FreeBSD Handbook

Resumo

Bem vindo ao FreeBSD! Este manual cobre a instalação e o uso diário do *FreeBSD 12.1-RELEASE* e do *FreeBSD 11.4-RELEASE*. Este livro é o resultado do trabalho contínuo de muitas pessoas. Algumas seções podem estar desatualizadas. Os interessados em ajudar a atualizar e expandir este documento devem enviar e-mails para a lista de discussão do projeto de documentação do FreeBSD.

A última versão deste livro está disponível no site do FreeBSD. Versões anteriores podem ser obtidas em https://docs.FreeBSD.org/doc/. O livro pode ser baixado em uma variedade de formatos e opções de compressão do servidor FTP do FreeBSD ou de um dos inúmeros sites espelho. Cópias impressas podem ser adquiridas da FreeBSD Mall. As pesquisas podem ser realizadas no manual e em outros documentos na página de busca.

Índice

Prefácio . . . 10
Audiência Pretendida . . . 10
Mudanças desde a Terceira Edição . . . 10
Mudanças desde a Segunda Edição (2004) . . . 10
Mudanças desde a Primeira Edição (2001) . . . 11
Organização deste Livro . . . 12
Convenções utilizadas neste livro . . . 15
Agradecimentos . . . 16
I: Primeiros Passos . . . 17
1. Introdução . . . 18
1.1. Sinopse . . . 18
1.2. Bem vindo ao FreeBSD! . . . 18
1.3. Sobre o Projeto FreeBSD . . . 23
2. Instalando o FreeBSD . . . 28
2.1. Sinopse . . . 28
2.2. Requisitos mínimos de hardware . . . 29
2.3. Tarefas de Pré-instalação . . . 30
2.4. Iniciando a instalação . . . 33
2.5. Usando o bsdinstall . . . 37
2.6. Alocando o espaço em disco . . . 43
2.7. Fazendo o download dos arquivos de distribuição . . . 66
2.8. Contas, Time Zone, Serviços e Hardening . . . 69
2.9. Interfaces de Rede . . . 87
2.10. Solução de problemas . . . 98
2.11. Usando o Live CD . . . 99
3. Fundamentos do FreeBSD . . . 100
3.1. Sinopse . . . 100
3.2. Consoles e Terminais Virtuais . . . 100
3.3. Usuários e Gerenciamento Básico de Contas . . . 103
3.4. Permissões . . . 112
3.5. Estrutura de Diretórios . . . 117
3.6. Organização dos Discos . . . 119
3.7. Montando e Desmontando Sistemas de Arquivos . . . 125
3.8. Processos e Daemons . . . 128
3.9. Shells . . . 131
3.10. Editores de Texto . . . 134
3.11. Dispositivos e nós de dispositivos . . . 135
3.12. Páginas de Manual . . . 135

4. Instalando Aplicativos: Pacotes e Ports . . . 138
4.1. Sinopse . . . 138
4.2. Visão geral sobre a Instalação de Software . . . 138
4.3. Encontrando Software . . . 140
4.4. Usando o pkg para o gerenciamento de pacotes binários . . . 142
4.5. Usando a Coleção de Ports . . . 149
4.6. Compilando Pacotes com o Poudriere . . . 159
4.7. Considerações pós-instalação . . . 162
4.8. Lidando com ports quebrados . . . 162
5. O sistema X Window . . . 164
5.1. Sinopse . . . 164
5.2. Terminologia . . . 164
5.3. Instalando o Xorg . . . 166
5.4. Configuração do Xorg . . . 166
5.5. Usando fontes no Xorg . . . 175
5.6. O Gerenciador de Display X . . . 179
5.7. Ambientes de desktop . . . 181
5.8. Instalando o Compiz Fusion . . . 185
5.9. Solução de problemas . . . 188
II: Tarefas comuns . . . 194
6. Aplicações de Desktop . . . 195
6.1. Sinopse . . . 195
6.2. Navegadores . . . 195
6.3. Produtividade . . . 197
6.4. Visualizadores de Documentos . . . 200
6.5. Finanças . . . 203
7. Multimídia . . . 205
7.1. Sinopse . . . 205
7.2. Configurando a Placa de Som . . . 205
7.3. Áudio MP3 . . . 211
7.4. Reprodução de Vídeo . . . 213
7.5. Placas de TV . . . 219
7.6. MythTV . . . 221
7.7. Scanners de Imagem . . . 222
8. Configurando o kernel do FreeBSD . . . 227
8.1. Sinopse . . . 227
8.2. Por que compilar um kernel personalizado? . . . 227
8.3. Encontrando o hardware do sistema . . . 228
8.4. O Arquivo de Configuração . . . 229
8.5. Criando e Instalando um Kernel Customizado . . . 231
8.6. Se algo der errado . . . 232

9. Impressão . . . 234
9.1. Inicio Rápido . . . 234
9.2. Conexões de Impressora . . . 235
9.3. Linguagens de Descrição de Página Comuns . . . 236
9.4. Impressão Direta . . . 238
9.5. LPD (Daemon de impressora de linha) . . . 238
9.6. Outros sistemas de impressão . . . 248
10. Compatibilidade binária com o Linux® . . . 249
10.1. Sinopse . . . 249
10.2. Configurando a compatibilidade binária com o Linux™ . . . 249
10.3. Tópicos Avançados . . . 252
III: Administração do Sistema . . . 255
11. Configuração e Ajuste . . . 256
11.1. Sinopse . . . 256
11.2. Inicialização de Serviços . . . 256
11.3. Configurando o cron(8) . . . 257
11.4. Gerenciando Serviços no FreeBSD . . . 260
11.5. Configurando Placas de Interface de Rede . . . 262
11.6. Hosts Virtuais . . . 269
11.7. Configurando o log do sistema . . . 270
11.8. Arquivos de Configuração . . . 277
11.9. Efetuando ajustes com o sysctl(8) . . . 279
11.10. Otimização de Discos . . . 281
11.11. Ajustando os Limites do Kernel . . . 285
11.12. Adicionando Espaço de Swap . . . 288
11.13. Gerenciamento de energia e recursos . . . 290
12. O processo de inicialização do FreeBSD . . . 297
12.1. Sinopse . . . 297
12.2. Processo de Inicialização do FreeBSD . . . 297
12.3. Configurando telas iniciais de inicialização . . . 304
12.4. Sugestões de dispositivos . . . 305
12.5. Sequência de Desligamento . . . 306
13. Segurança . . . 307
13.1. Sinopse . . . 307
13.2. Introdução . . . 307
13.3. Senhas de Uso Unico . . . 315
13.4. TCP Wrapper . . . 319
13.5. Kerberos . . . 321
13.6. OpenSSL . . . 329
13.7. VPN Sobre IPsec . . . 332
13.8. OpenSSH . . . 339

13.9. Listas de Controle de Acesso . . . 345
13.10. Monitorando Problemas de Segurança de Terceiros . . . 347
13.11. Avisos de Segurança do FreeBSD . . . 348
13.12. Auditoria de Processo . . . 352
13.13. Limites de Recursos . . . 353
13.14. Administração Compartilhada com Sudo . . . 357
14. Jails . . . 361
14.1. Sinopse . . . 361
14.2. Termos Relacionados à Jails . . . 362
14.3. Criando e Controlando Jails . . . 363
14.4. Tuning e Administração . . . 365
14.5. Atualizando Múltiplas Jails . . . 367
14.6. Gerenciando Jails com o ezjail . . . 373
15. Controle de acesso obrigatório . . . 384
15.1. Sinopse . . . 384
15.2. Termos chave . . . 385
15.3. Entendendo os rótulos MAC . . . 386
15.4. Planejando a configuração de segurança . . . 390
15.5. Políticas MAC Disponíveis . . . 392
15.6. Bloqueio do Usuário . . . 400
15.7. Nagios em Jail MAC . . . 400
15.8. Solução de problemas do framework MAC . . . 404
16. Auditoria de Evento de Segurança . . . 406
16.1. Sinopse . . . 406
16.2. Termos chave . . . 407
16.3. Configuração de Auditoria . . . 407
16.4. Trabalhando com Trilhas de Auditoria . . . 412
17. Armazenamento . . . 415
17.1. Sinopse . . . 415
17.2. Adicionando Discos . . . 415
17.3. Redimensionando e Ampliando Discos . . . 416
17.4. Dispositivos de Armazenamento USB . . . 419
17.5. Criando e Usando Mídia em CD . . . 423
17.6. Criando e Usando Mídia de DVD . . . 428
17.7. Criando e Usando Disquetes . . . 434
17.8. Noções Básicas de Backup . . . 435
17.9. Discos de Memória . . . 439
17.10. Snapshots de Sistemas de Arquivos . . . 441
17.11. Cotas de Disco . . . 443
17.12. Criptografando Partições de Disco . . . 446
17.13. Criptografando Swap . . . 452

17.14. Alta Disponibilidade de Armazenamento (HAST) . . . 454
18. GEOM: Framework de Transformação de Disco Modular . . . 463
18.1. Sinopse . . . 463
18.2. RAID0 - Striping . . . 463
18.3. RAID1 - Espelhamento . . . 465
18.4. RAID3 - Distribuição em Nível de Byte com Paridade Dedicada . . . 475
18.5. Dispositivos RAID por Software . . . 477
18.6. GEOM Network Gate . . . 482
18.7. Rotulando Dispositivos de Disco . . . 483
18.8. Journaling UFS através do GEOM . . . 486
19. O sistema de arquivos Z (ZFS) . . . 488
19.1. O que torna o ZFS diferente . . . 488
19.2. Guia de Início Rápido . . . 489
19.3. Administração `zpool` . . . 495
19.4. Administração do `zfs` . . . 515
19.5. Administração Delegada . . . 536
19.6. Tópicos Avançados . . . 537
19.7. Recursos adicionais . . . 540
19.8. Recursos e terminologia do ZFS . . . 540
20. Outros Sistemas de Arquivos . . . 551
20.1. Sinopse . . . 551
20.2. Sistemas de arquivos do Linux™ . . . 551
21. Virtualização . . . 553
21.1. Sinopse . . . 553
21.2. FreeBSD como Sistema Operacional Convidado no Parallels para Mac OS™ X . . . 553
21.3. FreeBSD como sistema convidado no Virtual PC para Windows™ . . . 563
21.4. FreeBSD como Sistema Operacional Convidado no VMware Fusion para Mac OS™ . . . 572
21.5. FreeBSD como Sistema Operacional Convidado no VirtualBox™ . . . 583
21.6. FreeBSD como Host com VirtualBox™ . . . 585
21.7. FreeBSD como um Host bhyve . . . 588
21.8. FreeBSD como Host Xen™ . . . 594
22. Localização - Uso e Configuração do i18n/L10n . . . 601
22.1. Sinopse . . . 601
22.2. Usando Localização . . . 601
22.3. Encontrando Aplicações i18n . . . 608
22.4. Configuração de Localização para Idiomas Específicos . . . 608
23. Atualização e Upgrade do FreeBSD . . . 611
23.1. Sinopse . . . 611
23.2. Atualização do FreeBSD . . . 611
23.3. Atualizando o Conjunto de Documentação . . . 619
23.4. Acompanhando um ramo de desenvolvimento . . . 622

23.5. Atualizando o FreeBSD a partir do código fonte . . . 625
23.6. Atualização de várias máquinas . . . 631
24. DTrace . . . 633
24.1. Sinopse . . . 633
24.2. Diferenças de Implementação . . . 633
24.3. Ativando o Suporte do DTrace . . . 634
24.4. Usando o DTrace . . . 635
25. Modo de dispositivo USB/USB OTG . . . 638
25.1. Sinopse . . . 638
25.2. Portas Seriais Virtuais USB . . . 639
25.3. Interfaces de rede do modo de dispositivo USB . . . 641
25.4. Dispositivo de armazenamento virtual USB . . . 641
IV: Comunicação de rede . . . 644
26. Comunicações Seriais . . . 645
26.1. Sinopse . . . 645
26.2. Terminologia serial e hardware . . . 645
26.3. Terminais . . . 649
26.4. Serviço Dial-in . . . 653
26.5. Serviço de Dial-in . . . 657
26.6. Configurando o Console Serial . . . 660
27. PPP . . . 667
27.1. Sinopse . . . 667
27.2. Configurando o PPP . . . 667
27.3. Solução de problemas de conexões PPP . . . 675
27.4. Usando o PPP sobre Ethernet (PPPoE) . . . 679
27.5. Usando PPP sobre ATM (PPPoA) . . . 680
28. Correio Eletrônico . . . 684
28.1. Sinopse . . . 684
28.2. Componentes de Email . . . 684
28.3. Arquivos de Configuração do Sendmail . . . 686
28.4. Alterando o Mail Transfer Agent . . . 689
28.5. Solução de problemas . . . 691
28.6. Tópicos Avançados . . . 693
28.7. Configurando Apenas Envio . . . 695
28.8. Usando Email com uma Conexão Dialup . . . 696
28.9. Autenticação SMTP . . . 697
28.10. Mail User Agents . . . 699
28.11. Usando o fetchmail . . . 707
28.12. Usando o procmail . . . 708
29. Servidores de Rede . . . 710
29.1. Sinopse . . . 710

29.2. O super-servidor inetd . . . 710
29.3. Network File System (NFS) . . . 714
29.4. Sistema de Informação de Rede (NIS) . . . 719
29.5. Protocolo leve de acesso de diretório (LDAP) . . . 733
29.6. Protocolo de configuração dinâmica de hosts (DHCP) . . . 742
29.7. Sistema de Nomes de Domínio (DNS) . . . 746
29.8. Servidor HTTP Apache . . . 748
29.9. Protocolo de Transferência de Arquivos (FTP) . . . 755
29.10. Serviços de arquivos e impressão para clientes Microsoft™Windows™ Clients (Samba) . . . 757
29.11. Sincronização de Relógio com NTP . . . 759
29.12. Inicializador iSCSI e Configuração Alvo . . . 763
30. Firewalls . . . 769
30.1. Sinopse . . . 769
30.2. Conceitos de Firewall . . . 770
30.3. PF . . . 771
30.4. IPFW . . . 789
30.5. IPFILTER (IPF) . . . 804
30.6. Blacklistd . . . 817
31. Rede Avançada . . . 823
31.1. Sinopse . . . 823
31.2. Gateways e Rotas . . . 823
31.3. Rede sem fio . . . 829
31.4. USB Tethering . . . 850
31.5. Bluetooth . . . 851
31.6. Bridging . . . 860
31.7. Agregação de links e failover . . . 867
31.8. Operação Diskless com PXE . . . 872
31.9. IPv6 . . . 877
31.10. Protocolo Comum de Redundância de Endereços (CARP) . . . 882
31.11. VLANs . . . 885
V: Apêndices . . . 887
Apêndice A: Obtendo o FreeBSD . . . 888
A.1. CD and DVD Sets . . . 888
A.2. Sites de FTP . . . 888
A.3. Usando o Subversion . . . 895
A.4. Usando o rsync . . . 898
Apêndice B: Bibliografia . . . 900
B.1. Livros específicos para o FreeBSD . . . 900
B.2. Guias de usuários . . . 901
B.3. Guias de Administradores . . . 901

B.4. Guias de programadores . . . 901
B.5. Internals do sistema operacional . . . 902
B.6. Referências de segurança . . . 902
B.7. Referências de Hardware . . . 903
B.8. História do UNIX™ . . . 903
B.9. Periódicos, Jornais e Revistas . . . 904
Apêndice C: Recursos na Internet . . . 905
C.1. Websites . . . 905
C.2. Listas de Discussão . . . 905
C.3. Grupos de Notícias Usenet . . . 926
C.4. Espelhos Oficiais . . . 927
Apêndice D: Chaves OpenPGP . . . 930
D.1. Administradores . . . 930

Prefácio

Audiência Pretendida

O novato no FreeBSD descobrirá que a primeira seção deste livro guia o usuário através do processo de instalação do FreeBSD e gentilmente apresenta os conceitos e convenções que sustentam o UNIX™. Trabalhar através desta seção exige pouco mais do que o desejo de explorar, e a capacidade de incorporar novos conceitos à medida que eles são introduzidos.

Uma vez que você chegou até aqui, a segunda seção do Handbook, muito maior, é uma referência abrangente a todos os tópicos de interesse para administradores de sistemas FreeBSD. Alguns destes capítulos podem recomendar que você faça alguma leitura prévia, e isto é destacado na sinopse no início de cada capítulo.

Para uma lista de fontes adicionais de informação, por favor veja o Bibliografia.

Mudanças desde a Terceira Edição

A versão online atual do Handbook representa o esforço cumulativo de muitas centenas de contribuidores nos últimos 10 anos. A seguir estão algumas das mudanças significativas desde a publicação da terceira edição do volume em 2004:

- DTrace foi adicionado com informações sobre a poderosa ferramenta de análise de desempenho DTrace.
- Outros Sistemas de Arquivos foi adicionado com informações sobre sistemas de arquivos não-nativos no FreeBSD, como o ZFS da Sun™.
- Auditoria de Evento de Segurança foi adicionado para cobrir os novos recursos de auditoria no FreeBSD e explicar seu uso.
- Virtualização foi adicionado com informações sobre a instalação do FreeBSD em ambientes virtualizados.
- Instalando o FreeBSD foi adicionado para cobrir a instalação do FreeBSD usando o novo utilitário de instalação, bsdinstall.

Mudanças desde a Segunda Edição (2004)

A terceira edição foi o culminar de mais de dois anos de trabalho pelos membros dedicados do Projeto de Documentação do FreeBSD. A edição impressa cresceu a tal tamanho que foi necessário publicar como dois volumes separados. A seguir estão as principais mudanças nesta nova edição:

- Configuração e Ajuste foi expandido com novas informações sobre o gerenciamento de recursos e energia da ACPI, o utilitário de sistema `cron` e mais opções para ajuste do kernel.
- Segurança foi expandido com novas informações sobre redes virtuais privadas (VPNs), listas de controle de acesso (ACLs) do sistema de arquivos e avisos de segurança.
- Controle de acesso obrigatório é um novo capítulo desta edição. Ele explica o que é MAC e como esse mecanismo pode ser usado para proteger um sistema FreeBSD.

- Armazenamento foi expandido com novas informações sobre dispositivos de armazenamento USB, snapshots do sistema de arquivos, cotas do sistema de arquivos, arquivos e sistemas de arquivos com suporte de rede e partições de disco criptografadas.
- Uma seção de solução de problemas foi adicionada ao PPP.
- Correio Eletrônico foi expandido com novas informações sobre o uso de agentes de transporte alternativos, autenticação SMTP, UUCP, fetchmail, procmail e outros tópicos avançados.
- Servidores de Rede é novidade nesta edição. Este capítulo inclui informações sobre a configuração do Servidor HTTP Apache, ftpd e a configuração de um servidor para clientes Microsoft™Windows™ com Samba. Algumas seções do Rede Avançada foram movidas para cá para melhorar a apresentação.
- Rede Avançada foi expandido com novas informações sobre o uso de dispositivos Bluetooth™ com o FreeBSD, configuração de redes sem fio e redes ATM (Asynchronous Transfer Mode).
- Um glossário foi adicionado para fornecer um local central para as definições de termos técnicos utilizados ao longo do livro.
- Uma série de melhorias estéticas foram feitas nas tabelas e figuras ao longo do livro.

Mudanças desde a Primeira Edição (2001)

A segunda edição foi o culminar de mais de dois anos de trabalho pelos membros dedicados do Projeto de Documentação do FreeBSD. A seguir, as principais mudanças nesta edição:

- Um índice completo foi adicionado.
- Todas as figuras ASCII foram substituídas por diagramas gráficos.
- Uma sinopse padrão foi adicionada a cada capítulo para fornecer um resumo rápido de quais informações o capítulo contém e o que se espera que o leitor saiba.
- O conteúdo foi logicamente reorganizado em três partes: "Introdução", "Administração do Sistema" e "Apêndices".
- Fundamentos do FreeBSD foi expandido para conter informações adicionais sobre processos, daemons e sinais.
- Instalando Aplicativos. Pacotes e Ports foi expandido para conter informações adicionais sobre o gerenciamento de pacotes binários.
- O sistema X Window foi completamente reescrito com ênfase no uso de tecnologias de desktop modernas como KDE e GNOME sobre o XFree86™ 4.X.
- O processo de inicialização do FreeBSD foi expandido.
- Armazenamento foi escrito a partir do que costumava ser dois capítulos separados em "Discos" e "Backups". Sentimos que os tópicos são mais fáceis de compreender quando apresentados como um único capítulo. Uma seção sobre RAID (hardware e software) também foi adicionada.
- Comunicações Seriais foi completamente reorganizado e atualizado para o FreeBSD 4.X/5.X.
- PPP foi substancialmente atualizado.
- Muitas novas seções foram adicionadas ao Rede Avançada.
- Correio Eletrônico foi expandido para incluir mais informações sobre a configuração do

sendmail.

- Compatibilidade binária com o Linux® foi expandido para incluir informações sobre como instalar o Oracle™ e o SAP™R/3™.
- Os novos tópicos a seguir são abordados nesta segunda edição:
  - Configuração e Ajuste.
  - Multimídia.

Organização deste Livro

Este livro é dividido em cinco seções logicamente distintas. A primeira seção, *Introdução*, cobre a instalação e o uso básico do FreeBSD. Espera-se que o leitor siga estes capítulos em sequência, possivelmente ignorando capítulos que abordam tópicos familiares. A segunda seção, *Tarefas Comuns*, cobre alguns dos recursos mais usados do FreeBSD. Esta seção e todas as seções subsequentes podem ser lidas fora de ordem. Cada capítulo começa com uma sinopse sucinta que descreve o que o capítulo cobre e o que se espera que o leitor já conheça. Isso permite que o leitor casual pule para encontrar capítulos de interesse. A terceira seção, *Administração do Sistema*, cobre tópicos de administração. A quarta seção, *Comunicação de Rede*, aborda tópicos sobre redes e servidores. A quinta seção contém apêndices de informações de referência.

Introdução

Introduz o FreeBSD para um novo usuário. Descreve a história do projeto FreeBSD, seus objetivos e modelo de desenvolvimento.

Instalando o FreeBSD

Guia o usuário durante todo o processo de instalação do FreeBSD 9.*x* usando o bsdinstall.

Fundamentos do FreeBSD

Cobre os comandos básicos e a funcionalidade do sistema operacional FreeBSD. Se você está familiarizado com Linux™ ou outro tipo de UNIX™, provavelmente você pode pular este capítulo.

Instalando Aplicativos. Pacotes e Ports

Cobre a instalação de softwares de terceiros com a inovadora "Coleção de Ports" do FreeBSD, e com pacotes binários tradicionais.

O sistema X Window

Descreve o Sistema X Window em geral e usa o X11 no FreeBSD em particular. Também descreve ambientes comuns de desktop, como o KDE e GNOME.

Aplicações de Desktop

Lista alguns aplicativos comuns de desktop, como navegadores web e pacotes de produtividade, e descreve como instalá-los no FreeBSD.

Multimídia

Mostra como configurar o suporte a reprodução de som e vídeo para o seu sistema. Também descreve alguns exemplos de aplicativos de áudio e vídeo.

Configurando o kernel do FreeBSD

Explica o porque que você pode precisar configurar um novo kernel e fornece instruções detalhadas para configurar, compilar e instalar um kernel personalizado.

Impressão

Descreve o gerenciamento de impressoras no FreeBSD, incluindo informações sobre páginas de banner, contabilidade de impressoras e configuração inicial.

Compatibilidade binária com o Linux®

Descreve os recursos de compatibilidade Linux™ do FreeBSD. Também fornece instruções detalhadas de instalação para muitos aplicativos Linux™ populares, como o Oracle™ e o Mathematica™.

Configuração e Ajuste

Descreve os parâmetros disponíveis para os administradores do sistema ajustarem um sistema FreeBSD para um ótimo desempenho. Também descreve os vários arquivos de configuração usados no FreeBSD e onde encontrá-los.

O processo de inicialização do FreeBSD

Descreve o processo de inicialização do FreeBSD e explica como controlar este processo com opções de configuração.

Segurança

Descreve muitas ferramentas diferentes disponíveis para ajudar a manter seu sistema FreeBSD seguro, incluindo Kerberos, IPsec e OpenSSH.

Jails

Descreve o framework do jail e as suas vantagens sobre o chroot tradicional do FreeBSD.

Controle de acesso obrigatório

Explica o que é o Mandatory Access Control (MAC) e como esse mecanismo pode ser usado para proteger um sistema FreeBSD.

Auditoria de Evento de Segurança

Descreve o que é a Auditoria de Eventos do FreeBSD, como ela pode ser instalada, configurada e como as trilhas de auditoria podem ser inspecionadas ou monitoradas.

Armazenamento

Descreve como gerenciar mídias de armazenamento e sistemas de arquivos com o FreeBSD. Isto inclui discos físicos, matrizes RAID, mídias óticas e de fita, discos com suporte de memória e sistemas de arquivos de rede.

GEOM. Framework de Transformação de Disco Modular

Descreve o que é o framework GEOM do FreeBSD e como configurar os vários níveis suportados de RAID.

Outros Sistemas de Arquivos

Examina o suporte a sistemas de arquivos não-nativos no FreeBSD, como o Z File System da

Sun™.

Virtualização

Descreve o que os sistemas de virtualização oferecem e como eles podem ser usados com o FreeBSD.

Localização - Uso e Configuração do i18n/L10n

Descreve como usar o FreeBSD em outros idiomas além do inglês. Abrange a localização tanto em nível de sistema como em nível de aplicativo.

Atualização e Upgrade do FreeBSD

Explica as diferenças entre FreeBSD-STABLE, FreeBSD-CURRENT e FreeBSD releases. Descreve quais usuários se beneficiariam do uso de um sistema em desenvolvimento e descreve este processo. Cobre os métodos que os usuários podem usar para atualizar seu sistema para a última release de segurança.

DTrace

Descreve como configurar e usar a ferramenta DTrace da Sun™ no FreeBSD. O rastreamento dinâmico pode ajudar a localizar problemas de desempenho, realizando a análise do sistema em tempo real.

Comunicações Seriais

Explica como conectar terminais e modems ao seu sistema FreeBSD para conexões de discagem de entrada e de saída.

PPP

Descreve como usar o PPP para se conectar a sistemas remotos com o FreeBSD.

Correio Eletrônico

Explica os diferentes componentes de um servidor de e-mail e mergulha em tópicos simples de configuração do software mais popular de servidor de e-mails: o sendmail.

Servidores de Rede

Fornece instruções detalhadas e exemplos de arquivos de configuração para configurar sua máquina FreeBSD como um servidor de sistema de arquivos de rede, servidor de nome de domínio, servidor de sistema de informações de rede ou servidor de sincronização de horário.

Firewalls

Explica a filosofia por trás dos firewalls baseados em software e fornece informações detalhadas sobre a configuração dos diferentes firewalls disponíveis para o FreeBSD.

Rede Avançada

Descreve muitos tópicos de rede, incluindo o compartilhamento de uma conexão à Internet com outros computadores em sua LAN, tópicos avançados de roteamento, rede sem fio, Bluetooth™, ATM, IPv6 e muito mais.

Obtendo o FreeBSD

Lista diferentes fontes para obter a mídia de instalação do FreeBSD em CD-ROM ou DVD, bem

como diferentes sites na Internet que permitem que você baixe e instale o FreeBSD.

Bibliografia

Este livro aborda muitos assuntos diferentes que podem deixá-lo com a curiosidade de uma explicação mais detalhada. A bibliografia lista muitos livros excelentes que são referenciados no texto.

Recursos na Internet

Descreve os muitos fóruns disponíveis para usuários do FreeBSD postarem perguntas e se engajarem em conversas técnicas sobre o FreeBSD.

Chaves OpenPGP

Lista as fingerprints PGP de vários desenvolvedores do FreeBSD.

Convenções utilizadas neste livro

Para fornecer um texto consistente e fácil de ler, várias convenções são seguidas ao longo do livro.

Convenções Tipográficas

Itálico

Uma fonte *itálica* é usada para nomes de arquivos, URLs, textos enfatizados e o primeiro uso de termos técnicos.

`Monospace`

Uma fonte `monoespaçada` é usada para mensagens de erro, comandos, variáveis de ambiente, nomes de ports, nomes de host, nomes de usuários, nomes de grupos, nomes de dispositivos, variáveis e fragmentos de código.

Negrito

Uma fonte `negrita` é usada para aplicativos, comandos e chaves.

Entrada do Usuário

As teclas são mostradas em **negrito** para se destacar do restante do texto. As combinações de teclas que devem ser digitadas simultaneamente são mostradas com + entre as teclas, como:

`Ctrl` + `Alt` + `Del`

Isso significa que o usuário deve digitar as teclas `Ctrl`, `Alt` e `Del` ao mesmo tempo.

As teclas que devem ser digitadas em sequência serão separadas por vírgulas, por exemplo:

`Ctrl` + `X`, `Ctrl` + `S`

Significaria que o usuário deve digitar as teclas `Ctrl` e `X` simultaneamente e, em seguida, digitar as teclas `Ctrl` e `S` simultaneamente.

Exemplos

Exemplos começando com C:\> indicam um comando MS-DOS™. Salvo indicação em contrário, estes comandos podem ser executados a partir de uma janela de "Prompt de Comando" em um ambiente Microsoft™Windows™.

```
E:\> tools\fdimage floppies\kern.flp A:
```

Exemplos começando com `#` indicam um comando que deve ser executado como superusuário no FreeBSD. Você pode logar como `root` para digitar o comando, ou logar como sua conta normal e usar o comando su(1) para obter privilégios de superusuário.

```
# dd if=kern.flp of=/dev/fd0
```

Exemplos começando com `%` indicam um comando que deve ser chamado a partir de uma conta de usuário normal. Salvo indicação em contrário, a sintaxe C-shell é usada para definir variáveis de ambiente e outros comandos do shell.

```
% top
```

Agradecimentos

O livro que você está segurando representa os esforços de muitas centenas de pessoas em todo o mundo. Não importa se eles enviaram correções para erros de digitação ou submeteram capítulos completos, todas as contribuições foram úteis.

Várias empresas têm apoiado o desenvolvimento deste documento, pagando aos autores para trabalhar em tempo integral, pagando pela publicação, etc. Em particular, a BSDi (posteriormente adquirida pela Wind River Systems) pagou membros do Projeto de Documentação do FreeBSD para trabalhar na melhoria deste livro em tempo integral, levando à publicação da primeira edição impressa em março de 2000 (ISBN 1-57176-241-8). A Wind River Systems pagou vários autores adicionais para fazer uma série de melhorias na infraestrutura de impressão e adicionar capítulos adicionais ao texto. Este trabalho culminou com a publicação da segunda edição impressa em novembro de 2001 (ISBN 1-57176-303-1). Em 2003-2004, a FreeBSD Mall, Inc. pagou a vários contribuidores para melhorar o Handbook em preparação para a terceira edição impressa.

Parte I: Primeiros Passos

Esta parte do handbook é destinada aos usuários e administradores que são novos no FreeBSD. Estes capítulos:

- Apresentam o FreeBSD.
- Guiam os leitores através do processo de instalação.
- Ensinam conceitos básicos e fundamentais do UNIX™.
- Mostram como instalar a grande variedade de aplicativos de terceiros disponíveis para o FreeBSD.
- Apresenta o X, o sistema de janelas UNIX™ e detalha como configurar um ambiente de desktop para tornar os usuários mais produtivos.

O número de referências a tópicos futuros no texto foi mantido no mínimo, para que uma seção possa ser lida do começo ao fim com o mínimo de avanço desnecessário de páginas.

Capítulo 1. Introdução

1.1. Sinopse

Obrigado pelo seu interesse no FreeBSD! O capítulo seguinte cobre vários aspectos do Projeto FreeBSD, como seu histórico, objetivos, modelo de desenvolvimento e assim por diante.

Depois de ler este capítulo, você saberá:

- Como o FreeBSD se relaciona com outros sistemas operacionais de computadores.
- A história do projeto FreeBSD.
- Os objetivos do projeto FreeBSD.
- O básico do modelo de desenvolvimento de código aberto do FreeBSD.
- E claro: de onde o nome "FreeBSD" vem.

1.2. Bem vindo ao FreeBSD!

O FreeBSD é um Sistema Operacional de código aberto nos padrões Unix-Like para computadores de arquitetura x86 (32 and 64 bits), ARM™, AArch64, RISC-V™, MIPS™, POWER™, PowerPC™, and Sun UltraSPARC™. Ele fornece todos os recursos que são considerados comuns hoje em dia, como multitarefa preemptiva, proteção de memória, memória virtual, recursos para múltiplos usuários, suporte a SMP, todas as ferramentas de desenvolvimento de código aberto para diferentes linguagens e estruturas e recursos de área de trabalho centralizados no Sistema X Window, KDE ou GNOME. Seus pontos fortes são:

- *Licença Liberal Open Source*, que concede a você o direito de modificar e estender livremente seu código-fonte e incorporá-lo em projetos Open Source e produtos fechados, sem impor restrições típicas às licenças copyleft, bem como evita potenciais problemas de incompatibilidade de licença.
- *Rede TCP/IP forte* - O FreeBSD implementa protocolos padrões da indústria com desempenho e escalabilidade crescentes. Isso faz com que seja uma boa combinação tanto em funções de servidor quanto de roteamento/firewall - e, de fato, muitas empresas e fornecedores o utilizam precisamente para essa finalidade.
- *Suporte totalmente integrado ao OpenZFS*, incluindo root-on-ZFS, ZFS Boot Environments, gerenciamento de falhas, delegação administrativa, suporte a jails, documentação específica ao FreeBSD e suporte ao instalador do sistema.
- *Extensivos recursos de segurança*, do Mandatory Access Control ao Capsicum e mecanismos de sandbox.
- *Mais de 30 mil pacotes pré-compilados* para todas as arquiteturas suportadas, e a Coleção de Ports, que facilita a compilação de seus próprios pacotes personalizados.
- *Documentação* - além do Handbook e livros de diferentes autores que cobrem tópicos que vão da administração do sistema aos internals do kernel, há também as páginas man(), não apenas para daemons do userspace, utilitários e arquivos de configuração, mas também para APIs do

driver do kernel (seção 9) e drivers individuais (seção 4).

- *Estrutura de repositório simples e consistente e sistema de compilação* - O FreeBSD usa um único repositório para todos os seus componentes, tanto para o kernel quanto para o userspace. Isso, juntamente com um sistema de compilação unificado, fácil de personalizar e um processo de desenvolvimento bem pensado, facilita a integração do FreeBSD com a infraestrutura de compilação do seu próprio produto.
- *Mantem-se fiel à filosofia do Unix*, preferindo heterogeneidade ao invés de deamons monolíticos "all in one" com comportamento codificado (hardcoded).
- *Compatibilidade binária* com o Linux, o que torna possível executar muitos binários do Linux sem a necessidade de virtualização.

O FreeBSD é baseado na release 4.4BSD-Lite do Computer Systems Research Group (CSRG) da Universidade da Califórnia em Berkeley, e mantém a tradição distinta do desenvolvimento de sistemas BSD. Além do bom trabalho fornecido pelo CSRG, o Projeto FreeBSD colocou milhares de horas-homem para estender a funcionalidade e ajustar o sistema para o máximo desempenho e confiabilidade em situações de carga reais. O FreeBSD oferece desempenho e confiabilidade a altura de outras ofertas de código aberto e comerciais, combinadas com recursos de ponta não disponíveis em nenhum outro lugar.

1.2.1. O que o FreeBSD Pode Fazer?

As aplicações para as quais o FreeBSD pode ser colocado são verdadeiramente limitadas apenas pela sua própria imaginação. Do desenvolvimento de software à automação de fábrica, do controle de estoque à correção de azimute de antenas de satélite remotas; Se isso puder ser feito com um produto comercial UNIX™, é mais do que provável que você também possa fazê-lo com o FreeBSD! O FreeBSD também se beneficia significativamente de milhares de aplicativos de alta qualidade desenvolvidos por centros de pesquisa e universidades em todo o mundo, muitas vezes disponíveis com pouco ou nenhum custo.

Como o código-fonte do FreeBSD está disponível gratuitamente, o sistema também pode ser customizado em um grau quase inédito para aplicações ou projetos especiais, e de maneiras que geralmente não são possíveis com a maioria do sistemas operacionais dos principais fornecedores comerciais. Aqui está apenas uma amostra de algumas das aplicações em que as pessoas estão atualmente usando o FreeBSD:

- *Serviços de Internet:* A robusta rede TCP/IP incorporada ao FreeBSD torna-o uma plataforma ideal para uma variedade de serviços de Internet, tais como:
  - Servidores WEB
  - Roteamento IPv4 e IPv6
  - Firewalls e Gateways NAT ("IP masquerading")
  - Servidores FTP
  - Servidores de Email
  - E mais...
- *Educação:* Você é estudante de ciências da computação ou de engenharia relacionada? Não há melhor maneira de aprender sobre sistemas operacionais, arquitetura de computadores e redes

do que colocar as mãos no sistema, uma experiência que o FreeBSD pode oferecer. Os vários pacotes CAD, matemáticos e de design gráfico disponíveis gratuitamente também o tornam altamente útil para aqueles cujo principal interesse em um computador é fazer com que *outro* trabalho seja feito!

- *Pesquisa:* Com o código-fonte de todo o sistema disponível, o FreeBSD é uma excelente plataforma para pesquisa em sistemas operacionais, assim como em outros ramos da ciência da computação. A natureza livremente disponível do FreeBSD também possibilita que grupos remotos colaborem em ideias ou desenvolvimento compartilhado sem ter que se preocupar com acordos de licenciamento especiais ou limitações sobre o que pode ser discutido em fóruns abertos.
- *Rede:* Precisa de um novo roteador? Um servidor de nomes (DNS)? Um firewall para manter as pessoas fora de sua rede interna? O FreeBSD pode facilmente transformar esse PC não utilizado que está encostado em algum canto em um roteador avançado com recursos sofisticados de filtragem de pacotes.
- *Embarcado:* O FreeBSD é uma excelente plataforma para construir sistemas embarcados. Com suporte para plataformas ARM™, MIPS™ e PowerPC™, juntamente com uma pilha de rede robusta, recursos de ponta e a permissiva Licença BSD o FreeBSD é uma excelente base para a criação de roteadores embarcados, firewalls e outros dispositivos.
- *Desktop:* O FreeBSD é uma ótima opção para uma solução de desktop barata usando o servidor X11 disponível gratuitamente. O FreeBSD oferece várias opções de ambientes de desktop de código aberto, incluindo as interfaces gráficas de usuário padrão do GNOME e do KDE. O FreeBSD pode até inicializar "diskless" a partir de um servidor central, tornando as estações de trabalho individuais ainda mais baratas e fáceis de administrar.
- *Desenvolvimento de Software:* O sistema básico do FreeBSD vem com um conjunto completo de ferramentas de desenvolvimento, incluindo um completo compilador e depurador C/C++ . O suporte para muitas outras linguagens também está disponível por meio da coleção de ports e dos pacotes.

O FreeBSD está disponível para download gratuito, ou pode ser obtido em CD-ROM ou DVD. Por favor, consulte Obtendo o FreeBSD para maiores informações sobre como obter o FreeBSD.

1.2.2. Quem Usa o FreeBSD?

O FreeBSD é conhecido por seus recursos de serviço web - sites que rodam no FreeBSD incluem Hacker News, Netcraft, NetEase, Netflix, Sina, Sony Japan, Rambler, Yahoo!, e Yandex.

Os recursos avançados do FreeBSD, a segurança comprovada, o ciclo de release previsível e a licença permissiva levaram à sua utilização como plataforma para a construção de muitos appliances, dispositivos e produtos tanto comerciais quanto de código aberto. Muitas das maiores empresas de TI do mundo usam o FreeBSD:

- Apache - A Apache Software Foundation executa a maior parte de sua infraestrutura voltada para o público, incluindo possivelmente um dos maiores repositórios SVN do mundo, com mais de 1.4 milhões de commits, no FreeBSD.
- Apple - OS X utiliza muito do FreeBSD na sua pilha de rede, no seu sistema de arquivos virtuais e em muitos componentes userland. O Apple iOS também contém elementos emprestados do

FreeBSD.

- Cisco - Os appliances de segurança de rede e anti-spam IronPort executam um kernel modificado do FreeBSD.
- Citrix - A linha NetScaler de dispositivos de segurança fornece balanceamento de carga nas camadas 4-7, cache de conteúdo, firewall de aplicativos, VPN segura e acesso móvel à rede em nuvem, juntamente com o poder de um shell do FreeBSD.
- Dell EMC Isilon - Os dispositivos de armazenamento corporativo da Isilon são baseados no FreeBSD. A licença extremamente liberal do FreeBSD permitiu que a Isilon integrasse sua propriedade intelectual ao kernel e se concentrasse em construir seu produto ao invés de um sistema operacional.
- Quest KACE - Os appliances de gerenciamento de sistemas KACE executam o FreeBSD devido à sua confiabilidade, escalabilidade e a comunidade que apoia seu desenvolvimento contínuo.
- iXsystems - A linha TrueNAS de dispositivos de armazenamento unificado é baseada no FreeBSD. Além de seus produtos comerciais, a iXsystems também gerencia o desenvolvimento dos projetos de código aberto TrueOS e FreeNAS.
- Juniper - O sistema operacional JunOS que roda em todos os equipamentos de rede da Juniper (incluindo roteadores, switches, firewalls e dispositivos de rede) é baseado no FreeBSD. A Juniper é um dos muitos fornecedores que mostra a relação simbiótica entre o projeto e os fornecedores de produtos comerciais. Melhorias geradas na Juniper são enviadas para o FreeBSD para reduzir a complexidade de integrar novos recursos do FreeBSD ao JunOS no futuro.
- McAfee - O SecurOS, a base dos produtos de firewall corporativo da McAfee, incluindo o Sidewinder, é baseado no FreeBSD.
- NetApp - A linha de dispositivos de armazenamento Data ONTAP GX é baseada no FreeBSD. Além disso, a NetApp contribuiu com muitos recursos, incluindo o novo hipervisor licenciado pelo BSD, bhyve.
- Netflix - O appliance OpenConnect que a Netflix usa para transmitir filmes para seus clientes é baseado no FreeBSD. A Netflix fez extensas contribuições para a base de código e trabalha para manter um delta zero a partir do FreeBSD mainline. Os appliances Netflix OpenConnect são responsáveis por entregar mais de 32% de todo o tráfego de Internet na América do Norte.
- Sandvine - A Sandvine usa o FreeBSD como base de suas plataformas de processamento de rede em tempo real de alto desempenho que compõem seus produtos inteligentes de controle de política de rede.
- Sony - O console de videogame PlayStation 4 executa uma versão modificada do FreeBSD.
- Sophos - O produto Sophos Email Appliance é baseado em uma versão modificada (hardened) do FreeBSD e varre as mensagens de entrada em busca por spam e vírus, ao mesmo tempo em que monitora as mensagens de saída quanto a malware, bem como a perda acidental de informações confidenciais.
- Spectra Logic - A linha nTier de dispositivos de armazenamento de dados de arquivamento executa o FreeBSD e o OpenZFS.
- Stormshield - Os dispositivos Stormshield Network Security são baseados em uma versão modificada do FreeBSD. A licença BSD permite que eles integrem sua própria propriedade

intelectual ao sistema enquanto retornam uma grande quantidade de desenvolvimento interessante para a comunidade.

- The Weather Channel - O appliance IntelliStar que é instalado na central de cada provedor de cabo local e é responsável por injetar previsões meteorológicas locais na programação da rede de TV a cabo, executa o FreeBSD.
- Verisign - A Verisign é responsável por operar os registros de domínio raiz .com e .net, bem como a infra-estrutura de DNS que a acompanha. Eles contam com diversos sistemas operacionais de rede, incluindo o FreeBSD, para garantir que não haja um ponto comum de falha em sua infraestrutura.
- Voxer - A Voxer suporta sua plataforma de mensagem de voz móvel com o ZFS no FreeBSD. A Voxer mudou de um derivativo do Solaris para o FreeBSD por causa da sua documentação superior, comunidade maior e mais ativa e ao ambiente mais favorável ao desenvolvedor. Além de recursos críticos como o ZFS e o DTrace, o FreeBSD também oferece suporte a TRIM no ZFS.
- Fudo Security - O dispositivo de segurança FUDO permite que as empresas monitorem, controlem, registrem e façam auditoria de contratados e administradores que trabalham em seus sistemas. Baseado em todos os melhores recursos de segurança do FreeBSD, incluindo ZFS, GELI, Capsicum, HAST e auditdistd.

O FreeBSD também gerou vários projetos de código aberto relacionados:

- BSD Router - Um substituto baseado em FreeBSD para grandes roteadores corporativos projetados para rodar em hardware PC padrão.
- FreeNAS - Um FreeBSD personalizado projetado para ser usado como um dispositivo de servidor de arquivos de rede. Fornece uma interface web baseada em Python para simplificar o gerenciamento dos sistemas de arquivos UFS e ZFS. Inclui suporte para NFS, SMB/CIFS, AFP, FTP e iSCSI. Inclui um sistema extensível de plugins baseado em jails do FreeBSD.
- GhostBSD - é derivado do FreeBSD, usa o ambiente GTK para fornecer uma aparência bonita e uma experiência confortável na moderna plataforma BSD, oferecendo um ambiente de trabalho natural e nativo UNIX™.
- mfsBSD - Um kit de ferramentas para compilar uma imagem do sistema FreeBSD que roda inteiramente da memória.
- NAS4Free - Uma distribuição de servidor de arquivos baseada no FreeBSD com uma interface web PHP.
- OPNSense - OPNsense é um firewall e uma plataforma de roteamento open source, baseado em FreeBSD, fácil-de-usar e fácil-de-compilar. O OPNsense inclui a maioria dos recursos disponíveis em firewalls comerciais caros e, em muitos casos, muito mais. Ele traz o rico conjunto de recursos de ofertas comerciais com os benefícios de códigos fonte abertos e verificáveis.
- TrueOS - O TrueOS é baseado na lendária segurança e estabilidade do FreeBSD. O TrueOS segue o FreeBSD-CURRENT, com os drivers, atualizações de segurança e pacotes mais recentes disponíveis.
- FuryBSD - é um desktop FreeBSD de código aberto novinho em folha. O FuryBSD presta homenagem aos projetos de BSD de desktop do passado, como PC-BSD e TrueOS com sua interface gráfica e adiciona ferramentas adicionais, como uma imagem live USB/DVD híbrida. O FuryBSD é totalmente gratuito para uso e distribuído sob a licença BSD.

- MidnightBSD - é um sistema operacional derivado do FreeBSD desenvolvido com usuários de desktop em mente. Inclui todo o software que você esperaria para suas tarefas diárias: email, navegação web, processamento de texto, jogos e muito mais.
- pfSense - Uma distribuição de firewall baseada no FreeBSD com uma enorme variedade de recursos e amplo suporte a IPv6.
- ZRouter - Um firmware alternativo de código aberto para dispositivos embarcados baseado no FreeBSD. Projetado para substituir o firmware proprietário em roteadores prontos para uso.

Uma lista de depoimentos de empresas que baseiam seus produtos e serviços no FreeBSD pode ser encontrada no site da Fundação FreeBSD. A Wikipedia também mantém uma lista de produtos baseados no FreeBSD.

1.3. Sobre o Projeto FreeBSD

A seção a seguir fornece algumas informações básicas sobre o projeto, incluindo um breve histórico, metas do projeto e o modelo de desenvolvimento do projeto.

1.3.1. Uma Breve História do FreeBSD

O Projeto FreeBSD teve sua gênese no início de 1993, parcialmente como uma evolução natural do Unofficial 386BSD Patchkit por parte dos três últimos coordenadores: Nate Williams, Rod Grimes e Jordan Hubbard.

O objetivo original era produzir um snapshot intermediário do 386BSD, a fim de corrigir um grande número de problemas que o mecanismo do patchkit simplesmente não era capaz de resolver. O título inicial do projeto foi 386BSD 0.5 ou 386BSD Interim em referência a esse fato.

O 386BSD era o sistema operacional do Bill Jolitz, que havia até então sofrido bastante com quase um ano de negligência. Como o patchkit inchava cada vez mais desconfortavelmente a cada dia que passava, eles decidiram ajudar o Bill fornecendo este snapshot "limpo". Esses planos foram interrompidos quando, de repente, Bill Jolitz decidiu retirar sua sanção do projeto sem qualquer indicação clara do que seria feito em seu lugar.

O trio achou que a meta continuava valendo a pena, mesmo sem o apoio de Bill, e então adotaram o nome "FreeBSD" cunhado por David Greenman. Os objetivos iniciais foram definidos após consultar os usuários atuais do sistema e, uma vez que ficou claro que o projeto estava em vias de se tornar realidade, Jordan entrou em contato com a Walnut Creek CDROM com o objetivo de melhorar os canais de distribuição do FreeBSD para aqueles desafortunados sem acesso fácil à Internet. O Walnut Creek CDROM não apenas apoiou a ideia de distribuir o FreeBSD em CD, mas também chegou a fornecer ao projeto uma máquina para trabalhar e uma conexão rápida à Internet. Sem o grau de fé quase sem precedentes da Walnut Creek CDROM no que era, na época, um projeto completamente desconhecido, é bastante improvável que o FreeBSD tivesse chegado tão longe, tão rápido, como hoje.

A primeira distribuição em CD-ROM (e amplo pela rede) foi o FreeBSD 1.0, lançado em dezembro de 1993. Isto foi baseado na fita 4.3BSD-Lite ("Net/2") da U.C. Berkeley, com muitos componentes também fornecidos pelo 386BSD e pela Free Software Foundation. Foi um sucesso bastante razoável para uma primeira oferta, e e eles seguiram com o bem-sucedido FreeBSD 1.1 em maio de 1994.

Por esta altura, algumas nuvens de tempestade inesperadas formaram-se no horizonte, como a Novell e U.C. Berkeley resolveram seu longo processo judicial sobre o status legal da fita do Berkeley Net/2. Uma condição desse acordo foi a concessão da U.C. Berkeley de que grande parte do código Net/2 foi "onerado" e era de propriedade da Novell, que por sua vez o adquiriu da AT&T algum tempo antes. O que a Berkeley recebeu em troca foi a "bênção" da Novell de que o lançamento do 4.4BSD-Lite, quando finalmente fosse lançado, seria declarado livre e todos os atuais usuários do Net/2 seriam fortemente encorajados a mudar. Isso incluiu o FreeBSD, e foi dado ao projeto o tempo para interromper o envio de seu próprio produto baseado em Net/2 até o final de julho de 1994. Sob os termos desse acordo, o projeto recebeu um último lançamento antes do prazo final, sendo esse lançamento o FreeBSD 1.1.5.1.

O FreeBSD então começou a tarefa árdua de literalmente se reinventar de um conjunto completamente novo e incompleto de bits do 4.4BSD-Lite. As versões "Lite" foram leves em parte porque o CSRG da Berkeley removeu grandes pedaços de código necessários para realmente compilar um sistema inicializável (devido a vários requisitos legais) e o fato de que a port Intel do 4.4 era altamente incompleto. O projeto levou até novembro de 1994 para fazer essa transição, e em dezembro lançou o FreeBSD 2.0 para o mundo. Apesar de ainda ser um pouco mais difícil, o lançamento foi um sucesso significativo e foi seguido pela versão mais robusta e fácil de instalar o FreeBSD 2.0.5 em junho de 1995.

Desde aquela época, o FreeBSD fez uma série de lançamentos cada vez melhorando a estabilidade, a velocidade e o conjunto de recursos da versão anterior.

Por enquanto, os projetos de desenvolvimento de longo prazo continuam a ocorrer no ramo 10.X-CURRENT (trunk), e os snapshots de release 10.X são continuamente disponibilizados a partir do servidor de snapshots à medida que o trabalho progride.

1.3.2. Objetivos do Projeto FreeBSD

Os objetivos do Projeto FreeBSD são fornecer software que possa ser usado para qualquer propósito e sem amarras. Muitos de nós temos um investimento significativo no código (e projeto) e certamente não nos importaríamos com uma pequena compensação financeira de vez em quando, mas definitivamente não estamos preparados para insistir nisso. Acreditamos que a nossa primeira e principal "missão" é fornecer código a todos os participantes, e para qualquer finalidade, para que o código obtenha o maior uso possível e forneça o maior benefício possível. Este é, acredito, um dos objetivos mais fundamentais do Software Livre e um dos que apoiamos entusiasticamente.

O código em nossa árvore de código-fonte que se enquadra na GNU General Public License (GPL) ou na Library General Public License (LGPL) vem com um pouco mais de amarras, embora pelo menos do lado do acesso imposto, em vez do oposto usual. Devido às complexidades adicionais que podem evoluir no uso comercial de software GPL, no entanto, preferimos software submetido sob licença BSD quando é uma opção razoável fazê-lo.

1.3.3. O Modelo de Desenvolvimento do FreeBSD

O desenvolvimento do FreeBSD é um processo muito aberto e flexível, sendo construído literalmente a partir das contribuições de milhares de pessoas ao redor do mundo, como pode ser visto na nossa lista de contribuidores. A infraestrutura de desenvolvimento do FreeBSD permite que milhares de colaboradores colaborem pela Internet. Estamos constantemente à procura de

novos desenvolvedores e ideias, e os interessados em se envolver mais estreitamente com o projeto precisam simplesmente entrar em contato conosco pelas lista de discussões técnicas do FreeBSD. A lista de discussão de anúncios do FreeBSD também está disponível para aqueles que desejam fazer com que outros usuários do FreeBSD conheçam as principais áreas de trabalho.

Coisas úteis para saber sobre o Projeto FreeBSD e seu processo de desenvolvimento, seja trabalhando independentemente ou em estreita cooperação:

Os repositórios SVN

Por vários anos, a árvore de código-fonte central do FreeBSD foi mantida pelo CVS (Concurrent Versions Systems), uma ferramenta de controle de código-fonte disponível gratuitamente. Em junho de 2008, o Projeto mudou para o SVN (Subversion). A troca foi considerada necessária, pois as limitações técnicas impostas pelo CVS estavam se tornando óbvias devido à rápida expansão da árvore de código-fonte e à quantidade de histórico já armazenada. Os repositórios do Projeto de Documentação e da Coleção de Ports também foram movidos do CVS para o SVN em maio de 2012 e julho de 2012, respectivamente. Por favor, consulte a seção Atualizando o código fonte para maiores informações sobre como obter o repositório `src/` do FreeBSD e Usando a Coleção de Ports para detalhes sobre como obter a coleção de ports do FreeBSD.

A lista de committers

Os *committers* são as pessoas que têm acesso de *escrita* na árvore do Subversion, e estão autorizados a fazer modificações no código fonte do FreeBSD (o termo "committer" vem de `commit`, o comando de controle de código-fonte que é usado para trazer novas mudanças para o repositório). Qualquer um pode enviar um relatório de bug para o Banco de Dados de Bugs. Antes de enviar um relatório de bug, as listas de discussão, canais de IRC ou fóruns do FreeBSD podem ser usados para ajudar a verificar se um problema é realmente um bug.

O FreeBSD core team

O *FreeBSD core team* seria equivalente a um conselho de diretores se o Projeto FreeBSD fosse uma empresa. A principal tarefa do core team é garantir que o projeto, como um todo, esteja saudável e seguindo na direção certa. Convidar desenvolvedores dedicados e responsáveis a ingressar em nosso grupo de committers é uma das funções do core team, assim como o recrutamento de novos membros do core team à medida que os outros saiam. O core team atual foi eleito a partir de um grupo de committers candidatos em junho de 2020. As eleições são realizadas a cada dois anos.



Como a maioria dos desenvolvedores, a maioria dos membros do core team também são voluntários quando se trata de desenvolvimento do FreeBSD e não se beneficiam financeiramente do projeto, então o "compromisso" também não deve ser interpretado erroneamente como significando "suporte garantido". A analogia do "quadro de diretores" da diretriz acima não é muito precisa, e pode ser mais apropriado dizer que estas são as pessoas que deram suas vidas em favor do FreeBSD contra o seu melhor julgamento!

Contribuidores externos

Por último, mas definitivamente não menos importante, o maior grupo de desenvolvedores são os próprios usuários que fornecem feedback e correções de bugs para nós em uma base quase constante. A principal maneira de manter contato com o desenvolvimento não-centralizado do

FreeBSD é inscrever-se nas listas de discussões técnicas sobre o FreeBSD onde essas coisas são discutidas. Veja Recursos na Internet para maiores informações sobre as várias listas de discussão do FreeBSD.

A Lista de Colaboradores do FreeBSD é extensa e crescente, então por que não se juntar a ela contribuindo com algo para o FreeBSD hoje?

Fornecer código não é a única maneira de contribuir para o projeto; para uma lista mais completa de coisas que precisam ser feitas, por favor consulte o web site do Projeto FreeBSD.

Em resumo, nosso modelo de desenvolvimento é organizado como um conjunto solto de círculos concêntricos. O modelo centralizado é projetado para a conveniência dos *usuários* do FreeBSD, que são providos com uma maneira fácil de rastrear uma base de código central, e não para manter potenciais colaboradores fora! Nosso desejo é apresentar um sistema operacional estável com um grande conjunto de aplicações coerentes que os usuários possam facilmente instalar e usar - este modelo funciona muito bem em realizar isso.

Tudo o que pedimos para aqueles que se juntarem a nós como desenvolvedores do FreeBSD é a mesma dedicação que o pessoal atual tem para o seu sucesso contínuo!

1.3.4. Programas de Terceiros

Além das distribuições básicas, o FreeBSD oferece uma coleção de software portada com milhares de programas comumente procurados. No momento da redação deste texto, havia mais de 24.000 ports! A lista de ports varia de servidores http, a jogos, linguagens, editores e quase tudo no meio. Toda a coleção de ports requer aproximadamente 500 MB. Para compilar um port, simplesmente mude para o diretório do programa que você deseja instalar, digite `make install` e deixe o sistema fazer o resto. A distribuição original completa para cada port que você cria é baixada dinamicamente, para que você precise apenas de espaço em disco suficiente para compilar os ports desejados. Quase todos os ports também são fornecidos como um pacote "pré-compilado", que pode ser instalado com um comando simples (`pkg install`) por aqueles que não desejam compilar seus próprios ports pelo código fonte. Maiores informações sobre pacotes e ports podem ser encontradas em Instalando Aplicativos. Pacotes e Ports.

1.3.5. Documentação Adicional

Todas as versões suportadas do FreeBSD fornecem uma opção no instalador para instalar documentação adicional em /usr/local/shar/doc/freebsd durante a configuração inicial do sistema. A documentação também pode ser instalada posteriormente, usando os pacotes descritos em Atualizando a documentação a partir do ports. Você pode ver os manuais instalados localmente com qualquer navegador compatível com HTML usando as seguintes URLs:

O Handbook do FreeBSD

/usr/local/shared/doc/freebsd/handbook/index.html

O FAQ do FreeBSD

/usr/local/shared/doc/freebsd/faq/index.html

Você também pode visualizar as cópias principais (e atualizadas com mais frequência) em

https://www.FreeBSD.org/.

Capítulo 2. Instalando o FreeBSD

2.1. Sinopse

Existem diversos modos diferentes de colocar o FreeBSD para rodar, dependendo do ambiente. São eles:

- Imagens de Máquinas Virtuais, para baixar e importar em um ambiente virtual da sua escolha. Elas podem ser baixadas da página de Download do FreeBSD. Existem imagens para KVM ("qcow2"), VMWare ("vmdk"), Hyper-V ("vhd") e imagens de dispositivos brutos (raw device) que são universalmente suportadas . Estas não são imagens de instalação, mas sim as instâncias pré-configuradas ("já instaladas"), prontas para executar e realizar tarefas de pós-instalação.
- Imagens de máquinas virtuais disponíveis no AWS Marketplace, no Microsoft Azure Marketplace, e na Plataforma Google Cloud, para executar em seus respectivos serviços de hospedagem. Para obter maiores informações sobre como implantar o FreeBSD no Azure, consulte o capítulo relevante na Documentação do Azure.
- Imagens de cartão SD, para sistemas embarcados, como Raspberry Pi ou BeagleBone Black. Eles podem ser baixados da página de Download do FreeBSD. Esses arquivos devem ser descompactados e gravados como uma imagem bruta para um cartão SD, a partir do qual a placa será inicializada.
- Imagens de instalação, para instalar o FreeBSD no disco rígido de um desktop padrão, laptop ou servidor.

O resto deste capítulo descreve o quarto caso, explicando como instalar o FreeBSD usando o programa de instalação baseado em texto chamado bsdinstall.

Em geral, as instruções de instalação neste capítulo foram escritas para as arquiteturas i386™ e AMD64. Onde aplicável, instruções específicas para outras plataformas serão listadas. Pode haver pequenas diferenças entre o instalador e o que é mostrado aqui, portanto, use este capítulo como um guia geral, e não como um conjunto de instruções literais.



Usuários que preferem instalar o FreeBSD usando uma instaldor gráfico talvez possam se interessar no FuryBSD, GhostBSD ou MidnightBSD.

Depois de ler este capítulo, você saberá:

- Quais os requisitos mínimos de hardware e as arquiteturas suportadas pelo FreeBSD.
- Como criar a mídia de instalação do FreeBSD.
- Como iniciar o bsdinstall.
- As perguntas que o bsdinstall fará, o que elas significam e como respondê-las.
- Como solucionar problemas de uma instalação com falha.
- Como acessar uma versão live do FreeBSD antes de se comprometer com uma instalação.

Antes de ler este capítulo, você deve:

- Ler a lista de hardware suportado que acompanha a versão do FreeBSD que será instalada e verificar se o hardware do sistema é suportado.

2.2. Requisitos mínimos de hardware

Os requisitos de hardware para instalar o FreeBSD variam por arquitetura. Arquiteturas de hardware e dispositivos suportados por uma release do FreeBSD estão listados na página Informação de Release do FreeBSD. A página de download do FreeBSD também tem recomendações para escolha a imagem correta para as diferentes arquiteturas.

Uma instalação do FreeBSD requer um mínimo de 96 MB de RAM e 1,5 GB de espaço livre no disco rígido. No entanto, essas pequenas quantidades de memória e espaço em disco são realmente adequadas apenas para aplicativos personalizados, como dispositivos embarcados. Os sistemas de desktop de uso geral precisam de mais recursos. De 2 a 4 GB de RAM e pelo menos 8 GB de espaço no disco rígido é um bom ponto de partida.

Estes são os requisitos do processador para cada arquitetura:

amd64

Esse é o tipo de processador de desktop e laptop mais comum, usado na maioria dos sistemas modernos. A Intel ™ chama ele de Intel64. Outros fabricantes às vezes o chamam de x86-64.

Exemplos de processadores compatíveis com AMD64 incluem: AMD Athlon™ 64, AMD Opteron™, multi-core Intel™Xeon™ e processadores Intel™Core™ 2 e posteriores.

i386

Desktops e laptops mais antigos geralmente usam essa arquitetura x86 de 32 bits.

Quase todos os processadores compatíveis com i386 com uma unidade de ponto flutuante são suportados. Todos os processadores Intel™ 486 ou superior são suportados.

O FreeBSD irá aproveitar o suporte a Extensões de Endereços Físicos (PAE) em CPUs com este recurso. Um kernel com o recurso PAE ativado detectará memória acima de 4 GB e permitirá que ela seja usada pelo sistema. No entanto, o uso do PAE coloca restrições em drivers de dispositivos e outros recursos do FreeBSD.

powerpc

Todos os sistemas New World ROMApple™Mac™ com USB incorporados são suportados. O SMP é suportado em máquinas com vários CPUs.

Um kernel de 32 bits só pode usar os primeiros 2 GB de RAM.

sparc64

Os sistemas suportados pelo FreeBSD/sparc64 estão listados no Projeto FreeBSD/sparc64.

O SMP é suportado em todos os sistemas com mais de 1 processador. Um disco dedicado é necessário, pois não é possível compartilhar um disco com outro sistema operacional neste momento.

2.3. Tarefas de Pré-instalação

Uma vez determinado que o sistema atende aos requisitos mínimos de hardware para instalar o FreeBSD, o arquivo de instalação deve ser baixado e a mídia de instalação preparada. Antes de fazer isso, verifique se o sistema está pronto para uma instalação, verificando os itens nesta lista de controle:

1. Faça backup dos dados importantes

   Antes de instalar qualquer sistema operacional, *sempre* faça backup de todos os dados importantes primeiro. Não armazene o backup no sistema que está sendo instalado. Em vez disso, salve os dados em um disco removível, como uma unidade USB, outro sistema na rede ou um serviço de backup online. Teste o backup antes de iniciar a instalação para garantir que ele contenha todos os arquivos necessários. Depois que o instalador formatar o disco do sistema, todos os dados armazenados nesse disco serão perdidos.

2. Decida onde instalar o FreeBSD

   Se o FreeBSD for o único sistema operacional instalado, esta etapa pode ser ignorada. Mas se o FreeBSD compartilhar o disco com outro sistema operacional, decida qual disco ou partição será usado para o FreeBSD.

   Nas arquiteturas i386 e amd64, os discos podem ser divididos em várias partições usando um dos dois esquemas de particionamento. Um *registro de inicialização mestre* tradicional (MBR) contém uma tabela de partição que define até quatro *partições primárias*. Por razões históricas, o FreeBSD chama essas partições primárias de *slices*. Uma dessas partições primárias pode ser transformada em uma *partição estendida* contendo várias *partições lógicas*. A *Tabela de Partição GUID* (GPT) é um método mais novo e mais simples de particionar um disco. Implementações comuns de GPT permitem até 128 partições por disco, eliminando a necessidade de partições lógicas.

   O boot loader do FreeBSD requer uma partição primária ou GPT. Se todas as partições primárias ou GPT já estiverem em uso, uma deve ser liberada para o FreeBSD. Para criar uma partição sem excluir dados existentes, use uma ferramenta de redimensionamento de partição para reduzir uma partição existente e criar uma nova partição usando o espaço liberado.

   Uma variedade de ferramentas de redimensionamento de partições comerciais e gratuitas estão listadas em http://en.wikipedia.org/wiki/List_of_disk_partitioning_software. O GParted Live (http://gparted.sourceforge.net/livecd.php) é um live CD que inclui o editor de partições GParted. O GParted também está incluído em muitas outras distribuições live CD do Linux.

   Quando usados corretamente, os utilitários de encolhimento de disco podem criar espaço com segurança para criar uma nova partição. Como existe a possibilidade de selecionar a partição errada, sempre faça backup de todos os dados importantes e verifique a integridade do backup antes de modificar as partições do disco.

Partições de disco contendo diferentes sistemas operacionais tornam possível instalar vários sistemas operacionais em um computador. Uma alternativa é usar virtualização (Virtualização) o que permite que vários sistemas operacionais sejam executados ao mesmo tempo sem modificar nenhuma partição de disco.

3. Colete informações de rede

   Alguns métodos de instalação do FreeBSD requerem uma conexão de rede para baixar os arquivos de instalação. Após qualquer instalação, o instalador oferecerá a configuração das interfaces de rede do sistema.

   Se a rede tiver um servidor DHCP, ele poderá ser usado para fornecer configuração de rede automática. Se o DHCP não estiver disponível, as seguintes informações de rede para o sistema devem ser obtidas com o administrador de rede local ou com o provedor de serviços de Internet:

   a. Endereço IP
   b. Máscara de sub-rede
   c. Endereço do IP do gateway padrão
   d. Nome de domínio da rede
   e. Endereços IP dos servidores DNS da rede

4. Verifique a Errata do FreeBSD

   Embora o Projeto FreeBSD se esforce para garantir que cada versão do FreeBSD seja o mais estável possível, ocasionalmente, os bugs aparecem no processo. Em raras ocasiões, esses erros afetam o processo de instalação. A medida que esses problemas são descobertos e corrigidos, eles são anotados na Errata do FreeBSD (https://www.freebsd.org/releases/12.1R/errata/) no site do FreeBSD. Verifique a errata antes de instalar para certificar-se de que não existem problemas que possam afetar a instalação.

   Informações e erratas para todos os releases podem ser encontradas na seção de informações de release do site do FreeBSD (https://www.freebsd.org /releases/).

2.3.1. Prepare a mídia de instalação

O instalador do FreeBSD não é um aplicativo que pode ser executado dentro de outro sistema operacional. Em vez disso, baixe um arquivo de instalação do FreeBSD, grave-o na mídia associada ao seu tipo e tamanho (CD, DVD, ou USB), e inicialize o sistema para instalar a partir da mídia inserida.

Os arquivos de instalação do FreeBSD estão disponíveis em www.freebsd.org/where/. O nome de cada arquivo de instalação inclui a versão de Release do FreeBSD, a arquitetura e o tipo de arquivo. Por exemplo, para instalar o FreeBSD 12.1 em um sistema amd64 de um DVD, baixe o FreeBSD-12.1-RELEASE-amd64-dvd1.iso, grave este arquivo em um DVD, e inicialize o sistema com o DVD inserido.

Os arquivos de instalação estão disponíveis em vários formatos. Os formatos variam dependendo da arquitetura do computador e do tipo de mídia.

Arquivos de instalação adicionais são incluídos para computadores que inicializam com UEFI (Interface de Firmware Extensível Unificada). Os nomes desses arquivos incluem a string uefi.

Tipos de arquivo:

- `-bootonly.iso`: Este é o menor arquivo de instalação, pois contém apenas o instalador. É necessária uma conexão de Internet em funcionamento durante a instalação, pois o instalador fará o download dos arquivos necessários para concluir a instalação do FreeBSD. Este arquivo deve ser gravado em um CD usando um aplicativo de gravação CD.
- `-disc1.iso`: Este arquivo contém todos os arquivos necessários para instalar o FreeBSD, seu código-fonte e a coleção de ports. Ele deve ser gravado em um CD usando um aplicativo de gravação CD.
- `-dvd1.iso`: Este arquivo contém todos os arquivos necessários para instalar o FreeBSD, seu código-fonte e a coleção de ports. Ele também contém um conjunto de pacotes binários populares para instalar um gerenciador de janelas e alguns aplicativos para que um sistema completo possa ser instalado a partir da mídia sem a necessidade de uma conexão com a Internet. Este arquivo deve ser gravado em um DVD usando um aplicativo de gravação DVD.
- `-memstick.img`: Este arquivo contém todos os arquivos necessários para instalar o FreeBSD, seu código-fonte e a coleção de ports. Ele deve ser gravado em um pendrive USB usando as instruções abaixo.
- `-mini-memstick.img`: Como ` -bootonly.iso `, não inclui arquivos de instalação, mas faz o download conforme necessário. É necessária uma conexão de internet em funcionamento durante a instalação. Grave este arquivo para um pendrive USB como mostrado em Gravando um arquivo de imagem para um pendrive USB.

Depois de baixar o arquivo de imagem, baixe o CHECKSUM.SHA256 do mesmo diretório. Calcule o *checksum* para o arquivo de imagem. O FreeBSD fornece o sha256(1) para isso, usado como `sha256 imagefilename`. Outros sistemas operacionais possuem programas semelhantes.

Compare o checksum calculado com a mostrado em CHECKSUM.SHA256. Os checksum devem corresponder exatamente. Se os checksums não corresponderem, o arquivo de imagem está corrompido e deve ser baixado novamente.

2.3.1.1. Gravando um arquivo de imagem para um pendrive USB

O arquivo *. Img é uma *imagem* do conteúdo completo de um cartão de memória. Ele *não pode* ser copiado para o dispositivo de destino como um arquivo. Várias aplicações estão disponíveis para escrever o *. Img para um pendrive USB. Esta seção descreve dois destes utilitários.



Antes de continuar, faça backup de todos os dados importantes do pendrive USB. Este procedimento irá apagar todos os dados existentes no mesmo.

Procedure: Usando o `dd` para gravar a imagem



Este exemplo usa /dev/da0 como o dispositivo de destino em que a imagem será gravada. Seja *muito cuidadoso* para que o dispositivo correto seja usado, pois esse comando destruirá os dados existentes no dispositivo de destino especificado.

1. O utilitário de linha de comando dd(1) está disponível no BSD, no Linux™ e no Mac OS™. Para gravar a imagem usando o `dd`, insira o pendrive USB e determine o nome do dispositivo. Em seguida, especifique o nome do arquivo de instalação baixado e o nome do dispositivo para o pendrive USB. Este exemplo grava a imagem de instalação amd64 no primeiro dispositivo USB em um sistema FreeBSD existente.

```
# dd if=FreeBSD-12.1-RELEASE-amd64-memstick.img of=/dev/da0 bs=1M conv=sync
```

   Se este comando falhar, verifique se o pendrive USB não está montado e se o nome do dispositivo aponta para o disco, não para uma partição. Alguns sistemas operacionais podem requerer que este comando seja executado com o sudo(8). A sintaxe do dd(1) varia ligeiramente em diferentes plataformas; por exemplo, o Mac OS™ requer um `bs=1m` em minúsculas. Sistemas como o Linux™ podem gravar em buffer. Para forçar todas as gravações a serem concluídas, use o comando sync(8).

Procedure: Usando o Windows™ para gravar a imagem



Certifique-se de fornecer a letra da unidade correta, pois os dados existentes na unidade especificada serão sobrescritos e destruídos.

1. Obtendo o Image Writer para Windows™

   O Image Writer para Windows™ é um aplicativo gratuito que pode gravar corretamente um arquivo de imagem em um cartão de memória. Faça o download a partir de https://sourceforge.net/projects/win32diskimager/ e extraia-o em uma pasta.

2. Escrevendo a imagem com o Image Writer

   Clique duas vezes no ícone Win32DiskImager para iniciar o programa. Verifique se a letra da unidade mostrada em `Device` é a unidade com o cartão de memória. Clique no ícone da pasta e selecione a imagem a ser gravada no cartão de memória. Clique em **[Save]** para aceitar o nome do arquivo de imagem. Verifique se tudo está correto e se nenhuma pasta do cartão de memória está aberta em outras janelas. Quando tudo estiver pronto, clique em **[Write]** para gravar o arquivo de imagem no cartão de memória.

Agora você está pronto para começar a instalar o FreeBSD.

2.4. Iniciando a instalação



Por padrão, a instalação não fará alterações no(s) disco(s) antes da seguinte

mensagem:

```
Your changes will now be written to disk. If you
have chosen to overwrite existing data, it will
be PERMANENTLY ERASED. Are you sure you want to
commit your changes?
```

A instalação pode ser encerrada a qualquer momento antes deste aviso. Se houver uma preocupação de que algo esteja configurado incorretamente, basta desligar o computador antes desse ponto e nenhuma alteração será feita nos discos do sistema.

Esta seção descreve como inicializar o sistema a partir da mídia de instalação que foi preparada usando as instruções em Prepare a mídia de instalação. Ao usar um dispositivo USB inicializável, conecte o dispositivo USB antes de ligar o computador. Ao inicializar a partir do CD ou do DVD, ligue o computador e insira a mídia na primeira oportunidade. O procedimento para configurar o sistema para inicializar a partir da mídia inserida depende da arquitetura.

2.4.1. Inicializando em i386™ e amd64

Estas arquiteturas fornecem um menu BIOS para selecionar o dispositivo de inicialização. Dependendo da mídia de instalação usada, selecione o dispositivo de CD/DVD ou o USB como o primeiro dispositivo de inicialização. A maioria dos sistemas também fornece uma chave para selecionar o dispositivo durante a inicialização sem ter que entrar no BIOS. Normalmente, a chave é `F10`, `F11`, `F12` ou `Escape`.

Se o computador carregar o sistema operacional existente em vez do instalador do FreeBSD, então:

1. A mídia de instalação não foi inserida cedo o suficiente no processo de inicialização. Deixe a mídia inserida e tente reiniciar o computador.
2. As alterações do BIOS estavam incorretas ou não foram salvas. Verifique novamente se o dispositivo de inicialização correto está selecionado como o primeiro dispositivo de inicialização.
3. Este sistema é muito antigo para suportar a inicialização a partir da mídia escolhida. Neste caso, o Plop Boot Manager (http://www.plop.at/en/bootmanagers.html) pode ser usado para inicializar o sistema a partir da mídia selecionada.

2.4.2. Inicializando no PowerPC™

Na maioria das máquinas, manter pressionado o `C` no teclado durante a inicialização irá inicializar apartirdoCD.Casocontrário,mantenhapressionados `Command` + `Option` + `O` + `F`,ou `Windows` + `Alt` + `O` + `F` em teclados não-Apple™. No prompt `0 >`, digite

```
boot cd:,\ppc\loader cd:0
```

2.4.3. Menu de inicialização do FreeBSD

Quando o sistema inicializar a partir da mídia de instalação, um menu semelhante ao seguinte será exibido:



Figura 1. Menu do FreeBSD Boot Loader

Por padrão, o menu irá esperar dez segundos por uma ação do usuário antes de inicializar no instalador do FreeBSD ou, se o FreeBSD já estiver instalado, antes de inicializar no FreeBSD. Para pausar o cronômetro de inicialização para rever as seleções, pressione `Espaço`. Para selecionar uma opção, pressione seu número, caractere ou tecla destacada. As seguintes opções estão disponíveis.

- `Boot Multi User`: Isto irá continuar o processo de inicialização do FreeBSD. Se o temporizador de boot tiver sido pausado, pressione `1`, `B` maiúsculo ou minúsculo ou `Enter`.
- `Boot Single User`: Este modo pode ser usado para corrigir uma instalação existente do FreeBSD como descrito em Modo Single-User. Pressione `2` ou `S` maiúsculo ou minúsculo para entrar neste modo.
- `Escape to loader prompt`: Isso inicializará o sistema em um prompt de reparo que contém um número limitado de comandos de baixo nível. Este prompt é descrito em Estágio três. Pressione `3` ou `Esc` para inicializar neste prompt.
- `Reboot`: Reinicia o sistema.
- `Kernel`: Carrega um kernel diferente.
- `Configure Boot Options`: Abre o menu mostrado e descrito em Menu de Opções de Inicialização

do FreeBSD.



Figura 2. Menu de Opções de Inicialização do FreeBSD

O menu de opções de inicialização é dividido em duas seções. A primeira seção pode ser usada para retornar ao menu de inicialização principal ou para redefinir quaisquer opções que tenham sido alteradas de volta para seus valores padrões.

A próxima seção é usada para alternar as opções disponíveis para `On` ou `Off` pressionando o número ou caractere realçado da opção. O sistema sempre inicializará usando as configurações dessas opções até serem modificadas. Várias opções podem ser alternadas usando este menu:

- `ACPI Support`: Se o sistema travar durante a inicialização, tente alternar essa opção para `Off`.
- `Safe Mode`: Se o sistema ainda travar durante a inicialização, mesmo com `Suporte a ACPI` definido como `Off`, tente definir esta opção como `On`.
- `Single User`: Alterne esta opção para `On` para corrigir uma instalação existente do FreeBSD como descrito em Modo Single-User. Depois que o problema for corrigido, configure-o de volta para `Off`.
- `Verbose`: Alterne esta opção para `On` para ver mensagens mais detalhadas durante o processo de inicialização. Isso pode ser útil ao solucionar problemas de hardware.

Depois de fazer as seleções necessárias, pressione `1` ou `Backspace` para retornar ao menu de boot principal, então pressione `Enter` para continuar a inicialização no FreeBSD. Uma série de mensagens de inicialização irão aparecer enquanto o FreeBSD executa seus testes de dispositivos

de hardware e carrega o programa de instalação. Quando a inicialização estiver concluída, o menu de boas-vindas mostrado em Menu de boas-vindas será exibido.

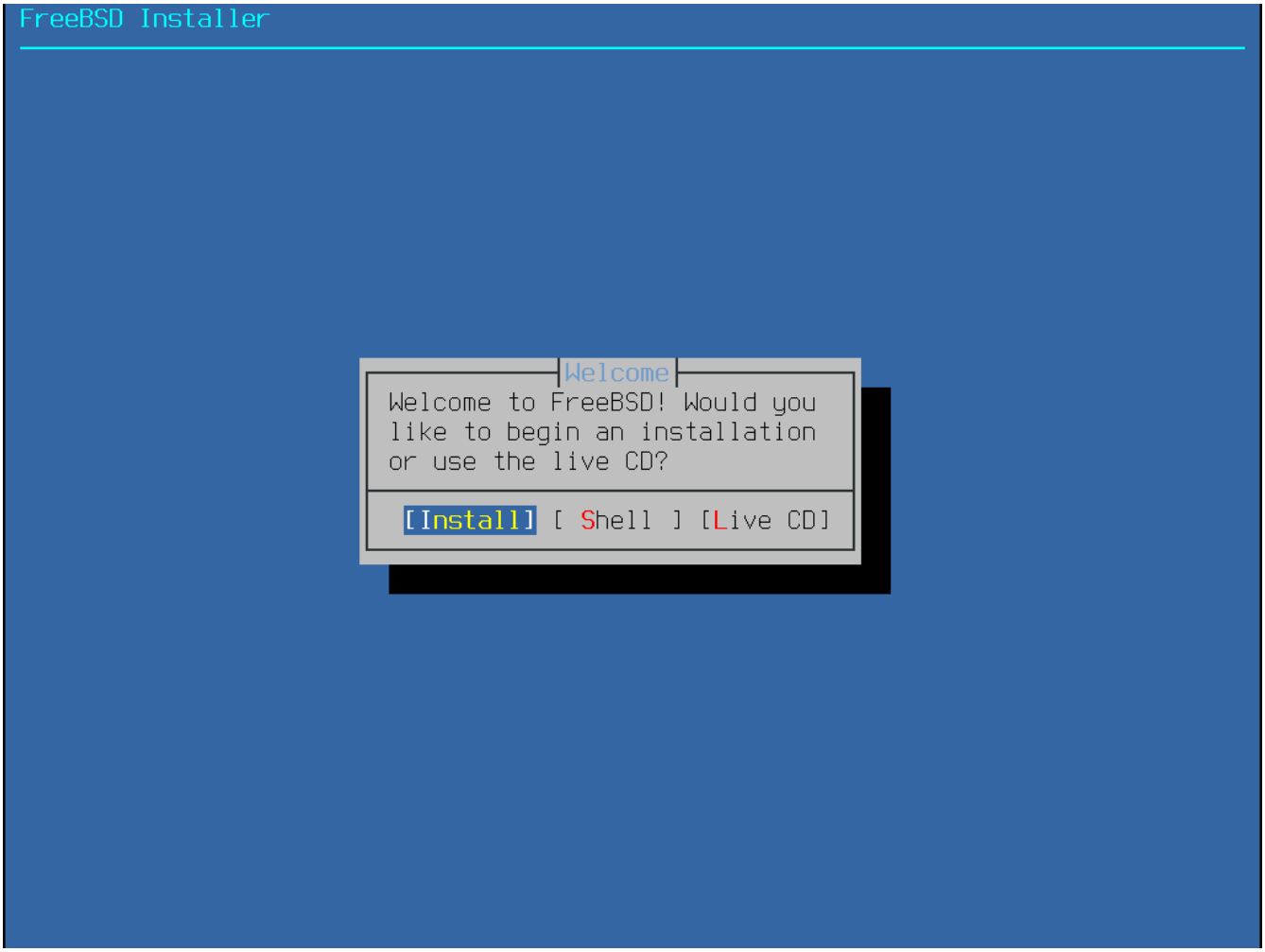


Figura 3. Menu de boas-vindas

Pressione `Enter` para selecionar o padrão de **[Install]** para entrar no instalador. O restante deste capítulo descreve como usar este instalador. Caso contrário, use as setas para a direita ou para a esquerda ou a letra colorida para selecionar o item de menu desejado. A opção **[Shell]** pode ser usada para acessar um shell do FreeBSD, a fim de usar utilitários de linha de comando para preparar os discos antes da instalação. A opção **[Live CD]** pode ser usada para testar o FreeBSD antes de instalá-lo. A versão live é descrita em Usando o Live CD.



Para revisar as mensagens de inicialização, incluindo o probe do dispositivo de hardware, pressione a tecla `S` maiúscula ou minúscula e, em seguida, `Enter` para acessar um shell. No prompt do shell, digite `more /var/run/dmesg.boot` e use a barra de espaço para rolar pelas mensagens. Quando terminar, digite `exit` para retornar ao menu de boas-vindas.

2.5. Usando o bsdinstall

Esta seção mostra a ordem dos menus do bsdinstall e o tipo de informação que será solicitada antes que o sistema seja instalado. Use as teclas de seta para realçar uma opção de menu e, em seguida, a barra de `Espaço` para selecionar ou desmarcar esse item de menu. Quando terminar, pressione `Enter` para salvar a seleção e passar para a próxima tela.

2.5.1. Selecionando o menu do Keymap (Mapa de teclas)

Antes de iniciar o processo, o bsdinstall carregará os arquivos de keymap como mostrado em Carregamento de Keymap.

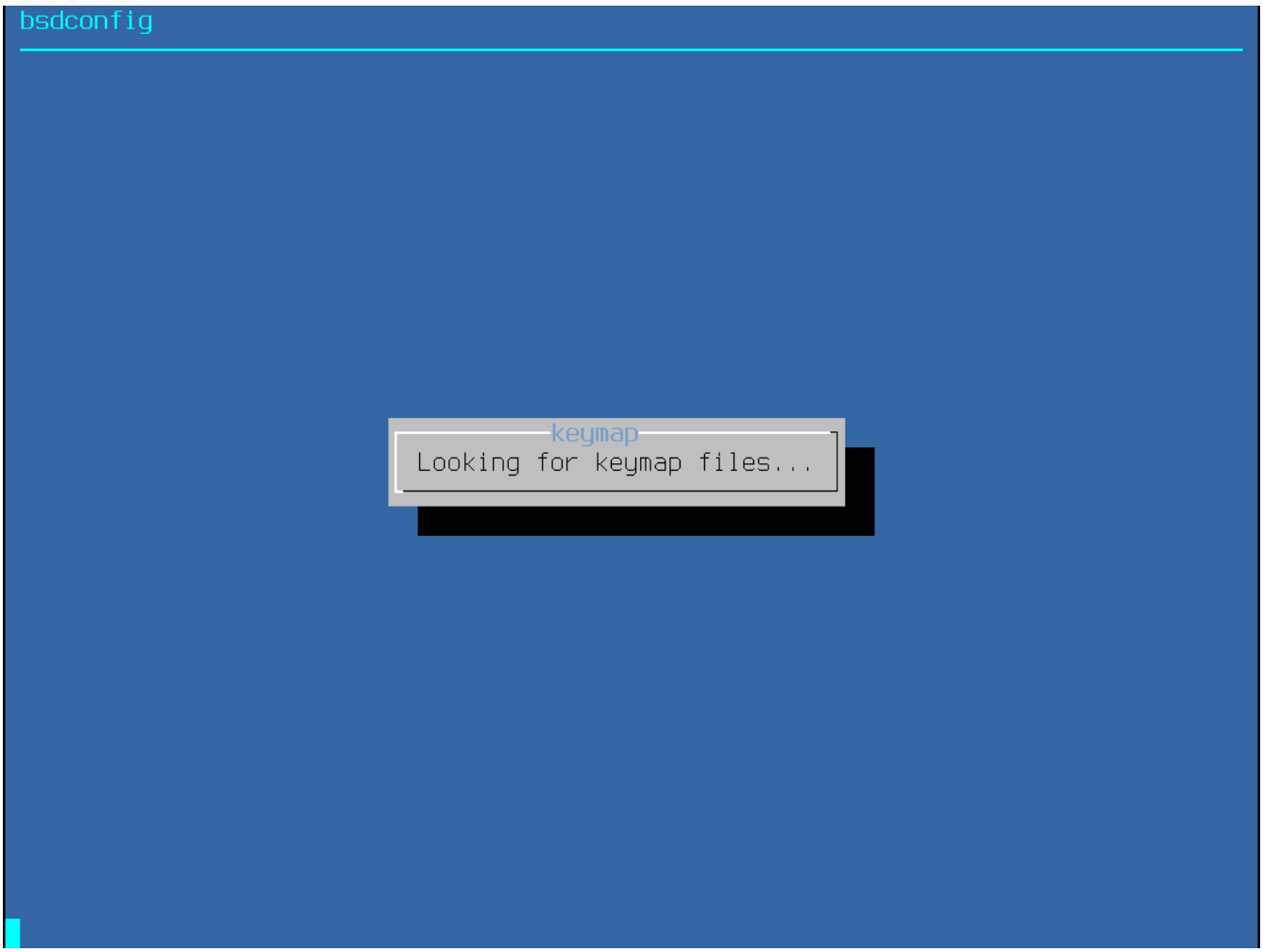


Figura 4. Carregamento de Keymap

Após o carregamento dos keymaps, o bsdinstall exibe o menu mostrado em Menu de Seleção do Keymap. Use as setas para cima e para baixo para selecionar o mapa de teclas que mais representa o mapeamento do teclado conectado ao sistema. Pressione `Enter` para salvar a seleção.

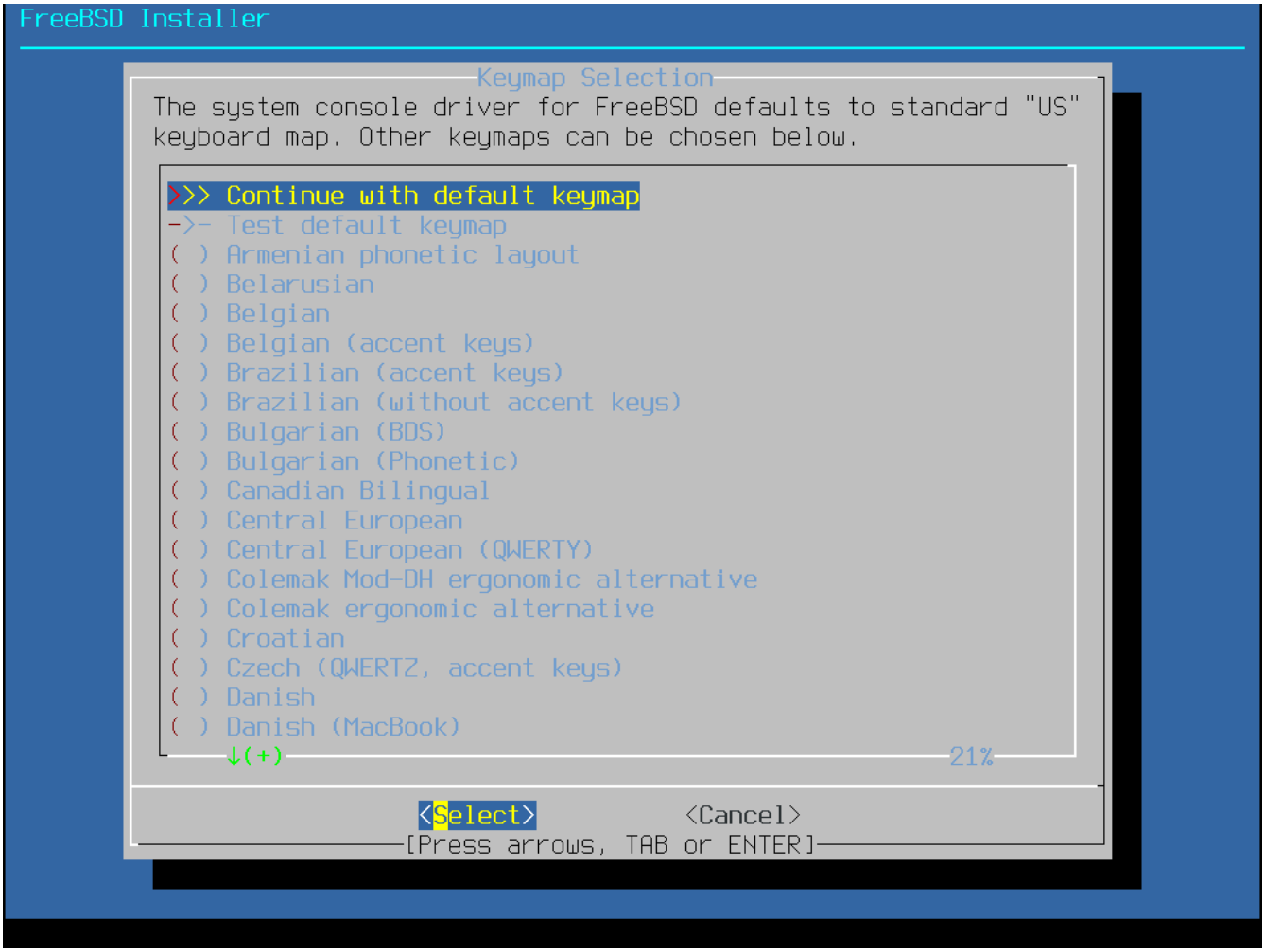


Figura 5. Menu de Seleção do Keymap



Pressionar `Esc` sairá deste menu e usará o mapa de teclas padrão. Se a escolha do mapa de teclado não for clara, a opção United States of America ISO-8859-1 é uma opção segura.

Além disso, ao selecionar um keymap diferente, o usuário pode testar o keymap e garantir que esteja correto antes de continuar, conforme mostrado em Menu de Teste do Keymap.

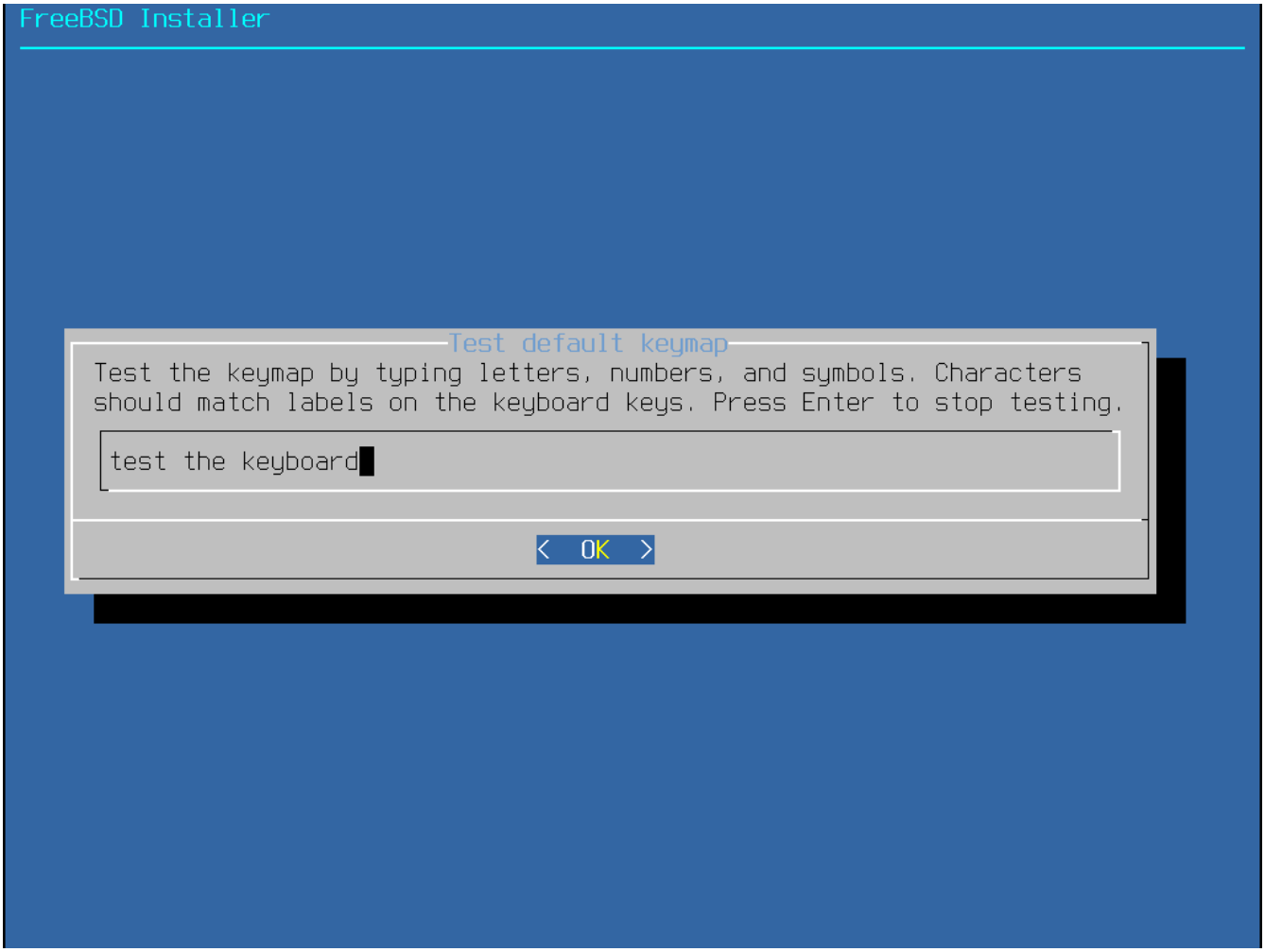


Figura 6. Menu de Teste do Keymap

2.5.2. Configurando o nome do host

O próximo menu do bsdinstall é usado para definir o nome do host para o sistema recém-instalado.

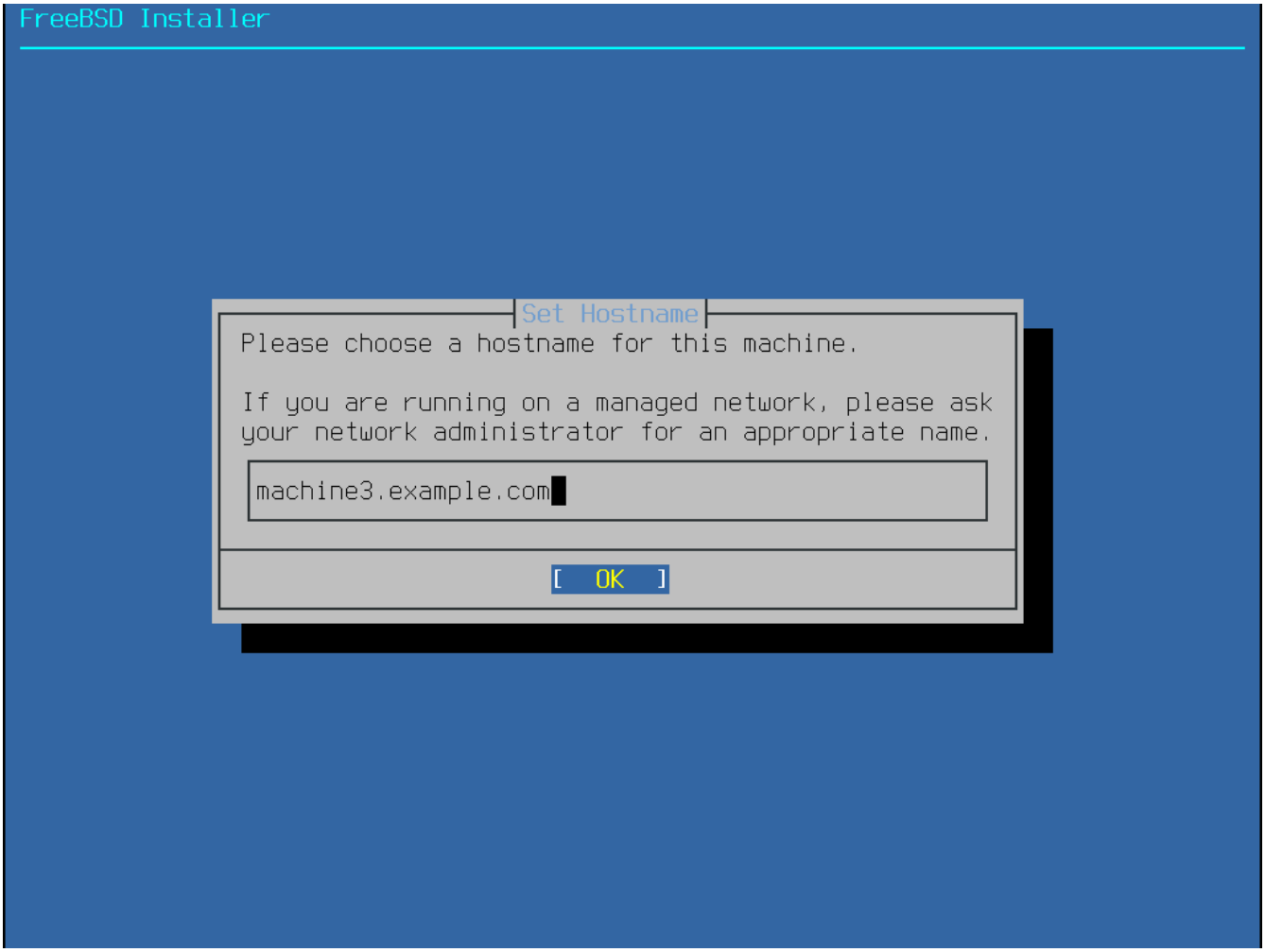


Figura 7. Configurando o nome do host

Digite um nome de host exclusivo para a rede. Ele deve ser um nome de host totalmente qualificado, como `machine3.example.com`.

2.5.3. Selecionando Componentes para Instalar

Em seguida, o bsdinstall solicitará a seleção de componentes opcionais para instalação.

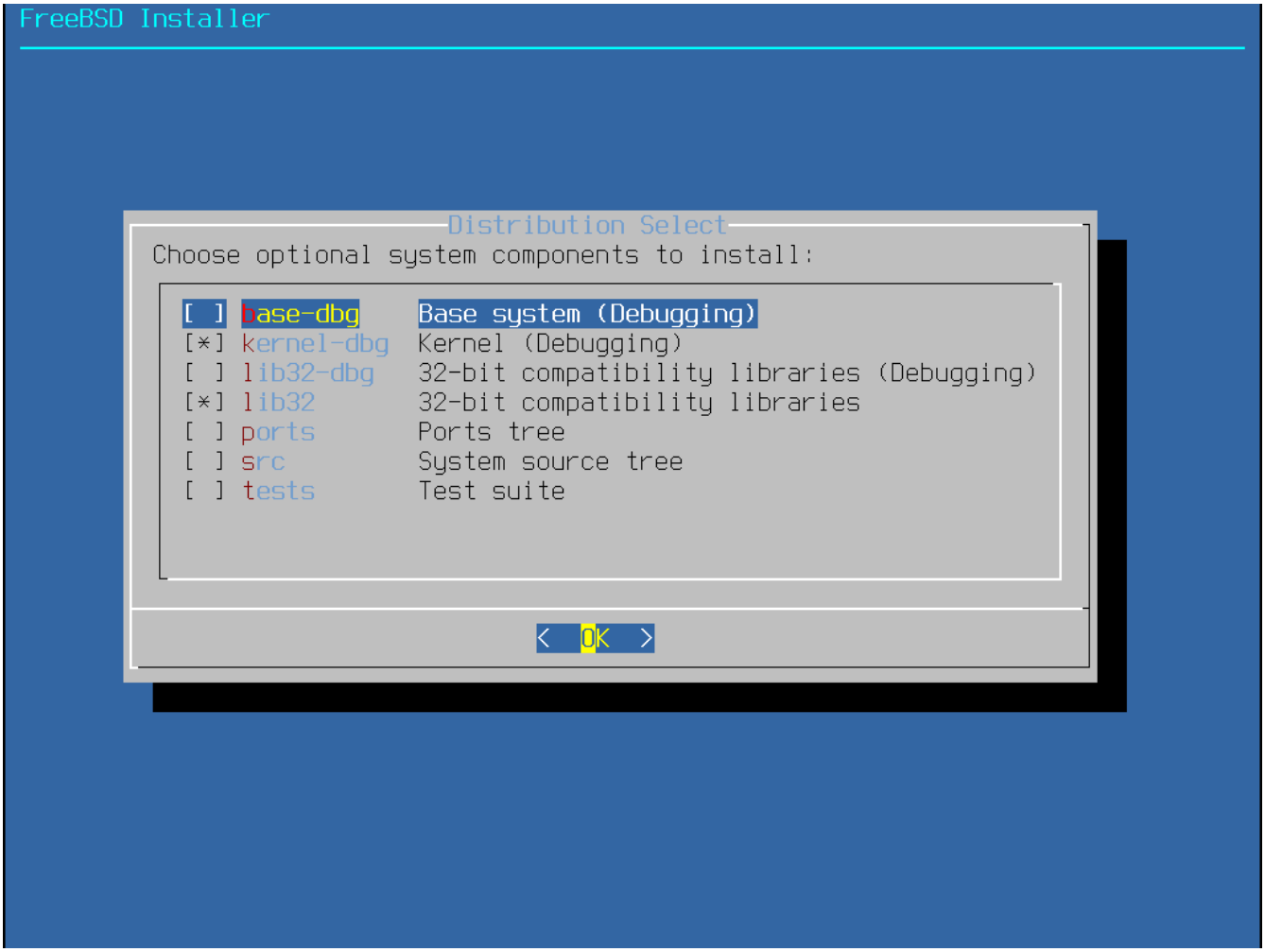


Figura 8. Selecionando Componentes para Instalar

Decidir quais componentes instalar dependerá em grande parte do uso pretendido para o sistema e da quantidade de espaço em disco disponível. O kernel do FreeBSD e o userland, coletivamente conhecidos como o *sistema base*, são sempre instalados. Dependendo da arquitetura, alguns desses componentes podem não aparecer:

- `base-dbg` - Ferramentas básicas como cat, ls entre outras com símbolos de depuração ativados.
- `kernel-dbg` - Kernel e módulos com símbolos de depuração ativados.
- `lib32-dbg` - Bibliotecas de compatibilidade para executar aplicativos de 32 bits em uma versão de 64 bits do FreeBSD com símbolos de depuração ativados.
- `lib32` - Bibliotecas de compatibilidade para executar aplicativos de 32 bits em uma versão de 64 bits do FreeBSD.
- `ports` - A Coleção de Ports do FreeBSD é uma coleção de arquivos que automatiza o download, a compilação e a instalação de pacotes de software de terceiros. Instalando Aplicativos: Pacotes e Ports discute como usar a coleção de ports.



O programa de instalação não verifica o espaço em disco adequado. Selecione esta opção apenas se houver espaço suficiente no disco rígido. A Coleção de Ports do FreeBSD ocupa cerca de 500 MB de espaço em disco.

- `src` - O código-fonte completo do FreeBSD para o kernel e para o userland. Embora não seja necessário para a maioria dos aplicativos, pode ser necessário para compilar drivers de

dispositivo, módulos do kernel ou alguns aplicativos da Coleção de Ports. Ele também é usado para desenvolver o próprio FreeBSD. A árvore de código-fonte completa requer 1 GB de espaço em disco e a recompilação de todo o sistema FreeBSD requer 5 GB adicionais de espaço.

- `tests` - FreeBSD Test Suite.

2.5.4. Instalando a partir da rede

O menu mostrado em Instalando a partir da rede apenas aparece ao instalar a partir de um -bootonly.iso ou -mini-memstick.img pois esta mídia de instalação não possui cópias dos arquivos de instalação. Como os arquivos de instalação devem ser recuperados através de uma conexão de rede, esse menu indica que a interface de rede deve ser configurada primeiro. Se o menu é exibido em qualquer etapa do processo lembre-se de seguir as instruções em Configurando as Interfaces de Rede.

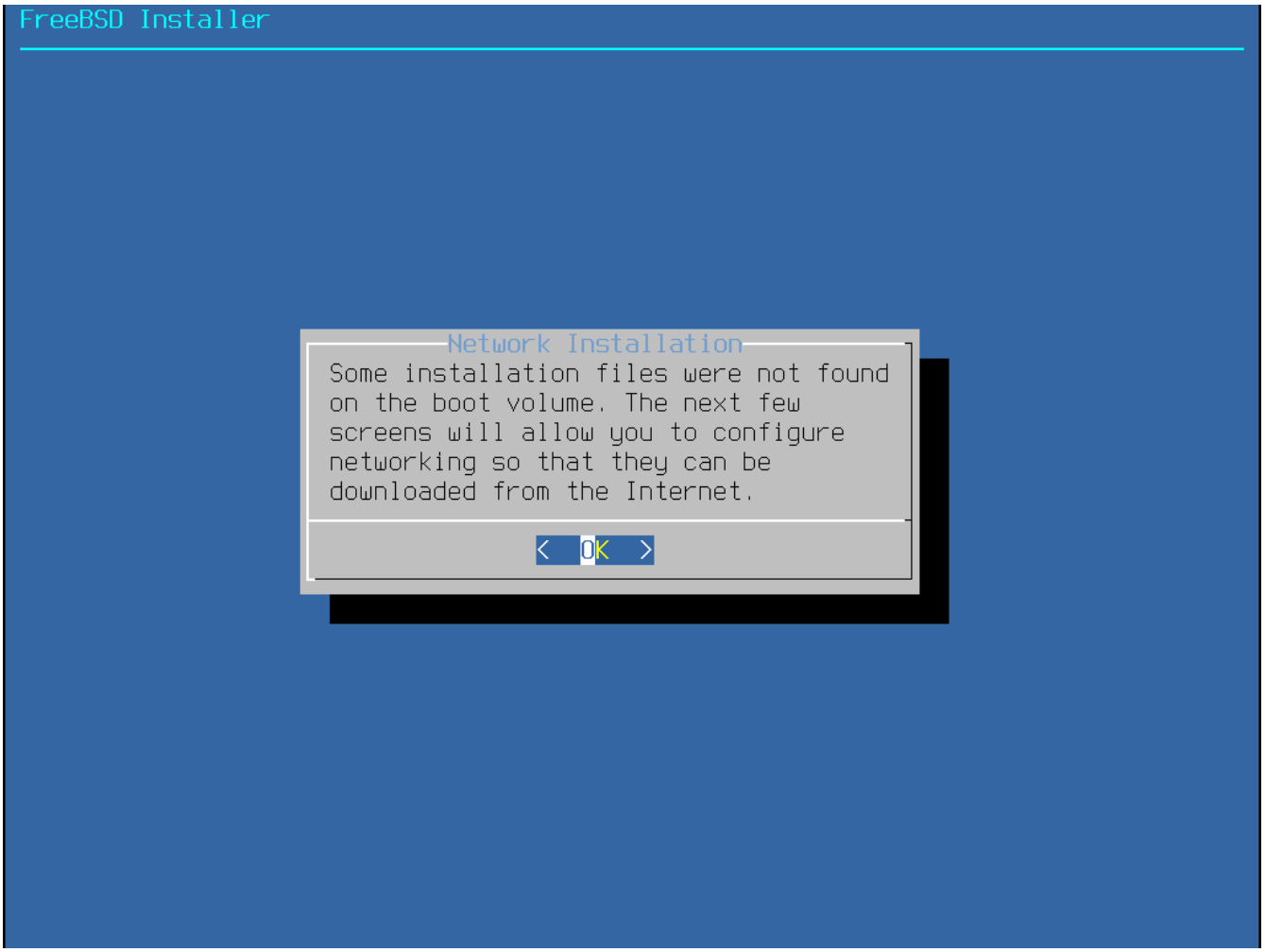


Figura 9. Instalando a partir da rede

2.6. Alocando o espaço em disco

O próximo menu é usado para determinar o método de alocação de espaço em disco. As opções disponíveis no menu dependem da versão do FreeBSD sendo instalada.

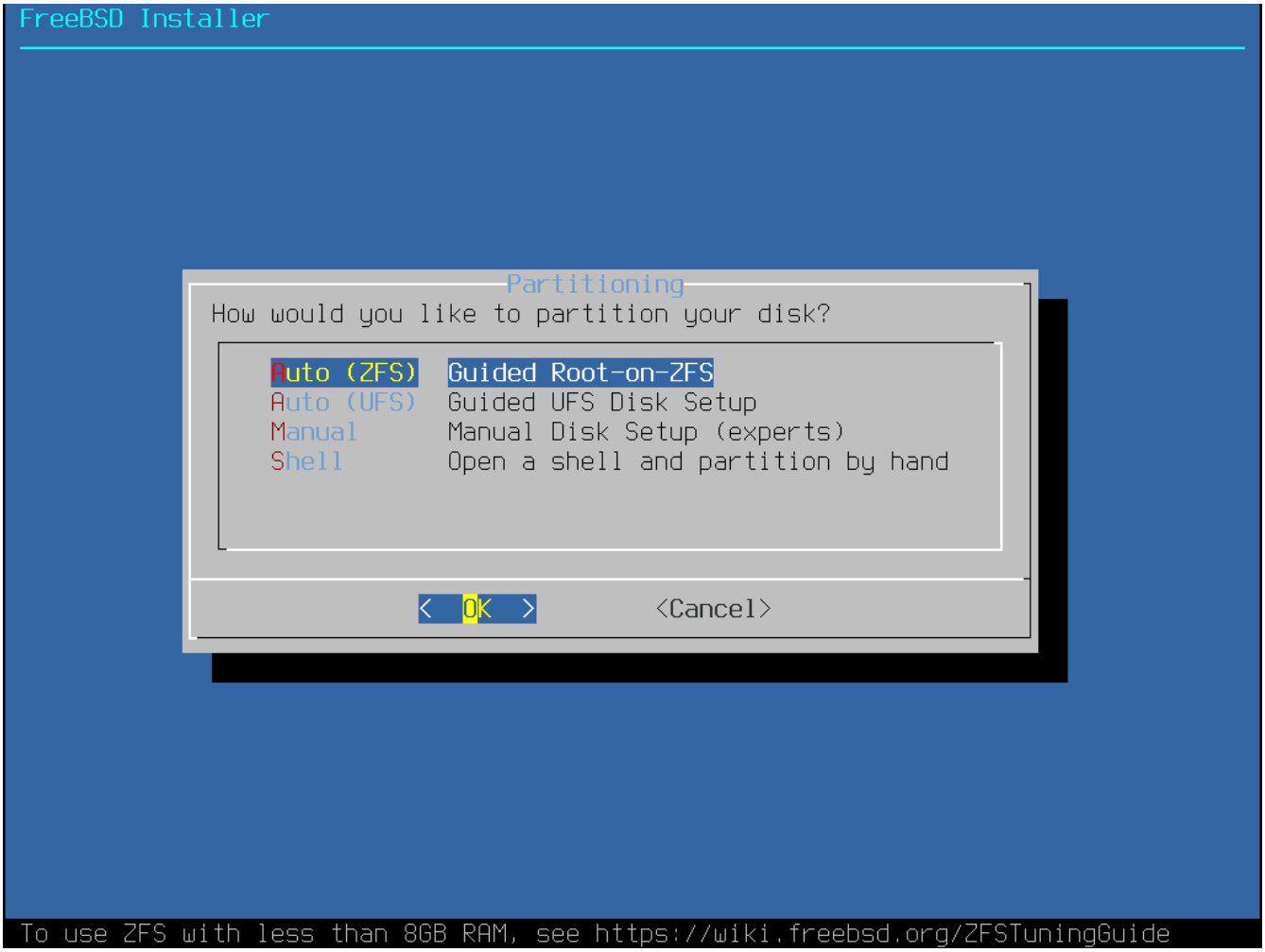


Figura 10. Opções de Particionamento

bsdinstall fornece ao usuário quatro métodos para alocar espaço em disco:

- O particionamento `Auto (UFS)` configura automaticamente as partições do disco usando o sistema de arquivos `UFS`.
- O particionamento `Manual` permite que usuários avançados criem partições personalizadas a partir das opções de menu.
- `Shell` abre um prompt de shell no qual usuários avançados podem criar partições personalizadas usando utilitários de linha de comando como gpart(8), fdisk(8), e bsdlabel(8).
- O particionamento `Auto (ZFS)` cria um sistema root-on-ZFS com suporte opcional à criptografia GELI para *boot environments*.

Esta seção descreve o que considerar ao definir as partições de disco. Em seguida, demonstra como usar os diferentes métodos de particionamento.

2.6.1. Criando o layout da partição

Ao criar os sistemas de arquivos, lembre-se de que os discos rígidos transferem dados mais rapidamente das trilhas externas para as internas. Assim, sistemas de arquivos menores e mais acessados devem estar mais próximos da parte externa da unidade, enquanto partições maiores, como /usr, devem ser colocadas em direção às partes internas do disco. É uma boa idéia criar partições em uma ordem similar a: /, swap, /var e /usr.

O tamanho da partição /var reflete o uso pretendido para a máquina. Esta partição é usada para armazenar caixas de correio, arquivos de log e spools de impressora. Caixas de correio e arquivos de log podem crescer até tamanhos inesperados, dependendo do número de usuários e de quanto tempo os arquivos de log são mantidos. Na média, a maioria dos usuários raramente precisa de mais do que cerca de um gigabyte de espaço livre em disco no /var.



Às vezes, é necessário muito espaço em disco no /var/tmp. Quando um novo software é instalado, as ferramentas de empacotamento extraem uma cópia temporária dos pacotes no /var/tmp. Grandes pacotes de software, como o Firefox ou LibreOffice podem ser difíceis de instalar se não houver espaço em disco suficiente no /var/tmp.

A partição /usr contém muitos dos arquivos que suportam o sistema, incluindo o a Coleção de Ports do FreeBSD e o código-fonte do sistema. Pelo menos 2 gigabytes de espaço são recomendados para esta partição.

Ao selecionar os tamanhos das partições, lembre-se dos requisitos de espaço. Ficar sem espaço em uma partição enquanto mal usa outra pode ser um aborrecimento.

Como regra geral, a partição swap deve ter o dobro do tamanho da memória física (RAM). Sistemas com pouca memória RAM podem ter um melhor desempenho com mais swap. Configurar um swap pequeno pode levar a ineficiências no código de verificação de página da VM e pode criar problemas mais tarde, se mais memória for adicionada.

Em sistemas maiores com vários discos SCSI ou vários discos IDE operando em diferentes controladoras, é recomendável que uma area de swap seja configurada em cada unidade, até quatro unidades. As partições de swap devem ter aproximadamente o mesmo tamanho. O kernel pode manipular tamanhos arbitrários, mas as estruturas internas de dados podem ser dimensionadas para 4 vezes a maior partição de swap. Manter as partições de swap próximas do mesmo tamanho permitirá que o kernel otimize o espaço de swap entre discos. Partições grandes de swap são uma coisa boa, mesmo se o swap não for muito usado. Pode ser mais fácil de se recuperar de um programa devorador de memória antes de ser forçado a reinicializar.

Ao particionar adequadamente um sistema, a fragmentação introduzida nas partições menores e intensas em gravação não vai prejudicar as partições que são maioritariamente de leitura. Manter as partições com maior carga de gravação mais próximas da borda do disco aumentará o desempenho de I/O nas partições onde ela é mais necessária. Embora o desempenho de I/O nas partições maiores possa ser necessário, mudá-las mais para a borda do disco não levará a uma melhoria de desempenho significativa em relação à movimentação de /var para a borda.

2.6.2. Particionamento Guiado Usando UFS

Quando este método é selecionado, um menu exibirá o(s) disco(s) disponível(s). Se vários discos estiverem conectados, escolha aquele em que o FreeBSD deve ser instalado.

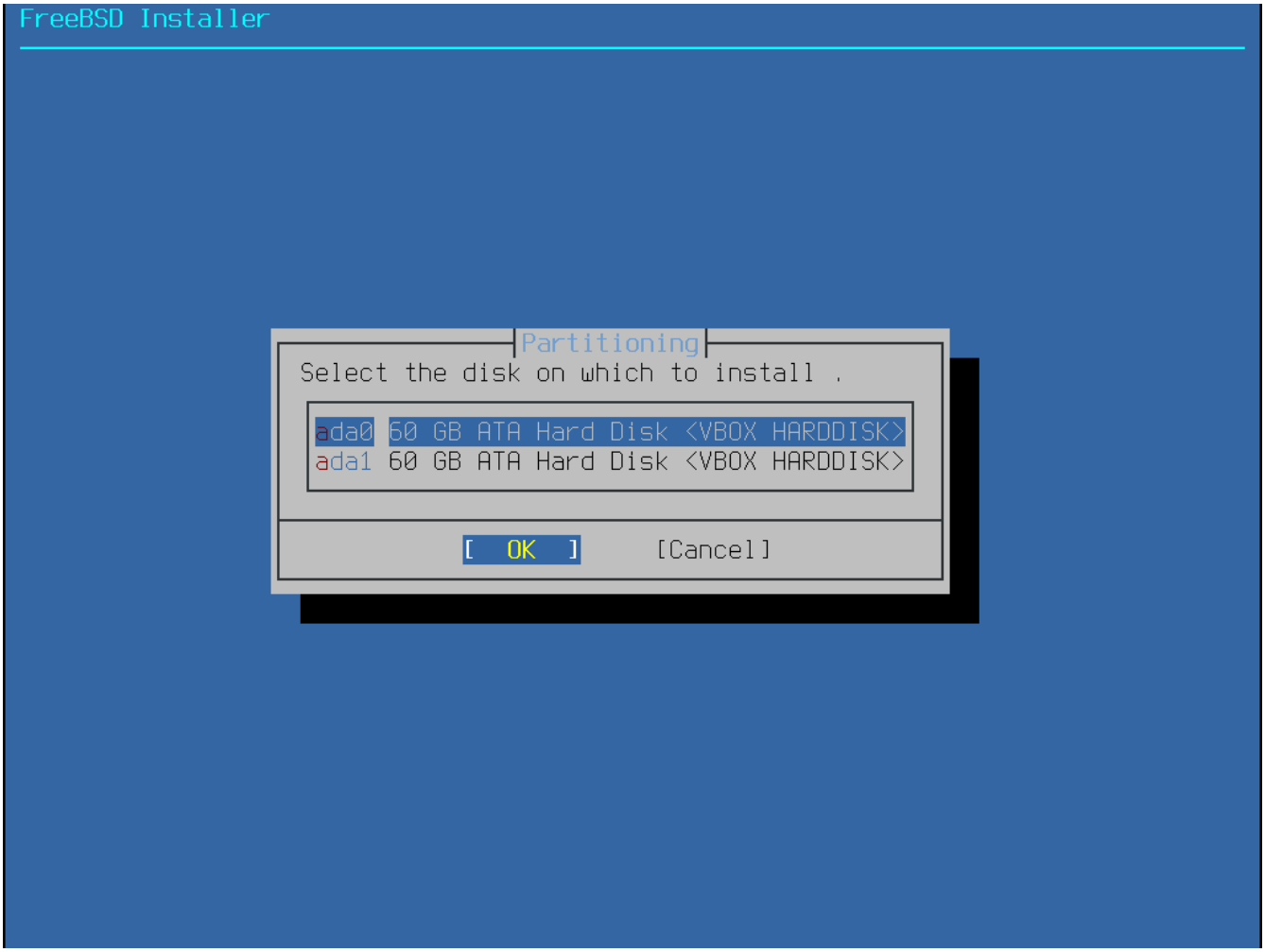


Figura 11. Selecionando a partir de vários discos

Depois que o disco é selecionado, o próximo menu solicita a instalação no disco inteiro ou a criação de uma partição usando o espaço livre. Se **[Entire Disk]** for escolhido, um layout de partição geral que preenche todo o disco é criado automaticamente. Selecionar **[Partition]** cria um layout de partição do espaço não utilizado no disco.

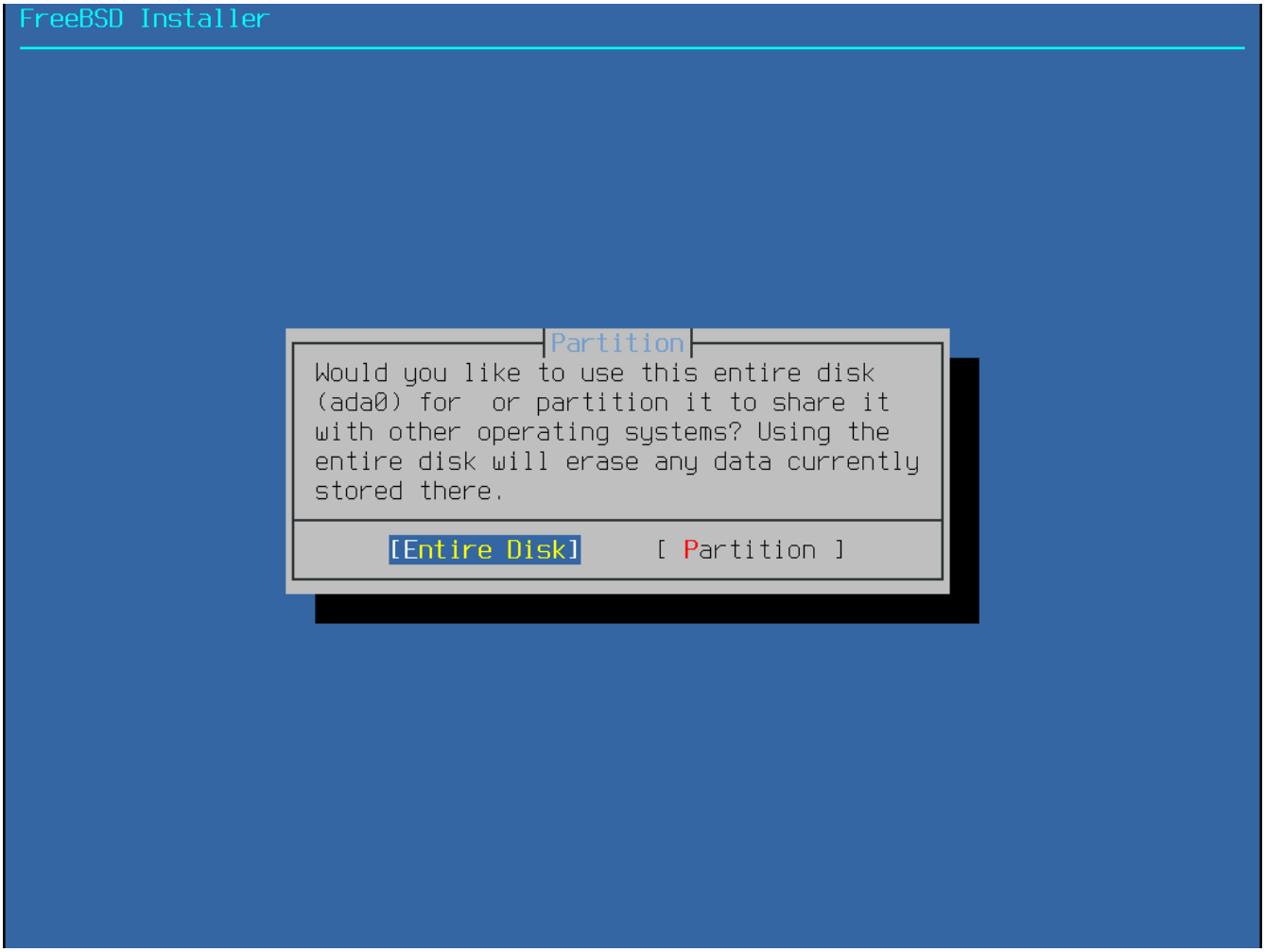


Figura 12. Selecionando todo o disco ou partição

Após **[Entire Disk]** ser escolhido, bsdinstall exibe uma caixa de diálogo indicando que o disco será apagado.

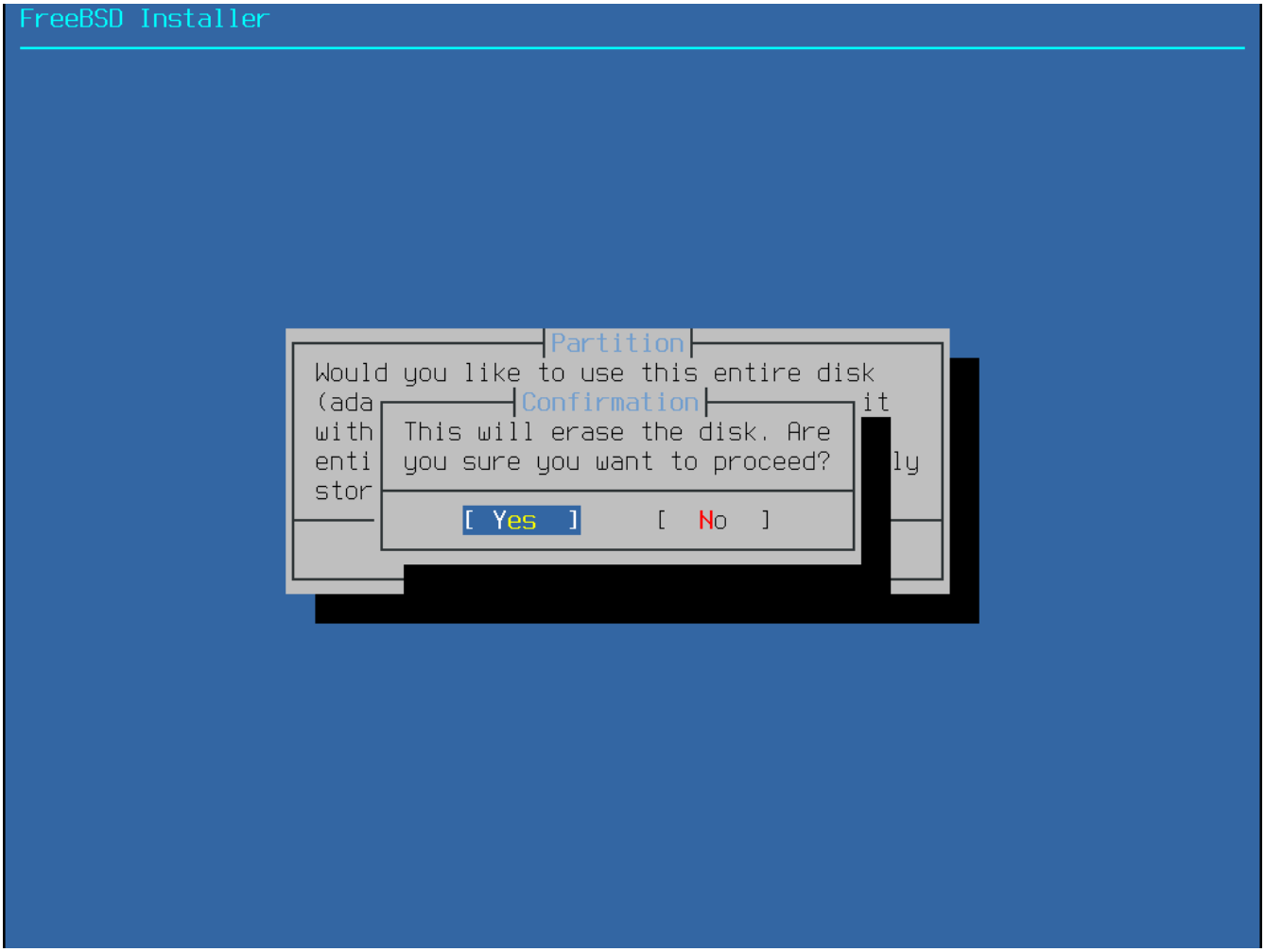


Figura 13. Confirmação

O próximo menu mostra uma lista com os tipos de esquema de partição. O GPT é geralmente a opção mais apropriada para computadores amd64. Computadores mais antigos que não são compatíveis com o GPT devem usar o MBR. Os outros esquemas de partição são geralmente usados para computadores incomuns ou antigos. Mais informações estão disponíveis em Esquemas de Particionamento.

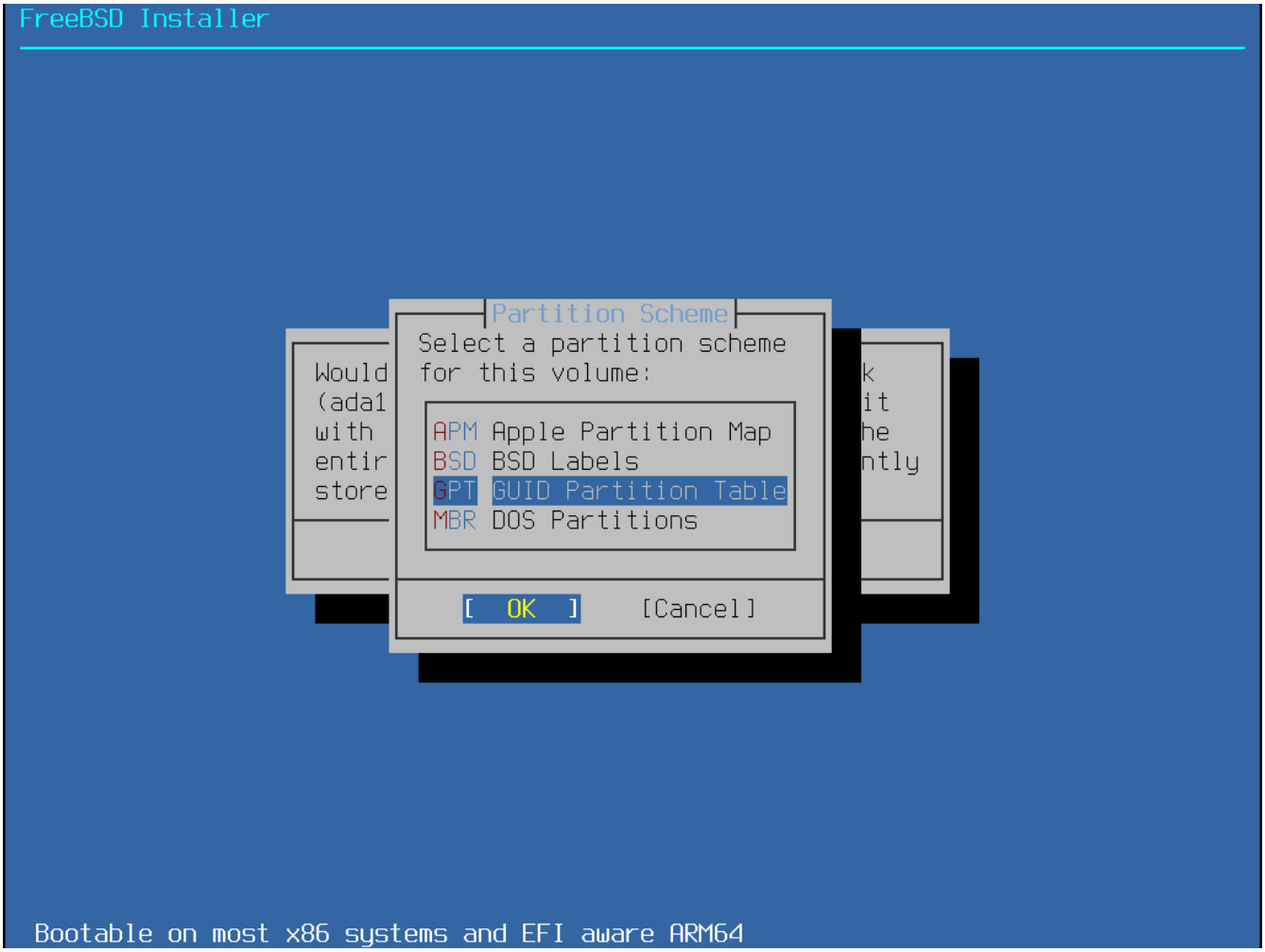


Figura 14. Selecionar Esquema de Particionamento

Depois que o layout da partição tiver sido criado, revise-o para garantir que ele atenda às necessidades da instalação. Selecionar **[Revert]** redefinirá as partições para seus valores originais e pressionar **[Auto]** recriará as partições automáticas do FreeBSD. As partições também podem ser criadas, modificadas ou excluídas manualmente. Quando o particionamento estiver correto, selecione **[Finish]** para continuar com a instalação.

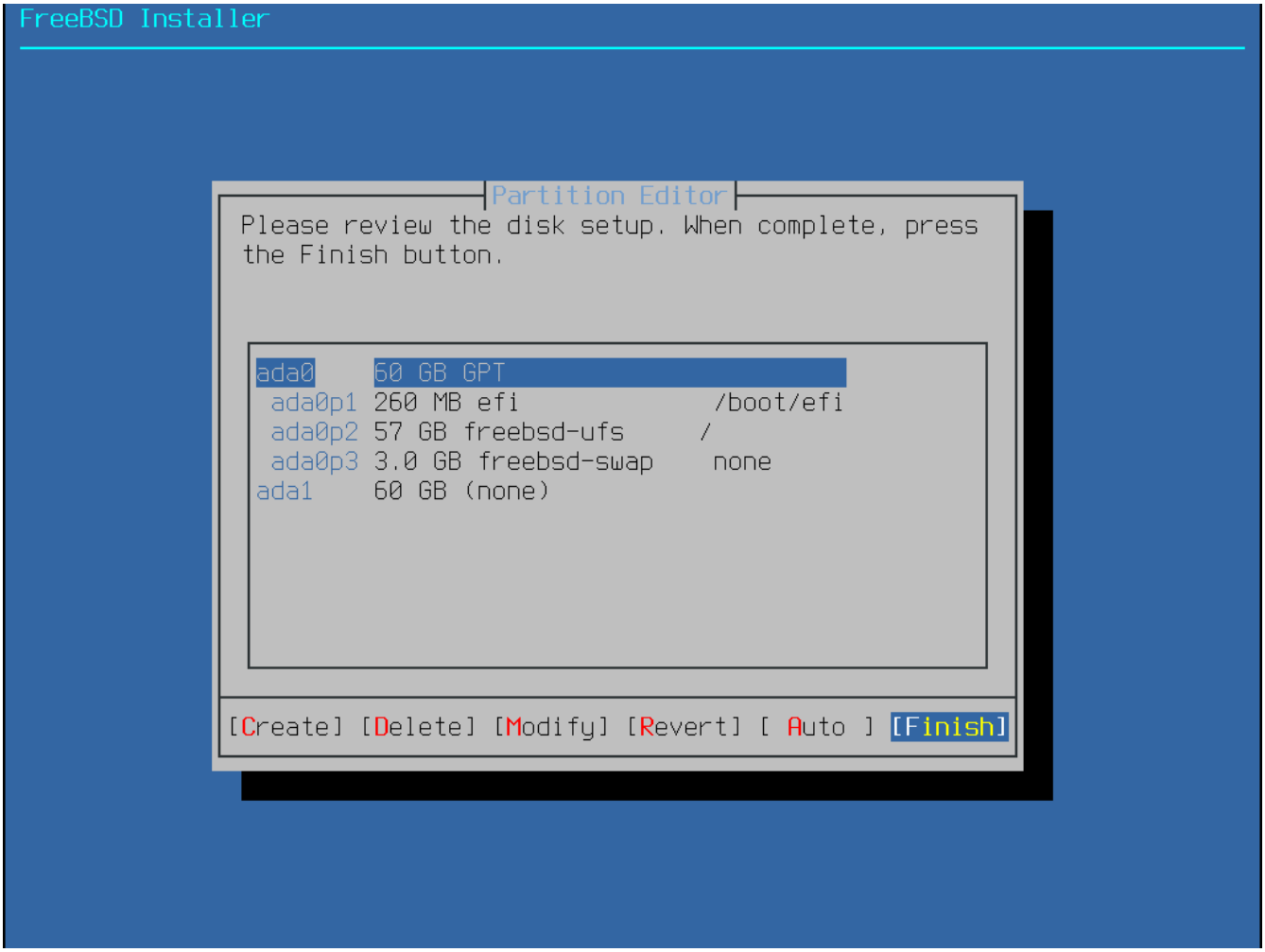


Figura 15. Revise as partições criadas

Depois que os discos são configurados, o próximo menu fornece a última chance de fazer alterações antes que os discos selecionados sejam formatados. Se for necessário fazer alterações, selecione **[Back]** para retornar ao menu principal de particionamento. **[Revert & Exit]** sairá do instalador sem fazer alterações no disco. Selecione **[Commit]** para iniciar o processo de instalação.

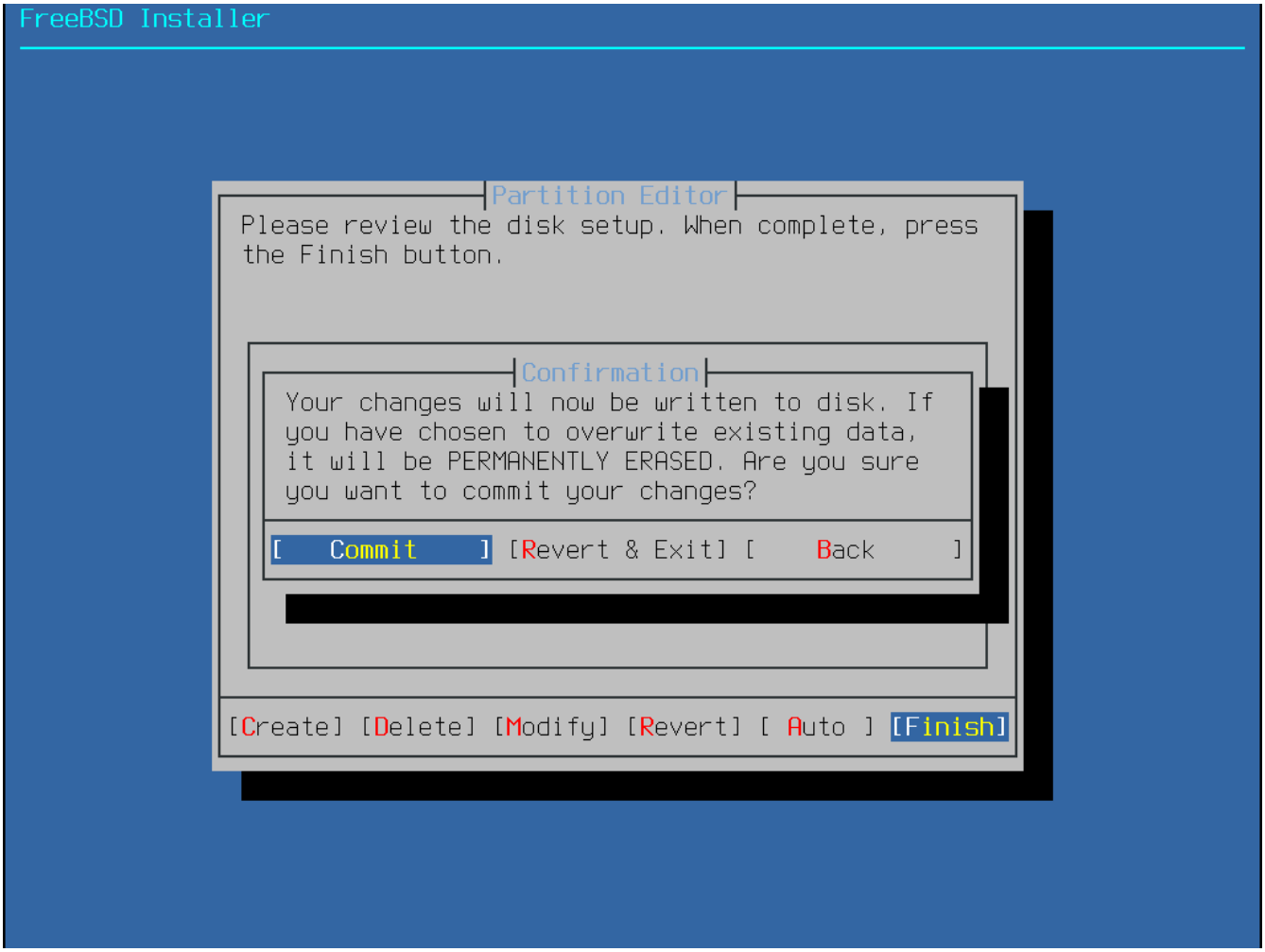


Figura 16. Confirmação final

Para continuar com o processo de instalação, vá para Fazendo o download dos arquivos de distribuição.

2.6.3. Particionamento Manual

Selecionar este método abre o editor de partições:

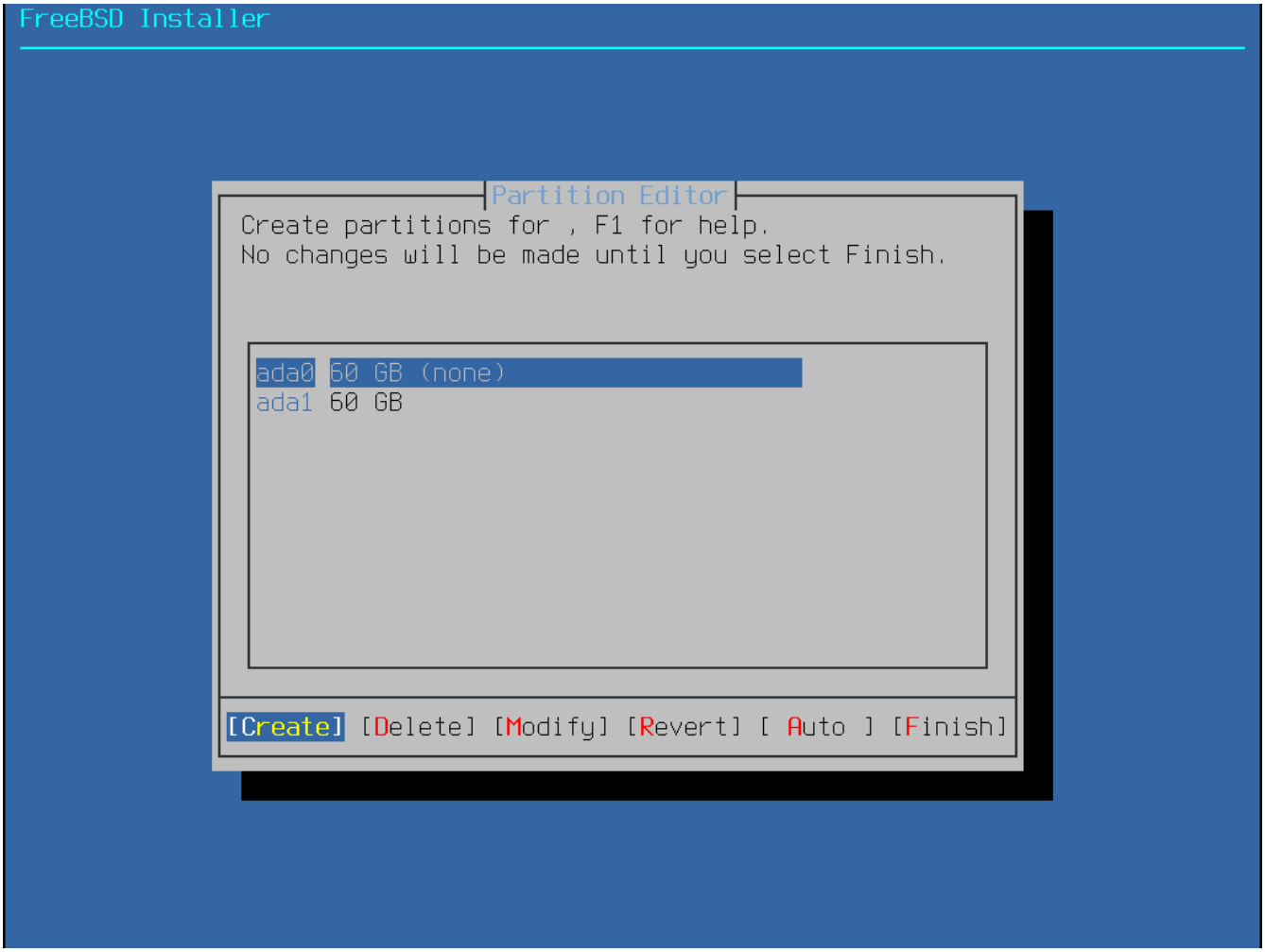


Figura 17. Criar partições manualmente

Realce a unidade de instalação (ada0 neste exemplo) e selecione **[Create]** para exibir um menu dos esquemas de partição disponíveis:

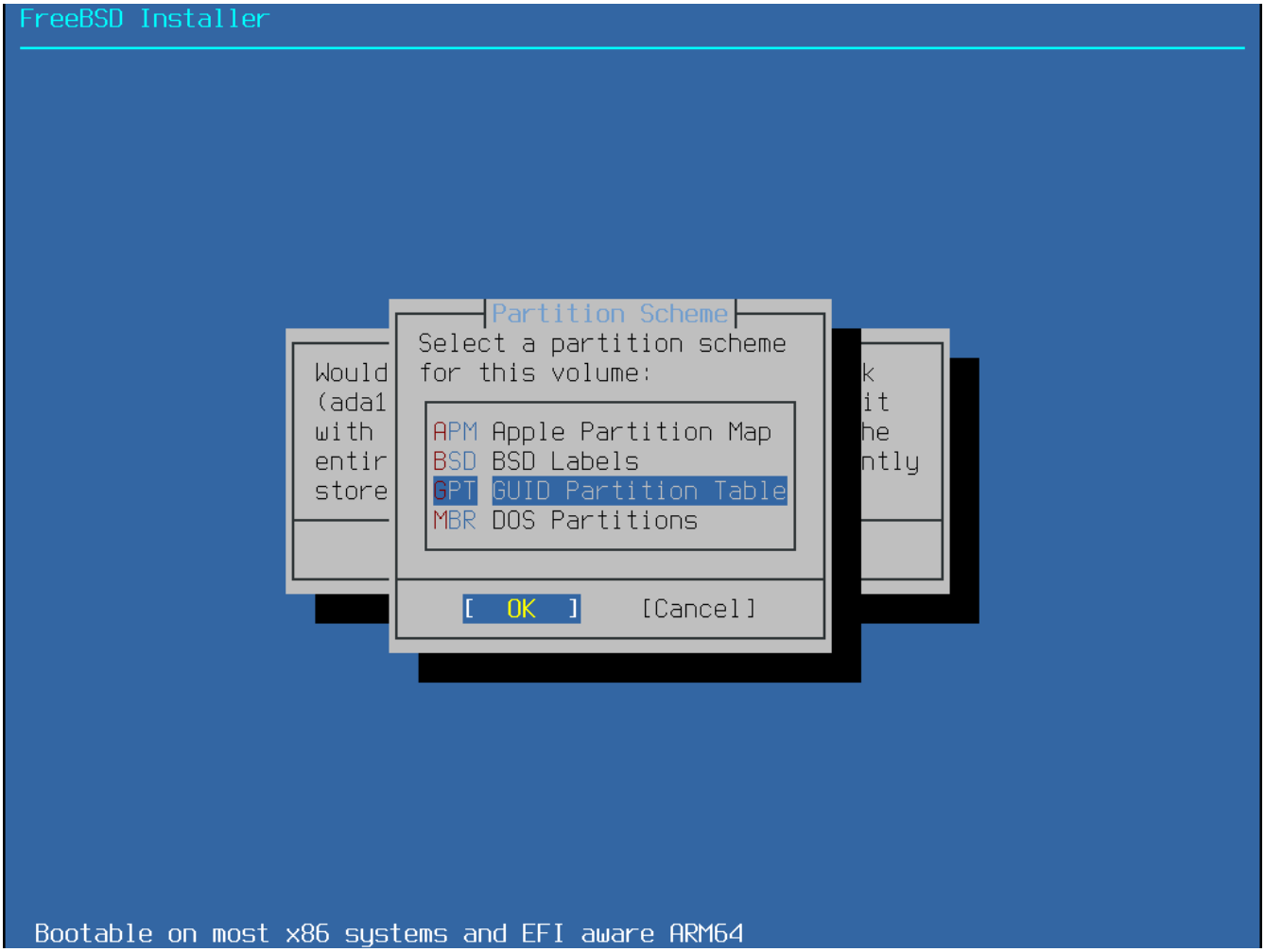


Figura 18. Criar partições manualmente

O GPT é geralmente a opção mais apropriada para computadores amd64. Computadores mais antigos que não são compatíveis com o GPT devem usar o MBR. Os outros esquemas de partição são geralmente usados para computadores incomuns ou antigos.

Tabela 1. Esquemas de Particionamento

Abreviação	**Descrição**
APM	Apple Partition Map, usado no PowerPC™.
BSD	O Label BSD sem um MBR, às vezes chamado de *modo perigosamente dedicado* porque os utilitários de discos não BSD podem não reconhecê-lo.
GPT	Tabela de Partição GUID (http://en.wikipedia.org/wiki/GUID_Partition_Table).
MBR	Registro mestre de inicialização ou MBR (http://en.wikipedia.org/wiki/Master_boot_record).
VTOC8	Tabela de Volume do Conteúdo usado pelos computadores Sun SPARC64 e UltraSPARC.

Depois que o esquema de particionamento for selecionado e criado, selecione **[Create]** novamente para criar as partições.A tecla `Tab` é utilizada para navegação entre os campos.

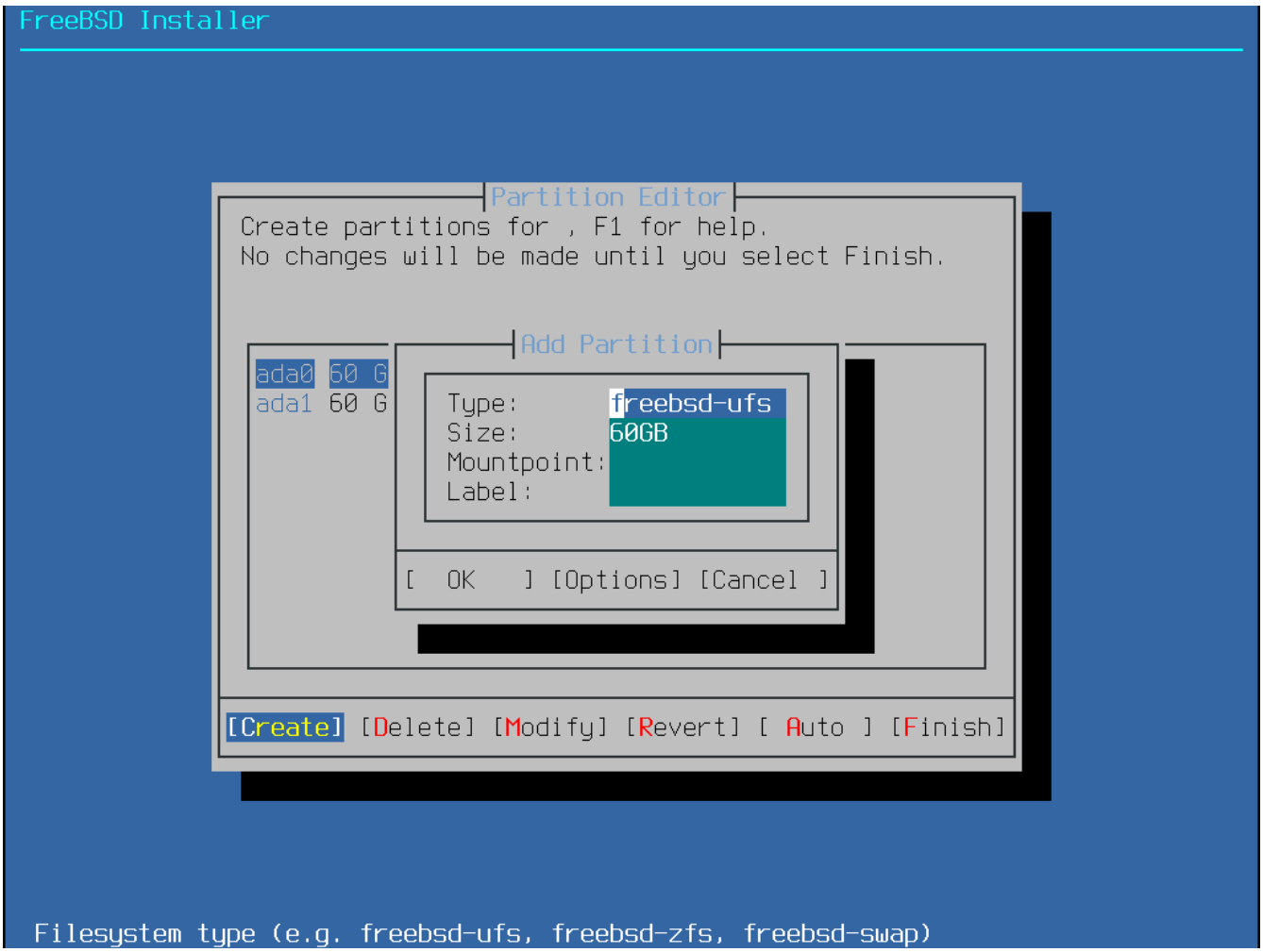


Figura 19. Criar partições manualmente

Uma instalação padrão do FreeBSD GPT usa pelo menos três partições:

- `freebsd-boot` - Mantém o código de inicialização do FreeBSD.
- `freebsd-ufs` - Um sistema de arquivos UFS do FreeBSD.
- `freebsd-zfs` - Um sistema de arquivos ZFS do FreeBSD. Mais informações sobre o ZFS estão disponíveis em O sistema de arquivos Z (ZFS).
- `freebsd-swap` - Espaço de swap do FreeBSD.

Consulte gpart(8) para obter informações de todos os tipos de partições GPT disponíveis.

Várias partições do sistema de arquivos podem ser criadas e algumas pessoas preferem um layout tradicional com partições separadas para /, /var, /tmp e /usr. Veja Criando partições tradicionais para um sistema de arquivos dividido para um exemplo.

O `tamanho` pode ser digitado com abreviações comuns: *K* para kilobytes, *M* para megabytes, ou *G* para gigabytes.



O alinhamento adequado do setor fornece o melhor desempenho, e ao definir os tamanhos das partições em múltiplos de 4K bytes ajuda a garantir o alinhamento

> em discos com setores de 512 ou 4 bytes. Geralmente, usar tamanhos de partições que são múltiplos de 1M ou 1G é a maneira mais fácil de garantir que cada partição comece em um múltiplo par de 4K. Há uma exceção: a partição *freebsd-boot* não deve ser maior que 512K devido às limitações atuais do código de inicialização.

Um `Mountpoint` é necessário se a partição contiver um sistema de arquivos. Se apenas uma única partição UFS for criada, o ponto de montagem deve ser /.

O `Label` é um nome pelo qual a partição será conhecida. Nomes ou números de unidades podem mudar se a unidade estiver conectada a um controlador ou porta diferente, mas a etiqueta da partição não muda. Referir-se a rótulos em vez de nomes de unidade e números de partição em arquivos como o /etc/fstab torna o sistema mais tolerante a alterações de hardware. Os rótulos GPT aparecem em /dev/gpt/ quando um disco é anexado. Outros esquemas de particionamento têm diferentes capacidades de rótulos e seus rótulos aparecem em diferentes diretórios no /dev/.



> Use um rótulo único e exclusivo para cada uma das partições para evitar conflitos de rótulos idênticos. Algumas letras do nome, uso ou localização do computador podem ser adicionadas ao rótulo. Por exemplo, use `labroot` ou `rootfslab` para a partição raiz UFS no computador chamado `lab`.

Exemplo 1. Criando partições tradicionais para um sistema de arquivos dividido

Para um layout de partição tradicional em que os diretórios /, /var, /tmp e /usr são sistemas de arquivos separados em suas próprias partições, crie um esquema de particionamento GPT e crie as partições conforme mostrado. Os tamanhos de partição mostrados são típicos para um disco de destino de 20G. Se houver mais espaço disponível no disco de destino, partições maiores de swap ou /var podem ser úteis. Os rótulos mostrados aqui são prefixados com `ex` para "exemplo", mas os leitores devem usar outros valores de rótulo exclusivos, conforme descrito acima.

Por padrão, o gptboot do FreeBSD espera que a primeira partição UFS seja a partição /.

Tipo de Partição	Tamanho	Ponto de montagem	Rótulo
`freebsd-boot`	`512K`		
`freebsd-ufs`	`2G`	/	`exrootfs`
`freebsd-swap`	`4G`		`exswap`
`freebsd-ufs`	`2G`	/var	`exvarfs`
`freebsd-ufs`	`1G`	/tmp	`extmpfs`
`freebsd-ufs`	aceite o padrão (restante do disco)	/usr	`exusrfs`

Depois que as partições personalizadas forem criadas, selecione **[Finish]** para continuar com a instalação e vá para Fazendo o download dos arquivos de distribuição.

2.6.4. Particionamento Guiado Usando Root-on-ZFS

Este modo de particionamento funciona apenas com discos inteiros e apagará por completo o conteúdo do disco. O menu de configuração principal do ZFS oferece várias opções para controlar a criação do pool.

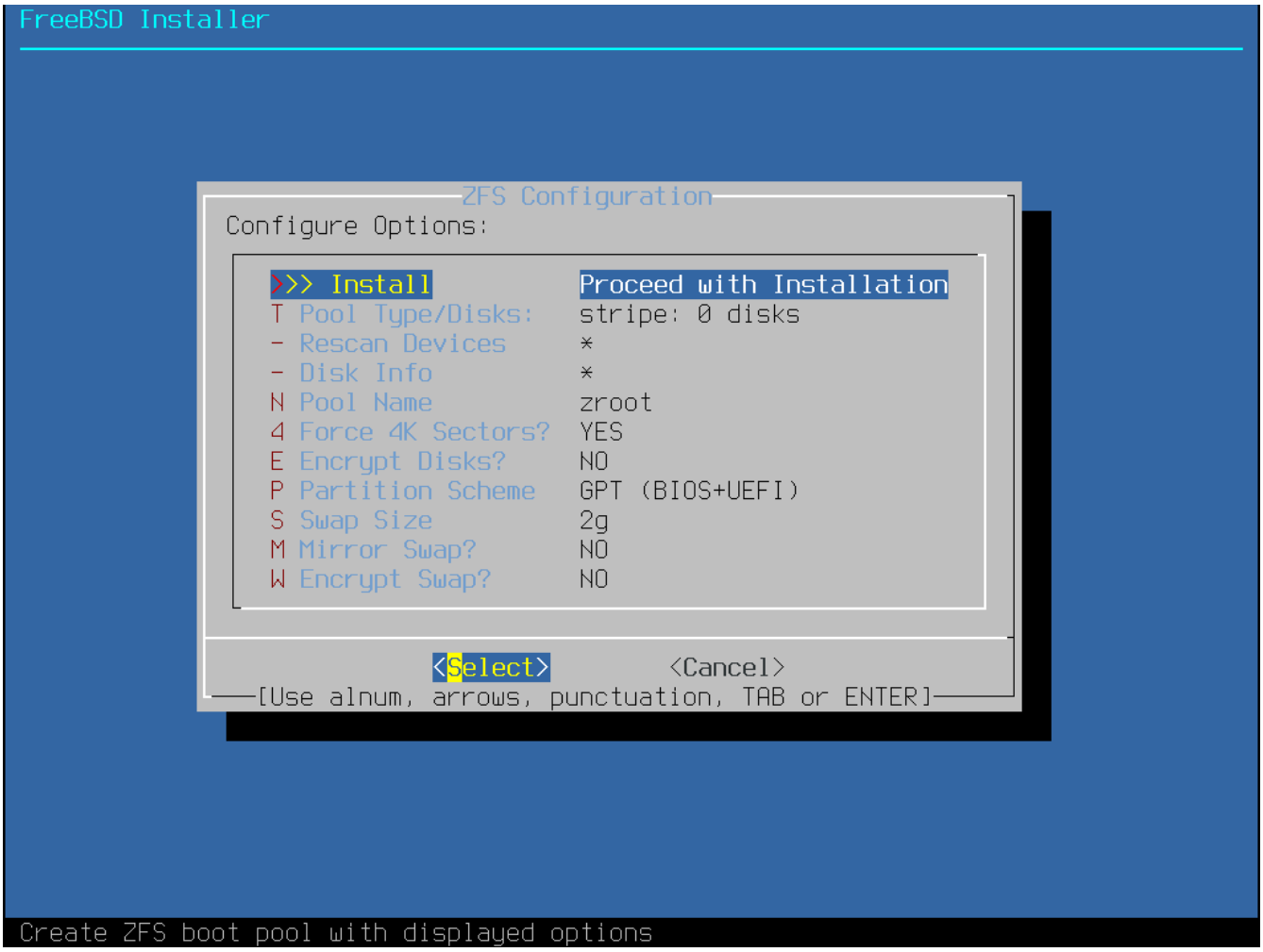


Figura 20. Menu de particionamento do ZFS

Aqui está um resumo das opções que podem ser usadas neste menu:

- `Instalar` - Prosseguir a instalação com as opções selecionadas.
- `Tipo de Pool/Discos` - Permite configurar o `Tipo de Pool` e o(s) disco(s) que irão constituir o pool. Atualmente o instalador ZFS automático suporta apenas a criação de uma única camada superior vdev, exceto em modo stripe. Para criar pools mais complexos, use as instruções em Particionamento do modo shell para criar o pool.
- `Re-escanear Dispositivos` - Re-popular a lista de discos disponíveis.
- `Disk Info` - Disk Info pode ser usado para inspecionar cada disco, incluindo sua tabela de partição e várias outras informações, como o número do modelo do dispositivo e o número de série, se disponíveis.
- `Pool Name` - Define o nome do pool. O nome default é *zroot*.
- `Force 4K Sectors?` - Forçar o uso de setores em 4K. Por padrão, o instalador irá automaticamente criar partições alinhadas com limites em 4K e forçar o ZFS a usar setores de 4K. Isto é seguro mesmo com discos de setores de 512 bytes, e tem o benefício adicional de garantir que pools

criados em discos de 512 bytes conseguirão ter setores de 4K adicionados no futuro, seja como espaço de armazenamento adicional ou como substituição ode discos e falha. Aperte a tecla `Enter` para escolher ativar isso ou não.

- `Encrypt Disks?` - A criptografia dos discos permite ao usuário criptografar os discos usando GELI. Mais informação sobre criptografia de discos está disponível em Criptografia de Disco com `geli`. Aperte a tecla `Enter` para escolher ativá-la ou não.
- `Partition Scheme` - Permitei escolher o esquema de partição. GPT é a opção recomendada na maioria dos casos. Aperte a tecla `Enter` para escolher entre diferentes opções.
- `Swap Size` - Determina a quantidade de espaço para swap.
- `Mirror Swap?` - Permite ao usuário espelhar o swap entre os discos. Fique atento, o espelhamento da swap irá quebrar dumps de crash. Pressione a tecla `Enter` para ativar ou não.
- `Encrypt Swap?` - Permite ao usuário criptografar a swap. Criptografa a swap com uma chave temporária toda vez que o sistema inicializa e a descarta na reinicialização. Pressione a tecla `Enter` para ativar ou não. Mais informação sobre criptografia de swap em Criptografando Swap.

Selecione `T` para configurar o `Pool Type` e o(s) disco(s) que irá constituir o pool.

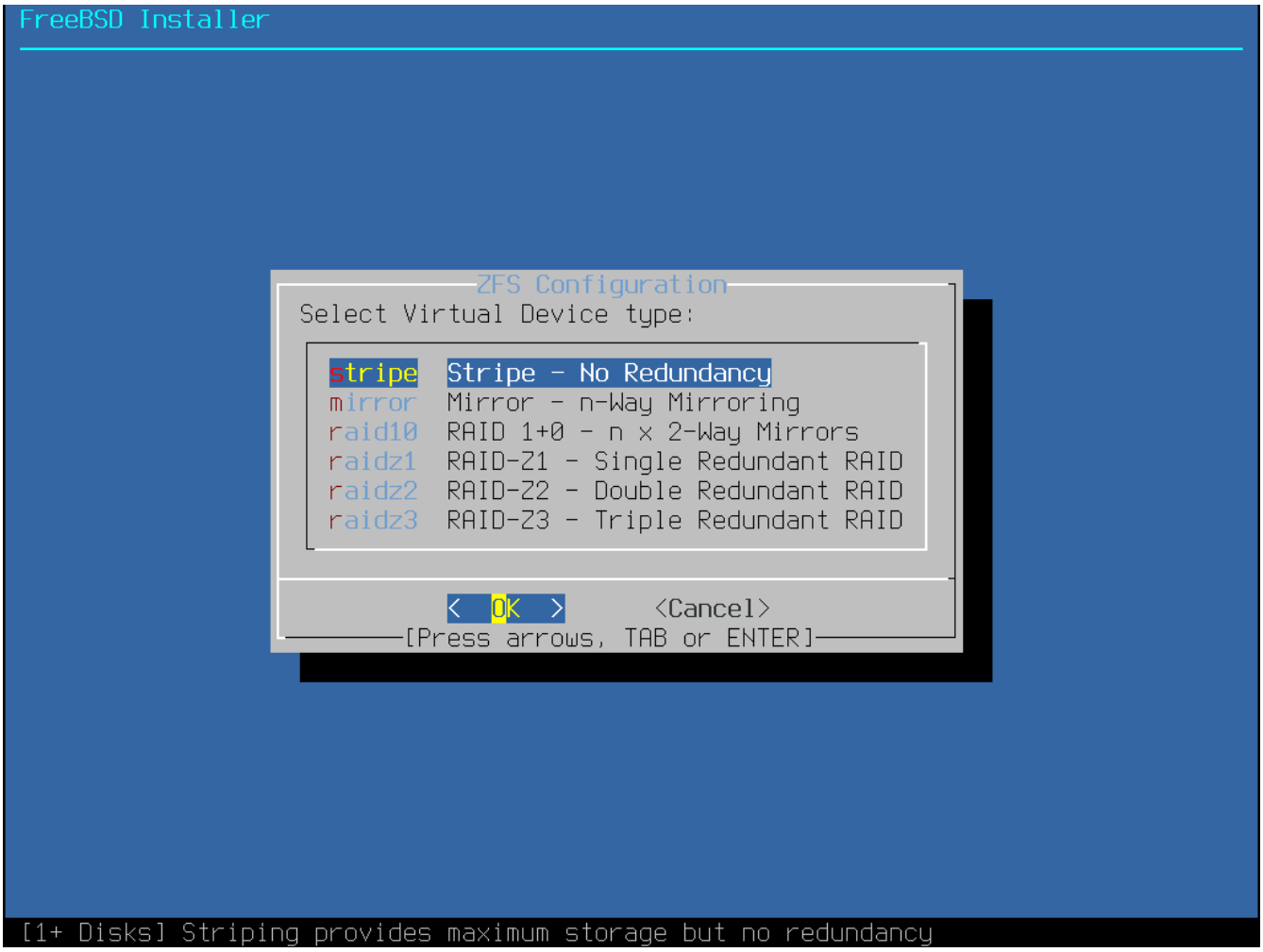


Figura 21. Tipo de pool ZFS

Aqui está um resumo dos `Pool Type` que podem ser selecionados neste menu:

- `stripe` - Striping provê a capacidade máxima de todos os dispositivos conectados, mas não redundância. Se um disco falhar os dados do pool estarão perdidos de forma irrevogável.

- `mirror` - O espelhamento armazena uma completa cópia de todos os dados em todos os discos. O espelhamento provê uma boa performance em leitura porque os dados são lidos the todos os discos em paralelo. A performance da escrita é mais lenta pois os dados precisam ser escritos em todos os discos do pool. Torna possível que haja falha nos discos, menos um. Esta opção requer ao menos dois discos.
- `raid10` - Striped mirrors. Provê a melhor performance, mas o menor armazenamento. Esta opção necessita de um número par de discos e no mínimo quatro discos.
- `raidz1` - RAID Único Redundante. Permite que haja falha concorrente de um disco. Esta opção necessita de ao menos três discos.
- `raidz2` - RAID Duplo Redundante. Permite que até dois discos falhem concorrentemente. Esta opção necessita de ao menos quatro discos.
- `raidz3` - RAID Triplo Redundante. Permite que até três discos falhem concorrentemente. Esta opção necessita de ao menos cinco discos.

Quando um `Pool Type` for selecionado, uma lista de discos disponíveis será exibida, e o usuário é solicitado a selecionar um ou mais discos para compor o pool. A configuração é então validada, para garantir que discos suficientes sejam selecionados. Caso contrário, selecione **[<Change Selection>]** para retornar à lista de discos ou **[<Backgt>]** para alterar o `Pool Type`

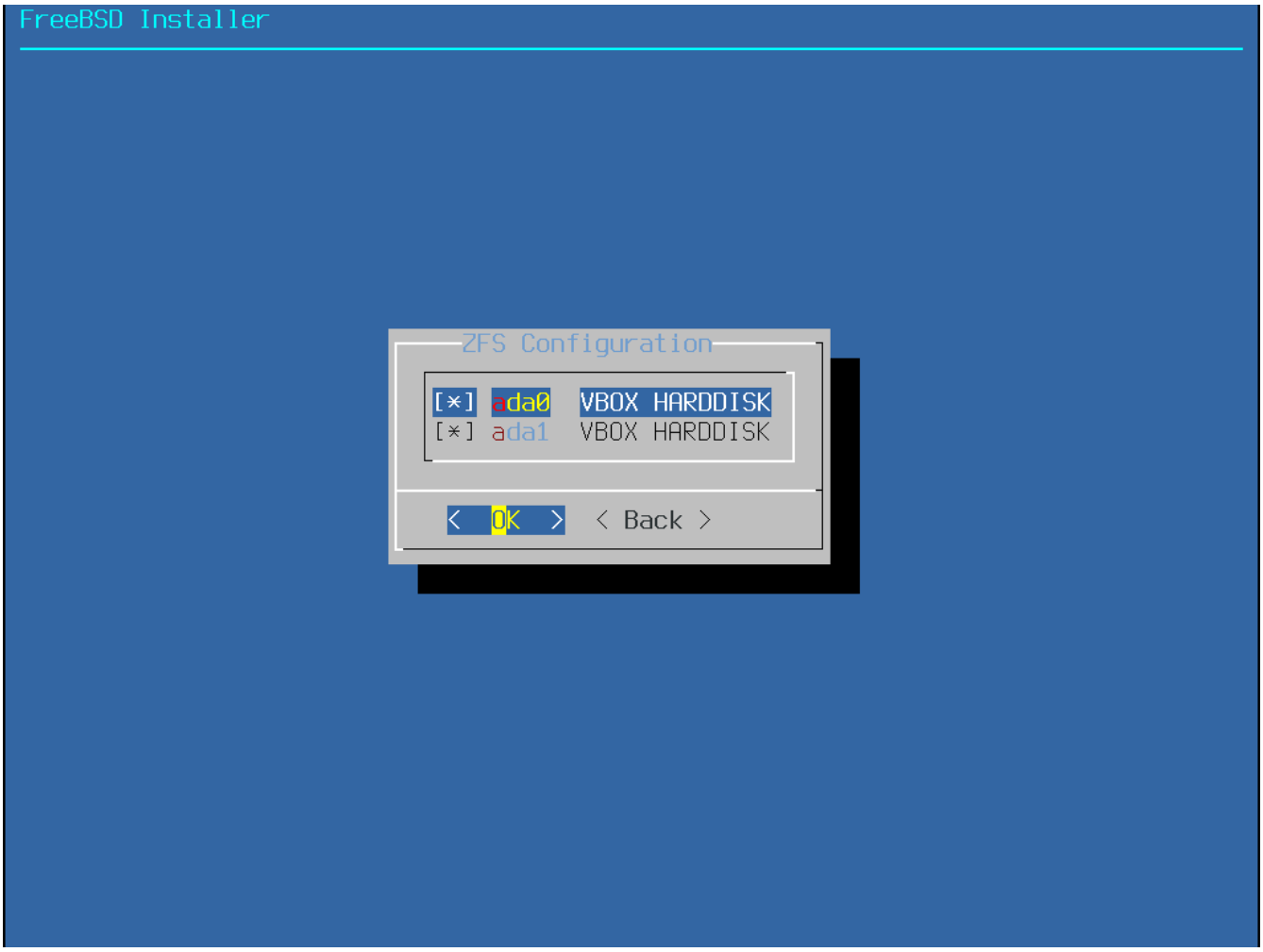


Figura 22. Seleção de disco

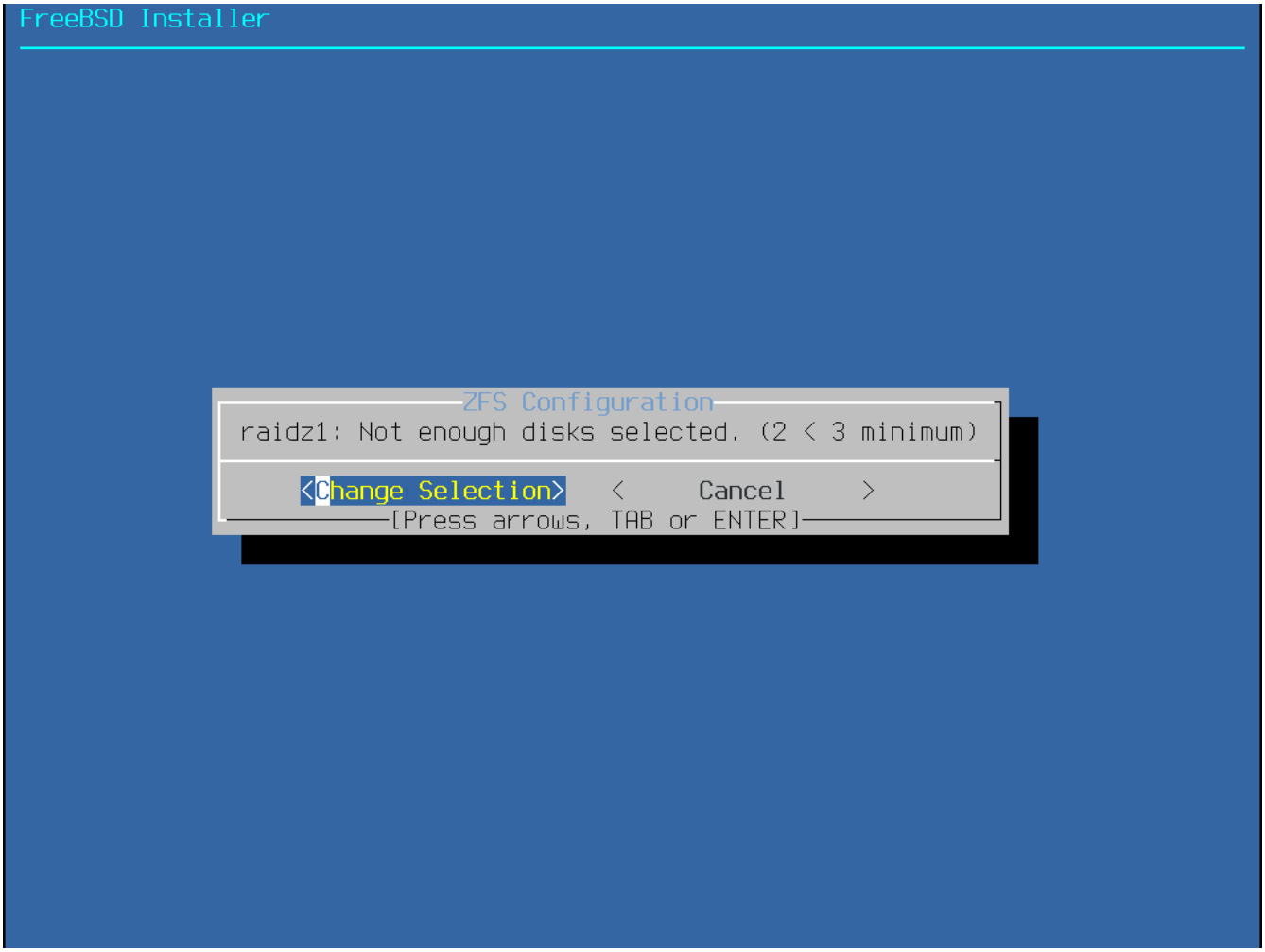


Figura 23. Seleção inválida

Se um ou mais discos estiverem faltando na lista, ou se os discos foram anexados depois que o instalador foi iniciado, selecione **[- Rescan Devices]** para preencher novamente a lista de discos disponíveis.

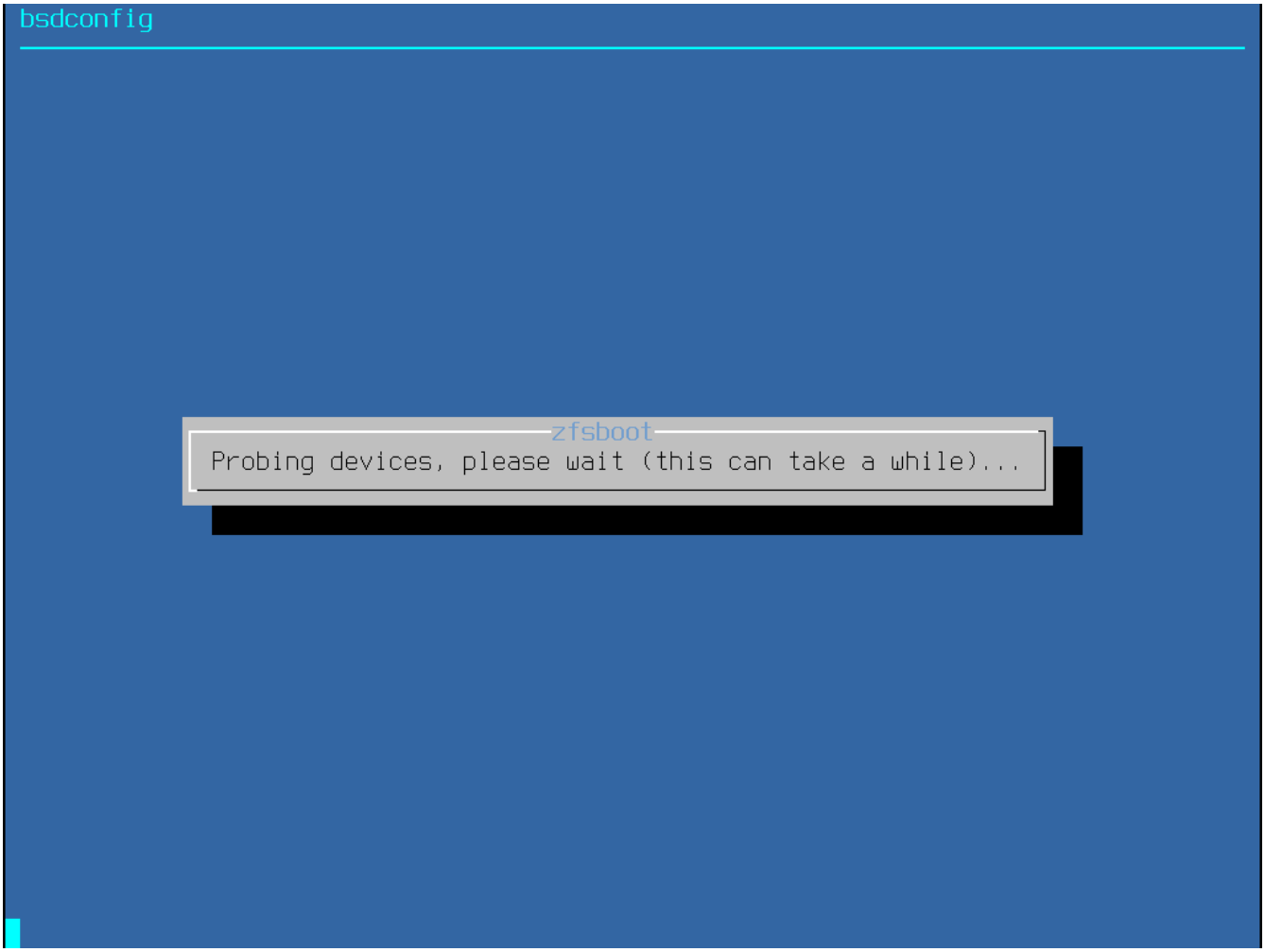


Figura 24. Dispositivos de Reescaneamento

Para evitar apagar acidentalmente o disco errado, o menu **[- Disk Info]** pode ser usado para inspecionar cada disco, incluindo sua tabela de partição e várias outras informações, como o número do modelo do dispositivo e o número de série, se disponíveis.

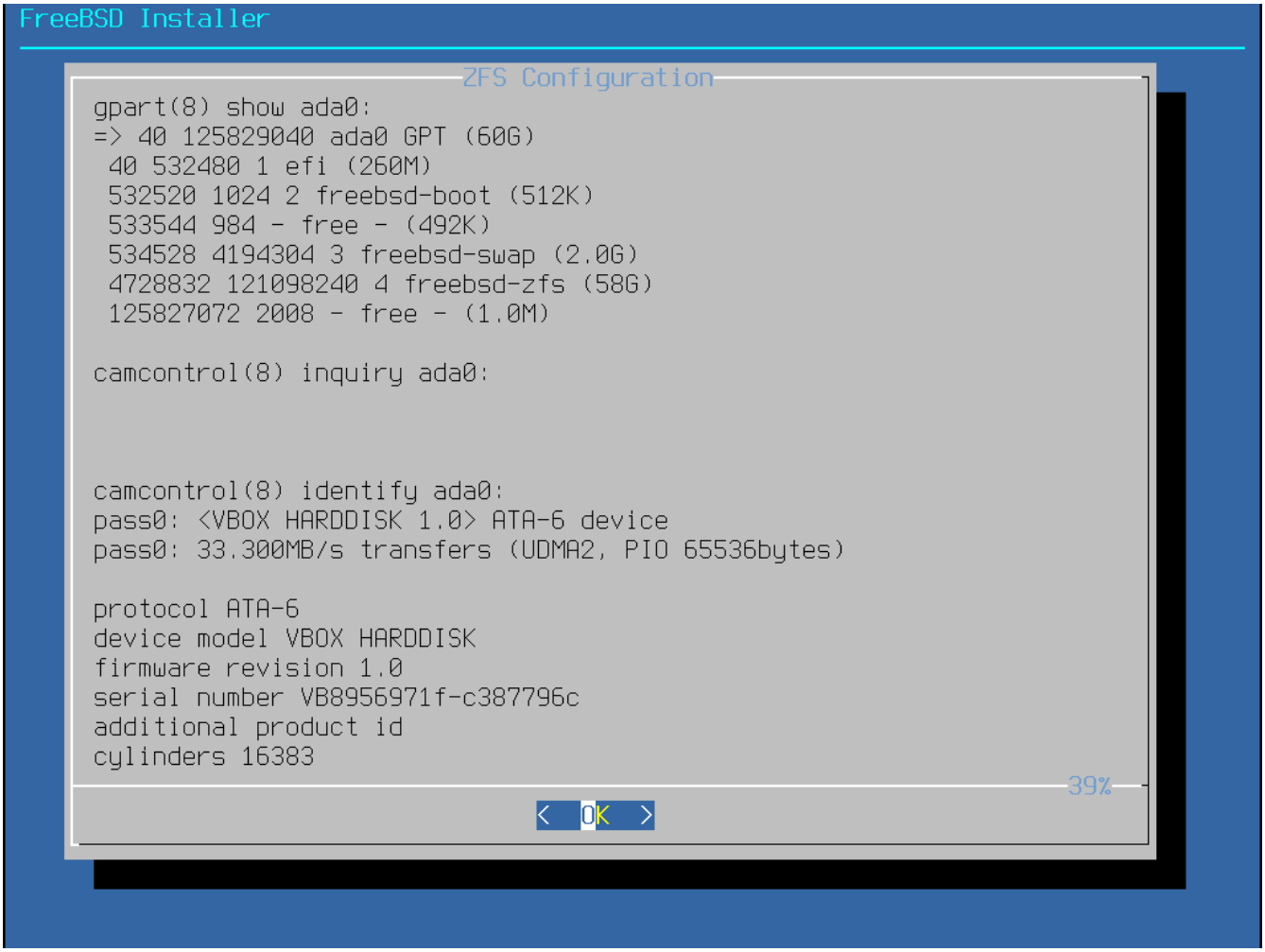


Figura 25. Analisando um disco

Selecione `N` para configurar o `Pool Name`. Entre com o nome desejado e então selecione **[<OK>]** para confirmar ou **[<Cancel>]** para retornar ao menu principal e deixar o nome padrão.

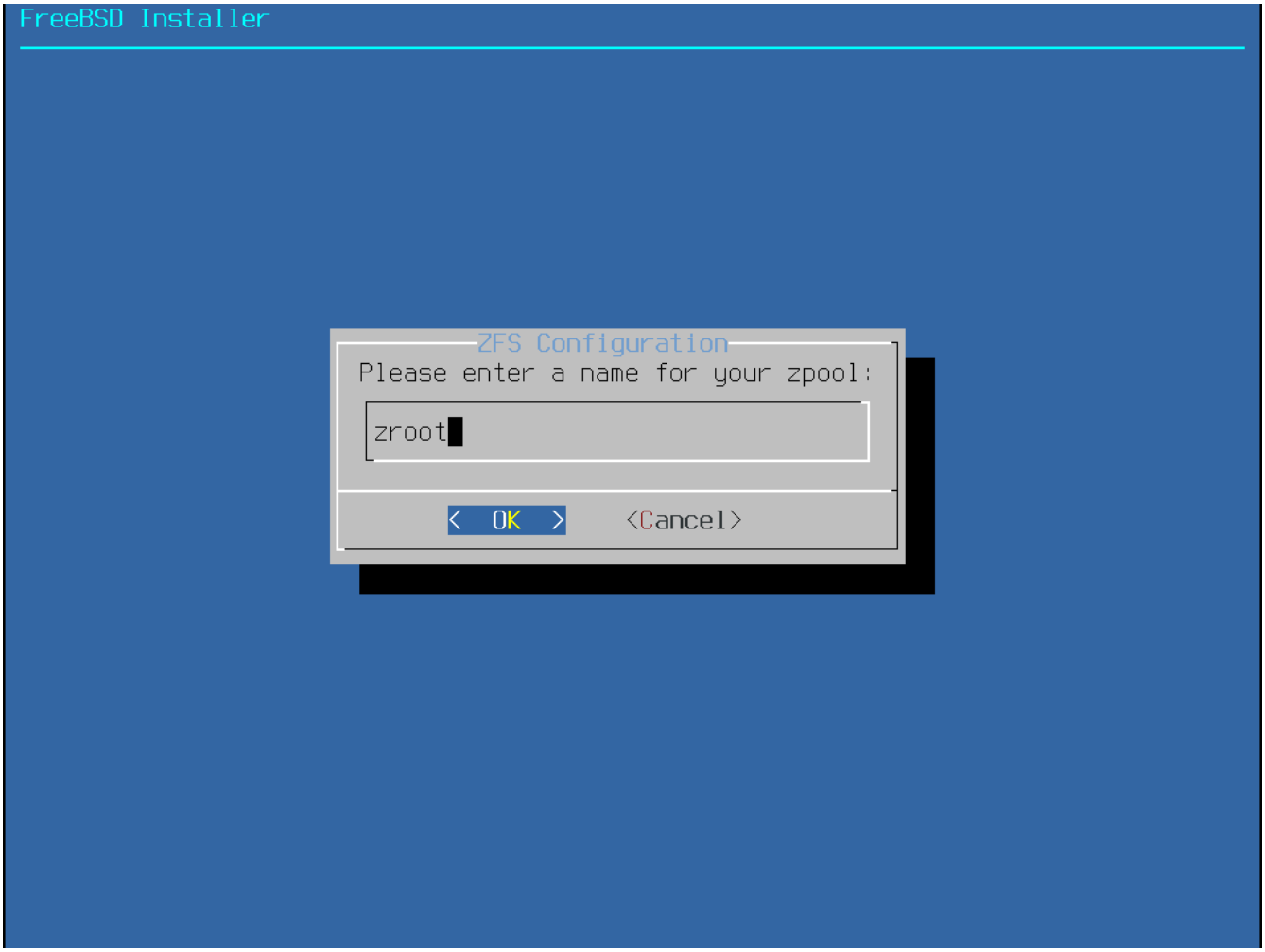


Figura 26. Nome do Pool

Selecione `S` para escolher a quantidade de swap. Entre com a quantidade desejada e então selecione **[<OK>]** para confirmar isto ou **[<Cancel>]** para retornar ao menu principal e deixar a quantidade padrão.

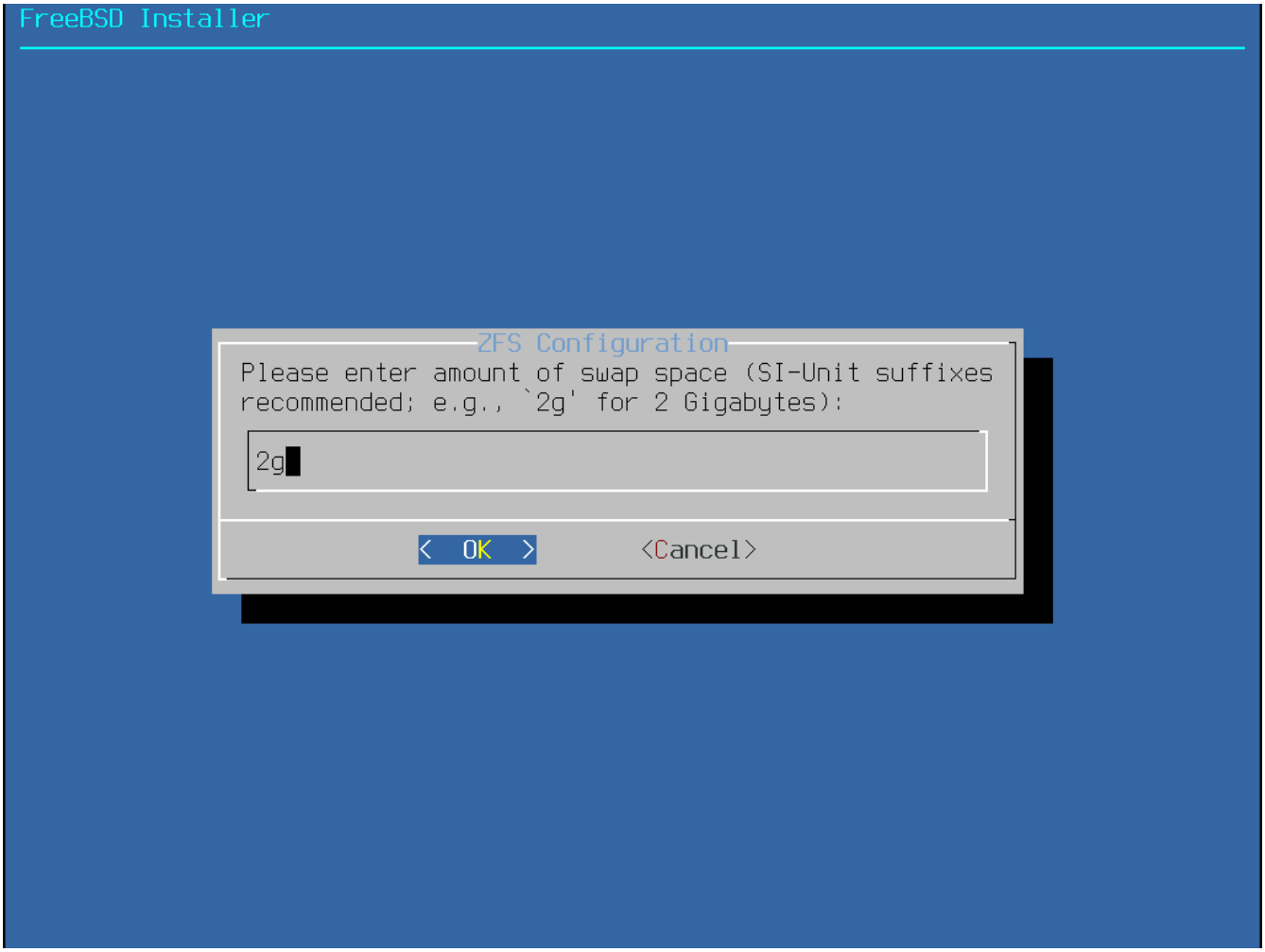


Figura 27. Quantidade de Swap

Uma vez que todas opções estejam setadas com os valores desejados, selecione a opção **[>>> Install]** no topo do menu. O instalador oferece uma última chance de cancelar antes que o conteúdo das unidades selecionadas seja destruído para criar o pool do ZFS.

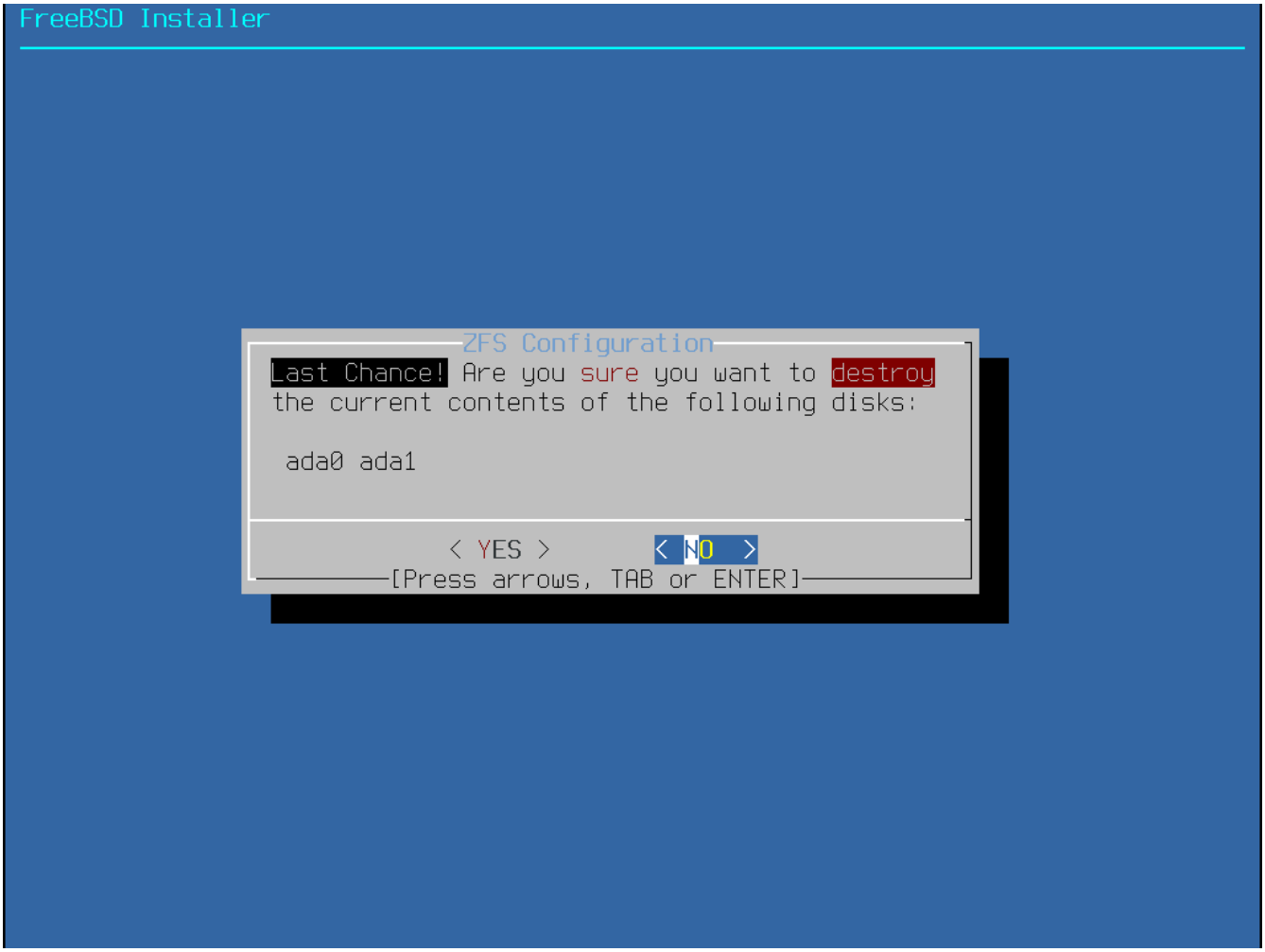


Figura 28. Última chance

Se a criptografia de disco GELI foi ativada, o instalador solicitará duas vezes que a frase secreta seja usada para criptografar os discos. E depois dissoa inicialização da criptografia é iniciado.

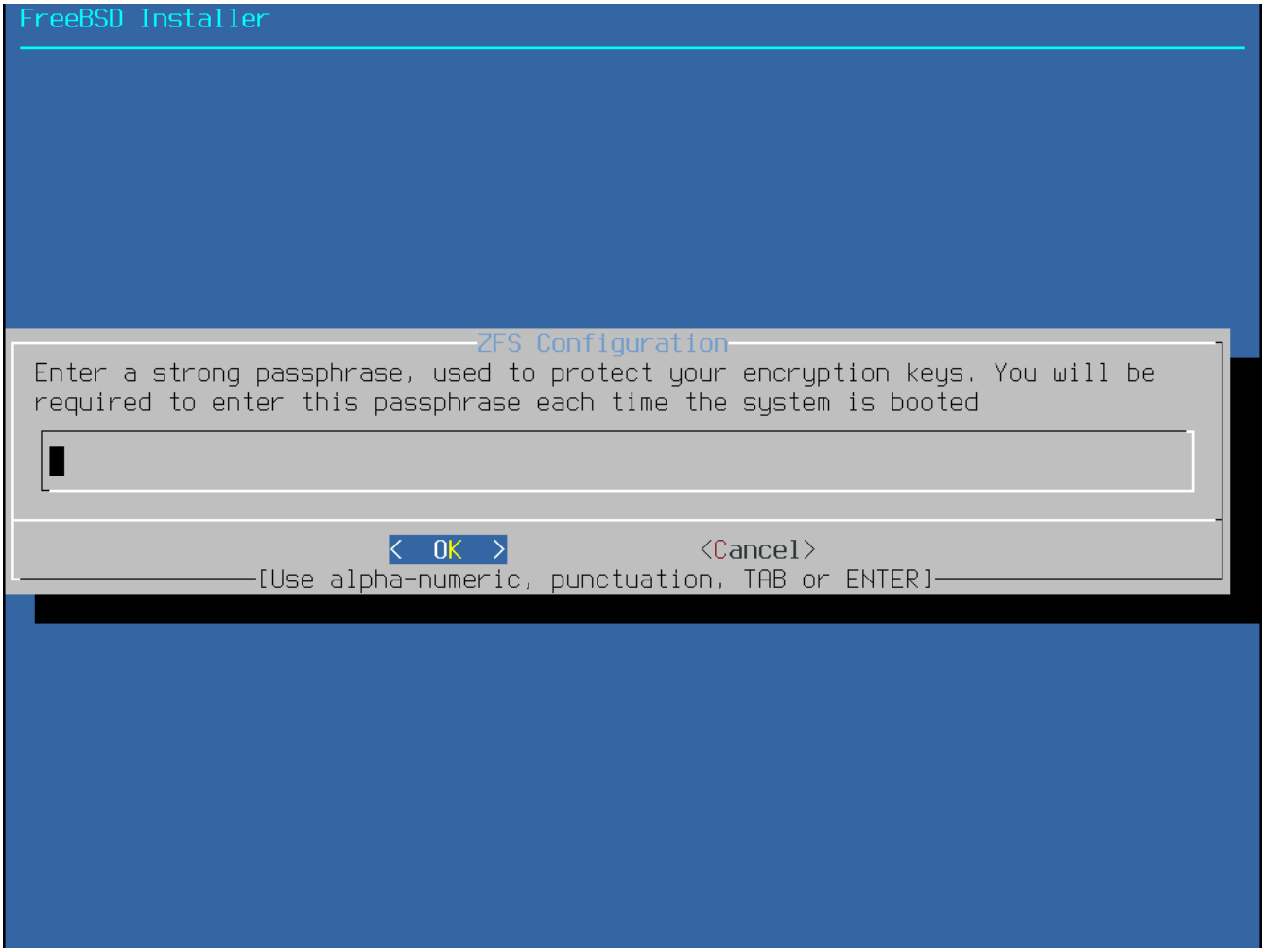


Figura 29. Senha de criptografia de disco

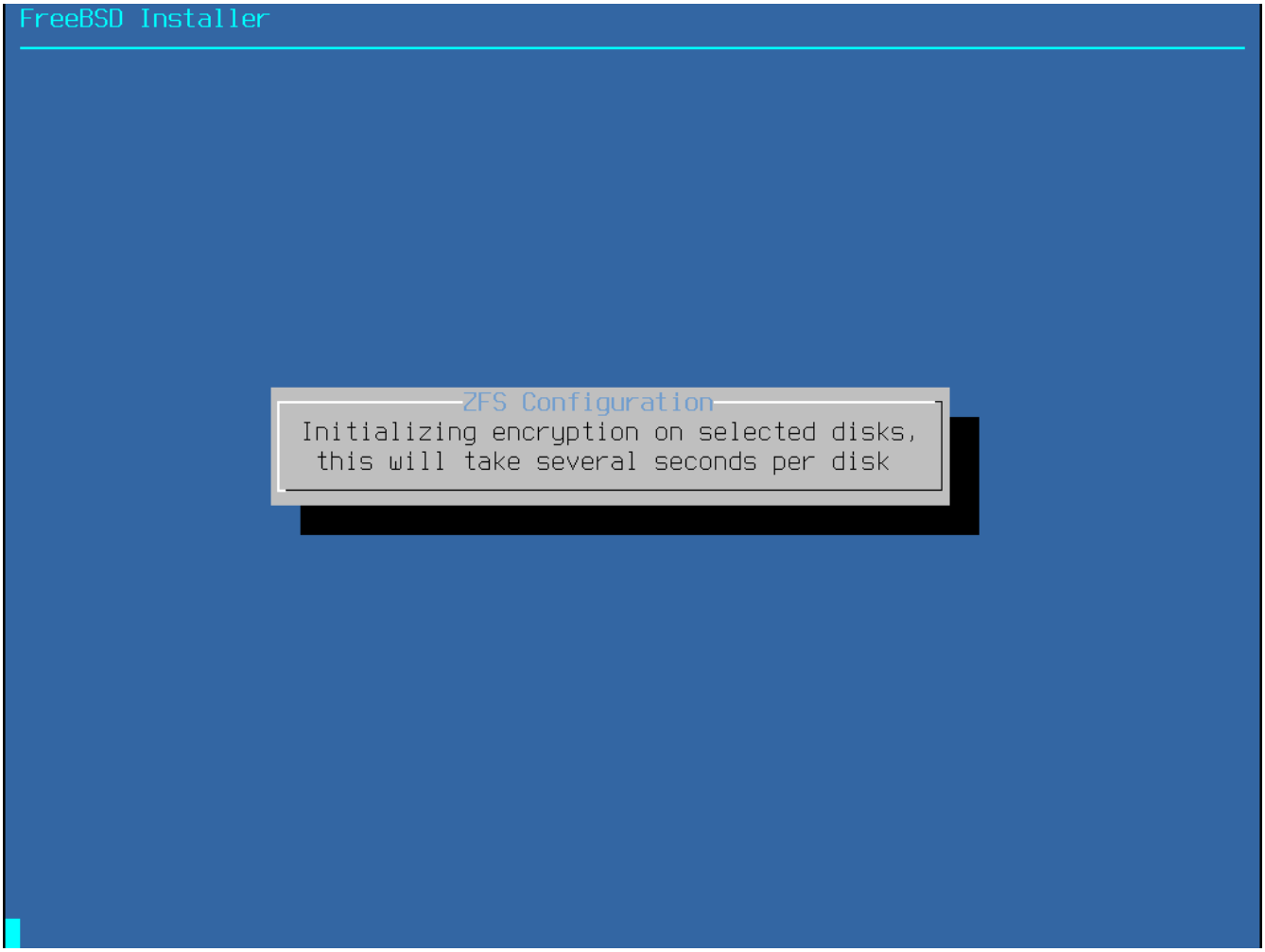


Figura 30. inicializando Criptografia

A instalação então prossegue normalmente. Para continuar com a instalação, vá para Fazendo o download dos arquivos de distribuição.

2.6.5. Particionamento do modo shell

Ao criar instalações avançadas, os menus de particionamento do bsdinstall podem não fornecer o nível de flexibilidade necessário. Usuários avançados podem selecionar a opção **[Shell]** no menu de particionamento para particionar manualmente as unidades, criar o(s) sistema(s) de arquivos, preencher o /tmp/bsdinstall_etc/fstab e montar os sistemas de arquivos em /mnt. Feito isso, digite `exit` para retornar ao bsdinstall e continue com a instalação.

2.7. Fazendo o download dos arquivos de distribuição

O tempo de instalação irá variar dependendo das distribuições escolhidas, mídia de instalação e velocidade do computador. Uma série de mensagens indicará o progresso.

Primeiro, o instalador formata o(s) disco(s) selecionado(s) e inicializa as partições. Em seguida, no caso de uma `bootonly media` ou `mini memstick`, ele faz o download dos componentes selecionados:

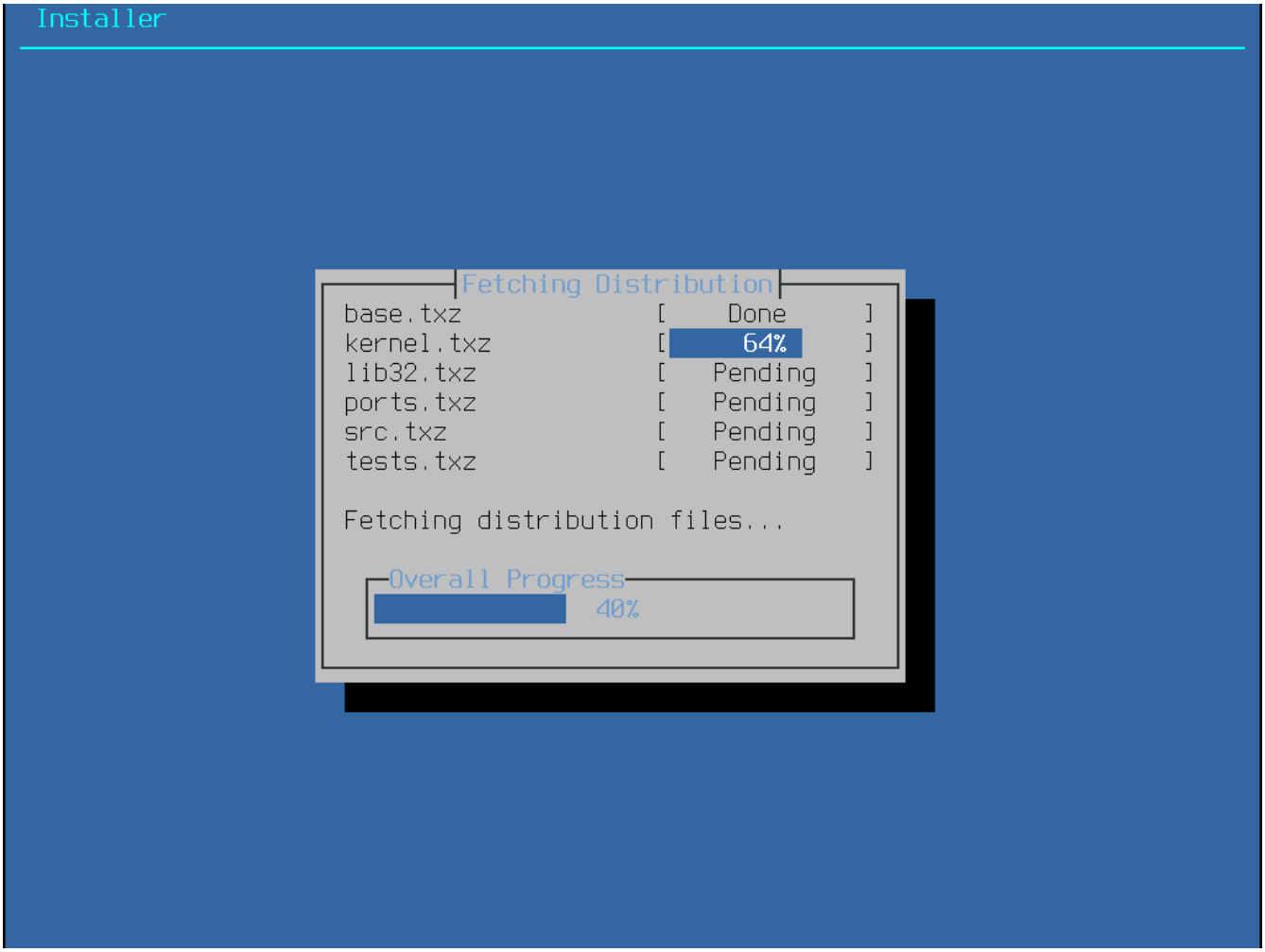


Figura 31. Fazendo o download dos arquivos de distribuição

Em seguida, a integridade dos arquivos de distribuição é verificada para garantir que eles não tenham sido corrompidos durante o download ou mal interpretados da mídia de instalação:

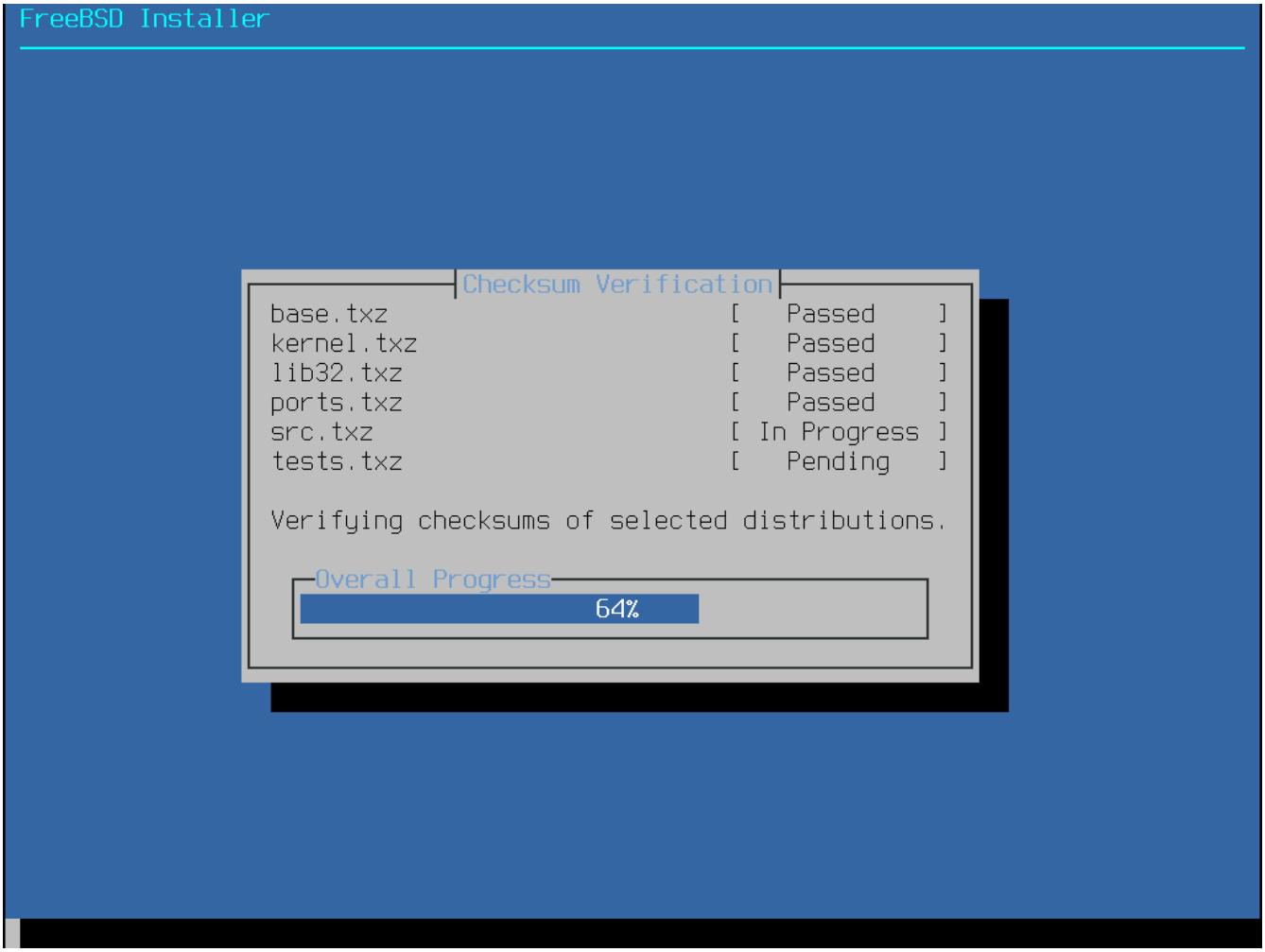


Figura 32. Verificando arquivos de distribuição

Finalmente, os arquivos de distribuição verificados são extraídos para o disco:

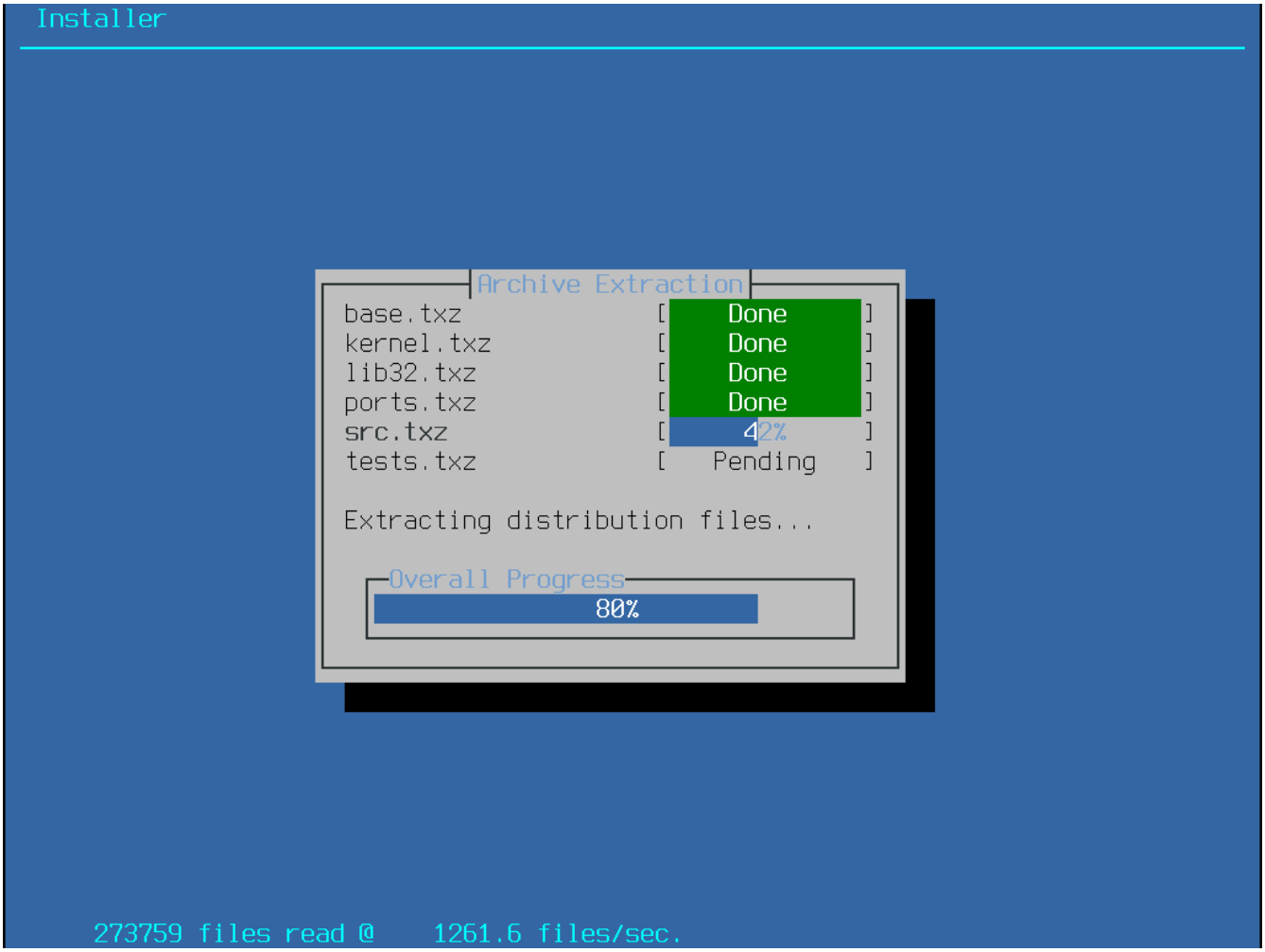


Figura 33. Extraindo arquivos de distribuição

Depois que todos os arquivos de distribuição solicitados tiverem sido extraídos, o bsdinstall exibirá a primeira tela de configuração pós-instalação. As opções de configuração pós-instalação disponíveis estão descritas na próxima seção.

2.8. Contas, Time Zone, Serviços e Hardening

2.8.1. Definindo a Senha de `root`

Primeiro, a senha do `root` deve ser definida. Ao digitar a senha, os caracteres digitados não são exibidos na tela. Depois que a senha for digitada, ela deve ser digitada novamente. Isso ajuda a evitar erros de digitação.

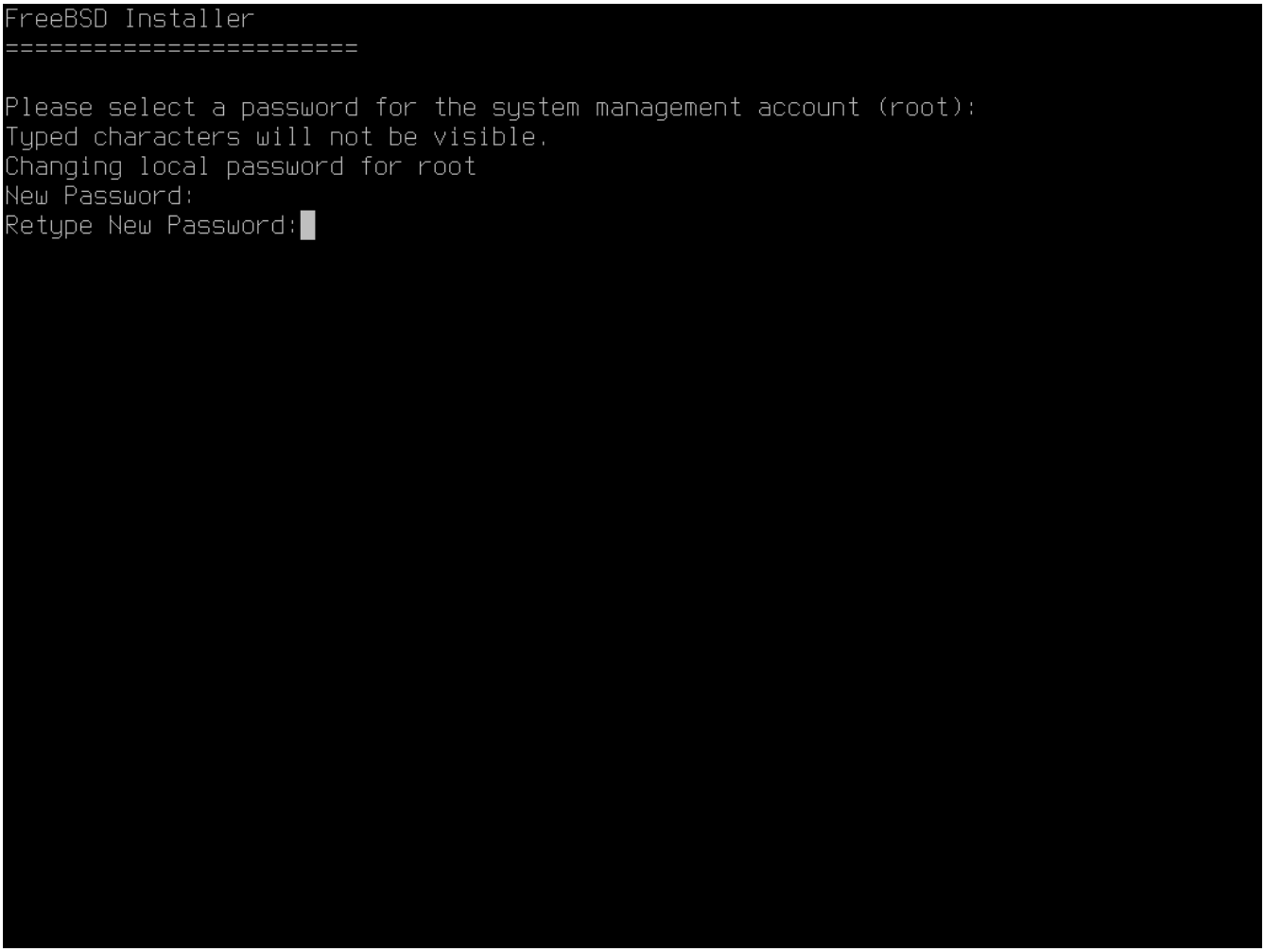


Figura 34. Definindo a Senha de `root`

2.8.2. Defina o fuso horário

A próxima série de menus é usada para determinar a hora local correta, selecionando a região geográfica, o país e o fuso horário. Definir o fuso horário permite que o sistema corrija automaticamente as alterações de horário regionais, como horário de verão, e execute outras funções relacionadas ao fuso horário corretamente.

O exemplo mostrado aqui é para uma máquina localizada no fuso horário do continente da Espanha, Europa. As seleções variam de acordo com a localização geográfica.

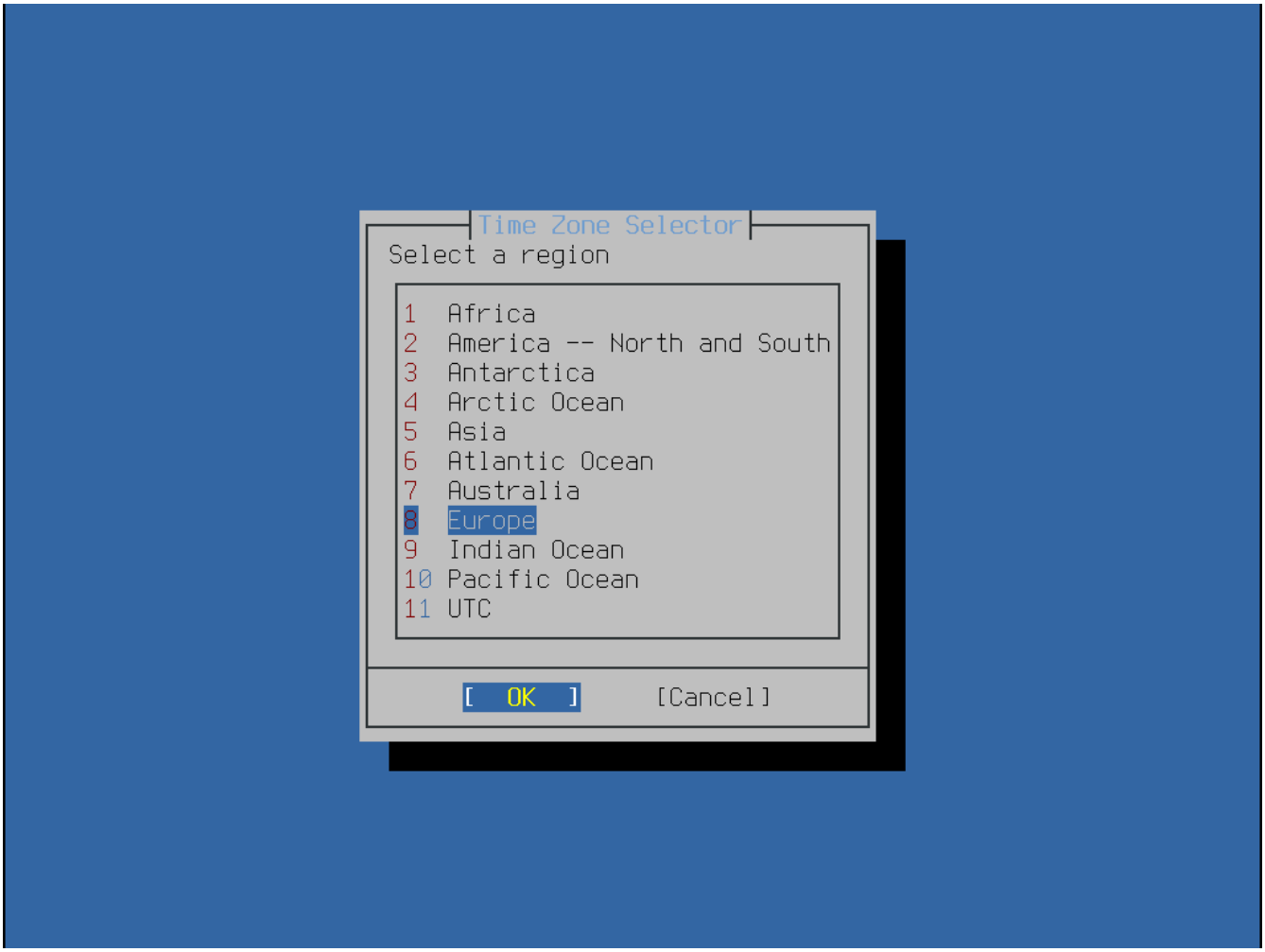


Figura 35. Selecione uma região

A região apropriada é selecionada usando as teclas de seta e depois pressionando `Enter`.

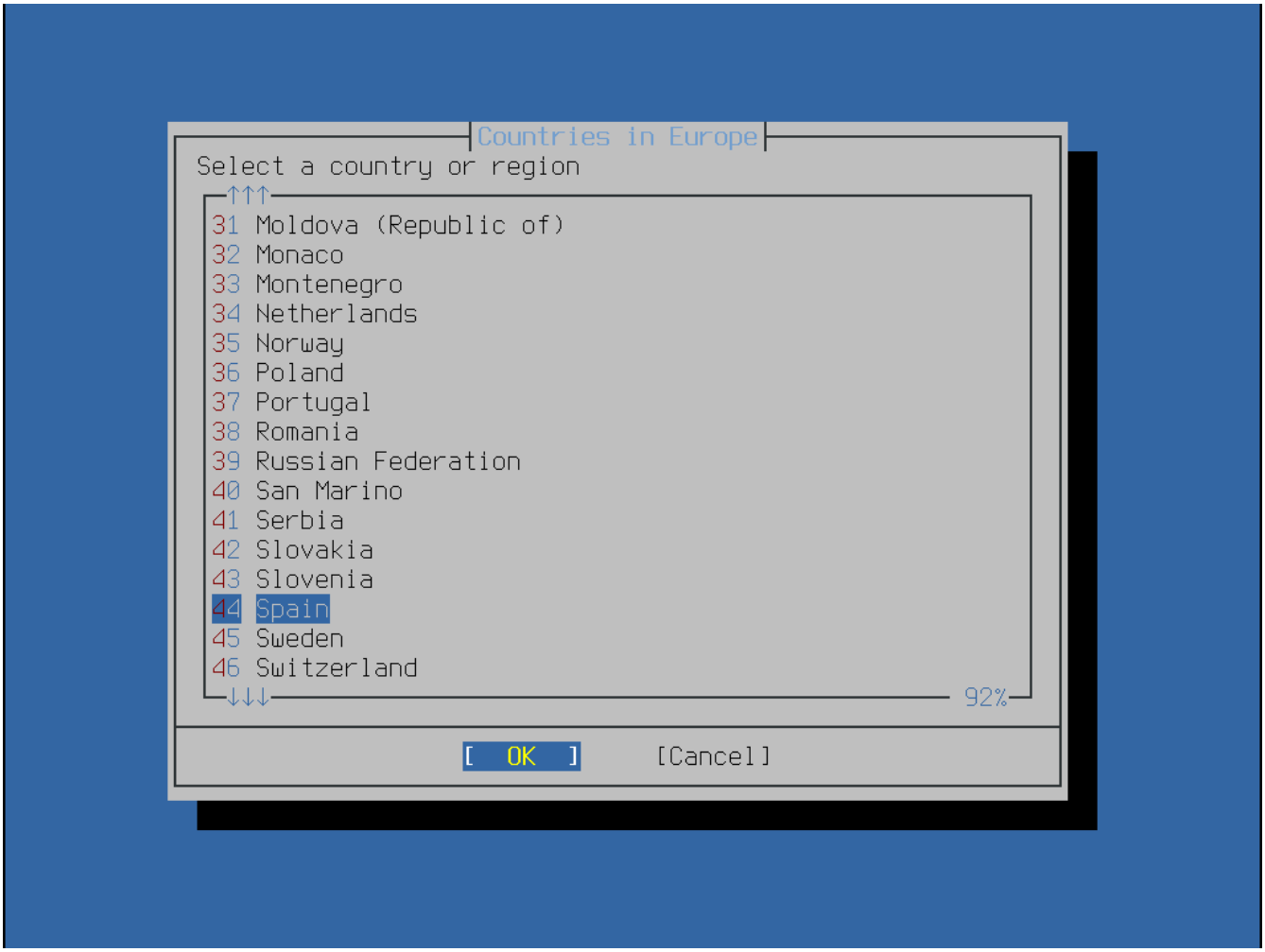


Figura 36. Selecione um pais

Selecione o país apropriado usando as teclas de seta e pressione `Enter`.

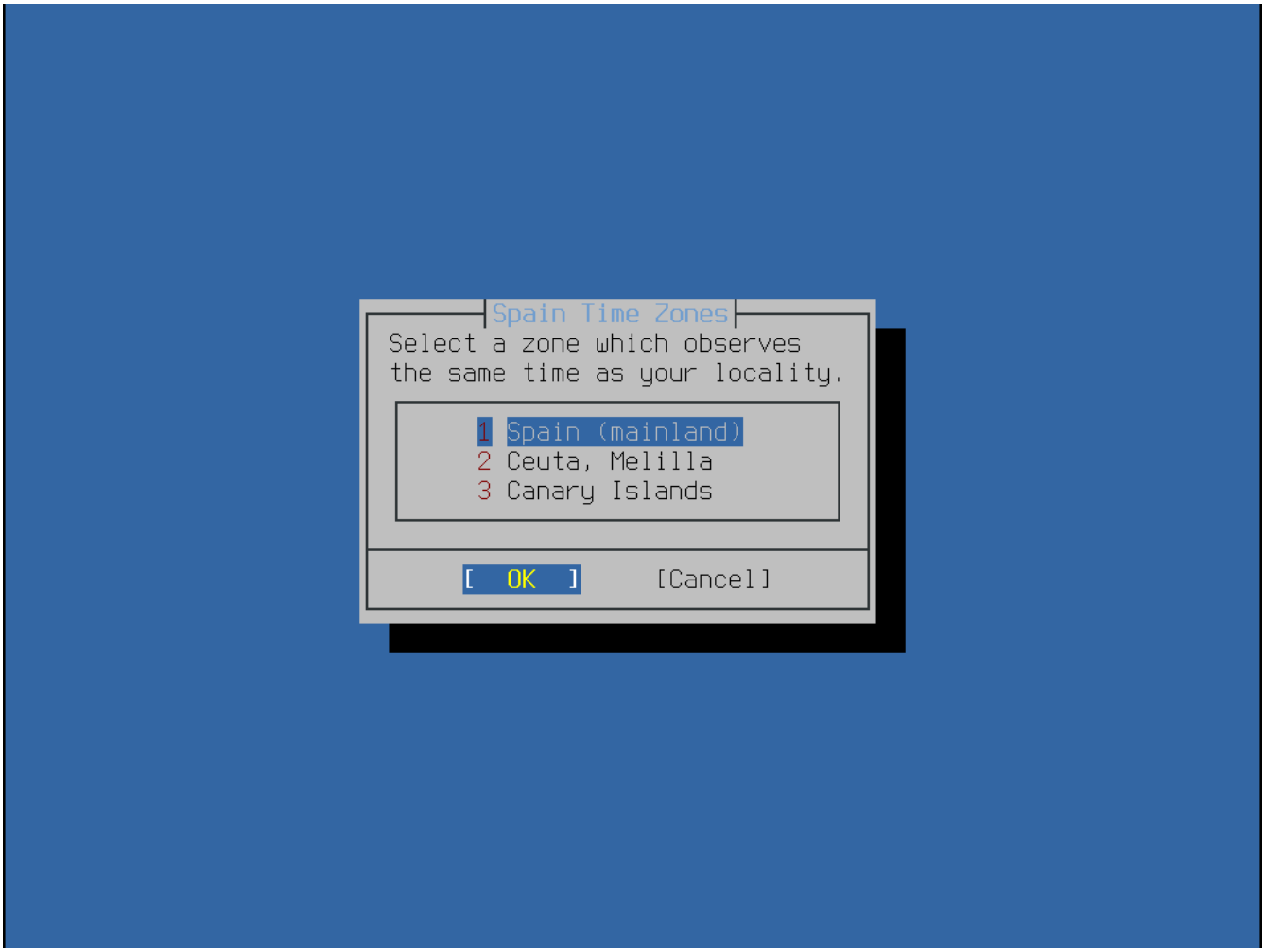


Figura 37. Selecione um fuso horário

O fuso horário apropriado é selecionado usando as teclas de seta e pressionando `Enter`.

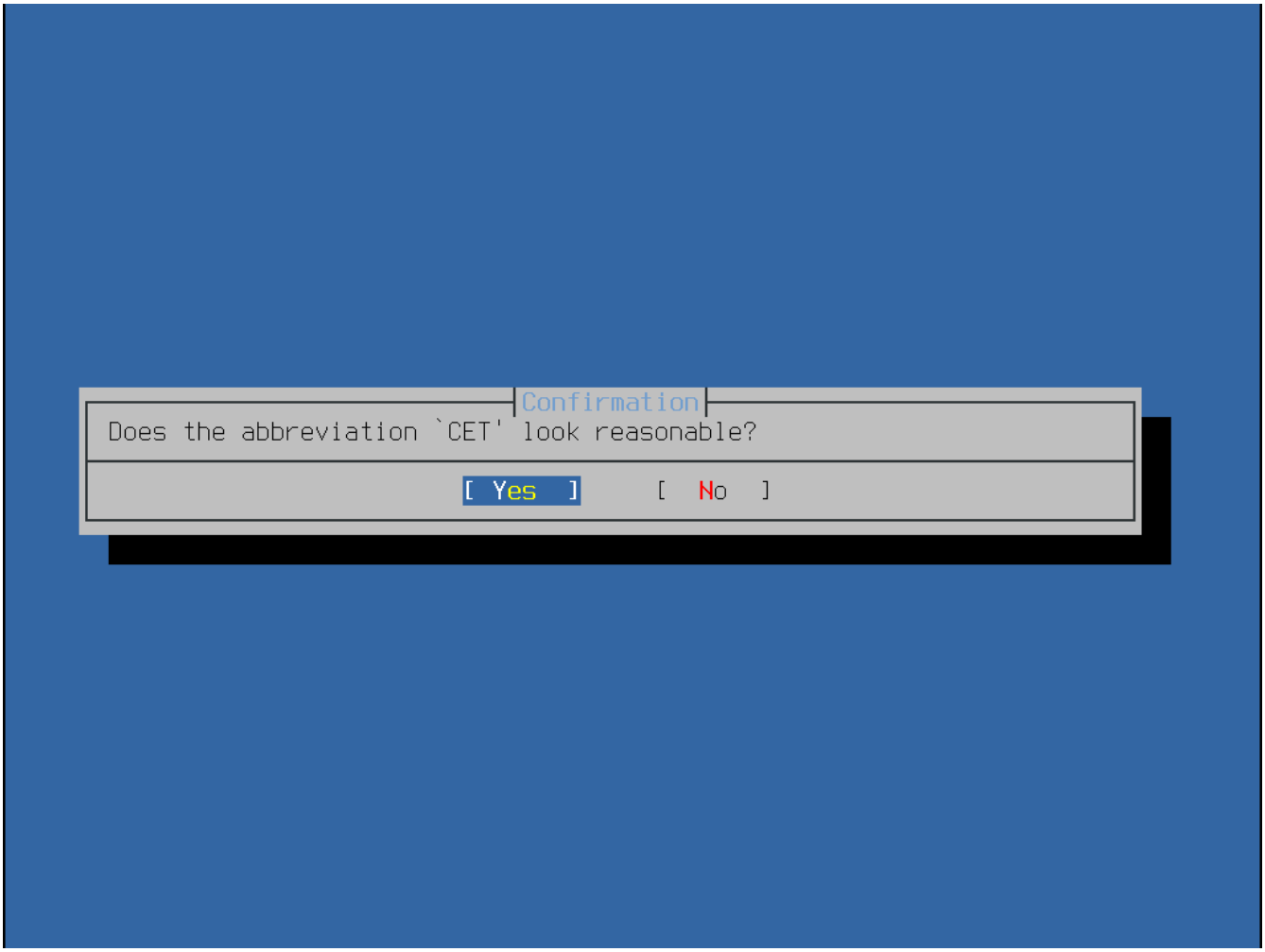


Figura 38. Confirme o fuso horário

Confirme se a abreviação do fuso horário está correta.

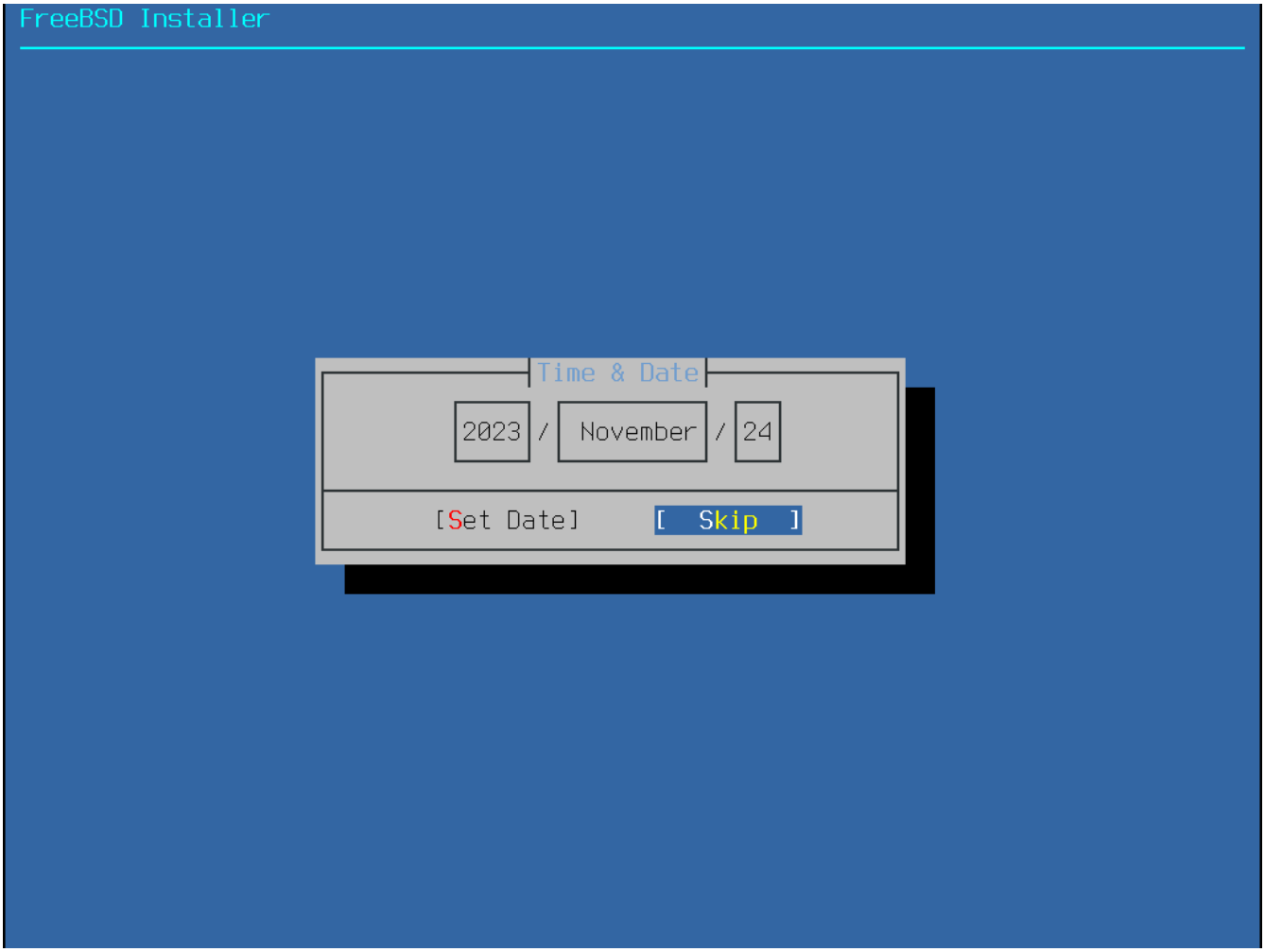


Figura 39. Selecionar Data

A data apropriada é selecionada usando as teclas de seta e pressionando **[Set Date]**. Caso contrário, a seleção de data pode ser pulada pressionando **[Skip]**.

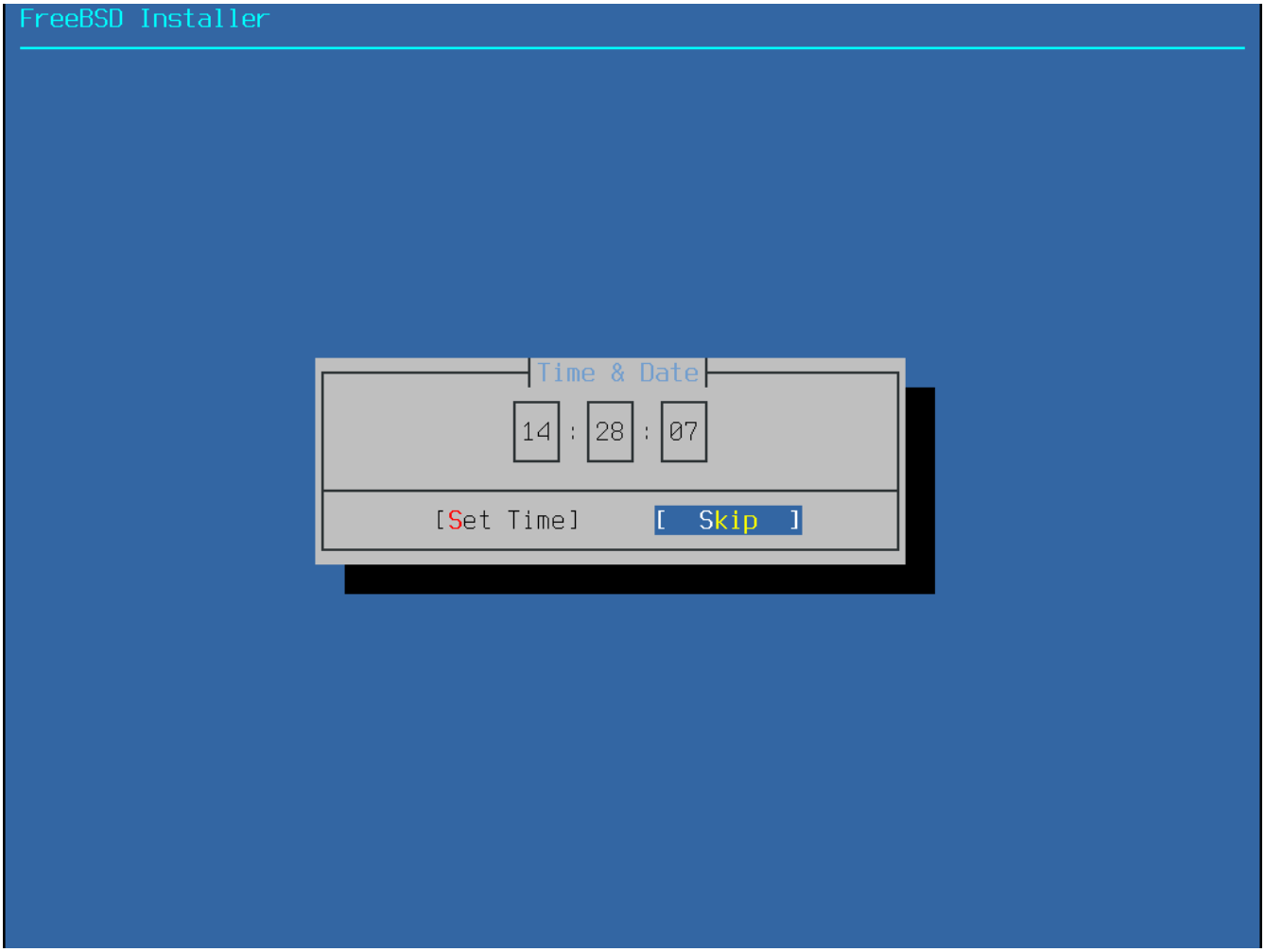


Figura 40. Selecionar Hora

O horário apropriado é selecionado usando as teclas de seta e, em seguida, pressionando **[Set Time]**. Caso contrário, a seleção da hora pode ser pulada pressionando **[Skip]**.

2.8.3. Ativando Serviços

O próximo menu é usado para configurar quais serviços do sistema serão iniciados sempre que o sistema for inicializado. Todos esses serviços são opcionais. Inicie apenas os serviços necessários para o funcionamento do sistema.

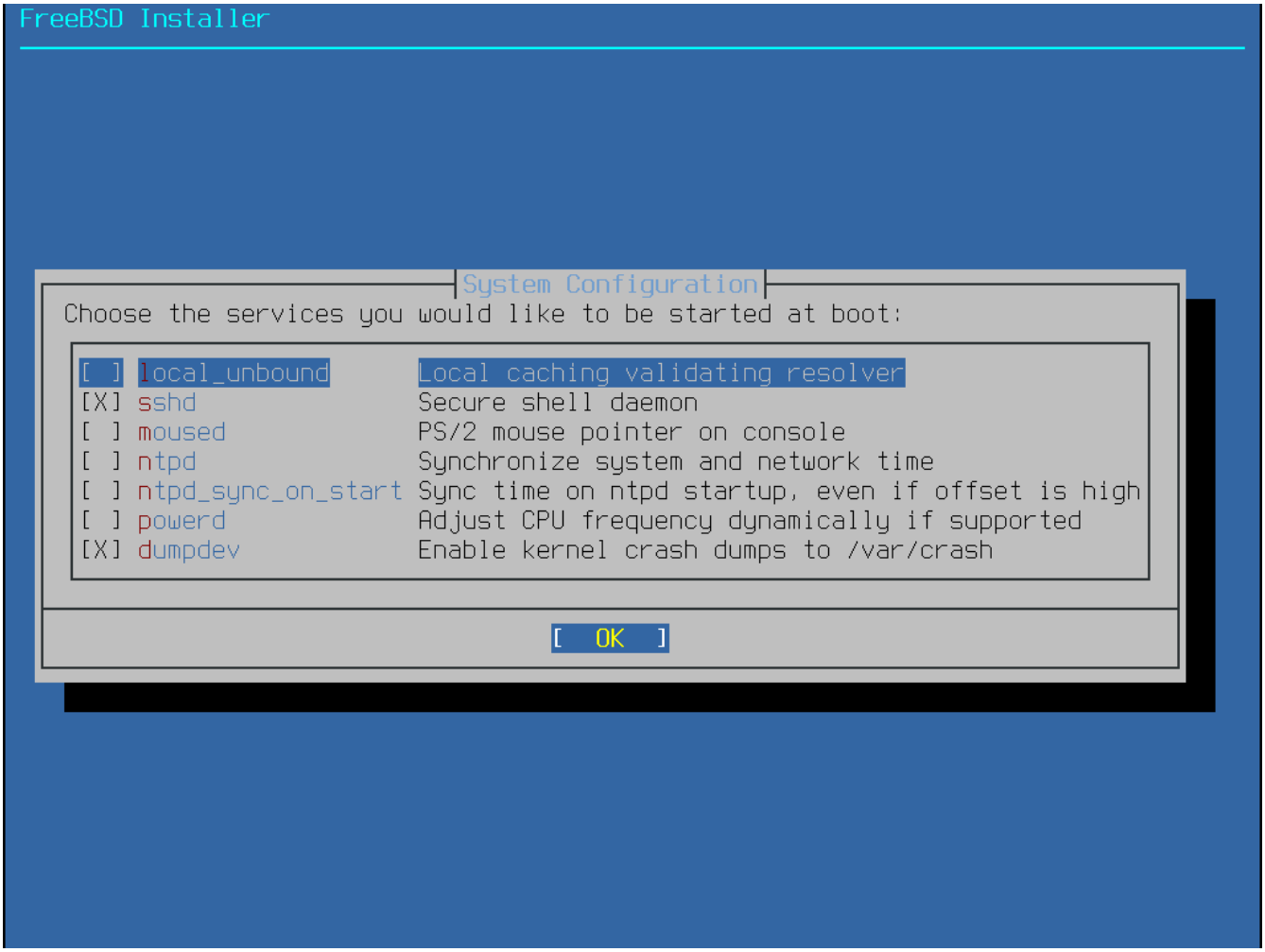


Figura 41. Selecionando Serviços Adicionais para Ativar

Aqui está um resumo dos serviços que podem ser ativados neste menu:

- `local_unbound` -Ative o DNS local unbound. É necessário ter em mente que esse é o unbound do sistema base e deve ser usado apenas como um cache local de consultas DNS. Se o objetivo é configurar um resolvedor para toda a rede, instale dns/unbound.
- `sshd` - O daemon Secure Shell (SSH) é usado para acessar remotamente um sistema através de uma conexão criptografada. Ative este serviço somente se o sistema estiver disponível para logins remotos.
- `moused` - Ative este serviço se o mouse for usado a partir do console do sistema de linha de comando.
- `ntpdate` - Ative a sincronização automática do relógio no momento do boot. A funcionalidade deste programa agora está disponível no daemon ntpd(8). Após um período considerável de luto, o utilitário ntpdate(8) será aposentado.
- `ntpd` - O daemon do Network Time Protocol (NTP) para sincronização automática do relógio. Ative este serviço se houver um servidor Windows™, Kerberos ou LDAP na rede.
- `powerd` - Utilitário de controle de energia do sistema para controle de energia e economia de energia.
- `dumpdev` - A habilitação de despejos de memória é útil na depuração de problemas com o sistema; portanto, os usuários são incentivados a habilitar despejos de memória.

2.8.4. Ativando Opções de Segurança (Hardening)

O próximo menu é usado para configurar quais opções de segurança serão ativadas. Todas essas opções são opcionais. Mas seu uso é incentivado.

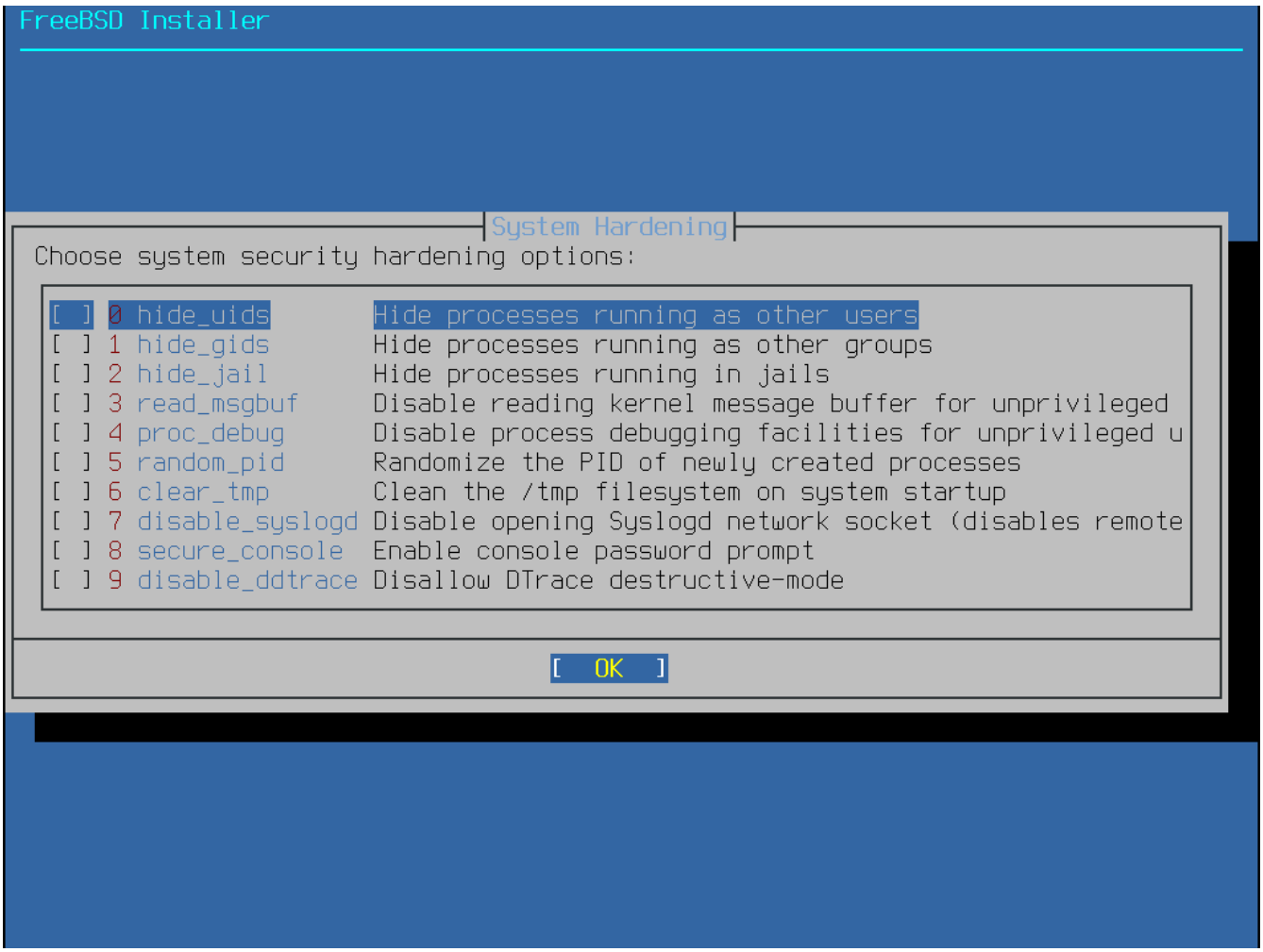


Figura 42. Selecionando Opções de Segurança (Hardening)

Aqui está um resumo das opções que podem ser ativadas neste menu:

- `hide_uids` - Oculta processos em execução de outros usuários para impedir que usuários sem privilégios vejam processos em execução de outros usuários (UID), impedindo o vazamento de informações.
- `hide_gids` - Oculta processos em execução de outros grupos para impedir que usuários sem privilégios vejam processos em execução de outros grupos (GID), impedindo o vazamento de informações.
- `hide_jail` - Oculta processos em execução em jails para impedir que usuários sem privilégios vejam processos em execução dentro das jails.
- `read_msgbuf` - Desativando a leitura do buffer de mensagens do kernel para usuários sem privilégios, impede o uso do dmesg(8) para exibir mensagens do log do kernel em buffer.
- `proc_debug` - Desativar os recursos de depuração de processo para usuários sem privilégios desativa uma variedade de serviços de depuração entre processos sem privilégios, incluindo algumas funcionalidades procfs, ptrace() e ktrace(). Observe que isso também irá bloquear ferramentas de depuração, como por exemplo, lldb(1), truss(1), procstat(1), bem como alguns

recursos de depuração integrados em certas linguagens de script como PHP, etc., de funcionar para usuários sem privilégios.

- `random_pid` - Randomize o PID dos processos recém-criados.
- `clear_tmp` - Limpar o /tmp na inicialização do sistema.
- `disable_syslogd` - Desative a criação de socket de rede do syslogd. Por padrão, o FreeBSD executa o syslogd de maneira segura com `-s`. Isso impede que o daemon atenda solicitações UDP recebidas na porta 514. Com esta opção ativada, o syslogd será executado com o sinalizador `-ss`, que impede o syslogd de abrir qualquer porta. Para obter mais informações, consulte syslogd(8).
- `disable_sendmail` - Desative o agente de transporte de email sendmail.
- `secure_console` - Quando esta opção está ativada, o prompt solicita a senha de `root` ao entrar em modo single-user.
- `disable_ddtrace` - O DTrace pode ser executado em um modo que realmente afetará o kernel em execução. Ações destrutivas não podem ser usadas, a menos que tenham sido explicitamente ativadas. Para habilitar esta opção ao usar o DTrace, use `-w`. Para obter mais informações, consulte dtrace(1).

2.8.5. Adicione usuários

O próximo menu pede para criar pelo menos uma conta de usuário. Recomenda-se fazer login no sistema usando uma conta de usuário em vez de utilizar diretamente o `root`. Quando logado como `root`, essencialmente não há limites ou proteção sobre o que pode ser feito. Fazer o login como um usuário normal é mais seguro.

Selecione **[Yes]** para adicionar novos usuários.

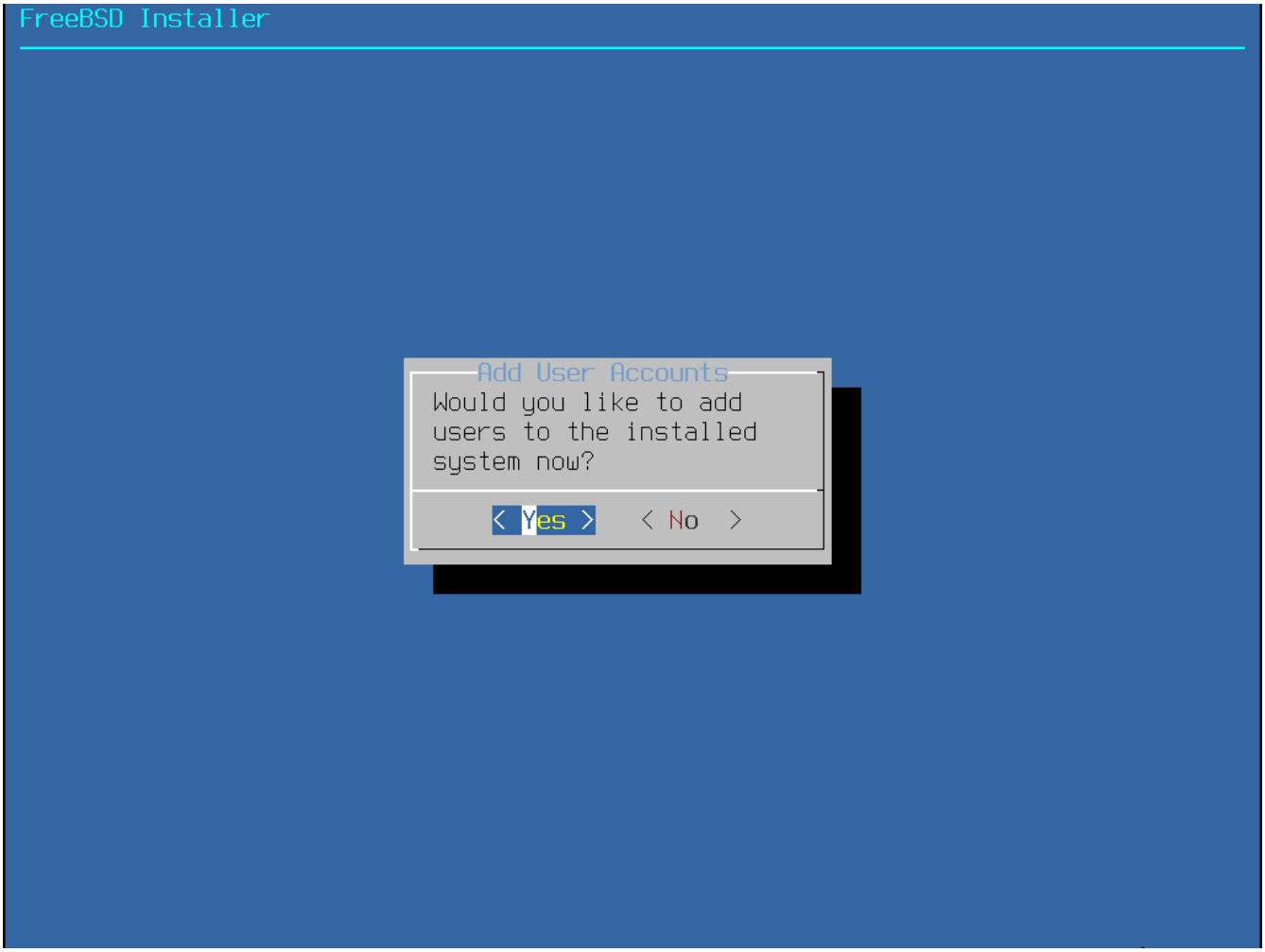


Figura 43. Adicione contas de usuário

Siga os prompts e insira as informações solicitadas para a conta do usuário. O exemplo mostrado em Insira as informações do usuário cria a conta de usuário `asample`.

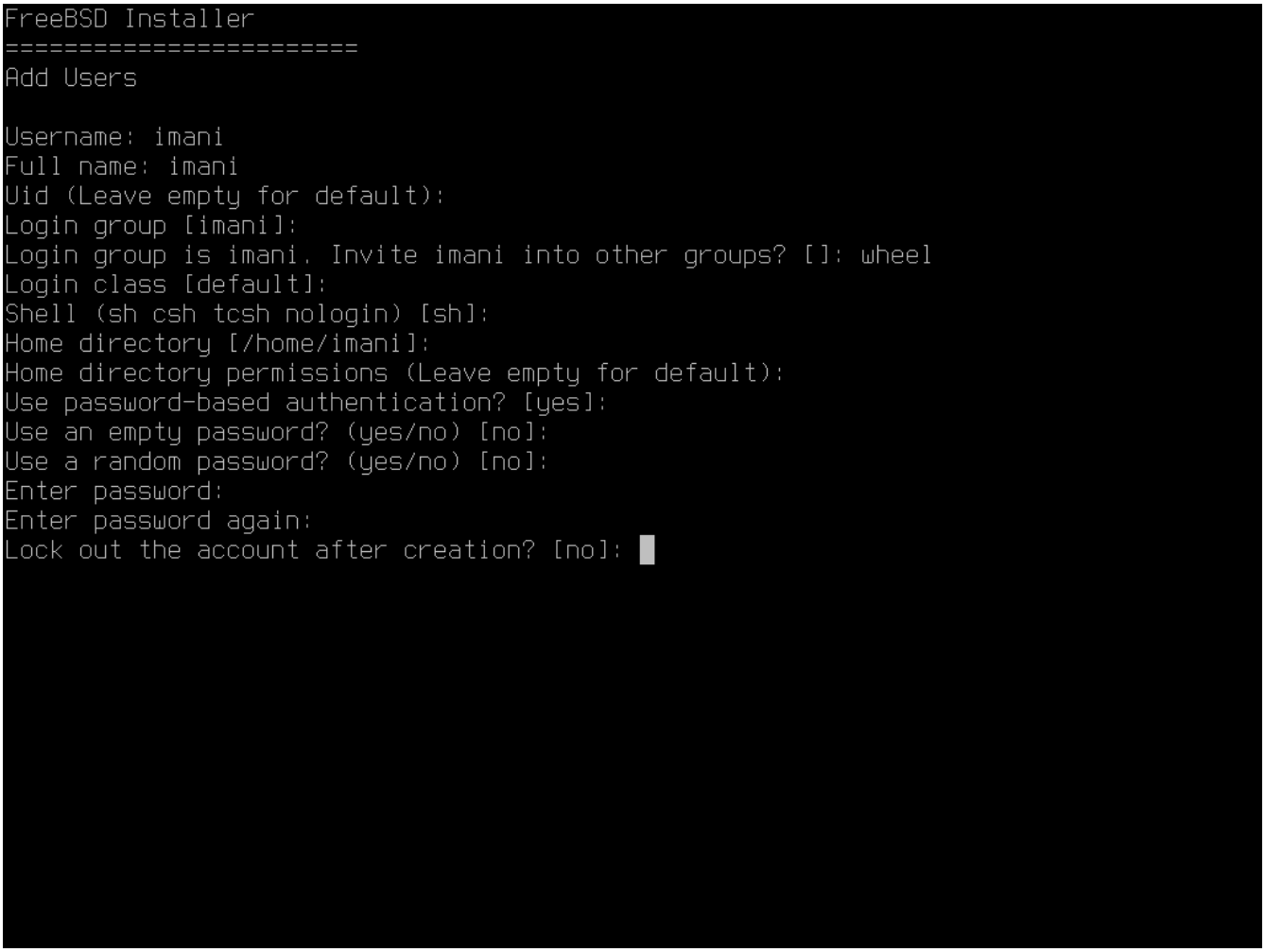


Figura 44. Insira as informações do usuário

Aqui está um resumo das informações para solicitadas:

- `Username` - O nome que o usuário digitará para efetuar login. Uma convenção comum é usar a primeira letra do primeiro nome combinada com o sobrenome, desde que cada nome de usuário seja exclusivo para o sistema. O nome de usuário faz distinção entre maiúsculas e minúsculas e não deve conter espaços.
- `Username` - O nome completo do usuário. Este campo pode conter espaços e é usado como uma descrição para a conta do usuário.
- `Uid` - ID do Usuário. Normalmente, isso é deixado em branco para que o sistema atribua um valor.
- `Login group` - O grupo do usuário. Normalmente, isso é deixado em branco para aceitar o padrão.
- `Invite user into other groups?` - Grupos adicionais aos quais o usuário será adicionado como membro. Se o usuário precisar de acesso administrativo, digite `wheel` aqui.
- `Login class` - normalmente deixado em branco para seguir com o padrão.
- `Shell` - Digite um dos valores listados para definir o shell interativo para o usuário. Consulte Shells para maiores informações sobre shells.
- `Home directory` - O diretório inicial do usuário. O padrão geralmente está correto.
- `Home directory permissions` - Permissões no diretório inicial do usuário. O padrão geralmente

está correto.

- `Use password-based authentication?` A resposta deve ser `Yes` para que o usuário seja solicitado a inserir sua senha no login.
- `Use an empty password?` - Normalmente a resposta será `No`, pois é inseguro ter uma senha em branco.
- `Use a random password?` - Normalmente a resposta será `No` para que o usuário possa definir sua própria senha no próximo prompt.
- `Enter password` - Escolha a senha para este usuário. Caracteres digitados não serão exibidos na tela.
- `Enter password again` - A senha deve ser digitada novamente para verificação.
- `Lock out the account after creation?` - A reposta normalmente será `No` para que o usuário possa fazer o login.

Depois de inserir tudo, um resumo será exibido para revisão. Se algum erro foi cometido, digite `no` e tente novamente. Se tudo estiver correto, digite `yes` para criar o novo usuário.

```
FreeBSD Installer
=======================
Add Users

Username: imani
Full name: imani
Uid (Leave empty for default):
Login group [imani]:
Login group is imani. Invite imani into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/imani]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : imani
Password   : *****
Full Name  : imani
Uid        : 1001
Class      :
Groups     : imani wheel
Home       : /home/imani
Home Mode  :
Shell      : /bin/sh
Locked     : no
OK? (yes/no) [yes]:
adduser: INFO: Successfully added (imani) to the user database.
Add another user? (yes/no) [no]:
```

Figura 45. Saia do gerenciamento de usuários e grupos

Se houver mais usuários para adicionar, responda a pergunta `Add another user?` com `yes`. Digite `no` para concluir a adição de usuários e continuar a instalação.

Para obter maiores informações sobre como adicionar usuários e sobre como gerenciá-los de

usuários, consulte Usuários e Gerenciamento Básico de Contas.

2.8.6. Configuração final

Depois que tudo tiver sido instalado e configurado, você terá uma chance final para modificar as configurações.

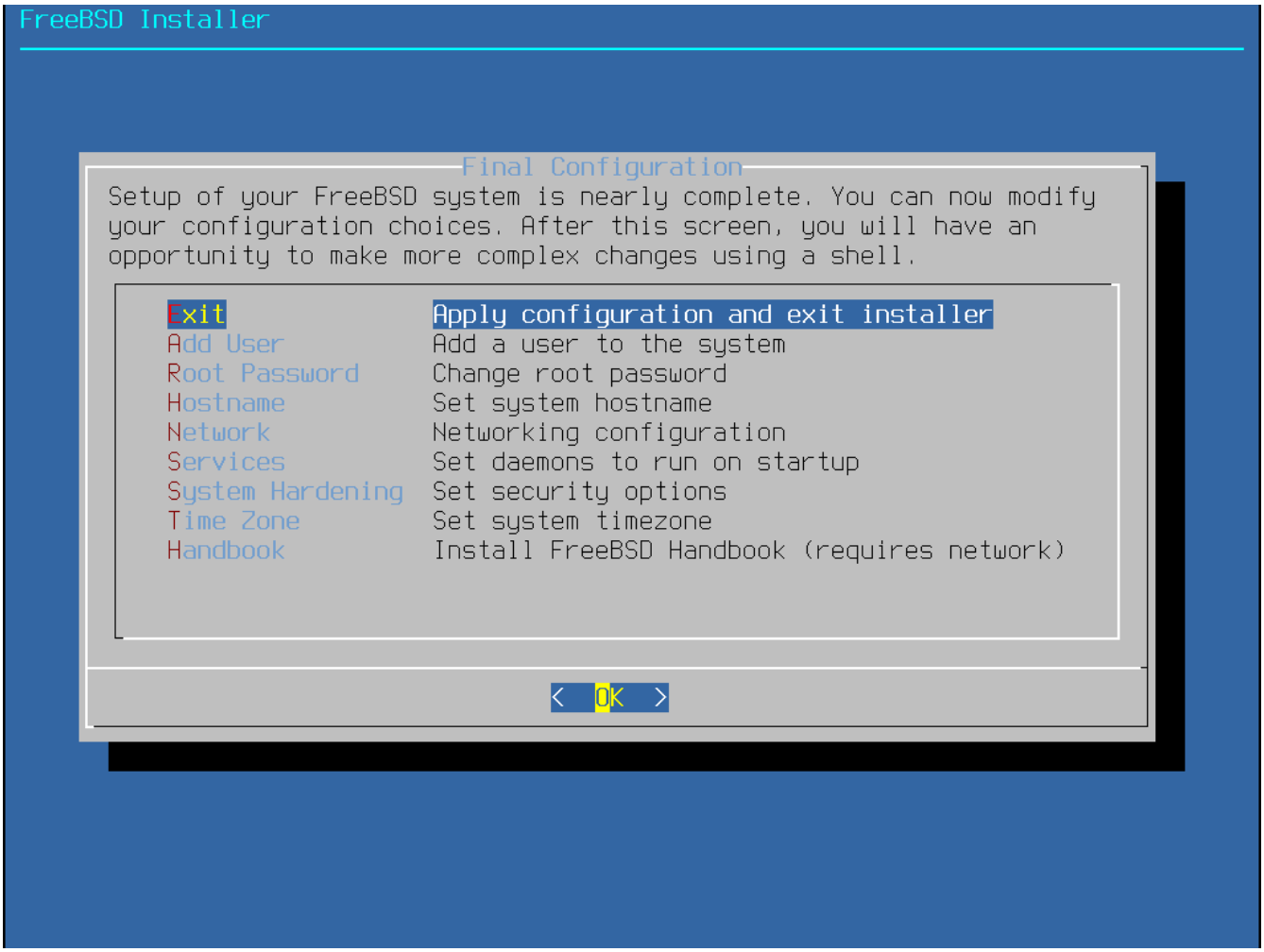


Figura 46. Configuração final

Use este menu para fazer alterações ou fazer qualquer configuração adicional antes de concluir a instalação.

- `Add User` - Descrito em Adicione usuários.
- `Root Password` - Descrito em Definindo a Senha de `root`.
- `Hostname` - Descrito em Configurando o nome do host.
- `Network` - Descrito em Configurando as Interfaces de Rede.
- `Services` - Descrito em Ativando Serviços.
- `System Hardening` - Descrito em Ativando Opções de Segurança (Hardening).
- `Time Zone` - Descrito em Defina o fuso horário.
- `Handbook` - Faça o download e instale o FreeBSD Handbook.

Depois que completar qualquer configuração final que tenha faltado, selecione **[Exit]**.

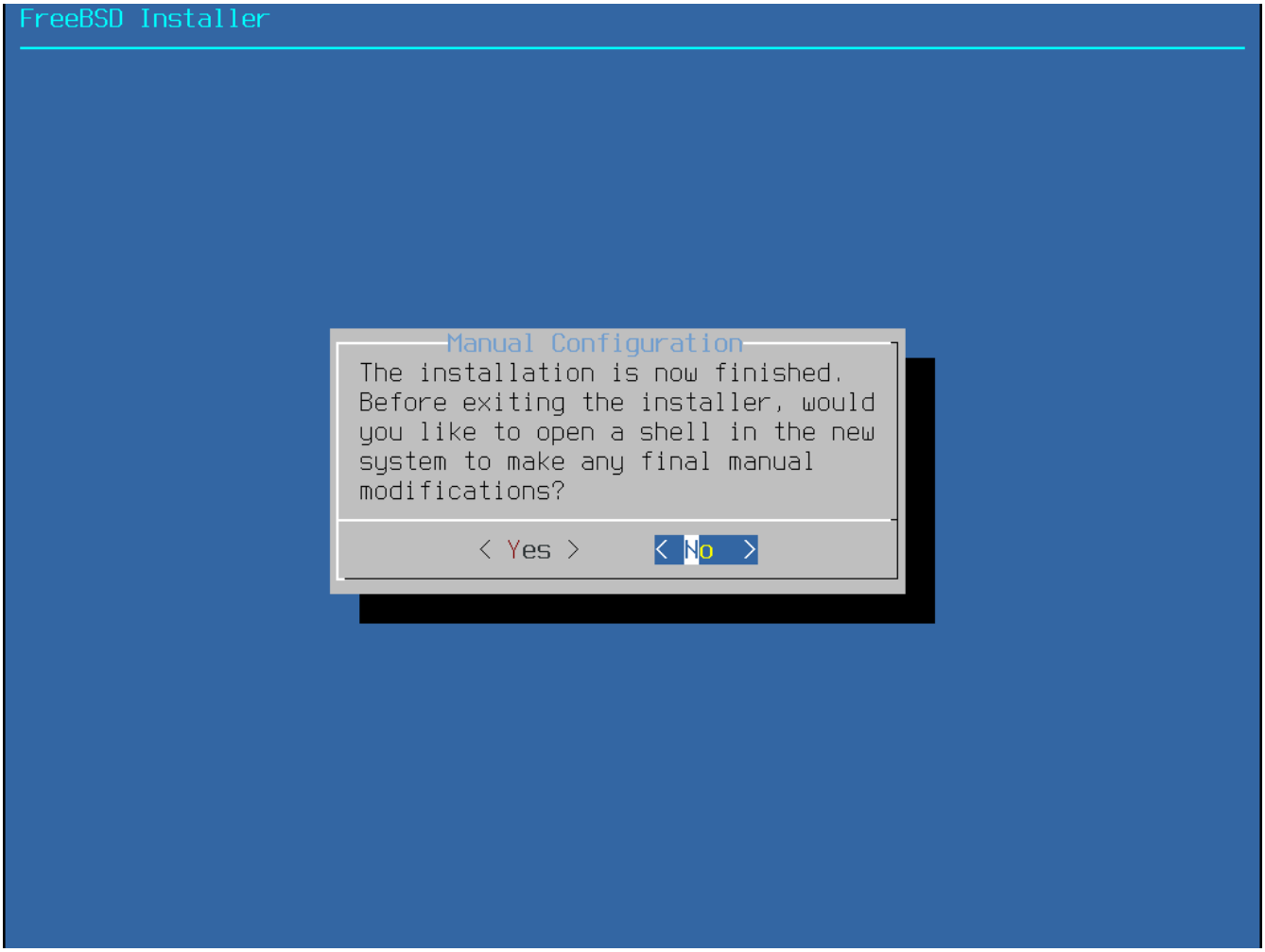


Figura 47. Configuração manual

O bsdinstall perguntará se há alguma configuração adicional que precise ser feita antes de reinicializar o novo sistema. Selecione **[Yes]** para sair para um shell dentro do novo sistema ou **[No]** para prosseguir para a última etapa da instalação.

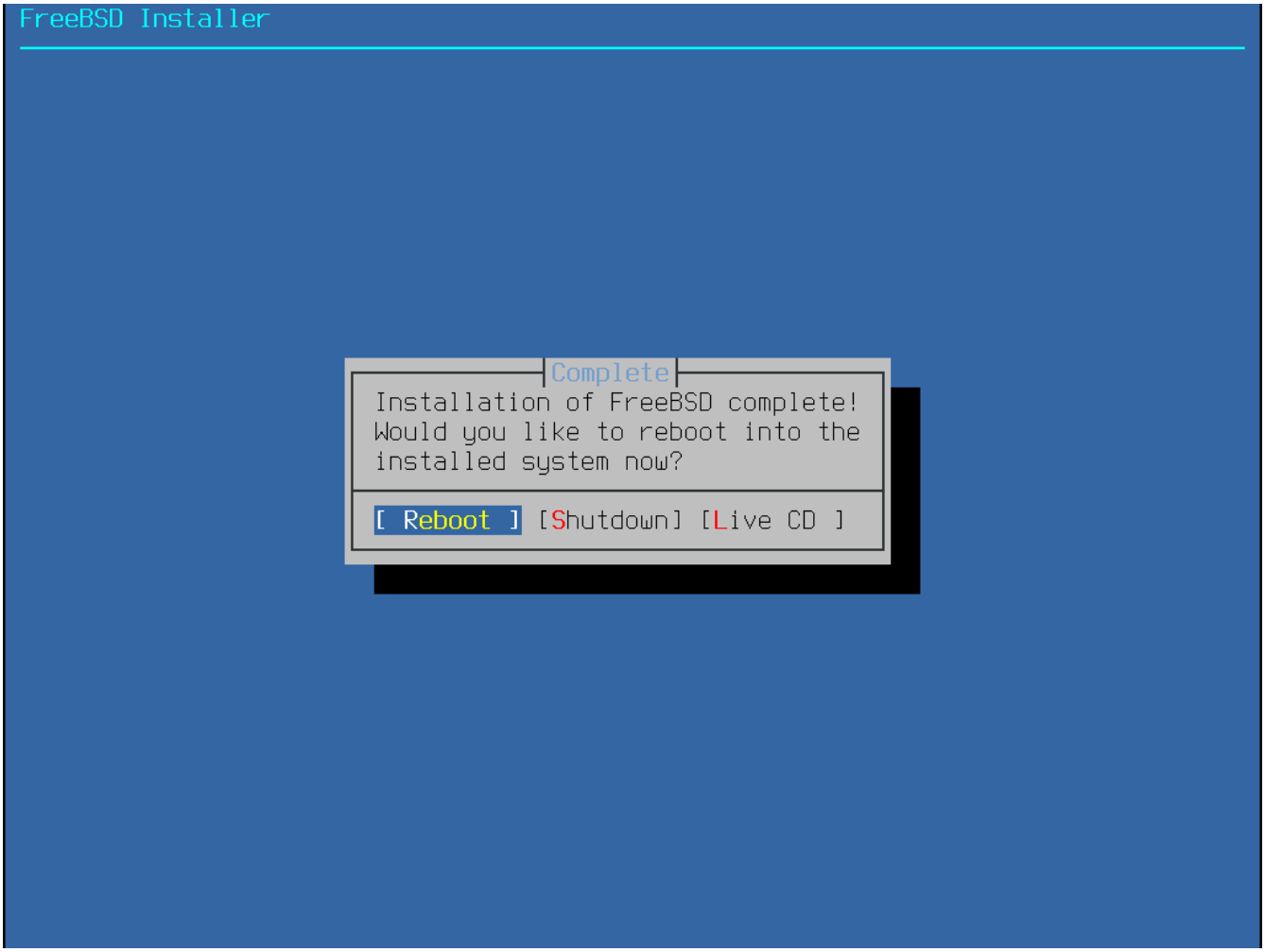


Figura 48. Conclua a instalação

Se outras configurações ou configurações especiais forem necessárias, selecione **[Live CD]** para inicializar a mídia de instalação no modo Live CD.

Se a instalação estiver completa, selecione **[Reboot]** para reiniciar o computador e iniciar o novo sistema FreeBSD. Não se esqueça de remover a mídia de instalação do FreeBSD ou o computador poderá inicializar novamente a partir dela.

Quando o FreeBSD inicializa, mensagens informativas são exibidas. Depois que o sistema concluir a inicialização, um prompt de login será exibido. No `login:`, insira o nome de usuário adicionado durante a instalação. Evite efetuar login como `root`. Consulte A conta de superusuário para instruções sobre como se tornar o superusuário quando o acesso administrativo for necessário.

As mensagens que apareceram durante a inicialização podem ser revisadas pressionando `Scroll-Lock` para ativar o buffer de rolagem para trás. As teclas `PgUp`, `PgDn` e setas podem ser usadas para rolar pelas mensagens. Quando terminar, pressione `Scroll-Lock` novamente para desbloquear o visor e retornar ao console. Para revisar essas mensagens depois que o sistema estiver ativo por algum tempo, digite `less /var/run/dmesg.boot` em um prompt de comando. Pressione `q` para retornar à linha de comando após a visualização.

Se o sshd foi habilitado em Selecionando Serviços Adicionais para Ativar, a primeira inicialização pode ser um pouco mais lenta, pois o sistema gerará as chaves RSA e DSA. As inicializações subseqüentes serão mais rápidas. As impressões digitais das chaves serão exibidas, conforme mostrado neste exemplo:

```
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|       o         |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|       ..     . .|
|      o  .   . + |
|     . ..   . E .|
|    . .  o o . . |
|     +  S = .    |
|    +  . = o     |
|     +  . * .    |
|    . .  o .     |
|      .o. .      |
+-----------------+
Starting sshd.
```

Consulte OpenSSH para maiores informações sobre fingerprints e o SSH.

O FreeBSD não instala um ambiente gráfico por padrão. Consulte O sistema X Window para maiores informações sobre como instalar e configurar um gerenciador gráfico de janelas.

O desligamento adequado de um computador FreeBSD ajuda a proteger os dados e o hardware contra danos. *Não desligue a energia antes do sistema ter sido desligado corretamente!* Se o usuário for membro do grupo `wheel`, torne-se o superusuário digitando `su` na linha de comando e inserindo a senha do usuário `root`. Em seguida, digite `shutdown -p now` e o sistema será desligado corretamente e, se o hardware suportar, irá se desliga-se.

2.9. Interfaces de Rede

2.9.1. Configurando as Interfaces de Rede

Em seguida, é mostrada uma lista das interfaces de rede encontradas no computador. Selecione a interface para configurar.

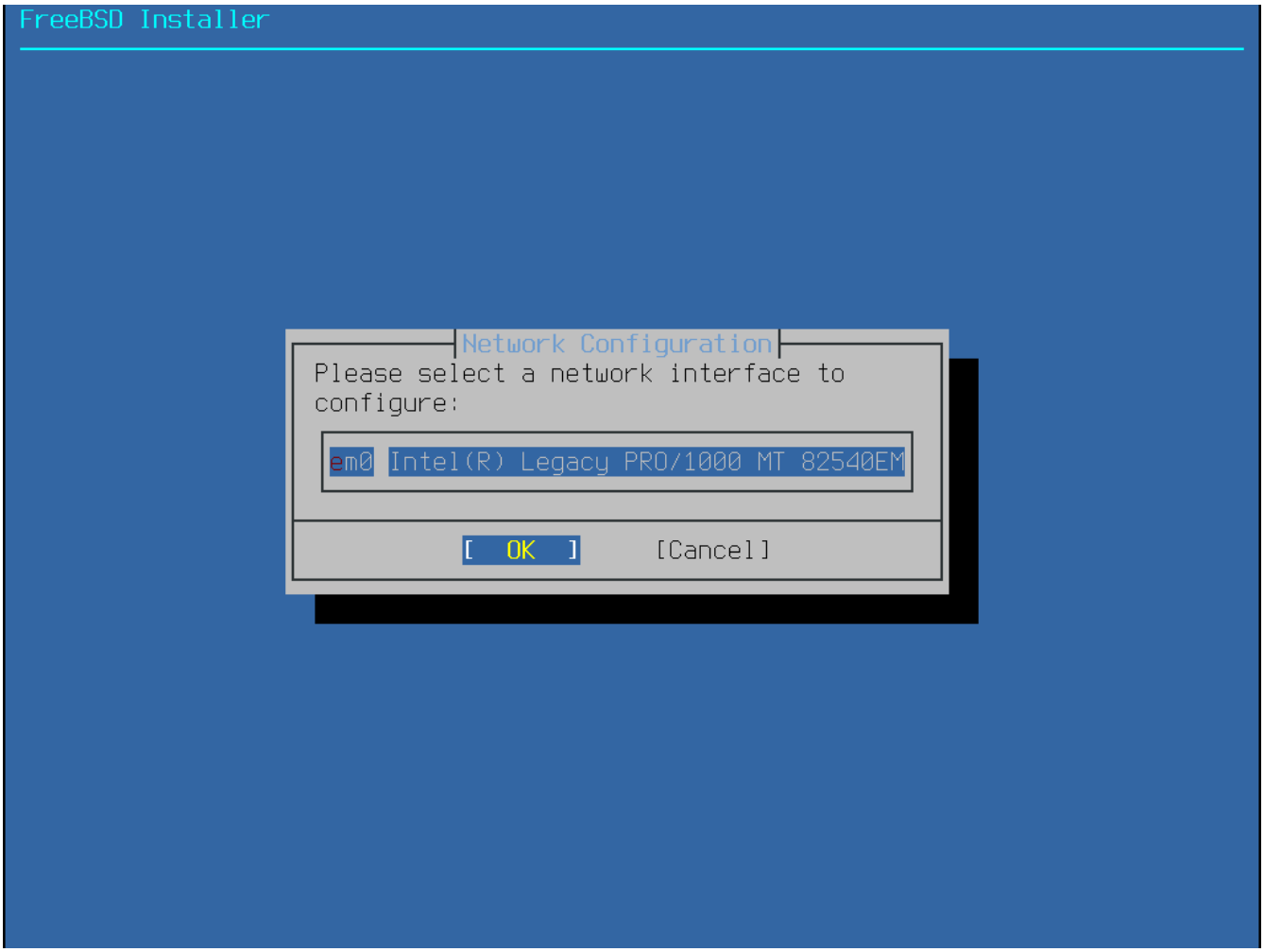


Figura 49. Escolha uma interface de rede

Se uma interface Ethernet for selecionada, o instalador irá pular para o menu mostrado em Escolha a rede IPv4. Se uma interface de rede sem fio for escolhida, o sistema procurará pontos de acesso sem fio:

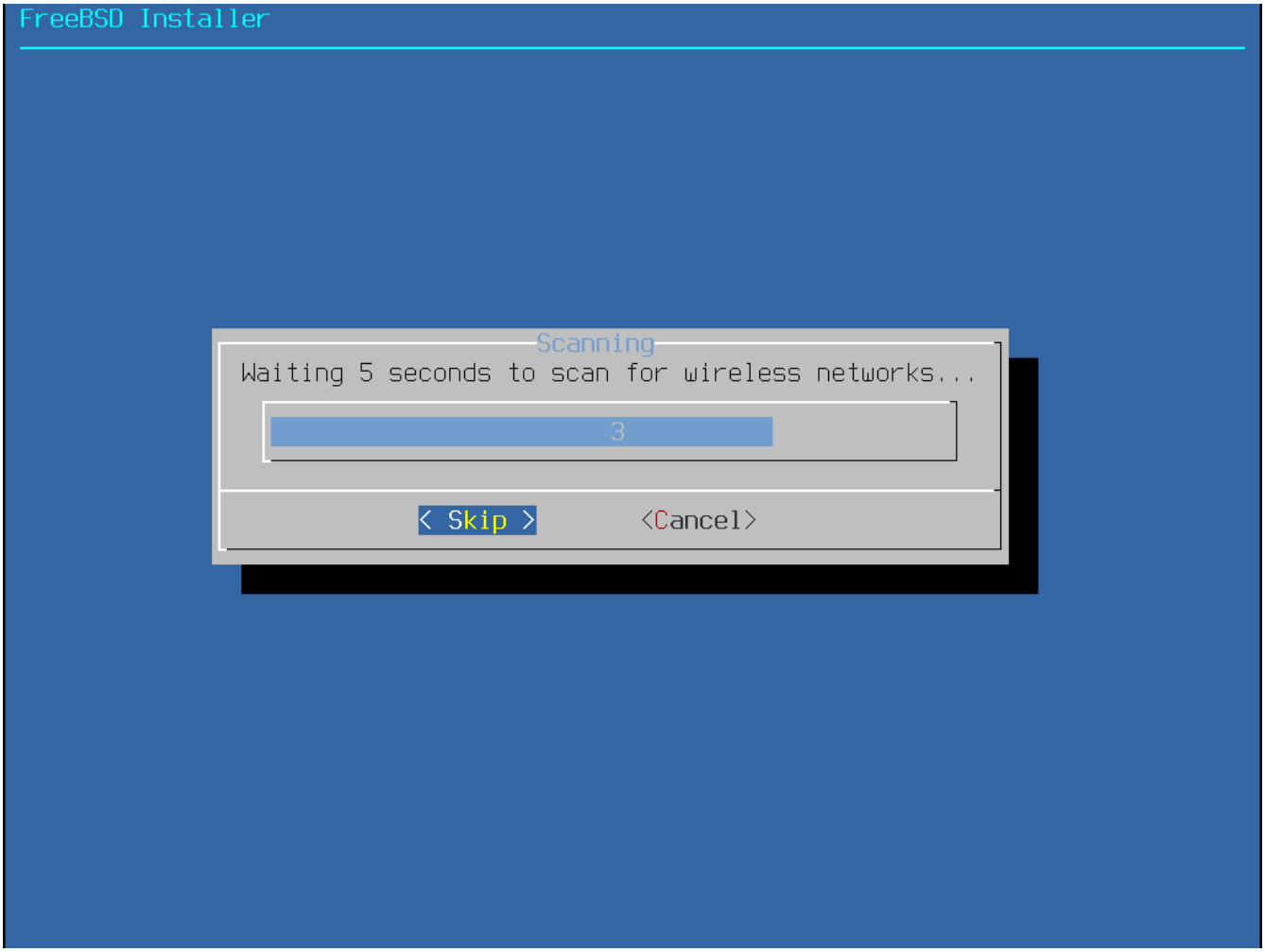


Figura 50. Buscando por pontos de acesso sem fio

As redes sem fio são identificadas por um identificador de conjunto de serviços (SSID), um nome curto e exclusivo dado a cada rede. Os SSIDs encontrados durante a busca serão listados, seguidos por uma descrição dos tipos de criptografia disponíveis para essa rede. Se o SSID desejado não aparecer na lista, selecione **[Rescan]** para buscar novamente. Se a rede desejada ainda não aparecer, verifique se há problemas com as conexões da antena ou tente mover o computador para mais perto do ponto de acesso. refaça a busca após cada alteração ser feita.

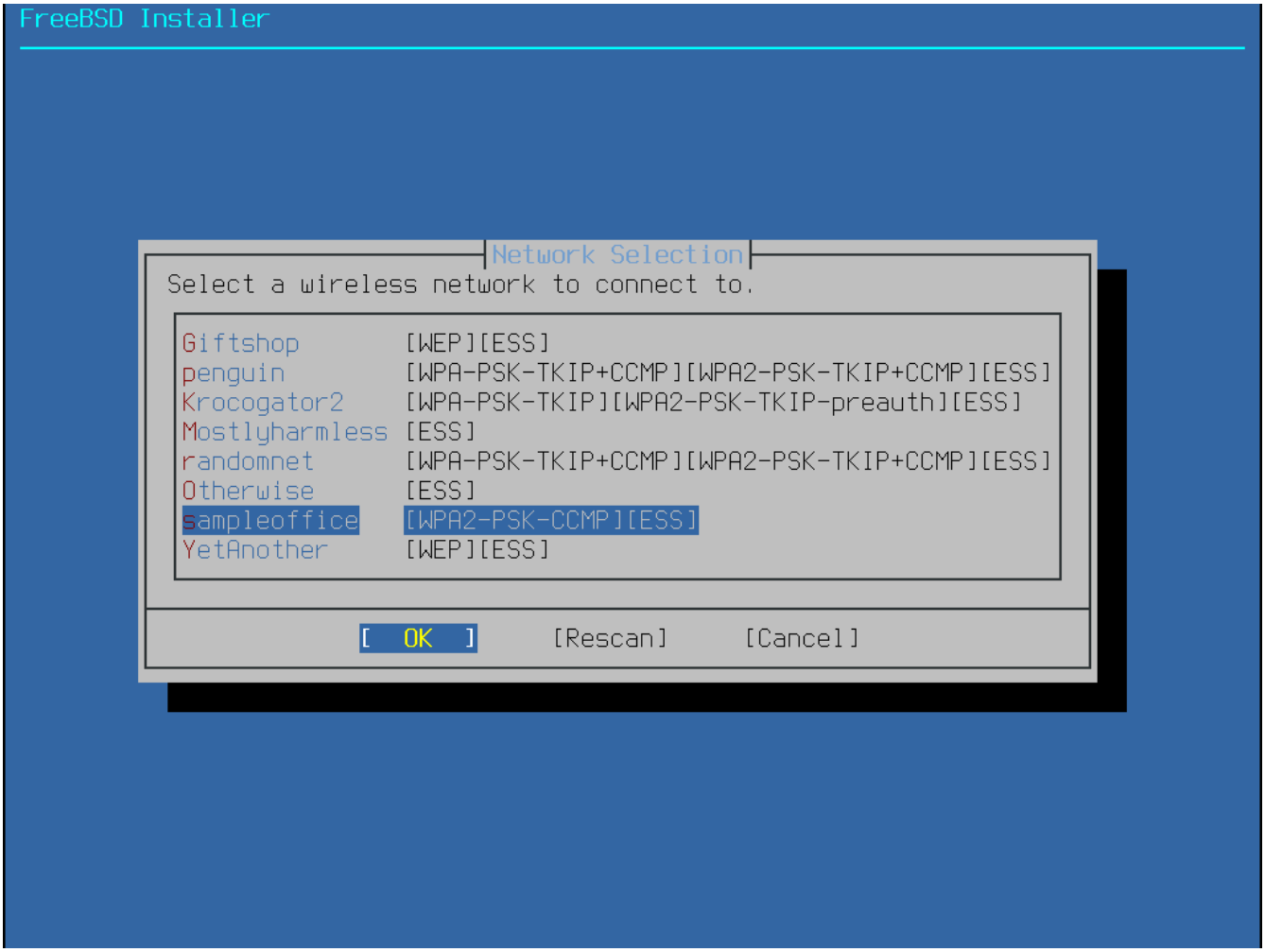


Figura 51. Escolhendo uma rede sem fio

Em seguida, insira as informações de criptografia para se conectar à rede sem fio selecionada. A encriptação WPA2 é fortemente recomendada, pois os tipos de encriptação mais antigos, como o WEP, oferecem pouca segurança. Se a rede usar WPA2, insira a senha, também conhecida como Chave Pré-Compartilhada (PSK). Por motivos de segurança, os caracteres digitados na caixa de entrada são exibidos como asteriscos.

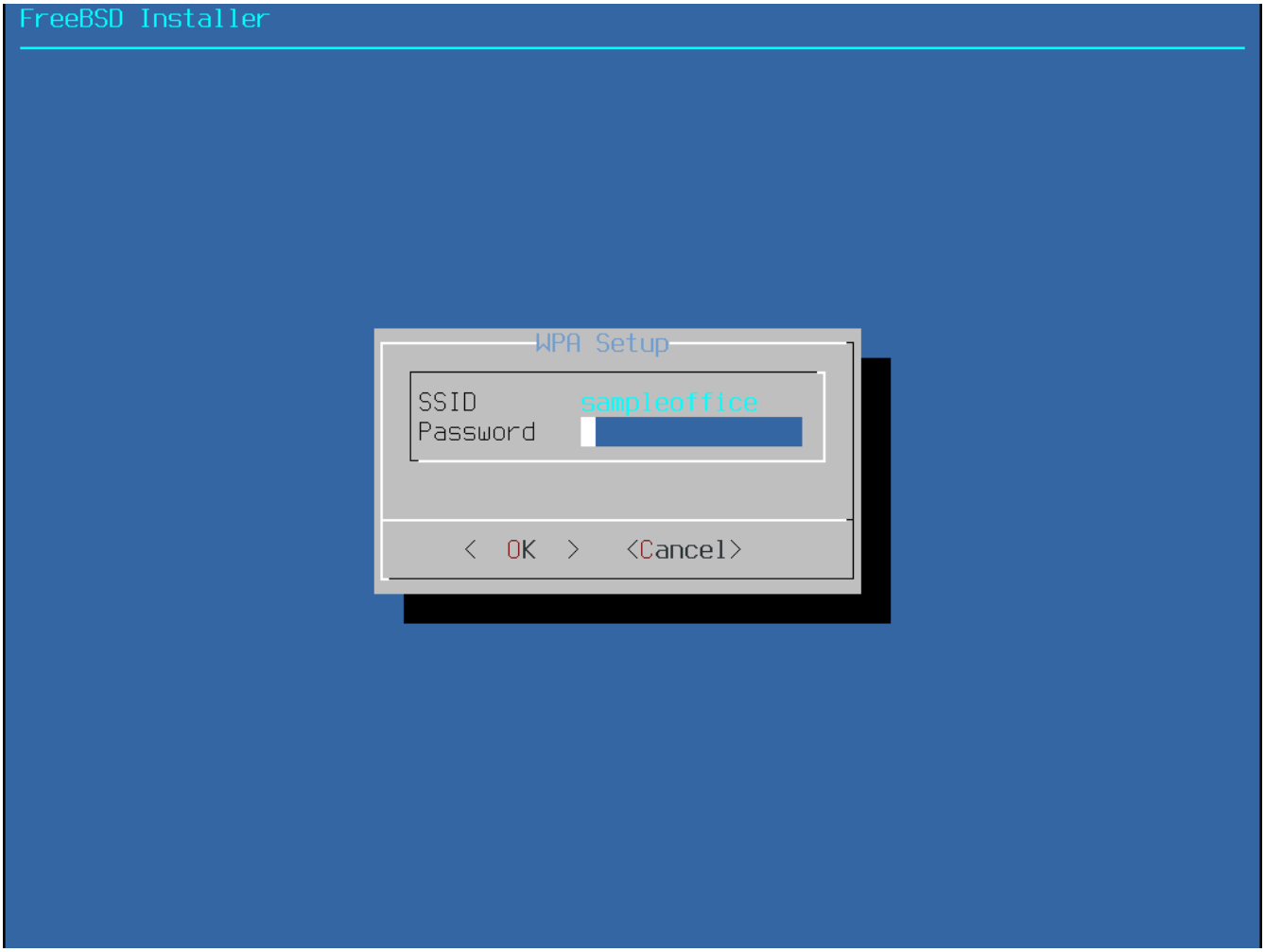


Figura 52. Configuração WPA2

Em seguida, escolha se um endereço IPv4 deve ou não ser configurado na interface Ethernet ou na interface sem fio:

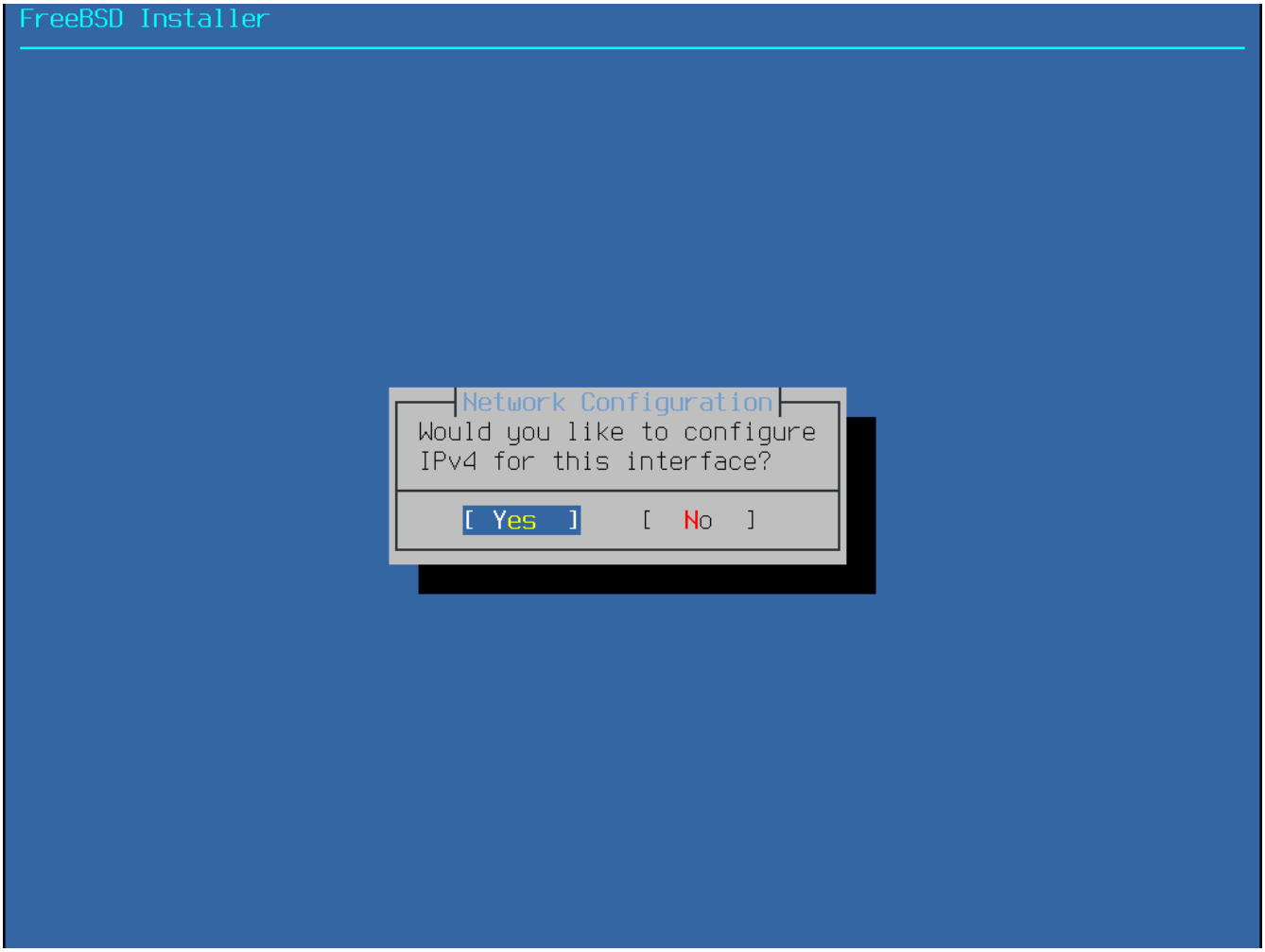


Figura 53. Escolha a rede IPv4

Existem dois métodos de configuração de IPv4. O DHCP configurará automaticamente a interface de rede da forma correta e deverá ser usado se a rede fornecer um servidor DHCP. Caso contrário, as informações de endereçamento precisam ser inseridas manualmente como em uma configuração estática.



Não insira informações de rede aleatórias, pois isso não funcionará. Se um servidor DHCP não estiver disponível, obtenha as informações listadas em [bsdinstall-collect-network-information] do administrador da rede ou do provedor de serviços de Internet.

Se um servidor DHCP estiver disponível, selecione **[Yes]** no próximo menu para configurar automaticamente a interface de rede. O instalador parecerá pausar por um minuto ou mais enquanto encontra o servidor DHCP e obtém as informações de endereçamento do sistema.

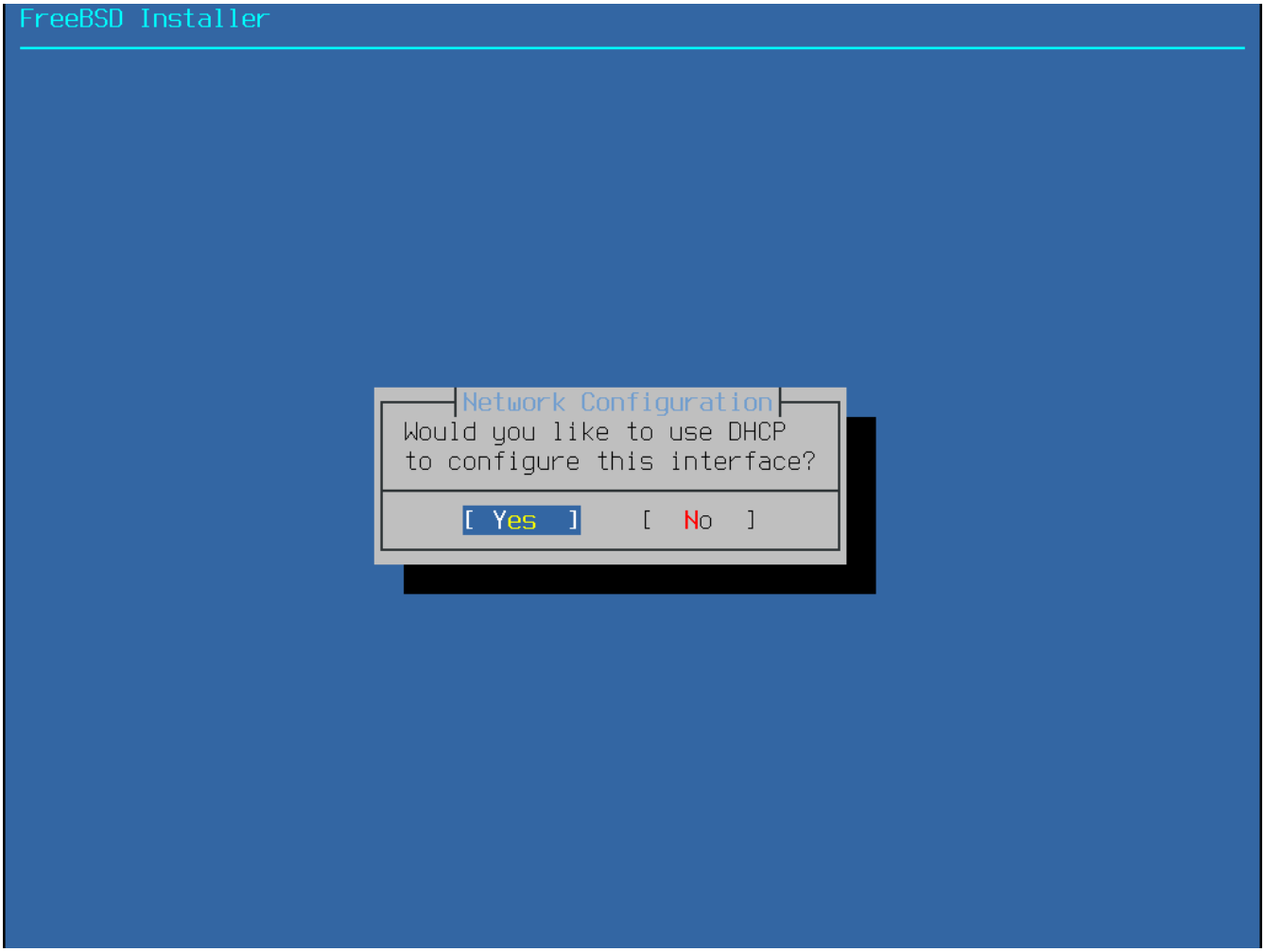


Figura 54. Escolha a configuração IPv4DHCP

Se um servidor DHCP não estiver disponível, selecione **[No]** e insira as seguintes informações de endereçamento neste menu:

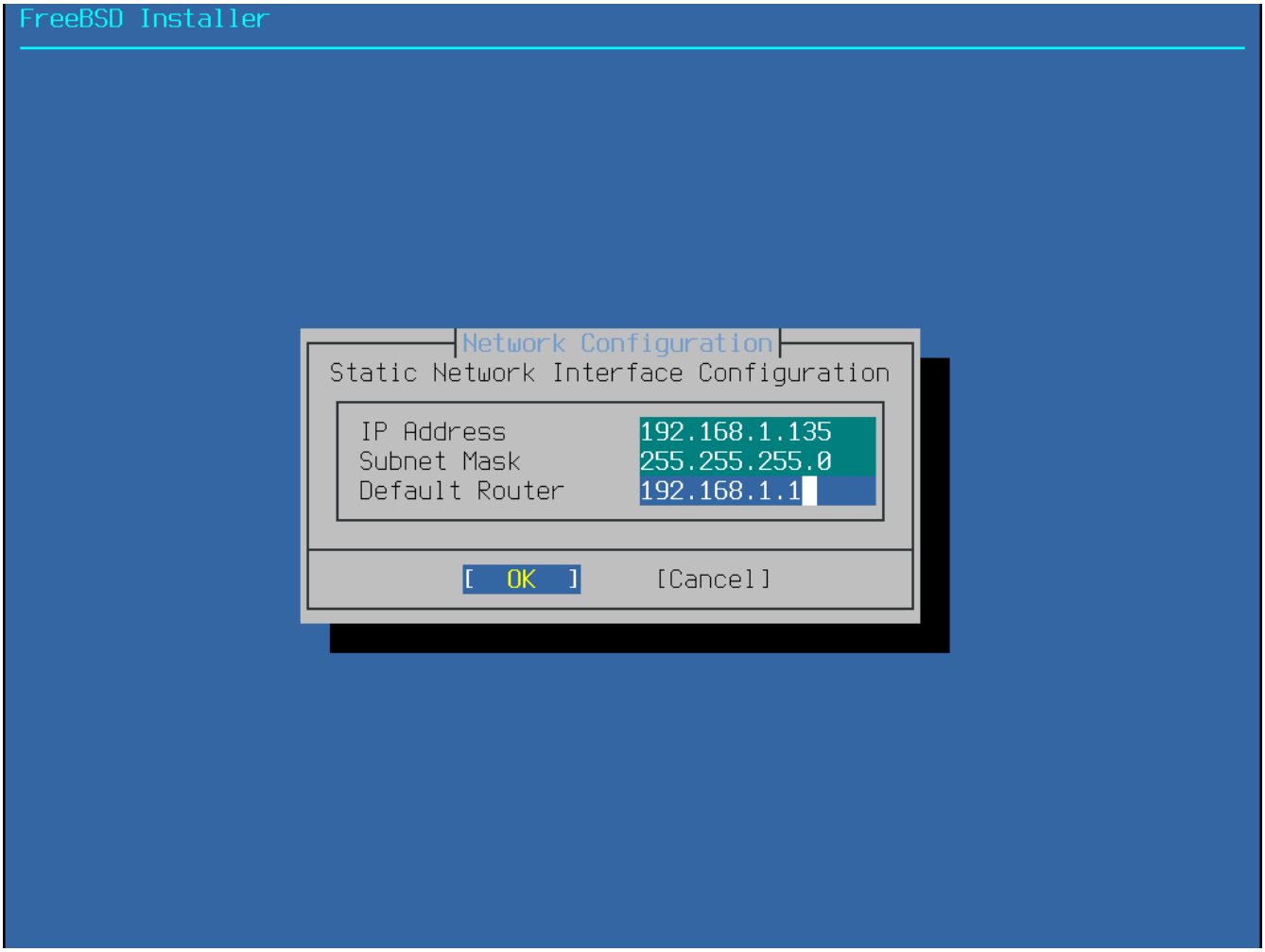


Figura 55. Configuração IPv4 estática

- `Endereço IP` - O endereço IPv4 atribuído a este computador. O endereço deve ser único e não estar em uso por outro equipamento na rede local.
- `Subnet Mask` - A máscara de sub-rede da rede.
- `Default Router` - O endereço IP do gateway padrão da rede.

A próxima tela perguntará se a interface deve ser configurada para IPv6. Se IPv6 estiver disponível e for desejado, escolha **[Yes]** para selecioná-lo.

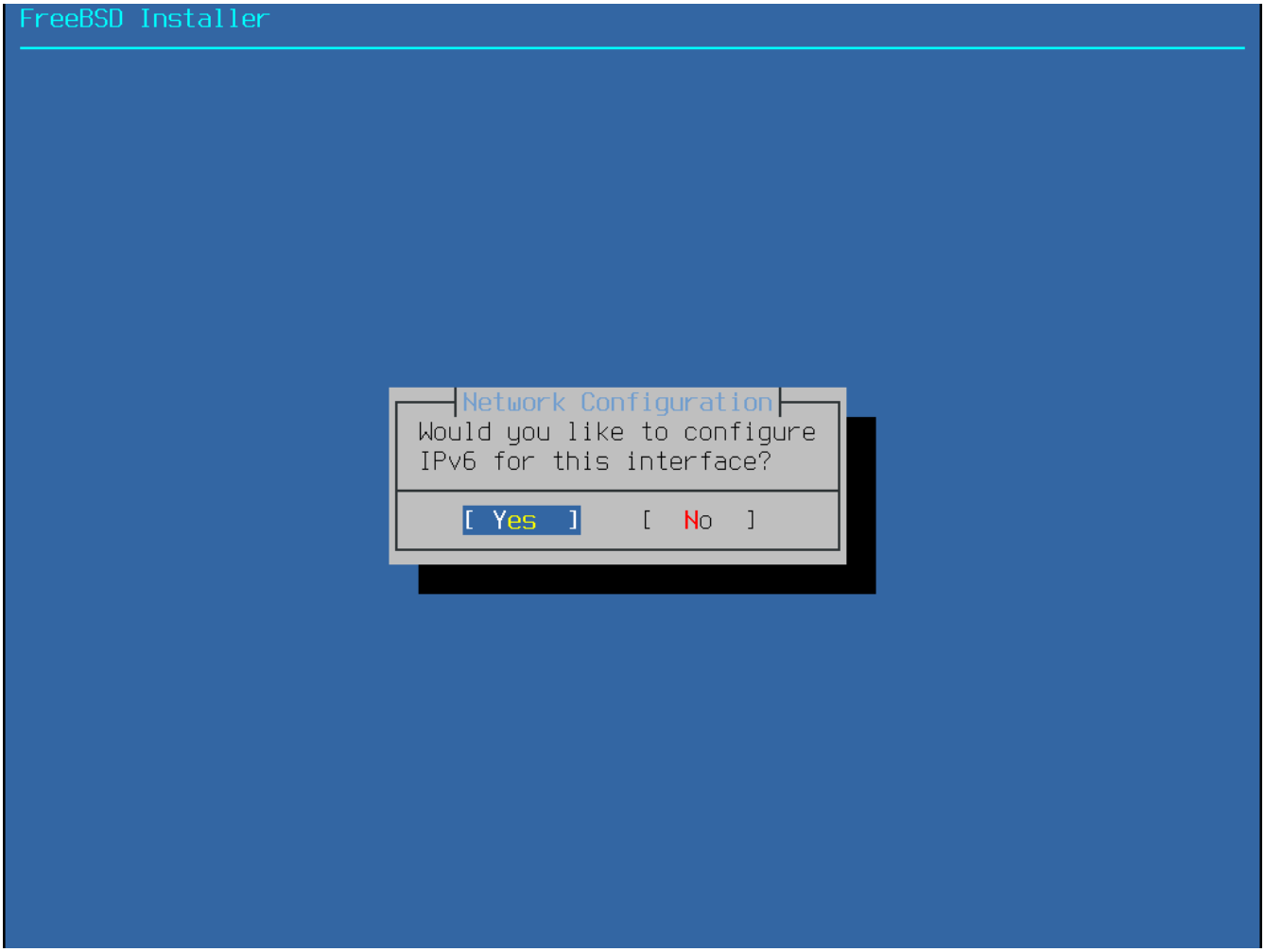


Figura 56. Escolha a rede IPv6

O IPv6 também possui dois métodos de configuração. A configuração automática de endereços sem estado (SLAAC) solicitará automaticamente as informações de configuração corretas de um roteador local. Consulte rfc4862 para maiores informações. A configuração estática requer entrada manual das informações da rede.

Se um roteador IPv6 estiver disponível, selecione **[Yes]** no próximo menu para configurar automaticamente a interface de rede. O instalador parecerá pausar por um minuto ou mais enquanto localiza o roteador e obtém as informações de endereçamento do sistema.

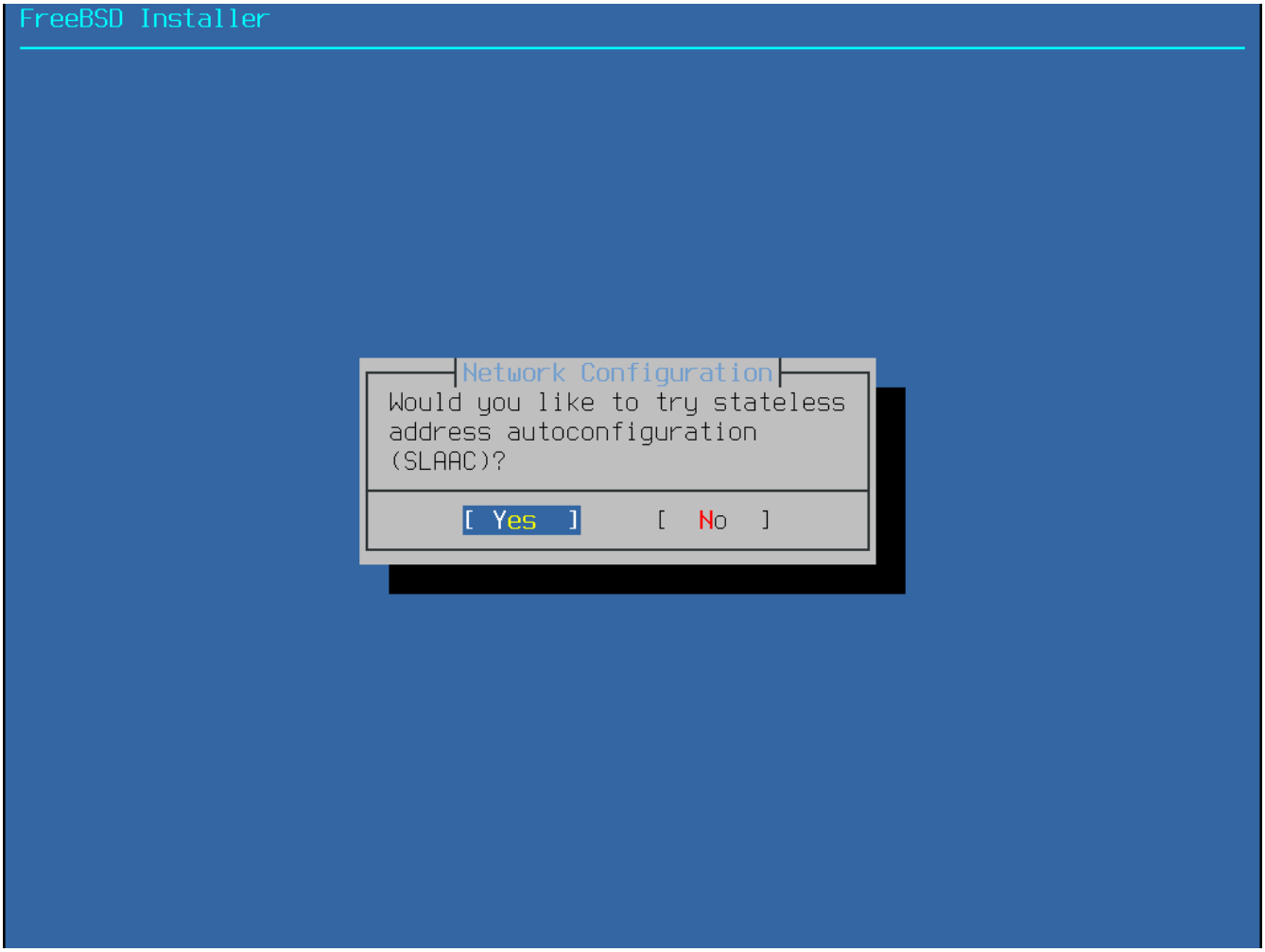


Figura 57. Escolha a configuração do SLAAC do IPv6

Se um roteador IPv6 não estiver disponível, selecione **[No]** e insira as seguintes informações de endereçamento neste menu:

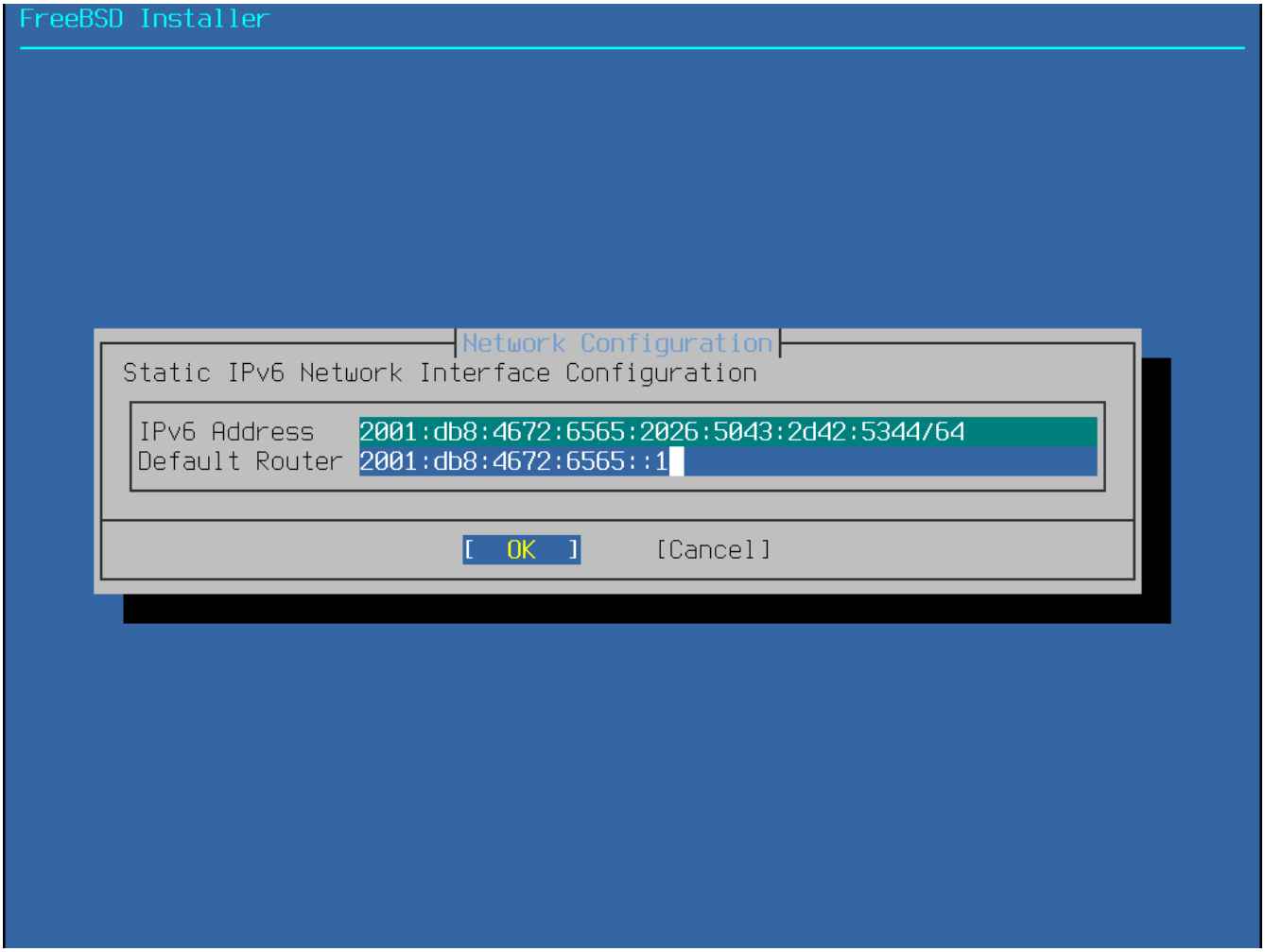


Figura 58. Configuração Estática do IPv6

- `Endereço IPv6` - O endereço IPv6 atribuído a este computador. O endereço deve ser único e não estar em uso por outro equipamento na rede local.
- `Default Router` - O endereço IPv6 do gateway padrão da rede.

O último menu de configuração de rede é usado para configurar o resolvedor do Sistema de Nomes de Domínio (DNS), que converte nomes de host de e para endereços de rede. Se o DHCP ou SLAAC foi usado para autoconfigurar a interface de rede, os valores do `Resolver Configuration` podem já estar preenchidos. Caso contrário, insira o domínio da rede local nome no campo `Search`. `DNS # 1` e `DNS # 2` são os endereços IPv4 e/ou IPv6 dos servidores de DNS. Pelo menos um servidor DNS é necessário.

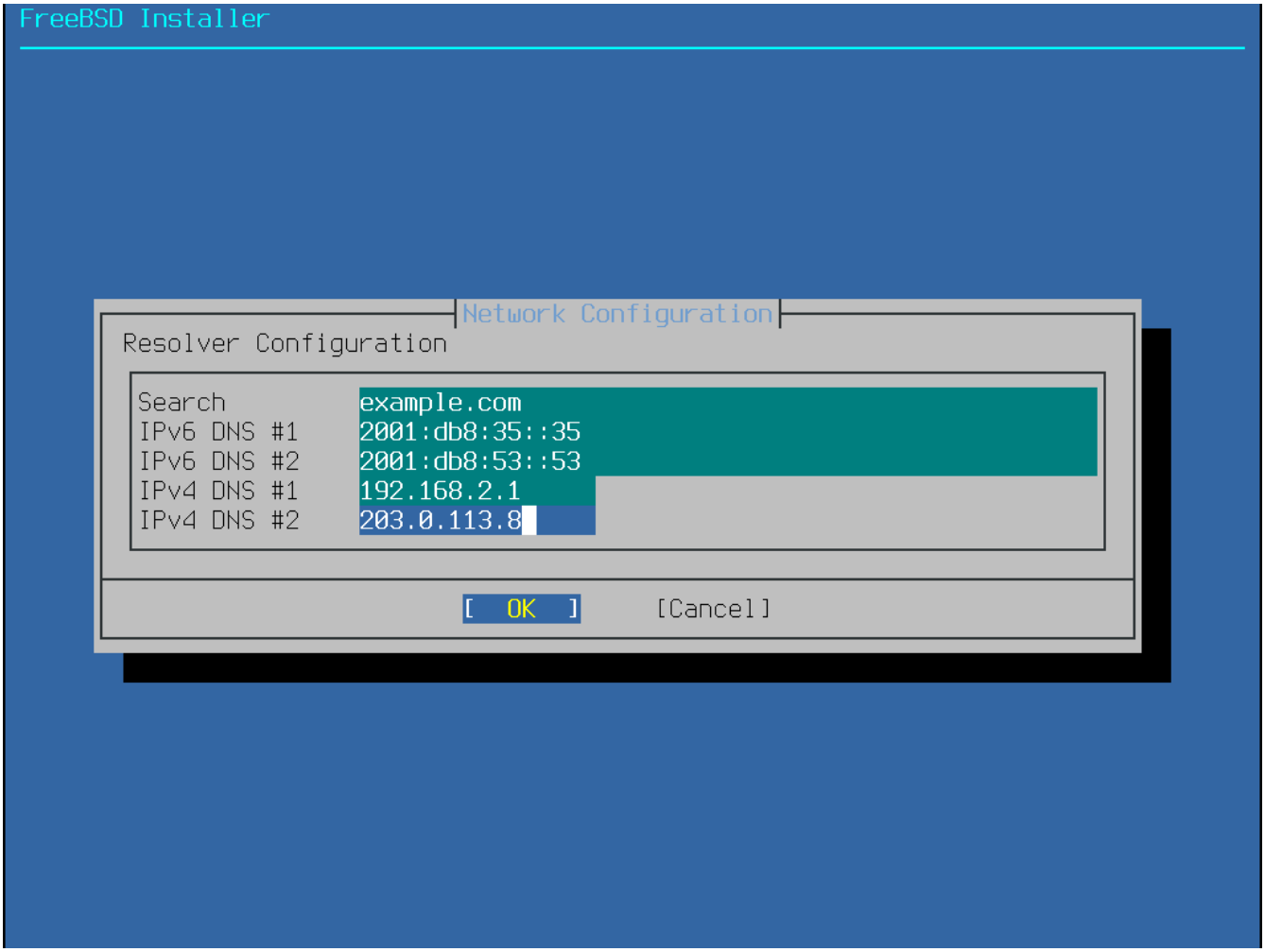


Figura 59. Configuração do DNS

Quando a interface estiver configurada, selecione um site espelho localizado na mesma região do mundo que o computador no qual o FreeBSD está sendo instalado. Os arquivos podem ser recuperados mais rapidamente quando o espelho está próximo ao computador de destino, reduzindo o tempo de instalação.

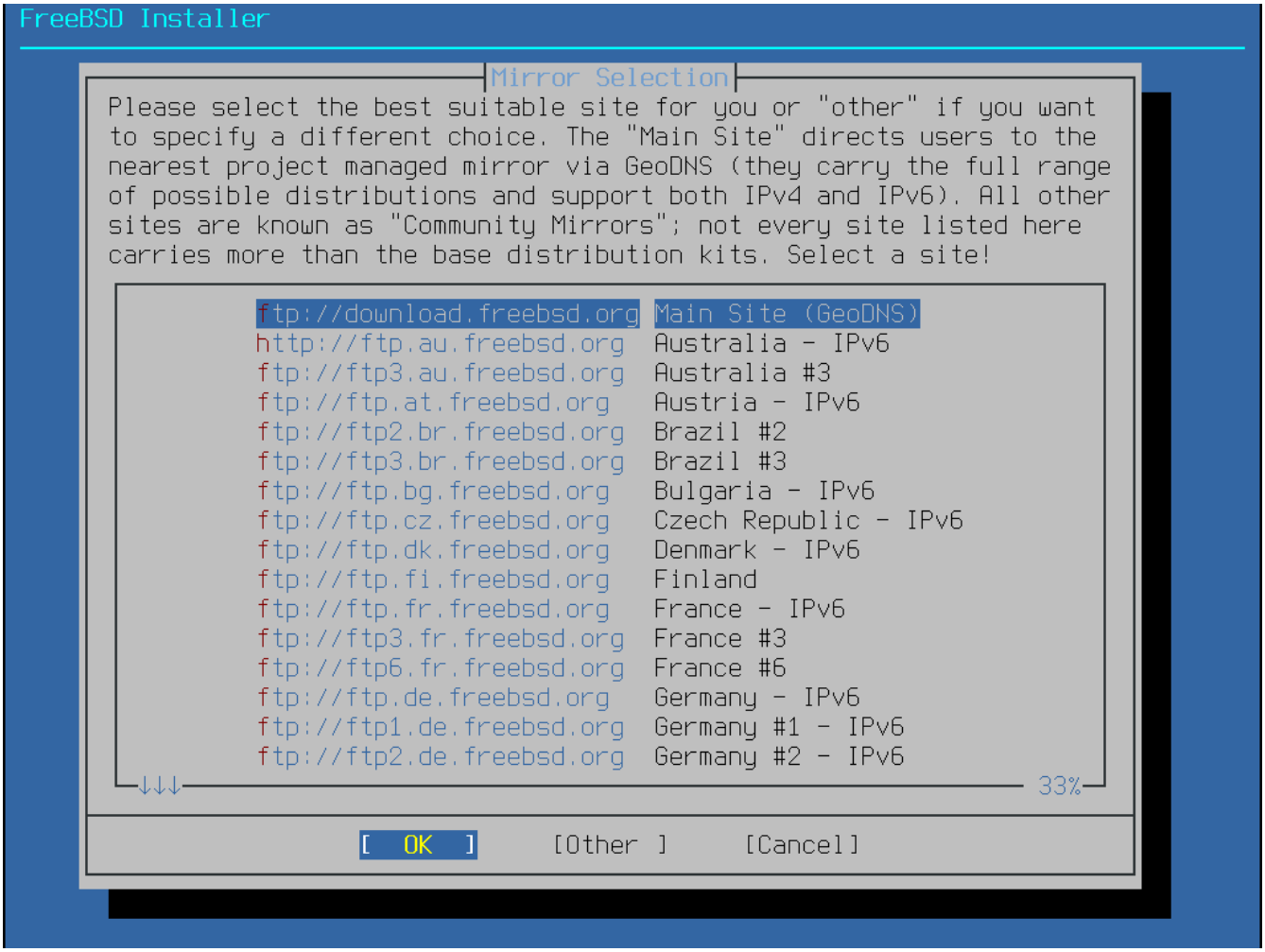


Figura 60. Escolhendo um Site Espelho

2.10. Solução de problemas

Esta seção aborda a solução de problemas básicos de instalação, tais como problemas comuns que as pessoas relataram.

Verifique o documento Notas de Hardware (https://www.freebsd.org/releases/) para a versão do FreeBSD para garantir que o hardware é suportado. Se o hardware for suportado e houver travamentos ou outros problemas, compile um kernel personalizado usando as instruções em Configurando o kernel do FreeBSD para adicionar suporte a dispositivos que não estão presentes no kernel GENERIC. O kernel padrão assume que a maioria dos dispositivos de hardware está na configuração padrão de fábrica em termos de IRQs, endereços de I/O e canais DMA. Se o hardware foi reconfigurado, um arquivo de configuração personalizado do kernel pode dizer ao FreeBSD onde encontrar os dispositivos.



Alguns problemas de instalação podem ser evitados ou aliviados com a atualização do firmware em vários componentes de hardware, principalmente na placa-mãe. O firmware da placa-mãe é geralmente chamado de BIOS. A maioria dos fabricantes de placas-mãe e computadores tem um site para atualizações e para informações sobre as atualizações.

Os fabricantes geralmente desaconselham a atualização da BIOS da placa-mãe, a menos que haja uma boa razão para isso, como uma atualização crítica. O processo

> de atualização *pode* dar errado, deixando o BIOS incompleto e o computador inoperante.

Se o sistema trava enquanto verifica o hardware durante a inicialização ou se comporta de maneira estranha durante a instalação, o ACPI pode ser o culpado. O FreeBSD faz uso extensivo do sistema ACPI nas plataformas i386 e amd64 para ajudar na configuração do sistema, caso seja detectado durante a inicialização. Infelizmente, alguns bugs ainda existem tanto no driver ACPI como nas placas-mãe do sistema e no firmware BIOS. O ACPI pode ser desativado configurando a opção `hint.acpi.0.disabled` no terceiro estágio do boot loader:

```
set hint.acpi.0.disabled="1"
```

Isso é redefinido toda vez que o sistema é inicializado, portanto é necessário adicionar `hint.acpi.0.disabled="1"` ao arquivo /boot/loader.conf . Maiores informações sobre o boot loader podem ser encontradas em Sinopse.

2.11. Usando o Live CD

O menu de boas-vindas do bsdinstall, mostrado em Menu de boas-vindas, fornece uma opção **[Live CD]**. Isto é útil para aqueles que ainda estão se perguntando se o FreeBSD é o sistema operacional correto para eles e quer testar alguns dos recursos antes de instalar.

Os seguintes pontos devem ser observados antes de usar o **[Live CD]**:

- Para obter acesso ao sistema, a autenticação é necessária. O nome de usuário é `root` e a senha está em branco.
- Como o sistema é executado diretamente da mídia de instalação, o desempenho será significativamente mais lento do que o de um sistema instalado em um disco rígido.
- Essa opção fornece apenas um prompt de comando e não uma interface gráfica.

Capítulo 3. Fundamentos do FreeBSD

3.1. Sinopse

Este capítulo cobre os comandos básicos e as funcionalidades do sistema operacional FreeBSD. Grande parte deste material é relevante para qualquer sistema operacional do tipo UNIX™. Novos usuários do FreeBSD são encorajados a ler este capítulo cuidadosamente.

Depois de ler este capítulo, você saberá:

- Como usar e configurar consoles virtuais.
- Como criar e gerenciar usuários e grupos no FreeBSD.
- Como funcionam as permissões de arquivo UNIX™ e as flags de arquivos do FreeBSD.
- O layout padrão do sistema de arquivos do FreeBSD.
- A organização do disco no FreeBSD.
- Como montar e desmontar sistemas de arquivos.
- O que são processos, daemons e sinais.
- O que é um shell e como alterar o ambiente de login padrão.
- Como usar editores de texto básicos.
- O que são devices e device nodes.
- Como ler páginas de manual para obter maiores informações.

3.2. Consoles e Terminais Virtuais

A menos que o FreeBSD tenha sido configurado para iniciar automaticamente um ambiente gráfico durante a inicialização, o sistema inicializará em um prompt de login da linha de comando, como visto neste exemplo:

```
FreeBSD/amd64 (pc3.example.org) (ttyv0)

login:
```

A primeira linha contém algumas informações sobre o sistema. O `amd64` indica que o sistema neste exemplo está executando uma versão de 64 bits do FreeBSD. O nome do host é `pc3.example.org`, e ttyv0 indica que este é o "console do sistema". A segunda linha é o prompt de login.

Como o FreeBSD é um sistema multiusuário, ele precisa de alguma maneira distinguir entre usuários diferentes. Isso é feito exigindo que todos os usuários façam login no sistema antes de obter acesso aos programas no sistema. Cada usuário tem um "nome de usuário" único e uma "senha" pessoal.

Para efetuar login no console do sistema, digite o nome de usuário que foi configurado durante a instalação do sistema, conforme descrito em Adicione usuários e pressione `Enter`. Em seguida,

insira a senha associada ao nome de usuário e pressione `Enter`. A senha não é *ecoada* por razões de segurança.

Uma vez que a senha correta é inserida, a mensagem do dia (MOTD) será exibida, seguida de um prompt de comando. Dependendo do shell que foi selecionado quando o usuário foi criado, este prompt será um caractere `#`, `$` ou `%`. O prompt indica que o usuário está logado no console do sistema FreeBSD e pronto para testar os comandos disponíveis.

3.2.1. Consoles Virtuais

Enquanto o console do sistema pode ser usado para interagir com o sistema, um usuário trabalhando a partir da linha de comando no teclado de um sistema FreeBSD normalmente irá efetuar login em um console virtual. Isso ocorre porque as mensagens do sistema são configuradas por padrão para serem exibidas no console do sistema. Essas mensagens serão exibidas por cima do comando ou arquivo em que o usuário estiver trabalhando, dificultando a concentração no trabalho em questão.

Por padrão, o FreeBSD é configurado para fornecer vários consoles virtuais para a entrada de comandos. Cada console virtual tem seu próprio prompt de login e shell e é fácil alternar entre os consoles virtuais. Isso essencialmente fornece a linha de comando equivalente a ter várias janelas abertas ao mesmo tempo em um ambiente gráfico.

As combinações de teclas `Alt` + `F1` até a `Alt` + `F8` foram reservadas pelo FreeBSD para alternar entre os consoles virtuais. Use `Alt` + `F1` para alternar para o console do sistema (ttyv0), `Alt` + `F2` para acessar o primeiro console virtual (ttyv1), `Alt` + `F3` para acessar o segundo console virtual (ttyv2) e assim por diante. Ao usar o Xorg como um console gráfico, a combinação `Ctrl` + `Alt` + `F1` é utilizada para retornar para um console virtual baseado em texto.

Ao mudar de um console para o próximo, o FreeBSD gerencia a saída da tela. O resultado é uma ilusão de ter várias telas virtuais e teclados que podem ser usados para digitar comandos para o FreeBSD rodar. Os programas executados em um console virtual não param de ser executados quando o usuário alterna para um console virtual diferente.

Consulte kbdcontrol(1), vidcontrol(1), atkbd:(4), syscons(4), e vt(4) para uma descrição mais técnica do console do FreeBSD e seus drivers de teclado.

No FreeBSD, o número de consoles virtuais disponíveis é configurado nesta seção do /etc/ttys:

```
# name    getty                           type  status comments
#
ttyv0   "/usr/libexec/getty Pc"         xterm   on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv2   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv3   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv4   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv5   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv6   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv7   "/usr/libexec/getty Pc"         xterm   on  secure
```

```
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

Para desativar um console virtual, coloque um símbolo de comentário (`#`) no início da linha que representa esse console virtual. Por exemplo, para reduzir o número de consoles virtuais disponíveis de oito para quatro, coloque um `#` na frente das últimas quatro linhas que representam os consoles virtuais de ttyv5 até ttyv8. *Não* comente a linha do console do sistema ttyv0. Note que o último console virtual (ttyv8) é usado para acessar o ambiente gráfico se o Xorg tiver sido instalado e configurado conforme descrito em O sistema X Window.

Para uma descrição detalhada de cada coluna neste arquivo e as opções disponíveis para os consoles virtuais, consulte ttys(5).

3.2.2. Modo "Single User"

O menu de inicialização do FreeBSD fornece uma opção chamada "Boot Single User". Se esta opção for selecionada, o sistema inicializará em um modo especial conhecido como "single user mode". Esse modo é normalmente usado para reparar um sistema que não inicializa ou para redefinir a senha de `root` quando ela é desconhecida. Quando em modo single user, a rede e outros consoles virtuais não estão disponíveis. No entanto, haverá acesso completo de `root` ao sistema e, por padrão, a senha de `root` não é necessária. Por estas razões, o acesso físico ao teclado é necessário para iniciar neste modo e determinar quem tem acesso físico ao teclado é algo a considerar ao proteger um sistema FreeBSD.

As configurações que controlam o modo de single user são encontradas nesta seção do /etc/ttys:

```
# name  getty                           type  status  comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown  off  secure
```

Por padrão, o status é definido como `secure`. Isso pressupõe que quem tem acesso físico ao teclado não é importante ou é controlado por uma política de segurança física. Se essa configuração for alterada para `insecure`, a suposição é que o ambiente em si é inseguro porque qualquer pessoa pode acessar o teclado. Quando esta linha é alterada para `insecure`, o FreeBSD irá solicitar a senha do `root` quando um usuário selecionar inicializar no modo single user.



Tenha cuidado ao alterar esta configuração para `inseguro`! Se a senha do `root` for esquecida, a inicialização no modo single user ainda é possível, mas pode ser difícil para alguém que não esteja familiarizado com o processo de inicialização do FreeBSD.

3.2.3. Alterar os modos de vídeo do console

O modo de vídeo padrão do console do FreeBSD pode ser ajustado para 1024x768, 1280x1024 ou qualquer outro tamanho suportado pelo chip gráfico e monitor. Para usar um modo de vídeo diferente, carregue o módulo `VESA`:

```
# kldload vesa
```

Para determinar quais modos de vídeo são suportados pelo hardware, use vidcontrol(1). Para obter uma lista de modos de vídeo suportados, execute o seguinte:

```
# vidcontrol -i mode
```

A saída deste comando lista os modos de vídeo suportados pelo hardware. Para selecionar um novo modo de vídeo, especifique o modo usando vidcontrol(1) como o usuário `root` :

```
# vidcontrol MODE_279
```

Se o novo modo de vídeo for aceitável, ele pode ser definido permanentemente na inicialização, adicionando-o ao /etc/rc.conf:

```
allscreens_flags="MODE_279"
```

3.3. Usuários e Gerenciamento Básico de Contas

O FreeBSD permite que múltiplos usuários usem o computador ao mesmo tempo. Enquanto apenas um usuário pode se sentar em frente à tela e usar o teclado a qualquer momento, qualquer número de usuários pode efetuar o login no sistema através da rede. Para usar o sistema, cada usuário deve ter sua própria conta de usuário.

Este capítulo descreve:

- Os diferentes tipos de contas de usuários em um sistema FreeBSD.
- Como adicionar, remover e modificar contas de usuários.
- Como definir limites para controlar os recursos que usuários e grupos podem acessar.
- Como criar grupos e adicionar usuários como membros de um grupo.

3.3.1. Tipos de conta

Como todo acesso ao sistema FreeBSD é obtido usando contas e todos os processos são executados por usuários, o gerenciamento de usuários e contas é importante.

Existem três tipos principais de contas: contas do sistema, contas de usuário e a conta de superusuário.

3.3.1.1. Contas do sistema

As contas do sistema são usadas para executar serviços como DNS, correio e servidores web. A razão para isso é a segurança; se todos os serviços fossem executados como superusuário, eles

poderiam agir sem restrições.

Exemplos de contas do sistema são `daemon`, `operador`, `bind`, `news`, e `www`.



É necessário ter cuidado ao usar o grupo operator, pois privilégios de acesso como o de superusuário podem ser concedidos, incluindo e não limitado a, desligamento, reinicialização e acesso a todos os itens em /dev para o grupo.

A `nobody` é uma conta genérica sem privilégios do sistema. No entanto, quanto mais serviços usarem `nobody`, a mais arquivos e processos esse usuário será associado e, portanto, mais privilegiado esse usuário se tornará.

3.3.1.2. Contas de usuário

As contas de usuários são atribuídas a pessoas reais e são usadas para efetuar login e usar o sistema. Todas as pessoas que acessam o sistema devem ter uma conta de usuário exclusiva. Isso permite que o administrador descubra quem está fazendo o que e impede que usuários alterem as configurações de outros usuários.

Cada usuário pode configurar seu próprio ambiente para adequar o sistema ao seu uso, utilizando suas opções padrão para o shell, editor, atalhos de teclado e idioma.

Cada conta de usuário em um sistema FreeBSD tem certas informações associadas:

Nome de usuário

O nome do usuário é digitado no prompt `login:`. Cada usuário deve ter um nome de usuário exclusivo. Existem diversas regras para criar nomes de usuário válidos que estão documentadas em passwd(5). Recomenda-se usar nomes de usuário que tenham oito ou menos caracteres, todos os caracteres devem ser minúsculos para manter a compatibilidade com aplicativos legados.

Senha

Cada conta tem uma senha associada.

ID do usuário (UID)

O ID do Usuário (UID) é um número usado para identificar unicamente o usuário no sistema FreeBSD. Comandos que permitem que um nome de usuário seja especificado o converterão para o UID. Recomenda-se usar um UID menor que 65535, já que valores mais altos podem causar problemas de compatibilidade com alguns softwares.

ID do grupo (GID)

O ID do grupo (GID) é um número usado para identificar unicamente o grupo principal ao qual o usuário pertence. Os grupos são um mecanismo para controlar o acesso a recursos com base no GID de um usuário, em vez de no seu UID. Isso pode reduzir significativamente o tamanho de alguns arquivos de configuração e permite que os usuários sejam membros de mais de um grupo. Recomenda-se usar um GID de 65535 ou inferior, pois GIDs mais altos podem não funcionar com alguns softwares.

Classe de login

As classes de login são uma extensão do mecanismo de grupo que fornece flexibilidade adicional ao configurar o sistema para diferentes usuários. As classes de login são discutidas em Configurando Classes de Login.

Tempo para mudança de senha

Por padrão as senhas não expiram. No entanto, a expiração de senha pode ser ativada por usuário, forçando alguns ou todos os usuários a alterar suas senhas após um determinado período de tempo.

Tempo de expiração da conta

Por padrão o FreeBSD não expira contas. Ao criar contas que precisam de uma vida útil limitada, como contas de alunos em uma escola, especifique a data de expiração da conta usando o pw(8). Após o tempo de expiração, a conta não poderá ser usada para efetuar login no sistema, embora os diretórios e arquivos da conta permaneçam no servidor.

Nome completo do usuário

O nome de usuário identifica a conta de forma única para o FreeBSD, mas não reflete necessariamente o nome real do usuário. Semelhante a um comentário, essas informações podem conter espaços, caracteres maiúsculos e ter mais de oito caracteres.

Diretório Inicial (home)

O diretório "home" é um caminho completo para um diretório no sistema. Este é o diretório inicial do usuário quando o usuário faz o login. Uma convenção comum é colocar todos os diretórios home dos usuários em /home/username ou /usr/home/username. Cada usuário armazena seus arquivos e subdiretórios pessoais em seu próprio diretório home.

Shell do usuário

O shell fornece o ambiente padrão do usuário para interagir com o sistema. Existem muitos tipos diferentes de shells e usuários experientes terão suas próprias preferências, que podem ser refletidas nas suas configurações da conta.

3.3.1.3. A conta de superusuário

A conta de superusuário, geralmente chamada de `root`, é usada para gerenciar o sistema sem limitações de privilégios. Por este motivo, não deve ser usado para tarefas do dia-a-dia, como enviar e receber e-mail, exploração geral do sistema ou programação.

O superusuário, ao contrário de outras contas de usuário, pode operar sem limites, e o uso indevido da conta de superusuário pode resultar em desastres espetaculares. As contas de usuário não podem destruir o sistema operacional por engano, por isso é recomendável fazer o login como uma conta de usuário e se tornar o superusuário somente quando um comando exigir privilégios extras.

Sempre cheque duas ou três vezes todos os comandos emitidos como superusuário, pois um espaço extra ou um caractere ausente pode causar uma perda de dados irreparável.

Existem várias maneiras de obter privilégios de superusuário. Embora seja possível efetuar login como `root`, isso é altamente desencorajado.

Em vez disso, use su(1) para se tornar o superusuário. Se `-` for especificado ao executar este comando, o usuário também herdará o ambiente do usuário root. O usuário que executa este comando deve estar no grupo `wheel` ou o comando falhará. O usuário também deve saber a senha da conta de usuário `root`.

Neste exemplo, o usuário só se torna superusuário para executar `make install`, pois isso requer privilégios de superusuário. Quando o comando é concluído, o usuário digita `exit` para deixar a conta de superusuário e retornar à sua conta de usuário.

Exemplo 2. Instalar um programa como superusuário

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

O framework integrado su(1) funciona bem para sistemas isolados ou redes pequenas com apenas um administrador. Uma alternativa é instalar o pacote ou port security/sudo. Este software fornece registro de atividades e permite ao administrador configurar quais usuários podem executar quais comandos como superusuário.

3.3.2. Gerenciando Contas

O FreeBSD fornece uma variedade de diferentes comandos para gerenciar contas de usuários. Os comandos mais comuns são descritos em Utilitários para gerenciar contas de usuários, seguidos por alguns exemplos de seu uso. Veja a página de manual para cada utilitário para maiores detalhes e exemplos de uso.

Tabela 2. Utilitários para gerenciar contas de usuários

Comando	Resumo
adduser(8)	Aplicativo de linha de comando recomendado para adicionar novos usuários.
rmuser(8)	Aplicativo de linha de comando recomendado para remover usuários.
chpass(1)	Uma ferramenta flexível para alterar as informações do usuário.
passwd(1)	Ferramenta de linha de comando para alterar senhas de usuários.
pw(8)	Uma ferramenta poderosa e flexível para modificar todos os aspectos das contas de usuário.

3.3.2.1. adduser

O programa recomendado para adicionar novos usuários é o adduser(8). Quando um novo usuário é adicionado, este programa atualiza automaticamente o /etc/passwd e o /etc/group. Ele também cria um diretório inicial para o novo usuário, copia os arquivos de configuração padrão de /usr/shared/skel e pode, opcionalmente, enviar uma nova mensagem de boas-vindas ao novo usuário. Este utilitário deve ser executado como o superusuário.

O utilitário adduser(8) é interativo e percorre as etapas para criar uma nova conta de usuário. Como visto em Adicionando um usuário no FreeBSD, insira as informações necessárias ou pressione `Enter` para aceitar o valor padrão mostrado entre colchetes. Neste exemplo, o usuário foi convidado para o grupo `wheel`, permitindo que ele se tornasse o superusuário com o uso do su(1). Quando terminar, o utilitário perguntará se deseja criar outro usuário ou finalizar o comando.

Exemplo 3. Adicionando um usuário no FreeBSD

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```



Como a senha não é mostrada quando digitada, tenha cuidado para não digitar a

> senha incorretamente ao criar a conta do usuário.

3.3.2.2. `rmuser`

Para remover completamente um usuário do sistema, execute o rmuser(8) como o superusuário. Este comando executa as seguintes etapas:

1. Remove a entrada crontab(1) do usuário, se existir.
2. Remove todas as tarefas at(1) pertencentes ao usuário.
3. Elimina todos os processos pertencentes ao usuário.
4. Remove o usuário do arquivo de senhas do sistema.
5. Opcionalmente, remove o diretório pessoal do usuário, se ele for de propriedade do usuário.
6. Remove os arquivos de mensagens recebidas pertencentes ao usuário de /var/mail.
7. Remove todos os arquivos pertencentes ao usuário das áreas de armazenamento de arquivos temporários, como /tmp.
8. Finalmente, remove o nome de usuário de todos os grupos aos quais ele pertence em /etc/group. Se um grupo ficar vazio e o nome do grupo for o mesmo que o nome de usuário, o grupo será removido. Isso complementa os grupos exclusivos por usuário criados por adduser(8).

O rmuser(8) não pode ser usado para remover contas de superusuário, pois isso quase sempre ocasiona uma de destruição em massa.

Por padrão, um modo interativo é usado, conforme mostrado no exemplo a seguir.

Exemplo 4. Remoção de contas interativas com o `rmuser`

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Removing user (jru): mailspool home passwd.
#
```

3.3.2.3. `chpass`

Qualquer usuário pode usar o chpass(1) para alterar seu shell padrão e informações pessoais associadas à sua conta de usuário. O superusuário pode usar esse utilitário para alterar informações adicionais da conta de qualquer usuário.

Quando não há opções, além de um nome de usuário opcional, o chpass(1) exibe um editor

contendo informações do usuário. Quando o usuário sai do editor, o banco de dados do usuário é atualizado com as novas informações.



Este utilitário solicitará a senha do usuário ao sair do editor, a menos que o utilitário seja executado como superusuário.

Em Usando o `chpass` como superusuário, o superusuário digitou `chpass jru` e agora está visualizando os campos que podem ser alterados para este usuário. Se `jru` executar este comando, apenas os últimos seis campos serão exibidos e estarão disponíveis para edição. Isso é mostrado em Usando o `chpass` como usuário regular.

Exemplo 5. Usando o `chpass` como superusuário

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Exemplo 6. Usando o `chpass` como usuário regular

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```



Os comandos chfn(1) e chsh(1) são links para chpass(1), como são ypchpass(1), ypchfn(1)eypchsh(1). Já que o suporte ao NIS é automático, colocar o `yp` antes do comando não é necessário. Os procedimentos para configurar o NIS está documentado em Servidores de Rede.

3.3.2.4. passwd

Qualquer usuário pode alterar facilmente sua senha usando o passwd(1). Para prevenir alterações acidentais ou não autorizadas, este comando irá solicitar a senha atual ao usuário antes de configurar uma nova senha:

Exemplo 7. Alterando Sua Senha

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

O superusuário pode alterar a senha de qualquer usuário especificando o nome de usuário ao executar o passwd(1). Quando este utilitário é executado como superusuário, ele não solicita a senha atual do usuário. Isso permite que a senha seja alterada quando um usuário não consegue lembrar a senha original.

Exemplo 8. Mudando a senha de outro usuário como superusuário

```
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```



Como com o chpass(1), o yppasswd(1) é um link para passwd(1), então o NIS funciona com ambos os comandos.

3.3.2.5. pw

O utilitário pw(8) pode criar, remover, modificar e exibir usuários e grupos. Funciona como um front-end para o usuário do sistema e para os arquivos de grupo. O pw(8) tem um conjunto muito poderoso de opções de linha de comando que o torna adequado para uso em shell scripts, mas novos usuários podem achar isso mais complicado que os outros comandos apresentados nesta seção.

3.3.3. Gerenciando Grupos

Um grupo é uma lista de usuários. Um grupo é identificado pelo nome do grupo e pelo GID. No FreeBSD, o kernel usa o UID de um processo, e a lista de grupos a que pertence, para determinar o que o processo pode fazer. Na maioria das vezes, o GID de um usuário ou processo geralmente

significa o primeiro grupo na lista.

O mapeamento do nome do grupo para o GID está listado em /etc/group. Este é um arquivo de texto simples com quatro campos delimitados por dois pontos. O primeiro campo é o nome do grupo, o segundo é a senha criptografada, o terceiro é o GID e o quarto é a lista de membros delimitados por vírgulas. Para uma descrição mais completa da sintaxe, consulte group(5).

O superusuário pode modificar o /etc/group usando um editor de texto. Alternativamente, o pw(8) pode ser usado para adicionar e editar grupos. Por exemplo, para adicionar um grupo chamado `teamtwo` e confirmar se ele existe:

Exemplo 9. Adicionando um grupo usando o pw(8)

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

Neste exemplo, `1100` é o GID de `teamtwo`. No momento, `teamtwo` não possui membros. Este comando adicionará `jru` como um membro de `teamtwo`.

Exemplo 10. Adicionando contas de usuários a um novo grupo usando o pw(8)

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

O argumento para a opção `-M` é uma lista de usuários, delimitada por vírgulas, a serem adicionados a um novo grupo (vazio) ou para substituir os membros de um grupo existente. Para o usuário, essa associação ao grupo é diferente (e adicional ao) do grupo principal do usuário listado no arquivo de senha. Isso significa que o usuário não aparecerá como membro ao usar a opção `groupshow` com o pw(8), mas mostrará quando a informação é consultada via id(1) ou uma ferramenta similar. Quando o pw(8) é usado para adicionar um usuário a um grupo, ele apenas manipula o /etc/group e não tenta ler dados adicionais do /etc/passwd.

Exemplo 11. Adicionando um novo membro a um grupo usando o pw(8)

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
teamtwo:*:1100:jru,db
```

Neste exemplo, o argumento para `-m` é uma lista delimitada por vírgulas de usuários que devem ser adicionados ao grupo. Ao contrário do exemplo anterior, esses usuários são adicionados ao grupo e não substituem usuários existentes no grupo.

Exemplo 12. Usando o id(1) para determinar a associação ao grupo

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

Neste exemplo, `jru` é um membro dos grupos `jru` e `teamtwo`.

Para obter mais informações sobre este comando e o formato do /etc/group, consulte pw(8) e group(5).

3.4. Permissões

No FreeBSD, todo arquivo e diretório tem um conjunto associado de permissões e vários utilitários estão disponíveis para visualizar e modificar essas permissões. É necessário entender como as permissões funcionam para garantir que os usuários consigam acessar os arquivos que precisam e não consigam acessar os arquivos usados pelo sistema operacional ou de propriedade de outros usuários.

Esta seção discute as permissões UNIX™ tradicionais usadas no FreeBSD. Para um controle de acesso ao sistema de arquivos mais refinado, consulte Listas de Controle de Acesso.

No UNIX™, as permissões básicas são atribuídas usando três tipos de acesso: ler, escrever e executar. Esses tipos de acesso são usados para determinar o acesso do arquivo ao proprietário, ao grupo e a outros usuários do arquivo (todos os outros). As permissões de leitura, gravação e execução podem ser representadas como as letras `r`, `w` e `x`. Elas também podem ser representados como números binários, pois cada permissão está ativada ou desativada (`0`). Quando representada como um número, a ordem é sempre lida como `rwx`, onde `r` é ativado com o valor `4`, `w` é ativado com o valor `2` e `x` é ativado com o valor `1`.

A Tabela 4.1 resume as possíveis possibilidades numéricas e alfabéticas. Ao ler a coluna "Listagem do Diretório", um `-` é usado para representar uma permissão que está desativada.

Tabela 3. Permissões UNIX™

Valor	Permissão	Listagem de diretório
0	Sem leitura, sem escrita, sem execução	`---`
1	Sem leitura, sem escrita, execução	`--x`
2	Sem leitura, escrita, sem execução	`-w-`
3	Sem leitura, escrita, execução	`-wx`
4	Leitura, sem escrita, sem execução	`r--`
5	Leitura, sem escrita, execução	`r-x`

Valor	Permissão	Listagem de diretório
6	Leitura, escrita, sem execução	`rw-`
7	Leitura, escrita, execução	`rwx`

Use o argumento `-l` com o ls(1) para exibir uma lista longa de diretórios que inclua uma coluna de informações sobre um permissões do arquivo para o proprietário, grupo e outros. Por exemplo, um `ls -l` em um diretório arbitrário pode mostrar:

```
% ls -l
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
```

O primeiro caractere (mais à esquerda) da primeira coluna indica se esse arquivo é um arquivo normal, um diretório, um dispositivo de caractere especial, um soquete ou qualquer outro dispositivo especial de pseudo-arquivo. Neste exemplo, o `-` indica um arquivo regular. Os próximos três caracteres, `rw-` neste exemplo, fornecem as permissões para o proprietário do arquivo. Os próximos três caracteres, `r--`, fornecem as permissões para o grupo ao qual o arquivo pertence. Os três últimos caracteres, `r--`, dão as permissões para o resto do mundo. Um traço significa que a permissão está desativada. Neste exemplo, as permissões são definidas para que o proprietário possa ler e gravar no arquivo, o grupo possa ler o arquivo e o resto do mundo só possa ler o arquivo. De acordo com a tabela acima, as permissões para este arquivo seriam `644`, onde cada dígito representa uma das três partes da permissão do arquivo.

Como o sistema controla as permissões nos dispositivos? O FreeBSD trata a maioria dos dispositivos de hardware como um arquivo nos quais os programas podem abrir, ler e gravar dados. Esses arquivos de dispositivos especiais são armazenados em /dev/.

Diretórios também são tratados como arquivos. Eles tem permissões de leitura, gravação e execução. O bit executável de um diretório tem um significado ligeiramente diferente que nos arquivos. Quando um diretório é marcado como executável, isso significa que é possível mudar para esse diretório usando cd(1). Isso também significa que é possível acessar os arquivos dentro desse diretório, sujeito às permissões dos próprios arquivos.

Para executar uma listagem de diretórios, a permissão de leitura deve estar ativada no diretório. Para deletar um arquivo que se conhece o nome, é necessário ter permissões de escrita *e* execução no diretório que contém o arquivo.

Há mais bits de permissão, mas eles são usados principalmente em circunstâncias especiais, como binários setuid e diretórios fixos. Para obter mais informações sobre permissões de arquivos e como configurá-las, consulte chmod(1).

3.4.1. Permissões simbólicas

Permissões simbólicas usam caracteres em vez de valores octais para atribuir permissões a arquivos ou diretórios. Permissões simbólicas usam a sintaxe de (quem) (ação) (permissões), onde

os seguintes valores estão disponíveis:

Opção	Letra	Representa
(quem)	u	Usuário
(quem)	g	Grupo
(quem)	o	Outros
(quem)	a	Todos ("resto do mundo")
(açao)	+	Adiciona permissões
(açao)	-	Remove permissões
(açao)	=	Permissões definidas explicitamente
(permissões)	r	Leitura
(permissões)	w	Escrita
(permissões)	x	Execução
(permissões)	t	bit fixador
(permissões)	s	Set UID ou GID

Esses valores são usados com o chmod(1), mas com letras em vez de números. Por exemplo, o comando a seguir impediria que outros usuários acessassem *FILE*:

```
% chmod go= FILE
```

Uma lista separada por vírgula pode ser fornecida quando mais de um conjunto de alterações em um arquivo precisar ser feito. Por exemplo, o comando a seguir remove as permissões de gravação do grupo e "resto do mundo" no *FILE* e adiciona as permissões de execução para todos:

```
% chmod go-w,a+x FILE
```

3.4.2. Flags de arquivos no FreeBSD

Além das permissões de arquivo, o FreeBSD suporta o uso de "flags de arquivo". Esses sinalizadores adicionam um nível a mais de segurança e controle sobre os arquivos, mas não nos diretórios. Com flags de arquivos, mesmo o `root` pode ser impedido de remover ou alterar arquivos.

Os sinalizadores de arquivo são modificados usando o chflags(1). Por exemplo, para ativar o sinalizador undeletable do sistema no arquivo file1, use o seguinte comando:

```
# chflags sunlink file1
```

Para desabilitar o sinalizador undeletable do sistema, coloque um "no" na frente do `sunlink`:

```
# chflags nosunlink file1
```

Para visualizar os sinalizadores de um arquivo, use `-lo` com o ls(1):

```
# ls -lo file1
```

```
-rw-r--r-- 1 trhodes trhodes sunlnk 0 Mar 1 05:54 file1
```

Vários flags de arquivo só podem ser adicionados ou removidos pelo usuário `root`. Em outros casos, o proprietário do arquivo pode definir seus sinalizadores. Consulte chflags(1) e chflags(2) para maiores informações.

3.4.3. As permissões `setuid`, `setgid` e `sticky`

Além das permissões já discutidas, existem três outras configurações específicas que todos os administradores devem conhecer. Eles são as permissões `setuid`, `setgid` e `sticky`.

Essas configurações são importantes para algumas operações UNIX™, pois fornecem funcionalidades normalmente não concedidas a usuários normais. Para compreendê-los, a diferença entre o ID real de usuário e o ID efetivo de usuário deve ser explicada.

O ID de usuário real é o UID que inicia ou é o dono do processo. O ID de usuário efetivo é o UID do usuário com o qual o processo é executado. Por exemplo, o passwd(1) é executado com o ID do usuário real quando um usuário altera sua senha. No entanto, para atualizar o banco de dados de senhas, o comando é executado como o ID efetivo do usuário `root`. Isso permite que os usuários alterem suas senhas sem ver um erro `Permission Denied`.

A permissão setuid pode ser definida prefixando um conjunto de permissões com o número quatro (4), conforme mostrado no exemplo a seguir:

```
# chmod 4755 suidexample.sh
```

As permissões em suidexample.sh agora se parecem com o seguinte:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidexample.sh
```

Observe que um `s` agora faz parte do conjunto de permissões designado para o proprietário do arquivo, substituindo o bit executável. Isso viabiliza utilitários que precisam de permissões elevadas, como o passwd(1).



A opção `nosuid` mount(8) fará com que esses binários falhem silenciosamente sem alertar o usuário. Essa opção não é totalmente confiável, já que um wrapper `nosuid` pode contorná-la.

Para ver isso em tempo real, abra dois terminais. Em um deles, digite `passwd` como um usuário normal. Enquanto aguarda uma nova senha, verifique a tabela de processos e observe as informações de usuário do passwd(1):

No terminal A:

```
Changing local password for trhodes
Old Password:
```

No terminal B:

```
# ps aux | grep passwd
```

```
trhodes  5232  0.0  0.2  3420  1608   0  R+    2:10AM   0:00.00 grep passwd
root     5211  0.0  0.2  3620  1724   2  I+    2:09AM   0:00.01 passwd
```

Embora passwd(1) seja executado como um usuário normal, ele está usando o UID do `root`.

A permissão `setgid` executa a mesma função que a permissão `setuid`; exceto que altera as configurações do grupo. Quando um aplicativo ou utilitário é executado com essa configuração, ele recebe as permissões com base no grupo do arquivo, não no usuário que iniciou o processo.

Para definir a permissão `setgid` em um arquivo, execute o chmod(1) com dois (2) no início:

```
# chmod 2755 sgidexample.sh
```

Na listagem a seguir, observe que o `s` está agora no campo designado para as configurações de permissão do grupo:

```
-rwxr-sr-x   1 trhodes  trhodes    44 Aug 31 01:49 sgidexample.sh
```



Nestes exemplos, mesmo que o shell script em questão seja um arquivo executável, ele não será executado com um EUID diferente ou um ID de usuário efetivo. Isso ocorre porque os shell scripts podem não acessar as chamadas de sistema setuid(2).

Os bits de permissão `setuid` e `setgid` podem diminuir a segurança do sistema, permitindo permissões elevadas. A terceira permissão especial, o `sticky bit`, pode fortalecer a segurança de um sistema.

Quando o `sticky bit` é definido em um diretório, ele permite a exclusão de arquivos apenas pelo proprietário do arquivo. Isso é útil para impedir a exclusão de arquivos em diretórios públicos, como /tmp, por usuários que não possuem o arquivo. Para utilizar essa permissão, use o um (1) no início das permissões:

```
# chmod 1777 /tmp
```

A permissão `sticky bit` será exibida como um `t` no final do conjunto de permissões:

```
# ls -al / | grep tmp
```

```
drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

3.5. Estrutura de Diretórios

Entender a hierarquia de diretórios do FreeBSD é fundamental para obter uma compreensão geral do sistema. O diretório mais importante é o root ou raiz ou "/". Esse diretório é o primeiro montado no momento da inicialização e contém a base do sistema necessária para preparar o sistema operacional para a operação multi-usuário. O diretório raiz também contém pontos de montagem para outros sistemas de arquivos que são montados durante a transição para a operação multi-usuário.

Um ponto de montagem é um diretório no qual sistemas de arquivos adicionais podem ser disponibilizados em um sistema de arquivos principal (geralmente o sistema de arquivos raiz). Isso é descrito em Organização dos Discos. Os pontos de montagem padrão incluem /usr/, /var/, /tmp/, /mnt/ e /cdrom/. Esses diretórios são geralmente associados a entradas em /etc/fstab. Este arquivo é uma tabela de vários sistemas de arquivos e pontos de montagem e é lido pelo sistema. A maioria dos sistemas de arquivos em /etc/fstab é montada automaticamente no momento da inicialização do script rc(8) a não ser que haja a opção `noauto`. Maiores detalhes em O arquivo fstab.

Uma descrição completa da hierarquia do sistema de arquivos está disponível em hier(7). A tabela a seguir fornece uma visão geral dos diretórios mais comuns.

Diretório	Descrição
/	Diretório raiz do sistema de arquivos.
/bin/	Utilitários de usuário fundamentais para ambientes mono e multi-usuário.
/boot/	Programas e arquivos de configuração usados durante o bootstrap do sistema operacional.
/boot/defaults/	Arquivos de configuração de inicialização padrão. Consulte loader.conf(5) para maiores detalhes.
/dev/	Nós de dispositivo (device nodes). Consulte intro(4) para detalhes.
/etc/	Arquivos de configuração do sistema e scripts.
/etc/defaults/	Arquivos padrão de configuração do sistema. Consulte rc(8) para maiores detalhes.

Diretório	Descrição
/etc/mail/	Arquivos de configuração para agentes de transporte de mensagens, como o sendmail(8).
/etc/periodic/	Scripts que são executados diariamente, semanalmente e mensalmente, por meio do cron(8). Consulte periodic(8) para maiores detalhes.
/etc/ppp/	Arquivos de configuração do ppp(8).
/mnt/	Diretório vazio comumente usado pelos administradores do sistema como um ponto de montagem temporário.
/proc/	Sistema de arquivos de processos. Consulte procfs(5), mount_procfs(8) para detalhes.
/rescue/	Programas vinculados estaticamente para recuperação de emergência, conforme descrito em rescue(8).
/root/	Diretório da conta `root`.
/sbin/	Programas do sistema e utilitários de administração fundamentais para ambientes mono e multi-usuário.
/tmp/	Arquivos temporários que normalmente *não* são preservados em uma reinicialização do sistema. Um sistema de arquivos baseado em memória é frequentemente montado em /tmp. Isso pode ser automatizado usando as variáveis relacionadas ao tmpmfs do rc.conf(5) ou com uma entrada em /etc/fstab; consulte mdmfs(8) para maiores detalhes.
/usr/	A maioria dos utilitários e aplicativos do usuário.
/usr/bin/	Utilitários comuns, ferramentas de programação e aplicativos.
/usr/include/	Arquivos para "include" do C padrão.
/usr/lib/	Arquivos de biblioteca.
/usr/libdata/	Diversos arquivos de dados de utilitários.
/usr/libexec/	Daemons do sistema e utilitários do sistema executados por outros programas.

Diretório	Descrição
/usr/local/	Executáveis e bibliotecas locais. Também é usado como o destino padrão para o framework do ports do FreeBSD. Dentro do /usr/local, o layout geral esboçado por hier(7) para /usr deve ser usado. Exceções são o diretório man, que está diretamente sob /usr/local em vez de sob /usr/local/share, e a documentação do ports está em share/doc/port.
/usr/obj/	Árvore de destino específica da arquitetura produzida pela construção da árvore /usr/src.
/usr/ports/	A Coleção de Ports do FreeBSD (opcional).
/usr/sbin/	Daemons do sistema e utilitários do sistema executados pelos usuários.
/usr/shared/	Arquivos independentes de arquitetura.
/usr/src/	Arquivos do código-fonte do BSD.
/var/	Arquivos de log de múltiplos propósitos, temporários, transientes e de spool. Um sistema de arquivos baseado em memória às vezes é montado em /var. Isso pode ser automatizado usando as variáveis relacionadas ao varmfs em rc.conf(5) ou com uma entrada em /etc/fstab; consulte mdmfs(8) para maiores detalhes.
/var/log/	Diversos arquivos de log do sistema.
/var/mail/	Arquivos de caixa de correio do usuário.
/var/spool/	Diretórios de spooling de impressoras e sistemas de email.
/var/tmp/	Arquivos temporários que geralmente são preservados em uma reinicialização do sistema, a menos que /var seja um sistema de arquivos baseado em memória.
/var/yp/	Mapas de NIS.

3.6. Organização dos Discos

A menor unidade de organização que o FreeBSD usa para encontrar arquivos é o nome do arquivo. Os nomes dos arquivos diferenciam maiúsculas de minúsculas, o que significa que readme.txt e README.TXT são dois arquivos distintos. O FreeBSD não usa a extensão de um arquivo para determinar se é um programa, documento ou alguma outra forma de dados.

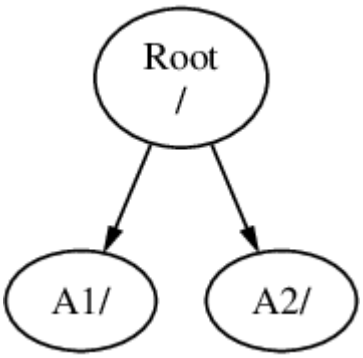
Os arquivos são armazenados em diretórios. Um diretório pode não conter arquivos ou pode conter centenas deles. Um diretório também pode conter outros diretórios, permitindo uma hierarquia de diretórios entre si para organizar os dados.

Arquivos e diretórios são referenciados por meio de um nome, seguido por uma barra, `/`, seguido por qualquer outro nome de diretório que seja necessário. Por exemplo, se o diretório foo contiver um diretório bar que contenha o arquivo readme.txt, o nome completo ou *caminho*, para o arquivo é foo/bar/readme.txt. Observe que isso é diferente do Windows™ que usa `\` para separar nomes de arquivos e diretórios. O FreeBSD não usa letras de unidades ou outros nomes de unidades no caminho. Por exemplo, não se deve digitar c:\foo\bar\readme.txt no FreeBSD.

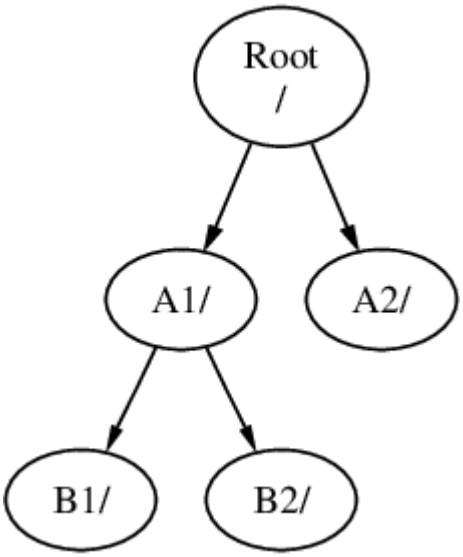
Diretórios e arquivos são armazenados em um sistema de arquivos. Cada sistema de arquivos contém exatamente um diretório no nível superior, chamado de *diretório raiz* para esse sistema de arquivos. Este diretório raiz pode conter outros diretórios. Um sistema de arquivos é designado como *sistema de arquivos raiz* ou `/`. Todos os outros sistemas de arquivos são *montados* no sistema de arquivos raiz. Não importa quantos discos estejam no sistema FreeBSD, cada diretório parece fazer parte do mesmo disco.

Considere três sistemas de arquivos, chamados `A`, `B` e `C`. Cada sistema de arquivos tem um diretório raiz, que contém dois outros diretórios, chamados `A1`, `A2` (e da mesma forma `B1`, `B2` e `C1`, `C2`).

Chame `A` de sistema de arquivos raiz. Se ls(1) for usado para visualizar o conteúdo deste diretório, ele mostrará dois subdiretórios, `A1` e `A2`. A árvore de diretórios tem esta aparência:

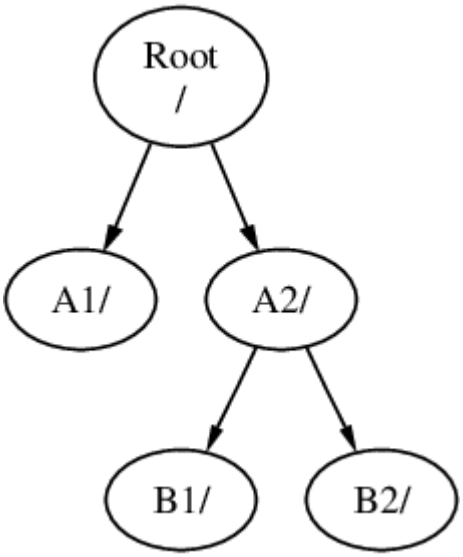


Um sistema de arquivos deve ser montado em um diretório em outro sistema de arquivos. Ao montar o sistema de arquivos `B` no diretório `A1`, o diretório raiz de `B` substitui `A1` e os diretórios em `B` aparecem de acordo:



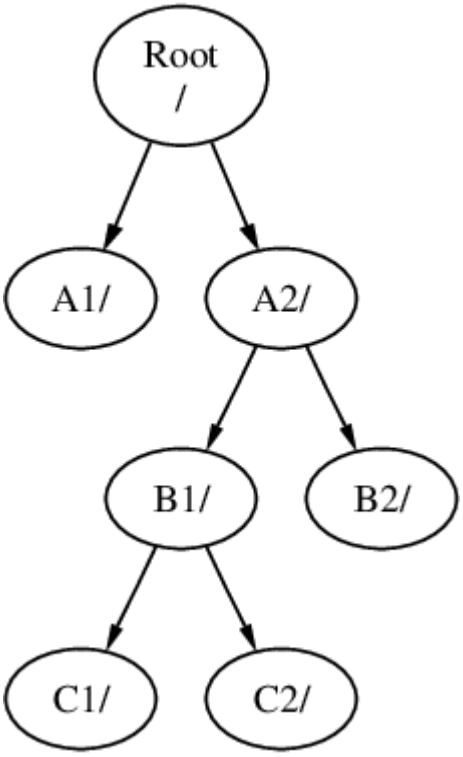
Todos os arquivos que estão nos diretórios B1 ou B2 podem ser alcançados com o caminho /A1/B1 ou /A1/B2, conforme necessário. Todos os arquivos que estavam em /A1 foram temporariamente ocultados. Eles reaparecerão se B for *desmontado* de A.

Se B tivesse sido montado em A2, o diagrama ficaria assim:

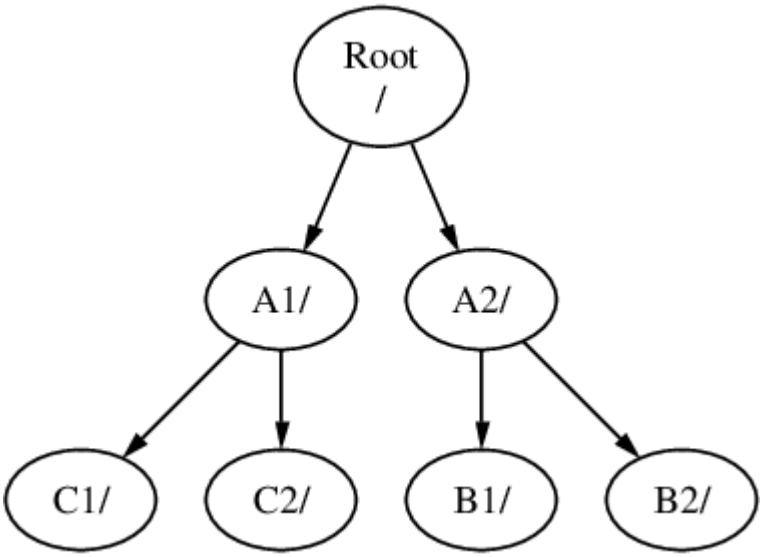


e os caminhos seriam /A2/B1 e /A2/B2 respectivamente.

Os sistemas de arquivos podem ser montados uns em cima dos outros. Continuando o último exemplo, o sistema de arquivos C pode ser montado no topo do diretório B1 no sistema de arquivos B, levando a esta disposição:



Ou C poderia ser montado diretamente no sistema de arquivos A, sob o diretório A1:



É perfeitamente possível ter um sistema de arquivos raiz grande e não precisar criar nenhum outro. Existem algumas desvantagens nessa abordagem e uma vantagem.

Benefícios de vários sistemas de arquivos

- Sistemas de arquivos diferentes podem ter diferentes *opções de montagem*. Por exemplo, o sistema de arquivos raiz pode ser montado somente para leitura, impossibilitando que os usuários excluam ou editem inadvertidamente um arquivo crítico. Separar sistemas de arquivos graváveis pelo usuário, como /home, de outros sistemas de arquivos permite que eles sejam montados como *nosuid*. Essa opção impede que os bits *suid*/*guid* dos executáveis armazenados no sistema de arquivos entrem em vigor, possivelmente melhorando a segurança.
- O FreeBSD otimiza automaticamente o layout dos arquivos em um sistema de arquivos, dependendo de como o sistema de arquivos está sendo usado. Portanto, um sistema de arquivos que contém muitos arquivos pequenos que são gravados com freqüência terá uma otimização diferente para um que contenha menos arquivos maiores. Ao ter um sistema de arquivos maior, essa otimização é quebrada.
- Os sistemas de arquivos do FreeBSD são robustos se a energia for perdida. No entanto, uma perda de energia em um ponto crítico ainda pode danificar a estrutura do sistema de arquivos. Ao dividir dados em vários sistemas de arquivos, é mais provável que o sistema ainda inicialize, facilitando a restauração do backup conforme necessário.

Benefício de um sistema de arquivos único

- Os sistemas de arquivos são de tamanho fixo. Se você cria um sistema de arquivos quando instala o FreeBSD e dá a ele um tamanho específico, você pode descobrir mais tarde que precisa aumentar a partição. Isso não é facilmente realizado sem um backup, recriando o sistema de arquivos com o novo tamanho e, em seguida, restaurando os dados de backup.



O FreeBSD possui o comando growfs(8), que torna possível aumentar o tamanho do sistema de arquivos enquanto montado, removendo essa limitação.

Os sistemas de arquivos estão contidos em partições. Isto não tem o mesmo significado que o uso comum do termo partição (por exemplo, a partição MS-DOS™), por causa da herança UNIX™ do

FreeBSD. Cada partição é identificada por uma letra de a até h. Cada partição pode conter apenas um sistema de arquivos, o que significa que os sistemas de arquivos geralmente são descritos por seu ponto de montagem típico na hierarquia do sistema de arquivos ou pela letra da partição em que estão contidos.

O FreeBSD também usa espaço em disco para *espaço de swap* para fornecer *memória virtual*. Isso permite que o seu computador se comporte como se tivesse muito mais memória do que realmente tem. Quando o FreeBSD fica sem memória, ele move alguns dos dados que não estão sendo usados atualmente para o espaço de swap, e os move de volta (removendo alguma outra coisa) quando precisa.

Algumas partições possuem certas convenções associadas a elas.

Partição	Convenção
a	Normalmente contém o sistema de arquivos raiz.
b	Normalmente contém espaço de swap.
c	Normalmente o mesmo tamanho da slice que a envolve. Isso permite que os programas que precisem trabalhar na slice inteira, como um scanner de bloco defeituoso, trabalhem na partição c. Um sistema de arquivos normalmente não seria criado nessa partição.
d	A partição d costumava ter um significado especial associado a ela, mas isso foi descontinuado e d pode funcionar como qualquer partição normal.

Os discos no FreeBSD são divididos em slices, referidas no Windows™ como partições, numeradas de 1 a 4. Estas são então divididas em partições, que contêm sistemas de arquivos, e são rotuladas usando letras.

Os números das slices seguem o nome do dispositivo, prefixado com um s, começando em 1. Então "da0*s1*" é a primeira slice na primeira unidade SCSI. Pode haver apenas quatro slices físicas em um disco, mas pode haver slices lógicas dentro de slices físicas do tipo apropriado. Essas slices estendidas são numeradas a partir de 5, então "ada0*s5*" é a primeira slice estendida no primeiro disco SATA. Esses dispositivos são usados por sistemas de arquivos que esperam ocupar uma slice.

Slices, unidades físicas "perigosamente dedicadas" e outras unidades contêm *partições*, que são representadas como letras de a até h. Esta letra é adicionada ao nome do dispositivo, então "da0*a*" é a partição a na primeira unidade da, que é "perigosamente dedicada". A "ada1s3*e*" é a quinta partição na terceira slice da segunda unidade de disco SATA.

Finalmente, cada disco no sistema é identificado. Um nome de disco começa com um código que indica o tipo de disco e, em seguida, um número, indicando qual é o disco. Ao contrário das slices, a numeração de discos começa em 0. Códigos usuais são listados em Nomes de dispositivos de disco.

Ao se referir a uma partição, inclua o nome do disco, s, o número da slice, em seguida, a letra da

partição. Exemplos são mostrados em Exemplo de Nomes de Disco, Slice e Partição.

Modelo conceitual de um disco mostra um modelo conceitual de um layout de disco.

Ao instalar o FreeBSD, configure as slices de disco, crie partições dentro da slice a ser usada para o FreeBSD, crie um sistema de arquivos ou espaço de swap em cada partição e decida onde cada sistema de arquivos será montado.

Tabela 4. Nomes de dispositivos de disco

Tipo de drive	Nome do drive
discos rígidos SATA e IDE	`ada` ou `ad`
Discos rígidos SCSI e dispositivos de armazenamento USB	`da`
drives de CD-ROMSATA e IDE	`cd` ou `acd`
Unidades SCSICD-ROM	`cd`
Unidades de disquete	`fd`
Unidades de CD-ROM não-padrão variadas	`mcd` para CD-ROM Mitsumi e `scd` para dispositivos de CD-ROM Sony
Unidades de fita SCSI	`sa`
Unidades de fita IDE	`ast`
Drives RAID	Exemplos incluem `aacd` para Adaptec™ AdvancedRAID, `mlxd` e `mlyd` para Mylex™, `amrd` para AMI MegaRAID™, `idad` para Compaq Smart RAID, `twed` para 3ware™ RAID.

Exemplo 13. Exemplo de Nomes de Disco, Slice e Partição

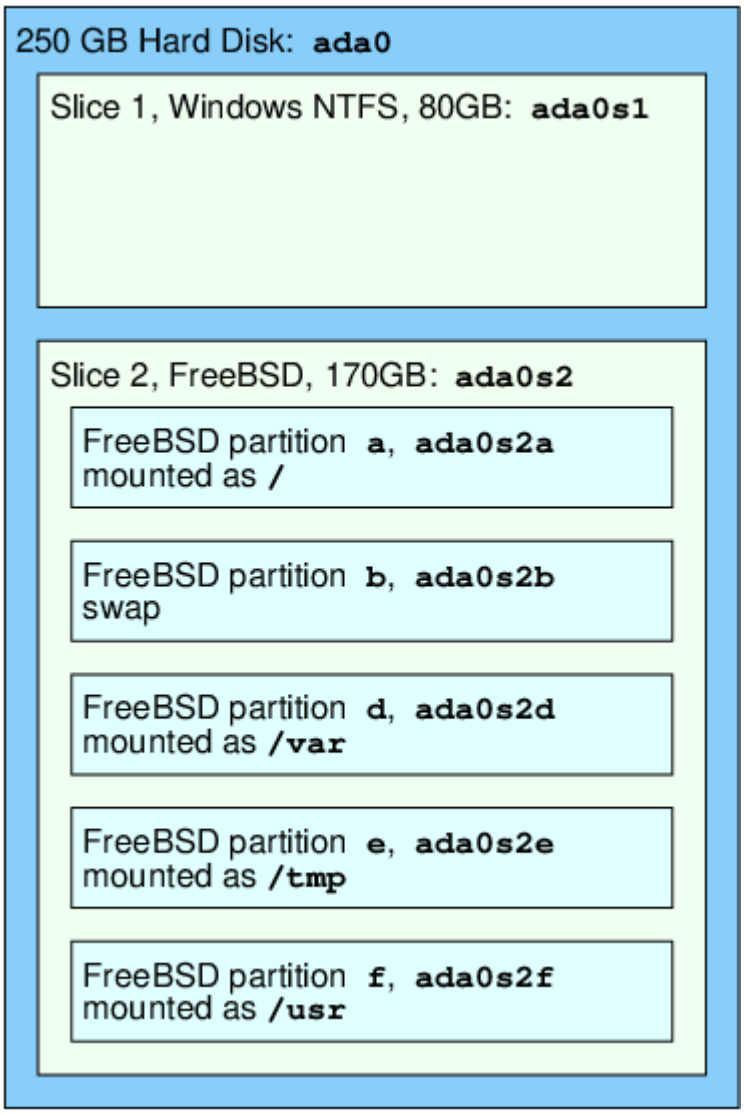
Nome	Significado
`ada0s1a`	A primeira partição (`a`) na primeira slice (`s1`) no primeiro disco SATA (`ada0`).
`da1s2e`	A quinta partição (`e`) na segunda slice (`s2`) no segundo disco SCSI (`da1`).

Exemplo 14. Modelo conceitual de um disco

Este diagrama mostra a visão do FreeBSD do primeiro disco SATA conectado ao sistema. Suponha que o disco tenha 250 GB de tamanho e contenha uma slice de 80 GB e uma slice de 170 GB (partições MS-DOS™). A primeira slice contém um sistema de arquivos Windows™NTFS, C:, e a segunda fatia contém uma instalação do FreeBSD. Este exemplo de instalação do FreeBSD possui quatro partições de dados e uma partição swap.

Cada uma das quatro partições contém um sistema de arquivos. A partição `a` é usada para o sistema de arquivos raiz, `d` para /var/, `e` para /tmp/ e `f` para /usr/. A letra de partição `c` refere-se

à fatia inteira e, portanto, não é usada para partições comuns.



3.7. Montando e Desmontando Sistemas de Arquivos

O sistema de arquivos é melhor visualizado como uma árvore, enraizada, por assim dizer, em /. O /dev, /usr, e os outros diretórios no diretório raiz são ramos, que podem ter suas próprias ramificações, como /usr/local e assim por diante.

Existem várias razões para abrigar alguns desses diretórios em sistemas de arquivos separados. O /var contém os diretórios log/, spool/ e vários tipos de arquivos temporários e, como tal, podem encher. Encher completamente o sistema de arquivos raiz não é uma boa ideia, então separar o /var do / geralmente é vantajoso.

Outro motivo comum para colocar determinadas árvores de diretório em outros sistemas de arquivos é se elas forem ser armazenadas em discos físicos separados ou se são discos virtuais separados, tal como montagens de NFS (Network File System), descritas em Network File System (NFS) ou unidades de CD-ROM.

3.7.1. O arquivo fstab

Durante o processo de inicialização (O processo de inicialização do FreeBSD), os sistemas de arquivos listados em /etc/fstab são automaticamente montados, exceto pelas entradas que contêm `noauto`. Este arquivo contém entradas no seguinte formato:

```
device       /mount-point fstype     options      dumpfreq     passno
```

`device`
: Um nome de dispositivo existente, conforme explicado em Nomes de dispositivos de disco.

`mount-point`
: Um diretório existente no qual montar o sistema de arquivos.

`fstype`
: O tipo de sistema de arquivos para passar para o mount(8). O sistema de arquivos padrão do FreeBSD é o `ufs`.

`options`
: `rw` para sistemas de arquivos de leitura/gravação, ou `ro` para sistemas de arquivos somente de leitura, seguidos por quaisquer outras opções que possam ser necessárias. Uma opção comum é `noauto` para sistemas de arquivos normalmente não montados durante a seqüência de inicialização. Outras opções estão listadas em mount(8).

`dumpfreq`
: Usado pelo dump(8) para determinar quais sistemas de arquivos requerem o dump. Se o campo estiver faltando, um valor zero será assumido.

`passno`
: Determina a ordem em que os sistemas de arquivos devem ser verificados. Os sistemas de arquivos que devem ser ignorados devem ter seu `passno` definido como zero. O sistema de arquivos raiz precisa ser verificado antes de todo o restante e deve ter seu `passno` definido como um. Os outros sistemas de arquivos devem ser configurados para valores maiores que um. Se mais de um sistema de arquivos tiver o mesmo `passno`, o fsck(8) tentará verificar os sistemas de arquivos em paralelo, se possível.

Consulte fstab(5) para obter maiores informações sobre o formato do /etc/fstab e suas opções.

3.7.2. Usando o mount(8)

Os sistemas de arquivos são montados usando o comando mount(8). A sintaxe mais básica é a seguinte:

```
# mount device mountpoint
```

Este comando fornece muitas opções que são descritas em mount(8). As opções mais usadas

incluem:

Opções de montagem

`-a`

Monte todos os sistemas de arquivos listados em /etc/fstab, exceto aqueles marcados como "noauto", excluídos pela opção `-t`, ou aqueles que já estão montados.

`-d`

Faz tudo, exceto a chamada real do sistema de montagem. Esta opção é útil em conjunto com a opção `-v` para determinar o que o mount(8) está realmente tentando fazer.

`-f`

Força a montagem de um sistema de arquivos sujo (perigoso) ou a revogação do acesso de gravação ao fazer o downgrade do status de montagem de um sistema de arquivos de leitura/gravação para somente leitura.

`-r`

Monta o sistema de arquivos somente para leitura. Isso é idêntico ao uso de `-o ro`.

`-t` ***fstype***

Monta o tipo de sistema de arquivos especificado ou monta somente sistemas de arquivos do tipo especificado, se `-a`estiver incluído. "ufs" é o tipo de sistema de arquivos padrão.

`-u`

Atualiza as opções de montagem no sistema de arquivos.

`-v`

Fica verboso (mostra mais informações).

`-w`

Monta o sistema de arquivos para leitura/gravação.

As seguintes opções podem ser passadas para `-o` como uma lista separada por vírgula:

nosuid

Não interprete flags setuid ou setgid no sistema de arquivos. Essa também é uma opção de segurança útil.

3.7.3. Usando o umount(8)

Para desmontar um sistema de arquivos use umount(8). Esse comando usa um parâmetro que pode ser um ponto de montagem, um nome do dispositivo, `-a` ou `-A`.

Todos os usos aceitam `-f` para forçar a desmontagem e `-v` para ver mais informações. Atenção, em geral `-f` não é uma boa opção, pois pode travar o computador ou danificar os dados no sistema de arquivos.

Para desmontar todos os sistemas de arquivos montados, ou apenas os tipos de sistema de arquivos listados após `-t`, use `-a` ou `-A`. Note que `-A` não tenta desmontar o sistema de arquivos raiz.

3.8. Processos e Daemons

O FreeBSD é um sistema operacional multitarefa. Cada programa em execução a qualquer momento é chamado de *processo*. Todo comando em execução inicia pelo menos um novo processo e há vários processos de sistema que são executados pelo FreeBSD.

Cada processo é identificado exclusivamente por um número chamado *ID do processo* (PID). Semelhante aos arquivos, cada processo tem um proprietário e um grupo, e as permissões de proprietário e grupo são usadas para determinar quais arquivos e dispositivos o processo pode abrir. A maioria dos processos também possui um processo pai que os iniciou. Por exemplo, o shell é um processo e qualquer comando iniciado no shell é um processo que tem o shell como seu processo pai. A exceção é um processo especial chamado init(8) que é sempre o primeiro processo a rodar na inicialização e que sempre possui um PID de 1.

Alguns programas não são projetados para serem executados com a entrada contínua do usuário e desconectam do terminal na primeira oportunidade. Por exemplo, um servidor da Web responde a solicitações da Web, em vez de entradas do usuário. Servidores de email são outro exemplo desse tipo de aplicativo. Esses tipos de programas são conhecidos como *daemons*. O termo daemon vem da mitologia grega e representa uma entidade que não é boa nem má, e que invisivelmente realiza tarefas úteis. É por isso que o mascote do BSD é o daemon de aparência alegre com tênis e um tridente.

Existe uma convenção para nomear programas que normalmente são executados como daemons com um "d" à direita. Por exemplo, BIND é o Berkeley Internet Name Domain, mas o programa real que é executado é `named`. O programa do servidor da web Apache é o `httpd` e o daemon de spooling da impressora de linha é o `lpd`. Esta é apenas uma convenção de nomenclatura. Por exemplo, o daemon de correio principal para o aplicativo Sendmail é o `sendmail` e não `maild`.

3.8.1. Visualizando Processos

Para ver os processos em execução no sistema, use o ps(1) ou o top(1). Para exibir uma lista estática dos processos em execução no momento, seus PIDs, quanta memória eles estão usando e o comando com o qual eles foram iniciados, use o ps(1). Para exibir todos os processos em execução e atualizar a exibição a cada poucos segundos para ver interativamente o que o computador está fazendo, use o top(1).

Por padrão, o ps(1) mostra apenas os comandos que estão em execução e que são de propriedade do usuário. Por exemplo:

```
% ps
  PID TT  STAT    TIME COMMAND
 8203  0  Ss   0:00.59 /bin/csh
 8895  0  R+   0:00.00 ps
```

A saída do ps(1) é organizada em várias colunas. A coluna `PID` exibe o ID do processo. Os PIDs são atribuídos a partir de 1, vão até 99999, e depois retornam ao início. No entanto, um PID não é reatribuído se já estiver em uso. A coluna `TT` mostra o tty em que o programa está sendo executado e `STAT` mostra o estado do programa. `TIME` é a quantidade de tempo que o programa foi executado na

CPU. Normalmente, esse não é o tempo decorrido desde que o programa foi iniciado, pois a maioria dos programas gasta muito tempo esperando que as coisas aconteçam antes que precisem gastar tempo na CPU. Finalmente, `COMMAND` é o comando que foi usado para iniciar o programa.

Várias opções diferentes estão disponíveis para alterar as informações exibidas. Um dos conjuntos mais úteis é `auxww`, onde `a` exibe informações sobre todos os processos em execução de todos os usuários, `u` exibe o nome de usuário e o uso de memória do proprietário do processo, `x` exibe informações sobre os processos do daemon e `ww` faz com que o ps(1) exiba a linha de comando completa para cada processo, em vez de truncá-la para caber na tela quando é muito longa.

A saída do top(1) é semelhante a abaixo:

```
% top
last pid:  9609;  load averages:  0.56,  0.45,  0.36              up 0+00:20:03
10:21:46
107 processes: 2 running, 104 sleeping, 1 zombie
CPU:  6.2% user,  0.1% nice,  8.2% system,  0.4% interrupt, 85.1% idle
Mem: 541M Active, 450M Inact, 1333M Wired, 4064K Cache, 1498M Free
ARC: 992M Total, 377M MFU, 589M MRU, 250K Anon, 5280K Header, 21M Other
Swap: 2048M Total, 2048M Free

  PID USERNAME    THR PRI NICE   SIZE    RES STATE   C   TIME   WCPU COMMAND
  557 root          1 -21  r31   136M 42296K select  0   2:20  9.96% Xorg
 8198 dru           2  52    0   449M 82736K select  3   0:08  5.96% kdeinit4
 8311 dru          27  30    0  1150M   187M uwait   1   1:37  0.98% firefox
  431 root          1  20    0 14268K  1728K select  0   0:06  0.98% moused
 9551 dru           1  21    0 16600K  2660K CPU3    3   0:01  0.98% top
 2357 dru           4  37    0   718M   141M select  0   0:21  0.00% kdeinit4
 8705 dru           4  35    0   480M    98M select  2   0:20  0.00% kdeinit4
 8076 dru           6  20    0   552M   113M uwait   0   0:12  0.00% soffice.bin
 2623 root          1  30   10 12088K  1636K select  3   0:09  0.00% powerd
 2338 dru           1  20    0   440M 84532K select  1   0:06  0.00% kwin
 1427 dru           5  22    0   605M 86412K select  1   0:05  0.00% kdeinit4
```

A saída é dividida em duas seções. O cabeçalho (as primeiras cinco ou seis linhas) mostra o PID do último processo executado, as médias de carga do sistema (que são uma medida de quão ocupado o sistema está), o tempo de atividade do sistema desde a última reinicialização) e a hora atual. As outras informações no cabeçalho se relacionam com quantos processos estão sendo executados, quanta memória e swap estão em uso e quanto tempo o sistema está gastando em diferentes estados da CPU. Se o módulo do sistema de arquivos ZFS foi carregado, uma linha `ARC` indica a quantidade de dados que foram lidos do cache de memória, e não do disco.

Abaixo do cabeçalho há uma série de colunas contendo informações semelhantes à saída do ps(1), como o PID, nome de usuário, quantidade de tempo de CPU e o comando que iniciou o processo. Por padrão, o top(1) também exibe a quantidade de espaço de memória ocupada pelo processo. Isso é dividido em duas colunas: uma para o tamanho total e outra para o tamanho do residente. O tamanho total é a quantidade de memória que o aplicativo precisa e o tamanho de residente é o quanto ele está realmente usando agora.

O top(1) atualiza automaticamente a exibição a cada dois segundos. Um intervalo diferente pode ser especificado com `-s`.

3.8.2. Matando Processos

Uma maneira de se comunicar com qualquer processo ou daemon em execução é enviar um *sinal* usando o kill(1). Existem vários sinais diferentes; alguns têm um significado específico, enquanto outros são descritos na documentação do comando. Um usuário só pode enviar um sinal para um processo que seja seu. Enviar um sinal para o processo de outra pessoa resultará em um erro de permissão negada. A exceção é o usuário `root`, que pode enviar sinais para os processos de qualquer pessoa.

O sistema operacional também pode enviar um sinal para um processo. Se um aplicativo estiver mal escrito e tentar acessar a memória que não deveria, o FreeBSD enviará ao processo o sinal de "Segmentation Violation" (`SIGSEGV`). Se uma aplicação foi escrita para usar a chamada de sistema alarm(3) para ser alertada após um período de tempo, será enviado o sinal "Alarm" (`SIGALRM`).

Dois sinais podem ser usados para interromper um processo: `SIGTERM` e `SIGKILL`. `SIGTERM` é a maneira educada de eliminar um processo, pois o processo pode ler o sinal, fechar quaisquer arquivos de log que possam estar abertos e tentar terminar o que está fazendo antes de desligar. Em alguns casos, um processo pode ignorar `SIGTERM` se estiver no meio de alguma tarefa que não pode ser interrompida.

`SIGKILL` não pode ser ignorado por um processo. Enviar um `SIGKILL` para um processo geralmente interromperá esse processo de uma vez por todas. .

Outros sinais comumente usados são `SIGHUP`, `SIGUSR1` e `SIGUSR2`. Como esses são sinais de finalidade geral, diferentes aplicativos responderão de maneira diferente.

Por exemplo, depois de alterar o arquivo de configuração de um servidor da Web, o servidor da Web precisa ser instruído a reler sua configuração. Reiniciar o `httpd` resultaria em um breve período de interrupção no servidor da web. Em vez disso, envie ao daemon o sinal `SIGHUP`. Esteja ciente de que diferentes daemons terão um comportamento diferente, então consulte a documentação do daemon para determinar se `SIGHUP` terá os resultados desejados.

Procedure: Enviando um sinal para um processo

Este exemplo mostra como enviar um sinal para o inetd(8). O arquivo de configuração do inetd(8) é o /etc/inetd.conf e o inetd(8) irá reler este arquivo de configuração quando for enviado um `SIGHUP`.

1. Encontre o PID do processo para enviar o sinal usando pgrep(1). Neste exemplo, o PID do inetd(8) é 198:

```
% pgrep -l inetd
198  inetd -wW
```

2. Use o kill(1) para enviar o sinal. Como o inetd(8) é de propriedade do `root`, use o su(1) para

se tornar `root` primeiro.

```
% su
Password:
# /bin/kill -s HUP 198
```

Como a maioria dos comandos UNIX™, o kill(1) não imprimirá nenhuma saída se for bem-sucedido. Se um sinal for enviado para um processo que não pertence ao usuário, a mensagem `kill: PID: Operation not permitted` será exibida. Errar o PID irá enviar o sinal para o processo errado, o que poderia ter resultados negativos, ou enviará o sinal para um PID que não esteja em uso no momento, resultando em o erro `kill: PID: No such process`.



Por que usar o `/bin/kill`?

Muitos shells fornecem o `kill` como um comando interno, o que significa que o shell enviará o sinal diretamente, em vez de executar o /bin/kill. Esteja ciente de que diferentes shells possuem uma sintaxe diferente para especificar o nome do sinal a ser enviado. Em vez de tentar aprender todos eles, pode ser mais simples especificar explicitamente o uso do `/bin/kill`.

Ao enviar outros sinais, substitua `TERM` ou `KILL` pelo nome do sinal.



Matar um processo aleatório no sistema é uma má ideia. Em particular, o init(8), PID 1, é especial. Executar `/bin/kill -s KILL 1` é uma maneira rápida e não recomendada de desligar o sistema. *Sempre* verifique os argumentos do kill(1)*antes* de pressionar a tecla `Enter`.

3.9. Shells

Um *shell* fornece uma interface de linha de comandos para interagir com o sistema operacional. Um shell recebe comandos do canal de entrada e os executa. Muitos shells fornecem funções incorporadas para ajudar nas tarefas diárias, como gerenciamento de arquivos, referenciamento de arquivos, edição de linha de comando, macros de comando e variáveis de ambiente. O FreeBSD vem com vários shells, incluindo o shell Bourne (sh(1)) e o shell C estendido (tcsh(1)). Outros shells estão disponíveis na Coleção de Ports do FreeBSD, como o `zsh` e o `bash`.

O shell usado é realmente uma questão de gosto. Um programador C pode se sentir mais confortável com um shell semelhante ao C, como o tcsh(1). Um usuário Linux™ pode preferir o `bash`. Cada shell tem propriedades únicas que podem ou não funcionar com o ambiente de trabalho preferido de um usuário, e é por isso que existe a opção de qual shell usar.

Um recurso de shell comum é a conclusão do nome do arquivo. Depois que um usuário digita as primeiras letras de um comando ou nome de arquivo e pressiona a tecla `Tab`, o shell completa o restante do comando ou nome do arquivo. Considere dois arquivos chamados foobar e football. Para excluir foobar, o usuário pode digitar `rm foo` e pressionar a tecla `Tab` para completar o nome do arquivo.

Mas se o shell mostrar apenas `rm foo`. Não foi possível completar o nome do arquivo porque ambos foobar e football começam com `foo`. Algumas shells emitem um sinal sonoro ou mostram todas as opções se houver mais de um nome. O usuário deve digitar mais caracteres para identificar o nome do arquivo desejado. Digitar um `t` e pressionar a tecla `Tab` novamente é suficiente para permitir que o shell determine qual nome de arquivo é desejado e preencha o resto.

Outra característica do shell é o uso de variáveis de ambiente. As variáveis de ambiente são um par de variável/chave armazenado no ambiente do shell. Esse ambiente pode ser lido por qualquer programa chamado pela shell e, portanto, contém muitas configurações de programas. Variáveis de Ambiente Comuns fornece uma lista de variáveis de ambiente comuns e seus significados. Observe que os nomes das variáveis de ambiente estão sempre em maiúsculas.

Tabela 5. Variáveis de Ambiente Comuns

Variável	Descrição
`USER`	Nome do usuário atual.
`PATH`	Lista de diretórios separados por dois pontos para pesquisa de binários (progamas).
`DISPLAY`	Nome de rede do display do Xorg para conexão, se disponível.
`SHELL`	O shell atual.
`TERM`	O nome do tipo de terminal do usuário. Usado para determinar os recursos do terminal.
`TERMCAP`	Acesso à base de dados dos códigos de escape do terminal para executar várias funções do terminal.
`OSTYPE`	Tipo de sistema operacional.
`MACHTYPE`	A arquitetura da CPU do sistema.
`EDITOR`	O editor de texto preferencial do usuário.
`PAGER`	O utilitário preferencial do usuário para visualização de texto página à página.
`MANPATH`	Lista de diretórios separados por dois pontos para pesquisar páginas de manual.

O processo para definir uma variável de ambiente difere entre as shells. Em tcsh(1) e csh(1), use `setenv` para definir variáveis de ambiente. Em sh(1) e no `bash`, use `export` para definir as variáveis de ambiente atuais. Este exemplo define o `EDITOR` padrão para /usr/local/bin/emacs para a shell tcsh(1):

```
% setenv EDITOR /usr/local/bin/emacs
```

O comando equivalente para `bash` seria:

```
% export EDITOR="/usr/local/bin/emacs"
```

Para expandir uma variável de ambiente para ver sua configuração atual, digite um caracter `$` na frente de seu nome na linha de comando. Por exemplo, `echo $TERM` exibe a configuração atual do `$TERM`.

Shells tratam caracteres especiais, conhecidos como meta-caracteres, como representações especiais de dados. O meta-caracter mais comum é `*`, que representa qualquer número de caracteres em um nome de arquivo. Meta-caracteres podem ser usados para executar a globalização de nomes de arquivos. Por exemplo, `echo *` é equivalente a `ls` porque a shell pega todos os arquivos que correspondem ao `*` e `echo` os lista na linha de comando.

Para evitar que a shell interprete um caractere especial, escape-o a partir da shell, iniciando-o com uma barra invertida (`\`). Por exemplo, `echo $TERM` imprime a configuração do terminal, enquanto `echo \$TERM` imprime literalmente a string `$TERM`.

3.9.1. Alterando a Shell

A maneira mais fácil de alterar permanentemente a shell padrão é usar o `chsh`. A execução desse comando abrirá o editor que está configurado na variável de ambiente `EDITOR`, que por padrão é definido como o vi(1). Altere a linha `Shell:` para o caminho completo da nova shell.

Como alternativa, use `chsh -s`, que irá definir a shell especificada sem abrir um editor. Por exemplo, para alterar a shell para `bash`:

```
% chsh -s /usr/local/bin/bash
```



A nova shell *deve* estar presente no arquivo /etc/shells. Se a shell foi instalada a partir da coleção de ports do FreeBSD, como descrito em Instalando Aplicativos. Pacotes e Ports, ela deve ser adicionada automaticamente a este arquivo. Se estiver faltando, adicione-a usando este comando, substituindo o caminho pelo caminho da shell:

```
# echo /usr/local/bin/bash >> /etc/shells
```

Em seguida, execute novamente o chsh(1).

3.9.2. Técnicas Avançadas de Shell

A shell UNIX™ não é apenas um interpretador de comandos, ela atua como uma ferramenta poderosa que permite aos usuários executar comandos, redirecionar sua saída, redirecionar sua entrada e encadear comandos para melhorar o resultado final. Quando essa funcionalidade é mesclada com comandos incorporados, é fornecido ao usuário um ambiente que pode maximizar a eficiência.

O redirecionamento de shell é a ação de enviar a saída ou a entrada de um comando para outro comando ou para um arquivo. Para capturar a saída do comando ls(1), por exemplo, em um arquivo, redirecione a saída:

```
% ls > directory_listing.txt
```

O conteúdo do diretório agora será listado em directory_listing.txt. Alguns comandos podem ser usados para ler entradas, como sort(1). Para classificar esta listagem, redirecione a entrada:

```
% sort < directory_listing.txt
```

A entrada será classificada e colocada na tela. Para redirecionar essa entrada para outro arquivo, pode-se redirecionar a saída de sort(1) misturando a direção:

```
% sort < directory_listing.txt > sorted.txt
```

Em todos os exemplos anteriores, os comandos estão executando o redirecionamento usando descritores de arquivos. Todo sistema UNIX™ possui descritores de arquivos, que incluem entrada padrão (stdin), saída padrão (stdout) e erro padrão (stderr). Cada um tem um propósito, onde a entrada pode ser um teclado ou um mouse, algo que fornece entrada. A saída pode ser uma tela ou papel em uma impressora. E erro seria tudo o que pode ser usado para mensagens de diagnóstico ou erro. Todos os três são considerados descritores de arquivos baseados em I/O e, às vezes, considerados fluxos.

Através do uso desses descritores, a shell permite que a saída e a entrada sejam passadas por vários comandos e redirecionadas para/ou a partir de um arquivo. Outro método de redirecionamento é o operador de pipe.

O operador pipe UNIX™, "|" permite que a saída de um comando seja transmitida diretamente ou direcionada para outro programa. Basicamente, um pipe permite que a saída padrão de um comando seja passada como entrada padrão para outro comando, por exemplo:

```
% cat directory_listing.txt | sort | less
```

Nesse exemplo, o conteúdo de directory_listing.txt será classificado e a saída será transmitida para less(1). Isso permite que o usuário role pela saída em seu próprio ritmo e evite que ela role para fora da tela.

3.10. Editores de Texto

A maioria das configurações do FreeBSD é feita através da edição de arquivos de texto. Por isso, é uma boa ideia familiarizar-se com um editor de texto. O FreeBSD vem com alguns como parte do sistema base, e muitos outros estão disponíveis na coleção do ports.

Um editor simples para aprender é o ee(1), que significa editor fácil (Ease Editor). Para iniciar este

editor, digite `ee filename` em que *filename* é o nome do arquivo a ser editado. Uma vez dentro do editor, todos os comandos para manipular as funções do editor são listados no topo da tela. O cursor (^) representa `Ctrl`, então `^e` expande para `Ctrl` + `e`. Para sair do ee(1), pressione `Esc` e escolha a opção "leave editor" no menu principal. O editor pedirá para salvar as alterações, caso o arquivo tenha sido modificado.

O FreeBSD também vem com editores de texto mais poderosos, como o vi(1), como parte do sistema base. Outros editores, como editors/emacs e editors/vim, fazem parte da coleção do ports do FreeBSD. Esses editores oferecem mais funcionalidade às custas de serem mais complicados de aprender. Aprender um editor mais poderoso como o vim ou o Emacs pode economizar mais tempo a longo prazo.

Muitos aplicativos que modificam arquivos ou exigem entrada digitada abrirão automaticamente um editor de texto. Para alterar o editor padrão, defina a variável de ambiente `EDITOR` conforme descrito em Shells.

3.11. Dispositivos e nós de dispositivos

Um dispositivo é um termo usado principalmente para atividades relacionadas a hardware em um sistema, incluindo discos, impressoras, placas gráficas e teclados. Quando o FreeBSD inicializa, a maioria das mensagens de inicialização se refere aos dispositivos sendo detectados. Uma cópia das mensagens de inicialização é salva em /var/run/dmesg.boot.

Cada dispositivo tem um nome e um número de dispositivo. Por exemplo, ada0 é o primeiro disco rígido SATA, enquanto kbd0 representa o teclado.

A maioria dos dispositivos no FreeBSD deve ser acessada através de arquivos especiais chamados nós de dispositivos (device nodes), que estão localizados em /dev.

3.12. Páginas de Manual

A documentação mais abrangente sobre o FreeBSD está na forma de páginas de manual. Quase todos os programas do sistema vêm com um breve manual de referência explicando a operação básica e os argumentos disponíveis. Estes manuais podem ser visualizados usando o `man`:

```
% man command
```

onde *command* é o nome do comando para aprender. Por exemplo, para saber mais sobre o ls(1), digite:

```
% man ls
```

As páginas de manual são divididas em seções que representam o tipo de tópico. No FreeBSD, as seguintes seções estão disponíveis:

1. Comandos de usuário.

2. Chamadas do sistema e números de erro.
3. Funções nas bibliotecas C.
4. Drivers de dispositivos.
5. Formatos de arquivo.
6. Jogos e outras diversões.
7. Informações diversas.
8. Comandos de manutenção e operação do sistema.
9. Interfaces do kernel do sistema.

Em alguns casos, o mesmo tópico pode aparecer em mais de uma seção do manual online. Por exemplo, existe um comando de usuário `chmod` e uma chamada de sistema `chmod()`. Para informar ao man(1) qual seção exibir, especifique o número da seção:

```
% man 1 chmod
```

Isto irá mostrar a página de manual do comando chmod(1). Referências a uma seção em particular do manual online são tradicionalmente colocadas entre parênteses na documentação escrita, então chmod(1) refere-se ao comando do usuário e chmod(2) refere-se à chamada do sistema.

Se o nome da página de manual for desconhecido, use `man -k` para procurar por palavras-chave nas descrições da página de manual:

```
% man -k mail
```

Este comando exibe uma lista de comandos que possuem a palavra-chave "mail" em suas descrições. Isso é equivalente a usar o apropos(1).

Para ler as descrições de todos os comandos em /usr/bin, digite:

```
% cd /usr/bin
% man -f * | more
```

ou

```
% cd /usr/bin
% whatis * |more
```

3.12.1. Arquivos GNU Info

O FreeBSD inclui vários aplicativos e utilitários produzidos pela Free Software Foundation (FSF). Além das páginas de manual, esses programas podem incluir documentos de hipertexto chamados arquivos `info`. Elas podem ser visualizadas usando info(1) ou, se o editors/emacs estiver instalado, o

modo info do emacs.

Para usar o info(1), digite:

```
% info
```

Para uma breve introdução, digite h. Para uma referência rápida de comandos, digite ?.

Capítulo 4. Instalando Aplicativos: Pacotes e Ports

4.1. Sinopse

O FreeBSD tem uma grande coleção de ferramentas dentro do sistema base. Além disso, o FreeBSD fornece duas ferramentas complementares para a instalação de software de terceiros: o a Coleção de Ports do FreeBSD, para instalação a partir do código-fonte, e pacotes, para instalação de binários pré-compilados. Qualquer um dos métodos pode ser usado para instalar um software de uma mídia local ou da rede.

Depois de ler este capítulo, você saberá:

- A diferença entre pacotes binários e ports.
- Como encontrar softwares de terceiros que tenham sido portados para o FreeBSD.
- Como gerenciar pacotes binários usando o pkg.
- Como compilar software de terceiros a partir do código-fonte usando a coleção de ports.
- Como encontrar os arquivos instalados do aplicativo para configuração pós-instalação.
- O que fazer se a instalação do software falhar.

4.2. Visão geral sobre a Instalação de Software

As etapas típicas para instalar um software de terceiros em um sistema UNIX™ incluem:

1. Encontre e baixe o software, que pode ser distribuído no formato de código-fonte ou como um binário.
2. Desempacote o software a partir do seu formato de distribuição. Tipicamente é um arquivo tarball compactado com um programa como compress(1),gzip(1), bzip2(1) ou xz(1).
3. Localize a documentação em INSTALL, README ou algum arquivo em um subdiretório doc/ e leia sobre como instalar o software.
4. Se o software foi distribuído como código-fonte, compile-o. Isso pode envolver a edição de um Makefile ou a execução de um script `configure`.
5. Teste e instale o software.

Um *port* do FreeBSD é uma coleção de arquivos projetados para automatizar o processo de compilação de um aplicativo a partir do código-fonte. Os arquivos que compõem um port contêm todas as informações necessárias para baixar, extrair, corrigir, compilar e instalar automaticamente o aplicativo.

Se o software ainda não foi adaptado e testado no FreeBSD, o código-fonte pode precisar ser editado para que seja instalado e executado corretamente.

No entanto, mais de 24.000 aplicativos de terceiros já foram portados para o FreeBSD. Quando possível, esses aplicativos são disponibilizados para download como *pacotes* pré-compilados.

Pacotes podem ser manipulados com os comandos de gerenciamento de pacotes do FreeBSD.

Ambos, pacotes e ports, entendem dependências. Se um pacote ou port for usado para instalar um aplicativo, e uma biblioteca dependente ainda não estiver instalada, a biblioteca será instalada automaticamente primeiro.

Um pacote do FreeBSD contém cópias pré-compiladas de todos os comandos para uma aplicação, assim como quaisquer arquivos de configuração e documentação. Um pacote pode ser manipulado com os comandos pkg(8), como `pkg install`.

Mesmo as duas tecnologias sendo semelhantes, os pacotes e os ports têm seus próprios pontos fortes. Selecione a tecnologia que melhor atenda aos seus requisitos para instalar um aplicativo específico.

Benefícios dos Pacotes

- Um tarball compactado de um pacote geralmente é menor que o tarball compactado que contém o código-fonte do aplicativo.
- Pacotes não requerem tempo de compilação. Para aplicativos grandes, como o Mozilla, KDE ou GNOME, isso pode ser importante em um sistema lento.
- Pacotes não requerem nenhum entendimento do processo envolvido na compilação de software no FreeBSD.

Benefícios dos Ports

- Os pacotes são normalmente compilados com opções conservadoras porque eles precisam ser executados no número máximo de sistemas. Ao compilar a partir do port, podem-se alterar as opções de compilação.
- Alguns aplicativos têm opções em tempo de compilação relacionadas a quais recursos estão instalados. Por exemplo, o Apache pode ser configurado com uma ampla variedade de diferentes opções internas.

  Em alguns casos, vários pacotes existirão para o mesmo aplicativo para especificar determinadas configurações. Por exemplo, o Ghostscript está disponível como um pacote ghostscript e um pacote ghostscript-nox11, dependendo se o Xorg está instalado ou não. Criar vários pacotes rapidamente se torna impossível se um aplicativo tiver mais de uma ou duas opções diferentes de tempo de compilação.

- As condições de licenciamento de alguns softwares proíbem sua distribuição em binário. Tais softwares devem ser distribuídos como código-fonte o qual deve ser compilado pelo usuário final.
- Algumas pessoas não confiam em distribuições binárias ou preferem ler o código-fonte para procurar possíveis problemas.
- O código-fonte é necessário para aplicar patches personalizados.

Para acompanhar a atualização dos ports, inscreva-se na lista de discussão dos ports do FreeBSD e

no link Lista de discussão de bugs no FreeBSD.



Antes de instalar qualquer aplicativo, verifique https://vuxml.freebsd.org/ para questões de segurança relacionadas ao aplicativo ou digite `pkg audit -F` para verificar todas as instâncias instaladas aplicativos para vulnerabilidades conhecidas.

O restante deste capítulo explica como usar pacotes e ports para instalar e gerenciar software de terceiros no FreeBSD.

4.3. Encontrando Software

A lista de aplicativos disponíveis do FreeBSD está crescendo o tempo todo. Existem várias maneiras de encontrar softwares para instalar:

- O site do FreeBSD mantém uma lista atualizada e pesquisável de todos os aplicativos disponíveis, em https://www.FreeBSD.org/ports/. Os ports podem ser pesquisados por nome do aplicativo ou por categoria de software.
- Dan Langille mantém o FreshPorts.org, que fornece um utilitário de pesquisa abrangente e também rastreia alterações nos aplicativos da Coleção de Ports. Os usuários registrados podem criar uma lista de observação personalizada para receber um e-mail automatizado quando seus ports sendo monitorados forem atualizados.
- Se encontrar um aplicativo específico se tornar desafiador, tente pesquisar um site como SourceForge.net ou GitHub.com então volte no site do FreeBSD para ver se o aplicativo foi portado.
- Para pesquisar o repositório de pacotes binários por um aplicativo:

```
# pkg search subversion
git-subversion-1.9.2
java-subversion-1.8.8_2
p5-subversion-1.8.8_2
py27-hgsubversion-1.6
py27-subversion-1.8.8_2
ruby-subversion-1.8.8_2
subversion-1.8.8_2
subversion-book-4515
subversion-static-1.8.8_2
subversion16-1.6.23_4
subversion17-1.7.16_2
```

Os nomes dos pacotes incluem o número da versão e, no caso de ports baseados em python, o número da versão do pacote python sobre o qual o pacote foi compilado. Alguns ports também possuem várias versões disponíveis. No caso do Subversion, existem diferentes versões disponíveis, bem como diferentes opções de compilação. Neste caso, a versão estaticamente vinculada do Subversion. Ao indicar qual pacote instalar, é melhor especificar o aplicativo pela origem do port, que é o caminho na árvore de ports. Repita o `pkg search` com `-o` para listar a

origem de cada pacote:

```
# pkg search -o subversion
devel/git-subversion
java/java-subversion
devel/p5-subversion
devel/py-hgsubversion
devel/py-subversion
devel/ruby-subversion
devel/subversion16
devel/subversion17
devel/subversion
devel/subversion-book
devel/subversion-static
```

Pesquisar por shell globs, expressões regulares, correspondência exata, por descrição ou qualquer outro campo no banco de dados do repositório também é suportado pelo `pkg search`. Depois de instalar o ports-mgmt/pkg ou o ports-mgmt/pkg-devel, veja pkg-search(8) para maiores detalhes.

- Se a Coleção de Ports já estiver instalada, existem vários métodos para consultar a versão local da árvore de ports. Para descobrir em qual categoria um port está, digite `whereisfile`, onde *file* é o programa a ser instalado:

```
# whereis lsof
lsof: /usr/ports/sysutils/lsof
```

Como alternativa, uma declaração echo(1) pode ser usada:

```
# echo /usr/ports/*/*lsof*
/usr/ports/sysutils/lsof
```

Observe que isso também retornará todos os arquivos correspondentes baixados no diretório /usr/ports/distfiles.

- Outra maneira de encontrar software é usando o mecanismo de pesquisa integrado da Coleção de Ports. Para usar o recurso de pesquisa, cd para /usr/ports, execute `make search name=program-name` onde *program-name* é o nome do software. Por exemplo, para procurar por `lsof`:

```
# cd /usr/ports
# make search name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
Maint:  ler@lerctr.org
Index:  sysutils
```

```
B-deps:
R-deps:
```



O mecanismo de pesquisa interna usa um arquivo de informações de índice. Se uma mensagem indicar que o INDEX é necessário, execute `make fetchindex` para baixar o arquivo de índice atual. Com o INDEX presente, o `make search` poderá realizar a pesquisa solicitada.

A linha "Path:" indica onde encontrar o port.

Para receber menos informações, use o recurso `quicksearch`:

```
# cd /usr/ports
# make quicksearch name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
```

Para uma busca mais aprofundada, use o `make search key=string` ou o `make quicksearch key=string`, onde *string* é algum texto para procurar. O texto pode estar em comentários, descrições ou dependências para encontrar ports relacionados a um assunto em particular quando o nome do programa é desconhecido.

Ao usar `pesquisa` ou `pesquisa rápida`, a cadeia de pesquisa não diferencia maiúsculas de minúsculas. Procurar por "LSOF" produzirá os mesmos resultados que procurar por "lsof".

4.4. Usando o pkg para o gerenciamento de pacotes binários

O pkg é o substituto da próxima geração para as tradicionais ferramentas de gerenciamento de pacotes do FreeBSD, oferecendo muitos recursos que tornam o processamento de pacotes binários mais rápido e fácil.

Para sites que desejam apenas usar pacotes binários pré-construídos a partir dos espelhos do FreeBSD, o gerenciamento de pacotes com pkg pode ser suficiente.

No entanto, para aqueles que optarem por compilar suas aplicações a partir do código-fonte ou que utilizarem seus próprios repositórios, será necessária uma ferramenta de gerenciamento de ports separada.

Como o pkg só funciona com pacotes binários, ele não é um substituto para tais ferramentas. Estas ferramentas podem ser usadas para instalar o software a partir de pacotes binários e da Coleção do Ports, enquanto o pkg instala apenas pacotes binários.

4.4.1. Introdução ao pkg

O FreeBSD inclui um utilitário de bootstrap que pode ser usado para baixar e instalar o pkg e suas

páginas de manual. Este utilitário foi projetado para funcionar com versões do FreeBSD começando com 10.*X*.



Nem todas as versões e arquiteturas do FreeBSD suportam este processo de bootstrap. A lista atual está em https://pkg.freebsd.org/. Para outros casos, o pkg deve ser instalado a partir da coleção de ports ou como um pacote binário.

Para inicializar o sistema, execute:

```
# /usr/sbin/pkg
```

Você deve ter uma conexão com a Internet para que o processo de inicialização seja bem-sucedido.

Caso contrário, para instalar o port, execute:

```
# cd /usr/ports/ports-mgmt/pkg
# make
# make install clean
```

Ao atualizar um sistema existente que usava originalmente as ferramentas pkg_* mais antigas, o banco de dados deve ser convertido para o novo formato, para que as novas ferramentas estejam cientes dos pacotes já instalados. Uma vez que o pkg tenha sido instalado, o banco de dados de pacotes deve ser convertido do formato tradicional para o novo formato, executando este comando:

```
# pkg2ng
```



Esta etapa não é necessária para novas instalações que ainda não possuem nenhum software de terceiros instalado.



Este passo não é reversível. Uma vez que o banco de dados de pacotes tenha sido convertido para o formato pkg, as ferramentas tradicionais `pkg_*` não devem mais ser usadas.



A conversão do banco de dados de pacotes pode emitir erros conforme o conteúdo é convertido para a nova versão. Geralmente, esses erros podem ser ignorados com segurança. No entanto, uma lista com os softwares que não foram convertidos com sucesso é mostrada após o `pkg2ng` terminar. Esses aplicativos devem ser reinstalados manualmente.

Para garantir que a Coleção de Ports registre novos softwares com o pkg ao invés do tradicional banco de dados de pacotes, versões do FreeBSD anteriores a 10.*X* requerem esta linha em /etc/make.conf:

```
WITH_PKGNG= yes
```

Por padrão, o pkg usa os pacotes binários dos espelhos de pacotes do FreeBSD (o *repositório*). Para obter informações sobre como criar um repositório de pacotes personalizados, consulte Compilando Pacotes com o Poudriere.

Opções adicionais de configuração do pkg são descritas em pkg.conf(5).

As informações de uso do pkg estão disponíveis na página de manual pkg(8) ou executando o `pkg` sem argumentos adicionais.

Cada argumento do comando pkg é documentado em uma página de manual específica do comando. Para ler a página de manual do `pkg install`, por exemplo, execute um destes comandos:

```
# pkg help install
```

```
# man pkg-install
```

O restante desta seção demonstra tarefas comuns de gerenciamento de pacotes binários que podem ser executadas usando o pkg. Cada comando demonstrado fornece muitos switches para personalizar seu uso. Consulte a ajuda de um comando ou a página do manual para obter detalhes e mais exemplos.

4.4.2. Branches Ports Trimestrais e Mais Recentes

As branches `Quarterly`(trimestrais) provê aos usuários uma experiência mais estável e previsível para instalação e upgrade de ports e pacotes. Isto é feito essencialmente permitindo apenas atualizações que não contém novas features (non-features updates). Branches trimestrais visam receber correções se segurança (que talvez sejam atualizações de versão, ou commits de backports), correções de bugs e compliance de ports ou alterações de frameworks. A branch trimestral é baseada (anualmente) na HEAD no início de Janeiro, Abril, Julho e Outubro. As branches são nomeadas de acordo com o ano (YYYY) e o quarter (Q1-4) em que são criadas. Por exemplo, a branch trimestral criada em Janeiro de 2016, é nomeada 2016Q1. E a branch `Latest` provê as últimas versões dos pacotes para os usuários.

Para alternar de trimestral para latest execute os seguintes comandos:

```
# cp /etc/pkg/FreeBSD.conf /usr/local/etc/pkg/repos/FreeBSD.conf
```

Edite o arquivo /usr/local/etc/pkg/repos/FreeBSD.conf and change the string *quarterly* to *latest* in the `url:` line.

O resultado deve ser semelhante ao seguinte:

```
FreeBSD: {
  url: "pkg+http://pkg.FreeBSD.org/${ABI}/latest",
  mirror_type: "srv",
  signature_type: "fingerprints",
```

```
  fingerprints: "/usr/share/keys/pkg",
  enabled: yes
}
```

E finalmente rode este comando para atualizar do novo (ultimo) meta dado do repositório.

```
# pkg update -f
```

4.4.3. Obtendo informações sobre os pacotes instalados

Informações sobre os pacotes instalados em um sistema podem ser visualizadas executando `pkg info` que, quando executado sem qualquer opção, listará a versão do pacote para todos os pacotes instalados ou para o pacote especificado.

Por exemplo, para ver qual versão do pkg está instalada, execute:

```
# pkg info pkg
pkg-1.1.4_1
```

4.4.4. Instalando e removendo pacotes

Para instalar um pacote binário, use o seguinte comando, em que *packagename* é o nome do pacote a ser instalado:

```
# pkg install packagename
```

Esse comando usa os dados do repositório para determinar qual versão do software instalar e se ele possui alguma dependência faltando. Por exemplo, para instalar o curl:

```
# pkg install curl
Updating repository catalogue
/usr/local/tmp/All/curl-7.31.0_1.txz          100% of 1181 kB 1380 kBps 00m01s

/usr/local/tmp/All/ca_root_nss-3.15.1_1.txz   100% of  288 kB 1700 kBps 00m00s

Updating repository catalogue
The following 2 packages will be installed:

        Installing ca_root_nss: 3.15.1_1
        Installing curl: 7.31.0_1

The installation will require 3 MB more space

0 B to be downloaded

Proceed with installing packages [y/N]: y
```

```
Checking integrity... done
[1/2] Installing ca_root_nss-3.15.1_1... done
[2/2] Installing curl-7.31.0_1... done
Cleaning up cache files...Done
```

O novo pacote e quaisquer pacotes adicionais que foram instalados como dependências podem ser vistos na lista de pacotes instalados:

```
# pkg info
ca_root_nss-3.15.1_1    The root certificate bundle from the Mozilla Project
curl-7.31.0_1   Non-interactive tool to get files from FTP, GOPHER, HTTP(S) servers
pkg-1.1.4_6 New generation package manager
```

Pacotes que não são mais necessários podem ser removidos com `pkg delete`. Por exemplo:

```
# pkg delete curl
The following packages will be deleted:

    curl-7.31.0_1

The deletion will free 3 MB

Proceed with deleting packages [y/N]: y
[1/1] Deleting curl-7.31.0_1... done
```

4.4.5. Atualizando os Pacotes Instalados

Os pacotes instalados podem ser atualizados para as versões mais recentes executando:

```
# pkg upgrade
```

Este comando irá comparar as versões instaladas com as disponíveis no catálogo do repositório e atualizá-las a partir do repositório.

4.4.6. Auditando os Pacotes Instalados

Vulnerabilidades de software são regularmente descobertas em aplicativos de terceiros. Para resolver isso, o pkg inclui um mecanismo de auditoria integrado. Para determinar se há alguma vulnerabilidade conhecida para o software instalado no sistema, execute:

```
# pkg audit -F
```

4.4.7. Removendo Pacotes Não Utilizados Automaticamente

Remover um pacote pode deixar dependências que não são mais necessárias. Pacotes desnecessários que foram instalados como dependências podem ser automaticamente detectados e removidos usando:

```
# pkg autoremove
Packages to be autoremoved:
    ca_root_nss-3.15.1_1

The autoremoval will free 723 kB

Proceed with autoremoval of packages [y/N]: y
Deinstalling ca_root_nss-3.15.1_1... done
```

Os pacotes instalados como dependências são chamados de pacotes *automáticos*. Pacotes não automáticos, ou seja, os pacotes que não foram instalados como uma dependência para outro pacote, podem ser listados usando:

```
# pkg prime-list
nginx
openvpn
sudo
```

O `pkg prime-list` é um alias de comando declarado no /usr/local/etc/pkg.conf. Existem muitos outros que podem ser usados para consultar o banco de dados de pacotes do sistema. Por exemplo, o comando `pkg prime-origins` pode ser usado para obter o diretório de origem dos ports da lista mencionada acima:

```
# pkg prime-origins
www/nginx
security/openvpn
security/sudo
```

Esta lista pode ser usada para recompilar todos os pacotes instalados em um sistema usando ferramentas de compilação como o ports-mgmt/poudriere ou o ports-mgmt/synth.

Marcar um pacote instalado como automático pode ser feito usando:

```
# pkg set -A 1 devel/cmake
```

Uma vez que um pacote é um pacote orfão e está marcado como automático, ele será selecionado por `pkg autoremove`.

Marcar um pacote instalado como *não* automático pode ser feito usando:

```
# pkg set -A 0 devel/cmake
```

4.4.8. Restaurando o banco de dados de pacotes

Ao contrário do sistema tradicional de gerenciamento de pacotes, o pkg inclui seu próprio mecanismo de backup de banco de dados de pacotes. Essa funcionalidade é habilitada por padrão.



Para desabilitar o script que faz o backup periódico do banco de dados de pacotes, defina `daily_backup_pkgdb_enable="NO"` no periodic.conf(5).

Para restaurar o conteúdo de um backup anterior do banco de dados de pacotes, execute o seguinte comando substituindo */path/to/pkg.sql* pelo local do backup:

```
# pkg backup -r /path/to/pkg.sql
```



Se estiver restaurando um backup feito pelo script periódico, ele deve ser descompactado antes de ser restaurado.

Para executar um backup manual do banco de dados pkg, execute o seguinte comando, substituindo */path/to/pkg.sql* por um nome de arquivo e local adequados:

```
# pkg backup -d /path/to/pkg.sql
```

4.4.9. Removendo Pacotes Obsoletos

Por padrão, o pkg armazena pacotes binários em um diretório de cache definido por `PKG_CACHEDIR` no pkg.conf(5). Somente cópias dos últimos pacotes instalados são mantidas. Versões mais antigas do pkg mantinham todos os pacotes anteriores. Para remover esses pacotes binários desatualizados, execute:

```
# pkg clean
```

O cache inteiro pode ser limpo executando:

```
# pkg clean -a
```

4.4.10. Modificando Metadados de Pacotes

Os softwares dentro da Coleção de Ports do FreeBSD podem passar por grandes mudanças no número de versão. Para resolver isso, o pkg possui um comando interno para atualizar as origens do pacote. Isto pode ser útil, por exemplo, se lang/php5 for renomeado para lang/php53 para que lang/php5 possa agora representar a versão `5.4`.

Para alterar a origem do pacote para o exemplo acima, execute:

```
# pkg set -o lang/php5:lang/php53
```

Como outro exemplo, para atualizar lang/ruby18 para lang/ruby19, execute:

```
# pkg set -o lang/ruby18:lang/ruby19
```

Como um exemplo final, para alterar a origem das bibliotecas compartilhadas libglut de graphics/libglut para graphics/freeglut, execute:

```
# pkg install -Rf graphics/freeglut
```



Ao alterar as origens do pacote, é importante reinstalar os pacotes que dependem do pacote com a origem modificada. Para forçar uma reinstalação dos pacotes dependentes, execute:

```
# pkg install -Rf graphics/freeglut
```

4.5. Usando a Coleção de Ports

A Coleção de Ports é um conjunto de arquivos Makefiless, patches e arquivos de descrição. Cada conjunto desses arquivos é usado para compilar e instalar um aplicativo individual no FreeBSD, e é chamado de *port*.

Por padrão, a própria coleção de ports é armazenada como um subdiretório de /usr/ports.



Before installing and using the Ports Collection, please be aware that it is generally ill-advised to use the Ports Collection in conjunction with the binary packages provided via pkg to install software. pkg, by default, tracks quarterly branch-releases of the ports tree and not HEAD. Dependencies could be different for a port in HEAD compared to its counterpart in a quarterly branch release and this could result in conflicts between dependencies installed by pkg and those from the Ports Collection. If the Ports Collection and pkg must be used in conjunction, then be sure that your Ports Collection and pkg are on the same branch release of the ports tree.

Antes que um aplicativo possa ser compilado usando um port, a Coleção de Ports deve primeiro ser instalada. Se ela não foi instalada durante a instalação do FreeBSD, use um dos seguintes métodos para instalá-la:

Procedure: Método Portsnap

O sistema base do FreeBSD inclui o Portsnap. Esta é uma ferramenta rápida e de fácil utilização para obter a Coleção de Ports e é a escolha recomendada para a maioria dos usuários que não estão executando o FreeBSD-CURRENT. Este utilitário se conecta a um site do FreeBSD, verifica a chave segura e faz o download de uma nova cópia da Coleção de Ports. A chave é usada para verificar a integridade de todos os arquivos baixados.

1. Para baixar um snapshot compactado da coleção de ports em /var/db/portsnap:

```
# portsnap fetch
```

2. Ao executar o Portsnap pela primeira vez, extraia o snapshot em /usr/ports:

```
# portsnap extract
```

3. Após o primeiro uso do Portsnap ter sido concluído, como mostrado acima, o /usr/ports pode ser atualizado conforme necessário executando:

```
# portsnap fetch
# portsnap update
```

   Ao usar `fetch`, a operação `extract` ou `update` pode ser executada consecutivamente, da seguinte forma:

```
# portsnap fetch update
```

Procedure: Método Subversion

Se for necessário mais controle sobre a árvore de ports ou se as mudanças locais precisarem ser mantidas, ou se estiver executando o FreeBSD-CURRENT, o Subversion pode ser usado para obter a coleção de ports. Consulte O Subversion Primer para uma descrição detalhada do Subversion.

1. O Subversion deve ser instalado antes de poder ser usado para fazer o check-out da árvore de ports. Se uma cópia da árvore de ports já estiver presente, instale o Subversion desta forma:

```
# cd /usr/ports/devel/subversion
# make install clean
```

   Se a árvore de ports não estiver disponível, ou o pkg estiver sendo usado para gerenciar pacotes, o Subversion poderá ser instalado como um pacote:

```
# pkg install subversion
```

2. Check out a copy of the HEAD branch of the ports tree:

```
# svn checkout https://svn.FreeBSD.org/ports/head /usr/ports
```

3. Or, check out a copy of a quarterly branch:

```
# svn checkout https://svn.FreeBSD.org/ports/branches/2020Q3 /usr/ports
```

4. Conforme necessário, atualize o /usr/ports após o check out inicial do Subversion:

```
# svn update /usr/ports
```

5. As needed, switch /usr/ports to a different quarterly branch:

```
# svn switch http://svn.freebsd.org/ports/branches/2020Q4/ /usr/ports
```

A coleção de ports contém diretórios para categorias de software. Dentro de cada categoria estão subdiretórios para aplicativos individuais. Cada subdiretório de aplicativo contém um conjunto de arquivos que informa ao FreeBSD como compilar e instalar esse programa, chamado de *esqueleto do ports*. Cada esqueleto de port inclui esses arquivos e diretórios:

- Makefile: contém instruções que especificam como o aplicativo deve ser compilado e onde seus componentes devem ser instalados.
- distinfo: contém os nomes e checksums dos arquivos que devem ser baixados para compilar o port.
- files/: este diretório contém quaisquer patches necessários para o programa compilar e instalar no FreeBSD. Esse diretório também pode conter outros arquivos usados para compilar o port.
- pkg-descr: fornece uma descrição mais detalhada do programa.
- pkg-plist: uma lista de todos os arquivos que serão instalados pelo port. Ele também informa ao sistema de ports quais arquivos devem ser removidos após a desinstalação.

Alguns ports incluem pkg-message ou outros arquivos para lidar com situações especiais. Para obter mais detalhes sobre esses arquivos e sobre os ports em geral, consulte o FreeBSD Porter's Manual.

O port não inclui o código-fonte real, também conhecido como distfile. A etapa de extração da compilação de um port salvará automaticamente o código-fonte transferido por download para /usr/ports/distfiles.

4.5.1. Instalando Ports

Esta seção fornece instruções básicas sobre o uso da Coleção de Ports para instalar ou remover software. A descrição detalhada dos targets disponíveis do `make` e das variáveis de ambiente está disponível em ports(7).



Antes de compilar qualquer port, certifique-se de atualizar a Coleção de Ports conforme descrito na seção anterior. Como a instalação de qualquer software de terceiros pode introduzir vulnerabilidades de segurança, recomenda-se primeiro verificar https://vuxml.freebsd.org/ para problemas de segurança conhecidos relacionados ao port. Alternativamente, execute `pkg -f` antes de instalar um novo port. Esse comando pode ser configurado para executar automaticamente uma auditoria de segurança e uma atualização do banco de dados de vulnerabilidades durante a verificação diária do sistema de segurança. Para obter maiores informações, consulte pkg-audit(8) e periodic(8).

O uso da coleção de ports pressupõe uma conexão de Internet ativa. Também requer privilégios de superusuário.

Para compilar e instalar o port, mude para o diretório do port a ser instalado e, em seguida, digite `make install` no prompt. Mensagens indicarão o progresso:

```
# cd /usr/ports/sysutils/lsof
# make install
>> lsof_4.88D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.88
...
[extraction output snipped]
...
>> Checksum OK for lsof_4.88D.freebsd.tar.gz.
===>  Patching for lsof-4.88.d,8
===>  Applying FreeBSD patches for lsof-4.88.d,8
===>  Configuring for lsof-4.88.d,8
...
[configure output snipped]
...
===>  Building for lsof-4.88.d,8
...
[compilation output snipped]
...

===>  Installing for lsof-4.88.d,8
...
[installation output snipped]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.88.d,8
===>   Registering installation for lsof-4.88.d,8
```

```
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
/usr/local/sbin/lsof
#
```

Como o `lsof` é um programa que é executado com privilégios aumentados, um aviso de segurança é exibido quando é instalado. Quando a instalação estiver concluída, o prompt será retornado.

Algumas shells mantêm um cache dos comandos que estão disponíveis nos diretórios listados na variável de ambiente `PATH`, para acelerar as operações de pesquisa do arquivo executável desses comandos. Os usuários do shell `tcsh` devem digitar `rehash` para que um comando recém-instalado possa ser usado sem especificar seu caminho completo. Use `hash -r` para o shell `sh`. Consulte a documentação do shell para mais informações.

Durante a instalação, é criado um subdiretório de trabalho que contém todos os arquivos temporários usados durante a compilação. A remoção desse diretório economiza espaço em disco e minimiza a possibilidade de problemas mais tarde ao atualizar para a versão mais recente do port:

```
# make clean
===>  Cleaning for lsof-88.d,8
#
```



Para evitar esta etapa extra, use `make install clean` ao compilar o port.

4.5.1.1. Personalizando a instalação de ports

Alguns ports fornecem opções de compilação que podem ser usadas para habilitar ou desabilitar componentes do aplicativo, fornecer opções de segurança ou permitir outras personalizações. Os exemplos incluem o www/firefox, security/gpgme, e mail/sylpheed-claws. Se o port depender de outros ports que tenham opções configuráveis, ela poderá pausar várias vezes para interação do usuário, pois o comportamento padrão é solicitar ao usuário que selecione opções de um menu. Para evitar isso e fazer toda a configuração em um lote, execute `make config-recursive` dentro do diretório do port. Em seguida, execute `make install [clean]` para compilar e instalar o port.



Ao usar `config-recursive`, a lista de ports a serem configurados é reunida pelo target `all-depends-list`. É recomendado executar o `make config-recursive` até que todas as opções dos ports dependentes tenham sido definidas, e as telas de opções de ports não apareçam mais, para ter certeza de que todas as opções das dependência foram configuradas.

Há várias maneiras de revisitar o menu de opções de compilação de um port para adicionar, remover ou alterar essas opções após a compilação de um port. Um método é efetuar `cd` no diretório que contém o port e digitar `make config`. Outra opção é usar o `make showconfig`. Outra opção é executar `make rmconfig`, o que removerá todas as opções selecionadas e permitirá que você comece de novo. Todas essas opções, e outras, são explicadas detalhadamente em ports(7).

O sistema de ports usa o fetch(1) para fazer o download dos arquivos com o código-fonte, que suportam várias variáveis de ambiente. As variáveis `FTP_PASSIVE_MODE`, `FTP_PROXY` e `FTP_PASSWORD` podem precisar ser definidas se o sistema FreeBSD estiver por trás de um firewall ou proxy FTP/HTTP. Veja fetch(3) para a lista completa de variáveis suportadas.

Para usuários que não podem estar conectados à Internet o tempo todo, o `make fetch` pode ser executado dentro do /usr/ports, para buscar todos os distfiles, ou dentro de uma categoria, como /usr/ports/net, ou dentro do diretório de um port especifico. Observe que, se um port tiver alguma dependência, executar este comando em uma categoria ou diretório de um port especifico *não* buscará os distfiles de ports de outra categoria. Em vez disso, use `make fetch-recursive` para também buscar os distfiles para todas as dependências de um port.

Em casos raros, como quando uma organização tem um repositório local de distfiles, a variável `MASTER_SITES` pode ser usada para substituir os locais de download especificados no Makefile. Ao usar, especifique o local alternativo:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.organization.org/pub/FreeBSD/ports/distfiles/ fetch
```

As variáveis `WRKDIRPREFIX` e `PREFIX` podem substituir os diretórios de trabalho e de destino padrão. Por exemplo:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

irá compilar o port em /usr/home/example/ports e instalar tudo sob /usr/local.

```
# make PREFIX=/usr/home/example/local install
```

irá compilar o port em /usr/ports e instalá-lo em /usr/home/example/local. E:

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

vai combinar os dois.

Estes também podem ser definidos como variáveis ambientais. Consulte a página de manual do seu shell para obter instruções sobre como definir uma variável de ambiente.

4.5.2. Removendo Ports Instalados

Ports instalados podem ser desinstalados usando `pkg delete`. Exemplos para usar este comando podem ser encontrados na página de manual pkg-delete(8).

Alternativamente, o `make deinstall` pode ser executado no diretório do port:

```
# cd /usr/ports/sysutils/lsof
# make deinstall
===>  Deinstalling for sysutils/lsof
===>   Deinstalling
Deinstallation has been requested for the following 1 packages:

    lsof-4.88.d,8

The deinstallation will free 229 kB
[1/1] Deleting lsof-4.88.d,8... done
```

Recomenda-se ler as mensagens quando o port for desinstalado. Se o port tiver algum aplicativo que dependa dele, essas informações serão exibidas, mas a desinstalação continuará. Nesses casos, talvez seja melhor reinstalar o aplicativo para evitar dependências quebradas.

4.5.3. Atualizando os Ports

Com o tempo, novas versões de software ficam disponíveis na coleção de ports. Esta seção descreve como determinar qual software pode ser atualizado e como executar a atualização.

Para determinar se versões mais recentes dos ports instalados estão disponíveis, verifique se a versão mais recente da árvore de ports está instalada, usando o comando de atualização descrito em Método Portsnap ou Método Subversion. No FreeBSD 10 e posterior, ou se o sistema foi convertido para pkg, o seguinte comando listará os ports instalados que estão desatualizadas:

```
# pkg version -l "<"
```

Para o FreeBSD 9.*X* e menor, o seguinte comando listará os ports instalados que estão desatualizados:

```
# pkg_version -l "<"
```



Antes de tentar uma atualização, leia o /usr/ports/UPDATING da parte superior do arquivo até a data mais próxima da última vez em que os ports foram atualizados ou o sistema foi instalado. Este arquivo descreve vários problemas e etapas adicionais que os usuários podem encontrar e precisar executar ao atualizar um port, incluindo coisas como alterações de formato de arquivo, alterações nos locais de arquivos de configuração ou incompatibilidades com versões anteriores. Anote quaisquer instruções que correspondam a qualquer um dos ports que precisam de atualização e siga estas instruções ao executar a atualização.

4.5.3.1. Ferramentas para atualizar e gerenciar ports

A coleção de ports contém vários utilitários para executar a atualização real. Cada um tem seus pontos fortes e fracos.

Historicamente, a maioria das instalações usava o Portmaster ou o Portupgrade. O Synth é uma alternativa mais recente.



A escolha da melhor ferramenta para um determinado sistema depende do administrador do sistema. Recomenda-se a prática de fazer backup de seus dados antes de usar qualquer uma dessas ferramentas.

4.5.3.2. Atualizando Ports Usando o Portmaster

O ports-mgmt/portmaster é um utilitário muito pequeno para atualizar os ports instalados. Ele é projetado para usar as ferramentas instaladas com o sistema base do FreeBSD sem depender de outros ports ou bancos de dados. Para instalar este utilitário como um port:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

O Portmaster define quatro categorias de ports:

- Port Root: não tem dependências e não é uma dependência de outros ports.
- Port Trunk: não tem dependências, mas outros ports dependem dele.
- Port Branch: tem dependências e outros ports dependem dele.
- Port Leaf: tem dependências, mas nenhum outro port depende dele.

Para listar essas categorias e procurar atualizações:

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache22-2.2.3
        ===>>> New version available: apache22-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

Este comando é usado para atualizar todos os ports desatualizados:

```
# portmaster -a
```



Por padrão, o Portmaster faz um backup do pacote antes de excluir o port existente. Se a instalação da nova versão for bem-sucedida, o Portmaster excluirá o backup. O uso de `-b` instrui o Portmaster a não excluir automaticamente o backup. Adicionar `-i` inicia o Portmaster no modo interativo, solicitando a confirmação antes de atualizar cada port. Muitas outras opções estão disponíveis. Leia a página de manual para o portmaster(8) para obter detalhes sobre seu uso.

Se forem encontrados erros durante o processo de atualização, adicione `-f` para atualizar e recompilar todos os ports:

```
# portmaster -af
```

O Portmaster também pode ser usado para instalar novos ports no sistema, atualizando todas as dependências antes de compilar e instalar o novo port. Para usar essa função, especifique o local do port na coleção de ports:

```
# portmaster shells/bash
```

Maiores informações sobre ports-mgmt/portmaster podem ser encontradas no pkg-descr.

4.5.3.3. Atualizando Ports Usando o Portupgrade

O ports-mgmt/portupgrade é outro utilitário que pode ser usado para atualizar ports. Ele instala um conjunto de aplicativos que podem ser usados para gerenciar ports. No entanto, ele depende do Ruby. Para instalar o port:

```
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean
```

Antes de executar uma atualização usando esse utilitário, é recomendável verificar a lista de ports instalados usando o `pkgdb -F` e corrigir todas as inconsistências relatadas.

Para atualizar todos os ports desatualizados instalados no sistema, use o `portupgrade -a`. Como alternativa, inclua `-i` para ser solicitado da confirmação de cada atualização individual:

```
# portupgrade -ai
```

Para atualizar apenas um aplicativo especifico em vez de todos os ports disponíveis, use `portupgrade pkgname`. É muito importante incluir `-R` para primeiro atualizar todos os ports requeridos pelo aplicativo fornecido:

```
# portupgrade -R firefox
```

Se `-P` estiver incluído, o Portupgrade procura pacotes disponíveis nos diretórios locais listados em `PKG_PATH`. Se nenhum estiver disponível localmente, ele buscará pacotes de um site remoto. Se os pacotes não puderem ser encontrados localmente ou buscados remotamente, o Portupgrade utilizará os ports. Para evitar completamente o uso do ports, especifique `-PP`. Este último conjunto de opções diz ao Portupgrade para cancelar se nenhum pacote estiver disponível:

```
# portupgrade -PP gnome3
```

Para obter apenas os distfiles do port, ou pacotes, se `-P` for especificado, sem compilar ou instalar nada, use `-F`. Para mais informações sobre todas as opções disponíveis, consulte a página de manual do `portupgrade`.

Maiores informações sobre o ports-mgmt/portupgrade podem ser encontradas no pkg-descr.

4.5.4. Ports e o Espaço em Disco

A utilização da coleção de ports irá ocupar espaço em disco ao longo do tempo. Depois de compilar e instalar um port, executar `make clean` dentro do diretório de um port limpará o diretório temporário de trabalho. Se o Portmaster for usado para instalar um port, ele removerá automaticamente esse diretório, a menos que `-K` seja especificado. Se o Portupgrade estiver instalado, este comando removerá todos os diretórios de trabalho encontrados na cópia local da coleção de ports:

```
# portsclean -C
```

Além disso, arquivos de distribuição de código-fonte desatualizados se acumulam no /usr/ports/distfiles ao longo do tempo. Para usar Portupgrade para excluir todos os distfiles que não são mais referenciados por nenhum port:

```
# portsclean -D
```

O Portupgrade pode remover todos os distfiles não referenciados por qualquer port atualmente instalado no sistema:

```
# portsclean -DD
```

Se o Portmaster estiver instalado, use:

```
# portmaster --clean-distfiles
```

Por padrão, esse comando é interativo e solicita que o usuário confirme se um distfile deve ser

excluído.

Além desses comandos, o ports-mgmt/pkg_cutleaves automatiza a tarefa de remover os ports instalados que não são mais necessários.

4.6. Compilando Pacotes com o Poudriere

O Poudriere é um utilitário licenciado sob a licença BSD para criar e testar pacotes do FreeBSD. Ele usa o jails do FreeBSD para configurar ambientes de compilação isolados. Esses jails podem ser usados para compilar pacotes para versões do FreeBSD que são diferentes do sistema no qual ele está instalado, e também para construir pacotes para o i386 se o host for um sistema amd64. Uma vez que os pacotes são compilados, eles estão em um layout idêntico aos espelhos oficiais. Esses pacotes podem ser usados pelo pkg(8) e por outras ferramentas de gerenciamento de pacotes.

O Poudriere é instalado usando o pacote ou port ports-mgmt/poudriere. A instalação inclui um arquivo de configuração de exemplo, /usr/local/etc/poudriere.conf.sample. Copie este arquivo para /usr/local/etc/poudriere.conf. Edite o arquivo copiado de acordo com a configuração local.

Embora o ZFS não seja necessário no sistema que executa o poudriere, o seu uso é benéfico. Quando o ZFS é usado, o `ZPOOL` deve ser especificado em /usr/local/etc/poudriere.conf e o `FREEBSD_HOST` deve ser definido para um espelho próximo. A definição de `CCACHE_DIR` permite o uso de devel/ccache para armazenar em cache a compilação e reduzir os tempos de compilação para o código compilado com freqüência. Pode ser conveniente colocar os conjuntos de dados do poudriere em uma árvore isolada montada em /poudriere. Os valores padrões para as outras variáveis de configuração são adequados.

O número de núcleos do processador detectados é usado para definir quantas compilações serão executadas em paralelo. Forneça memória virtual suficiente, seja por meio de RAM ou espaço de swap. Se a memória virtual se esgotar, as jails de compilação serão interrompidas e desativadas, resultando em mensagens de erro estranhas.

4.6.1. Inicializar o Jails e o Port Trees

Após a configuração, inicialize o poudriere para que ele instale um jail com a árvore do FreeBSD requerida e uma árvore de ports. Especifique um nome para o jail usando `-j` e a versão do FreeBSD com `-v`. Em sistemas que executam o FreeBSD/amd64, a arquitetura pode ser definida com `-a` para `i386` ou `amd64`. O padrão é a arquitetura mostrada pelo `uname`.

```
# poudriere jail -c -j 11amd64 -v 11.4-RELEASE
[00:00:00] Creating 11amd64 fs at /poudriere/jails/11amd64... done
[00:00:00] Using pre-distributed MANIFEST for FreeBSD 11.4-RELEASE amd64
[00:00:00] Fetching base for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/base.txz              125 MB 4110 kBps    31s
[00:00:33] Extracting base... done
[00:00:54] Fetching src for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/src.txz               154 MB 4178 kBps    38s
[00:01:33] Extracting src... done
[00:02:31] Fetching lib32 for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/lib32.txz              24 MB 3969 kBps    06s
```

```
[00:02:38] Extracting lib32... done
[00:02:42] Cleaning up... done
[00:02:42] Recording filesystem state for clean... done
[00:02:42] Upgrading using ftp
/etc/resolv.conf -> /poudriere/jails/11amd64/etc/resolv.conf
Looking up update.FreeBSD.org mirrors... 3 mirrors found.
Fetching public key from update4.freebsd.org... done.
Fetching metadata signature for 11.4-RELEASE from update4.freebsd.org... done.
Fetching metadata index... done.
Fetching 2 metadata files... done.
Inspecting system... done.
Preparing to download files... done.
Fetching 124
patches.....10....20....30....40....50....60....70....80....90....100....110....120..
done.
Applying patches... done.
Fetching 6 files... done.
The following files will be added as part of updating to
11.4-RELEASE-p1:
/usr/src/contrib/unbound/.github
/usr/src/contrib/unbound/.github/FUNDING.yml
/usr/src/contrib/unbound/contrib/drop2rpz
/usr/src/contrib/unbound/contrib/unbound_portable.service.in
/usr/src/contrib/unbound/services/rpz.c
/usr/src/contrib/unbound/services/rpz.h
/usr/src/lib/libc/tests/gen/spawnp_enoexec.sh
The following files will be updated as part of updating to
11.4-RELEASE-p1:
[…]
Installing updates...Scanning //usr/share/certs/blacklisted for certificates...
Scanning //usr/share/certs/trusted for certificates...
 done.
11.4-RELEASE-p1
[00:04:06] Recording filesystem state for clean... done
[00:04:07] Jail 11amd64 11.4-RELEASE-p1 amd64 is ready to be used
```

```
# poudriere ports -c -p local -m svn+https
[00:00:00] Creating local fs at /poudriere/ports/local... done
[00:00:00] Checking out the ports tree... done
```

Em um único computador, o poudriere pode construir ports com várias configurações, em vários jails e de diferentes árvores de ports. Configurações customizadas para estas combinações são chamadas de *sets*. Veja a seção CUSTOMIZAÇÃO do poudriere(8) para detalhes depois que o ports-mgmt/poudriere ou o ports-mgmt/poudriere-devel estiver instalado.

A configuração básica mostrada aqui coloca um único jail-, port-, e um set específico make.conf em /usr/local/etc/poudriere.d. O nome do arquivo neste exemplo é criado combinando o nome do jail, o nome do port e o nome do set: 11amd64-local-workstation-make.conf. O sistema make.conf e este novo arquivo são combinados em tempo de compilação para criar o make.conf usado pela jail de

compilação.

Os pacotes a serem criados são inseridos em 11amd64-local-workstation-pkglist:

```
editors/emacs
devel/git
ports-mgmt/pkg
...
```

Opções e dependências para os ports especificados são configuradas:

```
# poudriere options -j 11amd64 -p local -z workstation -f 11amd64-local-workstation-
pkglist
```

Finalmente, os pacotes são compilados e um repositório de pacotes é criado:

```
# poudriere bulk -j 11amd64 -p local -z workstation -f 11amd64-local-workstation-
pkglist
```

Durante a execução, pressionar `Ctrl` + `t` exibe o estado atual da compilação. O Poudriere também cria arquivos em /poudriere/logs/bulk/jailname que podem ser usados com um servidor da Web para exibir informações de compilação.

Após a conclusão, os novos pacotes estão agora disponíveis para instalação a partir do repositório poudriere.

Para obter maiores informações sobre o uso do poudriere, consulte poudriere(8) e o site principal, https://github.com/freebsd/poudriere/wiki.

4.6.2. Configurando Clientes do pkg para usar um repositório de Poudriere

Embora seja possível usar um repositório personalizado ao lado do repositório oficial, às vezes é útil desativar o repositório oficial. Isso é feito criando um arquivo de configuração que substitui e desativa o arquivo de configuração oficial. Crie o /usr/local/etc/pkg/repos/FreeBSD.conf que deverá conter o seguinte:

```
FreeBSD: {
    enabled: no
}
```

Geralmente é mais fácil disponibilizar um repositório poudriere para as máquinas clientes via HTTP. Configure um servidor web para disponibilizar o diretório de pacotes, por exemplo: /usr/local/poudriere/data/packages/11amd64, onde 11amd64 é o nome da compilação.

Se a URL para o repositório de pacotes for: `http://pkg.example.com/11amd64`, o arquivo de configuração do repositório em /usr/local/etc/pkg/repos/custom.conf ficaria assim:

```
custom: {
    url: "http://pkg.example.com/11amd64",
    enabled: yes,
}
```

4.7. Considerações pós-instalação

Independentemente do software ter sido instalado a partir de um pacote binário ou de um port, a maioria dos aplicativos de terceiros requer algum nível de configuração após a instalação. Os seguintes comandos e locais podem ser usados para ajudar a determinar o que foi instalado com o aplicativo.

- A maioria dos aplicativos instala pelo menos um arquivo de configuração padrão em /usr/local/etc. Nos casos em que um aplicativo possui um grande número de arquivos de configuração, um subdiretório será criado para mantê-los. Geralmente, os arquivos de configuração de exemplo são instalados e terminam com um sufixo, como .sample. Os arquivos de configuração devem ser revisados e possivelmente editados para atender às necessidades do sistema. Para editar um arquivo de amostra, primeiro copie-o sem a extensão .sample.
- As aplicações que fornecem documentação irão instalá-la em /usr/local/shared/doc e muitos aplicativos também instalam páginas de manual. Esta documentação deve ser consultada antes de continuar.
- Alguns aplicativos executam serviços que devem ser adicionados ao /etc/rc.conf antes de iniciar o aplicativo. Esses aplicativos geralmente instalam um script de inicialização em /usr/local/etc/rc.d. Veja Iniciando Serviços para maiores informações.



Por padrão, os aplicativos não executam o script de inicialização durante a instalação, nem executam o script de parada após a desinstalação ou atualização. Essa decisão é deixada para o administrador do sistema.

- Os usuários de csh(1) devem executar `rehash` para reconstruir a lista dos binários conhecidos nos shells `PATH`.
- Use `pkg info` para determinar quais arquivos, páginas man e binários foram instalados com o aplicativo.

4.8. Lidando com ports quebrados

Quando um port não é compilado ou instalado, tente o seguinte:

1. Procure para ver se há uma correção pendente para o port no Banco de Dados do Relatório de Problemas. Nesse caso, implementar a correção proposta pode corrigir o problema.
2. Peça ajuda ao mantenedor do port. Digite `make maintainer` no diretório do port ou leia o Makefile do port para encontrar o endereço de e-mail do mantenedor. Lembre-se de incluir a linha `$FreeBSD:` do Makefile do port e a saída que leva ao erro no e-mail para o mantenedor.



Alguns ports não são mantidos por um indivíduo, mas sim por um grupo de

> mantenedores representado por uma lista de discussão. Muitos, mas não todos, esses endereços se parecem com freebsd-listname@FreeBSD.org. Por favor, leve isso em consideração ao enviar um email.
>
> Em particular, os ports mantidos por ports@FreeBSD.org não são mantidos por um indivíduo específico. Em vez disso, quaisquer correções e suporte vêm da comunidade geral que se inscreve nessa lista de discussão. Mais voluntários são sempre necessários!

Se não houver resposta ao email, use o Bugzilla para enviar um relatório de bug usando as instruções em Escrevendo Relatórios de Problemas do FreeBSD.

3. Conserte-o! O Porters Handbook inclui informações detalhadas sobre a infra-estrutura da arvore de ports para que você possa corrigir possiveis erros na compilacao de um ports que quebrou ou ocasionou um erro de compilacao ou até mesmo submeta seu próprio projeto!
4. Instale o pacote em vez do port usando as instruções em Usando o pkg para o gerenciamento de pacotes binários.

Capítulo 5. O sistema X Window

5.1. Sinopse

Uma instalação padrão do FreeBSD usando o bsdinstall não irá instalar automaticamente uma interface gráfica para o usuário. Este capítulo descreve como instalar e configurar o Xorg, que fornece o sistema X Window open source usado para fornecer um ambiente gráfico. Em seguida, descreve como encontrar e instalar um ambiente de área de trabalho ou um gerenciador de janelas.



Os usuários que preferem um método de instalação que configure automaticamente o Xorg devem consultar FuryBSD, GhostBSD ou MidnightBSD.

Para obter maiores informações sobre o hardware de vídeo suportado pelo Xorg, consulte o web site x.org.

Depois de ler este capítulo, você saberá:

- Quais são os vários componentes do Sistema X Window e como eles interoperam.
- Como instalar e configurar o Xorg.
- Como instalar e configurar vários gerenciadores de janelas e ambientes de desktop.
- Como usar fontes TrueType™ no Xorg.
- Como configurar seu sistema para usar um sistema de login gráfico (XDM).

Antes de ler este capítulo, você deve:

- Saber como instalar softwares adicionais de terceiros, conforme descrito em Instalando Aplicativos. Pacotes e Ports.

5.2. Terminologia

Embora não seja necessário entender todos os detalhes dos vários componentes do Sistema X Window e como eles interagem, algum conhecimento básico desses componentes pode ser útil.

Servidor X

O X foi projetado desde o início para ser centrado em rede e para adotar um modelo "cliente-servidor". Neste modelo, o "Servidor X" é executado no computador que possui o teclado, o monitor e o mouse conectados. A responsabilidade do servidor inclui tarefas como gerenciar o monitor, manipular a entrada do teclado e do mouse e manipular a entrada ou saída de outros dispositivos, como um tablet ou um projetor de vídeo. Isso confunde algumas pessoas, porque a terminologia X é exatamente o oposto do que eles esperam. Eles esperam que o "X server" seja a grande máquina poderosa no final das contas, e o "Ciente X" seja a máquina em sua mesa.

Cliente X

Cada aplicativo X, como o XTerm ou o Firefox, é um "cliente". Um cliente envia mensagens para o servidor, como "Por favor, desenhe uma janela nessas coordenadas", e o servidor envia de volta mensagens como "O usuário apenas clicou no botão OK".

Em um ambiente doméstico ou de uma pequena empresa, o servidor X e os clientes X geralmente são executados no mesmo computador. Também é possível executar o servidor X em um computador menos potente e executar os aplicativos X em um sistema mais poderoso. Nesse cenário, a comunicação entre o cliente X e o servidor ocorre através da rede.

Gerenciador de janelas

O X não dita como as janelas devem se parecer na tela, como movê-las com o mouse, quais teclas devem ser usadas para mover-se entre as janelas, como devem ficar as barras de título em cada janela, se elas têm ou não botões para fechar nelas e assim por diante. Em vez disso, o X delega essa responsabilidade para um gerenciador de janelas separado. Existem dezenas de gerenciadores de janelas disponíveis. Cada gerenciador de janelas oferece uma aparência diferente: alguns oferecem suporte a desktops virtuais, alguns permitem pressionamentos de tecla personalizados para gerenciar a área de trabalho, alguns têm um botão "Iniciar" e alguns são personalizáveis, permitindo uma alteração completa da aparência da área de trabalho. Os gerenciadores de janelas estão disponíveis na categoria x11-wm da coleção de ports.

Cada gerenciador de janelas usa um mecanismo de configuração diferente. Alguns esperam que o arquivo de configuração seja escrito à mão, enquanto outros fornecem ferramentas gráficas para a maioria das tarefas de configuração.

Ambiente de desktop

O KDE e o GNOME são considerados ambientes de desktop, pois incluem um conjunto completo de aplicativos para executar tarefas comuns de desktop. Estes podem incluir pacotes de escritório, navegadores da web e jogos.

Política de foco

O gerenciador de janelas é responsável pela política de foco do mouse. Essa política fornece alguns meios para escolher qual janela está recebendo ativamente as teclas digitadas e também deve indicar visivelmente qual janela está ativa no momento.

Uma política de foco é chamada "click-to-focus". Neste modelo, uma janela fica ativa ao receber um clique do mouse. Na política "focus-follows-mouse", a janela que está sob o ponteiro do mouse tem foco e o foco é alterado apontando para outra janela. Se o mouse estiver sobre a janela raiz, esta janela estará focada. No modelo "sloppy-focus", se o mouse for movido sobre a janela raiz, a janela usada mais recentemente ainda terá o foco. Com sloppy-focus, o foco só é alterado quando o cursor entra em uma nova janela, e não ao sair da janela atual. Na política de "click-to-focus", a janela ativa é selecionada pelo clique do mouse. A janela pode então ser destacada para aparecer na frente de todas as outras janelas. Todas as teclas digitadas serão direcionadas para esta janela, mesmo se o cursor for movido para outra janela.

Diferentes gerenciadores de janela suportam diferentes modelos de foco. Todos eles suportam click-to-focus, e a maioria deles também suporta outras políticas. Consulte a documentação do gerenciador de janelas para determinar quais modelos de foco estão disponíveis.

Widgets

Widget é um termo para todos os itens na interface do usuário que podem ser clicados ou manipulados de alguma forma. Isso inclui botões, caixas de seleção, botões de opção, ícones e listas. Um kit de ferramentas de widget é um conjunto de widgets usado para criar aplicativos

gráficos. Existem vários toolkits de widgets populares, incluindo o Qt, usado pelo KDE, e o GTK+, usado pelo GNOME. Como resultado, os aplicativos terão uma aparência e comportamentos diferentes, dependendo de qual kit de ferramentas de widget foi usado para criar o aplicativo.

5.3. Instalando o Xorg

No FreeBSD, o Xorg pode ser instalado como um pacote ou port.

O pacote binário pode ser instalado rapidamente, mas com menos opções de personalização:

```
# pkg install xorg
```

Para compilar e instalar a partir da Coleção de Ports:

```
# cd /usr/ports/x11/xorg
# make install clean
```

Qualquer uma dessas instalações resulta no sistema completo do Xorg sendo instalado. Pacotes binários são a melhor opção para a maioria dos usuários.

Uma versão menor do sistema X adequada para usuários experientes está disponível em x11/xorg-minimal. A maioria dos documentos, bibliotecas e aplicativos não será instalada. Algumas aplicações requerem esses componentes adicionais para funcionarem.

5.4. Configuração do Xorg

5.4.1. Inicio Rápido

O Xorg suporta as placas de vídeo, teclados e dispositivos USB mais comuns.



Placas de vídeo, monitores e dispositivos de entrada são detectados automaticamente e não exigem nenhuma configuração manual. Não crie o xorg.conf ou execute o passo `-configure` a menos que a configuração automática falhe.

1. Se o Xorg tiver sido usado neste computador antes, mova ou remova qualquer arquivo de configuração existente:

   ```
   # mv /etc/X11/xorg.conf ~/xorg.conf.etc
   # mv /usr/local/etc/X11/xorg.conf ~/xorg.conf.localetc
   ```

2. Adicione o usuário que executará o Xorg ao grupo `video` ou `wheel` para ativar a aceleração 3D quando disponível. Para adicionar o usuário *jru* ao grupo que estiver disponível:

```
# pw groupmod video -m jru || pw groupmod wheel -m jru
```

3. O gerenciador de janelas twm é incluído por padrão. Ele é chamado quando o Xorg se inicia:

```
% startx
```

4. Em algumas versões mais antigas do FreeBSD, o console do sistema deve ser definido como vt(4) antes que a volta para o console de texto funcione corretamente. Veja Configuração do Modo Kernel (KMS).

5.4.2. Grupo de Usuários para Vídeo Acelerado

O acesso ao /dev/dri é necessário para permitir a aceleração 3D nas placas de vídeo. Geralmente é mais simples adicionar o usuário que estará executando o X no grupo `video` ou no `wheel`. Aqui, o pw(8) é usado para adicionar o usuário *slurms* ao grupo `video`, ou ao grupo `wheel` se não houver nenhum grupo `video`:

```
# pw groupmod video -m slurms || pw groupmod wheel -m slurms
```

5.4.3. Configuração do Modo Kernel (KMS)

Quando o computador alterna a exibição do console para uma resolução de tela mais alta para o X, ele deve definir o *modo* da saída de vídeo. Versões recentes do Xorg usam um sistema dentro do kernel para fazer essas mudanças de modo mais eficiente. Versões mais antigas do FreeBSD usam o sc(4), que não tem conhecimento do sistema KMS. O resultado final é que depois de fechar o X, o console do sistema fica em branco, embora ainda esteja funcionando. O console vt(4) mais recente evita esse problema.

Adicione esta linha ao /boot/loader.conf para ativar o vt(4):

```
kern.vty=vt
```

5.4.4. Arquivos de Configuração

A configuração manual geralmente não é necessária. Por favor, não crie manualmente arquivos de configuração, a menos que a autoconfiguração não funcione.

5.4.4.1. Diretório

O Xorg procura em vários diretórios por arquivos de configuração. O /usr/local/etc/X11/ é o diretório recomendado para esses arquivos no FreeBSD. Usar esse diretório ajuda a manter os arquivos dos aplicativos separados dos arquivos do sistema operacional.

Armazenar arquivos de configuração no diretório legado /etc/X11/ ainda funciona. No entanto, isso combina arquivos de aplicativos com os arquivos básicos do FreeBSD e não é recomendado.

5.4.4.2. Arquivos Únicos ou Múltiplos

É mais fácil usar multiplos arquivos em que cada um controla uma configuração específica ao invés do único e tradicional xorg.conf. Esses arquivos são armazenados no subdiretório xorg.conf.d/ do diretório principal do arquivo de configuração. O caminho completo é tipicamente /usr/local/etc/X11/xorg.conf.d/.

Exemplos desses arquivos serão mostrados posteriormente nesta seção.

O tradicional e único arquivo xorg.conf ainda funciona, mas não é tão claro e nem tão flexível quanto vários arquivos no subdiretório xorg.conf.d/.

5.4.5. Placas de Vídeo

Devido as mudanças feitas nas versões recentes do FreeBSD, agora é possível usar drivers gráficos fornecidos pelo Framework do Ports, assim como pelos pacotes. Assim sendo, os usuários podem usar um dos seguintes drivers disponíveis em graphics/drm-kmod.

Intel KMS driver

A aceleração 2D e 3D é suportada na maioria das placas gráficas do driver Intel KMS fornecidas pela Intel.

Nome do driver: `i915kms`

A aceleração 2D e 3D é suportada na maioria das placas gráficas de driver Radeon KMS mais antigas fornecidas pela AMD.

Nome do Driver: `radeonkms`

A aceleração 2D e 3D é suportada nas mais recentes placas gráficas do driver AMD KMS fornecidas pela AMD.

Nome do Driver: `amdgpu`

Para referência, veja https://en.wikipedia.org/wiki/List_of_Intel_graphics_processing_units ou https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units para uma lista das GPUs suportadas.

Intel™

A aceleração 3D é suportada na maioria dos chipsets gráficos da Intel™ até o Ivy Bridge (HD Graphics 2500, 4000 e P4000), incluindo Iron Lake (HD Graphics) e Sandy Bridge (HD Graphics 2000) .

Nome do driver: `intel`

Para referência, veja https://en.wikipedia.org/wiki/List_of_Intel_graphics_processing_units.

AMD™ Radeon

Aceleração 2D e 3D é suportada em placas Radeon das mais antigas até a série HD6000.

Nome do Driver: `radeon`

Para referência, veja https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units.

NVIDIA

Vários drivers da NVIDIA estão disponíveis na categoria x11 da Coleção de Ports. Instale o driver que corresponde à sua placa de vídeo.

Para referência, veja https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units.

Gráficos Híbridos de Combinação

Alguns notebooks adicionam unidades de processamento gráfico adicionais àquelas incorporadas ao chipset ou ao processador. O *Optimus* combina o hardware da Intel™ e da NVIDIA. O *Switchable Graphics* ou *Hybrid Graphics* são uma combinação dos processadores Intel™ ou AMD™ e uma GPUAMD™ Radeon.

As implementações desses sistemas gráficos híbridos variam e o Xorg no FreeBSD não é capaz de controlar todas as versões deles.

Alguns computadores fornecem uma opção no BIOS para desativar um dos adaptadores gráficos ou selecionar um modo *discreto* que pode ser usado com um dos drivers de placa de vídeo padrão. Por exemplo, às vezes é possível desativar a GPU NVIDIA em um sistema Optimus. O vídeo Intel™ pode então ser usado com um driver Intel™.

Configurações de BIOS dependem do modelo do computador. Em algumas situações, ambas GPUs podem ser deixadas ativadas, mas criar um arquivo de configuração que use apenas a GPU principal na seção `Device` é o suficiente para tornar esse sistema funcional.

Outras placas de vídeo

Drivers para algumas placas de vídeo menos comuns podem ser encontrados no diretório x11-drivers da Coleção de Ports.

PLacas que não são suportadas por um driver específico ainda podem ser usadas com o driver x11-drivers/xf86-video-vesa. Este driver é instalado pelo x11/xorg. Ele também pode ser instalado manualmente como x11-drivers/xf86-video-vesa. O Xorg tenta usar este driver quando um driver específico não é encontrado para a placa de vídeo.

O x11-drivers/xf86-video-scfb é um driver de vídeo não especializado similar que funciona em muitos computadores UEFI e ARM™.

Configurando o driver de vídeo em um arquivo

Para definir o driver Intel™ em um arquivo de configuração:

Exemplo 15. Selecione o driver de vídeo Intel™ em um arquivo

```
/usr/local/etc/X11/xorg.conf.d/driver-intel.conf
```

```
Section "Device"
    Identifier "Card0"
    Driver     "intel"
    # BusID    "PCI:1:0:0"
EndSection
```

Se mais de uma placa de vídeo estiver presente, o identificador `BusID` pode ser descomentado e configurado para selecionar a placa desejada. Uma lista de barramento de placa de vídeo ID pode ser exibida com `pciconf -lv | grep -B3 display`.

Para definir o driver Radeon em um arquivo de configuração:

Exemplo 16. Selecione o driver de vídeo Radeon em um arquivo

/usr/local/etc/X11/xorg.conf.d/driver-radeon.conf

```
Section "Device"
    Identifier "Card0"
    Driver     "radeon"
EndSection
```

Para definir o driver VESA em um arquivo de configuração:

Exemplo 17. Selecione o driver de vídeo VESA em um arquivo

/usr/local/etc/X11/xorg.conf.d/driver-vesa.conf

```
Section "Device"
    Identifier "Card0"
    Driver     "vesa"
EndSection
```

Para definir o driver `scfb` para uso com um computador UEFI ou ARM™:

Exemplo 18. Selecione o driver de vídeo `scfb` em um arquivo

/usr/local/etc/X11/xorg.conf.d/driver-scfb.conf

```
Section "Device"
    Identifier "Card0"
    Driver     "scfb"
EndSection
```

5.4.6. Monitores

Quase todos os monitores suportam o padrão Extended Display Identification Data (EDID). O Xorg usa o EDID para se comunicar com o monitor e detectar as resoluções e taxas de atualização suportadas. Em seguida, seleciona a combinação mais adequada de configurações para usar com esse monitor.

Outras resoluções suportadas pelo monitor podem ser escolhidas definindo a resolução desejada nos arquivos de configuração, ou após o servidor X ter sido iniciado com xrandr(1).

Usando xrandr(1)

Execute o xrandr(1) sem nenhum parâmetro para ver uma lista de saídas de vídeo e modos de monitor detectados:

```
% xrandr
Screen 0: minimum 320 x 200, current 3000 x 1920, maximum 8192 x 8192
DVI-0 connected primary 1920x1200+1080+0 (normal left inverted right x axis y axis)
495mm x 310mm
   1920x1200     59.95*+
   1600x1200     60.00
   1280x1024     85.02    75.02    60.02
   1280x960      60.00
   1152x864      75.00
   1024x768      85.00    75.08    70.07    60.00
   832x624       74.55
   800x600       75.00    60.32
   640x480       75.00    60.00
   720x400       70.08
DisplayPort-0 disconnected (normal left inverted right x axis y axis)
HDMI-0 disconnected (normal left inverted right x axis y axis)
```

Isso mostra que a saída `DVI-0` está sendo usada para exibir uma resolução de tela de 1920x1200 pixels a uma taxa de atualização de cerca de 60 Hz. Os monitores não estão conectados aos conectores `DisplayPort-0` e `HDMI-0`.

Qualquer um dos outros modos de exibição pode ser selecionado com xrandr(1). Por exemplo, para mudar para 1280x1024 a 60 Hz:

```
% xrandr --mode 1280x1024 --rate 60
```

Uma tarefa comum é usar a saída de vídeo externa em um notebook para um projetor de vídeo.

O tipo e a quantidade de conectores de saída variam entre os dispositivos, e o nome dado a cada saída varia de driver para driver. O que um driver chama de `HDMI-1`, outro pode chamar de `HDMI1`. Portanto, o primeiro passo é executar xrandr(1) para listar todas as saídas disponíveis:

```
% xrandr
Screen 0: minimum 320 x 200, current 1366 x 768, maximum 8192 x 8192
```

```
LVDS1 connected 1366x768+0+0 (normal left inverted right x axis y axis) 344mm x
193mm
   1366x768      60.04*+
   1024x768      60.00
   800x600       60.32    56.25
   640x480       59.94
VGA1 connected (normal left inverted right x axis y axis)
   1280x1024     60.02 +  75.02
   1280x960      60.00
   1152x864      75.00
   1024x768      75.08    70.07    60.00
   832x624       74.55
   800x600       72.19    75.00    60.32    56.25
   640x480       75.00    72.81    66.67    60.00
   720x400       70.08
HDMI1 disconnected (normal left inverted right x axis y axis)
DP1 disconnected (normal left inverted right x axis y axis)
```

Quatro saídas foram encontradas: os conectores `LVDS1` e `VGA1`, `HDMI1` e `DP1` do painel interno.

O projetor foi conectado à saída `VGA1`. O xrandr(1) agora é usado para definir essa saída para a resolução nativa do projetor e adicionar o espaço adicional à direita da área de trabalho:

```
% xrandr --output VGA1 --auto --right-of LVDS1
```

A opção `--auto` escolhe a resolução e a taxa de atualização detectadas pelo EDID. Se a resolução não for detectada corretamente, um valor fixo pode ser fornecido com `--mode` em vez da instrução `--auto`. Por exemplo, a maioria dos projetores pode ser usada com uma resolução de 1024x768, que é definida com `--mode 1024x768`.

O xrandr(1) geralmente é executado a partir do .xinitrc para definir o modo apropriado quando o X é iniciado.

Configurando a resolução do monitor em um arquivo

Para definir uma resolução de tela de 1024x768 em um arquivo de configuração:

Exemplo 19. Defina a resolução de tela em um arquivo

/usr/local/etc/X11/xorg.conf.d/screen-resolution.conf

```
Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    SubSection "Display"
    Modes      "1024x768"
    EndSubSection
EndSection
```

Os poucos monitores que não possuem EDID podem ser configurados setando o `HorizSync` e o `VertRefresh` para o intervalo de frequências suportado pelo monitor.

Exemplo 20. Configurando Manualmente as Frequências do Monitor

/usr/local/etc/X11/xorg.conf.d/monitor0-freq.conf

```
Section "Monitor"
    Identifier   "Monitor0"
    HorizSync    30-83   # kHz
    VertRefresh  50-76   # Hz
EndSection
```

5.4.7. Dispositivos de Entrada

5.4.7.1. Teclados

Layout do Teclado

A localização padronizada das teclas em um teclado é chamada de *layout*. Layouts e outros parâmetros ajustáveis são listados em xkeyboard-config(7).

Um layout dos Estados Unidos é o padrão. Para selecionar um layout alternativo, defina as opções `XkbLayout` e `XkbVariant` em um `InputClass`. Isso será aplicado a todos os dispositivos de entrada que correspondam à classe.

Este exemplo seleciona um layout de teclado Francês.

Exemplo 21. Definindo um layout de teclado

/usr/local/etc/X11/xorg.conf.d/keyboard-fr.conf

```
Section "InputClass"
    Identifier  "KeyboardDefaults"
    MatchIsKeyboard "on"
    Option      "XkbLayout" "fr"
EndSection
```

Exemplo 22. Definindo vários layouts de teclado

Define os layouts de teclado para Estados Unidos, Espanhol e Ucraniano. Alterne entre esses layouts pressionando `Alt` + `Shift`. O x11/xxkb ou x11/sbxkb pode ser usado para um melhor controle da mudança de layout e dos indicadores do layout atual.

/usr/local/etc/X11/xorg.conf.d/kbd-layout-multi.conf

```
Section "InputClass"
```

```
    Identifier  "All Keyboards"
    MatchIsKeyboard "yes"
    Option      "XkbLayout" "us, es, ua"
EndSection
```

Fechando o Xorg pelo teclado

X pode ser fechado com uma combinação de teclas. Por padrão, essa combinação de teclas não está definida porque entra em conflito com os comandos do teclado para alguns aplicativos. Ativar essa opção requer alterações na seção `InputDevice` do teclado:

Exemplo 23. Ativando o fechamento de X pelo teclado

/usr/local/etc/X11/xorg.conf.d/keyboard-zap.conf

```
Section "InputClass"
    Identifier  "KeyboardDefaults"
    MatchIsKeyboard "on"
    Option      "XkbOptions" "terminate:ctrl_alt_bksp"
EndSection
```

5.4.7.2. Mouse e Dispositivos Similares



Se ao usar xorg-server 1.20.8 ou maior no FreeBSD 12.1 e não usar moused(8), adicione `kern.evdev.rcpt_mask=12` ao arquivo /etc/sysctl.conf:

Muitos parâmetros do mouse podem ser ajustados com opções de configuração. Veja mousedrv(4) para obter uma lista completa.

Botões do Mouse

O número de botões em um mouse pode ser definido na seção `InputDevice` do xorg.conf. Para definir o número de botões para 7:

Exemplo 24. Definindo o número de botões do mouse

/usr/local/etc/X11/xorg.conf.d/mouse0-buttons.conf

```
Section "InputDevice"
    Identifier  "Mouse0"
    Option      "Buttons" "7"
EndSection
```

5.4.8. Configuração manual

Em alguns casos, a autoconfiguração do Xorg não funciona com alguns hardwares específicos, ou

uma configuração diferente é desejada. Para esses casos, um arquivo de configuração personalizado pode ser criado.



Não crie arquivos de configuração manualmente, a menos que seja necessário. A configuração manual desnecessária pode impedir o funcionamento adequado.

Um arquivo de configuração pode ser gerado pelo Xorg baseado no hardware detectado. Esse arquivo geralmente é um ponto de partida útil para configurações personalizadas.

Gerando um arquivo xorg.conf:

```
# Xorg -configure
```

O arquivo de configuração é salvo em /root/xorg.conf.new. Faça as alterações desejadas e teste esse arquivo(usando `-retro` assim será exibido um fundo visível) com:

```
# Xorg -retro -config /root/xorg.conf.new
```

Após a nova configuração ter sido ajustada e testada, ela pode ser dividida em arquivos menores no diretorio, /usr/local/etc/X11/xorg.conf.d/.

5.5. Usando fontes no Xorg

5.5.1. Fontes Type1

As fontes padrões que vem com o Xorg não são adequadas para muitas aplicações desktop. As fontes grandes aparecem irregulares e com aparência não profissional, e as fontes pequenas são quase ilegíveis. Contudo existem muitas fontes Type1 (PostScript™) gratuitas de alta qualidade prontas para uso no Xorg. Por exemplo, a coleção de fontes URW (x11-fonts/urwfonts) inclui versões de alta qualidade de fontes type1 padrão (Times Roman™, Helvetica™, Palatino™ e outras). A coleção Freefonts (x11-fonts/freefonts) inclui muito mais fontes, mas a maioria delas direcionadas para uso em softwares gráficos como o Gimp, e não são tão completas para servir como fontes de tela. Além disso, o Xorg pode ser configurado para usar fontes TrueType™ com um minimo esforço. Para maiores detalhes sobre isso veja a página de manual do X(7) ou Fontes TrueType™.

Para instalar as coleções de fontes Type1 usando pacotes binários, execute os seguintes comandos:

```
# pkg install urwfonts
```

Como alternativa, para compilar a partir da coleção de Ports, execute os seguintes comandos:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

Proceda da mesma forma com a freefont ou outras coleções. Para que o servidor X detecte essas fontes, adicione uma linha apropriada ao arquivo de configuração do servidor X (/etc/X11/xorg.conf):

```
FontPath "/usr/local/shared/fonts/urwfonts/"
```

Alternativamente, na linha de comando de execução da sessão X:

```
% xset fp+ /usr/local/shared/fonts/urwfonts
% xset fp rehash
```

Isso funcionará, mas será perdido quando a sessão X for fechada, a menos que seja adicionada ao arquivo de inicialização (~/.xinitrc para uma sessão `startx` normal ou ~/.xsession ao efetuar login através de um gerenciador de login gráfico como o XDM). Uma terceira forma é usar o novo /usr/local/etc/fonts/local.conf, como demonstrado em Fontes com Anti-Alias.

5.5.2. Fontes TrueType™

O Xorg tem suporte nativo para renderizar fontes TrueType™. Existem dois módulos diferentes que podem ativar essa funcionalidade. O módulo freetype é usado neste exemplo porque é mais consistente com os outros backends de renderização de fonte. Para habilitar o módulo freetype, basta adicionar a seguinte linha à seção `"Module"` do /etc/X11/xorg.conf.

```
Load  "freetype"
```

Agora crie um diretório para as fontes TrueType™ (por exemplo, /usr/local/shared/fonts/TrueType) e copie todas as fontes TrueType™ para este diretório. Tenha em mente que as fontes TrueType™ não podem ser obtidas diretamente de um Apple™Mac™; elas devem estar no formato UNIX™/MS-DOS™/Windows™ para uso pelo Xorg. Uma vez que os arquivos foram copiados para este diretório, use mkfontdir para criar um fonts.dir, para que o renderizador de fontes do X saiba que esses novos arquivos foram instalados. O `mkfontdir` pode ser instalado como um pacote binário com o comando:

```
# pkg install mkfontdir
```

Em seguida, crie um índice de arquivos de fontes X em um diretório:

```
# cd /usr/local/shared/fonts/TrueType
# mkfontdir
```

Agora adicione o diretório TrueType™ ao caminho da fonte. Isso é exatamente o mesmo descrito em Fontes Type1:

```
% xset fp+ /usr/local/shared/fonts/TrueType
% xset fp rehash
```

ou adicione uma linha de `FontPath` ao xorg.conf.

Agora, o Gimp, o Apache OpenOffice e todos os outros aplicativos X devem reconhecer as fontes TrueType™ instaladas. Fontes extremamente pequenas (como o texto em uma tela de alta resolução em uma página da Web) e fontes extremamente grandes (dentro do StarOffice™) ficarão muito melhores agora.

5.5.3. Fontes com Anti-Alias

Todas as fontes do Xorg que são encontradas em /usr/local/shared/fonts/ e ~/.fonts/ são automaticamente disponibilizadas para anti-aliasing para aplicativos compatíveis com Xft-aware. Os aplicativos mais recentes são compatíveis com o Xft-aware, incluindo o KDE, o GNOME e o Firefox.

Para controlar quais fontes são anti-aliased, ou para configurar as propriedades do anti-alias, crie (ou edite, se já existir) o arquivo /usr/local/etc/fonts/local.conf. Vários recursos avançados do sistema de fontes Xft podem ser ajustados usando este arquivo; Esta seção descreve apenas algumas possibilidades simples. Para maiores detalhes, por favor veja fonts-conf(5).

Este arquivo deve estar no formato XML. Preste muita atenção ao uso de letras maiusculas e minusculas e certifique-se de que todas as tags estejam corretamente fechadas. O arquivo começa com o cabeçalho XML usual seguido por uma definição DOCTYPE e, em seguida, a tag `<fontconfig>`:

```
<?xml version="1.0"?>
      <!DOCTYPE fontconfig SYSTEM "fonts.dtd">
      <fontconfig>
```

Como dito anteriormente, todas as fontes em /usr/local/shared/fonts/ e ~/.fonts/ já estão disponíveis para aplicativos Xft-aware. Para adicionar outro diretório fora dessas duas árvores de diretórios, adicione uma linha como essa a /usr/local/etc/fonts/local.conf:

```
<dir>/path/to/my/fonts</dir>
```

Depois de adicionar novas fontes e especialmente novos diretórios de fontes, reconstrua os caches de fontes:

```
# fc-cache -f
```

O anti-aliasing torna as bordas um pouco confusas, o que torna o texto muito pequeno mais legível e remove os "serrilhados" do texto grande, mas pode causar fadiga ocular se aplicado ao texto normal. Para excluir tamanhos de fonte menores que 14 pontos do anti-aliasing, inclua estas linhas:

```
        <match target="font">
        <test name="size" compare="less">
        <double>14</double>
        </test>
        <edit name="antialias" mode="assign">
        <bool>false</bool>
        </edit>
    </match>
    <match target="font">
        <test name="pixelsize" compare="less" qual="any">
        <double>14</double>
        </test>
        <edit mode="assign" name="antialias">
        <bool>false</bool>
        </edit>
    </match>
```

O espaçamento para algumas fontes monoespaçadas também pode ser inadequado com o anti-aliasing. Este parece ser um problema com o KDE, em particular. Uma possível correção é forçar o espaçamento dessas fontes para que seja 100. Adicione essas linhas:

```
    <match target="pattern" name="family">
       <test qual="any" name="family">
           <string>fixed</string>
       </test>
       <edit name="family" mode="assign">
           <string>mono</string>
       </edit>
    </match>
    <match target="pattern" name="family">
        <test qual="any" name="family">
        <string>console</string>
        </test>
        <edit name="family" mode="assign">
        <string>mono</string>
        </edit>
    </match>
```

(isto cria um apelido para outros nomes comuns para fontes fixas como `"mono"`), e então adicione:

```
         <match target="pattern" name="family">
         <test qual="any" name="family">
         <string>mono</string>
         </test>
         <edit name="spacing" mode="assign">
         <int>100</int>
         </edit>
```

```
    </match>
```

Determinadas fontes, como Helvetica, podem ter um problema com o anti-alias. Geralmente isso se manifesta como uma fonte que parece cortada ao meio na vertical. Na pior das hipóteses, pode causar falhas nos aplicativos. Para evitar isso, considere adicionar o seguinte ao local.conf:

```
        <match target="pattern" name="family">
        <test qual="any" name="family">
        <string>Helvetica</string>
        </test>
        <edit name="family" mode="assign">
        <string>sans-serif</string>
        </edit>
    </match>
```

Depois de editar o local.conf, certifique-se de finalizar o arquivo com a tag `</fontconfig>`. Não fazer isso fará com que as alterações sejam ignoradas.

Os usuários podem adicionar configurações personalizadas criando seus próprios arquivos ~/.config/fontconfig/fonts.conf. Este arquivo usa o mesmo formato XML descrito acima.

Um último ponto: com uma tela de LCD, a amostragem de sub-pixels pode ser desejada. Isso basicamente trata os componentes vermelho, verde e azul (separados horizontalmente) separadamente para melhorar a resolução horizontal; os resultados podem ser dramáticos. Para habilitar isso, adicione a linha em algum lugar do local.conf:

```
    <match target="font">
        <test qual="all" name="rgba">
        <const>unknown</const>
        </test>
        <edit name="rgba" mode="assign">
        <const>rgb</const>
        </edit>
    </match>
```



Dependendo do tipo de display, o `rgb` pode precisar ser alterado para `bgr`, `vrgb` ou `vbgr`: experimente e veja qual funciona melhor.

5.6. O Gerenciador de Display X

O Xorg fornece um Gerenciador de Display X, o XDM, que pode ser usado para o gerenciamento de sessões de login. O XDM fornece uma interface gráfica para escolher em qual servidor de display se conectar para inserir informações de autorização, tal como uma combinação de login e senha.

Esta seção demonstra como configurar o X Display Manager no FreeBSD. Alguns ambientes de desktop fornecem seu próprio gerenciador de login gráfico. Consulte GNOME para instruções sobre

como configurar o GNOME Display Manager e KDE para instruções sobre como configurar o KDE Display Manager.

5.6.1. Configurando o XDM

Para instalar o XDM, use o pacote ou ports x11/xdm. Uma vez instalado, o XDM pode ser configurado para ser executado quando a máquina for inicializada editando esta entrada em /etc/ttys:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

Altere o `off` para `on` e salve a edição. O `ttyv8` nesta entrada indica que o XDM será executado no nono terminal virtual.

O diretório de configuração do XDM está localizado em /usr/local/lib/X11/xdm. Esse diretório contém diversos arquivos usados para alterar o comportamento e a aparência do XDM, bem como alguns scripts e programas usados para configurar a área de trabalho quando o XDM está em execução. Arquivos de Configuração do XDM resume a função de cada um desses arquivos. A sintaxe exata e o uso desses arquivos são descritos em xdm(1).

Tabela 6. Arquivos de Configuração do XDM

Arquivo	Descrição
Xaccess	O protocolo para conectar ao XDM é chamado de X Display Manager Connection Protocol (XDMCP). Este arquivo é um conjunto de regras de autorização do cliente para controlar conexões de XDMCP de máquinas remotas. Por padrão, esse arquivo não permite a conexão de nenhum cliente remoto.
Xresources	Este arquivo controla a aparência do seletor de display XDM e das telas de login. A configuração padrão é uma janela de login retangular simples com o nome do host da máquina exibido na parte superior em uma fonte grande e "Login:" e "Senha:" solicitado abaixo. O formato deste arquivo é idêntico ao arquivo app-defaults descrito na documentação do Xorg.
Xservers	A lista de exibições locais e remotas que o seletor deve fornecer como opções de login.
Xsession	Script de sessão padrão para logins que é executado pelo XDM após um usuário realizar o login. Isso aponta para um script de sessão personalizado em ~/.xsession.

Arquivo	Descrição
Xsetup_*	Script para iniciar automaticamente os aplicativos antes de exibir as interfaces de seleção ou de login. Há um script para cada exibição sendo usada, denominada Xsetup_*, em que * é o número de exibição local. Geralmente, esses scripts executam um ou dois programas em segundo plano, como `xconsole`.
xdm-config	Configuração global para todos os monitores executados nesta máquina.
xdm-errors	Contém os erros gerados pelo programa do servidor. Se um display que o XDM está tentando iniciar travar, procure neste arquivo por mensagens de erro. Essas mensagens também são gravadas no ~/.xsession-errors do usuário.
xdm-pid	O ID do processo XDM em execução.

5.6.2. Configurando o acesso remoto

Por padrão, somente usuários no mesmo sistema podem efetuar login usando o XDM. Para permitir que os usuários em outros sistemas se conectem ao servidor de Display, edite as regras de controle de acesso e ative o listener de conexão.

Para configurar o XDM para escutar qualquer conexão remota, comente a linha `DisplayManager.requestPort` em /usr/local/etc/X11/xdm/xdm-config colocando um `!` na frente dele:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

Salve as edições e reinicie o XDM. Para restringir o acesso remoto, veja as entradas de exemplo em /usr/local/lib/X11/xdm/Xaccess e consulte xdm(1) para mais informações.

5.7. Ambientes de desktop

Esta seção descreve como instalar três ambientes de desktop populares em um sistema FreeBSD. Um ambiente de desktop pode variar de um gerenciador de janelas simples a um conjunto completo de aplicativos de desktop. Mais de cem ambientes de área de trabalho estão disponíveis na categoria x11-wm da Coleção de Ports.

5.7.1. GNOME

O GNOME é um ambiente de área de trabalho amigável. Ele inclui um painel para iniciar aplicativos e exibir status, uma área de trabalho, um conjunto de ferramentas e aplicativos e um conjunto de convenções que facilitam a cooperação entre os aplicativos e a compatibilidade entre

eles. Mais informações sobre o GNOME no FreeBSD podem ser encontradas em https://www.FreeBSD.org/gnome. Esse site contém documentação adicional sobre instalação, configuração e gerenciamento do GNOME no FreeBSD.

Este ambiente de desktop pode ser instalado a partir de um pacote binario:

```
# pkg install gnome3
```

Para instalar o GNOME a partir do ports, use o seguinte comando. O GNOME é um aplicativo grande e levará algum tempo para compilar, mesmo em um computador rápido.

```
# cd /usr/ports/x11/gnome3
# make install clean
```

O GNOME requer que o /proc seja montado. Adicione esta linha ao /etc/fstab para montar este sistema de arquivos automaticamente durante a inicialização do sistema:

```
proc           /proc       procfs  rw  0   0
```

O GNOME usa o D-Bus e o HAL para barramento de mensagens e abstração de hardware. Esses aplicativos são instalados automaticamente como dependências do GNOME. Habilite-os em /etc/rc.conf para que eles sejam iniciados quando o sistema inicializar:

```
dbus_enable="YES"
hald_enable="YES"
```

Após a instalação, configure o Xorg para iniciar o GNOME. A maneira mais fácil de fazer isso é habilitar o Gerenciador de Display do GNOME, o GDM, que é instalado como parte do pacote ou ports do GNOME. Pode ser ativado adicionando esta linha ao /etc/rc.conf:

```
gdm_enable="YES"
```

Geralmente é desejável também iniciar todos os serviços do GNOME. Para conseguir isso, adicione uma segunda linha ao /etc/rc.conf:

```
gnome_enable="YES"
```

O GDM será iniciado automaticamente quando o sistema for inicializado.

Um segundo método para iniciar o GNOME é digitar `startx` na linha de comando depois de configurar o ~/.xinitrc. Se este arquivo já existir, substitua a linha que inicia o gerenciador de janelas atual por uma que inicie o /usr/local/bin/gnome-session. Se este arquivo não existir, crie-o com este comando:

```
% echo "exec /usr/local/bin/gnome-session" > ~/.xinitrc
```

Um terceiro método é usar o XDM como o gerenciador de Display. Neste caso, crie um executável ~/.xsession:

```
% echo "exec /usr/local/bin/gnome-session" > ~/.xsession
```

5.7.2. KDE

O KDE é outro ambiente de trabalho fácil de usar. Essa área de trabalho fornece um conjunto de aplicativos com aparência e comportamento consistentes, um menu e barras de ferramentas padronizadas, atalhos de teclado, esquemas de cores, internacionalização e uma configuração de área de trabalho centralizada e orientada a diálogos. Mais informações sobre o KDE podem ser encontradas em http://www.kde.org/. Para informações específicas do FreeBSD, consulte http://freebsd.kde.org.

Para instalar o pacote KDE, digite:

```
# pkg install x11/kde5
```

Para instalar o KDE via ports, use o seguinte comando. A instalação do ports fornecerá um menu para selecionar quais componentes instalar. O KDE é um aplicativo grande e levará algum tempo para compilar, mesmo em um computador rápido.

```
# cd /usr/ports/x11/kde5
# make install clean
```

O KDE requer que o /proc esteja montado. Adicione esta linha ao /etc/fstab para montar este sistema de arquivos automaticamente durante a inicialização do sistema:

```
proc           /proc       procfs  rw  0   0
```

O KDE usa o D-Bus e o HAL para barramento de mensagens e abstração de hardware. Estas aplicações são automaticamente instaladas como dependências do KDE. Habilite-os em /etc/rc.conf para que eles sejam iniciados quando o sistema inicializar:

```
dbus_enable="YES"
hald_enable="YES"
```

Desde o KDE Plasma 5, o Gerenciador de Display do KDE, KDM, não é mais desenvolvido. Uma possível substituição é o SDDM. Para instalá-lo, digite:

```
# pkg install x11/sddm
```

Adicione esta linha em /etc/rc.conf:

```
sddm_enable="YES"
```

Um segundo método para iniciar o KDE Plasma é digitar `startx` na linha de comando. Para que isso funcione, a seguinte linha é necessária em ~/.xinitrc:

```
exec ck-launch-session startplasma-x11
```

Um terceiro método para iniciar o KDE Plasma é através do XDM. Para fazer isso, crie um arquivo executável ~/.xsession da seguinte maneira:

```
% echo "exec ck-launch-session startplasma-x11" > ~/.xsession
```

Uma vez iniciado o KDE Plasma , consulte o sistema de ajuda integrado para obter mais informações sobre como usar seus diversos menus e aplicativos.

5.7.3. Xfce

O Xfce é um ambiente de desktop baseado no kit de ferramentas GTK+ usado pelo GNOME. No entanto, é mais leve e fornece um desktop simples, eficiente e fácil de usar. É totalmente configurável, possui um painel principal com menus, applets e lançadores de aplicativos, fornece um gerenciador de arquivos e um gerenciador de som e é personalizável. Como é rápido, leve e eficiente, é ideal para máquinas mais antigas ou mais lentas com limitações de memória. Mais informações sobre o Xfce podem ser encontradas em http://www.xfce.org.

Para instalar o pacote Xfce:

```
# pkg install xfce
```

Alternativamente , para compilar o port:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

O Xfce usa o D-Bus para barramento de mensagens. Este aplicativo é instalado automaticamente como dependência do Xfce. Habilite-o em /etc/rc.conf para que ele seja iniciado quando o sistema inicializar:

```
dbus_enable="YES"
```

Ao contrário do GNOME ou KDE, o Xfce não disponibiliza seu próprio gerenciador de login. Para iniciar o Xfce à partir da linha de comando digitando `startx`, mas primeiro adicione sua entrada ao ~/.xinitrc:

```
% echo ". /usr/local/etc/xdg/xfce4/xinitrc" > ~/.xinitrc
```

Um método alternativo é usar o XDM. Para configurar este método, crie um executável ~/.xsession:

```
% echo ". /usr/local/etc/xdg/xfce4/xinitrc" > ~/.xsession
```

5.8. Instalando o Compiz Fusion

Uma maneira de tornar o uso de um computador desktop mais agradável é com bons efeitos 3D.

Instalar o pacote Compiz Fusion é fácil, mas a configuração requer alguns passos que não estão descritos na documentação do ports.

5.8.1. Configurando o Driver nVidia no FreeBSD

Os efeitos da área de trabalho podem causar uma carga considerável na placa gráfica. Para uma placa gráfica baseada na nVidia, o driver proprietário é necessário para um bom desempenho. Usuários de outras placas gráficas podem pular esta seção e continuar com a configuração do xorg.conf.

Para determinar qual o driver nVidia é necessário, consulte a Perguntas frequentes sobre o assunto.

Tendo determinado o driver correto para usar em sua placa gráfica, a instalação é tão simples quanto instalar qualquer outro pacote.

Por exemplo, para instalar o driver mais recente:

```
# pkg install x11/nvidia-driver
```

O driver irá criar um módulo do kernel, que precisa ser carregado na inicialização do sistema. Adicione a seguinte linha ao /boot/loader.conf:

```
nvidia_load="YES"
```



Para carregar imediatamente o módulo no kernel em execução, você pode executar o comando `kldload nvidia`. No entanto, foi observado que algumas versões do Xorg não funcionarão corretamente se o driver não for carregado no momento da inicialização. Desta forma, depois de editar o /boot/loader.conf, é recomendado reiniciar o sistema.

Com o módulo do kernel carregado, você normalmente só precisa alterar uma única linha no xorg.conf para habilitar o driver proprietário:

Encontre a seguinte linha no /etc/X11/xorg.conf:

```
Driver      "nv"
```

e mude para:

```
Driver      "nvidia"
```

Inicie a GUI como de costume, e você será saudado pelo splash da nVidia. Tudo deve funcionar como de costume.

5.8.2. Configurando o xorg.conf para Efeitos de Desktop

Para ativar o Compiz Fusion, o /etc/X11/xorg.conf precisa ser modificado:

Adicione a seguinte seção para habilitar os efeitos compostos:

```
Section "Extensions"
    Option         "Composite" "Enable"
EndSection
```

Localize a seção "Screen", que deve ser semelhante à abaixo:

```
Section "Screen"
    Identifier     "Screen0"
    Device         "Card0"
    Monitor        "Monitor0"
    ...
```

e adicione as duas linhas seguintes (após "Monitor"):

```
DefaultDepth    24
Option         "AddARGBGLXVisuals" "True"
```

Localize a "Subsection" que se refere à resolução da tela que você deseja usar. Por exemplo, se você deseja usar 1280x1024, localize a seção a seguir. Se a resolução desejada não aparecer em nenhuma subseção, você pode adicionar a entrada relevante à mão:

```
SubSection     "Display"
    Viewport    0 0
    Modes      "1280x1024"
```

```
EndSubSection
```

Uma profundidade de cor de 24 bits é necessária para a composição do desktop, altere a subseção acima para:

```
SubSection     "Display"
    Viewport    0 0
    Depth       24
    Modes      "1280x1024"
EndSubSection
```

Finalmente, confirme que os módulos "glx" e "extmod" estão carregados na seção "Module":

```
Section "Module"
    Load           "extmod"
    Load           "glx"
    ...
```

A configuração acima pode ser feita automaticamente com o x11/nvidia-xconfig (executando como `root`):

```
# nvidia-xconfig --add-argb-glx-visuals
# nvidia-xconfig --composite
# nvidia-xconfig --depth=24
```

5.8.3. Instalando e Configurando o Compiz Fusion

Instalar o Compiz Fusion é tão simples quanto qualquer outro pacote:

```
# pkg install x11-wm/compiz-fusion
```

Quando a instalação estiver concluída, inicie o Desktop Gráfico e, em um terminal, digite os seguintes comandos (como usuário normal):

```
% compiz --replace --sm-disable --ignore-desktop-hints ccp &
% emerald --replace &
```

Sua tela piscará por alguns segundos, pois o gerenciador de janelas (por exemplo, Metacity se você estiver usando o GNOME) será substituído pelo Compiz Fusion. O Emerald cuida das decorações da janela (isto é, botões de fechar, minimizar, maximizar, barras de título e assim por diante).

Você pode converter isso em um script trivial e executá-lo na inicialização automaticamente (por exemplo, adicionando a "Sessions" em um Desktop do GNOME):

```
#! /bin/sh
compiz --replace --sm-disable --ignore-desktop-hints ccp &
emerald --replace &
```

Salve isso no seu diretório home como, por exemplo, start-compiz e torne-o executável:

```
% chmod +x ~/start-compiz
```

Em seguida, utilize a GUI para adicioná-lo a Startup Programs (localizado em System, Preferences, Sessions em um desktop GNOME).

Para selecionar realmente todos os efeitos desejados e suas configurações, execute (novamente como um usuário normal) o Compiz Config Settings Manager:

```
% ccsm
```



No GNOME, isso também pode ser encontrado no menu System, Preferences.

Se você selecionou "gconf support" durante a compilação, você também será capaz de ver estas configurações usando o `gconf-editor` sob `apps/compiz`.

5.9. Solução de problemas

Se o mouse não funcionar, você precisará primeiro configurá-lo antes de prosseguir. Em versões recentes do Xorg, as seções `InputDevice` em xorg.conf são ignoradas em favor dos dispositivos autodetectados. Para restaurar o comportamento antigo, adicione a seguinte linha à seção `ServerLayout` ou `ServerFlags` deste arquivo:

```
Option "AutoAddDevices" "false"
```

Os dispositivos de entrada podem então ser configurados como nas versões anteriores, juntamente com quaisquer outras opções necessárias (por exemplo, troca do layout de teclado).



Como explicado anteriormente, o daemon hald irá, por padrão, detectar automaticamente o seu teclado. Há chances de que o layout ou modelo do teclado não esteja correto, ambientes de Desktop como o GNOME, KDE ou Xfce fornecem ferramentas para configurar o teclado. No entanto, é possível definir as propriedades do teclado diretamente com a ajuda do utilitário setxkbmap(1) ou com uma regra de configuração do aplicativohald.

Por exemplo, se alguém quiser usar um teclado de teclas PC 102 vindo com um layout francês, temos que criar um arquivo de configuração de teclado para o hald chamado x11-input.fdi e salva-lo no diretório /usr/local/etc/hal/fdi/policy. Este arquivo deve conter as seguintes linhas:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbModel"
type="string">pc102</merge>
      <merge key="input.x11_options.XkbLayout" type="string">fr</merge>
    </match>
  </device>
</deviceinfo>
```

Se este arquivo já existir, apenas copie e adicione ao seu arquivo as linhas referentes à configuração do teclado.

Você terá que reinicializar sua máquina para forçar o hald a ler este arquivo.

É possível fazer a mesma configuração a partir de um terminal X ou um script com esta linha de comando:

```
% setxkbmap -model pc102 -layout fr
```

O /usr/local/shared/X11/xkb/rules/base.lst lista os vários teclados, layouts e opções disponíveis.

O arquivo de configuração xorg.conf.new pode agora ser ajustado para o seu gosto. Abra o arquivo em um editor de texto, como emacs(1) ou o ee(1). Se o monitor for um modelo antigo ou incomum que não suporta a detecção automática de frequências de sincronização, essas configurações podem ser adicionadas ao xorg.conf.new na seção "Monitor":

```
Section "Monitor"
    Identifier   "Monitor0"
    VendorName   "Monitor Vendor"
    ModelName    "Monitor Model"
    HorizSync    30-107
    VertRefresh  48-120
EndSection
```

A maioria dos monitores suporta autodetecção de frequência de sincronização, tornando desnecessária a entrada manual desses valores. Para os poucos monitores que não suportam a detecção automática, evite possíveis danos inserindo apenas valores fornecidos pelo fabricante.

O X permite que os recursos do DPMS (Energy Star) sejam usados com monitores capazes. O programa xset(1) controla os tempos limite e pode forçar os modos de espera, suspensão ou desativação. Se você deseja habilitar recursos de DPMS para o seu monitor, você deve adicionar a seguinte linha à seção do monitor:

```
Option      "DPMS"
```

Enquanto o arquivo de configuração xorg.conf.new ainda estiver aberto em um editor, selecione a resolução padrão e a profundidade de cor desejada. Isso é definido na seção `"Screen"`:

```
Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    Monitor    "Monitor0"
    DefaultDepth 24
    SubSection "Display"
        Viewport  0 0
        Depth     24
        Modes     "1024x768"
    EndSubSection
EndSection
```

A palavra-chave `DefaultDepth` descreve a profundidade de cor a ser executada por padrão. Isto pode ser sobrescrito com a opção de linha de comando `-depth` para Xorg(1). A palavra-chave `Modes` descreve a resolução a ser executada na profundidade de cor especificada. Observe que somente os modos padrão VESA são suportados, conforme definido pelo hardware gráfico do sistema de destino. No exemplo acima, a profundidade de cor padrão é de vinte e quatro bits por pixel. Nesta profundidade de cor, a resolução aceita é 1024 por 768 pixels.

Finalmente, escreva o arquivo de configuração e teste-o usando o modo de teste dado acima.



Uma das ferramentas disponíveis para ajudá-lo durante o processo de solução de problemas são os arquivos de log do Xorg, que contêm informações sobre cada dispositivo ao qual o servidor Xorg se conecta. Os nomes de arquivos de log do Xorg estão no formato /var/log/Xorg.0.log. O nome exato do log pode variar de Xorg.0.log para Xorg.8.log e assim por diante.

Se tudo estiver bem, o arquivo de configuração precisa ser instalado em um local comum onde o Xorg(1) possa encontrá-lo. Isto é tipicamente /etc/X11/xorg.conf ou /usr/local/etc/X11/xorg.conf.

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

O processo de configuração do Xorg agora está completo. O Xorg pode agora ser iniciado com o utilitário startx(1). O servidor Xorg também pode ser iniciado com o uso de xdm(1).

5.9.1. Configuração com Chipsets gráficos Intel™ i810

A configuração com chipsets integrados i810 da Intel™ requer a interface de programação AGP agpgart para o Xorg para conduzir a placa. Consulte a página de manual do driver agp(4) para obter maiores informações.

Isso permitirá a configuração do hardware como qualquer outra placa gráfica. Observe que nos sistemas sem o driver agp(4) compilado no kernel, tentar carregar o módulo com kldload(8) não funcionará. Este driver tem que estar no kernel no momento da inicialização, através da compilação ou usando o /boot/loader.conf.

5.9.2. Adicionando um Flatpanel Widescreen ao Mix

Esta seção pressupõe um pouco de conhecimento avançado de configuração. Se as tentativas de usar as ferramentas de configuração padrão acima não resultaram em uma configuração funcional, há informações suficientes nos arquivos de log para serem úteis para fazer a configuração funcionar. O uso de um editor de texto será necessário.

Os formatos widescreen atuais (WSXGA, WSXGA+, WUXGA, WXGA, WXGA+, etc.) suportam formatos ou proporções de formato 16:10 e 10:9 que podem ser problemáticos. Exemplos de algumas resoluções de tela comuns para proporções de 16:10 são:

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

Em algum momento, será tão fácil quanto adicionar uma dessas resoluções como um possível `Mode` na `Section "Screen"` como tal:

```
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
    Viewport  0 0
    Depth     24
    Modes     "1680x1050"
EndSubSection
EndSection
```

O Xorg é inteligente o suficiente para extrair as informações de resolução da tela widescreen via informações I2C/DDC, para que ele saiba o que o monitor pode suportar em termos de freqüências e resoluções.

Se aqueles `ModeLines` não existem nos drivers, pode ser necessário dar ao Xorg uma pequena dica. Usando o /var/log/Xorg.0.log pode-se extrair informações suficientes para criar manualmente um `ModeLine` que funcionará. Basta procurar informações semelhantes:

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
```

```
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border:
0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border:
0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max
170 MHz
```

Esta informação é chamada de informação EDID. Criar uma `ModeLine` a partir disso é apenas uma questão de colocar os números na ordem correta:

```
ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

Assim, o `ModeLine` na `Section "Monitor"` para este exemplo ficaria assim:

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

Agora, tendo completado estes passos simples de edição, o X deve iniciar no seu novo monitor widescreen.

5.9.3. Solução de problemas do Compiz Fusion

5.9.3.1. Eu instalei o Compiz Fusion, e depois de executar os comandos que você mencionou, minhas janelas ficaram sem barras de título e botões. O que está errado?

Provavelmente está faltando alguma configuração em /etc/X11/xorg.conf. Revise este arquivo cuidadosamente e verifique especialmente as diretivas `DefaultDepth` e `AddARGBGLXVisuals`.

5.9.3.2. Quando executo o comando para iniciar o Compiz Fusion, o servidor X trava e eu volto ao console. O que está errado?

Se você verificar o /var/log/Xorg.0.log, você provavelmente encontrará mensagens de erro durante a inicialização do X. As mais comuns seriam:

```
(EE) NVIDIA(0):     Failed to initialize the GLX module; please check in your X
(EE) NVIDIA(0):     log file that the GLX module has been loaded in your X
(EE) NVIDIA(0):     server, and that the module is the NVIDIA GLX module.  If
(EE) NVIDIA(0):     you continue to encounter problems, Please try
(EE) NVIDIA(0):     reinstalling the NVIDIA driver.
```

Este é geralmente o caso quando você atualiza o Xorg. Você precisará reinstalar o pacote

x11/nvidia-driver para que o glx seja compilado novamente.

Parte II: Tarefas comuns

Agora que o básico foi abordado, esta parte do livro discute alguns recursos freqüentemente usados do FreeBSD. Estes capítulos:

- Introduzem aplicativos de desktop populares e úteis: navegadores, ferramentas de produtividade, visualizadores de documentos e muito mais.
- Introduzem uma série de ferramentas multimídia disponíveis para o FreeBSD.
- Explicam o processo de compilação de um kernel customizado do FreeBSD para habilitar funcionalidades extras.
- Descrevem o sistema de impressão em detalhes, tanto configurações de impressoras conectadas em desktops quanto impressoras conectadas à rede.
- Mostram como executar aplicativos Linux no sistema FreeBSD.

Alguns destes capítulos recomendam leituras prévias, e isso é destacado na sinopse no início de cada capítulo.

Capítulo 6. Aplicações de Desktop

6.1. Sinopse

Embora o FreeBSD seja popular como um servidor por seu desempenho e estabilidade, ele também é adequado para o uso diário como desktop. Com mais de 24.000 aplicativos disponíveis como pacotes ou ports para o FreeBSD, é fácil construir um desktop personalizado que executa uma ampla variedade de aplicativos de desktop. Este capítulo demonstra como instalar vários aplicativos de desktop, incluindo navegadores da Web, software de produtividade, visualizadores de documentos e softwares financeiros.



Os usuários que preferem instalar uma versão de desktop pré-configurada do FreeBSD em vez de configurar um do zero devem consultar FuryBSD., GhostBSD or MidnightBSD.

Os leitores deste capítulo devem saber como:

- Instalar software adicional usando pacotes ou ports, conforme descrito em Instalando Aplicativos. Pacotes e Ports.
- Instalar o X e um gerenciador de janelas, conforme descrito em O sistema X Window.

Para obter informações sobre como configurar um ambiente multimídia, consulte Multimídia.

6.2. Navegadores

O FreeBSD não vem com um navegador Web pré-instalado. Em vez disso, a categoria www da Coleção de Ports contém muitos navegadores que podem ser instalados como um pacote ou compilados a partir da coleção de Ports.

Os ambientes de área de trabalho KDE e GNOME incluem seu próprio navegador HTML. Consulte Ambientes de desktop para mais informações sobre como configurar esses desktops completos.

Alguns navegadores leves incluem o www/dillo2, o www/links e o www/w3m.

Esta seção demonstra como instalar os seguintes navegadores Web populares e indica se o aplicativo é pesado em recursos, se leva tempo para compilar a partir do Ports ou se possui dependências importantes.

Nome da aplicação	Recursos necessários	Instalação a partir do Ports	Notas
Firefox	médio	pesado	FreeBSD, Linux™, e versões localizadas estão disponíveis
Konqueror	médio	pesado	Requer bibliotecas do KDE
Chromium	médio	pesado	Requer Gtk+

6.2.1. Firefox

O Firefox é um navegador de código-fonte aberto que apresenta um mecanismo de exibição HTML compatível com os padrões, navegação por guias, bloqueio de pop-up, extensões, segurança aprimorada e muito mais. O Firefox é baseado na base de código Mozilla.

Para instalar o pacote da versão mais recente do Firefox, digite:

```
# pkg install firefox
```

Para instalar a versão ESR (Extended Support Release) do Firefox, use:

```
# pkg install firefox-esr
```

A Coleção de Ports pode ser usada para compilar a versão desejada do Firefox a partir do código-fonte. Este exemplo compila o www/firefox, onde o `firefox` pode ser substituído pelo ESR ou pela versão localizada para instalar.

```
# cd /usr/ports/www/firefox
# make install clean
```

6.2.2. Konqueror

O Konqueror é mais do que um navegador Web, pois também é um gerenciador de arquivos e um visualizador de multimídia. Suporta WebKit assim como seu próprio KHTML. WebKit é um motor de renderização usado por diversos navegadores modernos incluiindo o Chromium.

O Konqueror pode ser instalado como um pacote digitando:

```
# pkg install konqueror
```

Para instalar a partir da Coleção de Ports:

```
# cd /usr/ports/x11-fm/konqueror/
# make install clean
```

6.2.3. Chromium

O Chromium é um projeto de navegador de código aberto que visa criar uma experiência de navegação na Web mais segura, mais rápida e mais estável. O Chromium apresenta navegação com guias, bloqueio de pop-up, extensões e muito mais. O Chromium é o projeto de código-fonte aberto no qual o navegador Web do Google Chrome é baseado.

O Chromium pode ser instalado como um pacote digitando:

```
# pkg install chromium
```

Alternativamente, o Chromium pode ser compilado a partir do código-fonte usando a Coleção de Ports:

```
# cd /usr/ports/www/chromium
# make install clean
```



O executável do Chromium é /usr/local/bin/chrome, não /usr/local/bin/chromium.

6.3. Produtividade

Quando se trata de produtividade, os usuários geralmente procuram uma suíte de escritório ou um processador de texto fácil de usar. Embora alguns ambientes de desktop como o KDE forneçam uma suíte de escritório, não há um pacote de produtividade padrão. Várias suítes de escritório e processadores de texto gráficos estão disponíveis para o FreeBSD, independentemente do gerenciador de janelas instalado.

Esta seção demonstra como instalar os seguintes softwares populares de produtividade e indica se o aplicativo é pesado em recursos, se leva tempo para compilar a partir do ports ou se possui dependências importantes.

Nome da aplicação	Recursos necessários	Instalação a partir do Ports	Principais Dependências
Calligra	leve	pesado	KDE
AbiWord	leve	leve	Gtk+ ou GNOME
The Gimp	leve	pesado	Gtk+
Apache OpenOffice	pesado	enorme	JDK™ e Mozilla
LibreOffice	um pouco pesado	enorme	Gtk+, ou KDE/ GNOME, ou JDK™

6.3.1. Calligra

O ambiente de área de trabalho do KDE inclui uma suíte de escritório que pode ser instalada separadamente do KDE. O Calligra inclui componentes padrões que podem ser encontrados em outros pacotes de escritório. O Words é o processador de texto, Sheets é o programa de planilha eletrônica, o Stage gerencia apresentações de slides e Karbon é usado para desenhar documentos gráficos.

No FreeBSD, o editors/calligra pode ser instalado como um pacote ou um port. Para instalar o pacote:

```
# pkg install calligra
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/editors/calligra
# make install clean
```

6.3.2. AbiWord

O AbiWord é um programa gratuito de processamento de texto semelhante em aparência ao Microsoft™ Word. É rápido, contém muitos recursos e é de fácil utilização.

O AbiWord pode importar ou exportar muitos formatos de arquivo, incluindo alguns formatos proprietários como o Microsoft™.rtf.

Para instalar o pacote do AbiWord:

```
# pkg install abiword
```

Se o pacote não estiver disponível, ele pode ser compilado a partir da Coleção de Ports:

```
# cd /usr/ports/editors/abiword
# make install clean
```

6.3.3. O GIMP

Para autoria ou retoque de imagens, o GIMP fornece um sofisticado programa de manipulação de imagens. Ele pode ser usado como um programa de pintura simples ou como um pacote de qualidade para retoque de fotos. Ele suporta um grande número de plugins e possui uma interface de script. O GIMP pode ler e gravar uma grande variedade de formatos de arquivos e suporta interfaces com scanners e tablets.

Para instalar o pacote:

```
# pkg install gimp
```

Como alternativa, use a Coleção de Ports:

```
# cd /usr/ports/graphics/gimp
# make install clean
```

A categoria de programas gráficos (freebsd.org/ports/) da Coleção de Ports contém vários plugins relacionados ao GIMP, arquivos de ajuda e manuais do usuário.

6.3.4. Apache OpenOffice

O Apache OpenOffice é uma suíte de escritório de código-fonte aberto que é desenvolvida sob a asa da Incubadora da Apache Software Foundation. Ele inclui todos os aplicativos encontrados em um pacote completo de produtividade de escritório: um processador de texto, uma planilha eletrônica, um gerenciador de apresentação e um programa de desenho. Sua interface de usuário é semelhante a outros pacotes de escritório e pode importar e exportar em vários formatos de arquivo populares. Está disponível em vários idiomas diferentes e a internacionalização foi estendida para interfaces, corretores ortográficos e dicionários.

O processador de texto do Apache OpenOffice usa um formato de arquivo XML nativo para maior portabilidade e flexibilidade. O programa de planilha eletrônica possui uma linguagem de macros que pode ser conectada a bancos de dados externos. O Apache OpenOffice é estável e roda nativamente em Windows™, Solaris™, Linux™, FreeBSD, e Mac OS™ X. Maiores informações sobre o Apache OpenOffice podem ser encontradas em openoffice.org. Para informações específicas do FreeBSD, consulte porting.openoffice.org/freebsd/.

Para instalar o pacote Apache OpenOffice:

```
# pkg install apache-openoffice
```

Depois que o pacote for instalado, digite o seguinte comando para iniciar o Apache OpenOffice:

```
% openoffice-X.Y.Z
```

onde *X.Y.Z* é o número da versão instalada do Apache OpenOffice. Na primeira vez que o Apache OpenOffice for iniciado, algumas perguntas serão feitas e uma pasta .openoffice.org será criada no diretório pessoal do usuário.

Se o pacote do Apache OpenOffice desejado não estiver disponível, a compilação do port ainda será uma opção. No entanto, isso requer muito espaço em disco e um tempo bastante longo para compilar:

```
# cd /usr/ports/editors/openoffice-4
# make install clean
```



Para compilar uma versão localizada, substitua o comando anterior por:

```
# make LOCALIZED_LANG=your_language install clean
```

Substitua *your_language* pelo código ISO do idioma correto. Uma lista de códigos de idiomas suportados está disponível em files/Makefile.localized, localizado no diretório do port.

6.3.5. LibreOffice

O LibreOffice é um pacote de software livre desenvolvido por documentfoundation.org. É compatível com outras grandes suítes de escritórios e está disponível em diversas plataformas. Ele é um fork renomeado do Apache OpenOffice e inclui aplicativos encontrados em um pacote completo de produtividade de escritório: processador de texto, planilha, gerenciador de apresentação, programa de desenho, programa de gerenciamento de banco de dados e uma ferramenta para criar e editar fórmulas matemáticas. Está disponível em vários idiomas diferentes e a internacionalização foi estendida para interfaces, corretores ortográficos e dicionários.

O processador de texto do LibreOffice usa um formato de arquivo XML nativo para maior portabilidade e flexibilidade. O programa de planilha eletrônica possui uma linguagem de macros que pode ser conectada a bancos de dados externos. O LibreOffice é estável e roda nativamente em Windows™, Linux™, FreeBSD e Mac OS™ X. Maiores informações sobre o LibreOffice podem ser encontradas em libreoffice.org.

Para instalar a versão em inglês do pacote LibreOffice:

```
# pkg install libreoffice
```

A categoria de editores de texto (freebsd.org/ports/) da Coleção de Ports contém várias versões versões localizadas do LibreOffice. Ao instalar um pacote localizado, substitua `libreoffice` pelo nome do pacote localizado.

Quando o pacote estiver instalado, digite o seguinte comando para executar o LibreOffice:

```
% libreoffice
```

Durante a primeira execução, algumas perguntas serão feitas e uma pasta .libreoffice será criada no diretório pessoal do usuário.

Se o pacote LibreOffice desejado não estiver disponível, a compilação do port ainda será uma opção. No entanto, isso requer muito espaço em disco e um tempo bastante longo para compilar. Este exemplo compila a versão em inglês:

```
# cd /usr/ports/editors/libreoffice
# make install clean
```



Para compilar uma versão localizada, faça `cd` para o diretório do port do idioma desejado. Os idiomas suportados podem ser encontrados na categoria de editores (freebsd.org/ports/) da Coleção de Ports.

6.4. Visualizadores de Documentos

Alguns novos formatos de documentos ganharam popularidade desde o advento do UNIX™ e os

visualizadores que eles exigem podem não estar disponíveis no sistema base. Esta seção demonstra como instalar os seguintes visualizadores de documentos:

Nome da aplicação	Recursos necessários	Instalação a partir do Ports	Principais Dependências
Xpdf	leve	leve	FreeType
gv	leve	leve	Xaw3d
Geeqie	leve	leve	Gtk+ ou GNOME
ePDFView	leve	leve	Gtk+
Okular	leve	pesado	KDE

6.4.1. Xpdf

Para os usuários que preferem um pequeno visualizador de PDF do FreeBSD, o Xpdf fornece um visualizador leve e eficiente que requer poucos recursos. Ele usa as fontes X padrão e não requer nenhum kit de ferramentas adicional.

Para instalar o pacote Xpdf:

```
# pkg install xpdf
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

Quando a instalação estiver concluída, inicie o `xpdf` e use o botão direito do mouse para ativar o menu.

6.4.2. gv

O gv é um visualizador de arquivos PostScript™ e PDF. Ele é baseado no ghostview, mas tem uma aparência mais agradável, pois é baseado no kit de ferramentas do widget Xaw3d. O gv possui muitos recursos configuráveis, como orientação, tamanho do papel, escala e anti-aliasing. Quase qualquer operação pode ser executada com o teclado ou com o mouse.

Para instalar o gv como um pacote:

```
# pkg install gv
```

Se um pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/print/gv
```

```
# make install clean
```

6.4.3. Geeqie

O Geeqie é um fork do projeto abandonado GQView, em um esforço para levar o desenvolvimento adiante e integrar os patches existentes. O Geeqie é um gerenciador de imagens que suporta a visualização de um arquivo com um único clique, a execução de um editor externo e a visualização de miniaturas. Ele também possui um modo de apresentação de slides e algumas operações básicas de arquivo, facilitando o gerenciamento das coleções de imagens e a localização de arquivos duplicados. O Geeqie suporta visualização em tela cheia e a internacionalização.

Para instalar o pacote Geeqie:

```
# pkg install geeqie
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/graphics/geeqie
# make install clean
```

6.4.4. ePDFView

O ePDFView é um visualizador de documentos PDF leve que usa somente as bibliotecas Gtk+ e Poppler. Ele está atualmente em desenvolvimento, mas já abre a maioria dos arquivos PDF (até os criptografados), salva cópias de documentos e tem suporte para impressão usando o CUPS.

Para instalar o ePDFView como um pacote:

```
# pkg install epdfview
```

Se um pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/graphics/epdfview
# make install clean
```

6.4.5. Okular

O Okular é um visualizador de documentos universal baseado no KPDF para KDE. Ele pode abrir muitos formatos de documentos, incluindo PDF, PostScript™, DjVu, CHM, XPS e ePub.

Para instalar o Okular como um pacote:

```
# pkg install okular
```

Se um pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/graphics/okular
# make install clean
```

6.5. Finanças

Para gerenciar finanças pessoais em um desktop FreeBSD, alguns aplicativos poderosos e fáceis de usar podem ser instalados. Alguns são compatíveis com formatos de arquivos comuns, como os formatos usados pelo Quicken e Excel.

Esta seção cobre estes programas:

Nome da aplicação	Recursos necessários	Instalação a partir do Ports	Principais Dependências
GnuCash	leve	pesado	GNOME
Gnumeric	leve	pesado	GNOME
KMyMoney	leve	pesado	KDE

6.5.1. GnuCash

O GnuCash faz parte do esforço do GNOME para fornecer aplicativos fáceis de usar, mas poderosos, para usuários finais. O GnuCash pode ser usado para acompanhar receitas e despesas, contas bancárias e ações. Ele apresenta uma interface intuitiva, mantendo-se profissional.

O GnuCash fornece um registro inteligente, um sistema hierárquico de contas e muitos aceleradores de teclado e métodos de preenchimento automático. Ele pode dividir uma única transação em várias partes mais detalhadas. O GnuCash pode importar e mesclar arquivos QIF do Quicken. Ele também lida com a maioria dos formatos internacionais de data e moeda.

Para instalar o pacote GnuCash:

```
# pkg install gnucash
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

6.5.2. Gnumeric

O Gnumeric é um programa de planilha eletrônica desenvolvido pela comunidade GNOME. Ele possui adivinhação automática e conveniente de entrada do usuário de acordo com o formato da célula para muitas sequências. Ele pode importar arquivos em vários formatos populares,

incluindo Excel, Lotus 1-2-3 e Quattro Pro. Ele tem um grande número de funções internas e permite todos os formatos usuais de célula, como número, moeda, data, hora e muito mais.

Para instalar o Gnumeric como um pacote:

```
# pkg install gnumeric
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

6.5.3. KMyMoney

O KMyMoney é uma aplicação de finanças pessoais criada pela comunidade KDE. O KMyMoney tem como objetivo fornecer os recursos importantes encontrados em aplicativos comerciais de gerenciamento de finanças pessoais. Ele também destaca a facilidade de uso e a contabilidade adequada de dupla entrada entre seus recursos. O KMyMoney importa a partir de arquivos QIF padrão do Quicken, rastreia investimentos, manipula várias moedas e fornece diversos relatórios.

Para instalar o KMyMoney como um pacote:

```
# pkg install kmymoney-kde4
```

Se o pacote não estiver disponível, use a Coleção de Ports:

```
# cd /usr/ports/finance/kmymoney-kde4
# make install clean
```

Capítulo 7. Multimídia

7.1. Sinopse

O FreeBSD suporta uma ampla variedade de placas de som, permitindo que os usuários aproveitem a saída de alta fidelidade de um sistema FreeBSD. Isso inclui a capacidade de gravar e reproduzir áudio MPEG Layer 3 (MP3), arquivo de áudio Waveform (WAV), Ogg Vorbis e outros formatos. A coleção de Ports do FreeBSD contém muitas aplicações para editar áudio gravado, adicionar efeitos sonoros e controlar dispositivos MIDI conectados.

O FreeBSD também suporta a reprodução de arquivos de vídeo e DVDs. A coleção de Ports do FreeBSD contém aplicativos para codificar, converter e reproduzir várias mídias de vídeo.

Este capítulo descreve como configurar placas de som, reprodução de vídeo, placas sintonizadoras de TV e scanners no FreeBSD. Também descreve algumas das aplicações que estão disponíveis para usar esses dispositivos.

Depois de ler este capítulo, você irá saber como:

- Configurar uma placa de som no FreeBSD.
- Solucionar problemas de configuração de som.
- Reproduzir e codificar MP3 e outros áudios.
- Preparar um sistema FreeBSD para reprodução de vídeo.
- Reproduzir DVDs, arquivos .mpg e .avi.
- Copiar o conteúdo de um CD ou DVD em arquivos arquivos.
- Configurar uma placa de TV.
- Instale e configure o MythTV no FreeBSD
- Configurar um scanner de imagem.
- Configurar um headset Bluetooth.

Antes de ler este capítulo, você deve:

- Saber como instalar aplicativos conforme descrito em Instalando Aplicativos. Pacotes e Ports.

7.2. Configurando a Placa de Som

Antes de iniciar a configuração, determine o modelo da placa de som e o chip usado. O FreeBSD suporta uma ampla variedade de placas de som. Verifique a lista de dispositivos de áudio compatíveis nas Notas de Hardware, para ver se a placa de som é suportada e quais drivers do FreeBSD que ela usa.

Para usar um dispositivo de som, seu driver deve ser carregado. A maneira mais fácil é carregar o módulo do kernel para a placa de som com o kldload(8). Este exemplo carrega o driver para um chipset de áudio integrado baseado na especificação Intel:

```
# kldload snd_hda
```

Para automatizar o carregamento desse driver no momento da inicialização, faça edição adicionando a seguinte linha ao arquivo /boot/loader.conf:

```
snd_hda_load="YES"
```

Outros módulos de som disponíveis estão listados no arquivo /boot/defaults/loader.conf. Quando não tiver certeza de qual driver usar, carregue o módulo snd_driver:

```
# kldload snd_driver
```

Este é um metadriver que carrega todos os drivers de som mais comuns e pode ser usado para acelerar a busca pelo driver correto. Também é possível carregar todos os drivers de som adicionando o metadriver no arquivo /boot/loader.conf.

Para determinar qual driver foi selecionado para a placa de som após carregar o metadriver snd_driver, digite, `cat /dev/sndstat`.

7.2.1. Configurando um kernel Personalizado com Suporte de Som

Esta seção é para usuários que preferem compilar estaticamente em suporte para a placa de som em um kernel personalizado. Para mais informações sobre como recompilar um kernel, consulte Configurando o kernel do FreeBSD.

Ao usar um kernel personalizado para fornecer suporte ao som, verifique se o driver do framework de áudio existe no arquivo de configuração do kernel personalizado:

```
device sound
```

Em seguida, adicione suporte para a placa de som. Para continuar o exemplo do chipset de áudio integrado baseado na especificação Intel da seção anterior, use a seguinte linha no arquivo de configuração do kernel personalizado:

```
device snd_hda
```

Certifique-se de ler a página de manual do driver para o nome do dispositivo a ser usado pelo driver.

Placas de som ISA não-PnP podem requerer que as configurações de porta IRQ e I/O da placa sejam adicionadas ao arquivo /boot/device.hints. Durante o processo de inicialização, o loader(8) lê este arquivo e passa as configurações para o kernel. Por exemplo, uma placa antiga ISA não-PnP da Creative SoundBlaster™ usará o driver snd_sbc(4) em conjunto com `snd_sb16`. Para esta placa, as seguintes linhas devem ser adicionadas ao arquivo de configuração do kernel:

```
device snd_sbc
device snd_sb16
```

Se a placa usar a porta de I/O `0x220` e a IRQ `5`, essas linhas também deverão ser adicionadas ao arquivo /boot/device.hints:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

A sintaxe usada no arquivo /boot/device.hints é descrita em sound(4) e na página de manual do driver da placa de som.

As configurações mostradas acima são os padrões. Em alguns casos, a IRQ ou outras configurações podem precisar ser alterados para corresponder à placa. Consulte snd_sbc(4) para obter mais informações sobre esta placa.

7.2.2. Testando o Som

Depois de carregar o módulo necessário ou reinicializar no kernel personalizado, a placa de som deve ser detectada. Para confirmar, execute `dmesg | grep pcm`. Este exemplo é de um sistema com um chipset integrado Conexant CX20590:

```
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 5 on hdaa0
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 6 on hdaa0
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> at nid 31,25 and 35,27 on hdaa1
```

O status da placa de som também pode ser verificado usando este comando:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm: 64bit 2009061500/amd64)
Installed devices:
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> (play/rec) default
```

A saída irá variar dependendo da placa de som. Se nenhum dispositivo pcm estiver listado, verifique se o driver de dispositivo correto foi carregado ou compilado no kernel. A próxima seção lista alguns problemas comuns e suas soluções.

Se tudo correr bem, a placa de som deverá funcionar no FreeBSD. Se a unidade de CD ou DVD estiver corretamente conectada à placa de som, é possível inserir um CD de áudio na unidade e reproduzi-lo com cdcontrol(1):

```
% cdcontrol -f /dev/acd0 play 1
```



CD de audio têm codificações especializadas, o que significa que não devem ser montados usando mount(8).

Várias aplicações, como audio/workman, fornecem uma interface mais amigável. O Port audio/mpg123 pode ser instalado para ouvir arquivos de áudio MP3.

Outra maneira rápida de testar a placa é enviar dados para /dev/dsp:

```
% cat filename > /dev/dsp
```

onde filename pode ser qualquer tipo de arquivo. Este comando deve produzir algum ruído, confirmando que a placa de som está funcionando.



Os nós de dispositivo /dev/dsp* serão criados automaticamente conforme necessário. Quando não estão em uso, eles não existem e não aparecerão na saída de ls(1).

7.2.3. Configurando Dispositivos de Som Bluetooth

Conectar a um dispositivo Bluetooth está fora do escopo deste capítulo. Consulte a Bluetooth para mais informações.

Para que o dispositivo Bluetooth funcione com o sistema de som do FreeBSD, os usuários precisam primeiramente instalar o audio/virtual_oss:

```
# pkg install virtual_oss
```

audio/virtual_oss requer `cuse` para ser carregado no kernel:

```
# kldload cuse
```

Para carregar o `cuse` durante a inicialização do sistema, execute o comando:

```
# sysrc -f /boot/loader.conf cuse_load=yes
```

Para usar fones de ouvido como reprodutor de som com audio/virtual_oss, os usuários precisam criar um dispositivo virtual depois de se conectarem a um dispositivo de áudio Bluetooth:

```
# virtual_oss -C 2 -c 2 -r 48000 -b 16 -s 768 -R /dev/null -P
/dev/bluetooth/headphones -d dsp
```



headphones neste exemplo é o nome de host de /etc/bluetooth/hosts. `BT_ADDR` também poderia ser usado.

Consulte virtual_oss(8) para mais informações.

7.2.4. Solução de Problemas de Som

Mensagens de Erros Comuns lista algumas mensagens de erros comuns e suas soluções:

Tabela 7. Mensagens de Erros Comuns

Erro	Solução
`sb_dspwr(XX) timed out`	A porta de I/O não está configurada corretamente.
`bad irq XX`	A IRQ está definida incorretamente. Certifique-se de que a IRQ definido e a IRQ do som são as mesmas.
`xxx: gus pcm not attached, out of memory`	Não há memória disponível suficiente para usar o dispositivo.
`xxx: can't open /dev/dsp!`	Digite `fstat | grep dsp` para verificar se outro aplicativo está mantendo o dispositivo aberto. Os causadores de problemas notáveis são o suporte a som do esound e do KDE.

Placas gráficas modernas geralmente vêm com seu próprio driver de som para uso com HDMI. Às vezes, esse dispositivo de som é enumerado antes da placa de som, o que significa que a placa de som não será usada como o dispositivo de reprodução padrão. Para verificar se este é o caso, execute dmesg e procure por `pcm`. A saída é algo como isto:

```
...
hdac0: HDA Driver Revision: 20100226_0142
hdac1: HDA Driver Revision: 20100226_0142
hdac0: HDA Codec #0: NVidia (Unknown)
hdac0: HDA Codec #1: NVidia (Unknown)
hdac0: HDA Codec #2: NVidia (Unknown)
hdac0: HDA Codec #3: NVidia (Unknown)
pcm0: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 0 nid 1 on hdac0
pcm1: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 1 nid 1 on hdac0
pcm2: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 2 nid 1 on hdac0
pcm3: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 3 nid 1 on hdac0
hdac1: HDA Codec #2: Realtek ALC889
pcm4: <HDA Realtek ALC889 PCM #0 Analog> at cad 2 nid 1 on hdac1
pcm5: <HDA Realtek ALC889 PCM #1 Analog> at cad 2 nid 1 on hdac1
pcm6: <HDA Realtek ALC889 PCM #2 Digital> at cad 2 nid 1 on hdac1
pcm7: <HDA Realtek ALC889 PCM #3 Digital> at cad 2 nid 1 on hdac1
...
```

Neste exemplo, a placa gráfica (NVidia) foi enumerada antes da placa de som (Realtek ALC889). Para usar a placa de som como o dispositivo de reprodução padrão, altere a variável hw.snd.default_unit para a unidade que deve ser usada para reprodução:

```
# sysctl hw.snd.default_unit=n
```

onde n é o número do dispositivo de som a ser usado. Neste exemplo, deve ser 4. Deixe esta mudança permanente adicionando a seguinte linha ao arquivo /etc/sysctl.conf:

```
hw.snd.default_unit=4
```

7.2.5. Utilizando Múltiplas Fontes de Som

Muitas vezes é desejável ter várias fontes de som capazes de tocar simultaneamente. O FreeBSD usa "Canais de Som Virtuais" para multiplexar a reprodução da placa de som mixando o som no kernel.

Três variáveis no sysctl(8) estão disponíveis para configurar canais virtuais:

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

Este exemplo aloca quatro canais virtuais, que é um número prático para o uso diário. Ambos dev.pcm.0.play.vchans=4 e dev.pcm.0.rec.vchans=4 são configuráveis depois que um dispositivo foi anexado e representa o número de canais virtuais pcm0 para reprodução e gravação. Como o módulo pcm pode ser carregado independentemente dos drivers de hardware, hw.snd.maxautovchans indica quantos canais virtuais serão dados a um dispositivo de áudio quando ele estiver conectado. Consulte pcm(4) para obter mais informações.



O número de canais virtuais para um dispositivo não pode ser alterado enquanto estiver em uso. Primeiramente, feche todos os programas usando o dispositivo, como players de música ou daemons de som.

O dispositivo pcm correto será automaticamente alocado de forma transparente para um programa que solicite /dev/dsp0.

7.2.6. Configurando Valores Padrões para Canais de Mixer

Os valores padrões para os diferentes canais do mixer são codificados permanentemente no código-fonte do driver pcm(4). Embora os níveis do mixer da placa de som possam ser alterados usando mixer(8) ou aplicativos e daemons de terceiros, essa não é uma solução permanente. Para definir os valores padrões do mixer no nível do driver, defina os valores apropriados no arquivo /boot/device.hints, conforme mostrado neste exemplo:

```
hint.pcm.0.vol="50"
```

Isso definirá o canal de volume como um valor padrão de `50` quando o módulo pcm(4) for carregado.

7.3. Áudio MP3

Esta seção descreve alguns players MP3 disponíveis para o FreeBSD, como ripar trilhas de CD de áudio e como codificar e decodificar MP3.

7.3.1. Players de MP3

Um popular reprodutor gráfico de MP3 é o Audacious. Ele suporta skins do Winamp e plugins adicionais. A interface é intuitiva, com uma lista de reprodução, equalizador gráfico e muito mais. Para aqueles que estão familiarizados com o Winamp, acharão o Audacious simples de usar. No FreeBSD, o Audacious pode ser instalado a partir de pacotes ou coleção de Ports multimedia/audacious. Audacious é descendente do XMMS.

O pacote ou Port audio/mpg123 fornece um reprodutor de MP3 alternativo em linha de comando. Uma vez instalado, especifique o arquivo MP3 para reproduzir na linha de comando. Se o sistema tiver vários dispositivos de áudio, o dispositivo de som também pode ser especificado:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
        version 1.18.1; written and copyright by Michael Hipp and others
        free software (LGPL) without any warranty but with best wishes

Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

Outros players MP3 estão disponíveis na Coleção de Ports do FreeBSD.

7.3.2. Copiando Trilhas de CD de Áudio

Antes de codificar um CD ou CD para MP3, os dados de áudio no CD devem ser copiados para o disco rígido. Isso é feito copiando os dados brutos do CD Digital Audio (CDDA) como arquivos WAV.

A ferramenta `cdda2wav`, que é instalada com o suite de programas sysutils/cdrtools, pode ser usada para extrair informações de áudio do CD.

Com o CD de áudio na unidade, o seguinte comando pode ser executado como `root` para ripar um CD inteiro em arquivos WAV, por trilhas:

```
# cdda2wav -D 0,1,0 -B
```

Neste exemplo, o `-D 0,1,0` indica o dispositivo SCSI 0,1,0 contendo o CD para ripar. Use o comando

`cdrecord -scanbus` para determinar os parâmetros corretos do dispositivo para o sistema.

Para ripar trilhas individuais, use a opção `-t` para especificar a trilha:

```
# cdda2wav -D 0,1,0 -t 7
```

Para extrair um intervalo de trilhas, como as trilhas de um a sete, especifique um intervalo:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Para extrair de uma unidade ATAPI(IDE) CDROM, especifique o nome do dispositivo no lugar dos números da unidade SCSI. Por exemplo, para extrair a trilha 7 de uma unidade IDE:

```
# cdda2wav -D /dev/acd0 -t 7
```

Alternativamente, o comando `dd` pode ser usado para extrair trilhas de áudio em unidades ATAPI, conforme descrito em Duplicando CDs de Áudio.

7.3.3. Codificação e Decodificação de MP3

Lame é um codificador popular para MP3 que pode ser instalado a partir do port audio/lame. Devido a problemas de patente, o pacote não está disponível.

O comando a seguir converterá o arquivo WAV audio01.wav para audio01.mp3:

```
# lame -h -b 128 --tt "Foo Song Title" --ta "FooBar Artist" --tl "FooBar Album" \
--ty "2014" --tc "Ripped and encoded by Foo" --tg "Genre" audio01.wav audio01.mp3
```

Os 128 kbits especificados são uma taxa de bits padrão para MP3, enquanto as taxas de bits 160 e 192 fornecem maior qualidade. Quanto maior a taxa de bits, maior o tamanho do arquivo MP3 resultante. A opção `-h` ativa o modo " de maior qualidade, mas um pouco mais lento ". As opções que começam com `--t` indicam as tags ID3, que geralmente contêm informações sobre músicas, para serem incorporadas no arquivo MP3. Opções adicionais de codificação podem ser encontradas na página de manual do lame.

Para gravar um CD de áudio usando arquivos MP3, primeiro estes devem ser convertidos em um formato de arquivo não compactado. O XMMS pode ser usado para converter para o formato WAV, enquanto o mpg123 pode ser usado para converter para o formato de dados de áudio bruto, Pulse Code Modulation (PCM).

Para converter o arquivo audio01.mp3 usando mpg123, especifique o nome do arquivo PCM:

```
# mpg123 -s audio01.mp3 > audio01.pcm
```

Para usar XMMS para converter um arquivo MP3 para WAV, use esses passos:

Procedure: Convertendo para o Formato WAV no XMMS

1. Inicie o XMMS.
2. Clique com o botão direito do mouse na janela para abrir o menu XMMS.
3. Selecione `Preferences` abaixo de `Options`.
4. Altere o Plugin de Saída para "Disk Writer Plugin".
5. Pressione `Configure`.
6. Digite ou procure um diretório para gravar os arquivos descompactados.
7. Carregue o arquivo MP3 no XMMS como de costume, com o volume em 100% e as configurações de EQ desativadas.
8. Pressione `Play`. O XMMS aparecerá como se estivesse tocando o MP3, mas nenhuma música será ouvida. Na verdade, está tocando o MP3 para um arquivo.
9. Quando terminar, certifique-se de ajustar o Plugin de Saída padrão de volta ao que era antes para ouvir MP3 novamente.

Os formatos WAV e PCM podem ser usados com cdrecord. Ao usar arquivos WAV, haverá um pequeno som no início de cada trilha. Este som é o cabeçalho do arquivo WAV. O pacote binário ou port audio/sox pode ser usado para remover o cabeçalho:

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Consulte Criando e Usando Mídia em CD para mais informações sobre o uso de um gravador de CD no FreeBSD.

7.4. Reprodução de Vídeo

Antes de configurar a reprodução de vídeo, determine o modelo e o chipset da placa de vídeo. Embora o Xorg suporte uma ampla variedade de placas de vídeo, nem todas oferecem um bom desempenho de reprodução. Para obter uma lista de extensões suportadas pelo servidor Xorg usando a placa, execute `xdpyinfo` enquanto o Xorg está sendo executado.

É interessante ter um arquivo de teste MPEG pequeno para avaliar vários players e opções. Como alguns aplicativos de DVD procuram por mídia DVD em /dev/dvd por padrão, ou possuem esse nome de dispositivo codificado neles, pode ser útil fazer um link simbólico para o dispositivo adequado:

```
# ln -sf /dev/cd0 /dev/dvd
```

Devido à natureza do devfs(5), os links criados manualmente não persistirão após a reinicialização do sistema. Para recriar o link simbólico automaticamente quando o sistema inicializar, adicione a

seguinte linha ao arquivo /etc/devfs.conf:

```
link cd0 dvd
```

A decriptografia do DVD invoca certas funções que exigem permissão de gravação para o dispositivo DVD.

Para melhorar a interface de memória compartilhada do Xorg, recomenda-se aumentar os valores dessas variáveis sysctl(8):

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

7.4.1. Determinando os Recursos de Vídeo

Existem várias maneiras possíveis de exibir vídeo no Xorg e o que funciona é, em grande parte, dependente de hardware. Cada método descrito abaixo terá qualidade variável em diferentes hardwares.

Interfaces de vídeo comuns incluem:

1. Xorg: saída normal usando memória compartilhada.
2. XVideo: uma extensão para a interface Xorg que permite que o vídeo seja exibido diretamente em objetos que podem ser desenhados através de uma aceleração especial. Esta extensão oferece reprodução de boa qualidade, mesmo em máquinas de baixo custo. A próxima seção descreve como determinar se esta extensão está sendo executada.
3. SDL: Simple Directmedia Layer é uma camada de portabilidade para muitos sistemas operacionais, permitindo o desenvolvimento de aplicativos multiplataforma que fazem uso eficiente de som e gráficos. O SDL fornece uma abstração de baixo nível para o hardware, que às vezes pode ser mais eficiente que a interface do Xorg. No FreeBSD, o SDL pode ser instalado usando o pacote ou port devel/sdl20.
4. DGA: Direct Graphics Access é uma extensão do Xorg que permite que um programa contorne o servidor Xorg e altere diretamente o quadro de buffer. Como ele depende de um mapeamento de memória de baixo nível, os programas que o utilizam devem ser executados como `root`. A extensão DGA pode ser testada e comparada usando dga(1). Quando o `dga` está em execução, ele altera as cores do display sempre que uma tecla é pressionada. Para sair, pressione `q`.
5. SVGAlib: uma camada gráfica de console de baixo nível.

7.4.1.1. XVideo

Para verificar se esta extensão está em execução, use `xvinfo`:

```
% xvinfo
```

O XVideo é suportado pela placa de vídeo se o resultado for semelhante a:

```
X-Video Extension version 2.2
  screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
```

```
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x7e0, 0xf800
      id: 0x31313259 (Y211)
        guid: 59323131-0000-0010-8000-00aa00389b71
        bits per pixel: 6
        number of planes: 3
        type: YUV (packed)
      id: 0x0
        guid: 00000000-0000-0000-0000-000000000000
        bits per pixel: 0
        number of planes: 0
        type: RGB (packed)
        depth: 1
        red, green, blue masks: 0x0, 0x0, 0x0
```

Os formatos listados, como YUV2 e YUV12, não estão presentes em todas as implementações do XVideo e sua ausência pode atrapalhar alguns players.

Se o resultado, ao invés disso, se parecer com:

```
X-Video Extension version 2.2
screen 0
no adaptors present
```

O XVideo provavelmente não é compatível com a placa. Isso significa que será mais difícil para o monitor atender às demandas computacionais de renderização de vídeo, dependendo da placa de vídeo e do processador.

7.4.2. Ports e Pacotes Lidando com Vídeo

Esta seção apresenta alguns dos softwares disponíveis na Coleção de Ports do FreeBSD, que podem ser usados para reprodução de vídeo.

7.4.2.1. MPlayer e MEncoder

O MPlayer é um reprodutor de vídeo em linha de comando com uma interface gráfica opcional que visa oferecer velocidade e flexibilidade. Outros front-ends gráficos para o MPlayer estão disponíveis na Coleção de Ports do FreeBSD.

O MPlayer pode ser instalado usando o pacote ou port multimedia/mplayer. Várias opções de compilação estão disponíveis e uma variedade de verificações de hardware ocorre durante o processo de compilação. Por esses motivos, alguns usuários preferem compilar um port ao invés de instalar o pacote.

Ao compilar o port, as opções do menu devem ser revisadas para determinar o tipo de suporte a ser

compilado no port. Se uma opção não estiver selecionada, o MPlayer não poderá exibir esse tipo de formato de vídeo. Use as setas e a barra de espaço para selecionar os formatos necessários. Quando terminar, pressione `Enter` para continuar a compilação e instalação do port.

Por padrão, o pacote ou port construirá o utilitário de linha de comando `mplayer` e o utilitário gráfico `gmplayer`. Para codificar vídeos, compile o port multimedia/mencoder. Devido a restrições de licenciamento, um pacote não está disponível para o MEncoder.

A primeira vez que o MPlayer for executado, ele criará um arquivo ~/.mplayer no diretório pessoal do usuário. Esse subdiretório contém versões padrões dos arquivos de configurações específicos do usuário.

Esta seção descreve apenas alguns usos comuns. Consulte o mplayer(1) para uma descrição completa de suas inúmeras opções.

Para reproduzir o arquivo testfile.avi, especifique as interfaces de vídeo com `-vo`, conforme mostrado nos exemplos a seguir:

```
% mplayer -vo xv testfile.avi
```

```
% mplayer -vo sdl testfile.avi
```

```
% mplayer -vo x11 testfile.avi
```

```
# mplayer -vo dga testfile.avi
```

```
# mplayer -vo 'sdl:dga' testfile.avi
```

Vale a pena tentar todas essas opções, pois seu relativo desempenho depende de muitos fatores e varia significativamente com o hardware.

Para reproduzir um DVD, substitua testfile.avi por `dvd://N -dvd-device DEVICE`, em que *N* é o número do título a ser reproduzido e *DEVICE* é o nó do dispositivo para o DVD. Por exemplo, para reproduzir o filme 3 de /dev/dvd:

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```



O dispositivo padrão de DVD pode ser definido durante a construção do port MPlayer incluindo a opção `WITH_DVD_DEVICE=/path/to/desired/device`. Por padrão, o dispositivo é /dev/cd0. Mais detalhes podem ser encontrados no Makefile.options do port.

Para parar, pausar, avançar e assim por diante, use uma tecla de atalho. Para ver a lista de atalhos de teclado, execute `mplayer -h` ou leia o mplayer (1).

Opções de reprodução adicionais incluem `-fs -zoom`, que ativa o modo de tela cheia e `-framedrop`, o que ajuda no desempenho.

Cada usuário pode adicionar opções comumente usadas ao seu ~/.mplayer/config assim:

```
vo=xv
fs=yes
zoom=yes
```

O `mplayer` pode ser usado para copiar um filme de DVD para um arquivo .vob. Para gravar o filme em um segundo DVD:

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

O arquivo de saída, out.vob, estará no formato MPEG.

Qualquer pessoa que deseje obter um alto nível de experiência com vídeo UNIX™ deve consultar mplayerhq.hu/DOCS. Como é tecnicamente informativa, esta documentação deve ser considerada como leitura obrigatória antes de enviar qualquer relatório de bug.

Antes de usar o `mencoder`, é interessante familiarizar-se com as opções descritas em mplayerhq.hu/DOCS/HTML/en/mencoder.html. Existem inúmeras maneiras de melhorar a qualidade, diminuir a taxa de bits e alterar os formatos, e algumas dessas opções podem fazer a diferença entre bom ou mau desempenho. Combinações impróprias de opções de linha de comando podem produzir arquivos de saída que não podem ser reproduzidos até mesmo por `mplayer`.

Aqui está um exemplo de uma cópia simples:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Para copiar para um arquivo, use `-dumpfile` com o `mplayer`.

Para converter input.avi para o codec MPEG4 com codificação de áudio MPEG3, primeiro instale o port audio/lame. Devido a restrições de licenciamento, um pacote não está disponível. Uma vez instalado, digite:

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
     -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Isso produzirá uma saída reproduzível por aplicativos como `mplayer` e `xine`.

input.avi pode ser substituído por `dvd://1 -dvd-device /dev/dvd` e executado como `root` para recodificar um filme de DVD diretamente. Como pode levar algumas tentativas para obter o

resultado desejado, recomenda-se gravar o arquivo de um filme e trabalhar nele.

7.4.2.2. O Player (reprodutor) de Vídeo xine

O xine é um reprodutor de vídeo com uma biblioteca base reutilizável e um executável modular que pode ser estendido com plug-ins. Pode ser instalado usando o pacote ou port multimedia/xine.

Na prática, o xine requer uma CPU rápida com uma placa de vídeo rápida ou suporte para a extensão XVideo. O player de vídeo xine apresenta melhor desempenho nas interfaces XVideo.

Por padrão, o player xine inicia uma interface gráfica com o usuário. Os menus podem então ser usados para abrir um arquivo específico.

Alternativamente, o xine pode ser executado a partir da linha de comando, especificando o nome do arquivo a ser reproduzido:

```
% xine -g -p mymovie.avi
```

Consulte xine-project.org/faq para mais informações e dicas de solução de problemas.

7.4.2.3. As Utilidades do Transcode

O Transcode fornece um conjunto de ferramentas para recodificar arquivos de vídeo e áudio. O Transcode pode ser usado para mesclar arquivos de vídeo ou reparar arquivos quebrados usando ferramentas de linha de comando com interfaces de fluxo stdin/stdout.

No FreeBSD, o Transcode pode ser instalado usando o pacote ou port multimedia/transcode. Muitos usuários preferem compilar o port, pois fornece um menu de opções de compilação para especificar o suporte e os codecs a serem compilados. Se uma opção não for selecionada, o Transcode não poderá codificar esse formato. Use as setas e a barra de espaço para selecionar os formatos necessários. Quando terminar, pressione `Enter` para continuar a compilação e instalação do port.

Este exemplo demonstra como converter um arquivo DivX em um arquivo PAL MPEG-1 (PAL VCD):

```
% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa
```

O arquivo MPEG resultante, output_vcd.mpg, está pronto para ser executado com o MPlayer. O arquivo pode ser gravado em uma mídia (CD), para criar um CD de vídeo usando um utilitário como multimedia/vcdimager ou sysutils/cdrdao.

Além da página de manual do `transcode`, consulte transcoding.org/cgi-bin/transcode para mais informações e exemplos.

7.5. Placas de TV

As placas de TV podem ser usadas para assistir à transmissão ou à TV a cabo em um computador. A

maioria das placas aceitam vídeo composto por meio de uma entrada RCA ou S-video e algumas placas incluem um sintonizador de rádio FM.

O FreeBSD fornece suporte para placas de TV baseadas em PCI usando um chip de captura de vídeo Brooktree Bt848/849/878/879 com o driver bktr(4). Este driver suporta a maioria das placas de vídeo Pinnacle PCTV. Antes de comprar uma placa de TV, consulte bktr(4) para obter uma lista dos sintonizadores suportados.

7.5.1. Carregando o Driver

Para usar a placa, o driver bktr(4) deve ser carregado. Para automatizar isso no momento da inicialização, adicione a seguinte linha ao arquivo /boot/loader.conf:

```
bktr_load="YES"
```

Como alternativa, pode-se compilar estaticamente o suporte para a placa de TV em um kernel personalizado. Nesse caso, adicione as seguintes linhas ao arquivo de configuração do kernel personalizado:

```
device   bktr
device  iicbus
device  iicbb
device  smbus
```

Esses dispositivos adicionais são necessários, pois os componentes da placa são interconectados por meio de um barramento I2C. Em seguida, crie e instale um novo kernel.

Para testar se o sintonizador foi detectado corretamente, reinicialize o sistema. A placa de TV deve aparecer nas mensagens de inicialização, conforme mostrado neste exemplo:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

As mensagens serão diferentes de acordo com o hardware. Se necessário, é possível substituir alguns dos parâmetros detectados usando sysctl(8) ou opções de configuração de kernel personalizadas. Por exemplo, para forçar o sintonizador a usar um sintonizador SECAM da Philips, adicione a seguinte linha a um arquivo de configuração de kernel personalizado:

```
options OVERRIDE_TUNER=6
```

ou, use sysctl(8):

```
# sysctl hw.bt848.tuner=6
```

Consulte bktr(4) para obter uma descrição disponível dos parâmetros do sysctl(8) e opções do kernel.

7.5.2. Aplicações Úteis

Para usar a placa de TV, instale um dos seguintes aplicativos:

- multimedia/fxtv oferece recursos de captura de imagem/áudio/vídeo numa transmissão de TV no monitor do computador.
- O multimedia/xawtv é outro aplicativo de TV com recursos semelhantes.
- O audio/xmradio fornece uma aplicação para usar o sintonizador de rádio FM de uma placa de TV.

Mais aplicações estão disponíveis na Coleção de Ports do FreeBSD.

7.5.3. Solução de problemas

Se forem encontrados problemas com a placa de TV, verifique se o chip de captura de vídeo e o sintonizador são compatíveis com bktr(4) e que as opções corretas de configuração foram usadas. Para obter mais suporte ou para fazer perguntas sobre as placas de TV suportadas, consulte a lista de discussão freebsd-multimedia.

7.6. MythTV

MythTV é um popular aplicativo de gravação de vídeo pessoal (PVR). Esta seção demonstra como instalar e configurar o MythTV no FreeBSD. Consulte mythtv.org/wiki para mais informações sobre como usar o MythTV.

MythTV requer um frontend e um backend. Esses componentes podem ser instalados no mesmo sistema ou em máquinas diferentes.

O frontend pode ser instalado no FreeBSD usando o pacote ou port multimedia/mythtv-frontend. O Xorg também deve ser instalado e configurado conforme descrito em O sistema X Window. Idealmente, este sistema tem uma placa de vídeo que suporta Compensação de Movimento de X-Vídeo (XvMC) e, opcionalmente, um controle remoto compatível com o Controle Remoto Infravermelho do Linux (LIRC).

Para instalar o backend e o frontend no FreeBSD, use o pacote ou port multimedia/mythtv. Um servidor de banco de dados MySQL™ também é necessário e deve ser instalado automaticamente como uma dependência. Opcionalmente, este sistema deve ter uma placa sintonizadora e armazenamento suficiente para armazenar os dados gravados.

7.6.1. Hardware

O MythTV usa o Video for Linux (V4L) para acessar dispositivos de entrada de vídeo, como

codificadores e sintonizadores. No FreeBSD, o MythTV funciona melhor com placas USB DVB-S/C/T, pois são bem suportadas pelo pacote multimedia/webcamd ou pelo port que forneça uma aplicação V4L userland. Qualquer placa de transmissão de vídeo digital (DVB) suportada pelo webcamd deve funcionar com o MythTV. Uma lista de placas suportadas conhecidas pode ser encontrada em wiki.freebsd.org/WebcamCompat. Drivers também estão disponíveis para placas Hauppauge nos ports multimedia/pvr250 e multimedia/pvrxxx, mas eles fornecem uma interface de driver não padronizados que não funcionam com versões do MythTV posteriores à 0.23. Devido a restrições de licenciamento, nenhum pacote está disponível e esses dois ports devem ser compilados.

A página wiki.freebsd.org/HTPC contém uma lista de todos os drivers DVB disponíveis.

7.6.2. Configurando o Backend MythTV

Para instalar o MythTV usando pacotes binários:

```
# pkg install mythtv
```

Como alternativa, para instalar a partir da Coleção de Ports:

```
# cd /usr/ports/multimedia/mythtv
# make install
```

Uma vez instalado, configure o banco de dados do MythTV:

```
# mysql -uroot -p < /usr/local/shared/mythtv/database/mc.sql
```

Em seguida, configure o backend:

```
# mythtv-setup
```

Finalmente, inicie o backend:

```
# sysrc mythbackend_enable=yes
# service mythbackend start
```

7.7. Scanners de Imagem

No FreeBSD, o acesso aos scanners de imagens é fornecido pelo SANE (Scanner Access Now Easy), que está disponível na Coleção de Ports do FreeBSD. O SANE também usará alguns drivers de dispositivos do FreeBSD para fornecer acesso ao hardware do scanner.

O FreeBSD suporta os scanners SCSI e USB. Dependendo da interface do scanner, são necessários drivers de dispositivos diferentes. Certifique-se de que o scanner seja suportado pelo SANE antes de executar qualquer configuração. Consulte http://www.sane-project.org/sane-supported-devices.html

para obter mais informações sobre os scanners suportados.

Este capítulo descreve como determinar se o scanner foi detectado pelo FreeBSD. Em seguida, ele fornece uma visão geral de como configurar e usar o SANE em um sistema FreeBSD.

7.7.1. Verificando o Scanner

O kernel GENERIC inclui os drivers de dispositivos necessários para suportar scanners USB. Usuários com um kernel personalizado devem garantir que as seguintes linhas estejam presentes no arquivo de configuração do kernel personalizado:

```
device usb
device uhci
device ohci
device ehci
device xhci
```

Para verificar se o scanner USB foi detectado, conecte-o e execute o comando `dmesg`, sendo então possível ver se o scanner aparece no buffer de mensagens do sistema. Em caso afirmativo, deve ser exibida uma mensagem semelhante a esta:

```
ugen0.2: <EPSON> at usbus0
```

Neste exemplo, um scanner EPSON Perfection™ 1650 USB foi detectado em /dev/ugen0.2.

Se o scanner usar uma interface SCSI, é importante saber qual placa controladora SCSI será usada. Dependendo do chipset SCSI, um arquivo de configuração do kernel personalizado pode ser necessário. O kernel GENERIC suporta os controladores SCSI mais comuns. Consulte /usr/src/sys/conf/NOTES para determinar a linha correta a ser adicionada a um arquivo de configuração de kernel personalizado. Além do driver de adaptador SCSI, as seguintes linhas são necessárias em um arquivo de configuração de kernel personalizado:

```
device scbus
device pass
```

Verifique se o dispositivo é exibido no buffer de mensagens do sistema:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Se o scanner não foi ligado na inicialização do sistema, ainda é possível forçar manualmente a detecção executando uma varredura de barramento SCSI com o comando `camcontrol`:

```
# camcontrol rescan all
```

```
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

O scanner deve agora aparecer na lista de dispositivos SCSI:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

Consulte scsi(4) e camcontrol(8) para mais detalhes sobre dispositivos SCSI no FreeBSD.

7.7.2. Configuração do SANE

O sistma SANE provê o acesso ao scanner via backends (graphics/sane-backends). Consulte http://www.sane-project.org/sane-supported-devices.html para determinar qual backend suporta o scanner. Uma interface gráfica é fornecida por aplicações terceiras como Kooka (graphics/kooka) ou XSane (graphics/xsane). Os backends do SANE são suficientes para testar o scanner.

Para instalar os backends do pacote binário:

```
# pkg install sane-backends
```

Alternativamente, para instalar a partir da Coleção de Ports

```
# cd /usr/ports/graphics/sane-backends
# make install clean
```

Depois de instalar o pacote ou port graphics/sane-backends, use o comando `sane-find-scanner` para verificar a detecção do scanner pelo sistema SANE:

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

A saída deve mostrar o tipo de interface do scanner e o nó do dispositivo usado para conectar o scanner ao sistema. O fornecedor e o modelo do produto podem ou não aparecer.



Alguns scanners USB exigem que o firmware seja carregado. Consulte sane-find-scanner(1) e sane(7) para mais detalhes.

Em seguida, verifique se o scanner será identificado por uma interface de digitalização. Os backends SANE incluem o comando `scanimage`, que pode ser usado para listar os dispositivos e

realizar uma aquisição de imagens. Use a opção `-L` para listar os dispositivos do scanner. O primeiro exemplo é para um scanner SCSI e o segundo é para um scanner USB:

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
# scanimage -L
device 'epson2:libusb:000:002' is a Epson GT-8200 flatbed scanner
```

Neste segundo exemplo,`epson2` é o nome do backend e `libusb:000:002` significa que `/dev/ugen0.2` é o dispositivo usado pelo scanner.

Se o comando `scanimage` não conseguir identificar o scanner, esta mensagem será exibida:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Se isso acontecer, edite o arquivo de configuração de backend em /usr/local/etc/sane.d/ e defina o dispositivo de scanner usado. Por exemplo, se o modelo de scanner não detectado for um EPSON Perfection™ 1650 e usar o backend `epson2`, edite o arquivo /usr/ local/etc/sane.d/epson2.conf. Ao editar, adicione uma linha especificando a interface e o nó do dispositivo usado. Nesse caso, adicione a seguinte linha:

```
usb /dev/ugen0.2
```

Salve as edições e verifique se o scanner está identificado com o nome do back-end correto e com o nó do dispositivo:

```
# scanimage -L
device 'epson2:libusb:000:002' is a Epson GT-8200 flatbed scanner
```

Depois que o comando `scanimage -L` identificar o scanner, a configuração estará completa e o scanner estará pronto para ser usado.

Embora o `scanimage` possa ser usado para realizar uma digitalização de imagem a partir da linha de comando, muitas vezes é preferível usar uma interface gráfica para executar o escaneamento. Aplicações como Kooka ou XSane são interfaces de digitalização populares. Eles oferecem recursos avançados, como vários modos de digitalização, correção de cores e digitalizações em lote. O XSane também pode ser usado como um plugin GIMP.

7.7.3. Permissões do Scanner

Para ter acesso ao scanner, o usuário precisa ler e gravar as permissões no nó do dispositivo usado pelo scanner. No exemplo anterior, o scanner USB usa o nó do dispositivo /dev/ugen0.2 que é realmente um link simbólico para o nó do dispositivo real /dev/usb/0.2.0. O link simbólico e o nó do dispositivo pertencem, respectivamente, aos grupos `wheel` e `operator`. Adicionando o usuário a esses grupos, será permitido o acesso ao scanner, considera-se inseguro adicionar um usuário a `wheel`. Uma solução melhor é criar um grupo e tornar o dispositivo de scanner acessível aos membros desse grupo.

Este exemplo cria um grupo chamado `usb`:

```
# pw groupadd usb
```

Então, crie um link simbólico para /dev/ugen0.2 e o nó do dispositivo /dev/usb/0.2.0 para ficarem acessíveis ao grupo `usb` com permissões de gravação `0660` ou `0664` adicionando as seguintes linhas ao /etc/devfs.rules:

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0666 group usb
```



Acontece do nó do dispositivo mudar com a adição ou remoção de dispositivos, então você pode querer dar acesso a todos os dispositivos USB usando esse conjunto de regas:

```
[system=5]
add path 'ugen*' mode 0660 group usb
add path 'usb/*' mode 0666 group usb
```

Finalmente, adicione os usuários a `usb` para permitir acesso ao scanner:

```
# pw groupmod usb -m joe
```

Para mais detalhes, consulte pw(8).

Capítulo 8. Configurando o kernel do FreeBSD

8.1. Sinopse

O kernel é o núcleo do sistema operacional do FreeBSD. Ele é responsável pelo gerenciamento de memória, aplicação de controles de segurança, rede, acesso ao disco e muito mais. Embora grande parte do FreeBSD seja configurável dinamicamente, ainda é necessário configurar e compilar um kernel personalizado ocasionalmente.

Depois de ler este capítulo, você saberá:

- Quando compilar um kernel personalizado.
- Como obter um inventário do hardware.
- Como personalizar um arquivo de configuração do kernel.
- Como usar o arquivo de configuração do kernel para criar e compilar um novo kernel.
- Como instalar o novo kernel.
- Como solucionar problemas se as coisas derem errado.

Todos os comandos listados nos exemplos deste capítulo devem ser executados como `root`.

8.2. Por que compilar um kernel personalizado?

Tradicionalmente, o FreeBSD usava um kernel monolítico. O kernel era um grande programa, suportava uma lista fixa de dispositivos e, para mudar o comportamento do kernel, era preciso compilar e depois reinicializar em um novo kernel.

Hoje, a maior parte da funcionalidade do kernel do FreeBSD está contida em módulos que podem ser dinamicamente carregados e descarregados do kernel, conforme necessário. Isso permite que o kernel em execução se adapte imediatamente ao novo hardware e que novas funcionalidades sejam trazidas para o kernel. Isso é conhecido como um kernel modular.

Ocasionalmente, ainda é necessário executar a configuração do kernel estático. Às vezes, a funcionalidade necessária é tão ligada ao kernel que não pode ser carregada dinamicamente. Alguns ambientes de segurança impedem o carregamento e descarregamento de módulos do kernel e exigem que apenas a funcionalidade necessária seja estaticamente compilada no kernel.

Construir um kernel personalizado é muitas vezes um rito de passagem para usuários avançados do BSD. Este processo, embora consuma tempo, pode fornecer benefícios ao sistema FreeBSD. Ao contrário do kernel GENERIC, que deve suportar uma ampla gama de hardware, um kernel personalizado pode ser reduzido para fornecer suporte apenas para o hardware desse computador. Isso tem vários benefícios, tais como:

- Tempo de inicialização mais rápido. Uma vez que o kernel irá verificar apenas o hardware existente no sistema, o tempo que o sistema leva para inicializar pode diminuir.

- Diminuir o uso de memória. Um kernel personalizado geralmente usa menos memória que o kernel GENERIC ao omitir recursos e drivers de dispositivo que não são utilizados. Isso é importante porque o código do kernel permanece residente na memória física o tempo todo, impedindo que a memória seja usada pelos aplicativos. Por esse motivo, um kernel personalizado é útil em um sistema com uma pequena quantidade de RAM.
- Suporte adicional de hardware. Um kernel personalizado pode adicionar suporte para dispositivos que não estão presentes no kernel GENERIC.

Antes de criar um kernel personalizado, considere a razão para isso. Se houver necessidade de suporte para um hardware específico, ele já pode existir como um módulo.

Os módulos do kernel existem em /boot/kernel e podem ser dinamicamente carregados no kernel em execução usando o kldload(8). A maioria dos drivers do kernel tem um módulo carregável e uma página de manual. Por exemplo, o driver Ethernet sem fio ath(4) tem as seguintes informações em sua página de manual:

```
Como alternativa, para carregar o driver como um módulo no momento da inicialização,
coloque o a seguinte linha no loader.conf(5):

    if_ath_load="YES"
```

Adicionar `if_ath_load="YES"` ao /boot/loader.conf carregará este módulo dinamicamente no momento da inicialização.

Em alguns casos, não há nenhum módulo associado em /boot/kernel. Isso é verdade principalmente para certos subsistemas.

8.3. Encontrando o hardware do sistema

Antes de editar o arquivo de configuração do kernel, é recomendável realizar um inventário do hardware da máquina. Em um sistema de inicialização dupla, o inventário pode ser criado a partir do outro sistema operacional. Por exemplo, o Device Manager da Microsoft™ contém informações sobre os dispositivos instalados.



Algumas versões do Microsoft ™Windows™ têm um ícone System que pode ser usado para acessar o Device Manager.

Se o FreeBSD for o único sistema operacional instalado, use o dmesg(8) para determinar o hardware que foi encontrado e listado durante a verificação de inicialização. A maioria dos drivers de dispositivos no FreeBSD tem uma página de manual que lista o hardware suportado pelo driver. Por exemplo, as seguintes linhas indicam que o driver psm(4) encontrou um mouse:

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

Como esse hardware existe, esse driver não deve ser removido de um arquivo de configuração de kernel personalizado.

Se a saída do `dmesg` não exibir os resultados da saída da verificação de inicialização, leia o conteúdo do /var/run/dmesg.boot.

Outra ferramenta para encontrar hardware é o pciconf(8), que fornece uma saída mais detalhada. Por exemplo:

```
% pciconf -lv
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01 hdr
=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

Esta saída mostra que o driver ath localizou um dispositivo Ethernet sem fio.

O sinalizador `-k` do man(1) pode ser usado para fornecer informações úteis. Por exemplo, ele pode ser usado para exibir uma lista de páginas de manual que contêm uma marca ou um nome de dispositivo específico:

```
# man -k Atheros
ath(4)                  - Atheros IEEE 802.11 wireless network driver
ath_hal(4)              - Atheros Hardware Access Layer (HAL)
```

Depois que a lista de inventário de hardware for criada, consulte-a para garantir que os drivers para o hardware instalado não sejam removidos à medida que a configuração do kernel personalizado é editada.

8.4. O Arquivo de Configuração

Para criar um arquivo de configuração do kernel personalizado e compilar um kernel personalizado, a árvore de código-fonte completa do FreeBSD deve ser instalada primeira.

Se o /usr/src/ não existir ou estiver vazio, o código-fonte não foi instalado. O fonte pode ser instalado usando o Subversion e as instruções em Usando o Subversion.

Depois que o código-fonte for instalado, revise o conteúdo do /usr/src/sys. Este diretório contém vários subdiretórios, incluindo aqueles que representam as seguintes arquiteturas suportadas: amd64, i386, powerpc e sparc64. Tudo dentro do diretório de uma arquitetura em particular lida apenas com essa arquitetura e o restante do código é código independente de máquina comum a todas as plataformas. Cada arquitetura suportada tem um subdiretório conf que contém o arquivo de configuração do kernel GENERIC para essa arquitetura.

Não faça edições no GENERIC. Em vez disso, copie o arquivo para um nome diferente e faça edições na cópia. A convenção é usar um nome do host com todas as letras maiúsculas. Ao manter várias

máquinas FreeBSD com hardware diferente, é uma boa idéia nomeá-lo com o nome do host da máquina. Este exemplo cria uma cópia, denominada MYKERNEL, do arquivo de configuração GENERIC para a arquitetura `amd64`:

```
# cd /usr/src/sys/amd64/conf
# cp GENERIC MYKERNEL
```

O MYKERNEL agora pode ser personalizado com qualquer editor de texto ASCII. O editor padrão é o vi, embora um editor mais fácil para iniciantes, chamado ee, também seja instalado com o FreeBSD.

O formato do arquivo de configuração do kernel é simples. Cada linha contém uma palavra-chave que representa um dispositivo ou subsistema, um argumento e uma breve descrição. Qualquer texto depois de um `é considerado um comentário e ignorado. Para remover o suporte do kernel para um dispositivo ou subsistema, coloque um` no início da linha que representa esse dispositivo ou subsistema. Não adicione ou remova um `#` para qualquer linha que você não entenda.



É fácil remover o suporte para um dispositivo ou opção e acabar com um kernel quebrado. Por exemplo, se o driver ata(4) for removido do arquivo de configuração do kernel, um sistema usando os drivers de disco ATA pode não inicializar. Em caso de dúvida, basta deixar o suporte no kernel.

Além das breves descrições fornecidas neste arquivo, descrições adicionais estão contidas no arquivo NOTES, o qual pode ser encontrado no mesmo diretório que o GENERIC para aquela arquitetura. Para opções independentes de arquitetura, consulte /usr/src/sys/conf/NOTES.



Quando terminar de personalizar o arquivo de configuração do kernel, salve uma cópia de backup em um local fora do /usr/src.

Como alternativa, mantenha o arquivo de configuração do kernel em outro lugar e crie um link simbólico para o arquivo:

```
# cd /usr/src/sys/amd64/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

Uma diretiva `include` está disponível para uso em arquivos de configuração. Isso permite que outro arquivo de configuração seja incluído no arquivo atual, facilitando a manutenção de pequenas alterações em relação a um arquivo existente. Se apenas um pequeno número de opções ou drivers adicionais forem necessários, isso permitirá que um delta seja mantido com relação ao GENERIC, conforme mostrado neste exemplo:

```
include GENERIC
ident MYKERNEL

options         IPFIREWALL
```

```
options         DUMMYNET
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPDIVER
```

Usando este método, o arquivo de configuração local expressa as diferenças locais em relação ao kernel GENERIC. Conforme as atualizações são realizadas, os novos recursos adicionados ao GENERIC também serão adicionados ao kernel local, a menos que sejam especificamente evitados usando `nooptions` ou `nodevice`. Uma lista abrangente de diretivas de configuração e suas descrições pode ser encontrada em config(5).



Para compilar um arquivo que contém todas as opções disponíveis, execute o seguinte comando como `root`:

```
# cd /usr/src/sys/arch/conf && make LINT
```

8.5. Criando e Instalando um Kernel Customizado

Depois que as edições no arquivo de configuração personalizada forem salvas, o código-fonte do kernel poderá ser compilado usando as seguintes etapas:

Procedure: Compilando um Kernel

1. Mude para este diretório:

   ```
   # cd /usr/src
   ```

2. Compile o novo kernel especificando o nome do arquivo de configuração do kernel personalizado:

   ```
   # make buildkernel KERNCONF=MYKERNEL
   ```

3. Instale o novo kernel associado ao arquivo de configuração do kernel especificado. Este comando irá copiar o novo kernel para /boot/kernel/kernel e salvar o kernel antigo para /boot/kernel.old/kernel:

   ```
   # make installkernel KERNCONF=MYKERNEL
   ```

4. Desligue o sistema e reinicie no novo kernel. Se algo der errado, consulte O kernel não inicializa.

Por padrão, quando um kernel personalizado é compilado, todos os módulos do kernel são reconstruídos. Para atualizar um kernel mais rapidamente ou para construir apenas módulos

customizados, edite o /etc/make.conf antes de começar a construir o kernel.

Por exemplo, esta variável especifica a lista de módulos para compilar em vez de usar o padrão de construir todos os módulos:

```
MODULES_OVERRIDE = linux acpi
```

Como alternativa, essa variável lista quais módulos excluir do processo de criação:

```
WITHOUT_MODULES = linux acpi sound
```

Variáveis adicionais estão disponíveis. Consulte make.conf(5) para detalhes.

8.6. Se algo der errado

Existem quatro categorias de problemas que podem ocorrer ao criar um kernel personalizado:

falhas na `config`

Se o `config` falhar, ele imprimirá o número da linha que está incorreta. Como exemplo, para a seguinte mensagem, certifique-se de que a linha 17 seja digitada corretamente, comparando-a com GENERIC ou NOTES:

```
config: line 17: syntax error
```

falha no `make`

Se o `make` falhar, geralmente é devido a um erro no arquivo de configuração do kernel que não é grave o suficiente para o `config` capturar. Revise a configuração, e se o problema não for aparente, envie um email para a lista de discussão de questões gerais do FreeBSD contendo o arquivo de configuração do kernel.

O kernel não inicializa

Se o novo kernel não inicializar ou não reconhecer os dispositivos, não entre em pânico! Felizmente, o FreeBSD possui um excelente mecanismo para recuperação de kernels incompatíveis. Simplesmente escolha o kernel para inicializar a partir do gerenciador de inicialização do FreeBSD. Isso pode ser acessado quando o menu de inicialização do sistema aparecer, selecionando a opção "Escape to a loader prompt". No prompt, digite `boot kernel.old` ou o nome de qualquer outro kernel que seja conhecido por inicializar corretamente.

Após inicializar com um kernel correto, verifique o arquivo de configuração e tente construí-lo novamente. Um recurso útil é o /var/log/messages que registra as mensagens do kernel de cada inicialização bem-sucedida. Além disso, o dmesg(8) imprimirá as mensagens do kernel a partir da inicialização atual.



Ao solucionar problemas de um kernel, certifique-se de manter uma cópia do GENERIC, ou algum outro kernel que funcione, como um nome diferente que

não será apagado na próxima compilação. Isto é importante porque toda vez que um novo kernel é instalado, o kernel.old é sobrescrito com o último kernel instalado, que pode ou não ser inicializável. Assim que possível, mova o kernel funcional renomeando o diretório que contém o kernel correto:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

O kernel funciona, mas o ps(1) não

Se a versão do kernel for diferente daquela com a qual os utilitários do sistema foram construídos, por exemplo, um kernel compilado a partir do código-fonte do -CURRENT é instalado em um sistema -RELEASE, muitos comandos de status do sistema como ps(1) e vmstat(8) não funcionarão. Para corrigir isso, recompile e instale o world usando a mesma versão da árvore de código-fonte que o kernel. Nunca é uma boa ideia usar uma versão diferente do kernel do que o resto do sistema operacional.

Capítulo 9. Impressão

Colocar informações no papel é uma função vital, apesar de muitas tentativas de eliminá-la. A impressão tem dois componentes básicos. Os dados devem ser entregues à impressora e devem estar em um formato que a impressora possa entender.

9.1. Inicio Rápido

A impressão básica pode ser configurada rapidamente. A impressora deve ser capaz de imprimir texto simples ASCII. Para imprimir em outros tipos de arquivos, consulte Filtros.

1. Crie um diretório para armazenar arquivos enquanto eles estão sendo impressos:

```
# mkdir -p /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
# chmod 770 /var/spool/lpd/lp
```

2. Como `root`, crie /etc/printcap com estes conteúdos:

```
lp:\
    :lp=/dev/unlpt0:\  ①
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:
```

① Esta linha é para uma impressora conectada a uma porta USB.Para uma impressora conectada a uma porta paralela ou uma porta de "impressora", use:Para uma impressora conectada diretamente a uma rede, use:Substitua *network-printer-name* pelo nome de host DNS da impressora de rede.

3. Ative o `lpd` editando o /etc/rc.conf, adicionando esta linha:

```
lpd_enable="YES"
```

Inicie o serviço:

```
# service lpd start
Starting lpd.
```

4. Imprima um teste:

```
# printf "1. This printer can print.\n2. This is the second line.\n" | lpr
```



Se ambas as linhas não iniciarem na borda esquerda, mas em "degrau", consulte Impedindo degraus em impressoras de texto simples.

Arquivos de texto agora podem ser impressos com `lpr`. Dê o nome do arquivo na linha de comando ou canalize a saída diretamente no `lpr`.

```
% lpr textfile.txt
% ls -lh | lpr
```

9.2. Conexões de Impressora

As impressoras são conectadas a sistemas de computadores de várias maneiras. Geralmente, as impressoras desktop pequenas são conectadas diretamente à porta USB do computador. As impressoras mais antigas são conectadas a uma porta paralela ou a porta de "impressora". Algumas impressoras estão diretamente conectadas a uma rede, facilitando o compartilhamento com vários computadores. Algumas impressoras usam uma rara conexão de porta serial.

O FreeBSD pode se comunicar com todos esses tipos de impressoras.

USB

As impressoras USB podem ser conectadas a qualquer porta USB disponível no computador.

Quando o FreeBSD detecta uma impressora USB, duas entradas de dispositivos são criadas: /dev/ulpt0 e /dev/unlpt0. Os dados enviados para qualquer dispositivo serão retransmitidos para a impressora. Após cada trabalho de impressão, o ulpt0 reseta a porta USB. O reset da porta pode causar problemas em algumas impressoras, portanto, o dispositivo unlpt0 é normalmente usado em seu lugar. O unlpt0 não reseta a porta USB.

Paralela (IEEE-1284)

O dispositivo da porta paralela é o /dev/lpt0. Este dispositivo aparece independentemente se uma impressora está ou não conectada, ela não é autodetectada.

A maior parte dos fabricantes se afastou destas portas "legadas" e muitos computadores não as têm mais. Adaptadores podem ser usados para conectar uma impressora paralela a uma porta USB. Com este tipo de adaptador, a impressora pode ser tratada como se fosse uma impressora USB. Dispositivos chamados *servidores de impressão* também podem ser usados para conectar impressoras paralelas diretamente a uma rede.

Serial (RS-232)

Portas seriais são outro tipo de porta legada, raramente usada para impressoras, exceto em determinadas aplicações de nicho. Os cabos, os conectores e a fiação necessária variam muito.

Para portas seriais incorporadas em uma placa-mãe, o nome do dispositivo serial é /dev/cuau0 ou /dev/cuau1. Os adaptadores Seriais USB também podem ser usados, e eles aparecerão como /dev/cuaU0.

Vários parâmetros de comunicação devem ser conhecidos para se comunicar com uma

impressora serial. Os mais importantes são *baud rate* ou BPS (Bits por segundo) e *paridade*. Os valores variam, mas as impressoras seriais típicas usam uma taxa de transmissão de 9600 e nenhuma paridade.

Rede

As impressoras de rede estão conectadas diretamente à rede de computadores local.

O nome de host DNS da impressora deve ser conhecido. Se a impressora tiver um endereço dinâmico atribuído por DHCP, o DNS deverá ser atualizado dinamicamente para que o nome do host tenha sempre o endereço IP correto. As impressoras de rede geralmente recebem endereços IP estáticos para evitar esse problema.

A maioria das impressoras de rede entende os trabalhos de impressão enviados com o protocolo LPD. Um nome de fila de impressão também pode ser especificado. Algumas impressoras processam dados de maneira diferente, dependendo de qual fila é usada. Por exemplo, uma fila `raw` imprime os dados inalterados, enquanto a fila `text` adiciona retornos de carro aos textos simples.

Muitas impressoras de rede também podem imprimir dados enviados diretamente para a porta 9100.

9.2.1. Resumo

As conexões de rede com fio geralmente são as mais fáceis de configurar e oferecem a impressão mais rápida. Para conexão direta com o computador, a conexão USB é preferida em função da velocidade e da simplicidade. As conexões paralelas funcionam, mas têm limitações no comprimento do cabo e na velocidade. Conexões seriais são mais difíceis de configurar. A configuração do cabo difere entre os modelos, e os parâmetros de comunicação, como taxa de transmissão e bits de paridade, se somam a complexidade. Felizmente, as impressoras seriais são raras.

9.3. Linguagens de Descrição de Página Comuns

Os dados enviados a uma impressora devem estar em um idioma que a impressora possa entender. Esses idiomas são chamados de Linguagens de Descrição de Página ou PDLs.

ASCII

Texto ASCII simples é a maneira mais simples de enviar dados para uma impressora. Os caracteres correspondem um a um com o que será impresso: um `A` nos dados imprime um `A` na página. Muito pouca formatação está disponível. Não há como selecionar uma fonte ou espaçamento proporcional. A simplicidade forçada do texto ASCII simples significa que o texto pode ser impresso diretamente do computador com pouca ou nenhuma codificação ou tradução. A saída impressa corresponde diretamente ao que foi enviado.

Algumas impressoras baratas não conseguem imprimir texto ASCII simples. Isso as torna mais difíceis de configurar, mas geralmente ainda é possível fazê-lo.

PostScript™

O PostScript™ é quase o oposto do ASCII. Em vez de um texto simples, um programa PostScript™ é um conjunto de instruções que desenham o documento final. Fontes e gráficos diferentes podem ser usados. No entanto, esse poder tem um preço. O programa que desenha a página deve ser escrito. Geralmente este programa é gerado pelo software aplicativo, portanto, o processo é invisível para o usuário.

Impressoras baratas às vezes deixam de fora a compatibilidade com o PostScript™ como uma medida para economia de custos.

PCL (linguagem de comando de impressora)

A PCL é uma extensão do ASCII, adicionando sequências de escape para formatação, seleção de fontes e impressão de gráficos. Muitas impressoras fornecem suporte para PCL5. Algumas suportam o mais recente PCL6 ou o PCLXL. Essas versões posteriores são superconjuntos do PCL5 e podem fornecer uma impressão mais rápida.

Baseado em Host

Os fabricantes podem reduzir o custo de uma impressora, oferecendo um processador simples e muito pouca memória. Essas impressoras não são capazes de imprimir texto simples. Em vez disso, bitmaps de texto e gráficos são desenhados por um driver no computador host e, em seguida, enviados para a impressora. Estas são chamadas de impressoras *baseadas em host*.

A comunicação entre o driver e uma impressora baseada em host geralmente ocorre por meio de protocolos proprietários ou não documentados, tornando-os funcionais apenas nos sistemas operacionais mais comuns.

9.3.1. Convertendo PostScript™ para outros PDLs

Muitas aplicações da Coleção de Ports e muitos utilitários do FreeBSD produzem uma saída em PostScript™. Esta tabela mostra os utilitários disponíveis para converter o postscript em outros PDLs comuns:

Tabela 8. Saída PDLs

Saída PDL	Gerado por	Notas
PCL ou PCL5	print/ghostscript9	`-sDEVICE=ljet4` para monocromático, e `-sDEVICE=cljet5` para colorido
PCLXL ou PCL6	print/ghostscript9	`-sDEVICE=pxlmono` para monocromático, `-sDEVICE=pxlcolor` para colorido
ESC/P2	print/ghostscript9	`-sDEVICE=uniprint`
XQX	print/foo2zjs	

9.3.2. Resumo

Para facilitar a impressão, escolha uma impressora que suporte PostScript™. Impressoras que

suportam PCL são as próximas preferidas. Com o print/ghostscript, essas impressoras podem ser usadas como se entendessem nativamente PostScript™. Impressoras que suportam PostScript™ ou PCL diretamente quase sempre suportam a impressão direta de arquivos de texto simples ASCII também.

Impressoras baseadas em linha, como as jatos de tinta comuns, geralmente não suportam PostScript™ ou PCL. Elas geralmente podem imprimir arquivos de texto plano ASCII. O print/ghostscript suporta os PDL usados por algumas dessas impressoras. Entretanto, a impressão de uma página inteira baseada em gráficos nessas impressoras costuma ser muito lenta devido à grande quantidade de dados a serem transferidos e impressos.

Geralmente, as impressoras baseadas em host são mais difíceis de configurar. Algumas não podem ser usadas por causa de PDLs proprietários. Evite essas impressoras quando possível.

Descrições de muitos PDLs podem ser encontradas em http://www.undocprint.org/formats/page_description_languages. O PDL específico usado por vários modelos de impressoras pode ser encontrado em http://www.openprinting.org/printers.

9.4. Impressão Direta

Para impressão ocasional, os arquivos podem ser enviados diretamente para um dispositivo de impressora sem qualquer configuração. Por exemplo, um arquivo chamado exemplo.txt pode ser enviado para uma impressora USB:

```
# cp sample.txt /dev/unlpt0
```

A impressão direta para impressoras de rede depende das capacidades da impressora, mas a maioria aceita trabalhos de impressão na porta 9100, e o nc(1) pode ser usado com eles. Para imprimir o mesmo arquivo em uma impressora com o nome de host DNS de *netlaser*:

```
# nc netlaser 9100 < sample.txt
```

9.5. LPD (Daemon de impressora de linha)

A impressão de um arquivo em segundo plano é chamada de *spooling*. Um spooler permite que o usuário continue com outros programas no computador sem ter de esperar que a impressora conclua lentamente o trabalho de impressão.

O FreeBSD inclui um spooler chamado lpd(8). Os trabalhos de impressão são enviados com o comando lpr(1).

9.5.1. Configuração inicial

Um diretório para armazenar trabalhos de impressão é criado, a propriedade é definida e as permissões são definidas para impedir que outros usuários visualizem o conteúdo desses arquivos:

```
# mkdir -p /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
# chmod 770 /var/spool/lpd/lp
```

As impressoras são definidas no /etc/printcap. Uma entrada para cada impressora inclui detalhes como um nome, a porta onde ela está conectada e várias outras configurações. Crie /etc/printcap com estes conteúdos:

```
lp:\                ①
    :lp=/dev/unlpt0:\   ②
    :sh:\           ③
    :mx#0:\         ④
    :sd=/var/spool/lpd/lp:\ ⑤
    :lf=/var/log/lpd-errs:  ⑥
```

① O nome desta impressora. O lpr(1) envia trabalhos de impressão para a impressora `lp`, a menos que outra impressora seja especificada com `-P`, portanto, a impressora padrão deve ser denominada `lp`.

② O dispositivo em que a impressora está conectada. Substitua esta linha pela apropriada para o tipo de conexão mostrado aqui.

③ Suprimir a impressão de uma página de cabeçalho no início de um trabalho de impressão.

④ Não limite o tamanho máximo de um trabalho de impressão.

⑤ O caminho para o diretório de spooling desta impressora. Cada impressora usa seu próprio diretório de spooling.

⑥ O arquivo de log no qual os erros nesta impressora serão relatados.

Depois de criar o /etc/printcap, use chkprintcap(8) para testar se há erros:

```
# chkprintcap
```

Corrija quaisquer problemas relatados antes de continuar.

Ative o lpd(8) no /etc/rc.conf:

```
lpd_enable="YES"
```

Inicie o serviço:

```
# service lpd start
```

9.5.2. Imprimindo com o lpr(1)

Os documentos são enviados para a impressora com o `lpr`. Um arquivo a ser impresso pode ser nomeado na linha de comando ou canalizado para o `lpr`. Esses dois comandos são equivalentes, enviando o conteúdo de doc.txt para a impressora padrão:

```
% lpr doc.txt
% cat doc.txt | lpr
```

Impressoras podem ser selecionadas com `-P`. Para imprimir em uma impressora chamada *laser*:

```
% lpr -Plaser doc.txt
```

9.5.3. Filtros

Os exemplos mostrados até agora enviaram o conteúdo de um arquivo de texto diretamente para a impressora. Contanto que a impressora entenda o conteúdo desses arquivos, a saída será impressa corretamente.

Algumas impressoras não são capazes de imprimir texto simples, e o arquivo de entrada pode nem ser texto simples.

Filtros permitem que os arquivos sejam traduzidos ou processados. O uso típico é traduzir um tipo de entrada, como texto simples, em um formato que a impressora possa entender, como PostScript™ ou PCL. Os filtros também podem ser usados para fornecer recursos adicionais, como adicionar números de página ou destacar o código-fonte para facilitar a leitura.

Os filtros discutidos aqui são *filtros de entrada* ou *filtros de texto*. Esses filtros convertem o arquivo recebido em diferentes formatos. Use su(1) para se tornar `root` antes de criar os arquivos.

Os filtros são especificados em /etc/printcap com o identificador `if=`. Para usar /usr/local/libexec/lf2crlf como um filtro, modifique o /etc/printcap assim:

```
lp:\
    :lp=/dev/unlpt0:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/libexec/lf2crlf:\   ①
    :lf=/var/log/lpd-errs:
```

① `if=` identifica o *filtro de entrada* que será usado no texto recebido.



Os caracteres backslash de *continuação de linha* no final das linhas nas entradas do printcap revelam que uma entrada para uma impressora é na verdade apenas uma linha longa com entradas delimitadas por dois pontos. O exemplo anterior pode ser reescrito como uma única linha menos legível:

```
lp:lp=/dev/unlpt0:sh:mx#0:sd=/var/spool/lpd/lp:if=/usr/local/libexec/lf
2crlf:lf=/var/log/lpd-errs:
```

9.5.3.1. Impedindo degraus em impressoras de texto simples

Os arquivos de texto típicos do FreeBSD contêm apenas um único caractere de feed de linha no final de cada linha. Estas linhas vão ficar em "degraus" em uma impressora padrão:

```
Um arquivo impresso parece
                    como os degraus de uma escada
                                            espalhados pelo vento
```

Um filtro pode converter os caracteres de nova linha em retornos de carro e novas linhas. Os retornos de carro fazem a impressora retornar para a esquerda após cada linha. Crie o /usr/local/libexec/lf2crlf com este conteúdo:

```
#!/bin/sh
CR=$'\r'
/usr/bin/sed -e "s/$/${CR}/g"
```

Defina as permissões e torne-o executável:

```
# chmod 555 /usr/local/libexec/lf2crlf
```

Modifique o /etc/printcap para usar o novo filtro:

```
:if=/usr/local/libexec/lf2crlf:\
```

Teste o filtro imprimindo o mesmo arquivo de texto simples. O procedimento fará com que cada linha comece no lado esquerdo da página.

9.5.3.2. Texto simples chique em impressoras PostScript™ com print/enscript

O GNUEnscript converte arquivos de texto simples em arquivos formatados como PostScript™ para impressão em impressoras PostScript™. Ele adiciona números de página, quebra as linhas longas e fornece vários outros recursos para facilitar a leitura dos arquivos de texto impressos. Dependendo do tamanho do papel local, instale o print/enscript-letter ou o print/enscript-a4 da coleção Ports.

Crie o /usr/local/libexec/enscript com este conteúdo:

```
#!/bin/sh
/usr/local/bin/enscript -o -
```

Defina as permissões e torne-o executável:

```
# chmod 555 /usr/local/libexec/enscript
```

Modifique o /etc/printcap para usar o novo filtro:

```
:if=/usr/local/libexec/enscript:\
```

Teste o filtro imprimindo um arquivo de texto simples.

9.5.3.3. Imprimindo PostScript™ em impressoras PCL

Muitos programas produzem documentos PostScript™. No entanto, impressoras baratas geralmente só entendem texto simples ou PCL. Este filtro converte os arquivos PostScript™ para o formato PCL antes de enviá-los para a impressora.

Instale o interpretador de PostScript™ Ghostscript, print/ghostscript9, através da Coleção de Ports.

Crie o /usr/local/libexec/ps2pcl com este conteúdo:

```
#!/bin/sh
/usr/local/bin/gs -dSAFER -dNOPAUSE -dBATCH -q -sDEVICE=ljet4 -sOutputFile=- -
```

Defina as permissões e torne-o executável:

```
# chmod 555 /usr/local/libexec/ps2pcl
```

A entrada PostScript™ enviada para este script será processada e convertida em PCL antes de ser enviada para a impressora.

Modifique o /etc/printcap para usar este novo filtro de entrada:

```
:if=/usr/local/libexec/ps2pcl:\
```

Teste o filtro enviando um pequeno programa PostScript™ para ele:

```
% printf "%%\!PS \n /Helvetica findfont 18 scalefont setfont \
72 432 moveto (PostScript printing successful.) show showpage \004" | lpr
```

9.5.3.4. Filtros Inteligentes

Um filtro que detecta o tipo de entrada e converte automaticamente para o formato correto da impressora pode ser muito conveniente. Os dois primeiros caracteres de um arquivo PostScript™ são geralmente `%!`. Um filtro pode detectar esses dois caracteres. Os arquivos PostScript™ podem ser

enviados de forma inalterada para uma impressora PostScript™. Arquivos de texto podem ser convertidos para PostScript™ com o Enscript como mostrado anteriormente. Crie o /usr/local/libexec/psif com este conteúdo:

```
#!/bin/sh
#
#  psif - Print PostScript or plain text on a PostScript printer
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

case "$first_two_chars" in
%!)
    # %! : PostScript job, print it.
    echo "$first_line" && cat && exit 0
    exit 2
    ;;
*)
    # otherwise, format with enscript
    ( echo "$first_line"; cat ) | /usr/local/bin/enscript -o - && exit 0
    exit 2
    ;;
esac
```

Defina as permissões e torne-o executável:

```
# chmod 555 /usr/local/libexec/psif
```

Modifique o /etc/printcap para usar este novo filtro de entrada:

```
:if=/usr/local/libexec/psif:\
```

Teste o filtro imprimindo PostScript™ e arquivos de texto simples.

9.5.4. Múltiplas filas

As entradas no /etc/printcap são na verdade definições de *filas*. Pode haver mais de uma fila para uma única impressora. Quando combinadas com filtros, múltiplas filas fornecem aos usuários um maior controle sobre como seus trabalhos são impressos.

Por exemplo, considere uma impressora laser PostScript™ em rede num escritório. A maioria dos usuários deseja imprimir texto simples, mas alguns usuários avançados querem poder imprimir diretamente os arquivos PostScript™. Duas entradas podem ser criadas para a mesma impressora no /etc/printcap:

```
textprinter:\
```

```
    :lp=9100@officelaser:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/textprinter:\
    :if=/usr/local/libexec/enscript:\
    :lf=/var/log/lpd-errs:

psprinter:\
    :lp=9100@officelaser:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/psprinter:\
    :lf=/var/log/lpd-errs:
```

Os documentos enviados para a fila `textprinter` serão formatados pelo filtro /usr/local/libexec/enscript mostrado em um exemplo anterior. Usuários avançados podem imprimir arquivos PostScript™ em `psprinter`, onde nenhuma filtragem é feita.

Esta técnica de múltiplas filas pode ser usada para fornecer acesso direto a todos os tipos de recursos da impressora. Uma impressora com um duplexador pode usar duas filas, uma para impressões em apenas um lado da folha e outra com um filtro que envia a seqüência de comandos para habilitar a impressão frente e verso e, em seguida, envia o arquivo recebido.

9.5.5. Monitoramento e controle de impressão

Vários utilitários estão disponíveis para monitorar trabalhos de impressão e verificar e controlar a operação da impressora.

9.5.5.1. lpq(1)

O lpq(1) mostra o status das tarefas de impressão de um usuário. Trabalhos de impressão de outros usuários não são mostrados.

Mostra os trabalhos pendentes do usuário atual em uma única impressora:

```
% lpq -Plp
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     0    (standard input)                      12792 bytes
```

Mostra os trabalhos pendentes do usuário atual em todas as impressoras:

```
% lpq -a
lp:
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     1    (standard input)                      27320 bytes

laser:
Rank   Owner      Job  Files                                 Total Size
```

```
1st    jsmith     287  (standard input)                    22443 bytes
```

9.5.5.2. lprm(1)

O lprm(1) é usado para remover trabalhos de impressão. Usuários normais só podem remover seus próprios trabalhos. O `root` pode remover qualquer um ou todos os trabalhos.

Remova todos os trabalhos pendentes de uma impressora:

```
# lprm -Plp -
dfA002smithy dequeued
cfA002smithy dequeued
dfA003smithy dequeued
cfA003smithy dequeued
dfA004smithy dequeued
cfA004smithy dequeued
```

Remova um único trabalho de uma impressora. O lpq(1) é usado para encontrar o número do trabalho.

```
% lpq
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     5    (standard input)                      12188 bytes
% lprm -Plp 5
dfA005smithy dequeued
cfA005smithy dequeued
```

9.5.5.3. lpc(8)

O lpc(8) é usado para verificar e modificar o status da impressora. O `lpc` é seguido por um comando e um nome de impressora opcional. O parâmetro `all` pode ser usado em vez de um nome de impressora específico, e o comando será aplicado a todas as impressoras. Usuários normais podem visualizar o status com lpc(8). Somente o `class="username">root` pode usar comandos que modificam o status da impressora.

Mostrar o status de todas as impressoras:

```
% lpc status all
lp:
    queuing is enabled
    printing is enabled
    1 entry in spool area
    printer idle
laser:
    queuing is enabled
    printing is enabled
    1 entry in spool area
```

```
    waiting for laser to come up
```

Impedindo que uma impressora aceite novos trabalhos e fazendo com que ela comece a aceitar novos trabalhos novamente:

```
# lpc disable lp
lp:
    queuing disabled
# lpc enable lp
lp:
    queuing enabled
```

Pare de imprimir, mas continue aceitando novos trabalhos. Em seguida, comece a imprimir novamente:

```
# lpc stop lp
lp:
    printing disabled
# lpc start lp
lp:
    printing enabled
    daemon started
```

Reinicie uma impressora após alguma condição de erro:

```
# lpc restart lp
lp:
    no daemon to abort
    printing enabled
    daemon restarted
```

Desative a fila de impressão e desative a impressão, com uma mensagem para explicar o problema aos usuários:

```
# lpc down lp Repair parts will arrive on Monday
lp:
    printer and queuing disabled
    status message is now: Repair parts will arrive on Monday
```

Reative uma impressora que esteja inativa:

```
# lpc up lp
lp:
    printing enabled
```

```
    daemon started
```

Veja lpc(8) para mais comandos e opções.

9.5.6. Impressoras Compartilhadas

As impressoras costumam ser compartilhadas por vários usuários em empresas e escolas. Recursos adicionais são fornecidos para tornar as impressoras compartilhadas mais convenientes.

9.5.6.1. Aliases

O nome da impressora é definido na primeira linha da entrada em /etc/printcap. Nomes adicionais, ou *aliases*, podem ser adicionados após esse nome. Os aliases são separados do nome e um do outro por barras verticais:

```
lp|repairsprinter|salesprinter:\
```

Os aliases podem ser usados no lugar do nome da impressora. Por exemplo, os usuários do departamento de vendas imprimem em sua impressora com

```
% lpr -Psalesprinter sales-report.txt
```

Usuários do departamento de Reparos podem imprimir na *sua* impressora com

```
% lpr -Prepairsprinter repairs-report.txt
```

Todos os documentos são impressos nessa única impressora. Quando o departamento de vendas cresce o suficiente para precisar de sua própria impressora, o alias pode ser removido da entrada da impressora compartilhada e usado como o nome de uma nova impressora. Os usuários nos dois departamentos continuam usando os mesmos comandos, mas os documentos de vendas são enviados para a nova impressora.

9.5.6.2. Páginas de cabeçalho

Pode ser difícil para os usuários localizarem seus documentos na pilha de páginas produzidas por uma impressora compartilhada ocupada. *Páginas de cabeçalho* foram criadas para resolver este problema. Uma página de cabeçalho com o nome de usuário e o nome do documento é impressa antes de cada trabalho de impressão. Estas páginas são por vezes chamadas de páginas *banner* ou *separadoras*.

A ativação das páginas de cabeçalho é diferente, dependendo se a impressora está conectada diretamente ao computador com um cabo USB, paralelo ou serial, ou se está conectada remotamente por uma rede.

As páginas de cabeçalho em impressoras conectadas diretamente são ativadas removendo-se a linha `:sh:\` (Suprimir Cabeçalho) da entrada no /etc/printcap. Essas páginas de cabeçalho usam

apenas caracteres de feed de linha para novas linhas. Algumas impressoras precisarão do filtro /usr/shared/examples/printing/hpif para evitar imprimir o texto em escada. O filtro configura impressoras PCL para imprimir retornos de carro e alimentações de linha quando um feed de linha é recebido.

As páginas de cabeçalho das impressoras de rede devem ser configuradas na própria impressora. Entradas de página de cabeçalho no /etc/printcap são ignoradas. As configurações geralmente estão disponíveis no painel frontal da impressora ou em uma página da web de configuração acessível com um navegador da web.

9.5.7. Referências

Arquivos de exemplo: /usr/shared/examples/printing/.

O *Manual do Spooler de Impressora de Linha do 4.3BSD*, /usr/shared/doc/smm/07.lpd/paper.ascii.gz.

Páginas de manual: printcap(5), lpd(8), lpr(1), lpc(8), lprm(1), lpq(1).

9.6. Outros sistemas de impressão

Vários outros sistemas de impressão estão disponíveis, além do lpd(8). Esses sistemas oferecem suporte para outros protocolos ou recursos adicionais.

9.6.1. CUPS (Sistema de impressão comum UNIX™)

O CUPS é um sistema de impressão popular disponível em muitos sistemas operacionais. Usar o CUPS no FreeBSD está documentado em um artigo separado: CUPS

9.6.2. HPLIP

A Hewlett Packard fornece um sistema de impressão que suporta muitas de suas impressoras a jato de tinta e laser. O port é o print/hplip. A página principal da web está em http://hplipopensource.com/hplip-web/index.html. O port lida com todos os detalhes de instalação no FreeBSD. As informações de configuração são mostradas em http://hplipopensource.com/hplip-web/install/manual/hp_setup.html.

9.6.3. LPRng

O LPRng foi desenvolvido como uma alternativa aprimorada para o lpd(8). O port é sysutils/LPRng. Para detalhes e documentação, veja https://lprng.sourceforge.net/.

Capítulo 10. Compatibilidade binária com o Linux®

10.1. Sinopse

O FreeBSD fornece compatibilidade binária com o Linux™, permitindo que os usuários instalem e executem a maioria dos binários do Linux™ em um sistema FreeBSD sem ter que primeiro modificar o binário. Foi até relatado que, em algumas situações, os binários Linux™ têm melhor desempenho no FreeBSD do que no Linux™.

No entanto, alguns recursos do sistema operacional específicos do Linux™ não são suportados no FreeBSD. Por exemplo, os binários Linux™ não funcionarão no FreeBSD se usarem chamadas específicas i386™, mesmo ativando o modo 8086 virtual.



O suporte para compatibilidade binária de 64 bits com o Linux™ foi adicionado no FreeBSD 10.3.

Depois de ler este capítulo, você saberá:

- Como habilitar a compatibilidade binária com o Linux™ em um sistema FreeBSD.
- Como instalar bibliotecas compartilhadas adicionais do Linux™.
- Como instalar aplicativos Linux™ em um sistema FreeBSD.
- Os detalhes de implementação da compatibilidade com o Linux™ no FreeBSD.

Antes de ler este capítulo, você deve:

- Saiber como instalar software adicional de terceiros.

10.2. Configurando a compatibilidade binária com o Linux™

Por padrão, as bibliotecas do Linux™ não estão instaladas e a compatibilidade binária com o Linux™ não está ativada. As bibliotecas Linux™ podem ser instaladas manualmente ou a partir da coleção de Ports do FreeBSD.

Antes de tentar compilar o port, carregue o módulo de kernel Linux™, caso contrário a compilação irá falhar:

```
# kldload linux
```

Para compatibilidade com 64 bits:

```
# kldload linux64
```

Para verificar se o módulo está carregado:

```
% kldstat
      Id Refs Address    Size     Name
      1    2 0xc0100000 16bdb8   kernel
      7    1 0xc24db000 d000     linux.ko
```

O pacote ou port emulators/linux_base-c7 é a maneira mais fácil de instalar um conjunto básico de bibliotecas e binários do Linux™ em um sistema FreeBSD. Para instalar o port:

```
# pkg install emulators/linux_base-c7
```

Para que a compatibilidade com o Linux™ seja ativada durante a inicialização, adicione esta linha ao /etc/rc.conf:

```
linux_enable="YES"
```

Em máquinas de 64 bits, o /etc/rc.d/abi carregará automaticamente o módulo para emulação de 64 bits.

Como a camada de compatibilidade binária do Linux™ ganhou suporte para a execução de binários Linux™ de 32 e 64 bits (em hosts x86 de 64 bits), não é mais possível vincular estaticamente a funcionalidade de emulação a um kernel personalizado.

10.2.1. Instalando Bibliotecas Adicionais Manualmente

Se um aplicativo Linux™ reclamar sobre a falta de bibliotecas compartilhadas após configurar a compatibilidade binária do Linux™, determine quais bibliotecas compartilhadas o Linux™ precisa e instale-as manualmente.

A partir de um sistema Linux™, o `ldd` pode ser usado para determinar quais bibliotecas compartilhadas o aplicativo precisa. Por exemplo, para verificar quais bibliotecas compartilhadas o `linuxdoom` precisa, execute este comando a partir de um sistema Linux™ que tenha o Doom instalado:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

Então, copie todos os arquivos listados na última coluna da saída do comando no sistema Linux™ para o diretório /compat/linux no sistema FreeBSD. Depois de copiados, crie links simbólicos para os nomes na primeira coluna. Este exemplo irá resultar nos seguintes arquivos no sistema FreeBSD:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
```

```
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Se uma biblioteca compartilhada Linux™ já existir com um número de revisão principal correspondente à primeira coluna da saída do comando `ldd`, ela não precisará ser copiada para a arquivo nomeado na última coluna, pois a biblioteca existente deve funcionar. No entanto é aconselhável copiar a biblioteca compartilhada se for uma versão mais nova. O arquivo antigo pode ser removido, desde que o link simbólico aponte para o novo.

Por exemplo, essas bibliotecas já existem no sistema FreeBSD:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

e o `ldd` indica que um binário requer uma versão posterior:

```
libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
```

Como a biblioteca existente tem apenas uma ou duas versões desatualizadas no último dígito, o programa ainda deve funcionar com a versão um pouco mais antiga. No entanto, é seguro substituir o libc.so existente pela versão mais nova:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Geralmente, será necessário procurar as bibliotecas compartilhadas que os binários do Linux™ dependem apenas das primeiras vezes que um programa Linux™ é instalado no FreeBSD. Depois de um tempo, haverá um conjunto suficiente de bibliotecas Linux™ compartilhadas no sistema para poder executar binários Linux™ atualizados sem qualquer trabalho extra.

10.2.2. Instalando os binários Linux™ELF

Os binários ELF requerem por vezes um passo extra. Quando um binário ELF sem marca for executado, ele gerará uma mensagem de erro:

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

Para ajudar o kernel do FreeBSD a distinguir entre um binário do FreeBSD ELF e um binário Linux™, use brandelf(1):

```
% brandelf -t Linux my-linux-elf-binary
```

Como o conjunto de ferramentas GNU coloca as informações de branding apropriadas em binários ELF automaticamente, essa etapa geralmente não é necessária.

10.2.3. Instalando um aplicativo baseado em Linux™RPM

Para instalar um aplicativo baseado em Linux™RPM, primeiro instale o pacote ou o port archivers/rpm4. Uma vez instalado, o usuário `root` pode usar este comando para instalar um .rpm:

```
# cd /compat/linux
# rpm2cpio < /path/to/linux.archive.rpm | cpio -id
```

Se necessário, use o `brandelf` nos binários ELF instalados . Observe que isso impedirá uma desinstalação limpa.

10.2.4. Configurando o Resolver do Hostname

Se o DNS não funcionar ou este erro aparecer:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

configure o /compat/linux/etc/host.conf como segue:

```
order hosts, bind
multi on
```

Isso especifica que o /etc/hosts deve ser pesquisado primeiro e o DNS deve ser pesquisado em segundo lugar. Quando o /compat/linux/etc/host.conf não existe, os aplicativos Linux™ usam o /etc/host.conf e avisam sobre a sintaxe incompatível do FreeBSD. Remova o `bind` se um servidor de nomes não estiver configurado usando o /etc/resolv.conf.

10.3. Tópicos Avançados

Esta seção descreve como funciona a compatibilidade binária com o Linux™ e é baseada em um email escrito para Lista de discussão do chat do FreeBSD por Terry Lambert tlambert@primenet.com (Message ID: <199906020108.SAA07001@usr09.primenet.com>).

O FreeBSD tem uma abstração chamada "loader de classes de execução". Esta é uma cunha na chamada de sistema execve(2) .

Historicamente, o loader UNIX™ examinava o número mágico (geralmente os primeiros 4 ou 8 bytes do arquivo) para ver se era um binário conhecido pelo sistema e, em caso afirmativo, invocava o loader binário.

Se o arquivo não fosse o tipo binário adequado para o sistema, a chamada execve(2) retornava uma falha, e o shell tentava iniciar a execução do mesmo como um comando do shell. A suposição era um padrão de "qualquer que seja o shell atual".

Posteriormente, foi feito um hack para que o sh(1) examinasse os dois primeiros caracteres e se eles fossem `:\n`, ele invocava o shell csh(1) em seu lugar.

O FreeBSD tem uma lista de loaders, em vez de um único loader, com um fallback para o loader `#!` para executar interpretadores de shell ou scripts de shell.

Para o suporte ao Linux™ABI, o FreeBSD vê o número mágico como um binário ELF. O loader ELF procura por uma *marca* especializada, que é uma seção de comentários na imagem ELF e que não esteja presente nos binários ELF SVR4/Solaris™.

Para que os binários Linux™ funcionem, eles devem ser *marcados* como tipo `Linux` usando o comando brandelf(1):

```
# brandelf -t Linux file
```

Quando o loader ELF vê a marca `Linux`, ele substitui um ponteiro na estrutura `proc`. Todas as chamadas do sistema são indexadas por esse ponteiro. Além disso, o processo é sinalizado para manipulação especial do vetor trap para o código de trampolim de sinal, e vários outros (menores) reparos que são manipulados pelo módulo do kernel Linux™.

O vetor de chamada do sistema Linux™ contém, entre outras coisas, uma lista de entradas `sysent[]` cujos endereços residem no módulo do kernel.

Quando uma chamada de sistema é acionada pelo binário Linux™, o código de interceptação desreferencia o ponteiro de função de chamada do sistema da estrutura `proc` e obtém a classe Linux™, não a FreeBSD, como ponto de entrada para a chamada do sistema.

O modo Linux™ procura fazer *reroots* dinamicamente. Isso é, na verdade, equivalente ao `union` para montagens de sistema de arquivos. Primeiro, é feita uma tentativa de procurar o arquivo em /compat/linux/original-path. Se isso falhar, a pesquisa será feita em /original-path. Isso garante que os binários que exigem outros binários possam ser executados. Por exemplo, o conjunto de ferramentas Linux™ pode ser executado sob o suporte da Linux™ABI. Isso também significa que os binários Linux™ podem carregar e executar binários do FreeBSD, se não houver binários Linux™ correspondentes, e que o comando uname(1) pode ser colocado na árvore de diretórios /compat/linux para garantir que os binários Linux™ não possam dizer que não estão rodando em Linux™.

De fato, existe um kernel Linux™ no kernel do FreeBSD. As várias funções subjacentes que implementam todos os serviços fornecidos pelo kernel são idênticas às entradas da tabela de chamada do sistema FreeBSD, e às entradas da tabela de chamada do sistema Linux™: operações do sistema de arquivos, operações de memória virtual, entrega de sinal e System V IPC. A única diferença é que os binários do FreeBSD obtêm as funções de *cola* do FreeBSD, e os binários Linux™ recebem as funções de *cola* do Linux™. As funções de *cola* do FreeBSD estão estaticamente ligadas ao kernel, e as funções de *cola* do Linux ™ podem ser estaticamente ligadas, ou podem ser acessadas através de um módulo do kernel.

Tecnicamente, isso não é realmente emulação, é uma implementação de ABI. Às vezes é chamado de "emulação™ Linux " porque a implementação foi feita num momento em que não havia outra palavra para descrever o que estava acontecendo. Dizer que o FreeBSD executava os binários do Linux™ não era verdade, já que o código não era compilado nele.

Parte III: Administração do Sistema

Os capítulos restantes cobrem todos os aspectos da administração do sistema FreeBSD. Cada capítulo começa descrevendo o que será aprendido como resultado da leitura do capítulo e também detalha o que o leitor deve saber antes de abordar o material.

Estes capítulos são projetados para serem lidos conforme as informações são necessárias. Eles não precisam ser lidos em nenhuma ordem específica, nem todos precisam ser lidos antes de começar a usar o FreeBSD.

Capítulo 11. Configuração e Ajuste

11.1. Sinopse

Um dos aspectos importantes do FreeBSD é a configuração adequada do sistema. Este capítulo explica muito do processo de configuração do FreeBSD, incluindo alguns dos parâmetros que podem ser configurados para ajustar um sistema FreeBSD.

Depois de ler este capítulo, você saberá:

- O básico da configuração do rc.conf e dos scripts de inicialização /usr/local/etc/rc.d.
- Como configurar e testar uma placa de rede.
- Como configurar hosts virtuais em dispositivos de rede.
- Como usar os vários arquivos de configuração em /etc.
- Como ajustar o FreeBSD usando variáveis sysctl(8).
- Como ajustar o desempenho do disco e modificar as limitações do kernel.

Antes de ler este capítulo, você deve:

- Entender os fundamentos do UNIX™ e do FreeBSD (Fundamentos do FreeBSD).
- Estar familiarizado com os conceitos básicos de configuração e compilação do kernel (Configurando o kernel do FreeBSD).

11.2. Inicialização de Serviços

Muitos usuários instalam software de terceiros no FreeBSD a partir da coleção de Ports e precisam que os serviços instalados sejam iniciados durante a inicialização do sistema. Serviços como mail/postfix ou www/apache22 são apenas dois dos muitos pacotes de software que podem ser iniciados durante a inicialização do sistema. Esta seção explica os procedimentos disponíveis para iniciar o software de terceiros.

No FreeBSD, a maioria dos serviços incluídos, como o cron(8), são iniciados através dos scripts de inicialização do sistema.

11.2.1. Configuração Estendida dos Aplicativos

Agora que o FreeBSD inclui o rc.d, a configuração da inicialização do aplicativo é mais fácil e fornece mais recursos. Usando as palavras-chave discutidas em Gerenciando Serviços no FreeBSD, os aplicativos podem ser configurados para iniciar depois de certos outros serviços e flags extras poderem ser passadas através do /etc/rc.conf no lugar de sinalizadores codificados no script de inicialização. Um script básico pode ser semelhante ao seguinte:

```
#!/bin/sh
#
# PROVIDE: utility
```

```
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=utility
rcvar=utility_enable

command="/usr/local/sbin/utility"

load_rc_config $name

#
# DO NOT CHANGE THESE DEFAULT VALUES HERE
# SET THEM IN THE /etc/rc.conf FILE
#
utility_enable=${utility_enable-"NO"}
pidfile=${utility_pidfile-"/var/run/utility.pid"}

run_rc_command "$1"
```

Este script irá garantir que o `utilitário` fornecido será iniciado após o pseudo-serviço `DAEMON`. Ele também fornece um método para definir e rastrear o ID do processo (PID).

Esta aplicação poderia então ter a seguinte linha colocada no /etc/rc.conf:

```
utility_enable="YES"
```

Este método permite a manipulação mais fácil de argumentos de linha de comando, inclusão das funções padrões fornecidas em /etc/rc.subr, compatibilidade com o rcorder(8), e fornece uma configuração mais fácil via rc.conf.

11.2.2. Usando o Services para Inicializar Serviços

Outros serviços podem ser iniciados usando o inetd(8). O uso do inetd(8) e sua configuração é descrita em profundidade em O super-servidor inetd.

Em alguns casos, pode fazer mais sentido usar o cron(8) para iniciar os serviços do sistema. Esta abordagem tem várias vantagens, pois o cron(8) executa estes processos como o proprietário do crontab(5). Isto permite que usuários regulares iniciem e mantenham seus próprios aplicativos.

O recurso `@reboot` do cron(8), pode ser usado no lugar da especificação de hora. Isso faz com que o job seja executado quando cron(8) é iniciado, normalmente durante a inicialização do sistema.

11.3. Configurando o cron(8)

Um dos utilitários mais úteis no FreeBSD é o cron. Este utilitário é executado em segundo plano e verifica regularmente o /etc/crontab para que as tarefas sejam executadas e procura /var/cron/tabs

para arquivos crontab personalizados. Estes arquivos são usados para planejar tarefas que o cron executa nos horários especificados. Cada entrada em um crontab define uma tarefa para ser executada e é conhecida como uma *tarefa do cron*.

Dois tipos diferentes de arquivos de configuração são usados: o crontab do sistema, que não deve ser modificado, e crontabs de usuário, que podem ser criados e editados conforme necessário. O formato usado por esses arquivos está documentado em crontab(5). O formato do sistema crontab, /etc/crontab inclui uma coluna `who` que não existe nos crontabs de usuário. No crontab do sistema , o cron executa o comando como o usuário especificado nesta coluna. Em um crontab de usuário, todos os comandos são executados como o usuário que criou o crontab.

Os crontabs de usuário permitem que usuários individuais programem suas próprias tarefas. O usuário `root` também pode ter um crontab de usuário que pode ser usado para agendar tarefas que não existem no crontab do sistema .

Aqui está uma entrada de amostra do crontab do sistema, /etc/crontab:

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
dbaio $
#①
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ②
#
#minute hour    mday    month   wday    who command ③
#
*/5 *   *   *   *   root    /usr/libexec/atrun ④
```

① Linhas que começam com o caractere `#` são comentários. Um comentário pode ser colocado no arquivo como um lembrete do que uma ação faz e do porque a sua execução é desejada. Comentários não podem estar na mesma linha que um comando ou então serão interpretados como parte do comando; eles devem estar em uma nova linha. Linhas em branco são ignoradas.

② O caractere igual (`=`) é usado para definir qualquer configuração de ambiente. Neste exemplo, ele é usado para definir o `SHELL` e o `PATH`. Se o `SHELL` for omitido, o cron usará o shell Bourne padrão. Se o `PATH` for omitido, o caminho completo deverá ser fornecido ao comando ou script a ser executado.

③ Esta linha define os sete campos usados em um crontab do sistema: `minute`, `hora`, `mday`, `month`, `wday`, `who` e `command`. O campo `minute` é o tempo em minutos quando o comando especificado será executado, a `hour` é a hora em que o comando especificado será executado, o `mday` é o dia do mês, `month` é o mês e `wday` é o dia da semana. Estes campos devem ser valores numéricos, representando o relógio de vinte e quatro horas, ou um `*`, representando todos os valores desse campo. O campo `who` existe somente no crontab do sistema e especifica com qual usuário o comando deve ser executado. O último campo é o comando a ser executado.

④ Esta entrada define os valores para este trabalho do cron. O `/5, seguido por vários outros caracteres`, especifica que `/usr/libexec/atrun` é invocado pelo `root` a cada cinco minutos de cada hora, de cada dia e dia da semana, de cada mês.Comandos podem incluir qualquer número de

opções. No entanto, os comandos que se estendem a várias linhas precisam ser quebrados com o caractere de continuação da barra invertida "\".

11.3.1. Criando um Crontab de Usuário

Para criar um crontab de usuário, invoque o `crontab` no modo editor:

```
% crontab -e
```

Isto irá abrir o crontab do usuário usando o editor de texto padrão. A primeira vez que um usuário executa este comando, ele abre um arquivo vazio. Depois que um usuário cria um crontab, esse comando abrirá este arquivo para edição.

É útil adicionar estas linhas a parte superior do arquivo crontab para configurar as variáveis de ambiente e lembrar os significados dos campos no crontab:

```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
# Order of crontab fields
# minute    hour    mday    month   wday    command
```

Em seguida, adicione uma linha para cada comando ou script a ser executado, especificando o horário para executar o comando. Este exemplo executa o script de shell Bourne personalizado especificado todos os dias às duas da tarde. Como o caminho para o script não está especificado em `PATH`, o caminho completo para o script é fornecido:

```
0   14  *   *   *   /usr/home/dru/bin/mycustomscript.sh
```



Antes de usar um script personalizado, verifique se ele é executável e teste-o com o conjunto limitado de variáveis de ambiente definidas pelo cron. Para replicar o ambiente que seria usado para executar a entrada do cron acima, use:

```
env -i SHELL=/bin/sh PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
HOME=/home/dru LOGNAME=dru /usr/home/dru/bin/mycustomscript.sh
```

O ambiente definido pelo cron é discutido em crontab(5). Verificar se os scripts operam corretamente em um ambiente cron é especialmente importante se incluírem quaisquer comandos que excluam arquivos usando curingas.

Quando terminar de editar o crontab, salve o arquivo. Ele será instalado automaticamente e o cron lerá o crontab e executará seus cron jobs nos horários especificados. Para listar as tarefas agendadas em um crontab, use este comando:

```
% crontab -l
```

```
0   14  *   *   *   /usr/home/dru/bin/mycustomscript.sh
```

Para remover todas as tarefas cron em um crontab de usuário:

```
% crontab -r
remove crontab for dru? y
```

11.4. Gerenciando Serviços no FreeBSD

O FreeBSD usa o sistema rc(8) de scripts de inicialização durante a inicialização do sistema e para gerenciar serviços. Os scripts listados em /etc/rc.d fornecem serviços básicos que podem ser controlados com `start`, `stop` e `restart` opções para service(8). Por exemplo, sshd(8) pode ser reiniciado com o seguinte comando:

```
# service sshd restart
```

Este procedimento pode ser usado para iniciar serviços em um sistema em execução. Os serviços serão iniciados automaticamente no momento da inicialização, conforme especificado em rc.conf(5). Por exemplo, para ativar o natd(8) na inicialização do sistema, adicione a seguinte linha ao /etc/rc.conf:

```
natd_enable="YES"
```

Se uma linha `natd_enable="NO"` já estiver presente, altere o `NO` para `YES`. Os scripts rc(8) carregarão automaticamente todos os serviços dependentes durante a próxima inicialização, conforme descrito abaixo.

Como o sistema rc(8) é destinado principalmente a iniciar e parar serviços na inicialização do sistema e no tempo de desligamento, o `start`, as opções `stop` e `restart` somente executarão suas ações se a variável apropriada estiver configurada no /etc/rc.conf. Por exemplo, o `sshd restart` só funcionará se `sshd_enable` estiver definido como `YES` em /etc/rc.conf. Para `iniciar`, `parar` ou `reiniciar` um serviço independente das configurações em /etc/rc.conf, estes comandos deve ser prefixado com "one". Por exemplo, para reiniciar sshd(8) independentemente da configuração atual do /etc/rc.conf, execute o seguinte comando:

```
# service sshd onerestart
```

Para verificar se um serviço está habilitado em /etc/rc.conf, execute o script apropriado rc(8) com `rcvar`. Este exemplo verifica se o sshd(8) está habilitado no /etc/rc.conf:

```
# service sshd rcvar
# sshd
#
```

```
sshd_enable="YES"
#   (default: "")
```



A linha `# sshd` é gerada pelo comando acima, não pelo console do `root`.

Para determinar se um serviço está ou não em execução, use `status`. Por exemplo, para verificar se o sshd(8) está em execução:

```
# service sshd status
sshd is running as pid 433.
```

Em alguns casos, também é possível fazer o `reload` denum serviço. Isso tenta enviar um sinal para um serviço individual, forçando o serviço a recarregar seus arquivos de configuração. Na maioria dos casos, isso significa enviar ao serviço um sinal `SIGHUP`. O suporte para esse recurso não está incluído para todos os serviços.

O sistema rc(8) é usado para serviços de rede e também contribui para a maior parte da inicialização do sistema. Por exemplo, quando o script /etc/rc.d/bgfsck é executado, ele imprime a seguinte mensagem:

```
Starting background file system checks in 60 seconds.
```

Esse script é usado para verificações do sistema de arquivos em segundo plano, que ocorrem apenas durante a inicialização do sistema.

Muitos serviços do sistema dependem de outros serviços para funcionar corretamente. Por exemplo, o yp(8) e outros serviços baseados em RPC podem falhar ao iniciar até que o rpcbind(8) seja iniciado. Para resolver esse problema, informações sobre dependências e outros meta-dados são incluídas nos comentários na parte superior de cada script de inicialização. O programa rcorder(8) é usado para analisar esses comentários durante a inicialização do sistema para determinar a ordem na qual os serviços do sistema devem ser invocados para satisfazer as dependências.

A seguinte palavra-chave deve ser incluída em todos os scripts de inicialização, conforme exigido pelo rc.subr(8) para "habilitar" o script de inicialização:

- `PROVIDE`: Especifica os serviços que este arquivo fornece.

As seguintes palavras-chave podem ser incluídas na parte superior de cada script de inicialização. Eles não são estritamente necessárias, mas são úteis como sugestões para rcorder(8):

- `REQUIRE`: lista os serviços necessários para este serviço. O script que contém esta palavra chave será executado *após* os serviços especificados.
- `BEFORE`: lista os serviços que dependem deste serviço. O script que contém esta palavra chave será executado *antes* dos serviços especificados.

Ao definir cuidadosamente essas palavras-chave para cada script de inicialização, um

administrador passa a ter um nível refinado de controle da ordem de inicialização dos scripts, sem a necessidade dos "runlevels" usados por alguns sistemas operacionais UNIX™.

Informações adicionais podem ser encontradas em rc(8) e rc.subr(8). Consulte este artigo para obter instruções sobre como criar um script rc(8) personalizado.

11.4.1. Gerenciando a configuração específica do sistema

A localização principal das informações de configuração do sistema é arquivo /etc/rc.conf. Este arquivo contém uma ampla gama de informações de configuração e é lido na inicialização do sistema para configurar o sistema. Ele fornece as informações de configuração para os arquivos rc*.

As entradas em /etc/rc.conf substituem as configurações padrões em /etc/defaults/rc.conf. O arquivo contendo as configurações padrões não deve ser editado. Ao invés disso, todas as alterações específicas do sistema devem ser feitas em /etc/rc.conf.

Várias estratégias podem ser aplicadas em aplicativos em cluster para separar as configurações que afetam todo o site da configuração específica do sistema para reduzir a sobrecarga de administração. A abordagem recomendada é colocar a configuração específica do sistema em /etc/rc.conf.local. Por exemplo, estas entradas em /etc/rc.conf aplicam-se a todos os sistemas:

```
sshd_enable="YES"
keyrate="fast"
defaultrouter="10.1.1.254"
```

Considerando que estas entradas em /etc/rc.conf.local se aplicam somente a este sistema:

```
hostname="node1.example.org"
ifconfig_fxp0="inet 10.1.1.1/8"
```

Distribua o /etc/rc.conf para cada sistema usando um aplicativo como o rsync ou o puppet, enquanto o /etc/rc.conf.local permanece único.

A atualização do sistema não sobrescreverá o /etc/rc.conf, portanto as informações de configuração do sistema não serão perdidas.



Ambos /etc/rc.conf e /etc/rc.conf.local são analisados pelo sh(1). Isto permite que os operadores do sistema criem cenários de configuração complexos. Consulte rc.conf(5) para obter mais informações sobre este tópico.

11.5. Configurando Placas de Interface de Rede

Adicionar e configurar uma placa de interface de rede (NIC) é uma tarefa comum para qualquer administrador do FreeBSD.

11.5.1. Localizando o Driver Correto

Primeiro, determine o modelo da NIC e o chip utilizado. O FreeBSD suporta uma ampla variedade de NICs. Verifique a lista de compatibilidade de hardware para a release do FreeBSD para ver se a NIC é suportada.

Se a NIC é suportada, determine o nome do driver do FreeBSD para a NIC. Consulte /usr/src/sys/conf/NOTES e /usr/src/sys/arch/conf/NOTES para a lista de Drivers NIC com algumas informações sobre os chipsets suportados. Em caso de dúvida, leia a página de manual do driver, pois ele fornecerá mais informações sobre o hardware suportado e quaisquer limitações conhecidas do driver.

Os drivers para as NICs comuns já estão presentes no kernel GENERIC, o que significa que a NIC deve ser verificada durante a inicialização. As mensagens de inicialização do sistema podem ser visualizadas digitando `more /var/run/dmesg.boot` e usando a barra de espaço para percorrer o texto. Neste exemplo, duas NICs Ethernet que utilizam o driver dc(4) estão presentes no sistema:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
miibus0: <MII bus> on dc0
bmtphy0: <BCM5201 10/100baseTX PHY> PHY 1 on miibus0
bmtphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc0: Ethernet address: 00:a0:cc:da:da:da
dc0: [ITHREAD]
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
miibus1: <MII bus> on dc1
bmtphy1: <BCM5201 10/100baseTX PHY> PHY 1 on miibus1
bmtphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: Ethernet address: 00:a0:cc:da:da:db
dc1: [ITHREAD]
```

Se o driver da NIC não estiver presente em GENERIC, mas houver um driver disponível, o driver precisará ser carregado antes que a NIC possa ser configurada e usada. Isso pode ser feito de duas maneiras:

- A maneira mais fácil é carregar um módulo do kernel para a NIC usando o kldload(8). Para carregar automaticamente o driver no momento da inicialização, adicione a linha apropriada ao /boot/loader.conf. Nem todos os drivers NIC estão disponíveis como módulos.
- Como alternativa, compile estaticamente o suporte para a NIC em um kernel personalizado. Consulte /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES e a página de manual do driver para determinar qual linha adicionar ao arquivo de configuração do kernel personalizado. Para mais informações sobre como recompilar o kernel, consulte Configurando o kernel do FreeBSD. Se a NIC foi detectada na inicialização, o kernel não precisa ser recompilado.

11.5.1.1. Utilizando os Drivers Windows™NDIS

Infelizmente, ainda existem muitos fornecedores que não fornecem esquemas para seus drivers

para a comunidade de código aberto porque consideram essas informações como segredos comerciais. Consequentemente, os desenvolvedores do FreeBSD e de outros sistemas operacionais são deixados com duas opções: desenvolver os drivers por um processo longo e complexo de engenharia reversa ou usar os binários de drivers existentes disponíveis para plataforma Microsoft™ Windows™.

O FreeBSD fornece suporte "nativo" para a especificação de interface de driver de rede (NDIS). Ele inclui o ndisgen(8) que pode ser utilizado para converter um driver Windows™ XP num formato que pode ser usado no FreeBSD. Como o driver ndis(4) usa um binário Windows™ XP, ele só é executado em sistemas i386™ e amd64. Dispositivos PCI, CardBus, PCMCIA e USB são suportados.

Para usar o ndisgen(8), três coisas são necessárias:

1. Código-fonte do kernel do FreeBSD.
2. Um binário do driver do Windows™ XP com uma extensão .SYS.
3. Um arquivo de configuração do driver do Windows™ XP com uma extensão .INF.

Faça o download dos arquivos .SYS e .INF para a NIC específica. Geralmente, eles podem ser encontrados no CD do driver ou no site do fornecedor. Os exemplos a seguir usam o W32DRIVER.SYS e o W32DRIVER.INF.

A largura do bit do driver deve corresponder à versão do FreeBSD. Para FreeBSD/i386, use um driver de 32 bits Windows™. Para o FreeBSD/amd64, é necessário um driver de 64 bits do Windows™.

O próximo passo é compilar o binário do driver em um módulo do kernel carregável. Como `root`, use ndisgen(8):

```
# ndisgen /path/to/W32DRIVER.INF /path/to/W32DRIVER.SYS
```

Este comando é interativo e solicita qualquer informação extra necessária. Um novo módulo do kernel será gerado no diretório atual. Use kldload(8) para carregar o novo módulo:

```
# kldload ./W32DRIVER_SYS.ko
```

Além do módulo do kernel gerado, os módulos ndis.ko e if_ndis.ko devem ser carregados. Isso deve acontecer automaticamente quando qualquer módulo que dependa do ndis(4) for carregado. Caso contrário, carregue-os manualmente, usando os seguintes comandos:

```
# kldload ndis
# kldload if_ndis
```

O primeiro comando carrega o wrapper do driver da miniporta ndis(4) e o segundo carrega o driver NIC gerado.

Execute o comando dmesg(8) para ver se houve algum erro de carregamento. Se tudo correu bem, a

saída deve ser semelhante à seguinte:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

A partir daqui, o ndis0 pode ser configurado como qualquer outra NIC.

Para configurar o sistema para carregar os módulos ndis(4) no momento da inicialização, copie o módulo gerado, W32DRIVER_SYS.ko, para /boot/modules. Em seguida, adicione a seguinte linha ao /boot/loader.conf:

```
W32DRIVER_SYS_load="YES"
```

11.5.2. Configurando a placa de rede

Quando o driver correto é carregado para a NIC, a placa precisa ser configurada. Ele pode ter sido configurado no momento da instalação por bsdinstall(8).

Para exibir a configuração da NIC, digite o seguinte comando:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8802<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:db
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet 10baseT/UTP
        status: no carrier
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
```

Neste exemplo, os seguintes dispositivos foram exibidos:

- dc0: A primeira interface Ethernet.
- dc1: A segunda interface Ethernet.

- lo0: o dispositivo de loopback.

O FreeBSD usa o nome do driver seguido da ordem em que a placa é detectada na inicialização para nomear a NIC. Por exemplo, sis2 é a terceira NIC no sistema usando driver sis(4).

Neste exemplo, o dc0 está ativo e em execução. Os principais indicadores são:

1. `UP` significa que a placa está configurada e pronta.
2. A placa tem um endereço da Internet (`inet`), `192.168.1.3`.
3. Ela tem uma máscara de sub-rede válida (`netmask`), onde `0xffffff00` é o mesmo que `255.255.255.0` .
4. Tem um endereço de broadcast válido, `192.168.1.255`.
5. O endereço MAC da placa (`ether`) é `00:a0:cc:da:da:da`.
6. A seleção de mídia física está no modo de seleção automática (`media:Ethernet autoselect (100baseTX <full-duplex>)`). Neste exemplo, o dc1 está configurado para ser executado com a mídia `10baseT/UTP`. Para obter mais informações sobre tipos de mídia disponíveis para um driver, consulte sua página de manual.
7. O status do link (`status`) é `active`, indicando que o sinal da portadora foi detectado. Para dc1, o status `status: no carrier` é normal quando um cabo Ethernet não está conectado à placa.

Se a saída ifconfig(8) tivesse mostrado algo semelhante a:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=80008<VLAN_MTU,LINKSTATE>
    ether 00:a0:cc:da:da:da
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
```

isso indicaria que a placa não foi configurada.

A placa deve ser configurada como `root`. A configuração da NIC pode ser realizada a partir da linha de comando com o ifconfig(8), mas não persistirá após uma reinicialização, a menos que a configuração também seja adicionada ao /etc/rc.conf. Se um servidor DHCP estiver presente na LAN, basta adicionar esta linha:

```
ifconfig_dc0="DHCP"
```

Substitua *dc0* com o valor correto para o sistema.

A linha adicionada, então, segue as instruções dadas em Teste e solução de problemas.



Se a rede foi configurada durante a instalação, algumas entradas para a NIC podem já estar presentes. Verifique novamente o /etc/rc.conf antes de adicionar novas linhas.

Se não existir um servidor DHCP, a NIC deve ser configurada manualmente. Adicione uma linha para cada NIC presente no sistema, conforme mostrado neste exemplo:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Substitua dc0 e dc1 e as informações de endereço IP com os valores corretos para o sistema. Consulte a man page do driver, ifconfig(8) e rc.conf(5) para maiores detalhes sobre as opções permitidas e a sintaxe de /etc/rc.conf.

Se a rede não estiver usando DNS, edite o /etc/hosts para adicionar os nomes e endereços IP dos hosts na LAN, se eles ainda não estiverem lá. Para maiores informações, consulte hosts(5) e /usr/shared/examples/etc/hosts.



Se não houver um servidor DHCP e o acesso à Internet for necessário, configure manualmente o gateway padrão e o nameserver:

```
# echo 'defaultrouter="your_default_router"' >> /etc/rc.conf
# echo 'nameserver your_DNS_server' >> /etc/resolv.conf
```

11.5.3. Teste e solução de problemas

Uma vez que as alterações necessárias no /etc/rc.conf sejam salvas, uma reinicialização pode ser usada para testar a configuração de rede e verificar se o sistema é reiniciado sem nenhum erro. Como alternativa, aplique as configurações ao sistema de rede com este comando:

```
# service netif restart
```



Se um gateway padrão foi configurado no /etc/rc.conf, também execute este comando:

```
# service routing restart
```

Uma vez que o sistema de rede tiver sido reiniciado, teste as NIC.

11.5.3.1. Testando uma placa Ethernet

Para verificar se uma placa Ethernet está configurada corretamente, execute um ping(8) na própria interface e, em seguida, ping(8) outra máquina na LAN:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
```

```
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Para testar a resolução da rede, use o nome do host em vez do endereço IP. Se não houver nenhum servidor DNS na rede, o /etc/hosts deve ser configurado primeiro. Para este propósito, edite o /etc/hosts para adicionar os nomes e os endereços IP dos hosts na LAN, se eles ainda não estiverem lá . Para maiores informações, consulte hosts(5) e /usr/shared/examples/etc/hosts.

11.5.3.2. Solução de problemas

Ao solucionar problemas de configurações de hardware e software, verifique primeiro as coisas simples. O cabo de rede está conectado? Os serviços de rede estão configurados corretamente? O firewall está configurado corretamente? A NIC é suportada pelo FreeBSD? Antes de enviar um relatório de bug, sempre verifique as Notas de Hardware, atualize a versão do FreeBSD para a versão mais recente do STABLE, verifique os arquivos da lista de discussão e pesquise na Internet.

Se a placa funcionar, mas o desempenho for ruim, leia tuning(7). Além disso, verifique a configuração da rede, pois configurações de rede incorretas podem causar conexões lentas.

Alguns usuários experimentam uma ou duas mensagens de `device timeout`, o que é normal para algumas placas. Se eles continuarem ou forem incômodos, verifique se o dispositivo está em conflito com outro. Verifique novamente as conexões dos cabos. Considere tentar outra placa.

Para resolver erros de `watchdog timeout`, primeiro verifique o cabo de rede. Muitas placas requerem um slot PCI que suporte a masterização de barramento. Em algumas placas-mãe antigas, apenas um slot PCI permite, normalmente o slot 0. Verifique a NIC e a documentação da placa-mãe para determinar se esse pode ser o problema.

As mensagens `No route to host` ocorrem se o sistema não puder rotear um pacote para o host de destino. Isso pode acontecer se nenhuma rota padrão for especificada ou se um cabo for desconectado. Verifique a saída do `netstat -rn` e certifique-se de que haja uma rota válida para o

host. Se não houver, leia Gateways e Rotas.

As mensagens de erro `ping: sendto: Permission denied` são geralmente causadas por um firewall mal configurado. Se um firewall está habilitado no FreeBSD, mas nenhuma regra foi definida, a política padrão é negar todo o tráfego, mesmo o ping(8). Consulte Firewalls para maiores informações.

Às vezes, o desempenho da placa é ruim ou abaixo da média. Nesses casos, tente configurar o modo de seleção de mídia de `autoselect` para a seleção de mídia correta. Embora isso funcione para a maioria dos hardwares, isso pode ou não resolver o problema. Novamente, verifique todas as configurações de rede e consulte tuning(7).

11.6. Hosts Virtuais

Um uso comum do FreeBSD é a hospedagem de sites virtuais, onde um servidor aparece na rede como muitos servidores. Isso é conseguido atribuindo vários endereços de rede a uma única interface.

Uma determinada interface de rede tem um endereço "real" e pode ter qualquer número de endereços "alias". Esses aliases são normalmente adicionados colocando entradas de alias no /etc/rc.conf, como mostrado neste exemplo:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

As entradas de alias devem começar com `alias_0_` usando um número sequencial como `alias0`, `alias1` e assim por diante. O processo de configuração será interrompido no primeiro número ausente.

O cálculo de máscaras de alias é importante. Para uma determinada interface, deve haver um endereço que represente corretamente a máscara de rede da rede. Qualquer outro endereço dentro dessa rede deve ter uma máscara de rede toda de `1`s, expressa como `255.255.255.255` ou `0xffffffff`.

Por exemplo, considere o caso em que a interface fxp0 está conectada a duas redes: `10.1.1.0` com uma máscara de rede de `255.255.255.0` e `202.0.75.16` com uma máscara de rede de `255.255.255.240`. O sistema deve ser configurado para aparecer nos intervalos `10.1.1.1` até `10.1.1.5` e `202.0.75.17` até `202.0.75.20`. Somente o primeiro endereço em um determinado intervalo de rede deve ter uma máscara de rede real. Todo o resto de (`10.1.1.2` até `10.1.1.5` e de `202.0.75.18` até `202.0.75.20`) deve ser configurado com uma máscara de rede `255.255.255.255`.

As seguintes entradas /etc/rc.conf configuram o adaptador corretamente para este cenário:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
```

```
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

Uma maneira mais simples de expressar isso é com uma lista separada por espaço de intervalos de endereços IP. O primeiro endereço receberá a máscara de sub-rede indicada e os endereços adicionais terão uma máscara de sub-rede `255.255.255.255`.

```
ifconfig_fxp0_aliases="inet 10.1.1.1-5/24 inet 202.0.75.17-20/28"
```

11.7. Configurando o log do sistema

Gerar e ler logs do sistema é um aspecto importante da administração do sistema. As informações nos registros do sistema podem ser usadas para detectar problemas de hardware e software, bem como erros de configuração dos aplicativos e do sistema. Essas informações também desempenham um papel importante na auditoria de segurança e na resposta a incidentes. A maioria dos daemons e aplicativos do sistema geram entradas de log.

O FreeBSD fornece um registrador de sistema, o syslogd, para gerenciar o registro. Por padrão, o syslogd é iniciado quando o sistema é inicializado. Isto é controlado pela variável `syslogd_enable` no /etc/rc.conf. Existem vários argumentos de aplicação que podem ser definidos usando `syslogd_flags` no /etc/rc.conf. Consulte syslogd(8) para obter maiores informações sobre os argumentos disponíveis.

Esta seção descreve como configurar o criador de logs do sistema FreeBSD para log local e remoto e como executar a rotação de log e o gerenciamento de log.

11.7.1. Configurando os logs locais

O arquivo de configuração, /etc/syslog.conf, controla o que o syslogd faz com as entradas de log à medida que são recebidas. Existem vários parâmetros para controlar o tratamento de eventos recebidos. O *facility* descreve qual subsistema gerou a mensagem, como o kernel ou um daemon, e o *level* descreve a gravidade do evento que ocorreu. Isso possibilita configurar onde uma mensagem de log é registrada, dependendo da instalação e do nível. Também é possível executar uma ação dependendo do aplicativo que enviou a mensagem e, no caso de log remoto, o nome do host da máquina que gera o evento de log.

Este arquivo de configuração contém uma linha por ação, em que a sintaxe de cada linha é um campo seletor seguido por um campo de ação. A sintaxe do campo seletor é *facility.level*, que corresponderá às mensagens de log de *facility* no nível *level* ou superior. Também é possível adicionar um sinalizador de comparação opcional antes do nível para especificar mais precisamente o que está registrado. Vários campos seletores podem ser usados para a mesma ação e são separados por um ponto-e-vírgula (`;`). Usar `*` irá corresponder a tudo. O campo de ação indica para onde enviar a mensagem de log, como para um arquivo ou host de log remoto. Por exemplo, aqui está o syslog.conf padrão do FreeBSD:

```
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
```

```
dbaio $
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.
*.err;kern.warning;auth.notice;mail.crit                /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err   /var/log/messages
security.*                                      /var/log/security
auth.info;authpriv.info                         /var/log/auth.log
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
ftp.info                                        /var/log/xferlog
cron.*                                          /var/log/cron
!-devd
*.=debug                                        /var/log/debug.log
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost
# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice
# Uncomment this if you wish to see messages produced by devd
# !devd
# *.>=info
!ppp
*.*                                             /var/log/ppp.log
!*
```

Neste exemplo:

- A linha 8 combina todas as mensagens com um nível de `err` ou superior, bem como `kern.warning`, `auth.notice` e `mail.crit`, e envia essas mensagens de log para o console (/dev/console).
- A linha 12 combina todas as mensagens do recurso `mail` no nível `info` ou acima e registra as mensagens em /var/log/maillog.
- A linha 17 usa um sinalizador de comparação (`=`) para corresponder apenas as mensagens no nível `debug` e registrá-las em /var/log/debug.log.
- A linha 33 é um exemplo de uso de uma especificação de programa. Isso faz com que as regras que a seguem apenas sejam válidas para o programa especificado. Neste caso, somente as mensagens geradas pelo ppp são registradas em /var/log/ppp.log.

Os níveis disponíveis, na ordem dos mais para o menos críticos, são `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, and `debug`.

As facilities, em nenhuma ordem particular, são `auth`, `authpriv`, `console`, `cron`, `daemon`, `ftp`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp`, and `local0` through `local7`. Esteja ciente de que outros sistemas operacionais podem ter recursos diferentes.

Para registrar tudo do nível `notice` e superior para /var/log/daemon.log, adicione a seguinte entrada:

```
daemon.notice                                   /var/log/daemon.log
```

Para obter mais informações sobre os diferentes níveis e facilities, consulte syslog(3) e syslogd(8). Para maiores informações sobre /etc/syslog.conf, sua sintaxe e exemplos de uso mais avançados, veja syslog.conf(5).

11.7.2. Gerenciamento de log e rotação

Os arquivos de log podem crescer rapidamente, ocupando espaço em disco e dificultando a localização de informações úteis. O gerenciamento de log tenta atenuar isso. No FreeBSD, o newsyslog é usado para gerenciar arquivos de log. Este programa interno rotaciona e comprime periodicamente arquivos de log e, opcionalmente, cria arquivos de log ausentes e sinaliza os programas quando os arquivos de log são movidos. Os arquivos de log podem ser gerados pelo syslogd ou por qualquer outro programa que gere arquivos de log. Enquanto o newsyslog é normalmente executado a partir do cron(8), ele não é um daemon do sistema. Na configuração padrão, ele é executado a cada hora.

Para saber quais ações executar, o newsyslog lê seu arquivo de configuração, /etc/newsyslog.conf. Este arquivo contém uma linha para cada arquivo de log que o newsyslog gerencia. Cada linha indica o proprietário do arquivo, suas permissões, quando rotacionar esse arquivo, flags opcionais que afetam a rotação do log, como compactação, e programas para sinalizar quando o log é rotacionado. Aqui está a configuração padrão no FreeBSD:

```
# configuration file for newsyslog
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
dbaio $
#
# Entries which do not specify the '/pid_file' field will cause the
# syslogd process to be signalled when that log file is rotated.  This
# action is only appropriate for log files which are written to by the
# syslogd process (ie, files listed in /etc/syslog.conf).  If there
# is no process which needs to be signalled when a given log file is
# rotated, then the entry for that file should include the 'N' flag.
#
# The 'flags' field is one or more of the letters: BCDGJNUXZ or a '-'.
#
# Note: some sites will want to select more restrictive protections than the
# defaults.  In particular, it may be desirable to switch many of the 644
```

```
# entries to 640 or 600.  For example, some sites will consider the
# contents of maillog, messages, and lpd-errs to be confidential.  In the
# future, these defaults may change to more conservative ones.
#
# logfilename          [owner:group]    mode count size when  flags [/pid_file]
[sig_num]
/var/log/all.log                        600  7     *    @T00  J
/var/log/amd.log                        644  7     100  *     J
/var/log/auth.log                       600  7     100  @0101T JC
/var/log/console.log                    600  5     100  *     J
/var/log/cron                           600  3     100  *     JC
/var/log/daily.log                      640  7     *    @T00  JN
/var/log/debug.log                      600  7     100  *     JC
/var/log/kerberos.log                   600  7     100  *     J
/var/log/lpd-errs                       644  7     100  *     JC
/var/log/maillog                        640  7     *    @T00  JC
/var/log/messages                       644  5     100  @0101T JC
/var/log/monthly.log                    640  12    *    $M1D0 JN
/var/log/pflog                          600  3     100  *     JB
/var/run/pflogd.pid
/var/log/ppp.log        root:network    640  3     100  *     JC
/var/log/devd.log                       644  3     100  *     JC
/var/log/security                       600  10    100  *     JC
/var/log/sendmail.st                    640  10    *    168   B
/var/log/utx.log                        644  3     *    @01T05 B
/var/log/weekly.log                     640  5     1    $W6D0 JN
/var/log/xferlog                        600  7     100  *     JC
```

Cada linha começa com o nome do log a ser rotacionado, seguido opcionalmente por um proprietário e um grupo para arquivos rotacionados e recém-criados. O campo `mode` define as permissões no arquivo de log e `count` indica quantos arquivos de log rotacionados devem ser mantidos. Os campos `size` e `when` informam o newsyslog quando rotacionar o arquivo. Um arquivo de log é rotacionado quando seu tamanho é maior que o campo `size` ou quando o tempo no campo `when` tiver terminado. Um asterisco (`*`) significa que este campo é ignorado. O campo *flags* fornece instruções adicionais, por exemplo, como compactar o arquivo rotacionado ou criar o arquivo de log se ele estiver ausente. Os dois últimos campos são opcionais e especificam o nome do arquivo de ID de Processo (PID) e um número de sinal para enviar a esse processo quando o arquivo é rotacionado.

Para obter maiores informações sobre todos os campos, sinalizadores válidos e como sobre especificar o tempo de rotação, consulte newsyslog.conf(5). Como o newsyslog é executado a partir do cron(8), ele não pode rotacionar arquivos com mais frequência do que a que está planejada para ser executada no cron(8).

11.7.3. Configurando o log remoto

Monitorar os arquivos de log de vários hosts pode se tornar difícil à medida que o número de sistemas aumenta. Configurar o log centralizado pode reduzir parte da carga administrativa da administração dos arquivos de log.

No FreeBSD, a agregação, a fusão e a rotação centralizada de arquivos de log podem ser configuradas usando o syslogd e o newsyslog. Esta seção demonstra um exemplo de configuração, em que host o `A`, chamado `logserv.example.com`, coletará informações de log para a rede local. O host `B`, denominado `logclient.example.com`, será configurado para transmitir informações de log para o servidor de registro em log.

11.7.3.1. Configuração do servidor de log

Um servidor de log é um sistema que foi configurado para aceitar informações de log de outros hosts. Antes de configurar um servidor de log, verifique o seguinte:

- Se houver um firewall entre o servidor de log e qualquer cliente de log, certifique-se de que o conjunto de regras do firewall permita a porta 514 do UDP para os clientes e o servidor.
- O servidor de log e todas as máquinas clientes devem ter entradas de nome diretas e reversas no DNS local. Se a rede não tiver um servidor DNS, crie entradas no /etc/hosts de cada sistema. A resolução adequada de nomes é necessária para que as entradas de log não sejam rejeitadas pelo servidor de log.

No servidor de log, edite o /etc/syslog.conf para especificar o nome do cliente para receber as entradas de log, o recurso de log a ser usado e o nome do log para armazenar as entradas de log do host. Este exemplo adiciona o nome do host de `B`, registra todos os recursos e armazena as entradas de log em /var/log/logclient.log.

Exemplo 25. Configuração do servidor de log de exemplo

```
+logclient.example.com
*.*     /var/log/logclient.log
```

Ao adicionar vários clientes de log, adicione uma entrada semelhante de duas linhas para cada cliente. Maiores informações sobre os recursos disponíveis podem ser encontradas em syslog.conf(5).

Em seguida, configure o /etc/rc.conf:

```
syslogd_enable="YES"
syslogd_flags="-a logclient.example.com -v -v"
```

A primeira entrada inicia o syslogd na inicialização do sistema. A segunda entrada permite entradas de log do cliente especificado. A opção `-v -v` aumenta a verbosidade das mensagens registradas. Isso é útil para ajustar os recursos, pois os administradores podem ver o tipo de mensagens que estão sendo registradas em cada facility.

Múltiplas opções `-a` podem ser especificadas para permitir o registro de múltiplos clientes. Endereços IP e netblocks inteiros também podem ser especificados. Consulte syslogd(8) para obter uma lista completa de opções possíveis.

Finalmente, crie o arquivo de log:

```
# touch /var/log/logclient.log
```

Neste ponto, o syslogd deve ser reiniciado e verificado:

```
# service syslogd restart
# pgrep syslog
```

Se um PID for retornado, o servidor será reiniciado com êxito e a configuração do cliente poderá ser iniciada. Se o servidor não reiniciar, consulte /var/log/messages para visualizar o erro.

11.7.3.2. Configuração do cliente de log

Um cliente de log envia entradas de log para um servidor de log na rede. O cliente também mantém uma cópia local de seus próprios logs.

Uma vez que o servidor de log foi configurado, edite o /etc/rc.conf no cliente de registro:

```
syslogd_enable="YES"
syslogd_flags="-s -v -v"
```

A primeira entrada ativa o syslogd na inicialização. A segunda entrada impede que os logs sejam aceitos por esse cliente de outros hosts (`-s`) e aumenta a verbosidade das mensagens registradas.

Em seguida, defina o servidor de log no /etc/syslog.conf do cliente. Neste exemplo, todos os facilities registrados são enviados para um sistema remoto, indicado pelo símbolo `@`, com o nome do host especificado:

```
*.*     @logserv.example.com
```

Depois de salvar a edição, reinicie o syslogd para que as alterações entrem em vigor:

```
# service syslogd restart
```

Para testar se as mensagens de log estão sendo enviadas pela rede, use o logger(1) no cliente para enviar uma mensagem para syslogd:

```
# logger "Test message from logclient"
```

Esta mensagem agora deve existir tanto no /var/log/messages do cliente e no /var/log/logclient.log do servidor de log.

11.7.3.3. Debugando servidores de log

Se nenhuma mensagem estiver sendo recebida no servidor de log, a causa provavelmente é um problema de conectividade de rede, um problema de resolução de nome de host ou um erro de digitação em um arquivo de configuração. Para isolar a causa, certifique-se de que o servidor de log e o cliente de log sejam capazes de comunicarem através do `ping` usando o nome do host especificado em seu /etc/rc.conf. Se isso falhar, verifique o cabeamento da rede, o conjunto de regras do firewall e as entradas de nome de host no servidor DNS ou /etc/hosts no servidor de log e nos clientes. Repita até que o `ping` seja bem-sucedido em ambos os hosts.

Se o `ping` for bem-sucedido em ambos os hosts, mas as mensagens de log ainda não estiverem sendo recebidas, aumente temporariamente o detalhamento do log para diminuir o problema de configuração. No exemplo a seguir, o /var/log/logclient.log no servidor de log está vazio e o /var/log/messages no cliente de log não indica uma razão para a falha. Para aumentar a saída de debug, edite a entrada `syslogd_flags` no servidor de log e execute uma reinicialização:

```
syslogd_flags="-d -a logclient.example.com -v -v"
```

```
# service syslogd restart
```

Dados de debug semelhantes aos seguintes irão aparecer no console imediatamente após a reinicialização:

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
Logging to FILE /var/log/messages
syslogd: kernel boot file is /boot/kernel/kernel
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
rejected in rule 0 due to name mismatch.
```

Neste exemplo, as mensagens de log estão sendo rejeitadas devido a um erro de digitação que resulta em uma incompatibilidade de nome de host. O nome do host do cliente deve ser `logclient`, não `logclien`. Corrija o erro de digitação, execute uma reinicialização e verifique os resultados:

```
# service syslogd restart
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
syslogd: kernel boot file is /boot/kernel/kernel
logmsg: pri 166, flags 17, from logserv.example.com,
msg Dec 10 20:55:02 <syslog.err> logserv.example.com syslogd: exiting on signal 2
cvthname(192.168.1.10)
```

```
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
accepted in rule 0.
logmsg: pri 15, flags 0, from logclient.example.com, msg Dec 11 02:01:28 trhodes: Test
message 2
Logging to FILE /var/log/logclient.log
Logging to FILE /var/log/messages
```

Neste ponto, as mensagens estão sendo recebidas e colocadas corretamente no arquivo correto.

11.7.3.4. Considerações de segurança

Como com qualquer serviço de rede, os requisitos de segurança devem ser considerados antes de implementar um servidor de log. Os arquivos de log podem conter dados confidenciais sobre serviços ativados no host local, contas de usuário e dados de configuração. Os dados enviados pela rede do cliente para o servidor não serão criptografados nem protegidos por senha. Se houver necessidade de criptografia, considere o uso do security/stunnel, que transmitirá os dados de log em um túnel criptografado.

A segurança local também é um problema. Os arquivos de log não são criptografados durante o uso ou após a rotação do log. Usuários locais podem acessar arquivos de log para obter informações adicionais sobre a configuração do sistema. Definir permissões adequadas nos arquivos de log é crítico. O rotacionador de log integrado, newsyslog, suporta a configuração de permissões em arquivos de log recém-criados e rotacionados. A configuração de arquivos de log no modo `600` deve impedir o acesso indesejado por usuários locais. Consulte newsyslog.conf(5) para obter informações adicionais.

11.8. Arquivos de Configuração

11.8.1. Layout do /etc

Existem vários diretórios nos quais as informações de configuração são mantidas. Estes incluem:

/etc	Informações de configuração específica do sistema genérico.
/etc/defaults	Versões padrão dos arquivos de configuração do sistema.
/etc/mail	Configuração extra do sendmail(8) e outros arquivos de configuração MTA.
/etc/ppp	Configuração para ambos os programas, user- e kernel-ppp.
/usr/local/etc	Arquivos de configuração para aplicativos instalados. Pode conter subdiretórios para cada aplicativo.
/usr/local/etc/rc.d	scripts rc(8) para os aplicativos instalados.

/var/db	Arquivos de banco de dados específicos do sistema gerados automaticamente, como o banco de dados de pacotes e o banco de dados locate(1).

11.8.2. Hostnames

11.8.2.1. /etc/resolv.conf

Como um sistema FreeBSD acessa o Sistema de Nomes de Domínio da Internet (Internet Domain Name System - DNS) é controlado por resolv.conf(5).

As entradas mais comuns para o /etc/resolv.conf são:

`nameserver`	O endereço IP de um servidor de nomes que o resolvedor deve consultar. Os servidores são consultados na ordem listada com um máximo de três.
`search`	Lista de pesquisa, para busca de nomes de host. Isso é normalmente determinado pelo domínio do nome do host local.
`domain`	O nome do domínio local.

Um típico /etc/resolv.conf é assim:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```



Apenas uma das opções `search` e `domain` deve ser usada.

Ao usar o DHCP, o dhclient(8) geralmente reescreve o /etc/resolv.conf com informações recebidas do servidor DHCP.

11.8.2.2. /etc/hosts

O /etc/hosts é um banco de dados de texto simples que funciona em conjunto com o DNS e o NIS para fornecer o nome do host aos mapeamentos de endereços IP. Entradas para computadores locais conectados através de uma LAN podem ser adicionadas a este arquivo para propósitos simplistas de nomeação em vez de configurar um servidor named(8). Além disso, o /etc/hosts pode ser usado para fornecer um registro local de nomes da Internet, reduzindo a necessidade de consultar servidores DNS externos para nomes comumente acessados.

```
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
dbaio $
#
```

```
#
# Host Database
#
# This file should contain the addresses and aliases for local hosts that
# share this file.  Replace 'my.domain' below with the domainname of your
# machine.
#
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1         localhost localhost.my.domain
127.0.0.1       localhost localhost.my.domain
#
# Imaginary network.
#10.0.0.2       myname.my.domain myname
#10.0.0.3       myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#   10.0.0.0    -   10.255.255.255
#   172.16.0.0  -   172.31.255.255
#   192.168.0.0 -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  Do not try to invent your own network
# numbers but instead get one from your network provider (if any) or
# from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfriNIC.)
#
```

O formato do /etc/hosts é o seguinte:

```
[Internet address] [official hostname] [alias1] [alias2] ...
```

Por exemplo:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

Consulte hosts(5) para obter maiores informações.

11.9. Efetuando ajustes com o sysctl(8)

O sysctl(8) é usado para fazer mudanças em um sistema FreeBSD em execução. Isso inclui muitas opções avançadas da stack TCP/IP e do sistema de memória virtual as quais podem melhorar drasticamente o desempenho do FreeBSD para um administrador de sistema experiente. Mais de quinhentas variáveis do sistema podem ser lidas e definidas usando o sysctl(8).

Em sua essência, o sysctl(8) serve duas funções: ler e modificar as configurações do sistema.

Para ver todas as variáveis legíveis:

```
% sysctl -a
```

Para ler uma variável específica, especifique seu nome:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Para definir uma variável específica, use a sintaxe *variable=value*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

As configurações das variáveis sysctl são geralmente strings, números ou booleanos, onde um booleano é `1` para sim `0` para não.

Para definir automaticamente algumas variáveis sempre que a máquina inicializar, adicione-as ao /etc/sysctl.conf. Para maiores informações, consulte sysctl.conf(5) e sysctl.conf.

11.9.1. sysctl.conf

O arquivo de configuração para o sysctl(8), /etc/sysctl.conf, se parece muito com o /etc /rc.conf. Os valores são definidos na forma `variable=value`. Os valores especificados são definidos após o sistema entrar no modo multiusuário. Nem todas as variáveis são configuráveis neste modo.

Por exemplo, para desativar o log de saídas de sinais fatais e impedir que os usuários vejam processos iniciados por outros usuários, os seguintes ajustes podem ser configurados em /etc/sysctl.conf:

```
# Do not log fatal signal exits (e.g., sig 11)
kern.logsigexit=0

# Prevent users from seeing information about processes that
# are being run under another UID.
security.bsd.see_other_uids=0
```

11.9.2. Variáveis sysctl(8) apenas de leitura

Em alguns casos, pode ser desejável modificar os valores de variáveis do sysctl(8) que são somente de leitura, o que exigirá uma reinicialização do sistema.

Por exemplo, em alguns modelos de laptops, o dispositivo cardbus(4) não examinará os intervalos de memória e falhará com erros semelhantes a:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

A correção requer a modificação de uma configuração definida por uma variável do sysctl(8) que é somente de leitura. Adicione `hw.pci.allow_unsupported_io_range=1` ao arquivo /boot/loader.conf e reinicie. Agora o cardbus(4) deve funcionar corretamente.

11.10. Otimização de Discos

A seção a seguir discutirá vários mecanismos e opções de ajuste que podem ser aplicados a dispositivos de disco. Em muitos casos, discos com partes mecânicas, como unidades SCSI, serão o gargalo que reduz o desempenho geral do sistema. Embora a solução seja instalar uma unidade sem peças mecânicas, como uma unidade de estado sólido, as unidades mecânicas não irão desaparecer num futuro próximo. Quando estiver otimizando discos, é aconselhável utilizar os recursos do comando iostat(8) para testar várias mudanças no sistema. Este comando permitirá ao usuário obter informações valiosas sobre o sistema IO.

11.10.1. Variáveis Sysctl

11.10.1.1. `vfs.vmiodirenable`

A variável `vfs.vmiodirenable` sysctl(8) pode ser definida como `0` (off) ou `1` (on). Está definida para `1` por padrão. Esta variável controla como os diretórios são armazenados em cache pelo sistema. A maioria dos diretórios é pequena, usando apenas um único fragmento (normalmente 1K) no sistema de arquivos e, normalmente, 512 bytes no cache de buffer. Com esta variável desativada, o cache de buffer armazenará apenas um número fixo de diretórios, mesmo que o sistema tenha uma quantidade enorme de memória. Quando ativado, este sysctl(8) permite que o cache de buffer use o cache de página VM para armazenar em cache os diretórios, disponibilizando toda a memória para fazer cache dos diretórios. No entanto, a memória mínima no núcleo usada para armazenar em cache um diretório é o tamanho da página física (geralmente 4K) em vez de 512 bytes. Manter esta opção ativada é recomendado se o sistema estiver executando quaisquer serviços que manipulem um grande número de arquivos. Esses serviços podem incluir caches da web, grandes sistemas de correio e sistemas de notícias. Manter essa opção geralmente não reduz o desempenho, mesmo com a memória desperdiçada, mas deve-se experimentar para descobrir.

11.10.1.2. `vfs.write_behind`

A variável `vfs.write_behind` sysctl(8) é padronizada para `1` (ligada). Isso informa ao sistema de arquivos para emitir gravações de mídia à medida que clusters completos são coletados, o que normalmente ocorre ao gravar arquivos sequenciais grandes. Isso evita saturar o cache de buffer com buffers sujos quando não beneficia o desempenho de I/O. No entanto, isso pode atrasar os processos e, sob certas circunstâncias, deve ser desativado.

11.10.1.3. `vfs.hirunningspace`

A variável `vfs.hirunningspace` sysctl(8) determina quanto de I/O de gravação pendente pode ser enfileirado no sistema de controladores de disco como um todo em qualquer instância. O padrão é

geralmente suficiente, mas em máquinas com muitos discos, tente aumentar para quatro ou cinco *megabytes*. Definir um valor muito alto, que exceda o limite de gravação do cache de buffer, pode levar a um mau desempenho de cluster. Não defina esse valor arbitrariamente alto, pois valores de gravação mais altos podem adicionar latência a leituras que ocorrem ao mesmo tempo.

Há vários outros caches de buffer e valores de cache de página VM relacionados a sysctl(8). Modificar esses valores não é recomendado, pois o sistema VM faz um bom trabalho de ajuste automático.

11.10.1.4. `vm.swap_idle_enabled`

A variável `vm.swap_idle_enabled` sysctl(8) é útil em grandes sistemas multiusuários com muitos usuários ativos, e muitos processos ociosos. Tais sistemas tendem a gerar pressão contínua nas reservas de memória livre. Ativar esse recurso e aprimorar a histerese de troca (em segundos ociosos) por meio de `vm.swap_idle_threshold1` e `vm.swap_idle_threshold2` reduz a prioridade das páginas de memória associadas aos processos inativos mais rapidamente do que no algoritmo de pageout normal. Isso dá uma ajuda ao daemon de pageout. Apenas ative essa opção se necessário, porque a compensação é essencialmente fazer o pre-page da memoria mais cedo, o que consome mais swap e largura de banda de disco. Em um sistema pequeno, esta opção terá um efeito determinável, mas em um sistema grande que já está paginando moderadamente, esta opção permite que o sistema VM instale processos inteiros dentro e fora da memória facilmente.

11.10.1.5. `hw.ata.wc`

Desativar o cache de gravação IDE reduz a largura de banda de gravação em discos IDE, mas às vezes pode ser necessário devido a problemas de consistência de dados introduzidos por fornecedores de disco rígido. O problema é que algumas unidades IDE mentem sobre quando uma gravação é concluída. Com o cache de gravação IDE ativado, os discos rígidos IDE gravam os dados fora de ordem e às vezes atrasam a gravação de alguns blocos indefinidamente quando estão sob carga pesada de disco. Uma falha ou falha de energia pode causar corrupção séria do sistema de arquivos. Verifique o padrão no sistema observando a variável `hw.ata.wc` sysctl(8). Se o cache de gravação IDE estiver desativado, pode-se definir essa variável somente leitura como `1` em /boot/loader.conf para ativar no momento da inicialização.

Para maiores informações, consulte ata(4).

11.10.1.6. `SCSI_DELAY` (`kern.cam.scsi_delay`)

A opção de configuração do kernel `SCSI_DELAY` pode ser usada para reduzir os tempos de inicialização do sistema. Os padrões são razoavelmente altos e podem ser responsáveis por `15` segundos de atraso no processo de inicialização. Reduzindo-o para `5` segundos geralmente funciona com unidades modernas. A variável de tempo de inicialização `kern.cam.scsi_delay` deve ser usada. A opção de configuração ajustável e a configuração do kernel aceitam valores em termos de *milissegundos* e *não__segundos*.

11.10.2. Soft Updates

Para ajustar um sistema de arquivos, use tunefs(8). Este programa tem muitas opções diferentes. Para ativar e desativar o Soft Updates, use:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Um sistema de arquivos não pode ser modificado com tunefs(8) enquanto estiver montado. Um bom momento para ativar o Soft Updates é antes que qualquer partição tenha sido montada, no modo de single-user.

O Soft Updates é recomendado para sistemas de arquivos UFS, pois melhora drasticamente o desempenho de metadados, principalmente a criação e exclusão de arquivos, através do uso de um cache em memória. Há duas desvantagens no Soft Updates que você deve conhecer. Primeiro, o Soft Updates garante a consistência do sistema de arquivos no caso de uma falha, mas pode facilmente levar vários segundos ou até um minuto para atualizar o disco físico. Se o sistema falhar, os dados não gravados poderão ser perdidos. Em segundo lugar, os Soft Updates atrasam a liberação de blocos do sistema de arquivos. Se o sistema de arquivos raiz estiver quase cheio, a execução de uma atualização importante, como `make installworld`, poderá causar a falta de espaço do sistema de arquivos e a atualização falhará.

11.10.2.1. Mais detalhes sobre soft updates

As atualizações de metadados são atualizações para dados que não são de conteúdo, como inodes ou diretórios. Existem duas abordagens tradicionais para gravar os metadados de um sistema de arquivos em disco.

Historicamente, o comportamento padrão era gravar atualizações de metadados de forma síncrona. Se um diretório fosse alterado, o sistema aguardava até que a alteração fosse gravada no disco. Os buffers de dados do arquivo (conteúdo do arquivo) foram passados pelo cache de buffer e foram copiados para o disco posteriormente de maneira assíncrona. A vantagem dessa implementação é que ela opera com segurança. Se houver uma falha durante uma atualização, os metadados estarão sempre em um estado consistente. Um arquivo é criado completamente ou não é de todo. Se os blocos de dados de um arquivo não encontrarem saída do cache de buffer para o disco no momento da falha, o fsck(8) reconhece isso e repara o sistema de arquivos definindo o comprimento do arquivo como `0`. Além disso, a implementação é clara e simples. A desvantagem é que as alterações nos metadados são lentas. Por exemplo, `rm -r` toca todos os arquivos em um diretório sequencialmente, mas cada alteração de diretório será gravada de forma síncrona no disco. Isso inclui atualizações para o próprio diretório, para a tabela de inode e possivelmente para blocos indiretos alocados pelo arquivo. Considerações semelhantes aplicam-se ao desenrolar hierarquias grandes usando `tar -x`.

A segunda abordagem é usar atualizações de metadados assíncronas. Este é o padrão para um sistema de arquivos UFS montado com `mount -o async`. Como todas as atualizações de metadados também são passadas pelo cache de buffer, elas serão mescladas com as atualizações dos dados de conteúdo do arquivo. A vantagem dessa implementação é que não há necessidade de esperar até que cada atualização de metadados seja gravada no disco, portanto, todas as operações que causam grandes quantidades de atualizações de metadados funcionam muito mais rápido do que no caso síncrono. Essa implementação ainda é clara e simples, portanto, há um baixo risco de erros se infiltrarem no código. A desvantagem é que não há garantia para um estado consistente do sistema de arquivos. Se houver uma falha durante uma operação que atualizou grandes quantidades de metadados, como uma falha de energia ou alguém pressionando o botão de reinicialização, o

sistema de arquivos será deixado em um estado imprevisível. Não há oportunidade de examinar o estado do sistema de arquivos quando o sistema é reativado, pois os blocos de dados de um arquivo já podem ter sido gravados no disco enquanto as atualizações da tabela de inodes ou do diretório associado não foram. É impossível implementar um fsck(8) que é capaz de limpar o caos resultante porque as informações necessárias não estão disponíveis no disco. Se o sistema de arquivos foi danificado além do reparo, a única opção é reformatá-lo e restaurá-lo a partir do backup.

A solução usual para este problema é implementar *dirty region logging*, que também é chamado de *journaling*. As atualizações de metadados ainda são gravadas de forma síncrona, mas apenas em uma pequena região do disco. Mais tarde, eles são movidos para o local apropriado. Como a área de registro é uma região pequena e contígua no disco, não há longas distâncias para as cabeças de disco se moverem, mesmo durante operações pesadas, portanto, essas operações são mais rápidas do que as atualizações síncronas. Além disso, a complexidade da implementação é limitada, portanto, o risco de erros estarem presentes é baixo. Uma desvantagem é que todos os meta-dados são gravados duas vezes, uma vez na região de registro e uma vez no local apropriado, portanto, pode resultar em "piora" na performance. Por outro lado, em caso de falha, todas as operações de metadados pendentes podem ser rapidamente recuperadas ou concluídas a partir da área de registro depois que o sistema for ativado novamente, resultando em uma inicialização rápida do sistema de arquivos.

Kirk McKusick, o desenvolvedor do Berkeley FFS, resolveu esse problema com o Soft Updates. Todas as atualizações de meta-dados pendentes são mantidas na memória e gravadas no disco em uma sequência ordenada ("atualizações de metadados ordenadas"). Isso tem o efeito de que, no caso de operações pesadas de meta-dados, atualizações posteriores em um item "catch" as anteriores que ainda estão na memória e ainda não foram gravadas no disco. Todas as operações são geralmente executadas na memória antes da atualização ser gravada no disco e os blocos de dados são classificados de acordo com sua posição, de modo que não estarão no disco antes de seus meta-dados. Se o sistema travar, um "reenvio de log" implícito faz com que todas as operações que não foram gravadas no disco apareçam como se nunca tivessem acontecido. Um estado consistente do sistema de arquivos é mantido e parece ser o de 30 a 60 segundos antes. O algoritmo usado garante que todos os recursos em uso sejam marcados como tal em seus blocos e inodes. Após uma falha, o único erro de alocação de recursos que ocorre é que os recursos são marcados como "used", que são, na verdade, "free". O fsck(8) reconhece essa situação e libera os recursos que não são mais usados. É seguro ignorar o estado sujo do sistema de arquivos após uma falha forçando a montagem com `mount -f`. Para liberar recursos que podem não ser utilizados, O fsck(8) precisa ser executado posteriormente. Esta é a idéia por trás do *fsck(8) em background*: no momento da inicialização do sistema, apenas um *snapshot* do sistema de arquivos é gravado e o fsck(8) é executado posteriormente. Todos os sistemas de arquivos podem ser montados "sujos", para que a inicialização do sistema prossiga no modo multiusuário. Em seguida, o fsck(8) em background é planejado para todos os sistemas de arquivos em que isso é necessário, para liberar recursos que podem não ser utilizados. Os sistemas de arquivos que não usam Soft Updates ainda precisam executar o fsck(8) em primeiro plano de forma usual .

A vantagem é que as operações de meta-dados são quase tão rápidas quanto as atualizações assíncronas e são mais rápidas que o *logging*, que precisa escrever os meta-dados duas vezes. As desvantagens são a complexidade do código, um maior consumo de memória e algumas idiosincrasias. Depois de uma falha, o estado do sistema de arquivos parece ser um pouco mais "velho". Em situações onde a abordagem síncrona padrão teria causado a existencia de alguns

arquivos de comprimento zero após o fsck(8), esses arquivos sequer chegam a existir com Soft Updates porque nem os metadados e nem o conteúdo do arquivo foram gravados no disco. O espaço em disco não é liberado até que as atualizações tenham sido gravadas no disco, o que pode ocorrer algum tempo depois de executar rm(1). Isso pode causar problemas ao instalar grandes quantidades de dados em um sistema de arquivos que não tenha espaço livre suficiente para armazenar todos os arquivos duas vezes.

11.11. Ajustando os Limites do Kernel

11.11.1. Limites de arquivos/processos

11.11.1.1. `kern.maxfiles`

A variável `kern.maxfiles` sysctl(8) pode ser aumentada ou diminuída com base nos requisitos do sistema. Essa variável indica o número máximo de descritores de arquivos no sistema. Quando a tabela do descritor de arquivos estiver cheia, o erro `file: table is full` aparecerá repetidamente no buffer de mensagem do sistema, que pode ser visualizado usando dmesg(8).

Cada arquivo aberto, socket ou fifo usa um descritor de arquivo. Um servidor de produção em larga escala pode facilmente exigir muitos milhares de descritores de arquivos, dependendo do tipo e número de serviços executados simultaneamente.

Em versões mais antigas do FreeBSD, o valor padrão de `kern.maxfiles` é derivado do `maxusers` no arquivo de configuração do kernel. O `kern.maxfiles` cresce proporcionalmente ao valor do `maxusers`. Ao compilar um kernel personalizado, considere configurar esta opção de configuração do kernel de acordo com o uso do sistema. A partir desse número, o kernel recebe a maioria dos seus limites predefinidos. Mesmo que uma máquina de produção não tenha 256 usuários simultâneos, os recursos necessários podem ser semelhantes a um servidor da Web de alta escala.

A variável read-only sysctl(8)`kern.maxusers` é dimensionada automaticamente na inicialização com base na quantidade de memória disponível no sistema, e pode ser determinado em tempo de execução, inspecionando o valor de `kern.maxusers`. Alguns sistemas requerem valores maiores ou menores de `kern.maxusers` e valores de `64`, `128`, e `256` não são incomuns. Ir acima de `256` não é recomendado, a menos que seja necessário um grande número de descritores de arquivos. Muitos dos valores ajustáveis definidos para seus padrões por `kern.maxusers` podem ser individualmente sobrescritos no tempo de inicialização ou em tempo de execução no /boot/loader.conf. Consulte loader.conf(5) e /boot/defaults/loader.conf para mais detalhes e algumas dicas.

Em versões mais antigas, o sistema ajustará automaticamente o `maxusers` se ele estiver definido como `0`. . Ao configurar esta opção, configure o `maxusers` para pelo menos `4`, especialmente se o sistema executar o Xorg ou se for usado para compilar software. A tabela mais importante definida por `maxusers` é o número máximo de processos, que é definido como `20 + 16 * maxusers`. Se `maxusers` for definido como `1`, só podem existir `36` processos simultâneos, incluindo `18` ou mais para que o sistema seja iniciado no boot ou `15` usado pelo Xorg. Até mesmo uma tarefa simples, como ler uma página de manual, iniciará nove processos para filtrar, descompactar e visualizar. Configurar o `maxusers` para `64` permite até `1044` processos simultâneos, o que deve ser suficiente para quase todos os usos. Se, no entanto, o erro for exibido ao tentar iniciar outro programa ou se um servidor estiver sendo executado com um grande número de usuários simultâneos, aumente o número e

recompile.



O `maxusers` *não* limita o número de usuários que podem logar na máquina. Em vez disso, ele configura vários tamanhos de tabela para valores razoáveis, considerando o número máximo de usuários no sistema e quantos processos cada usuário executará.

11.11.1.2. `kern.ipc.soacceptqueue`

A variável `kern.ipc.soacceptqueue` do sysctl(8) limita o tamanho da fila de escuta para aceitar novas conexões `TCP`. O valor padrão de `128` é normalmente muito baixo para o manuseio robusto de novas conexões em um servidor Web com carga pesada. Para tais ambientes, recomenda-se aumentar este valor para `1024` ou superior. Um serviço como o sendmail(8), ou Apache pode limitar por ele mesmo o tamanho da fila de escuta, mas frequentemente terá uma diretiva em seu arquivo de configuração para ajustar o tamanho da fila. Filas de escuta grandes fazem um trabalho melhor evitando ataques de negação de serviço (Denial of Service - DoS).

11.11.2. Limites de rede

A opção de configuração do kernel `NMBCLUSTERS` determina a quantidade de Mbufs de rede disponível para o sistema. Um servidor com muito tráfego e um baixo número de Mbufs prejudicará o desempenho. Cada cluster representa aproximadamente 2K de memória, portanto, um valor de `1024` representa `2` megabytes de memória do kernel reservada para buffers de rede. Um cálculo simples pode ser feito para descobrir quantos são necessários. Um servidor web que suporte um maximo de `1000` conexões simultâneas onde cada conexão usa um buffer de envio de 16K e recebe 6K, requer aproximadamente 32 MB de buffers de rede para cobrir o servidor web. Uma boa regra é multiplicar por `2`, então 2x32MB / 2KB = 64MB / 2kB = `32768`. Valores entre `4096` e `32768` são recomendados para máquinas com maiores quantidades de memória. Nunca especifique um valor arbitrariamente alto para este parâmetro, pois isso pode levar a uma falha no tempo de inicialização. Para observar o uso do cluster de rede, use a opção `-m` com o netstat(1).

A variável `kern.ipc.nmbclusters` deve ser usada para configurar isso no momento da inicialização. Apenas as versões mais antigas do FreeBSD irão requerer o uso da opção `NMBCLUSTERS` no config(8) do kernel.

Para servidores ocupados que fazem uso extensivo da chamada de sistema sendfile(2), pode ser necessário aumentar o número de buffers sendfile(2) através da opção de configuração do kernel `NSFBUFS` ou definindo seu valor no /boot/loader.conf (veja loader(8) para detalhes). Um indicador comum de que esse parâmetro precisa ser ajustado é quando os processos são vistos no estado `sfbufa`. A variável sysctl(8)`kern.ipc.nsfbufs` é somente de leitura. Este parâmetro nominalmente escala com o `kern.maxusers`, no entanto, pode ser necessário ajustar de acordo.



Mesmo que um socket tenha sido marcado como non-blocking, chamar o sendfile(2) em um socket non-blocking pode resultar no bloqueio do sendfile(2) até que sejam disponibilizados `struct sf_buf` suficientes.

11.11.2.1. net.inet.ip.portrange.*

As variáveis net.inet.ip.portrange.* do sysctl(8) controlam os intervalos de números de porta automaticamente ligados a sockets TCP e UDP. Existem três intervalos: um intervalo baixo, um intervalo padrão e um intervalo alto. A maioria dos programas de rede usam o intervalo padrão que é controlado por net.inet.ip.portrange.first e net.inet.ip.portrange.last, cujo padrão é 1024 e 5000, respectivamente. Intervalos de porta ligados são usados para conexões de saída e é possível executar o sistema fora das portas sob certas circunstâncias. Isso ocorre mais comumente ao executar um proxy web com muita carga. O intervalo de portas não é um problema ao executar um servidor que lida principalmente com conexões de entrada, como um servidor Web, ou que tenha um número limitado de conexões de saída, como um mail relay. Para situações em que há falta de portas, é recomendado aumentar modestamente o net.inet.ip.portrange.last. Um valor de 10000, 20000 ou 30000 pode ser razoável. Considere os efeitos do firewall ao alterar o intervalo de portas. Alguns firewalls podem bloquear grandes intervalos de portas, geralmente portas de numeração baixa, e esperam que os sistemas usem intervalos mais altos de portas para conexões de saída. Por esta razão, não é recomendado que o valor de net.inet.ip.portrange.first seja diminuído.

11.11.2.2. Produto de atraso de largura de banda TCP

A limitação do produto de atraso de largura de banda TCP pode ser ativada configurando a variável net.inet.tcp.inflight.enable do sysctl(8) para 1. Isso instrui o sistema a tentar calcular o produto de atraso de largura de banda para cada conexão e a limitar a quantidade de dados na fila para envio à rede para a quantidade necessária para manter o rendimento ideal.

Esse recurso é útil ao servir dados sobre modems, Gigabit Ethernet, links WAN de alta velocidade ou qualquer outro link com um produto de atraso de largura de banda alta, especialmente quando também estiver usando dimensionamento de janela ou quando uma janela de envio grande tiver sido configurado. Ao habilitar essa opção, defina também a variável net.inet.tcp.inflight.debug para 0 para desabilitar a depuração. Para uso em produção, definir a variável net.inet.tcp.inflight.min para pelo menos 6144 pode ser benéfico. Definir valores mínimos altos pode efetivamente desabilitar a limitação de largura de banda, dependendo do link. O recurso de limitação reduz a quantidade de dados acumulados nas rotas intermediárias e nas filas de pacotes de switchs e reduz a quantidade de dados acumulados na fila de interface do host local. Com menos pacotes enfileirados, as conexões interativas, especialmente os modems lentos, funcionarão com menores *Round Trip Times*. Esse recurso afeta apenas a transmissão de dados do lado do servidor, como o upload. Não tem efeito na recepção ou download de dados.

Ajustar o valor da variável net.inet.tcp.inflight.stab *não* é recomendado. Este parâmetro é padronizado para 20, representando 2 pacotes máximos adicionados ao cálculo da janela de produto de atraso de largura de banda. A janela adicional é necessária para estabilizar o algoritmo e melhorar a capacidade de resposta às mudanças de condições, mas também pode resultar em um ping(8) mais alto em links lentos , embora ainda muito menor do que sem o algoritmo inflight. Nesses casos, tente reduzir esse parâmetro para 15, 10 ou 5 e reduza a variável net.inet.tcp.inflight.min para um valor como 3500 para obter o efeito desejado. A redução desses parâmetros deve ser feita apenas como último recurso.

11.11.3. Memória virtual

11.11.3.1. kern.maxvnodes

Um vnode é a representação interna de um arquivo ou diretório. Aumentar o número de vnodes disponíveis para o sistema operacional reduz a I/O do disco. Normalmente, isso é tratado pelo sistema operacional e não precisa ser alterado. Em alguns casos em que o I/O de disco é um gargalo e o sistema está ficando sem vnodes, essa configuração precisa ser aumentada. A quantidade de RAM inativa e livre precisará ser levada em conta.

Para ver o número atual de vnodes em uso:

```
# sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

Para ver o máximo de vnodes:

```
# sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

Se o uso atual do vnode estiver próximo do máximo, tente aumentar o `kern.maxvnodes` por um valor de `1000`. Fique de olho no número de `vfs.numvnodes`. Se subir até o máximo novamente, o `kern.maxvnodes` precisará ser aumentado ainda mais. Caso contrário, uma mudança no uso da memória como reportado pelo top(1) deve estar visível e mais memória deve estar ativa.

11.12. Adicionando Espaço de Swap

Às vezes, um sistema requer mais espaço de swap. Esta seção descreve dois métodos para aumentar o espaço de troca: adicionar swap a uma partição existente ou em um novo disco rígido e criar um arquivo de swap em uma partição existente.

Para obter informações sobre como criptografar o espaço de swap, quais opções existem e por que isso deve ser feito, consulte Criptografando Swap.

11.12.1. Swap em um novo disco rígido ou partição existente

Adicionar um novo disco rígido para swap resulta em um melhor desempenho do que usando uma partição em uma unidade existente. A configuração de partições e discos rígidos é explicada em Adicionando Discos enquanto Criando o layout da partição discute layouts de partições e considerações sobre o tamanho de partições de swap.

Use o `swapon` para adicionar uma partição swap ao sistema. Por exemplo:

```
# swapon /dev/ada1s1b
```



É possível usar qualquer partição que não esteja atualmente montada, mesmo que já contenha dados. O uso do `swapon` em uma partição que contém dados sobrescreverá e destruirá esses dados. Certifique-se de que a partição a ser

> incluída como swap seja realmente a partição pretendida antes de executar o `swapon`.

Para adicionar automaticamente essa partição swap na inicialização, adicione uma entrada ao /etc/fstab:

```
/dev/ada1s1b    none    swap    sw  0   0
```

Veja fstab(5) para uma explicação das entradas do /etc/fstab. Maiores informações sobre `swapon` podem ser encontradas em swapon(8).

11.12.2. Criando um arquivo de swap

Esses exemplos criam um arquivo de swap de 512M chamado /usr/swap0 em vez de usar uma partição.

O uso de arquivos de swap requer que o módulo necessário pelo md(4) tenha sido embutido no kernel ou tenha sido carregado antes do swap ser ativado. Veja Configurando o kernel do FreeBSD para informações sobre como compilar um kernel customizado.

Exemplo 26. Criando um arquivo de swap

1. Crie o arquivo de swap:

   ```
   # dd if=/dev/zero of=/usr/swap0 bs=1m count=512
   ```

2. Defina as permissões adequadas no novo arquivo:

   ```
   # chmod 0600 /usr/swap0
   ```

3. Informe o sistema sobre o arquivo de swap adicionando uma linha ao /etc/fstab:

   ```
   md99    none    swap    sw,file=/usr/swap0,late 0   0
   ```

   O dispositivo md99 do md(4) é usado, deixando números de dispositivos inferiores disponíveis para uso interativo.

4. O espaço de swap será adicionado na inicialização do sistema. Para adicionar espaço de swap imediatamente, use o swapon(8):

   ```
   # swapon -aL
   ```

11.13. Gerenciamento de energia e recursos

É importante utilizar recursos de hardware de maneira eficiente. O gerenciamento de energia e recursos permite que o sistema operacional monitore os limites do sistema e, possivelmente, forneça um alerta se a temperatura do sistema aumentar inesperadamente. Uma especificação anterior para fornecer gerenciamento de energia foi o recurso Gerenciamento Avançado de Energia (Advanced Power Management - APM). O APM controla o uso de energia de um sistema com base em sua atividade. No entanto, era difícil e inflexível para os sistemas operacionais gerenciar o uso de energia e as propriedades térmicas de um sistema. O hardware era gerenciado pelo BIOS e o usuário tinha configuração e visibilidade limitadas nas configurações de gerenciamento de energia. O APMBIOS fornecido é específico da plataforma de hardware. Um driver APM no sistema operacional intermedia o acesso à interface do software APM, que permite o gerenciamento dos níveis de energia.

Existem quatro problemas principais no APM. Primeiro, o gerenciamento de energia é feito pelo BIOS específico do fornecedor, separado do sistema operacional. Por exemplo, o usuário pode definir valores de tempo ocioso para um disco rígido no APMBIOS para que, quando excedido, o BIOS diminua o disco rígido sem o consentimento do sistema operacional. Segundo, a lógica do APM é incorporada no BIOS e opera fora do escopo do sistema operacional. Isso significa que os usuários só podem corrigir problemas no APMBIOS, fazendo o flash de um novo ROM, que é um procedimento perigoso com potencial para deixar o sistema em um estado irrecuperável se falhar. Terceiro, o APM é uma tecnologia específica do fornecedor, o que significa que há muita duplicidade de esforços e que os erros encontrados no BIOS de um fornecedor podem não serem resolvidos em outros. Por fim, o APMBIOS não tinha espaço suficiente para implementar uma política de energia sofisticada ou que pudesse se adaptar bem ao propósito da máquina.

O BIOS plug and play (PNPBIOS) não era confiável em muitas situações. O PNPBIOS é uma tecnologia de 16 bits, portanto, o sistema operacional precisa usar a emulação de 16 bits para fazer interface com os métodos PNPBIOS. O FreeBSD fornece um driver APM, pois o APM ainda deve ser usado para sistemas fabricados antes do ano 2000. O driver está documentado em apm(4).

O sucessor do APM é a Interface Avançada de Configuração e Energia (Advanced Configuration and Power Interface - ACPI). O ACPI é um padrão escrito por uma aliança de fornecedores para fornecer uma interface para recursos de hardware e gerenciamento de energia. É um elemento-chave na *configuração direcionada do sistema operacional e gerenciamento de energia*, pois proporciona mais controle e flexibilidade ao sistema operacional.

Este capítulo demonstra como configurar o ACPI no FreeBSD. Em seguida, ele oferece algumas dicas sobre como depurar o ACPI e como enviar um relatório de problemas contendo informações de depuração para que os desenvolvedores possam diagnosticar e corrigir problemas no ACPI.

11.13.1. Configurando o ACPI

No FreeBSD, o driver acpi(4) é carregado por padrão na inicialização do sistema e *não* deve ser compilado no kernel. Este driver não pode ser descarregado após a inicialização porque o barramento do sistema o utiliza para várias interações de hardware. No entanto, se o sistema estiver com problemas, o ACPI pode ser desativado completamente ao reinicializar após a configurar `hint.acpi.0.disabled="1"` no /boot/loader.conf ou definindo esta variável no prompt do

loader, como descrito em Estágio três.



O ACPI e o APM não podem coexistir e devem ser usados separadamente. O último a ser carregado terminará se o driver perceber que o outro já está sendo executado.

O ACPI pode ser usado para colocar o sistema em modo de suspensão com o `acpiconf`, a opção `-s` e um número de `1` a `5`. A maioria dos usuários só precisa de `1` (suspensão rápida para RAM) ou `3` (suspender para RAM). A opção `5` executa um soft-off que é o mesmo que executar `halt -p`.

Outras opções estão disponíveis usando o `sysctl`. Consulte acpi(4) e acpiconf(8) para maiores informações.

11.13.2. Problemas comuns

O ACPI está presente em todos os computadores modernos que estão em conformidade com as arquiteturas ia32 (x86) e amd64 (AMD). O padrão completo tem muitos recursos, incluindo gerenciamento de desempenho da CPU, controle de planos de energia, zonas térmicas, vários sistemas de bateria, controladores incorporados e enumeração de barramento. A maioria dos sistemas implementa menos que o padrão completo. Por exemplo, um sistema de desktop geralmente só implementa a enumeração de barramento, enquanto um laptop também pode ter suporte a refrigeração e gerenciamento de bateria. Os laptops também têm suspensão e retomada, com sua própria complexidade associada.

Um sistema compatível com ACPI possui vários componentes. Os fornecedores de BIOS e chipset fornecem várias tabelas fixas, como FADT, na memória que especificam coisas como o mapa APIC (usado para SMP), registros de configuração e valores simples de configuração. Além disso, uma tabela de bytecode, a Tabela de Descrição de Sistema Diferenciada DSDT, especifica um espaço de nome de dispositivos e métodos em forma de árvore.

O driver ACPI deve analisar as tabelas fixas, implementar um interpretador para o bytecode e modificar os drivers de dispositivos e o kernel para aceitar informações do subsistema ACPI. Para o FreeBSD, a Intel™ forneceu um interpretador (ACPI-CA) que é compartilhado com o Linux™ e o NetBSD. O caminho para o código-fonte ACPI-CA é src/sys/contrib/dev/acpica. O código especifico que permite que o ACPI-CA funcione no FreeBSD está em src/sys/dev/acpica/Osd. Finalmente, drivers que implementam vários dispositivos ACPI são encontrados em src/sys/dev/acpica.

Para que o ACPI funcione corretamente, todas as partes devem funcionar corretamente. Aqui estão alguns problemas comuns, em ordem de freqüência em que ocorrem, e algumas possíveis soluções ou correções. Se uma correção não resolver o problema, consulte Obtendo e enviando informações de depuração para obter instruções sobre como enviar um relatório de bug.

11.13.2.1. Problemas do mouse

Em alguns casos, retomar a partir de uma operação de suspensão fará com que o mouse falhe. Um work around conhecido é adicionar `hint.psm.0.flags="0x3000"` ao /boot/loader.conf.

11.13.2.2. Suspend/Resume

O ACPI tem três estados de suspensão para RAM (STR), S1-S3, e um de suspensão de estado para disco (STD), chamado S4. O STD pode ser implementado de duas maneiras separadas. O S4 BIOS é uma suspensão para disco auxiliada pelo BIOSe o S4OS é implementado inteiramente pelo sistema operacional. O estado normal em que o sistema se encontra quando conectado, mas não ligado, é "soft off" (S5).

Use o `sysctl hw.acpi` para verificar os itens relacionados à suspensão. Estes resultados de exemplo são de um Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Use o `acpiconf -s` para testar os estados S3, S4 e S5. Um `s4bios` de um (1) indica suporte ao S4 BIOS em vez do S4 suportado pelo sistema operacional.

Ao testar as ações de suspend/resume, inicie com o S1, se suportado. É mais provável que esse estado funcione, pois não requer muito suporte ao driver. Ninguém implementou S2, que é similar ao S1. Em seguida, tente o S3. Este é o estado mais profundo do STR e requer muito suporte ao driver para reinicializar corretamente o hardware.

Um problema comum com suspend/resume é que muitos drivers de dispositivo não salvam, restauram ou reinicializam seu firmware, registros ou memória do dispositivo adequadamente. Como primeira tentativa de depuração do problema, tente:

```
# sysctl debug.bootverbose=1
# sysctl debug.acpi.suspend_bounce=1
# acpiconf -s 3
```

Esse teste emula o ciclo de suspend/resume de todos os drivers de dispositivo sem entrar realmente no estado S3. Em alguns casos, problemas como perder o estado do firmware, tempo limite do watchdog do dispositivo e tentar novamente para sempre podem ser capturados com esse método. Note que o sistema não entrará realmente no estado S3, o que significa que os dispositivos não perderão energia, e muitos funcionarão bem mesmo se os métodos suspend/resume estiverem totalmente ausentes, ao contrário do real estado S3.

Casos mais difíceis requerem hardware adicional, como uma porta serial e um cabo para depuração através de um console serial, uma porta Firewire e um cabo para o uso do dcons(4) e habilidades de depuração do kernel.

Para ajudar a isolar o problema, descarregue o maior número possível de drivers. Se funcionar, diminua o driver que é o problema carregando os drivers até que ele falhe novamente. Normalmente, drivers binários como nvidia.ko, drivers de exibição e USB terão mais problemas, enquanto as interfaces Ethernet normalmente funcionam bem. Se os drivers puderem ser carregados e descarregados adequadamente, automatize isso colocando os comandos apropriados em /etc/rc.suspend e /etc/rc.resume. Tente configurar o `hw.acpi.reset_video` para 1 se a tela estiver desarrumada após a retomada. Tente definir valores mais longos ou mais curtos para

`hw.acpi.sleep_delay` para ver se isso ajuda.

Tente carregar uma distribuição recente do Linux™ para ver se o suspend/resume funciona no mesmo hardware. Se funciona no Linux™, é provável que seja um problema no driver do FreeBSD. Descobrir qual driver causa o problema ajudará os desenvolvedores a corrigir o problema. Como os mantenedores do ACPI raramente mantêm outros drivers, como som ou ATA, qualquer problema de driver também deve ser postado na lista freebsd-current e enviada para o mantenedor do driver. Os usuários avançados podem incluir os printf(3)s de debug do driver problemático para rastrear onde, em sua função de reinício, ele é interrompido.

Por fim, tente desativar o ACPI e ativar o APM. Se o comando suspend/resume funcionar com APM, use o APM, especialmente em hardware mais antigo (anterior a 2000). Demorou algum tempo para que os fornecedores obtivessem suporte ACPI correto e os hardwares antigos são mais prováveis de terem problemas de BIOS com ACPI.

11.13.2.3. Travamentos do sistema

A maioria dos travamentos do sistema é resultado de interrupções perdidas ou de uma tempestade de interrupções. Chipsets podem ter problemas com base na inicialização, como o BIOS configura as interrupções antes da correção da tabela APIC (MADT) e o roteamento do sistema de controle de interrupções (SCI).

Tempestades de interrupção podem ser distinguidas de interrupções perdidas, verificando a saída do `vmstat -i` e observando a linha que possui `acpi0`. Se o contador está aumentando em mais de um par por segundo, há uma tempestade de interrupção. Se o sistema parece travado, tente acessar o DDB (`CTRL` + `ALT` + `ESC` no console) e digite `show interrupts`.

Ao lidar com problemas de interrupção, tente desativar o suporte ao APIC com `hint.apic.0.disabled="1"` no /boot/loader.conf .

11.13.2.4. Panics

Os panics são relativamente raros para ACPI e são a prioridade máxima a ser corrigida. O primeiro passo é isolar as etapas para reproduzir o panic, se possível, e obter um backtrace. Siga as instruções para habilitar `options DDB` e configurar um console serial em Entrando no Depurador DDB da Linha Serial ou configurar uma partição de despejo. Para obter um backtrace no DDB, use `tr`. Ao escrever o backtrace, obtenha pelo menos as cinco últimas e as cinco principais linhas do rastro.

Em seguida, tente isolar o problema inicializando com ACPI desabilitado. Se isso funcionar, isole o subsistema ACPI usando vários valores de `debug.acpi.disable`. Veja acpi(4) para alguns exemplos.

11.13.2.5. O sistema é ativado após a sua suspensão ou desligamento

Primeiro, tente definir `hw.acpi.disable_on_poweroff="0"` no /boot/loader.conf. Isso impede que a ACPI desative vários eventos durante o processo de desligamento. Alguns sistemas precisam desse valor definido como `1` (o padrão) pelo mesmo motivo. Isso geralmente corrige o problema de um sistema ser ativado espontaneamente após uma suspensão ou desligamento.

11.13.2.6. BIOS contém Bytecode com bugs

Alguns fornecedores de BIOS fornecem bytecode incorreto ou com bugs. Isso geralmente é manifestado por mensagens do console do kernel como esta:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Geralmente, esses problemas podem ser resolvidos com a atualização do BIOS para a revisão mais recente. A maioria das mensagens do console é inofensiva, mas se houver outros problemas, como o status da bateria não estar funcionando, essas mensagens são um bom lugar para começar a procurar por problemas.

11.13.3. Substituindo o padrão AML

O bytecode do BIOS, conhecido como ACPI Machine Language (AML), é compilado de uma linguagem de origem chamada ACPI Source Language (ASL). O AML é encontrado na tabela conhecida como Tabela de Descrição do Sistema Diferenciado (Differentiated System Description Table - DSDT).

O objetivo do FreeBSD é que todos trabalhem com ACPI sem qualquer intervenção do usuário. Soluções alternativas ainda estão sendo desenvolvidas para erros comuns feitos pelos fornecedores de BIOS. O interpretador Microsoft™ (acpi.sys e acpiec.sys) não verifica rigorosamente a conformidade com o padrão e, portanto, muitos fornecedores de BIOS que testam apenas ACPI sob Windows™ nunca corrigem seu ASL. Os desenvolvedores do FreeBSD continuam a identificar e documentar qual comportamento não padrão é permitido pelo interpretador da Microsoft™ para replicá-lo para que o FreeBSD possa funcionar sem forçar os usuários a corrigir o ASL.

Para ajudar a identificar o comportamento de bugs e possivelmente corrigi-lo manualmente, uma cópia pode ser feita do ASL do sistema. Para copiar o ASL do sistema para um nome de arquivo especificado, use `acpidump` com `-t`, para mostrar o conteúdo das tabelas fixas e `-d`, para desmontar o AML:

```
# acpidump -td > my.asl
```

Algumas versões de AML assumem que o usuário está executando o Windows™. Para sobrescrever isto, defina `hw.acpi.osname="Windows 2009"` no /boot/loader.conf, usando a mais recente versão do Windows™ listada no ASL.

Outras soluções alternativas podem exigir que o my.asl seja personalizado. Se este arquivo for editado, compile o novo ASL usando o seguinte comando. Os avisos geralmente podem ser ignorados, mas erros são bugs que geralmente impedem que o ACPI funcione corretamente.

```
# iasl -f my.asl
```

Incluir `-f` força a criação do AML, mesmo que haja erros durante a compilação. Alguns erros, como

a falta de declarações de retorno, são automaticamente contornados pelo interpretador do FreeBSD.

O nome do arquivo de saída padrão para `iasl` é DSDT.aml. Carregue este arquivo em vez da cópia com bugs do BIOS, que ainda está presente na memória flash, editando o /boot/loader.conf como segue:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Certifique-se de copiar o DSDT.aml para /boot e, em seguida, reinicialize o sistema. Se isso resolver o problema, envie um diff(1) do antigo e novo ASL para a lista freebsd-acpi para que os desenvolvedores possam contornar o comportamento de bugs no acpica.

11.13.4. Obtendo e enviando informações de depuração

O driver ACPI possui um recurso de depuração flexível. Um conjunto de subsistemas e o nível de detalhamento podem ser especificados. Os subsistemas a serem depurados são especificados como camadas e são divididos em componentes (`ACPI_ALL_COMPONENTS`) e suporte de hardware ACPI (`ACPI_ALL_DRIVERS`). O detalhamento da saída de depuração é especificado como o nível e varia de apenas erros de relatório (`ACPI_LV_ERROR`) para tudo (`ACPI_LV_VERBOSE`). O nível é uma máscara de bits, por isso, várias opções podem ser definidas de uma só vez, separadas por espaços. Na prática, um console serial deve ser usado para registrar a saída para que ela não seja perdida quando o buffer de mensagem do console for liberado. Uma lista completa das camadas e níveis individuais é encontrada em acpi(4).

A saída de depuração não está ativada por padrão. Para ativá-la, adicione as opções `ACPI_DEBUG` ao arquivo de configuração do kernel personalizado se ACPI estiver compilado no kernel. Adicione `ACPI_DEBUG=1` ao /etc/make.conf para ativá-lo globalmente. Se um módulo for usado em vez de um kernel personalizado, recompile apenas o módulo acpi.ko como segue:

```
# cd /sys/modules/acpi/acpi && make clean && make ACPI_DEBUG=1
```

Copie o acpi.ko compilado para /boot/kernel e adicione o nível e camada desejados ao /boot/loader.conf. As entradas neste exemplo permitem mensagens de depuração para todos os componentes e drivers de hardware ACPI e mensagens de erro de saída no nível menos detalhado:

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Se as informações necessárias forem acionadas por um evento específico, como suspend e resume, não modifique o /boot/loader.conf. Em vez disso, use o `sysctl` para especificar o layer e o nível após inicializar e preparar o sistema para o evento específico. As variáveis que podem ser definidas usando `sysctl` são nomeadas da mesma forma que os parâmetros no /boot/loader.conf.

Depois que as informações de depuração forem coletadas, elas podem ser enviadas para a lista

freebsd-acpi para que possam ser usadas pelos mantenedores do FreeBSD ACPI para identificar a causa raiz do problema e desenvolver uma solução.



Antes de enviar as informações de depuração para esta lista, certifique-se de que a versão mais recente do BIOS esteja instalada e, se disponível, a versão do firmware do controlador incorporado.

Ao enviar um relatório de problemas, inclua as seguintes informações:

- Descrição do comportamento de bugs, incluindo tipo de sistema, modelo e qualquer coisa que faça com que o erro apareça. Explique com a maior precisão possível quando o bug começou a ocorrer se for novo.
- A saída do `dmesg` após executar `boot -v`, incluindo quaisquer mensagens de erro geradas pelo bug.
- A saída `dmesg` do `boot -v` com o ACPI desabilitado, se a desativação do ACPI ajudar a corrigir o problema.
- Saída do `sysctl hw.acpi`. Isso lista quais recursos o sistema oferece.
- A URL para uma versão do ASL do sistema hospedada na web. *Não* envie o ASL diretamente para a lista, pois pode ser muito grande. Gere uma cópia do ASL executando este comando:

  ```
  # acpidump -dt > name-system.asl
  ```

  Substitua o nome de login para *name* e fabricante/modelo para *system*. Por exemplo, use njl-FooCo6000.asl.

A maioria dos desenvolvedores do FreeBSD assina a lista de discussão FreeBSD-CURRENT, mas deve-se enviar os problemas para a lista freebsd-acpi para ter certeza de que ele será visto. Seja paciente ao esperar por uma resposta. Se o bug não for imediatamente aparente, envie um relatório de bug. Ao inserir um PR, inclua as mesmas informações solicitadas acima. Isso ajuda os desenvolvedores a rastrear o problema e resolvê-lo. Não envie um PR sem enviar primeiro um e-mail para a lista freebsd-acpi pois é provável que o problema já tenha sido relatado antes.

11.13.5. Referências

Mais informações sobre ACPI podem ser encontradas nos seguintes locais:

- Arquivos da lista de e-mail do FreeBSD ACPI (https://lists.freebsd.org/pipermail/freebsd-acpi/)
- A especificação ACPI
- acpi(4), acpi_thermal(4), acpidump(8), iasl(8), e acpidb(8)

Capítulo 12. O processo de inicialização do FreeBSD

12.1. Sinopse

O processo de iniciar um computador e carregar o sistema operacional é chamado de "processo de bootstrap", ou de "inicialização". O processo de boot do FreeBSD fornece uma grande flexibilidade na personalização do que acontece quando o sistema é iniciado, incluindo a capacidade de selecionar diferentes sistemas operacionais instalados no mesmo computador, diferentes versões do mesmo sistema operacional ou um kernel instalado diferente.

Este capítulo detalha as opções de configuração que podem ser definidas. Ele demonstra como personalizar o processo de inicialização do FreeBSD, incluindo tudo o que acontece até que o kernel do FreeBSD tenha iniciado, procurado por dispositivos e iniciado o init(8). Isso ocorre quando a cor do texto das mensagens de inicialização muda de branco brilhante para cinza.

Depois de ler este capítulo, você reconhecerá:

- Os componentes do sistema de boot do FreeBSD e como eles interagem.
- As opções que podem ser passadas para os componentes no bootstrap do FreeBSD para controlar o processo de inicialização.
- Como configurar uma tela personalizada de inicialização.
- O básico da configuração de device hints.
- Como inicializar no modo de usuário único e multiusuário e como encerrar corretamente um sistema FreeBSD.



Este capítulo descreve apenas o processo de inicialização do FreeBSD rodando em sistemas x86 e amd64.

12.2. Processo de Inicialização do FreeBSD

Ligar um computador e iniciar o sistema operacional representa um dilema interessante. Por definição, o computador não sabe como fazer nada até que o sistema operacional seja iniciado. Isso inclui executar programas a partir do disco. Se o computador não pode executar um programa a partir do disco sem o sistema operacional e os programas do sistema operacional estão no disco, como o sistema operacional é iniciado?

Este problema é semelhante ao do livro As Aventuras do Barão de Munchausen. Um personagem tinha caído no meio de um bueiro, e se retirou agarrando suas botas e levantando. Nos primeiros dias da computação, o termo *bootstrap* era aplicado ao mecanismo usado para carregar o sistema operacional. Desde então, foi encurtado para "booting".

No hardware x86, o Sistema Básico de Entrada/Saída (BIOS) é responsável por carregar o sistema operacional. O BIOS procura no disco rígido pelo Master Boot Record (MBR), que deve estar localizado em um local específico do disco. O BIOS tem conhecimento suficiente para carregar e

executar o MBR, e assume que o MBR pode então executar o restante das tarefas envolvidas no carregamento do sistema operacional, possivelmente com a ajuda do BIOS.



O FreeBSD permite inicializar a partir do padrão mais antigo do MBR e da nova Tabela de Partição GUID (GPT). O particionamento GPT geralmente é encontrado em computadores com a Interface de Firmware Unificada e Extensivel (UEFI). No entanto, o FreeBSD pode inicializar a partir de partições de GPT mesmo em máquinas com apenas BIOS legado com o gptboot(8). O trabalho está em andamento para fornecer a inicialização direta a partir do UEFI.

O código dentro do MBR é normalmente chamado de *gerenciador de inicialização*, especialmente quando ele interage com o usuário. O gerenciador de inicialização geralmente tem mais código na primeira faixa do disco ou dentro do sistema de arquivos. Exemplos de gerenciadores de inicialização incluem o gerenciador de boot padrão do FreeBSD boot0, também chamado Boot Easy, e o Grub, que é usado por muitas distribuições Linux™.

Se apenas um sistema operacional estiver instalado, o MBR procura pelo primeiro slice inicializável (ativo) no disco e, em seguida, executa o código nesse slice para carregar o restante do sistema operacional. Quando vários sistemas operacionais estão presentes, um gerenciador de inicialização diferente pode ser instalado para exibir uma lista de sistemas operacionais para que o usuário possa selecionar um para inicializar.

O restante do sistema de boot do FreeBSD é dividido em três estágios. O primeiro estágio sabe apenas o suficiente para colocar o computador em um estado específico e executar o segundo estágio. O segundo estágio pode fazer um pouco mais, antes de executar o terceiro estágio. O terceiro estágio termina a tarefa de carregar o sistema operacional. O trabalho é dividido em três etapas porque o MBR coloca limites no tamanho dos programas que podem ser executados nos estágios um e dois. Encadear as tarefas juntas permite que o FreeBSD forneça um carregador mais flexível.

O kernel é então iniciado e começa a sondar os dispositivos e inicializá-los para uso. Quando o processo de inicialização do kernel é finalizado, o kernel passa o controle para o processo de usuário init(8), que garante que os discos estejam em estado utilizável, inicia a configuração de recursos no nível de usuário que monta sistemas de arquivos, configura placas de rede para se comunicar na rede e inicia os processos que foram configurados para serem executados na inicialização.

Esta seção descreve esses estágios em mais detalhes e demonstra como interagir com o processo de inicialização do FreeBSD.

12.2.1. O gerenciador de inicialização

O código do gerenciador de inicialização no MBR é às vezes chamado de *estágio zero* do processo de inicialização. Por padrão, o FreeBSD usa o gerenciador de boot boot0.

O MBR instalado pelo instalador do FreeBSD é baseado no /boot/boot0. O tamanho e a capacidade do boot0 são restritos a 446 bytes devido à tabela de slices e ao identificador `0x55AA` no final do MBR. Se o boot0 e vários sistemas operacionais estiverem instalados, uma mensagem semelhante a este exemplo será exibida no momento da inicialização:

Exemplo 27. Captura de tela do boot0

```
F1 Win
F2 FreeBSD

Default: F2
```

Outros sistemas operacionais sobrescreverão um MBR existente se forem instalados após o FreeBSD. Se isto acontecer, ou para substituir o MBR existente com o MBR do FreeBSD, use o seguinte comando:

```
# fdisk -B -b /boot/boot0 device
```

onde *device* é o disco de inicialização, como ad0 para o primeiro disco IDE, ad2 para o primeiro disco IDE em um segundo controlador IDE, ou da0 para o primeiro disco SCSI. Para criar uma configuração personalizada do MBR, consulte boot0cfg(8).

12.2.2. Estágio Um e Estágio Dois

Conceitualmente, o primeiro e o segundo estágios fazem parte do mesmo programa na mesma área do disco. Por causa das restrições de espaço, eles foram divididos em dois, mas são sempre instalados juntos. Eles são copiados do combinado /boot/boot pelo instalador do FreeBSD ou pelo `bsdlabel`.

Estes dois estágios estão localizados fora do sistema de arquivos, na primeira trilha do slice de inicialização, começando pelo primeiro setor. É ali onde o boot0, ou qualquer outro gerenciador de inicialização, espera encontrar um programa para executar, o qual continuará o processo de inicialização.

O primeiro estágio, boot1, é muito simples, pois pode ter apenas 512 bytes de tamanho. Ele sabe o suficiente sobre o FreeBSD *bsdlabel*, que armazena informações sobre o slice, para localizar e executar o boot2.

O estágio dois, boot2, é um pouco mais sofisticado, e entende o sistema de arquivos do FreeBSD o suficiente para encontrar arquivos. Ele pode fornecer uma interface simples para escolher o kernel ou loader para ser executado. Ele executa o loader, que é muito mais sofisticado e fornece um arquivo de configuração de inicialização. Se o processo de inicialização for interrompido no estágio dois, a seguinte tela interativa será exibida:

Exemplo 28. Captura de tela do boot2

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Para substituir o boot1 e boot2 instalados, use o `bsdlabel`, onde *diskslice* é o disco e o slice para inicializar, como ad0s1 para o primeiro slice no primeiro disco IDE:

```
# bsdlabel -B diskslice
```



Se apenas o nome do disco for usado, como ad0, o `bsdlabel` criará o disco no "modo perigosamente dedicado", sem slices. Esta provavelmente não é a ação desejada, então verifique novamente o *diskslice* antes de pressionar `Return`.

12.2.3. Estágio três

O loader é o estágio final do processo de bootstrap de três estágios. Ele está localizado no sistema de arquivos, geralmente como /boot/loader.

O loader é projetado como um método interativo para configuração, usando um conjunto de comandos embutidos, auxiliado por um interpretador mais poderoso que possui um conjunto de comandos mais complexo.

Durante a inicialização, o loader procurará por um console e por discos, e descobrirá de qual disco está sendo inicializado. Ele irá definir as variáveis de acordo, e um interpretador é iniciado onde os comandos do usuário podem ser passados a partir de um script ou usados interativamente.

O loader então lerá o /boot/loader.rc, que por padrão lê o /boot/defaults/loader.conf que define padrões razoáveis para variáveis e lê o /boot/loader.conf para mudanças locais nessas variáveis. O loader.rc então age sobre essas variáveis, carregando os módulos e o kernel selecionados.

Finalmente, por padrão, o loader realiza uma espera de 10 segundos por pressionamentos de teclas, e inicializa o kernel se não for interrompido. Se interrompido, o usuário é apresentado a um prompt que compreende o conjunto de comandos, no qual o usuário pode ajustar variáveis, descarregar todos os módulos, carregar módulos e finalmente inicializar ou reinicializar. Comandos Internos do Loader lista os comandos do loader mais usados. Para uma discussão completa de todos os comandos disponíveis, consulte loader(8).

Tabela 9. Comandos Internos do Loader

Variável	Descrição
autoboot *segundos*	Prossegue para inicializar o kernel se não for interrompido dentro do intervalo de tempo dado, em segundos. Ele exibe uma contagem regressiva e o intervalo de tempo padrão é de 10 segundos.

Variável	Descrição
boot `[-options] [kernelname]`	Imediatamente prossegue a inicialização do kernel, com qualquer opção especificada ou nome do kernel. Fornecer um nome de kernel na linha de comando só é aplicável depois que um `unload` foi emitido. Caso contrário, o kernel previamente carregado será usado. Se o *nomedokernel* não estiver qualificado, ele será pesquisado em */boot/kernel* e */boot/modules*.
boot-conf	Passa pela mesma configuração automática de módulos baseada em variáveis especificadas, mais comumente `kernel`. Isso só faz sentido se `unload` for usado primeiro, antes de alterar algumas variáveis.
help `[tópico]`	Mostra mensagens de ajuda lidas de /boot/loader.help. Se o tópico fornecido for `index`, a lista de tópicos disponíveis será exibida.
include *nomedoarquivo...*	Lê o arquivo especificado e interpreta-o linha por linha. Um erro interrompe imediatamente o `include`.
load `[-t type]` *filename*	Carrega o kernel, módulo do kernel ou arquivo do tipo especificado, com o nome de arquivo especificado. Quaisquer argumentos após o *nomedoarquivo* são passados para o arquivo. Se *nomedoarquivo* não estiver qualificado, ele será pesquisado em */boot/kernel* e */boot/modules*.
ls [-l] `[path]`	Exibe uma listagem de arquivos do caminho fornecido ou do diretório raiz, se o caminho não for especificado. Se `-l` for especificado, os tamanhos dos arquivos também serão mostrados.
lsdev [-v]	Lista todos os dispositivos dos quais é possível carregar módulos. Se `-v` for especificado, mais detalhes serão impressos.
lsmod [-v]	Exibe os módulos carregados. Se `-v` for especificado, mais detalhes serão mostrados.
more *nomedoarquivo*	Exibe os arquivos especificados, com uma pausa em cada `LINES` exibidas.
reboot	Reinicia imediatamente o sistema.
set *variable*, set *variable=value*	Define as variáveis de ambiente especificadas.
unload	Remove todos os módulos carregados.

Aqui estão alguns exemplos práticos de uso do loader. Para inicializar o kernel usual no modo

single-user :

```
boot -s
```

Para descarregar o kernel e os módulos usuais e, em seguida, carregar o kernel anterior ou outro especificado:

```
unload
load kernel.old
```

Use o kernel.GENERIC para se referir ao kernel padrão que vem com uma instalação, ou kernel.old, para se referir ao kernel previamente instalado antes de uma atualização do sistema ou antes de configurar um kernel personalizado.

Use o seguinte para carregar os módulos usuais com outro kernel:

```
unload
set kernel="kernel.old"
boot-conf
```

Para carregar um script de configuração do kernel automatizado:

```
load -t userconfig_script /boot/kernel.conf
```

12.2.4. Último estágio

Quando o kernel é carregado pelo loader ou pelo boot2, que ignora o loader, ele examina qualquer flag de inicialização e ajusta seu comportamento conforme necessário. Interação do Kernel durante o Boot lista os flags de inicialização comumente usados. Consulte boot(8) para obter mais informações sobre os outros sinalizadores de inicialização.

Tabela 10. Interação do Kernel durante o Boot

Opção	Descrição
`-a`	Durante a inicialização do kernel, solicita que o dispositivo seja montado como o sistema de arquivos raiz.
`-C`	Inicialize o sistema de arquivos raiz a partir de um CDROM.
`-s`	Inicialize no modo single-user.
`-v`	Seja mais detalhado durante a inicialização do kernel.

Uma vez que o kernel terminou a inicialização, ele passa o controle para o processo de usuário

init(8), localizado em /sbin/init, ou o caminho do programa especificado na variável `init_path` no `loader`. Este é o último estágio do processo de inicialização.

A sequência de inicialização garante que os sistemas de arquivos disponíveis no sistema estejam consistentes. Se um sistema de arquivos UFS não estiver e o `fsck` não puder corrigir as inconsistências, o init jogará o sistema no modo single-user para que o administrador do sistema possa resolver o problema diretamente. Caso contrário, o sistema é inicializado no modo multi-user.

12.2.4.1. Modo Single-User

Um usuário pode especificar este modo inicializando com `-s` ou definindo a variável `boot_single` no loader. Ele também pode ser alcançado executando o `shutdown now` do modo multi-user. O modo single-user começa com esta mensagem:

```
Enter full pathname of shell or RETURN for /bin/sh:
```

Se o usuário pressionar `Enter`, o sistema entrará no Bourne shell padrão. Para especificar um shell diferente, insira o caminho completo para o shell.

O modo single-user é geralmente usado para reparar um sistema que não inicializa devido a um sistema de arquivos inconsistente ou a um erro em um arquivo de configuração de inicialização. Ele também pode ser usado para redefinir a senha do `root` quando ela é desconhecida. Essas ações são possíveis porque o prompt do modo single-user fornece acesso local completo ao sistema e seus arquivos de configuração. Não há rede neste modo.

Embora o modo single-user seja útil para reparar um sistema, ele representa um risco de segurança, a menos que o sistema esteja em um local fisicamente seguro. Por padrão, qualquer usuário que possa obter acesso físico a um sistema terá controle total desse sistema após a inicialização no modo single-user.

Se o `console` do sistema for alterado para `insecure` em /etc/ttys, o sistema solicitará primeiro a senha do `root` antes de iniciar o modo single-user. Isso adiciona uma medida de segurança ao remover a capacidade de redefinir a senha do `root` quando ela é desconhecida.

Exemplo 29. Configurando um Console Inseguro em /etc/ttys

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off insecure
```

Um console `inseguro` significa que a segurança física para o console é considerada insegura, portanto, apenas alguém que conheça a senha do `root` pode usar o modo single-user.

12.2.4.2. Modo Multi-User

Se o init encontrar os sistemas de arquivos em ordem, ou quando o usuário tiver concluído seus comandos no modo de usuário único e tiver digitado `exit` para deixar o modo single-user, o sistema entra no modo multi-user, no qual inicia a configuração de recursos do sistema.

O sistema de configuração de recursos lê os padrões de configuração do /etc/defaults/rc.conf e detalhes específicos do sistema a partir do /etc/rc.conf. Em seguida, ele monta os sistemas de arquivos do sistema listados em /etc/fstab. Ele inicia serviços de rede, daemons diversos do sistema e, em seguida, os scripts de inicialização dos pacotes instalados localmente.

Para saber mais sobre o sistema de configuração de recursos, consulte rc(8) e examine os scripts localizados em /etc/rc.d.

12.3. Configurando telas iniciais de inicialização

Normalmente, quando um sistema FreeBSD inicializa, ele exibe seu progresso com uma série de mensagens no console. Uma tela inicial de inicialização cria uma tela de inicialização alternativa que oculta todo o probe de inicialização e as mensagens de inicialização de serviços. Algumas mensagens do boot loader, incluindo o menu de opções de inicialização e um prompt de contagem regressiva de espera, são exibidas no momento da inicialização, mesmo quando a tela inicial está ativada. A exibição da tela inicial pode ser desativada pressionando qualquer tecla do teclado durante o processo de inicialização.

Existem dois ambientes básicos disponíveis no FreeBSD. O primeiro é o ambiente padrão de linha de comando do console virtual legado. Depois que o sistema conclui a inicialização, é exibido um prompt de login do console. O segundo ambiente é um ambiente gráfico configurado. Consulte O sistema X Window para obter maiores informações sobre como instalar e configurar um gerenciador gráfico de tela e um gerenciador gráfico de login.

Depois que o sistema inicializa, a tela inicial é definida como proteção de tela. Após um período sem uso, a tela inicial será exibida e passará por etapas de mudança de intensidade da imagem, de brilhante a muito escuro e vice-versa. A configuração do protetor de tela inicial pode ser sobrescrita, adicionando-se uma linha `saver=` ao /etc/rc.conf. Vários protetores de tela embutidos estão disponíveis e descritos em splash(4). A opção `saver=` aplica-se apenas aos consoles virtuais e não tem efeito nos gerenciadores gráficos de telas.

Ao instalar o pacote ou port sysutils/bsd-splash-changer, uma imagem inicial aleatória de uma coleção será exibida na inicialização. A função tela inicial suporta 256 cores nos formatos bitmap (.bmp), ZSoft PCX (.pcx), ou TheDraw (.bin). A imagem .bmp, .pcx, ou .bin tem que ser colocada na partição root, em /boot por exemplo. Os arquivos de imagens iniciais tem que ter a resolução de 320 por 200 pixels ou menos para funcionarem em adaptadores VGA padrão. Para a tela inicial padrão de 256 cores e 320 por 200 pixels ou menos, adicione as seguintes linhas ao /boot/loader.conf. Substitua *splash.bmp* com o nome do arquivo bitmap a ser utilizado:

```
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Para usar um arquivo PCX em vez de um arquivo bitmap:

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

Em vez disso, use ASCII art no formato https://en.wikipedia.org/wiki/TheDraw:

```
splash_txt="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bin"
```

Outras opções interessantes do arquivo loader.conf incluem:

`beastie_disable="YES"`
: Isso impedirá que o menu de opções de inicialização seja exibido, mas o prompt de contagem regressiva da espera programada ainda estará presente. Mesmo com a exibição do menu de opções de inicialização desabilitada, entrar com uma seleção de opção no prompt de contagem decrescente de tempo programado ativará a opção de inicialização correspondente.

`loader_logo="beastie"`
: Isso substituirá as palavras padrão "FreeBSD", que são exibidas à direita do menu de opções de inicialização, com o logotipo colorido do beastie.

Para maiores informações, consulte splash(4), loader.conf(5), and vga(4).

12.4. Sugestões de dispositivos

Durante o começo da inicialização do sistema, o boot loader(8) lê o device.hints(5). Este arquivo armazena informações de inicialização do kernel conhecidas como variáveis, algumas vezes referenciadas como "sugestão de devices". Estas "sugestões de devices" são usados pelos drivers de dispositivo para configuração do dispositivo.

As sugestões de dispositivos também são especificadas no estágio 3 do prompt do boot loader, conforme demonstrado em Estágio três. As variáveis podem ser adicionadas usando `set`, removidas com `unset` e visualizadas `show`. Variáveis configuradas no arquivo /boot/device.hints também podem ser sobrescritas. As sugestões de dispositivos inseridas no boot loader não são permanentes e não serão aplicadas na próxima reinicialização.

Uma vez que o sistema é inicializado, kenv(1) pode ser usado para despejar todas as variáveis.

A sintaxe para o arquivo /boot/device.hints é uma variável por linha, usando o hash "#" como marcadores de comentário. As linhas são construídas da seguinte forma:

```
hint.driver.unit.keyword="value"
```

A sintaxe para o estágio 3 do boot loader é:

```
set hint.driver.unit.keyword=value
```

onde `driver` é o nome do driver de dispositivo, `unit` é o número da unidade de driver do dispositivo, e `keyword` é a palavra-chave sugerida. A palavra-chave pode consistir das seguintes opções:

- `at`: especifica o barramento ao qual o dispositivo está conectado.
- `port`: especifica o endereço inicial de I/O a ser usado.
- `irq`: especifica o número da requisição de interrupção a ser usada.
- `drq`: especifica o número do canal DMA.
- `maddr`: especifica o endereço de memória física ocupado pelo dispositivo.
- `flags`: define vários bits de flags para o dispositivo.
- `disabled`: se definido como `1`, o dispositivo é desativado.

Como os drivers de dispositivo podem aceitar ou exigir mais sugestões não listadas aqui, é recomendável exibir uma página de manual do driver. Para obter mais informações, consulte device.hints(5), kenv(1), loader.conf(5), e loader(8).

12.5. Sequência de Desligamento

Após desligamento controlado usando shutdown(8), o init(8) tentará executar o script /etc/rc.shutdown e, em seguida, enviará a todos os processos o sinal `TERM` e, subsequentemente, o sinal `KILL` para qualquer um que não termine em tempo hábil.

Para desligar uma máquina FreeBSD em arquiteturas e sistemas que suportam gerenciamento de energia, use o `shutdown -p now` para desligar a energia imediatamente. Para reinicializar um sistema FreeBSD, use o `shutdown -r now`. É preciso ser `root` ou um membro de `operator` para executar shutdown(8). Também é possível usar halt(8) e reboot(8). Consulte as páginas de manual e o shutdown(8) para obter mais informações.

Modifique a associação ao grupo referindo-se a Usuários e Gerenciamento Básico de Contas.



O gerenciamento de energia requer que o acpi(4) seja carregado como um módulo ou estaticamente compilado em um kernel personalizado.

Capítulo 13. Segurança

13.1. Sinopse

A segurança, seja física ou virtual, é um tópico tão amplo que todo um setor evoluiu em torno dele. Centenas de práticas padrão foram criadas sobre como proteger sistemas e redes e, como usuário do FreeBSD, é essencial entender como se proteger contra ataques e intrusos.

Neste capítulo, vários fundamentos e técnicas serão discutidos. O sistema FreeBSD vem com múltiplas camadas de segurança, e muitos outros utilitários de terceiros podem ser adicionados para aumentar a segurança.

Depois de ler este capítulo, você saberá:

- Conceitos básicos de segurança do sistema FreeBSD.
- Os vários mecanismos de criptografia disponíveis no FreeBSD.
- Como configurar a autenticação de senha única.
- Como configurar o TCP Wrapper para uso com o inetd(8).
- Como configurar o Kerberos no FreeBSD.
- Como configurar o IPsec e criar uma VPN.
- Como configurar e usar o OpenSSH no FreeBSD.
- Como usar ACLs para o sistema de arquivos .
- Como usar o pkg para auditar pacotes de software de terceiros instalados a partir da Coleção de Ports.
- Como utilizar os alertas de segurança do FreeBSD.
- O que é Auditoria de Processos e como ativá-la no FreeBSD.
- Como controlar os recursos do usuário usando classes de login ou o banco de dados de limites de recursos.

Antes de ler este capítulo, você deve:

- Entender os conceitos básicos do FreeBSD e de Internet.

Tópicos de segurança adicionais são abordados em outras partes deste Manual. Por exemplo, o Controle de Acesso Obrigatório é discutido em Controle de acesso obrigatório e os firewalls da Internet são discutidos em Firewalls.

13.2. Introdução

Segurança é responsabilidade de todos. Um ponto de entrada fraco em qualquer sistema pode permitir que intrusos obtenham acesso a informações críticas e causem estragos em toda a rede. Um dos princípios centrais da segurança da informação é a tríade CIA, que significa Confidencialidade, Integridade e Disponibilidade dos sistemas de informação.

A tríade CIA é um conceito básico de segurança de computadores, pois os clientes e usuários esperam que seus dados sejam protegidos. Por exemplo, um cliente espera que as informações do cartão de crédito sejam armazenadas com segurança (confidencialidade), que os pedidos não sejam alterados nos bastidores (integridade) e que tenham acesso às informações do pedido em todos os momentos (disponibilidade).

Para fornecer CIA, os profissionais de segurança aplicam uma estratégia de defesa em profundidade. A ideia de defesa em profundidade é adicionar várias camadas de segurança para evitar que uma falha em uma única camada e faça com que todo o sistema de segurança entre em colapso. Por exemplo, um administrador do sistema não pode simplesmente ativar um firewall e considerar a rede ou o sistema seguro. É preciso também auditar contas, verificar a integridade dos binários e garantir que ferramentas maliciosas não estejam instaladas. Para implementar uma estratégia de segurança eficaz, é preciso entender as ameaças e como se defender delas.

O que é uma ameaça no que se refere à segurança do computador? As ameaças não se limitam a invasores remotos que tentam acessar um sistema sem permissão de um local remoto. As ameaças também incluem funcionários, softwares mal-intencionados, dispositivos de rede não autorizados, desastres naturais, vulnerabilidades de segurança e até corporações concorrentes.

Sistemas e redes podem ser acessados sem permissão, às vezes por acidente, ou por atacantes remotos e, em alguns casos, por meio de espionagem corporativa ou ex-funcionários. Como usuário, é importante se preparar e admitir quando um erro levou a uma violação de segurança e relatar possíveis problemas à equipe de segurança. Como administrador, é importante conhecer as ameaças e estar preparado para mitigá-las.

Ao aplicar a segurança aos sistemas, recomenda-se começar protegendo as contas básicas e a configuração do sistema e, em seguida, proteger a camada de rede de modo a aderir à política do sistema e aos procedimentos de segurança da organização. Muitas organizações já possuem uma política de segurança que abrange a configuração de dispositivos de tecnologia. A política deve incluir a configuração de segurança de estações de trabalho, desktops, dispositivos móveis, telefones, servidores de produção e servidores de desenvolvimento. Em muitos casos, procedimentos operacionais padrão (SOPs) já existem. Em caso de dúvida, pergunte à equipe de segurança.

O restante desta introdução descreve como algumas dessas configurações básicas de segurança são executadas em um sistema FreeBSD. O restante deste capítulo descreve algumas ferramentas específicas que podem ser usadas ao implementar uma política de segurança em um sistema FreeBSD.

13.2.1. Prevenindo Logins

Ao garantir a segurança de um sistema, um bom ponto de partida é uma auditoria de contas. Assegure-se de que o `root` tenha uma senha forte e que essa senha não seja compartilhada. Desabilite todas as contas que não precisam de acesso de para logar.

Para negar acesso de login a contas, existem dois métodos. O primeiro é bloquear a conta. Este exemplo bloqueia a conta `toor`:

```
# pw lock toor
```

O segundo método é impedir o acesso ao login alterando o shell para /usr/sbin/nologin. Apenas o superusuário pode alterar o shell para outros usuários:

```
# chsh -s /usr/sbin/nologin toor
```

O shell /usr/sbin/nologin impede que o sistema atribua um shell ao usuário quando ele tenta efetuar login.

13.2.2. Escalonamento de Contas Permitido

Em alguns casos, a administração do sistema precisa ser compartilhada com outros usuários. O FreeBSD tem dois métodos para lidar com isso. O primeiro, que não é recomendado, é uma senha de root compartilhada usada por membros do grupo `wheel`. Com esse método, um usuário digita `su` e insere a senha para `wheel` sempre que o acesso do superusuário for necessário. O usuário deve então digitar `exit` para deixar o acesso privilegiado após terminar os comandos que requereram acesso administrativo. Para adicionar um usuário a este grupo, edite /etc/group e adicione o usuário ao final da entrada `wheel`. O usuário deve ser separado por um caractere vírgula sem espaço.

O segundo e recomendado método para permitir o escalonamento de privilégios é instalar o pacote ou port security/sudo. Este software fornece auditoria adicional, controle de usuário mais refinado e pode ser configurado para bloquear os usuários para que executem apenas os comandos privilegiados especificados.

Após a instalação, use o `visudo` para editar o /usr/local/etc/sudoers. Este exemplo cria um novo grupo `webadmin`, adiciona a conta `trhodes` a esse grupo e configura esse acesso de grupo para reiniciar o apache24:

```
# pw groupadd webadmin -M trhodes -g 6000
# visudo
%webadmin ALL=(ALL) /usr/sbin/service apache24 *
```

13.2.3. Hashes de Senhas

As senhas são um mal necessário da tecnologia. Quando elas devem ser usadas, elas devem ser complexas e um poderoso mecanismo de hash deve ser usado para criptografar a versão armazenada no banco de dados de senhas. O FreeBSD suporta os algoritmos de DES, MD5, SHA256, SHA512 e Blowfish na sua biblioteca `crypt()`. O padrão de SHA512 não deve ser alterado para um algoritmo hash menos seguro, mas pode ser alterado para o algoritmo Blowfish mais seguro.



O Blowfish não faz parte do AES e não é considerado compatível com nenhum Federal Information Processing Standard (FIPS). Seu uso pode não ser permitido em alguns ambientes.

Para determinar qual algoritmo de hash é usado para criptografar a senha de um usuário, o superusuário pode visualizar o hash do usuário no banco de dados de senhas do FreeBSD. Cada hash começa com um símbolo que indica o tipo de mecanismo de hash usado para criptografar a senha. Se DES for usado, não haverá símbolo de início. Para MD5, o símbolo é `$`. Para SHA256 e SHA512, o símbolo é `$6$`. Para o Blowfish, o símbolo é `$2a$`. Neste exemplo, a senha para `dru` é criptografada usando o algoritmo SHA512 padrão quando o hash começa com `$6$`. Observe que o hash criptografado, não a senha em si, é armazenado no banco de dados de senhas:

```
# grep dru /etc/master.passwd
dru:$6$pzIjSvCAn.PBYQBA$PXpSeWPx3g5kscj3IMiM7tUEUSPmGexxta.8Lt9TGSi2lNQqYGKszsBPuGME0:
1001:1001::0:0:dru:/usr/home/dru:/bin/csh
```

O mecanismo de hash é definido na classe de login do usuário. Para este exemplo, o usuário está na classe de login `default` e o algoritmo de hash é definido com esta linha em /etc/login.conf:

```
        :passwd_format=sha512:\
```

Para alterar o algoritmo para Blowfish, modifique a linha para ficar assim:

```
        :passwd_format=blf:\
```

Em seguida, execute `cap_mkdb /etc/login.conf` conforme descrito em Configurando Classes de Login. Observe que essa alteração não afetará os hashes de senha existentes. Isso significa que todas as senhas devem ser refeitas pedindo aos usuários que executem `passwd` para alterar sua senha.

Para logins remotos, a autenticação de dois fatores deve ser usada. Um exemplo de autenticação de dois fatores é "algo que você tem", como uma chave, e "algo que você conhece", como a senha para essa chave. Como o OpenSSH é parte do sistema básico do FreeBSD, todos os logins de rede devem ser sobre uma conexão criptografada e usar autenticação baseada em chave em vez de senhas. Para mais informações, consulte OpenSSH. Os usuários do Kerberos podem precisar fazer alterações adicionais para implementar o OpenSSH em sua rede. Essas alterações são descritas em Kerberos.

13.2.4. Aplicação de Política de Senha

Aplicar uma política de senha forte para contas locais é um aspecto fundamental da segurança do sistema. No FreeBSD, o tamanho da senha, a força da senha e a complexidade da senha podem ser implementados usando os Módulos de Autenticação Conectáveis (PAM).

Esta seção demonstra como configurar o tamanho mínimo e máximo da senha e a imposição de caracteres mistos usando o módulo pam_passwdqc.so. Este módulo é aplicado quando um usuário altera sua senha.

Para configurar este módulo, torne-se o superusuário e remova o comentário da linha contendo `pam_passwdqc.so` em /etc/pam.d/passwd. Em seguida, edite essa linha para corresponder à política de senha:

```
password        requisite       pam_passwdqc.so
min=disabled,disabled,disabled,12,10 similar=deny retry=3 enforce=users
```

Este exemplo define vários requisitos para novas senhas. A configuração `min` controla o tamanho mínimo da senha. Ele tem cinco valores porque este módulo define cinco tipos diferentes de senhas com base em sua complexidade. Complexidade é definida pelo tipo de caracteres que devem existir em uma senha, como letras, números, símbolos e maiúsculas e minúsculas. Os tipos de senhas são descritos em pam_passwdqc(8). Neste exemplo, os três primeiros tipos de senha são desativados, o que significa que as senhas que atendem a esses requisitos de complexidade não serão aceitas, independentemente da sua duração. O `12` define uma política de senha mínima de pelo menos doze caracteres, se a senha também contiver caracteres com três tipos de complexidade. O `10` define a política de senha para também permitir senhas de pelo menos dez caracteres, se a senha contiver caracteres com quatro tipos de complexidade.

A configuração `similar` nega senhas semelhantes à senha anterior do usuário. A configuração `retry` fornece ao usuário três oportunidades para inserir uma nova senha.

Depois que este arquivo for salvo, um usuário que alterar sua senha verá uma mensagem semelhante a seguinte:

```
% passwd
Changing local password for trhodes
Old Password:

You can now choose the new password.
A valid password should be a mix of upper and lower case letters,
digits and other characters.  You can use a 12 character long
password with characters from at least 3 of these 4 classes, or
a 10 character long password containing characters from all the
classes.  Characters that form a common pattern are discarded by
the check.
Alternatively, if no one else can see your terminal now, you can
pick this as your password: "trait-useful&knob".
Enter new password:
```

Se uma senha que não corresponde à política for inserida, ela será rejeitada com um aviso e o usuário terá a oportunidade de tentar novamente, até o número configurado de novas tentativas.

A maioria das políticas de senha exige que as senhas expirem depois de tantos dias. Para definir um tempo de expiração da senha no FreeBSD, defina `passwordtime` para a classe de login do usuário em /etc/login.conf. A classe de login `default` contém um exemplo:

```
#       :passwordtime=90d:\
```

Portanto, para definir uma expiração de 90 dias para esta classe de login, remova o símbolo de comentário (`#`), salve a edição e execute o `cap_mkdb /etc/login.conf`.

Para definir a expiração em usuários individuais, passe uma data de expiração ou o número de dias para expirar e um nome de usuário para o comando `pw`:

```
# pw usermod -p 30-apr-2015 -n trhodes
```

Como visto aqui, uma data de expiração é definida na forma de dia, mês e ano. Para obter maiores informações, consulte pw(8).

13.2.5. Detectando Rootkits

Um *rootkit* é qualquer software não autorizado que tente obter acesso como `root` a um sistema. Uma vez instalado, esse software mal-intencionado normalmente abrirá outro caminho de entrada para um invasor. Realisticamente, uma vez que um sistema foi comprometido por um rootkit e uma investigação foi realizada, o sistema deve ser reinstalado do zero. Existe um tremendo risco de que mesmo o engenheiro de sistemas ou segurança mais prudente perca algo que um invasor deixou para trás.

Um rootkit faz uma coisa útil para administradores: uma vez detectado, é um sinal de que um comprometimento aconteceu em algum momento. Mas, esses tipos de aplicativos tendem a ser muito bem ocultos. Esta seção demonstra uma ferramenta que pode ser usada para detectar rootkits, security/rkhunter.

Após a instalação deste pacote ou port, o sistema pode ser verificado usando o seguinte comando. Ele produzirá muitas informações e exigirá uma entrada manual da tecla `ENTER`:

```
# rkhunter -c
```

Depois que o processo for concluído, uma mensagem de status será impressa na tela. Esta mensagem incluirá a quantidade de arquivos verificados, arquivos suspeitos, possíveis rootkits e mais. Durante a verificação, alguns avisos de segurança genéricos podem ser produzidos sobre arquivos ocultos, a seleção do protocolo OpenSSH e versões vulneráveis conhecidas do software instalado. Estes podem ser tratados agora ou após uma análise mais detalhada ter sido realizada.

Todo administrador deve saber o que está sendo executado nos sistemas pelos quais é responsável. Ferramentas de terceiros como o rkhunter e o sysutils/lsof e comandos nativos como o `netstat` e o `ps`, podem mostrar uma grande quantidade de informações sobre o sistema. Faça anotações sobre o que é normal, faça perguntas quando algo parecer fora do lugar e seja paranoico. Embora evitar um comprometimento seja ideal, detectar um comprometimento é imprescindível.

13.2.6. Verificação Binária

A verificação de arquivos e binários do sistema é importante porque fornece às equipes de administração e segurança do sistema informações sobre alterações no sistema. Uma aplicação de software que monitora o sistema para alterações é chamado de Sistema de Detecção de Intrusão (IDS).

O FreeBSD fornece suporte nativo para um sistema de IDS básico. Embora os emails de segurança

noturnos notifiquem o administrador sobre alterações, as informações são armazenadas localmente e há uma chance de que um usuário mal-intencionado modifique essas informações para ocultar suas alterações no sistema. Como tal, recomenda-se criar um conjunto separado de assinaturas binárias e armazená-las em um diretório de read-only, propriedade do root ou, de preferência, em um disco USB removível ou servidor rsync remoto.

O utilitário `mtree` embutido pode ser usado para gerar uma especificação do conteúdo de um diretório. Um seed, ou uma constante numérica, é usada para gerar a especificação e é necessária para verificar se a especificação não foi alterada. Isso possibilita determinar se um arquivo ou binário foi modificado. Como o valor inicial do seed é desconhecido por um invasor, disfarçar ou impossibilitar a verificação dos valores de checksum dos arquivos será difícil ou impossível. O exemplo a seguir gera um conjunto de hashes SHA256, um para cada sistema binário no diretório /bin e salva esses valores em um arquivo oculto no diretório inicial do `root`, /root/.bin_chksum_mtree:

```
# mtree -s 3483151339707503 -c -K cksum,sha256digest -p /bin > /root/.bin_chksum_mtree
# mtree: /bin checksum: 3427012225
```

O *3483151339707503* representa o seed. Este valor deve ser lembrado, mas não compartilhado.

Visualizar o arquivo /root/.bin_cksum_mtree deve produzir uma saída semelhante à seguinte:

```
#          user: root
#       machine: dreadnaught
#          tree: /bin
#          date: Mon Feb  3 10:19:53 2014

# .
/set type=file uid=0 gid=0 mode=0555 nlink=1 flags=none
.               type=dir mode=0755 nlink=2 size=1024 \
                time=1380277977.000000000
    \133        nlink=2 size=11704 time=1380277977.000000000 \
                cksum=484492447 \

sha256digest=6207490fbdb5ed1904441fbfa941279055c3e24d3a4049aeb45094596400662a
    cat         size=12096 time=1380277975.000000000 cksum=3909216944 \

sha256digest=65ea347b9418760b247ab10244f47a7ca2a569c9836d77f074e7a306900c1e69
    chflags     size=8168 time=1380277975.000000000 cksum=3949425175 \

sha256digest=c99eb6fc1c92cac335c08be004a0a5b4c24a0c0ef3712017b12c89a978b2dac3
    chio        size=18520 time=1380277975.000000000 cksum=2208263309 \

sha256digest=ddf7c8cb92a58750a675328345560d8cc7fe14fb3ccd3690c34954cbe69fc964
    chmod       size=8640 time=1380277975.000000000 cksum=2214429708 \

sha256digest=a435972263bf814ad8df082c0752aa2a7bdd8b74ff01431ccbd52ed1e490bbe7
```

O nome do host da máquina, a data e a hora em que a especificação foi criada e o nome do usuário que criou a especificação são incluídos neste relatório. Há um checksum, tamanho, hora e um digest SHA256 para cada binário no diretório.

Para verificar se as assinaturas binárias não foram alteradas, compare o conteúdo atual do diretório com a especificação gerada anteriormente e salve os resultados em um arquivo. Este comando requer o seed que foi usado para gerar a especificação original:

```
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree >>
/root/.bin_chksum_output
# mtree: /bin checksum: 3427012225
```

Isso deve produzir o mesmo checksum para /bin que foi produzido quando a especificação foi criada. Se nenhuma alteração tiver ocorrido nos binários nesse diretório, o arquivo de saída /root/.bin_chksum_output estará vazio. Para simular uma alteração, altere a data no arquivo /bin/cat usando o `touch` e execute o comando de verificação novamente:

```
# touch /bin/cat
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree >>
/root/.bin_chksum_output
# more /root/.bin_chksum_output
cat changed
    modification time expected Fri Sep 27 06:32:55 2013 found Mon Feb  3 10:28:43 2014
```

Recomenda-se criar especificações para os diretórios que contêm binários e arquivos de configuração, bem como quaisquer diretórios que contenham dados sensíveis. Normalmente, as especificações são criadas para /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /etc e /usr/local/etc.

Existem sistemas de IDS mais avançados, como o security/aide. Na maioria dos casos, o `mtree` fornece a funcionalidade que os administradores precisam. É importante manter o valor inicial e a saída do checksum oculta de usuários mal-intencionados. Maiores informações sobre o `mtree` podem ser encontradas em mtree(8).

13.2.7. Otimizando o Sistema para Segurança

No FreeBSD, muitos recursos do sistema podem ser ajustados usando o `sysctl`. Alguns dos recursos de segurança que podem ser ajustados para impedir ataques de negação de serviço (DoS) serão abordados nesta seção. Mais informações sobre o uso do `sysctl`, incluindo como alterar temporariamente os valores e como tornar as alterações permanentes após o teste, podem ser encontradas em Efetuando ajustes com o sysctl(8).



Sempre que uma configuração é alterada com o `sysctl`, a chance de causar danos indesejados é aumentada, afetando a disponibilidade do sistema. Todas as alterações devem ser monitoradas e, se possível, testadas em um sistema de teste antes de serem usadas em um sistema de produção.

Por padrão, o kernel do FreeBSD é inicializado com um nível de segurança `-1`. Isso é chamado de

"modo inseguro" porque as flags de arquivos imutáveis podem ser desativadas e todos os dispositivos podem ser lidos ou gravados. O nível de segurança permanecerá em `-1`, a menos que seja alterado através do `sysctl` ou por uma configuração nos scripts de inicialização. O nível de segurança pode ser aumentado durante a inicialização do sistema, definindo `kern_securelevel_enable` para `YES` no arquivo /etc/rc.conf, e o valor de `kern_securelevel` para o nível de segurança desejado. Veja security(7) e init(8) para maiores informações sobre essas configurações e os níveis de segurança disponíveis.



Aumentar o valor da variável `securelevel` pode quebrar o Xorg e causar outros problemas. Esteja preparado para fazer alguma depuração.

As configurações da variável `net.inet.tcp.blackhole` e `net.inet.udp.blackhole` podem ser usadas para descartar pacotes SYN de entrada em portas fechadas sem enviar uma resposta RST. O comportamento padrão é retornar um RST para mostrar que uma porta está fechada. A alteração do padrão fornece algum nível de proteção contra varreduras de portas, que são usadas para determinar quais aplicativos estão sendo executados em um sistema. Defina `net.inet.tcp.blackhole` para `2` e `net.inet.udp.blackhole` para `1`. Consulte blackhole(4) para obter maiores informações sobre essas configurações.

As configurações das variáveis `net.inet.icmp.drop_redirect` e `net.inet.ip.redirect` ajudam a evitar *ataques de redirecionamento*. Um ataque de redirecionamento é um tipo de DoS que envia um grande número de pacotes ICMP tipo 5. Como esses pacotes não são necessários, configure `net.inet.icmp.drop_redirect` para `1` e configure `net.inet.ip.redirect` para `0`.

O roteamento de origem é um método para detectar e acessar endereços não roteáveis na rede interna. Isso deve ser desativado, pois endereços não roteáveis normalmente não são roteáveis de propósito. Para desativar este recurso, defina `net.inet.ip.sourceroute` e `net.inet.ip.accept_sourceroute` como `0`.

Quando uma máquina na rede precisa enviar mensagens para todos os hosts em uma sub-rede, uma mensagem de solicitação echo do ICMP é enviada para o endereço de broadcast. No entanto, não há motivo para um host externo executar essa ação. Para rejeitar todas as solicitações externas de transmissão, defina `net.inet.icmp.bmcastecho` como `0`.

Algumas configurações adicionais estão documentadas em security(7).

13.3. Senhas de Uso Unico

Por padrão, o FreeBSD inclui suporte para senhas de uso único em tudo (OPIE). O OPIE é projetado para evitar ataques repetidos, nos quais um atacante descobre a senha de um usuário e a usa para acessar um sistema. Como uma senha é usada apenas uma vez em OPIE, uma senha descoberta é de pouca utilidade para um invasor. O OPIE usa um hash seguro e um sistema de desafio/resposta para gerenciar senhas. A implementação do FreeBSD usa o hash MD5 por padrão.

O OPIE usa três tipos diferentes de senhas. A primeira é a senha usual UNIX™ ou Kerberos. A segunda é a senha única que é gerada pelo `opiekey`. O terceiro tipo de senha é a "senha secreta" que é usada para gerar senhas de uso único. A senha secreta não tem nada a fazer com ela e deve ser diferente da senha UNIX™.

Existem duas outras partes de dados importantes para o OPIE. Uma é o "seed" ou "chave", composta por duas letras e cinco dígitos. A outra é a "contagem de iteração", um número entre 1 e 100. O OPIE cria a senha única concatenando o seed e a senha secreta, aplicando o hash MD5 quantas vezes forem especificadas pela contagem de iterações e transformando o resultado em seis palavras inglesas curtas que representam a senha de uso único. O sistema de autenticação controla a última senha descartável usada e o usuário é autenticado se o hash da senha fornecida pelo usuário for igual à senha anterior. Como um hash unidirecional é usado, é impossível gerar futuras senhas de uso único se uma senha usada com êxito for capturada. A contagem de iteração é diminuída após cada login bem-sucedido para manter o usuário e o programa de login em sincronia. Quando a contagem de iterações descer para 1, o OPIE deve ser reinicializado.

Existem alguns programas envolvidos neste processo. Uma senha de uso único, ou uma lista consecutiva de senhas de uso único, é gerada passando uma contagem de iteração, um seed e uma senha secreta para o opiekey(1). Além de inicializar o OPIE, o opiepasswd(1) é usado para alterar senhas, contagens de iteração ou seeds. Os arquivos de credenciais relevantes em /etc/opiekeys são examinados pelo opieinfo(1) o qual imprime a iteração atual e o seed do usuário solicitante atual.

Esta seção descreve quatro tipos diferentes de operações. A primeira é como configurar senhas de uso único pela primeira vez em uma conexão segura. A segunda é como usar o `opiepasswd` em uma conexão insegura. A terceira é como efetuar login em uma conexão insegura. A quarta é como gerar um número de chaves que podem ser escritas ou impressas para uso em locais inseguros.

13.3.1. Inicializando o OPIE

Para inicializar o OPIE pela primeira vez, execute este comando a partir de um local seguro:

```
% opiepasswd -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:

ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

A opção `-c` define o modo de console que assume que o comando está sendo executado de um local seguro, como um computador sob o controle do usuário ou uma sessão SSH para um computador sob o controle do usuário.

Quando solicitado, insira a senha secreta que será usada para gerar as chaves de login de uso único. Essa senha deve ser difícil de adivinhar e deve ser diferente da senha associada à conta de login do usuário. Deve ter entre 10 e 127 caracteres. Lembre-se desta senha.

A linha `ID` lista o nome de login (`unfurl`), a contagem de iterações padrão (`499`) e o seed padrão (`to4268`). Ao efetuar o login, o sistema lembrará esses parâmetros e os exibirá, o que significa que

eles não precisam ser memorizados. A última linha lista a senha única gerada que corresponde a esses parâmetros e a senha secreta. No próximo login, use essa senha única.

13.3.2. Inicialização de uma Conexão Insegura

Para inicializar ou alterar a senha secreta em um sistema inseguro, é necessária uma conexão segura em algum lugar onde o `opiekey` possa ser executado. Isso pode ser um prompt de shell em uma máquina confiável. Uma contagem de iteração é necessária, em que 100 é provavelmente um bom valor, e o seed pode ser especificado ou a gerado aleatoriamente. Na conexão insegura, a máquina sendo inicializada, use opiepasswd(1):

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
    otp-md5 498 to4268 ext
    Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
    otp-md5 499 to4269
    Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Para aceitar o seed padrão, pressione `Return`. Antes de inserir uma senha de acesso, passe para a conexão segura e forneça os mesmos parâmetros:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Volte para a conexão insegura e copie a senha única gerada para o programa relevante.

13.3.3. Gerando uma Senha de Uso Único

Depois de inicializar o OPIE e efetuar login, um prompt como este será exibido:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)
```

```
login: <username>
otp-md5 498 gr4269 ext
Password:
```

Os prompts do OPIE fornecem um recurso útil. Se o `Enter` for pressionado no prompt de senha, o prompt ativará o echo e exibirá o que foi digitado. Isso pode ser útil ao tentar digitar uma senha manualmente a partir de uma impressão.

Neste ponto, gere a senha de uso único para responder a este aviso de login. Isso deve ser feito em um sistema confiável em que seja seguro executar o opiekey(1). Existem versões deste comando para Windows™, Mac OS™ e FreeBSD. Esse comando precisa da contagem de iteração e do seed como opções da linha de comandos. Use recortar e colar no prompt de login da máquina que está sendo conectada.

No sistema confiável:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Depois que a senha descartável for gerada, continue a logar.

13.3.4. Gerando Múltiplas Senhas de Uso Único

Às vezes, não há acesso a uma máquina confiável ou conexão segura. Neste caso, é possível usar o opiekey(1) para gerar algumas de senhas de uso único antecipadamente. Por exemplo:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

A opção `-n 5` solicita cinco chaves em seqüência e `30` especifica qual deve ser o último número de iteração. Note que estes são impressos na ordem *reversa* de uso. O usuário realmente paranóico pode querer escrever os resultados manualmente; caso contrário, imprima a lista. Cada linha mostra a contagem de iteração e a senha de uso único. Risque as senhas conforme elas forem usadas.

13.3.5. Restringindo o Uso de Senhas UNIX™

O OPIE pode restringir o uso de senhas UNIX™ com base no endereço IP de uma sessão de login. O arquivo relevante é o /etc/opieaccess, que está presente por padrão. Consulte opieaccess(5) para obter maiores informações sobre esse arquivo e sobre quais considerações de segurança você deve estar ciente ao usá-lo.

Aqui está um exemplo do arquivo opieaccess:

```
permit 192.168.0.0 255.255.0.0
```

Esta linha permite que os usuários cujo endereço de origem IP (que é vulnerável a spoofing) corresponda ao valor e à máscara especificados, para usar as senhas UNIX™ a qualquer momento.

Se nenhuma regra do arquivo opieaccess for correspondida, o padrão é negar logins que não sejam OPIE.

13.4. TCP Wrapper

O TCP Wrapper é um sistema de controle de acesso baseado em host que estende as habilidades do O super-servidor inetd. Ele pode ser configurado para fornecer suporte de registro, mensagens de retorno e restrições de conexão para os daemons do servidor sob o controle do inetd. Consulte tcpd(8) para obter maiores informações sobre o TCP Wrapper e seus recursos.

O TCP Wrapper não deve ser considerado um substituto para um firewall configurado adequadamente. Em vez disso, TCP Wrapper deve ser usado em conjunto com um firewall e outros aprimoramentos de segurança para fornecer outra camada de proteção na implementação de uma política de segurança.

13.4.1. Configuração Inicial

Para ativar o TCP Wrapper no FreeBSD, adicione as seguintes linhas ao arquivo /etc/rc.conf:

```
inetd_enable="YES"
inetd_flags="-Ww"
```

Então, configure corretamente o arquivo /etc/hosts.allow.



Ao contrário de outras implementações do TCP Wrapper, o uso do arquivo hosts.deny foi preterido no FreeBSD. Todas as opções de configuração devem ser colocadas no arquivo /etc/hosts.allow.

Na configuração mais simples, as políticas de conexão do daemon são configuradas para permitir ou bloquear, dependendo das opções no arquivo /etc/hosts.allow. A configuração padrão no FreeBSD é permitir todas as conexões para os daemons iniciados com o inetd.

A configuração básica geralmente assume a forma de `daemon : address : action`, onde `daemon` é o

daemon que o inetd iniciou, `address` é um nome de host válido ou um endereço IP ou um endereço IPv6 entre colchetes ([]) e `action` é `allow` ou `deny`. O TCP Wrapper usa uma semântica de correspondência de primeira regra, o que significa que o arquivo de configuração é varrido desde o início para uma regra correspondente. Quando uma correspondência é encontrada, a regra é aplicada e o processo de pesquisa é interrompido.

Por exemplo, para permitir conexões POP3 através do daemon mail/qpopper, as seguintes linhas devem ser anexadas ao arquivo hosts.allow:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

Sempre que este arquivo for editado, reinicie o inetd:

```
# service inetd restart
```

13.4.2. Configuração Avançada

O TCP Wrapper fornece opções avançadas para permitir mais controle sobre o modo como as conexões são tratadas. Em alguns casos, pode ser apropriado retornar um comentário para determinados hosts ou conexões de daemon. Em outros casos, uma entrada de log deve ser registrada ou um email enviado ao administrador. Outras situações podem exigir o uso de um serviço apenas para conexões locais. Isso tudo é possível através do uso de opções de configuração conhecidas como wildcards, caracteres de expansão e execução de comandos externos.

Suponha que uma situação ocorra onde uma conexão deva ser negada, mas uma razão deve ser enviada ao host que tentou estabelecer essa conexão. Essa ação é possível com a opção `twist`. Quando uma tentativa de conexão é feita, o `twist` executa um comando ou script shell. Existe um exemplo no arquivo hosts.allow:

```
# The rest of the daemons are protected.
ALL : ALL \
    : severity auth.info \
    : twist /bin/echo "You are not welcome to use %d from %h."
```

Neste exemplo, a mensagem "You are not allowed to use *daemon name* from *hostname*." será retornada para qualquer daemon não configurado no hosts.allow. Isso é útil para enviar uma resposta de volta ao inicializador de conexão logo após a conexão estabelecida ser descartada. Qualquer mensagem a ser retornada *deve* ser delimitada por caracteres de aspas duplas (").



Pode ser possível iniciar um ataque de negação de serviço no servidor se um invasor inunda esses daemons com solicitações de conexão.

Outra possibilidade é usar a opção `spawn`. Como a opção `twist`, a opção `spawn` implicitamente nega a conexão e pode ser usado para executar comandos ou scripts externos do shell. Ao contrário da `twist`, a `spawn` não enviará uma resposta ao host que estabeleceu a conexão. Por exemplo, considere

a seguinte configuração:

```
# We do not allow connections from example.com:
ALL : .example.com \
    : spawn (/bin/echo %a from %h attempted to access %d >> \
      /var/log/connections.log) \
    : deny
```

Isso negará todas as tentativas de conexão de `*.example.com` e registrará o nome do host, endereço IP e o daemon ao qual o acesso foi tentado no arquivo /var/log/connections.log. Este exemplo usa os caracteres de substituição `%a` e `%h`. Consulte hosts_access(5) para a lista completa.

Para corresponder a cada instância de um daemon, domínio ou endereço IP, use `ALL`. Outro wildcard é o `PARANOID`, que pode ser usado para corresponder a qualquer host que forneça um endereço IP que possa ser forjado, porque o endereço IP difere do nome resolvido para o host. Neste exemplo, todas as solicitações de conexão para o Sendmail que possuem um endereço IP que varia de seu nome de host serão negadas:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```



Usar o wildcard `PARANOID` resultará em conexões negadas se o cliente ou servidor tiver uma configuração de DNS incorreta.

Para saber mais sobre wildcards e sua funcionalidade associada, consulte hosts_access(5).



Ao adicionar novas linhas de configuração, certifique-se de que quaisquer entradas desnecessárias para esse daemon sejam comentadas no arquivo hosts.allow.

13.5. Kerberos

O Kerberos é um protocolo de autenticação de rede que foi originalmente criado pelo Instituto de Tecnologia de Massachusetts (MIT) como uma maneira segura de fornecer autenticação em uma rede potencialmente hostil. O protocolo Kerberos usa criptografia robusta para que tanto um cliente quanto um servidor possam provar sua identidade sem enviar nenhum segredo não criptografado pela rede. O Kerberos pode ser descrito como um sistema proxy de verificação de identidade e como um sistema confiável de autenticação de terceiros. Depois que um usuário autentica com Kerberos, suas comunicações podem ser criptografadas para garantir privacidade e integridade dos dados.

A única função do Kerberos é fornecer a autenticação segura de usuários e servidores na rede. Ele não fornece funções de autorização ou auditoria. Recomenda-se que o Kerberos seja usado com outros métodos de segurança que forneçam serviços de autorização e auditoria.

A versão atual do protocolo é a versão 5, descrita na RFC 4120. Várias implementações gratuitas

deste protocolo estão disponíveis, abrangendo uma ampla gama de sistemas operacionais. O MIT continua desenvolvendo o pacote Kerberos. É comumente usado no US como um produto de criptografia e, historicamente, está sujeito aos regulamentos de exportação dos US. No FreeBSD, o MITKerberos está disponível como o pacote ou port security/krb5. A implementação do Kerberos do Heimdal foi explicitamente desenvolvida fora do US para evitar regulamentações de exportação. A distribuição Kerberos do Heimdal está incluída na instalação base do FreeBSD, e outra distribuição com opções mais configuráveis está disponível como security/heimdal na Coleção de Ports.

No Kerberos, os usuários e serviços são identificados como "principals", que estão contidos em um agrupamento administrativo chamado de "realm". Um usuário principal típico teria o formato `user@REALM` (os realms são tradicionalmente em caracteres maiúsculos).

Esta seção fornece um guia sobre como configurar o Kerberos usando a distribuição Heimdal incluída no FreeBSD.

Para fins de demonstração de uma instalação do Kerberos , os namespaces serão os seguintes:

- O domínio (zona) de domínio DNS será `example.org`.
- O realm Kerberos será `EXAMPLE.ORG`.



Use nomes de domínio reais ao configurar o Kerberos, mesmo que ele seja executado internamente. Isso evita problemas de DNS e garante a interoperabilidade com outros realms do Kerberos.

13.5.1. Configurando um KDC do Heimdal

O Centro de Distribuição de Chaves (KDC) é o serviço de autenticação centralizada que o Kerberos fornece, a "a parte de terceiros confiáveis" do sistema. É o computador que emite os tíquetes Kerberos, que são usados para autenticação dos clientes nos servidores. Como o KDC é considerado confiável por todos os outros computadores no realm do Kerberos, isso aumenta as preocupações com a segurança. O acesso direto ao KDC deve ser limitado.

Embora a execução de um KDC exija poucos recursos de computação, uma máquina dedicada que atua apenas como um KDC é recomendada por motivos de segurança.

Para começar, instale o pacote security/heimdal assim:

```
# pkg install heimdal
```

Em seguida, edite o /etc/rc.conf como a seguir:

```
# sysrc kdc_enable=yes
# sysrc kadmind_enable=yes
```

Em seguida, edite o arquivo /etc/krb5.conf como a seguir:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
    kdc = kerberos.example.org
    admin_server = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Neste exemplo, o KDC usará o nome completo do host `kerberos.example.org`. O nome do host do KDC precisa ser resolvido no DNS.

O Kerberos também pode usar o DNS para localizar os KDCs, em vez de uma seção `[realms]` no arquivo /etc/krb5.conf. Para grandes organizações que possuem seus próprios servidores DNS, o exemplo acima pode ser reduzido para:

```
[libdefaults]
      default_realm = EXAMPLE.ORG
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Com as seguintes linhas sendo incluídas no arquivo de zona do domínio `example.org`:

```
_kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
_kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
_kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
_kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
_kerberos           IN  TXT     EXAMPLE.ORG
```



Para que os clientes possam encontrar os serviços Kerberos, eles *devem* ter um /etc/krb5.conf totalmente configurado ou um /etc/krb5.conf minimamente configurado *e* um servidor DNS corretamente configurado.

Em seguida, crie o banco de dados do Kerberos que contém as chaves de todos os principals (usuários e hosts) criptografados com uma senha master. Não é necessário lembrar essa senha, pois ela será armazenada no arquivo /var/heimdal/m-key; Seria razoável usar uma senha aleatória de 45 caracteres para essa finalidade. Para criar a chave master, execute `kstash` e digite uma senha:

```
# kstash
Master key: xxxxxxxxxxxxxxxxxxxxxxx
Verifying password - Master key: xxxxxxxxxxxxxxxxxxxxxxx
```

Depois que a chave master é criada, o banco de dados deve ser inicializado. A ferramenta administrativa do Kerberoskadmin(8) pode ser usada no KDC em um modo que opera diretamente

no banco de dados, sem usar o serviço de rede kadmind(8), como `kadmin -l`. Isso resolve o problema do ovo e da galinha de tentar se conectar ao banco de dados antes de criá-lo. No prompt do `kadmin`, use o `init` para criar o banco de dados inicial do realm:

```
# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
```

Por fim, enquanto ainda estiver no `kadmin`, crie o primeiro principal usando `add`. Atenha-se às opções padrão para o principal por enquanto, pois elas podem ser alteradas posteriormente com `modify`. Digite `?` no prompt para ver as opções disponíveis.

```
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Em seguida, inicie o serviço KDC executando:

```
# service kdc start
# service kadmind start
```

Embora não tenha nenhum daemon do kerberos em execução neste ponto, é possível confirmar que o KDC está funcionando obtendo um ticket para o principal que acabou de ser criado:

```
% kinit tillman
tillman@EXAMPLE.ORG's Password:
```

Confirme se um ticket foi obtido com sucesso usando `klist`:

```
% klist
Credentials cache: FILE:/tmp/krb5cc_1001
    Principal: tillman@EXAMPLE.ORG

  Issued                Expires               Principal
Aug 27 15:37:58 2013  Aug 28 01:37:58 2013  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

O ticket temporário pode ser destruído quando o teste terminar:

```
% kdestroy
```

13.5.2. Configurando um Servidor para Usar o Kerberos

A primeira etapa na configuração de um servidor para usar a autenticação Kerberos é garantir que ele tenha a configuração correta no arquivo /etc/krb5.conf. A versão do KDC pode ser usada como está, ou pode ser regenerada no novo sistema.

Em seguida, crie o arquivo /etc/krb5.keytab no servidor. Esta é a parte principal de "Kerberizar" um serviço - ele corresponde a gerar uma chave secreta compartilhada entre o serviço e o KDC. O segredo é uma chave criptográfica, armazenada em um "keytab". O keytab contém a chave do host do servidor, que permite que ele e o KDC verifiquem a identidade um do outro. Ele deve ser transmitido para o servidor de maneira segura, pois a segurança do servidor pode ser quebrada se a chave for tornada pública. Normalmente, o keytab é gerado na máquina confiável de um administrador usando o `kadmin`, e então transferido com segurança para o servidor, por exemplo, com scp(1); Ele também pode ser criado diretamente no servidor, se isso for consistente com a política de segurança desejada. É muito importante que o keytab seja transmitido para o servidor de forma segura: se a chave for conhecida por outra parte, essa parte pode representar qualquer usuário para o servidor! Usar o `kadmin` diretamente no servidor é conveniente, porque a entrada para o principal do host no banco de dados do KDC também é criada usando o `kadmin`.

Naturalmente, o `kadmin` é um serviço kerberizado; um tíquete Kerberos é necessário para autenticar-se no serviço de rede, mas para garantir que o usuário que está executando o `kadmin` esteja presente (e sua sessão não tenha sido invadida), o `kadmin` solicitará a senha para obter um novo ticket. O principal autenticando no serviço kadmin deve ter permissão para usar a interface `kadmin`, conforme especificado no arquivo /var/heimdal/kadmind.acl. Veja a seção intitulada "Administração Remota" em `info heimdal` para detalhes sobre a criação de listas de controle de acesso. Em vez de ativar o acesso remoto ao `kadmin`, o administrador pode conectar-se com segurança ao KDC através do console local ou por ssh() e executar a administração localmente usando o `kadmin -l`.

Depois de instalar o arquivo /etc/krb5.conf, use o `add --random-key` no `kadmin`. Isso adiciona o principal do host do servidor ao banco de dados, mas não extrai uma cópia da chave principal do host para um keytab. Para gerar o keytab, use `ext` para extrair a chave principal do host do servidor para seu próprio keytab:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
kadmin> ext_keytab host/myserver.example.org
kadmin> exit
```

Note que o `ext_keytab` por padrão armazena a chave extraída no arquivo /etc/krb5.keytab. Isso é

bom quando executado no servidor que está sendo kerberizado, mas o argumento `--keytab path/to/file` deve ser usado quando o keytab estiver sendo extraído em outro lugar:

```
# kadmin
kadmin> ext_keytab --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

O keytab pode então ser copiado com segurança para o servidor usando o scp(1) ou uma mídia removível. Certifique-se de especificar um nome de keytab não padrão para evitar a inserção de chaves desnecessárias na keytab do sistema.

Neste ponto, o servidor pode ler mensagens criptografadas do KDC usando sua chave compartilhada, armazenada no arquivo krb5.keytab. Agora ele está pronto para ativar os serviços de uso do Kerberos. Um dos serviços mais comuns é o sshd(8), que suporta o Kerberos através do GSS-API. No arquivo /etc/ssh/sshd_config, adicione a linha:

```
GSSAPIAuthentication yes
```

Depois de fazer essa alteração, o sshd(8) deve ser reiniciado para que a nova configuração tenha efeito: `service sshd restart`.

13.5.3. Configurando um cliente para usar o Kerberos

Assim como foi no servidor, o cliente requer configuração no arquivo /etc/krb5.conf. Copie o arquivo no local (com segurança) ou insira-o novamente conforme necessário.

Teste o cliente usando o `kinit`, `klist` e `kdestroy` a partir do cliente para obter, mostrar e excluir um ticket para um principal existente. Os aplicativos Kerberos também devem poder se conectar a servidores habilitados pelo Kerberos. Se isso não funcionar, mas a obtenção de um ticket ocorrer, provavelmente o problema está no servidor e não no cliente ou no KDC. No caso do ssh(1) kerberizado, o GSS-API está desabilitado por padrão, portanto teste usando `ssh -o GSSAPIAuthentication=yes hostname`.

Ao testar um aplicativo Kerberizado, tente usar um sniffer de pacote, como o `tcpdump`, para confirmar que nenhuma informação confidencial é enviada sem proteção.

Várias aplicações Kerberos cliente estão disponíveis. Com o advento de uma ponte para que aplicações usando SASL para autenticação possam usar mecanismos GSS-API, grandes classes de aplicativos clientes podem usar o Kerberos para autenticação, de clientes Jabber a clientes IMAP.

Os usuários em um realm geralmente têm seu principal Kerberos mapeado para uma conta de usuário local. Ocasionalmente, é necessário conceder acesso a uma conta de usuário local a alguém que não tenha um principal Kerberos correspondente. Por exemplo, `tillman@EXAMPLE.ORG` pode precisar de acesso à conta de usuário local `webdevelopers`. Outros diretores também podem precisar de acesso a essa conta local.

Os arquivos .k5login e .k5users, colocados no diretório home de um usuário, podem ser usados para resolver este problema. Por exemplo, se o seguinte .k5login for colocado no diretório inicial de

`webdevelopers`, os dois principals listados terão acesso a essa conta sem exigir uma senha compartilhada:

```
tillman@example.org
jdoe@example.org
```

Consulte ksu(1) para obter maiores informações sobre o .k5users.

13.5.4. Diferenças com a implementação do MIT

A principal diferença entre as implementações do MIT e a Heimdal é que o `kadmin` tem um conjunto de comandos diferente, mas equivalente, e usa um protocolo diferente. Se o KDC for MIT, a versão Heimdal do `kadmin` não poderá ser usada para administrar o KDC remotamente, e vice versa.

Aplicações cliente também podem usar opções de linha de comando ligeiramente diferentes para realizar as mesmas tarefas. Seguir as instruções em http://web.mit.edu/Kerberos/www/ é recomendado. Cuidado com os problemas de caminho: o port MIT é instalado em /usr/local/ por padrão, e os aplicativos do sistema FreeBSD serão executados em vez das versões do MIT se o `PATH` listar os diretórios do sistema primeiro.

Ao usar o MIT Kerberos como um KDC no FreeBSD, as seguintes edições também devem ser feitas no rc.conf:

```
kdc_program="/usr/local/sbin/kdc"
kadmind_program="/usr/local/sbin/kadmind"
kdc_flags=""
kdc_enable="YES"
kadmind_enable="YES"
```

13.5.5. Dicas, Truques e Solução de Problemas do Kerberos

Ao configurar e solucionar problemas do Kerberos, tenha em mente os seguintes pontos:

- Ao usar o Heimdal ou MITKerberos do ports, certifique-se de que o `PATH` liste as versões do port dos aplicativos clientes antes das versões do sistema.
- Se todos os computadores no realm não tiverem configurações de horário sincronizadas, a autenticação poderá falhar. Sincronização de Relógio com NTP descreve como sincronizar os relógios usando o NTP.
- Se o nome do host for alterado, o `host/` principal deve ser alterado e o keytab atualizado. Isso também se aplica a entradas de keytab especiais como o `HTTP/` principal usado para o www/mod_auth_kerb do Apache.
- Todos os hosts no realm devem ser resolvidos tanto de forma direta quanto reversa no DNS ou, no mínimo, no arquivo /etc/hosts. Os CNAMEs funcionarão, mas os registros A e PTR devem estar corretos e no lugar. A mensagem de erro para hosts não resolvidos não é intuitiva: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.

- Alguns sistemas operacionais que agem como clientes para o KDC não definem as permissões para o `ksu` para serem setuid `root`. Isso significa que o `ksu` não funciona. Este é um problema de permissões, não um erro do KDC.
- Com o MITKerberos, para permitir que um principal tenha uma duração de ticket maior que a duração padrão de dez horas, use `modify_principal` no kadmin(8) para alterar o `maxlife` do principal em questão e do `krbtgt` principal. O principal pode então usar o `kinit -l` para solicitar um ticket com uma vida útil mais longa.
- Ao executar um sniffer de pacotes no KDC para auxiliar na solução de problemas enquanto executa `kinit` de uma estação de trabalho, o Ticket de Concessão de Tickets (TGT) é enviado imediatamente, mesmo antes da digitação da senha. Isso ocorre porque o servidor Kerberos transmite livremente um TGT para qualquer solicitação não autorizada. No entanto, cada TGT é criptografado em uma chave derivada da senha do usuário. Quando um usuário digita sua senha, ela não é enviada para o KDC, ela é usada para descriptografar o TGT que o `kinit` já obteve. Se o processo de descriptografia resultar em um tíquete válido com um registro de data e hora válido, o usuário terá credenciais do Kerberos válidas. Essas credenciais incluem uma chave de sessão para estabelecer comunicações seguras com o servidor Kerberos no futuro, bem como o TGT, que é criptografado com a chave do próprio servidor Kerberos. Essa segunda camada de criptografia permite que o servidor Kerberos verifique a autenticidade de cada TGT.
- Os principals do host podem ter uma vida útil maior do ticket. Se o usuário do principal tiver uma vida útil de uma semana, mas o host ao qual está conectado tiver uma vida útil de nove horas, o cache do usuário terá um host principal expirado e o cache do ticket não funcionará como esperado.
- Ao configurar o arquivo krb5.dict para evitar que senhas incorretas específicas sejam usadas, conforme descrito em kadmind(8), lembre-que só se aplica a entidades que tenham uma política de senha atribuída a elas. O formato usado em krb5.dict é uma string por linha. Criar um link simbólico para /usr/shared/dict/words pode ser útil.

13.5.6. Atenuando as Limitações do Kerberos

Uma vez que com o Kerberos a abordagem é tudo ou nada, cada serviço habilitado na rede deve ser modificado para funcionar com o Kerberos ou ser protegido contra ataques de rede. Isso impede que as credenciais do usuário sejam roubadas e reutilizadas. Um exemplo é quando o Kerberos está habilitado em todos os shells remotos, mas o servidor de email POP3 não-Kerberizado envia senhas em texto simples.

O KDC é um ponto único de falha. Por design, o KDC deve ser tão seguro quanto seu banco de dados de senhas master. O KDC não deve ter absolutamente nenhum outro serviço sendo executado e deve estar fisicamente seguro. O perigo é alto porque o Kerberos armazena todas as senhas criptografadas com a mesma chave mestra que é armazenada como um arquivo no KDC.

Uma chave mestra comprometida não é tão ruim quanto se pode temer. A chave mestra é usada apenas para criptografar o banco de dados do Kerberos e como um seed para o gerador de números aleatórios. Desde que o acesso ao KDC seja seguro, um invasor não poderá fazer muito com a chave mestra.

Se o KDC não estiver disponível, os serviços de rede não poderão ser utilizados, pois a autenticação não poderá ser executada. Isso pode ser mitigado com um único KDC master e um ou mais slaves, e

com a implementação cuidadosa da autenticação secundária ou de fallback usando PAM.

O Kerberos permite que usuários, hosts e serviços se autentiquem entre si. Ele não possui um mecanismo para autenticar o KDC para os usuários, hosts ou serviços. Isso significa que um `kinit` infectado por um trojan pode registrar todos os nomes de usuário e senhas. As ferramentas de verificação de integridade do sistema de arquivos, como security/tripwire, podem mitigar isso.

13.5.7. Recursos e Outras Informações

- A FAQ do Kerberos
- Criando um Sistema de Autenticação: um Diálogo em Quatro Cenas
- RFC 4120, O Serviço de Autenticação em Rede (V5) do Kerberos
- Página Web do Kerberos MIT
- Página web do Heimdal Kerberos

13.6. OpenSSL

O OpenSSL é uma implementação de software livre dos protocolos SSL e TLS. Ele fornece uma camada de transporte de criptografia sobre a camada de comunicação normal, permitindo que ela seja entrelaçada com muitos aplicativos e serviços de rede.

A versão do OpenSSL incluída no FreeBSD suporta os protocolos de segurança de redes Secure Sockets Layer 3.0 (SSLv3) e Transport Layer Security 1.0/1.1/1.2 (TLSv1/TLSv1.1/TLSv1.2) e pode ser usado como uma biblioteca de criptografia geral. No FreeBSD 12.0-RELEASE e posterior, OpenSSL também suporta Transport Layer Security 1.3 (TLSv1.3).

O OpenSSL é muitas vezes usado para encriptar a autenticação de clientes de email e proteger transações baseadas na web como pagamentos com cartões de crédito. Alguns ports, como o www/apache24 e databases/postgresql11-server, incluem uma opção de compilação para inserir o OpenSSL. Se selecionado, o port vai adicionar suporte ao OpenSSL da base do sistema. Para ter o port compilado com o suporte do OpenSSL do port security/openssl, adicione o seguinte ao arquivo /etc/make.conf:

```
DEFAULT_VERSIONS+= ssl=openssl
```

Outro uso comum do OpenSSL é fornecer certificados para uso com aplicaçõe de software. Os certificados podem ser usados para verificar as credenciais de uma empresa ou indivíduo. Se um certificado não tiver sido assinado por uma *Autoridade de Certificação* externa (CA), como http://www.verisign.com, o aplicativo que usa o certificado produzirá um aviso. Há um custo associado à obtenção de um certificado assinado e o uso de um certificado assinado não é obrigatório, pois os certificados podem ser auto-assinados. No entanto, o uso de uma autoridade externa evitará avisos e poderá deixar os usuários mais à vontade.

Esta seção demonstra como criar e usar certificados em um sistema FreeBSD. Consulte Configurando um servidor LDAP para um exemplo de como criar uma CA para assinar seus próprios certificados.

Para obter maiores informações sobre o SSL, leia o OpenSSL Cookbook gratuito.

13.6.1. Gerando Certificados

Para gerar um certificado que será assinado por uma CA externa, emita o seguinte comando e insira as informações solicitadas nos prompts. Esta informação de entrada será gravada no certificado. No prompt `Common Name`, insira o nome completo para o sistema que usará o certificado. Se esse nome não corresponder ao servidor, a aplicação que estiver verificando o certificado emitirá um aviso para o usuário, tornando a verificação provida pelo certificado inútil.

```
# openssl req -new -nodes -out req.pem -keyout cert.key -sha256 -newkey rsa:2048
Generating a 2048 bit RSA private key
..................+++
.............................................................+++
writing new private key to 'cert.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:Another Name
```

Outras opções, como o tempo de expiração e algoritmos de criptografia alternativos, estão disponíveis ao criar um certificado. Uma lista completa de opções é descrita em openssl(1).

Este comando irá criar dois arquivos no diretório atual. A solicitação de certificado, req.pem, pode ser enviada para uma CA que validará as credenciais inseridas, assinará a solicitação e retornará o certificado assinado. O segundo arquivo, cert.key, é a chave privada do certificado e deve ser armazenado em um local seguro. Se ele cair nas mãos de outros, ele pode ser usado para representar o usuário ou o servidor.

Como alternativa, se uma assinatura de uma CA não for necessária, um certificado auto-assinado poderá ser criado. Primeiro, gere a chave RSA:

```
# openssl genrsa -rand -genkey -out cert.key 2048
```

```
0 semi-random bytes loaded
Generating RSA private key, 2048 bit long modulus
.............................................+++
.................................................................................
...........................+++
e is 65537 (0x10001)
```

Use essa chave para criar um certificado auto-assinado. Siga os prompts usuais para criar um certificado:

```
# openssl req -new -x509 -days 365 -key cert.key -out cert.crt -sha256
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (e.g. server FQDN or YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org
```

Isso criará dois novos arquivos no diretório atual: um arquivo de chave privada cert.key e o próprio certificado, cert.crt. Estes devem ser colocados em um diretório, preferencialmente sob /etc/ssl/, que é legível somente pelo `root`. As permissões de `0700` são apropriadas para esses arquivos e podem ser definidas usando o `chmod`.

13.6.2. Usando Certificados

Um uso para um certificado é criptografar conexões do servidor de email Sendmail para evitar o trafego de informações de autenticação em texto não criptografado.



> Alguns clientes de email exibirão um erro se o usuário não tiver instalado uma cópia local do certificado. Consulte a documentação incluída com o software para obter maiores informações sobre a instalação do certificado.

No FreeBSD 10.0-RELEASE e posterior, é possível criar um certificado auto-assinado para o Sendmail automaticamente. Para habilitar isso, adicione as seguintes linhas ao arquivo /etc/rc.conf:

```
sendmail_enable="YES"
sendmail_cert_create="YES"
sendmail_cert_cn="localhost.example.org"
```

Isso criará automaticamente um certificado auto-assinado, /etc/mail/certs/host.cert, uma chave de assinatura, /etc/mail/certs/host.key, e um certificado CA, /etc/mail/certs/cacert.pem. O certificado usará o `Common Name` especificado em `sendmail_cert_cn`. Depois de salvar as edições, reinicie o Sendmail:

```
# service sendmail restart
```

Se tudo correr bem, não haverá mensagens de erro no arquivo /var/log/maillog. Para um teste simples, conecte-se à porta de escuta do servidor de correio usando o `telnet`:

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.14.7/8.14.7; Fri, 18 Apr 2014 11:50:32 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Se a linha `STARTTLS` aparecer na saída, tudo está funcionando corretamente.

13.7. VPN Sobre IPsec

O Protocolo de Segurança da Internet (IPsec) é um conjunto de protocolos que se situam no topo da camada do Protocolo da Internet (IP). Ele permite que dois ou mais hosts se comuniquem de maneira segura, autenticando e criptografando cada pacote IP de uma sessão de comunicação. A pilha de rede IPsec do FreeBSD é baseada na implementação do http://www.kame.net/ e suporta as sessões IPv4 e IPv6.

O IPsec é composto pelos seguintes sub-protocolos:

- *Encapsulated Securtity Payload (ESP)*: este protocolo protege os dados do pacote IP da interferência de terceiros, criptografando o conteúdo usando algoritmos de criptografia simétricos, como Blowfish e 3DES.
- _Authentication Header (AH) _: este protocolo protege o cabeçalho do pacote IP da interferência

e spoofing de terceiros calculando um checksum criptográfico e gerando o hash dos campos de cabeçalho do pacote IP com uma função de hash segura. Isso é seguido por um cabeçalho adicional que contém o hash, para permitir que as informações no pacote sejam autenticadas.

- *IP Payload Compression Protocol (IPComp):* este protocolo tenta aumentar o desempenho da comunicação comprimindo o payload IP para reduzir a quantidade de dados enviados .

Esses protocolos podem ser usados juntos ou separadamente, dependendo do ambiente.

O IPsec suporta dois modos de operação. O primeiro modo, *Modo de Transporte*, protege as comunicações entre dois hosts. O segundo modo, *Modo de túnel*, é usado para construir túneis virtuais, comumente conhecidos como redes privadas virtuais (VPNs). Consulte ipsec(4) para obter informações detalhadas sobre o subsistema IPsec no FreeBSD.

O suporte a IPsec é ativado por padrão no FreeBSD 11 e posteriores. Para versões anteriores do FreeBSD, adicione estas opções a um arquivo de configuração de kernel personalizado e recompile o kernel usando as instruções em Configurando o kernel do FreeBSD:

```
options   IPSEC        IP security
device    crypto
```

Se o suporte a depuração do IPsec for desejado, a seguinte opção de kernel também deve ser adicionada:

```
options   IPSEC_DEBUG  debug for IP security
```

Este restante deste capítulo demonstra o processo de configuração de uma VPNIPsec entre uma rede doméstica e uma rede corporativa. No cenário de exemplo:

- Ambos os sites estão conectados à Internet através de um gateway que está executando o FreeBSD.
- O gateway em cada rede tem pelo menos um endereço IP externo. Neste exemplo, o endereço IP externo da LAN corporativa é `172.16.5.4` e o IP externo da LAN doméstica é `192.168.1.12`.
- Os endereços internos das duas redes podem ser endereços IP públicos ou privados. No entanto, o espaço de endereço não deve colidir. Por exemplo, ambas as redes não podem usar `192.168.1.x`. Neste exemplo, o endereço IP interno da LAN corporativa é `10.246.38.1` e o endereço do IP interno da LAN doméstica é `10.0.0.5`.

13.7.1. Configurando uma VPN no FreeBSD

Para começar, o security/ipsec-tools deve ser instalado a partir da Coleção de Ports. Este software fornece várias aplicações que suportam a configuração.

O próximo requisito é criar dois pseudo-dispositivos gif(4) que serão usados para encapsular pacotes e permitir que ambas as redes se comuniquem adequadamente. Como `root`, execute os seguintes comandos, substituindo *internal* e *external* pelos endereços IP reais das interfaces internas e externas dos dois gateways:

```
# ifconfig gif0 create
# ifconfig gif0 internal1 internal2
# ifconfig gif0 tunnel external1 external2
```

Verifique a configuração em cada gateway, usando o `ifconfig`. Aqui está a saída do Gateway 1:

```
gif0: flags=8051 mtu 1280
tunnel inet 172.16.5.4 --> 192.168.1.12
inet6 fe80::2e0:81ff:fe02:5881%gif0 prefixlen 64 scopeid 0x6
inet 10.246.38.1 --> 10.0.0.5 netmask 0xffffff00
```

Aqui está a saída do Gateway 2:

```
gif0: flags=8051 mtu 1280
tunnel inet 192.168.1.12 --> 172.16.5.4
inet 10.0.0.5 --> 10.246.38.1 netmask 0xffffff00
inet6 fe80::250:bfff:fe3a:c1f%gif0 prefixlen 64 scopeid 0x4
```

Depois de concluídos, os dois endereços de IP internos devem ser acessados usando ping(8):

```
priv-net# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=64 time=42.786 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=19.255 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=20.440 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=21.036 ms
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.255/25.879/42.786/9.782 ms
```

```
corp-net# ping 10.246.38.1
PING 10.246.38.1 (10.246.38.1): 56 data bytes
64 bytes from 10.246.38.1: icmp_seq=0 ttl=64 time=28.106 ms
64 bytes from 10.246.38.1: icmp_seq=1 ttl=64 time=42.917 ms
64 bytes from 10.246.38.1: icmp_seq=2 ttl=64 time=127.525 ms
64 bytes from 10.246.38.1: icmp_seq=3 ttl=64 time=119.896 ms
64 bytes from 10.246.38.1: icmp_seq=4 ttl=64 time=154.524 ms
--- 10.246.38.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.106/94.594/154.524/49.814 ms
```

Como esperado, ambos os lados têm a capacidade de enviar e receber pacotes ICMP dos endereços configurados de forma privada. Em seguida, os dois gateways devem ser informados sobre como rotear pacotes para enviar corretamente o tráfego de qualquer rede. Os seguintes comandos

atingirão esse objetivo:

```
corp-net# route add 10.0.0.0 10.0.0.5 255.255.255.0
corp-net# route add net 10.0.0.0: gateway 10.0.0.5
priv-net# route add 10.246.38.0 10.246.38.1 255.255.255.0
priv-net# route add host 10.246.38.0: gateway 10.246.38.1
```

Neste ponto, as máquinas internas devem ser alcançadas de cada gateway, bem como das máquinas atrás dos gateways. Novamente, use o ping(8) para confirmar:

```
corp-net# ping 10.0.0.8
PING 10.0.0.8 (10.0.0.8): 56 data bytes
64 bytes from 10.0.0.8: icmp_seq=0 ttl=63 time=92.391 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=63 time=21.870 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=63 time=198.022 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=63 time=22.241 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=63 time=174.705 ms
--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.870/101.846/198.022/74.001 ms

priv-net# ping 10.246.38.107
PING 10.246.38.1 (10.246.38.107): 56 data bytes
64 bytes from 10.246.38.107: icmp_seq=0 ttl=64 time=53.491 ms
64 bytes from 10.246.38.107: icmp_seq=1 ttl=64 time=23.395 ms
64 bytes from 10.246.38.107: icmp_seq=2 ttl=64 time=23.865 ms
64 bytes from 10.246.38.107: icmp_seq=3 ttl=64 time=21.145 ms
64 bytes from 10.246.38.107: icmp_seq=4 ttl=64 time=36.708 ms
--- 10.246.38.107 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.145/31.721/53.491/12.179 ms
```

Configurar os túneis é a parte fácil. Configurar um link seguro é um processo mais aprofundado. A seguinte configuração usa chaves RSA pré-compartilhadas (PSK). Além dos endereços IP, o arquivo /usr/local/etc/racoon/racoon.conf em ambos os gateways será idêntico e será semelhante a:

```
path    pre_shared_key  "/usr/local/etc/racoon/psk.txt"; #location of pre-shared key
file
log     debug;  #log verbosity setting: set to 'notify' when testing and debugging is
complete

padding # options are not to be changed
{
        maximum_length  20;
        randomize       off;
        strict_check    off;
        exclusive_tail  off;
}
```

```
timer   # timing options. change as needed
{
        counter         5;
        interval        20 sec;
        persend         1;
#       natt_keepalive  15 sec;
        phase1          30 sec;
        phase2          15 sec;
}

listen  # address [port] that racoon will listen on
{
        isakmp          172.16.5.4 [500];
        isakmp_natt     172.16.5.4 [4500];
}

remote  192.168.1.12 [500]
{
        exchange_mode   main,aggressive;
        doi             ipsec_doi;
        situation       identity_only;
        my_identifier   address 172.16.5.4;
        peers_identifier        address 192.168.1.12;
        lifetime        time 8 hour;
        passive         off;
        proposal_check  obey;
#       nat_traversal   off;
        generate_policy off;

                        proposal {
                                encryption_algorithm    blowfish;
                                hash_algorithm          md5;
                                authentication_method   pre_shared_key;
                                lifetime time           30 sec;
                                dh_group                1;
                        }
}

sainfo  (address 10.246.38.0/24 any address 10.0.0.0/24 any)    # address
$network/$netmask $type address $network/$netmask $type ( $type being any or esp)
{                               # $network must be the two internal networks you are
joining.
        pfs_group       1;
        lifetime        time    36000 sec;
        encryption_algorithm    blowfish,3des;
        authentication_algorithm        hmac_md5,hmac_sha1;
        compression_algorithm   deflate;
}
```

Para descrições de cada opção disponível, consulte a página de manual do racoon.conf.

O Banco de Dados da Política de Segurança (SPD) precisa ser configurado para que o FreeBSD e o racoon consigam criptografar e descriptografar o tráfego de rede entre os hosts.

Isso pode ser obtido com um shell script, semelhante ao seguinte, no gateway corporativo. Este arquivo será usado durante a inicialização do sistema e deve ser salvo como /usr/local/etc/racoon/setkey.conf.

```
flush;
spdflush;
# To the home network
spdadd 10.246.38.0/24 10.0.0.0/24 any -P out ipsec esp/tunnel/172.16.5.4-
192.168.1.12/use;
spdadd 10.0.0.0/24 10.246.38.0/24 any -P in ipsec esp/tunnel/192.168.1.12-
172.16.5.4/use;
```

Uma vez que o arquivo estiver no seu lugar, o racoon pode ser iniciado em ambos os gateways usando o seguinte comando:

```
# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf -l
/var/log/racoon.log
```

A saída deve ser semelhante à seguinte:

```
corp-net# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf
Foreground mode.
2006-01-30 01:35:47: INFO: begin Identity Protection mode.
2006-01-30 01:35:48: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:35:55: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:36:04: INFO: ISAKMP-SA established 172.16.5.4[500]-192.168.1.12[500]
spi:623b9b3bd2492452:7deab82d54ff704a
2006-01-30 01:36:05: INFO: initiate new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=28496098(0x1b2d0e2)
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.12[0] spi=47784998(0x2d92426)
2006-01-30 01:36:13: INFO: respond new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=124397467(0x76a279b)
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.12[0] spi=175852902(0xa7b4d66)
```

Para garantir que o túnel esteja funcionando corretamente, mude para outro console e use o tcpdump(1) para exibir o tráfego de rede usando o comando a seguir. Substitua `em0` pela placa de

interface de rede conforme necessário:

```
# tcpdump -i em0 host 172.16.5.4 and dst 192.168.1.12
```

Dados semelhantes aos seguintes devem aparecer no console. Caso contrário, há um problema e a depuração dos dados retornados será necessária.

```
01:47:32.021683 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xa)
01:47:33.022442 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xb)
01:47:34.024218 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xc)
```

Neste ponto, as duas redes devem estar disponíveis e parecem fazer parte da mesma rede. Muito provavelmente ambas as redes estão protegidas por um firewall. Para permitir que o tráfego flua entre elas, regras precisam ser adicionadas para liberar a passagem dos pacotes. Para o firewall ipfw(8), adicione as seguintes linhas ao arquivo de configuração do firewall:

```
ipfw add 00201 allow log esp from any to any
ipfw add 00202 allow log ah from any to any
ipfw add 00203 allow log ipencap from any to any
ipfw add 00204 allow log udp from any 500 to any
```



Os números das regras podem precisar ser alterados dependendo da configuração atual do host.

Para usuários do pf(4) ou do ipf(8) , as seguintes regras devem fazer o truque:

```
pass in quick proto esp from any to any
pass in quick proto ah from any to any
pass in quick proto ipencap from any to any
pass in quick proto udp from any port = 500 to any port = 500
pass in quick on gif0 from any to any
pass out quick proto esp from any to any
pass out quick proto ah from any to any
pass out quick proto ipencap from any to any
pass out quick proto udp from any port = 500 to any port = 500
pass out quick on gif0 from any to any
```

Finalmente, para permitir que a máquina inicie o suporte para a VPN durante a inicialização do sistema, adicione as seguintes linhas ao arquivo /etc/rc.conf:

```
ipsec_enable="YES"
ipsec_program="/usr/local/sbin/setkey"
```

```
ipsec_file="/usr/local/etc/racoon/setkey.conf" # allows setting up spd policies on
boot
racoon_enable="yes"
```

13.8. OpenSSH

O OpenSSH é um conjunto de ferramentas de conectividade de rede usadas para fornecer acesso seguro a máquinas remotas. Além disso, as conexões TCP/IP podem ser encapsuladas ou encaminhadas com segurança através de conexões SSH. O OpenSSH criptografa todo o tráfego para eliminar efetivamente a interceptação, o sequestro de conexão e outros ataques no nível da rede.

O OpenSSH é mantido pelo projeto OpenBSD e é instalado por padrão no FreeBSD. É compatível com os protocolos de versão 1 e 2 do SSH.

Quando os dados são enviados pela rede em um formato não criptografado, sniffers de rede posicionados em qualquer lugar entre o cliente e o servidor podem roubar as informações do usuário/senha ou os dados transferidos durante a sessão. O OpenSSH oferece uma variedade de métodos de autenticação e criptografia para evitar que isso aconteça. Mais informações sobre o OpenSSH estão disponíveis em http://www.openssh.com/.

Esta seção fornece uma visão geral dos utilitários embutidos de cliente para acessar com segurança outros sistemas e transferir arquivos com segurança de um sistema FreeBSD. Em seguida, descreve como configurar um servidor SSH em um sistema FreeBSD. Maiores informações estão disponíveis nas páginas man mencionadas neste capítulo.

13.8.1. Usando os Utilitários de Cliente SSH

Para logar em um servidor SSH, use `ssh` e especifique um nome de usuário que exista naquele servidor e o endereço IP ou nome de host do servidor. Se esta for a primeira vez que uma conexão foi feita ao servidor especificado, o usuário será solicitado a primeiro verificar a impressão digital do servidor:

```
# ssh user@example.com
The authenticity of host 'example.com (10.0.0.1)' can't be established.
ECDSA key fingerprint is 25:cc:73:b5:b3:96:75:3d:56:19:49:d2:5c:1f:91:3b.
Are you sure you want to continue connecting (yes/no)? yes
Permanently added 'example.com' (ECDSA) to the list of known hosts.
Password for user@example.com: user_password
```

O SSH utiliza um sistema de impressão digital de chaves para verificar a autenticidade do servidor quando o cliente se conecta. Quando o usuário aceita a impressão digital da chave digitando `yes` ao conectar-se pela primeira vez, uma cópia da chave é salva em .ssh/known_hosts no diretório pessoal do usuário. Futuras tentativas de login são verificadas em relação à chave salva e o `ssh` exibirá um alerta se a chave do servidor não corresponder à chave salva. Se isso ocorrer, o usuário deve primeiro verificar por que a chave foi alterada antes de continuar com a conexão.

Por padrão, versões recentes do OpenSSH aceitam apenas conexões SSH v2. Por padrão, o cliente

usará a versão 2 se possível e voltará para a versão 1 se o servidor não suportar a versão 2. Para forçar o `ssh` a usar somente o protocolo especificado, inclua `-1` ou `-2`. Opções adicionais são descritas em ssh(1).

Use o scp(1) para copiar com segurança um arquivo para ou de uma máquina remota. Este exemplo copia o arquivo COPYRIGHT do sistema remoto para um arquivo com o mesmo nome no diretório atual do sistema local:

```
# scp user@example.com:/COPYRIGHT COPYRIGHT
Password for user@example.com: *******
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

Como a impressão digital já foi verificada para esse host, a chave do servidor é verificada automaticamente antes de solicitar a senha do usuário.

Os argumentos passados para o `scp` são semelhantes ao comando `cp`. O arquivo ou arquivos para copiar é o primeiro argumento e o destino para copiar é o segundo. Como o arquivo é buscado pela rede, um ou mais dos argumentos do arquivo assumem o formato `user@host:<path_to_remote_file>`. Esteja ciente ao copiar recursivamente diretórios que o `scp` usa a opção `-r`, enquanto `cp` usa a `-R`.

Para abrir uma sessão interativa para copiar arquivos, use o `sftp`. Consulte sftp(1) para obter uma lista de comandos disponíveis enquanto estiver em uma sessão `sftp`.

13.8.1.1. Autenticação Baseada em Chave

Em vez de usar senhas, um cliente pode ser configurado para se conectar à máquina remota usando chaves. Para gerar chaves de autenticação RSA, use o `ssh-keygen`. Para gerar um par de chaves pública e privada, especifique o tipo de chave e siga os prompts. Recomenda-se proteger as chaves com uma senha memorável, mas difícil de se adivinhar.

```
% ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):  ①
Enter same passphrase again:                 ②
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:54Xm9Uvtv6H4NOo6yjP/YCfODryvUU7yWHzMqeXwhq8 user@host.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|                 |
|        . o..    |
|       .S*+*o    |
|      . O=Oo . . |
```

```
|       = Oo= oo..|
|      .oB.* +.oo.|
|       =OE**.o..=|
+----[SHA256]-----+
```

① Digite uma senha aqui. Pode conter espaços e símbolos.

② Digite novamente a senha para verificá-la.

A chave privada é armazenada no arquivo ~/.ssh/id_rsa e a chave pública é armazenada no arquivo ~/.ssh/id_rsa.pub. A chave *publica* deve ser copiada para ~/.ssh/ authorized_keys na máquina remota para que a autenticação baseada em chave funcione.



Muitos usuários acreditam que as chaves são seguras por design e usarão uma chave sem uma senha. Este é um comportamento *perigoso*. Um administrador pode verificar se um par de chaves está protegido por uma senha, visualizando a chave privada manualmente. Se o arquivo de chave privada contiver a palavra `ENCRYPTED`, o dono da chave está usando uma senha. Além disso, para proteger melhor os usuários finais, o termo `from` pode ser colocado no arquivo de chave pública. Por exemplo, adicionar `from "192.168.10.5"` na frente do prefixo `ssh-rsa` só permitirá que esse usuário específico efetue login a partir desse endereço IP.

As opções e arquivos variam de acordo com as diferentes versões do OpenSSH. Para evitar problemas, consulte ssh-keygen(1).

Se uma senha for usada, o usuário será solicitado a inserir a senha toda vez que uma conexão for feita ao servidor. Para carregar as chaves de SSH na memória e remover a necessidade de digitar a senha toda vez, use o ssh-agent(1) e o ssh-add(1).

A autenticação é feita pelo `ssh-agent`, usando as chaves privadas que estão carregadas nele. O `ssh-agent` pode ser usado para iniciar outro aplicativo como um shell ou um gerenciador de janelas.

Para usar o `ssh-agent` em um shell, inicie-o com um shell como um argumento. Adicione a identidade executando `ssh-add` e inserindo a senha para a chave privada. O usuário então poderá executar o `ssh` para se conectar em qualquer host que tenha a chave pública correspondente instalada. Por exemplo:

```
% ssh-agent csh
% ssh-add
Enter passphrase for key '/usr/home/user/.ssh/id_rsa':  ①
Identity added: /usr/home/user/.ssh/id_rsa (/usr/home/user/.ssh/id_rsa)
%
```

① Digite a senha para a chave.

Para usar o `ssh-agent` no Xorg, adicione uma entrada para ele em ~/.xinitrc. Isso fornece os serviços do `ssh-agent` para todos os programas iniciados no Xorg. Um exemplo do arquivo ~/.xinitrc pode ter esta aparência:

```
exec ssh-agent startxfce4
```

Isso inicia o `ssh-agent`, que, por sua vez, ativa o XFCE, sempre que o Xorg é iniciado. Uma vez que o Xorg tenha sido reiniciado para que as mudanças entrem em vigor, execute `ssh-add` para carregar todas as chaves SSH.

13.8.1.2. Tunelamento SSH

O OpenSSH tem a capacidade de criar um tunel para encapsular outro protocolo em uma sessão criptografada.

O comando a seguir informa ao `ssh` para criar um túnel para o telnet:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Este exemplo usa as seguintes opções:

`-2`
: Força o comando `ssh` a usar a versão 2 para conectar-se ao servidor.

`-N`
: Indica nenhum comando ou apenas túnel. Se omitido, o `ssh` inicia uma sessão normal.

`-f`
: Força o comando `ssh` a ser executado em segundo plano.

`-L`
: Indica um túnel local no formato *localport:remotehost:remoteport*.

`user@foo.example.com`
: O nome de login para usar no servidor SSH remoto especificado.

Um túnel SSH funciona criando um socket de escuta em `localhost` na `localport` especificada. Em seguida, ele encaminha quaisquer conexões recebidas em `localport` por meio da conexão SSH com o `remotehost:remoteport` especificado. No exemplo, a porta `5023` no cliente é encaminhada para a porta `23` na máquina remota. Como a porta 23 é usada pelo telnet, isso cria uma sessão telnet criptografada através de um túnel SSH.

Esse método pode ser usado para agrupar qualquer número de protocolos TCP inseguros, como SMTP, POP3 e FTP, como visto nos exemplos a seguir.

Exemplo 30. Criar um Túnel Seguro para SMTP

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

Isso pode ser usado em conjunto com `ssh-keygen` e contas de usuário adicionais para criar um ambiente de encapsulamento SSH mais uniforme. As chaves podem ser usadas no lugar de digitar uma senha e os túneis podem ser executados como um usuário separado.

Exemplo 31. Acesso Seguro de um Servidor POP3

Neste exemplo, há um servidor SSH que aceita conexões de fora. Na mesma rede, existe um servidor de email que executa um servidor POP3. Para verificar o e-mail de maneira segura, crie uma conexão SSH com o servidor SSH e encaminhe para o servidor de e-mail:

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

Quando o túnel estiver ativo e em execução, aponte o cliente de e-mail para enviar solicitações POP3 para `localhost` na porta 2110. Essa conexão será encaminhada com segurança pelo encapsulamento para `mail.example.com`.

Exemplo 32. Ignorando um Firewall

Alguns firewalls filtram as conexões de entrada e saída. Por exemplo, um firewall pode limitar o acesso de máquinas remotas às portas 22 e 80 para permitir apenas o SSH e navegação na web. Isso impede o acesso a qualquer outro serviço que use uma porta diferente de 22 ou 80.

A solução é criar uma conexão SSH com uma máquina fora do firewall da rede e usá-la para encapsular o serviço desejado:

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *******
```

Neste exemplo, um cliente Ogg Vorbis de streaming pode agora ser apontado para `localhost` na porta 8888, que será encaminhado para `music.example.com` na porta 8000, ignorando com êxito o firewall.

13.8.2. Ativando o Servidor SSH

Além de fornecer utilitários de cliente SSH embutidos, um sistema FreeBSD pode ser configurado como um servidor SSH, aceitando conexões de outros clientes SSH.

Para ver se o sshd está operando, use o comando service(8):

```
# service sshd status
```

Se o serviço não estiver em execução, adicione a seguinte linha ao arquivo /etc/rc.conf.

```
sshd_enable="YES"
```

Isso iniciará o sshd, o programa daemon para o OpenSSH, na próxima vez que o sistema for inicializado. Para iniciá-lo agora:

```
# service sshd start
```

A primeira vez que o sshd inicia em um sistema FreeBSD, as chaves de host do sistema serão criadas automaticamente e a impressão digital será exibida no console. Forneça aos usuários a impressão digital para que eles possam verificá-la na primeira vez que se conectarem ao servidor.

Consulte o sshd(8) para obter a lista de opções disponíveis ao iniciar o sshd e uma discussão mais completa sobre autenticação, processo de login e os vários arquivos de configuração.

Neste ponto, o sshd deve estar disponível para todos os usuários com um nome de usuário e senha no sistema.

13.8.3. Segurança do Servidor SSH

Enquanto o sshd é o recurso de administração remota mais usado para o FreeBSD, a força bruta e o drive por ataques são comuns a qualquer sistema exposto a redes públicas. Vários parâmetros adicionais estão disponíveis para evitar o sucesso desses ataques e serão descritos nesta seção.

É uma boa ideia limitar quais usuários podem efetuar login no servidor SSH e de onde usar a palavra-chave `AllowUsers` no arquivo de configuração do servidor OpenSSH. Por exemplo, para permitir que somente o `root` efetue login de `192.168.1.32`, inclua esta linha no arquivo /etc/ssh/sshd_config:

```
AllowUsers root@192.168.1.32
```

Para permitir que o usuário `admin` efetue login de qualquer lugar, liste esse usuário sem especificar um endereço IP:

```
AllowUsers admin
```

Multiplos usuários devem ser listados na mesma linha, assim:

```
AllowUsers root@192.168.1.32 admin
```

Depois de fazer alterações no arquivo /etc/ssh/sshd_config, informe o sshd para recarregar seu arquivo de configuração executando:

```
# service sshd reload
```



Quando essa palavra-chave é usada, é importante listar cada usuário que precisa efetuar login nesta máquina. Qualquer usuário que não esteja especificado nessa linha será bloqueado. Além disso, as palavras-chave usadas no arquivo de configuração do servidor OpenSSH fazem distinção entre maiúsculas e minúsculas. Se a palavra-chave não estiver escrita corretamente, incluindo esse detalhe, ela será ignorada. Sempre teste as alterações neste arquivo para garantir que as edições estejam funcionando conforme o esperado. Consulte o sshd_config(5) para verificar a ortografia e o uso das palavras-chave disponíveis.

Além disso, os usuários podem ser forçados a usar a autenticação de dois fatores por meio do uso de uma chave pública e privada. Quando necessário, o usuário pode gerar um par de chaves usando o ssh-keygen(1) e enviar ao administrador a chave pública. Este arquivo de chave será colocado no arquivo authorized_keys como descrito acima na seção cliente. Para forçar os usuários a usar apenas as chaves, a seguinte opção pode ser configurada:

```
AuthenticationMethods publickey
```



Não confunda o arquivo /etc/ssh/sshd_config com /etc/ssh/ssh_config (observe o `d` extra no primeiro nome do arquivo). O primeiro arquivo configura o servidor e o segundo arquivo configura o cliente. Consulte o ssh_config(5) para obter uma listagem das configurações do cliente disponíveis.

13.9. Listas de Controle de Acesso

As Listas de Controle de Acesso (ACLs) estendem o modelo de permissão padrão do UNIX™ em um compatível com o modo POSIX™.1e. Isso permite que um administrador aproveite um modelo de permissões mais refinado.

O kernel FreeBSD GENERIC fornece suporte a ACL para sistemas de arquivos UFS. Usuários que preferem compilar um kernel personalizado devem incluir a seguinte opção em seu arquivo de configuração do kernel personalizado:

```
options UFS_ACL
```

Se esta opção não for ativada na compilação, uma mensagem de aviso será exibida ao tentar montar um sistema de arquivos com o suporte a ACL. As ACLs dependem de atributos estendidos que são suportados nativamente pelo UFS2.

Este capítulo descreve como ativar o suporte a ACL e fornece alguns exemplos de uso.

13.9.1. Ativando o Suporte a ACL

As ACLs são habilitadas pela flag administrativa de tempo de montagem, `acls`, que podem ser adicionadas ao arquivo /etc/fstab. As flags de tempo de montagem também podem ser configuradas automaticamente de forma persistente usando-se o tunefs(8) para modificar um superbloco de flags ACLs no cabeçalho do sistema de arquivos. Em geral, é preferível usar flags de superbloco por vários motivos:

- A flag de superbloco não pode ser alterada por um remount usando `mount -u`, pois requer um `umount` completo e um `mount` completo. Isso significa que as ACLs não podem ser ativadas no sistema de arquivos raiz após a inicialização. Isso também significa que o suporte a ACL em um sistema de arquivos não pode ser alterado enquanto o sistema estiver em uso.
- Definir a flag de superbloco faz com que o sistema de arquivos seja sempre montado com a ACL ativada, mesmo que não haja uma entrada no fstab ou se os dispositivos forem reordenados. Isso evita a montagem acidental do sistema de arquivos sem o suporte a ACL.



É desejável desencorajar a montagem acidental sem que a ACL esteja habilitada porque coisas desagradáveis podem acontecer se ACLs estiverem habilitadas, e então desabilitadas e então reativadas sem limpar os atributos estendidos. Em geral, uma vez que as ACLs forem habilitadas em um sistema de arquivos, elas não devem ser desabilitadas, pois as proteções de arquivos resultantes podem não ser compatíveis com aquelas pretendidas pelos usuários do sistema e ACLs reativadas podem reconectar as ACLs anteriores aos arquivos que tiveram suas permissões alteradas, resultando em um comportamento imprevisível.

Os sistemas de arquivos com a ACL ativada exibirão um sinal de mais (`+`) nas configurações de permissão:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Neste exemplo, o directory1, directory2 e directory3 estão todos fazendo uso de ACLs, enquanto public_html não está.

13.9.2. Usando ACLs

As ACLs de um sistema de arquivos podem ser visualizadas usando `getfacl`. Por exemplo, para visualizar as configurações de ACL no arquivo test:

```
% getfacl test
    #file:test
    #owner:1001
    #group:1001
    user::rw-
```

```
    group::r--
    other::r--
```

Para alterar as configurações de ACL neste arquivo, use `setfacl`. Para remover todos os ACLs atualmente definidos de um arquivo ou sistema de arquivos, inclua `-k`. No entanto, o método preferido é usar `-b`, pois ela deixa os campos básicos necessários para que as ACLs funcionem.

```
% setfacl -k test
```

Para modificar as entradas padrões das ACLs, use `-m`:

```
% setfacl -m u:trhodes:rwx,group:web:r--,o::--- test
```

Neste exemplo, não havia entradas predefinidas, pois elas foram removidas pelo comando anterior. Este comando restaura as opções padrões e atribui as opções listadas. Se um usuário ou grupo for adicionado e não existir no sistema, um erro de `Invalid argument` será exibido.

Consulte getfacl(1) e setfacl(1) para maiores informações sobre as opções disponíveis para esses comandos.

13.10. Monitorando Problemas de Segurança de Terceiros

Nos últimos anos, o mundo da segurança fez muitas melhorias em como a avaliação de vulnerabilidades é tratada. A ameaça de invasão do sistema aumenta à medida que utilitários de terceiros são instalados e configurados para praticamente qualquer sistema operacional disponível atualmente.

A avaliação de vulnerabilidade é um fator importante na segurança. Enquanto o FreeBSD libera avisos para o sistema base, fazê-lo para cada utilitário de terceiros está além da capacidade do Projeto FreeBSD. Existe uma maneira de mitigar vulnerabilidades de terceiros e avisar os administradores sobre problemas de segurança conhecidos. Um utilitário do FreeBSD conhecido como pkg inclui opções explicitamente para este propósito.

O pkg pesquisa um banco de dados em busca de problemas de segurança. O banco de dados é atualizado e mantido pela equipe de segurança do FreeBSD e pelos desenvolvedores de ports.

Por favor, consulte as instruções para instalar o pkg.

A instalação fornece arquivos de configuração do periodic(8) para manter o banco de dados de auditoria do pkg e fornece um método programático para mantê-lo atualizado . Esta funcionalidade é ativada se `daily_status_security_pkgaudit_enable` estiver definido como `YES` em periodic.conf(5) . Certifique-se de que os e-mails de execução de segurança diários, que são enviados para a conta de e-mail do `root`, estejam sendo lidos.

Após a instalação e para auditar utilitários de terceiros como parte da Coleção de Ports a qualquer

momento, um administrador pode optar por atualizar o banco de dados e visualizar as vulnerabilidades conhecidas dos pacotes instalados, invocando:

```
# pkg audit -F
```

O pkg exibe as vulnerabilidades publicadas dos pacotes instalados:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <https://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-
0001020eed82.html>

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Ao apontar um navegador da web para a URL exibida, um administrador pode obter mais informações sobre a vulnerabilidade. Isto incluirá as versões afetadas, pela versão do port do FreeBSD, juntamente com outros sites que podem conter avisos de segurança.

O pkg é um poderoso utilitário e é extremamente útil quando acoplado com o ports-mgmt/portmaster.

13.11. Avisos de Segurança do FreeBSD

Como muitos produtores de sistemas operacionais de qualidade, o Projeto FreeBSD tem uma equipe de segurança responsável por determinar a data de fim de vida (EoL) para cada versão do FreeBSD e para fornecer atualizações de segurança para versões suportadas que ainda não atingiram sua EoL. Mais informações sobre a equipe de segurança do FreeBSD e as versões suportadas estão disponíveis na página de segurança do FreeBSD.

Uma tarefa da equipe de segurança é responder às vulnerabilidades de segurança reportadas no sistema operacional FreeBSD. Quando uma vulnerabilidade é confirmada, a equipe de segurança verifica as etapas necessárias para corrigir a vulnerabilidade e atualiza o código-fonte com a correção. Em seguida, publica os detalhes como um "Aviso de Segurança". Os avisos de segurança são publicados no site do FreeBSD e enviados para as listas de discussão freebsd-security-notifications, freebsd-security, e freebsd-announce.

Esta seção descreve o formato de um alerta de segurança do FreeBSD.

13.11.1. Formato de um Comunicado de Segurança

Aqui está um exemplo de um aviso de segurança do FreeBSD:

```
=============================================================================
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512
```

```
=============================================================================
FreeBSD-SA-14:04.bind                                        Security Advisory
                                                           The FreeBSD Project

Topic:          BIND remote denial of service vulnerability

Category:       contrib
Module:         bind
Announced:      2014-01-14
Credits:        ISC
Affects:        FreeBSD 8.x and FreeBSD 9.x
Corrected:      2014-01-14 19:38:37 UTC (stable/9, 9.2-STABLE)
                2014-01-14 19:42:28 UTC (releng/9.2, 9.2-RELEASE-p3)
                2014-01-14 19:42:28 UTC (releng/9.1, 9.1-RELEASE-p10)
                2014-01-14 19:38:37 UTC (stable/8, 8.4-STABLE)
                2014-01-14 19:42:28 UTC (releng/8.4, 8.4-RELEASE-p7)
                2014-01-14 19:42:28 UTC (releng/8.3, 8.3-RELEASE-p14)
CVE Name:       CVE-2014-0591

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit <URL:http://security.FreeBSD.org/>.

I.   Background

BIND 9 is an implementation of the Domain Name System (DNS) protocols.
The named(8) daemon is an Internet Domain Name Server.

II.  Problem Description

Because of a defect in handling queries for NSEC3-signed zones, BIND can
crash with an "INSIST" failure in name.c when processing queries possessing
certain properties.  This issue only affects authoritative nameservers with
at least one NSEC3-signed zone.  Recursive-only servers are not at risk.

III. Impact

An attacker who can send a specially crafted query could cause named(8)
to crash, resulting in a denial of service.

IV.  Workaround

No workaround is available, but systems not running authoritative DNS service
with at least one NSEC3-signed zone using named(8) are not vulnerable.

V.   Solution

Perform one of the following:

1) Upgrade your vulnerable system to a supported FreeBSD stable or
```

```
release / security branch (releng) dated after the correction date.

2) To update your vulnerable system via a source code patch:

The following patches have been verified to apply to the applicable
FreeBSD release branches.

a) Download the relevant patch from the location below, and verify the
detached PGP signature using your PGP utility.

[FreeBSD 8.3, 8.4, 9.1, 9.2-RELEASE and 8.4-STABLE]
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-release.patch
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-release.patch.asc
# gpg --verify bind-release.patch.asc

[FreeBSD 9.2-STABLE]
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-stable-9.patch
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-stable-9.patch.asc
# gpg --verify bind-stable-9.patch.asc

b) Execute the following commands as root:

# cd /usr/src
# patch < /path/to/patch

Recompile the operating system using buildworld and installworld as
described in <URL:https://www.FreeBSD.org/handbook/makeworld.html>.

Restart the applicable daemons, or reboot the system.

3) To update your vulnerable system via a binary patch:

Systems running a RELEASE version of FreeBSD on the i386 or amd64
platforms can be updated via the freebsd-update(8) utility:

# freebsd-update fetch
# freebsd-update install

VI.  Correction details

The following list contains the correction revision numbers for each
affected branch.

Branch/path                                                      Revision
- -------------------------------------------------------------------------
stable/8/                                                         r260646
releng/8.3/                                                       r260647
releng/8.4/                                                       r260647
stable/9/                                                         r260646
releng/9.1/                                                       r260647
releng/9.2/                                                       r260647
```

```
- -------------------------------------------------------------------------

To see which files were modified by a particular revision, run the
following command, replacing NNNNNN with the revision number, on a
machine with Subversion installed:

# svn diff -cNNNNNN --summarize svn://svn.freebsd.org/base

Or visit the following URL, replacing NNNNNN with the revision number:

<URL:https://svnweb.freebsd.org/base?view=revision&revision=NNNNNN>

VII. References

<URL:https://kb.isc.org/article/AA-01078>

<URL:http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0591>

The latest revision of this advisory is available at
<URL:http://security.FreeBSD.org/advisories/FreeBSD-SA-14:04.bind.asc>
-----BEGIN PGP SIGNATURE-----

iQIcBAEBCgAGBQJS1ZTYAAoJEO1n7NZdz2rnOvQP/2/68/s9Cu35PmqNtSZVVxVG
ZSQP5EGWx/lramNf9566iKxOrLRMq/h3XWcC4goVd+gZFrvITJSVOWSa7ntDQ7TO
XcinfRZ/iyiJbs/Rg2wLHc/t5oVSyeouyccqODYFbOwOlk35JjOTMUG1YcX+Zasg
ax8RV+7Zt1QSBkMlOz/myBLXUjlTZ3Xg2FXVsfFQW5/g2CjuHpRSFx1bVNX6ysoG
9DT58EQcYxIS8WfkHRbbXKh9I1nSfZ7/Hky/kTafRdRMrjAgbqFgHkYTYsBZeav5
fYWKGQRJulYfeZQ90yMTvlpF42DjCC3uJYamJnwDIu8OhS1WRBI8fQfr9DRzmRua
OK3BK9hUiScDZOJB6OqeVzUTfe7MAA4/UwrDtTYQ+PqAenv1PK8DZqwXyxA9ThHb
zKO3OwuKOVHJnKvpOcr+eNwo7jbnHlis0oBksj/mrq2P9m2ueF9gzCiq5Ri5Syag
Wssb1HUoMGwqU0roS8+pRpNC8YgsWpsttvUWSZ8u6Vj/FLeHpiV3mYXPVMaKRhVm
067BA2uj4Th1JKtGleox+Em0R7OFbCc/9aWC67wiqI6KRyit9pYiF3npph+7D5Eq
7zPsUdDd+qc+UTiLp3liCRp5w6484wWdhZO6wRtmUgxGjNkxFoNnX8CitzF8AaqO
UWWemqWuz3lAZuORQ9KX
=OQzQ
-----END PGP SIGNATURE-----
```

Todo comunicado de segurança usa o seguinte formato:

- Cada aviso de segurança é assinado pela chave PGP do Oficial de Segurança. A chave pública para o Oficial de Segurança pode ser verificada em Chaves OpenPGP.
- O nome do alerta de segurança sempre começa com `FreeBSD-SA-` (para o FreeBSD Security Advisory), seguido pelo ano em formato de dois dígitos (`14:`), seguido pelo número de aviso para aquele ano (`04.`), seguido pelo nome do aplicativo ou subsistema afetado (`bind`). O comunicado mostrado aqui é o quarto comunicado de 2014 e afeta o BIND.
- O campo `Topic` resume a vulnerabilidade.
- O campo `Category` refere-se à parte afetada do sistema, que pode ser uma de `core`, `contrib`, ou `ports`. A categoria `core` significa que a vulnerabilidade afeta um componente principal do sistema operacional FreeBSD. A categoria `contrib` significa que a vulnerabilidade afeta um

software incluído no FreeBSD, como o BIND. A categoria `ports` indica que a vulnerabilidade afeta um software disponível através da coleção de ports.

- O campo `Module` refere-se ao local do componente. Neste exemplo, o módulo `bind` é afetado; Portanto, essa vulnerabilidade afeta um aplicativo instalado com o sistema operacional.
- O campo `Announced` reflete a data em que o comunicado de segurança foi publicado. Isto significa que a equipe de segurança verificou que o problema existe e que um patch foi disponibilizado no repositório do código fonte do FreeBSD.
- O campo `Credits` dá crédito ao indivíduo ou organização que encontrou a vulnerabilidade e a relatou.
- O campo `Affects` explica quais versões do FreeBSD são afetadas por esta vulnerabilidade.
- O campo `Corrected` indica a data, a hora, o deslocamento do horário e as releases que foram corrigidas. A seção entre parênteses mostra cada branch para a qual a correção foi mesclada e o número de versão da liberação correspondente dessa branch. O próprio identificador de release inclui o número da versão e, se apropriado, o nível do patch. O nível de correção é a letra `p` seguida de um número, indicando o número de seqüência do patch, permitindo que os usuários controlem quais patches já foram aplicados ao sistema.
- O campo `CVE Name` lista o número de aviso, se existir, no banco de dados público cve.mitre.org de vulnerabilidades de segurança.
- O campo `Background` fornece uma descrição do módulo afetado.
- O campo `Problem Description` explica a vulnerabilidade. Isso pode incluir informações sobre o código defeituoso e como o utilitário pode ser usado de maneira mal-intencionada.
- O campo `Impact` descreve o tipo de impacto que o problema pode ter em um sistema.
- O campo `Workaround` indica se uma solução alternativa está disponível para os administradores do sistema que não podem corrigir imediatamente o sistema.
- O campo `Solution` fornece as instruções para corrigir o sistema afetado. Este é um método testado e verificado passo a passo para obter um sistema corrigido e funcionando com segurança.
- O campo `Correction Details` exibe cada branch do Subversion afetada com o número de revisão que contém o código corrigido.
- O campo `References` oferece fontes de informações adicionais sobre a vulnerabilidade.

13.12. Auditoria de Processo

A auditoria de processos é um método de segurança no qual um administrador pode controlar os recursos do sistema utilizados e sua alocação entre os usuários, fornecer monitoramento do sistema e controlar minimamente os comandos de um usuário.

A auditoria de processos tem pontos positivos e negativos. Um dos pontos positivos é que uma intrusão pode ser rastreada ao ponto de entrada. Um valor negativo é a quantidade de logs gerados pela contabilidade do processo e o espaço em disco necessário. Esta seção conduz um administrador pelos fundamentos da contabilidade de processo.



Se uma auditoria mais detalhada for necessária, consulte Auditoria de Evento de

> Segurança.

13.12.1. Ativando e Utilizando a Auditoria de Processos

Antes de usar a auditoria de processos, ela deve ser ativada usando os seguintes comandos:

```
# sysrc accounting_enable=yes
# service accounting start
```

As informações de auditoria são armazenadas em arquivos localizados em /var/account, que são criados automaticamente, se necessário, na primeira vez em que o serviço de auditoria é iniciado. Esses arquivos contêm informações confidenciais, incluindo todos os comandos executados por todos os usuários. O acesso de escrita aos arquivos é limitado ao `root` e o acesso de leitura é limitado ao `root` e aos membros do grupo `wheel`. Para também impedir que membros do grupo `wheel` leiam os arquivos, altere a permissão do diretório /var/account para permitir acesso apenas de `root`.

Uma vez ativada, a auditoria começará a rastrear informações, como estatísticas de CPU e comandos executados. Todos os logs auditados estão em um formato não legível que pode ser visualizado usando `sa`. Se executado sem nenhuma opção, o `sa` imprime informações relacionadas ao número de chamadas por usuário, o tempo total decorrido em minutos, o total de CPU e o tempo do usuário em minutos, e o número médio de operações de I/O. Consulte sa(8) para obter a lista de opções disponíveis que controlam a saída.

Para exibir os comandos emitidos pelos usuários, use o `lastcomm`. Por exemplo, este comando imprime todo o uso do comando `ls` pelo usuário `trhodes` no terminal `ttyp1`:

```
# lastcomm ls trhodes ttyp1
```

Muitas outras opções úteis existem e são explicadas em lastcomm(1), acct(5) e sa(8).

13.13. Limites de Recursos

O FreeBSD fornece vários métodos para um administrador limitar a quantidade de recursos do sistema que um indivíduo pode usar. As cotas de disco limitam a quantidade de espaço em disco disponível para os usuários. As cotas são discutidas em Cotas de Disco.

Limites para outros recursos, como CPU e memória, podem ser definidos usando um arquivo simples ou um comando para configurar um banco de dados de limites de recursos. O método tradicional define classes de login editando o arquivo /etc/login.conf. Embora esse método ainda seja suportado, qualquer alteração requer um processo de várias etapas para editar esse arquivo, reconstruir o banco de dados de recursos, fazer as alterações necessárias no arquivo /etc/master.passwd e reconstruir o banco de dados de senhas. Isso pode se tornar demorado, dependendo do número de usuários a serem configurados.

O comando `rctl` pode ser usado para fornecer um método mais refinado para controlar limites de recursos. Esse comando suporta mais que limites de usuário, já que também pode ser usado para

definir restrições de recursos em processos e jails.

Esta seção demonstra os dois métodos para controlar recursos, começando com o método tradicional.

13.13.1. Configurando Classes de Login

No método tradicional, as classes de login e os limites de recursos a serem aplicados a uma classe de login são definidos no arquivo /etc/login.conf. Cada conta de usuário pode ser atribuída a uma classe de login, onde `default` é a classe de login padrão. Cada classe de login possui um conjunto de recursos de login associados a ele. Um recurso de login é um par `name=value`, em que *name* é um identificador conhecido e *value* é uma string arbitrária que é processada de acordo, dependendo do *name*.



Sempre que o arquivo /etc/login.conf for editado, o /etc/login.conf.db deve ser atualizado executando o seguinte comando:

```
# cap_mkdb /etc/login.conf
```

Os limites de recursos diferem dos recursos de login padrão de duas maneiras. Primeiro, para cada limite, existe um limite *soft* e um *hard*. Um limite soft pode ser ajustado pelo usuário ou aplicativo, mas não pode ser definido como superior ao limite hard. O limite hard pode ser baixado pelo usuário, mas só pode ser aumentado pelo root. Segundo, a maioria dos limites de recursos se aplica por processo a um usuário específico.

Limites de Recursos de Classe de Login lista os limites de recursos mais usados. Todos os limites de recursos disponíveis e capabilities são descritos em detalhes em login.conf(5).

Tabela 11. Limites de Recursos de Classe de Login

Limite de Recurso	Descrição
coredumpsize	O limite do tamanho de um arquivo core gerado por um programa é subordinado a outros limites de uso do disco, como `filesize` ou cotas de disco. Esse limite é frequentemente usado como um método menos severo de controle do consumo de espaço em disco. Como os usuários não geram arquivos core e geralmente não os excluem, essa configuração pode evitar que eles fiquem sem espaço em disco caso ocorra um grande travamento de programa.

Limite de Recurso	Descrição
cputime	A quantidade máxima de tempo de CPU que o processo de um usuário pode consumir. Os processos ofensivos serão eliminados pelo kernel. Este é um limite no *tempo* de CPU consumido, não a porcentagem do CPU como exibido em alguns dos campos gerados pelo `top` e `ps`.
filesize	O tamanho máximo de um arquivo que o usuário pode possuir. Ao contrário das cotas de disco (Cotas de Disco), esse limite é imposto em arquivos individuais, não no conjunto de todos os arquivos que um usuário possui.
maxproc	O número máximo de processos de primeiro plano e de plano de fundo que um usuário pode executar. Esse limite pode não ser maior que o limite do sistema especificado pela variável `kern.maxproc`. Definir um limite muito pequeno pode prejudicar a produtividade de um usuário, pois algumas tarefas, como compilar um programa grande, iniciam muitos processos.
memorylocked	A quantidade máxima de memória que um processo pode solicitar para ser bloqueado na memória principal usando o mlock(2). Alguns programas críticos do sistema, como amd(8), se bloqueiam na memória principal para que, se o sistema começar a fazer swap, eles não contribuam para surrar o disco.
memoryuse	A quantidade máxima de memória que um processo pode consumir a qualquer momento. Inclui tanto a memória principal quanto o uso de swap. Este não é um limite geral para restringir o consumo de memória, mas é um bom começo.
openfiles	O número máximo de arquivos que um processo pode ter aberto. No FreeBSD, os arquivos são usados para representar sockets e canais IPC, então tome cuidado para não definir isso muito baixo. O limite de todo o sistema para isso é definido por pela variável `kern.maxfiles`.
sbsize	O limite na quantidade de memória de rede que um usuário pode consumir. Isso geralmente pode ser usado para limitar as comunicações da rede.

Limite de Recurso	Descrição
stacksize	O tamanho máximo de uma pilha de processos. Isso por si só não é suficiente para limitar a quantidade de memória que um programa pode usar, por isso deve ser usado em conjunto com outros limites.

Existem algumas outras coisas para se lembrar ao definir limites de recursos:

- Os processos iniciados na inicialização do sistema pelo /etc/rc são atribuídos à classe `daemon` de login.
- Embora o arquivo /etc/login.conf padrão seja uma boa fonte de valores razoáveis para a maioria dos limites, eles podem não ser apropriados para todos os sistemas. Definir um limite muito alto pode abrir o sistema para uso abusivo, enquanto que defini-lo como muito baixo pode prejudicar a produtividade.
- O Xorg utiliza muitos recursos e incentiva os usuários a executarem mais programas simultaneamente.
- Muitos limites se aplicam a processos individuais, não ao usuário como um todo. Por exemplo, definir a variável `openfiles` como `50` significa que cada processo que o usuário executa pode abrir até `50` arquivos. A quantidade total de arquivos que um usuário pode abrir é o valor de `openfiles` multiplicado pelo valor de `maxproc`. Isso também se aplica ao consumo de memória.

Para mais informações sobre limites de recursos e classes de login e capacidades em geral, consulte cap_mkdb(1), getrlimit(2) e login.conf(5).

13.13.2. Ativando e Configurando Limites de Recursos

A variável configurável `kern.racct.enable` deve ser configurada para um valor diferente de zero. Kernels personalizados requerem configuração específica:

```
options         RACCT
options         RCTL
```

Depois que o sistema for reinicializado no novo kernel, o `rctl` poderá ser usado para definir regras para o sistema.

A sintaxe da regra é controlada por meio do uso de um subject, subject-id, resource e action, conforme visto nesta regra de exemplo:

```
user:trhodes:maxproc:deny=10/user
```

Nesta regra, o subject é `user`, o subject-id é `trhodes`, o resource, `maxproc`, é o número máximo de processos, e a action é `deny`, que bloqueia a criação de novos processos. Isso significa que o usuário, `trhodes`, será restrito a execução de no máximo `10` processos. Outras ações possíveis incluem o registro no console, passando uma notificação para o devd(8) ou enviando um sigterm para o

processo.

Algum cuidado deve ser tomado ao adicionar regras. Como esse usuário está restrito a `10` processos, este exemplo impedirá que o usuário execute outras tarefas depois de efetuar login e executar uma sessão `screen`. Quando um limite de recurso for atingido, um erro será impresso, como neste exemplo:

```
% man test
    /usr/bin/man: Cannot fork: Resource temporarily unavailable
eval: Cannot fork: Resource temporarily unavailable
```

Como outro exemplo, uma jail pode ser impedida de exceder um limite de memória. Esta regra pode ser escrita como:

```
# rctl -a jail:httpd:memoryuse:deny=2G/jail
```

As regras persistirão durante as reinicializações se tiverem sido adicionadas ao arquivo /etc/rctl.conf. O formato é uma regra, sem o comando anterior. Por exemplo, a regra anterior pode ser adicionada como:

```
# Block jail from using more than 2G memory:
jail:httpd:memoryuse:deny=2G/jail
```

Para remover uma regra, use o `rctl` para removê-la da lista:

```
# rctl -r user:trhodes:maxproc:deny=10/user
```

Um método para remover todas as regras é documentado em rctl(8). No entanto, se for necessário remover todas as regras para um único usuário, esse comando poderá ser emitido:

```
# rctl -r user:trhodes
```

Existem muitos outros recursos que podem ser usados para exercer controle adicional sobre vários `subjects`. Veja rctl(8) para aprender sobre eles.

13.14. Administração Compartilhada com Sudo

Os administradores do sistema geralmente precisam conceder permissões avançadas aos usuários para que eles possam executar tarefas privilegiadas. A ideia de que os membros da equipe tenham acesso a um sistema FreeBSD para executar suas tarefas específicas abre desafios únicos para cada administrador. Esses membros da equipe precisam apenas de um subconjunto de acesso além dos níveis normais de usuário final; no entanto, eles quase sempre dizem ao gerenciadores que eles são incapazes de executar suas tarefas sem acesso de superusuário. Felizmente, não há motivo para

fornecer tal acesso aos usuários finais porque existem ferramentas para gerenciar esse exato requisito.

Até este ponto, o capítulo de segurança cobriu o acesso a usuários autorizados e a tentativa de impedir o acesso não autorizado. Outro problema surge quando os usuários autorizados têm acesso aos recursos do sistema. Em muitos casos, alguns usuários podem precisar acessar os scripts de inicialização do aplicativo ou uma equipe de administradores precisa manter o sistema. Tradicionalmente, os usuários e grupos padrão, as permissões de arquivo e até mesmo o comando su() gerenciariam esse acesso. E como os aplicativos exigiam mais acesso, à medida que mais usuários precisavam usar recursos do sistema, era necessária uma solução melhor. A aplicação mais usada atualmente é o Sudo.

O Sudo permite que os administradores configurem um acesso mais rígido aos comandos do sistema e forneçam alguns recursos avançados de log. Como uma ferramenta, ele está disponível na coleção de ports como security/sudo ou usando o utilitário pkg(8). Para usar a ferramenta pkg(8):

```
# pkg install sudo
```

Após a conclusão da instalação, o `visudo` instalado abrirá o arquivo de configuração com um editor de texto. O uso do `visudo` é altamente recomendado, pois vem com um verificador de sintaxe incorporado para verificar se não há erros antes que o arquivo seja salvo.

O arquivo de configuração é composto de várias seções pequenas que permitem uma configuração extensiva. No exemplo a seguir, o mantenedor do aplicativo da web, user1, precisa iniciar, parar e reiniciar o aplicativo da web conhecido como *webservice*. Para conceder a este usuário permissão para executar estas tarefas, adicione esta linha ao final do arquivo /usr/local/etc/sudoers:

```
user1   ALL=(ALL)       /usr/sbin/service webservice *
```

O usuário pode agora iniciar o *webservice* usando este comando:

```
% sudo /usr/sbin/service webservice start
```

Embora essa configuração permita que um único usuário acesse o serviço webservice; No entanto, na maioria das organizações, existe uma equipe inteira da Web encarregada de gerenciar o serviço. Uma única linha também pode dar acesso a um grupo inteiro. Essas etapas criarão um grupo da Web, adicionarão um usuário a esse grupo e permitirão que todos os membros do grupo gerenciem o serviço:

```
# pw groupadd -g 6001 -n webteam
```

Usando o mesmo comando pw(8), o usuário é adicionado ao grupo webteam:

```
# pw groupmod -m user1 -n webteam
```

Finalmente, esta linha no arquivo /usr/local/etc/sudoers permite que qualquer membro do grupo webteam gerencie o *webservice*:

```
%webteam   ALL=(ALL)       /usr/sbin/service webservice *
```

Ao contrário do su(1), o Sudo requer apenas a senha do usuário final. Isso adiciona uma vantagem em que os usuários não precisarão de senhas compartilhadas, uma descoberta na maioria das auditorias de segurança e o que por si só já ruins em todos os aspectos.

Os usuários autorizados a executar aplicativos com o Sudo só inserem suas próprias senhas. Isso é mais seguro e oferece melhor controle do que o su(1), onde a senha de `root` é inserida e o usuário adquire todas as permissões de `root`.



A maioria das organizações está se movendo ou migrou para um modelo de autenticação de dois fatores. Nestes casos, o usuário pode não ter uma senha para entrar. O Sudo resolve estes casos com a variável `NOPASSWD`. Adicioná-lo à configuração acima permitirá que todos os membros do grupo *webteam* gerenciem o serviço sem o requisito de senha:

```
%webteam   ALL=(ALL)       NOPASSWD: /usr/sbin/service webservice *
```

13.14.1. Logando a Saída

Uma vantagem para implementar o Sudo é a capacidade de ativar o log de sessão. Usando os mecanismos de log integrados e o comando sudoreplay incluído, todos os comandos iniciados por meio de Sudo são registrados para verificação posterior. Para ativar esse recurso, adicione uma entrada de diretório de log padrão, este exemplo usa uma variável de usuário. Existem várias outras convenções de nome de arquivo de log, consulte a página de manual do sudoreplay para obter informações adicionais.

```
Defaults iolog_dir=/var/log/sudo-io/%{user}
```



Este diretório será criado automaticamente após o logging ser configurado. É melhor deixar o sistema criar o diretório com permissões padrão apenas para estar seguro. Além disso, essa entrada também registra os administradores que usam o comando sudoreplay. Para alterar esse comportamento, leia e descomente as opções de log dentro do arquivo sudoers.

Uma vez que esta diretiva tenha sido adicionada ao arquivo sudoers, qualquer configuração de usuário pode ser atualizada com a solicitação para acessar o log. No exemplo mostrado, a entrada *webteam* atualizada teria as seguintes alterações adicionais:

```
%webteam ALL=(ALL) NOPASSWD: LOG_INPUT: LOG_OUTPUT: /usr/sbin/service webservice *
```

Deste ponto em diante, todos os membros do grupo *webteam* que alteram o status do aplicativo *webservice* serão registrados. A lista de sessões anteriores e atuais pode ser exibida com:

```
# sudoreplay -l
```

Na saída, para reproduzir uma sessão específica, procure a entrada `TSID=` e passe-a para o sudoreplay sem outras opções para reproduzir a sessão na velocidade normal. Por exemplo:

```
# sudoreplay user1/00/00/02
```



Enquanto as sessões são registradas, qualquer administrador pode remover as sessões e deixar apenas uma questão de por que elas fizeram isso. Vale a pena adicionar uma verificação diária por meio de um sistema de detecção de intrusão (IDS) ou software semelhante para que outros administradores sejam alertados sobre alterações manuais.

O `sudoreplay` é extremamente extensível. Consulte a documentação para mais informações.

Capítulo 14. Jails

14.1. Sinopse

Como a administração de sistemas é uma tarefa difícil, muitas ferramentas foram desenvolvidas para facilitar a vida do administrador. Essas ferramentas geralmente aprimoram a maneira como os sistemas são instalados, configurados e mantidos. Uma das ferramentas que podem ser usadas para melhorar a segurança de um sistema FreeBSD é *jails*. Jails estão disponíveis desde o FreeBSD 4.X e continuam sendo aprimoradas em sua utilidade, desempenho, confiabilidade e segurança.

Jails são construídas em cima do conceito de chroot(2), que é usado para mudar o diretório raiz de um conjunto de processos. Isso cria um ambiente seguro, separado do resto do sistema. Os processos criados no ambiente chroot não podem acessar arquivos ou recursos fora dele. E por esse motivo, comprometer um serviço em execução em um ambiente chroot não deve permitir que o invasor comprometa todo o sistema. No entanto, um chroot tem várias limitações. É adequado para tarefas fáceis que não exigem muita flexibilidade ou recursos complexos e avançados. Ao longo do tempo, foram descobertas muitas maneiras de escapar de um ambiente chroot, tornando essa solução como não sendo a melhor para proteger os serviços.

As jails aprimoram o conceito do ambiente chroot tradicional de várias maneiras. Em um ambiente chroot tradicional, os processos são limitados apenas na parte do sistema de arquivos que eles podem acessar. O restante dos recursos do sistema, os usuários, os processos em execução e o subsistema de rede são compartilhados pelos processos chroot e pelos processos do sistema host. As jails expandem esse modelo virtualizando o acesso ao sistema de arquivos, ao conjunto de usuários e ao subsistema de rede. Controles mais refinados estão disponíveis para ajustar o acesso de um ambiente em jail. As jails podem ser consideradas como um tipo de virtualização no nível do sistema operacional.

Uma jail é caracterizada por quatro elementos:

- Uma subárvore de diretórios: o ponto de partida a partir do qual uma jail é inserida. Uma vez dentro da jail, não é permitido que um processo escape fora dessa subárvore.
- Um nome de host: que será usado pela jail.
- Um endereço IP: atribuído à jail. O endereço IP de uma jail é geralmente um endereço de alias de uma interface de rede existente.
- Um comando: o nome do caminho de um executável para ser executado dentro da jail. O caminho é relativo ao diretório raiz do ambiente da jail.

As Jails possuem seu próprio conjunto de usuários e sua própria conta de `root` que são limitados ao ambiente da jail. A conta `root` de uma jail não tem permissão para executar operações no sistema fora do ambiente da jail associada.

Este capítulo fornece uma visão geral da terminologia e dos comandos para gerenciar as jail do FreeBSD. As jails são uma ferramenta poderosa para administradores de sistemas e usuários avançados.

Depois de ler este capítulo, você saberá:

- O que é uma jail e qual finalidade ela pode servir nas instalações do FreeBSD.
- Como compilar, iniciar e parar uma jail.
- Os fundamentos da administração de jails, tanto de dentro como fora da jail.



As jails são uma ferramenta poderosa, mas não são uma panaceia de segurança. Embora não seja possível que um processo rodando em jail burle a segurança por conta própria, existem várias maneiras pelas quais um usuário não privilegiado fora da jail pode cooperar com um usuário privilegiado dentro da jail para obter privilégios elevados no ambiente host.

A maioria desses ataques podem ser mitigados apenas garantindo que o root da jail não seja acessível a usuários não privilegiados no ambiente host. Como regra geral, usuários não confiáveis com acesso privilegiado a uma jail não devem ter acesso ao ambiente do host.

14.2. Termos Relacionados à Jails

Para facilitar a compreensão de partes do sistema FreeBSD relacionadas a jails, seus componentes internos e a maneira como eles interagem com o resto do FreeBSD, os seguintes termos são usados mais adiante neste capítulo:

chroot(8) (comando)

Utilitário, que usa a chamada de sistema chroot(2) do FreeBSD para alterar o diretório raiz de um processo e todos os seus descendentes.

chroot(2) (ambiente)

O ambiente dos processos em execução em um "chroot". Isso inclui recursos como a parte do sistema de arquivos que é visível, IDs de usuário e grupo que estão disponíveis, interfaces de rede e outros mecanismos de IPC, etc.

jail(8) (comando)

O utilitário de administração do sistema que permite o lançamento de processos dentro de um ambiente jail.

host (sistema, processo, usuário, etc.)

O sistema de controle de um ambiente jail. O sistema host tem acesso a todos os recursos de hardware disponíveis e pode controlar processos fora e dentro de um ambiente jail. Uma das diferenças importantes do sistema host de uma jail é que as limitações que se aplicam aos processos de super-usuário dentro de uma jail não são aplicadas aos processos do sistema host.

hosted (sistema, processo, usuário, etc.)

Um processo, usuário ou outra entidade, cujo acesso a recursos é restrito por uma jail do FreeBSD.

14.3. Criando e Controlando Jails

Alguns administradores dividem as jails nos dois seguintes tipos: jails "completa", que se assemelham a um sistema real do FreeBSD, e jails de "serviço", dedicados a um aplicativo ou serviço, possivelmente executando com privilégios. Esta é apenas uma divisão conceitual e o processo de criação de uma jail não é afetado por ela. Ao criar uma jail "completa", há duas opções para a origem do userland: usar binários pré-compilados (como aqueles fornecidos em uma mídia de instalação) ou compila-los a partir do código fonte.

14.3.1. Instalando uma Jail

14.3.1.1. Para instalar uma Jail pela Internet

A ferramenta bsdinstall(8) pode ser usada para baixar e instalar os binários necessários para uma Jail. Será apresentando a seleção de um mirror, quais distribuições serão instaladas no diretório de destino e algumas configurações básicas da Jail:

```
# bsdinstall jail /here/is/the/jail
```

Uma vez que o comando finalize, o próximo passo é configurar o host para rodar a jail.

14.3.1.2. Instalar uma Jail por uma imagem ISO

Para instalar o userland da mídia de instalação, primeiro crie o diretório raiz da jail. Isso pode ser feito definindo a variável `DESTDIR` para o local adequado.

Inicie um shell e defina a variável `DESTDIR`:

```
# sh
# export DESTDIR=/here/is/the/jail
```

Monte a mídia de instalação como abordado em mdconfig(8) ao usar a ISO de instalação:

```
# mount -t cd9660 /dev/`mdconfig -f cdimage.iso` /mnt
# cd /mnt/usr/freebsd-dist/
```

Extraia os binários dos tarballs na mídia de instalação dentro do destino declarado. Minimamente, apenas o conjunto base precisa ser extraído, mas uma instalação completa pode ser executada quando preferida.

Para instalar apenas o sistema básico:

```
# tar -xf /mnt/usr/freebsd-dist/base.txz -C $DESTDIR
```

Para instalar tudo, exceto o kernel:

```
# for set in base ports; do tar -xf /mnt/usr/freebsd-dist/$set.txz -C $DESTDIR ; done
```

14.3.1.3. Para compilar e instalar uma Jail a partir do código fonte

A página de manual jail(8) explica o procedimento para compilar uma jail:

```
# setenv D /here/is/the/jail
# mkdir -p $D      ①
# cd /usr/src
# make buildworld  ②
# make installworld DESTDIR=$D  ③
# make distribution DESTDIR=$D  ④
# mount -t devfs devfs $D/dev   ⑤
```

① Selecionar um local para uma jail é o melhor ponto de partida. É aqui que a jail residirá fisicamente no sistema de arquivos do host da jail. Uma boa opção pode ser /usr/jail/jailname, onde *jailname* é o nome do host que identifica a jail. Normalmente, /usr/ tem espaço suficiente para o sistema de arquivos da jail, onde para jails "completa" é, essencialmente, uma replicação de todos os arquivos presentes em uma instalação padrão do sistema básico do FreeBSD.

② Se você já tiver recompilado seu userland usando `make world` ou `make buildworld`, você pode pular esta etapa e instalar seu userland existente na nova jail.

③ Esse comando preencherá a sub-árvore de diretórios escolhida como o local físico da jail no sistema de arquivos com os binários, bibliotecas, páginas de manual e assim por diante.

④ O target `distribuição` do make instala todos os arquivos de configuração necessários. Em palavras simples, ele instala cada arquivo instalável de /usr/src/etc/ no diretório /etc do ambiente jail: $D/etc/.

⑤ A montagem do sistema de arquivos devfs(8) dentro de uma jail não é necessária. Por outro lado, qualquer, ou quase qualquer aplicativo requer acesso a pelo menos um dispositivo, dependendo da finalidade do aplicativo fornecido. É muito importante controlar o acesso a dispositivos de dentro de uma jail, pois configurações inadequadas podem permitir que um invasor faça coisas desagradáveis na jail. O controle sobre devfs(8) é gerenciado por meio de conjuntos de regras que são descritos nas páginas de manual devfs(8) e devfs.conf(5).

14.3.2. Configurando o Host

Uma vez que a jail é instalada, ela pode ser iniciada usando o utilitário jail(8). O utilitário jail(8) possui quatro argumentos obrigatórios que são descritos em Sinopse. Outros argumentos podem ser especificados também, por exemplo, para executar o processo em jail com as credenciais de um usuário específico. O argumento de `command` depende do tipo de jail; para um *sistema virtual*, /etc/rc é uma boa escolha, já que ele irá replicar a sequência de inicialização de um sistema real do FreeBSD. Para uma jail de *serviço*, depende do serviço ou aplicativo que será executado dentro da jail.

As jails geralmente são iniciadas no boot e o mecanismo rc do FreeBSD fornece uma maneira fácil de fazer isso.

1. Configure os parâmetros da jail no arquivo jail.conf:

```
www {
    host.hostname = www.example.org;          # Hostname
    ip4.addr = 192.168.0.10;                  # IP address of the jail
    path ="/usr/jail/www";                    # Path to the jail
    devfs_ruleset = "www_ruleset";            # devfs ruleset
    mount.devfs;                              # Mount devfs inside the jail
    exec.start = "/bin/sh /etc/rc";           # Start command
    exec.stop = "/bin/sh /etc/rc.shutdown";   # Stop command
}
```

Configure as jails para iniciar no boot no arquivo rc.conf:

```
jail_enable="YES"   # Set to NO to disable starting of any jails
```

A inicialização padrão das jails configuradas no arquivo jail.conf(5), executará o script /etc/rc da jail, que assume que a jail é um sistema virtual completo. Para jails de serviço, o comando de inicialização padrão da jail deve ser alterado, definindo a opção `exec.start` apropriadamente.



Para obter uma lista completa das opções disponíveis, consulte a página de manual jail.conf(5).

service(8) pode ser usado para iniciar ou parar uma jail manualmente, se uma entrada para ela existir no arquivo jail.conf:

```
# service jail start www
# service jail stop www
```

As jails podem ser desligadas com o jexec(8). Use jls(8) para identificar o `JID` da jail e, em seguida, use jexec(8) para executar o script de desligamento nessa jail.

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.0.10    www                           /usr/jail/www
# jexec 3 /etc/rc.shutdown
```

Mais informações sobre isso podem ser encontradas na página de manual jail(8).

14.4. Tuning e Administração

Existem várias opções que podem ser configuradas para qualquer jail, e várias maneiras de

combinar um sistema host FreeBSD com jails, para produzir aplicações de alto nível. Esta seção apresenta:

- Algumas das opções disponíveis para ajustar as restrições de comportamento e segurança implementadas pela instalação de uma jail.
- Alguns das aplicações de alto nível para gerenciamento de jail, que estão disponíveis através da Coleção de Ports do FreeBSD, e que podem ser usados para implementar soluções globais baseadas em jails.

14.4.1. Ferramentas de Sistema para Tuning de Jail no FreeBSD

O tuning da configuração de uma jail é feito principalmente configurando variáveis sysctl(8). Uma sub-árvore especial do sysctl existe como base para organizar todas as opções relevantes: a hierarquia `security.jail.*` das opções do kernel do FreeBSD. Aqui está uma lista dos principais sysctls relacionados à jail, completas com seu valor padrão. Os nomes devem ser autoexplicativos, mas para obter mais informações sobre eles, consulte as páginas de manual jail(8) e sysctl(8).

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 0`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

Estas variáveis podem ser usadas pelo administrador de sistemas do *sistema host* para adicionar ou remover algumas das limitações impostas por padrão no usuário `root`. Note que existem algumas limitações que não podem ser removidas. O usuário `root` não tem permissão para montar ou desmontar sistemas de arquivos de dentro de uma jail(8). O `root` dentro de uma jail não pode carregar ou descarregar conjuntos de regras devfs(8), definir regras de firewall, ou fazer muitas outras tarefas administrativas que requerem modificações de dados no kernel, como a configuração do `securelevel` do kernel.

O sistema base do FreeBSD contém um conjunto básico de ferramentas para visualizar informações sobre as jails ativas e para se conectar a uma jail para executar comandos administrativos. Os comandos jls(8) e jexec(8) são parte do sistema base do FreeBSD, e podem ser usados para realizar as seguintes tarefas simples:

- Apresentar uma lista de jails ativas e seu identificador de jail correspondente (JID), endereço IP, hostname e path.
- Se conectar a uma jail em execução, a partir de seu sistema host, e executar um comando dentro da jaijl ou executar tarefas administrativas dentro da própria jail. Isso é especialmente útil quando o usuário `root` deseja desligar de maneira limpa uma jail. O utilitário jexec(8) também pode ser usado para iniciar um shell em uma jail para administração; por exemplo:

```
# jexec 1 tcsh
```

14.4.2. Ferramentas Administrativas de Alto Nível na Coleção de Ports do FreeBSD

Entre os muitos utilitários de terceiros para administração de jail, um dos mais completos e úteis é o sysutils/ezjail. É um conjunto de scripts que contribuem para o gerenciamento de jail(8). Consulte a seção ezjail do handbook para mais informações.

14.4.3. Mantendo as Jails com Alterações e Atualizadas

As jails devem ser atualizadas a partir do sistema operacional do host, pois a tentativa de aplicar patchs no userland de dentro da jail pode falhar, já que o comportamento padrão no FreeBSD é não permitir o uso de chflags(1) em uma jail, o que impede a substituição de alguns arquivos. É possível alterar esse comportamento, mas é recomendado usar o freebsd-update(8) para atualizar as jails. Use `-b` para especificar o caminho da jail a ser atualizada.

Para atualizar a Jail para o último release patch da versão do FreeBSD que já está em execução, execute os seguintes comandos no host:

```
# freebsd-update -b /here/is/the/jail fetch
# freebsd-update -b /here/is/the/jail install
```

Para atualizar a jail para uma versão maior ou menor, primeiro atualize o sistema hospedeiro com descrito em Realizando Upgrades de Versão Principais e Menores. Uma vez que o hospedeiro esteja atualizado e reiniciado, a jail pode então ser atualizada. Por exemplo, para atualizar de 12.0-RELEASE para 12.1-RELEASE, rode no hospedeiro:

```
# freebsd-update -b /here/is/the/jail --currently-running 12.0-RELEASE -r 12.1-RELEASE
upgrade
# freebsd-update -b /here/is/the/jail install
# service jail restart myjail
# freebsd-update -b /here/is/the/jail install
```

Então, se foi uma atualização de versão principal, reinstale todos os pacotes instalados e reinicie a jail novamente. Isso é necessário porque a versão ABI muda ao atualizar entre as versões principais do FreeBSD. Pelo host:

```
# pkg -j myjail upgrade -f
# service jail restart myjail
```

14.5. Atualizando Múltiplas Jails

O gerenciamento de várias jails pode se tornar problemático porque toda jail tem que ser

recompilada a partir do zero sempre que for atualizada. Isso pode ser demorado e entediante se muitas jails forem criadas e atualizadas manualmente.

Esta seção demonstra um método para resolver esse problema compartilhando com segurança o máximo possível entre jails usando montagens somente leitura mount_nullfs(8), para que a atualização seja mais simples. Isso torna mais atraente colocar serviços únicos, como HTTP, DNS e SMTP, em jails individuais. Além disso, fornece uma maneira simples de adicionar, remover e atualizar jails.



Existem soluções mais simples, como o ezjail, que fornece um método mais fácil de administrar as jails do FreeBSD, mas é menos versátil que essa configuração. O ezjail é coberto com mais detalhes em Gerenciando Jails com o ezjail.

Os objetivos da configuração descrita nesta seção são:

- Criar uma estrutura de jail simples e fácil de entender que não exija a execução de um installworld completo em todas as jails.
- Facilitar a adição de novas jails ou remoção das já existentes.
- Facilitar a atualização ou upgrade de jails existentes.
- Tornar possível a utilização de uma branch customizada do FreeBSD.
- Seja paranoico com a segurança, reduzindo ao máximo a possibilidade de comprometimento.
- Economize espaço e inodes, tanto quanto possível.

Esse design depende de um template master único, read-only, que é montado em cada jail e em um dispositivo read-write por jail. Um dispositivo pode ser um disco físico separado, uma partição ou um dispositivo de memória com suporte a vnode. Este exemplo usa montagens nullfs read-write.

O layout do sistema de arquivos é o seguinte:

- As jails são hospedadas na partição /home.
- Cada jail será montada no diretório /home/j.
- O template para cada jail e a partição read-only para todos as jails é /home/j/mroot.
- Um diretório em branco será criado para cada jail no diretório /home/j.
- Cada jail terá um diretório /s que será vinculado à parte de read-write do sistema.
- Cada jail terá seu próprio sistema de read-write baseado em /home/j/skel.
- A parte de read-write de cada jail será criada em /home/js.

14.5.1. Criando o Template

Esta seção descreve as etapas necessárias para criar o template master.

É recomendado primeiramente atualizar o sistema host FreeBSD para a branch -RELEASE mais recente usando as instruções em Atualizando o FreeBSD a partir do código fonte. Adicionalmente, este template usa o pacote ou port sysutils/cpdup e o portsnap será utilizado para baixar a Coleção de Ports do FreeBSD.

1. Primeiro, crie uma estrutura de diretório para o sistema de arquivo read-only que conterá os binários do FreeBSD para as jails. Em seguida, altere para o diretório de código-fonte do FreeBSD e instale o sistema de arquivos read-only no template das jails:

```
# mkdir /home/j /home/j/mroot
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot
```

2. Em seguida, prepare uma Coleção de Ports do FreeBSD para as jails, assim como uma árvore de código fonte do FreeBSD, que são necessários para o mergemaster:

```
# cd /home/j/mroot
# mkdir usr/ports
# portsnap -p /home/j/mroot/usr/ports fetch extract
# cpdup /usr/src /home/j/mroot/usr/src
```

3. Crie um esqueleto para a parte de read-write do sistema:

```
# mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr-X11R6
/home/j/skel/distfiles
# mv etc /home/j/skel
# mv usr/local /home/j/skel/usr-local
# mv tmp /home/j/skel
# mv var /home/j/skel
# mv root /home/j/skel
```

4. Use o mergemaster para instalar os arquivos de configuração ausentes. Em seguida, remova os diretórios extras criados pelo mergemaster:

```
# mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
# cd /home/j/skel
# rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
```

5. Agora, faça os links dos sistema de arquivos read-write ao sistema de arquivos read-only. Certifique-se de que os links simbólicos sejam criados nos locais corretos de s/, pois a criação de diretórios nos locais errados fará com que a instalação falhe.

```
# cd /home/j/mroot
# mkdir s
# ln -s s/etc etc
# ln -s s/home home
# ln -s s/root root
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
```

```
# ln -s ../../s/distfiles usr/ports/distfiles
# ln -s s/tmp tmp
# ln -s s/var var
```

6. Como último passo, crie um arquivo /home/j/skel/etc/make.conf genérico contendo esta linha:

```
WRKDIRPREFIX?=  /s/portbuild
```

   Isto torna possível compilar ports do FreeBSD dentro de cada jail. Lembre-se de que o diretório do ports faz parte do sistema somente leitura. O caminho customizado para o `WRKDIRPREFIX` permite que compilações sejam feitas na parte read-write de cada jail.

14.5.2. Criando Jails

O template jail agora pode ser usado para preparar e configurar as jails no arquivo /etc/rc.conf. Este exemplo demonstra a criação de 3 jails: `NS`, `MAIL` e `WWW`.

1. Adicione as seguintes linhas ao arquivo /etc/fstab, para que o template read-only e o espaço read-write das jails estejam disponível nas respectivas jails:

```
/home/j/mroot   /home/j/ns     nullfs  ro  0   0
/home/j/mroot   /home/j/mail   nullfs  ro  0   0
/home/j/mroot   /home/j/www    nullfs  ro  0   0
/home/js/ns     /home/j/ns/s   nullfs  rw  0   0
/home/js/mail   /home/j/mail/s nullfs  rw  0   0
/home/js/www    /home/j/www/s  nullfs  rw  0   0
```

   Para evitar que o fsck verifique as montagens nullfs durante a inicialização e o dump faça backup das montagens nullfs read-only das jails, as duas últimas colunas são ambos definidos para `0`.

2. Configure as jails no arquivo /etc/rc.conf:

```
jail_enable="YES"
jail_set_hostname_allow="NO"
jail_list="ns mail www"
jail_ns_hostname="ns.example.org"
jail_ns_ip="192.168.3.17"
jail_ns_rootdir="/usr/home/j/ns"
jail_ns_devfs_enable="YES"
jail_mail_hostname="mail.example.org"
jail_mail_ip="192.168.3.18"
jail_mail_rootdir="/usr/home/j/mail"
jail_mail_devfs_enable="YES"
```

```
jail_www_hostname="www.example.org"
jail_www_ip="62.123.43.14"
jail_www_rootdir="/usr/home/j/www"
jail_www_devfs_enable="YES"
```

A variável `jail_name_rootdir` é configurada como /usr/home em vez de /home porque o caminho físico de /home em uma instalação padrão do FreeBSD é /usr/home. A variável `jail_name_rootdir` *não* deve ser configurada para um caminho que inclua um link simbólico, caso contrário as jails não serão iniciadas.

3. Crie os pontos de montagem necessários para o sistema de arquivos read-only de cada jail:

```
# mkdir /home/j/ns /home/j/mail /home/j/www
```

4. Instale o template read-write em cada jail usando sysutils/cpdup:

```
# mkdir /home/js
# cpdup /home/j/skel /home/js/ns
# cpdup /home/j/skel /home/js/mail
# cpdup /home/j/skel /home/js/www
```

5. Nesta fase, as jails estão compiladas e preparadas para execução. Primeiro, monte os sistemas de arquivos necessários para cada jail e, em seguida, inicie-as:

```
# mount -a
# service jail start
```

As jails devem estar funcionando agora. Para verificar se eles foram iniciadas corretamente, use `jls`. Sua saída deve ser semelhante ao seguinte:

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.3.17    ns.example.org                /home/j/ns
     2  192.168.3.18    mail.example.org              /home/j/mail
     1  62.123.43.14    www.example.org               /home/j/www
```

Neste ponto, deve ser possível entrar em cada jail, adicionar novos usuários ou configurar daemons. A coluna `JID` indica o número de identificação da jail de cada jail em execução. Use o seguinte comando para executar tarefas administrativas na jail cujo JID é `3`:

```
# jexec 3 tcsh
```

14.5.3. Fazendo Upgrade

O design dessa configuração fornece uma maneira fácil de atualizar as jails existentes, minimizando o tempo de downtime. Além disso, fornece uma maneira de reverter para a versão mais antiga, caso ocorra algum problema.

1. O primeiro passo é atualizar o sistema host. Em seguida, crie um novo template temporário read-only em /home/j/mroot2.

   ```
   # mkdir /home/j/mroot2
   # cd /usr/src
   # make installworld DESTDIR=/home/j/mroot2
   # cd /home/j/mroot2
   # cpdup /usr/src usr/src
   # mkdir s
   ```

   O `installworld` cria alguns diretórios desnecessários, que devem ser removidos:

   ```
   # chflags -R 0 var
   # rm -R etc var root usr/local tmp
   ```

2. Recrie os links simbólicos read-write para o sistema de arquivos master:

   ```
   # ln -s s/etc etc
   # ln -s s/root root
   # ln -s s/home home
   # ln -s ../s/usr-local usr/local
   # ln -s ../s/usr-X11R6 usr/X11R6
   # ln -s s/tmp tmp
   # ln -s s/var var
   ```

3. Em seguida, pare as jails:

   ```
   # service jail stop
   ```

4. Desmonte os sistemas de arquivos originais, pois os sistemas read-write estão conectados ao sistema read-only (/s):

   ```
   # umount /home/j/ns/s
   # umount /home/j/ns
   # umount /home/j/mail/s
   # umount /home/j/mail
   # umount /home/j/www/s
   # umount /home/j/www
   ```

5. Mova o antigo sistema de arquivos read-only e substitua-o pelo novo. Isso servirá como backup e arquivamento do antigo sistema de arquivos read-only se algo der errado. A convenção de nomenclatura usada aqui corresponde a quando um novo sistema de arquivos read-only foi criado. Mova a Coleção de Ports do FreeBSD original para o novo sistema de arquivos para economizar espaço e inodes:

```
# cd /home/j
# mv mroot mroot.20060601
# mv mroot2 mroot
# mv mroot.20060601/usr/ports mroot/usr
```

6. Neste ponto, o novo template read-only está pronto, então a única tarefa restante é remontar os sistemas de arquivos e iniciar as jails:

```
# mount -a
# service jail start
```

Use `jls` para verificar se as jails foram iniciadas corretamente. Execute `mergemaster` em cada jail para atualizar os arquivos de configuração.

14.6. Gerenciando Jails com o ezjail

Criar e gerenciar múltiplas jails pode se tornar um trabalho tedioso e propenso a erros. O ezjail de Dirk Engling automatiza e simplifica muito as tarefas de jails. Uma *basejail* é criada como um template. Jails adicionais usam mount_nullfs(8) para compartilhar muitos dos diretórios da basejail sem usar espaço em disco adicional. Cada jail adicional leva apenas alguns megabytes de espaço em disco antes que os aplicativos sejam instalados. A atualização da cópia do userland na basejail atualiza automaticamente todas as outras jails.

Benefícios e recursos adicionais são descritos em detalhes no site do ezjail, https://erdgeist.org/arts/software/ezjail/.

14.6.1. Instalando o ezjail

A instalação do ezjail consiste na inclusão de uma interface de loopback para uso nas jails, instalação do port ou pacote e ativação do serviço.

1. Para manter o tráfego de loopback da jail fora da interface de rede de loopback do host `lo0`, uma segunda interface de loopback é criada adicionando uma entrada no arquivo /etc/rc.conf:

```
cloned_interfaces="lo1"
```

   A segunda interface de loopback `lo1` será criada quando o sistema for iniciado. Também

pode ser criado manualmente sem reiniciar:

```
# service netif cloneup
Created clone interfaces: lo1.
```

Jails podem ter permissão para usar aliases dessa interface de loopback secundária sem interferir no host.

Dentro de uma jail, o acesso ao endereço de loopback `127.0.0.1` é redirecionado para o primeiro endereço de IP atribuído à jail. Para fazer com que o loopback da jail corresponda à nova interface `lo1`, essa interface deve ser especificada primeiro na lista de interfaces e endereços IP fornecidos ao criar uma nova jail.

Dê a cada jail um endereço de loopback exclusivo no bloco de rede `127.0.0.0/8`.

2. Instale o sysutils/ezjail:

```
# cd /usr/ports/sysutils/ezjail
# make install clean
```

3. Ative o ezjail adicionando esta linha ao arquivo /etc/rc.conf:

```
ezjail_enable="YES"
```

4. O serviço será iniciado automaticamente na inicialização do sistema. Ele pode ser iniciado imediatamente na sessão atual:

```
# service ezjail start
```

14.6.2. Configuração inicial

Com o ezjail instalado, a estrutura do diretório basejail pode ser criada e preenchida. Esta etapa é necessária apenas uma vez no computador host da jail.

Em ambos os exemplos, `-p` faz com que a árvore de ports seja baixada com o portsnap(8) para a basejail. Essa cópia única do diretório de ports será compartilhada por todas as jails. Usar uma cópia separada do diretório de ports para jails isola-os do host. O ezjail é explicado com mais detalhes no FAQ: http://erdgeist.org/arts/software/ezjail/#FAQ.

1. Preencher a Jail com o FreeBSD-RELEASE

   Para uma basejail baseada na mesma versão FreeBSD RELEASE do computador host, use o comando `install`. Por exemplo, em um computador host executando o FreeBSD 10-STABLE, a versão mais recente do FreeBSD -10 será instalada na jail:

```
# ezjail-admin install -p
```

2. Preencher a Jail com o comando `installworld`

   A basejail pode ser instalada a partir de binários criados pelo `buildworld` no host com `ezjail-admin update`.

   Neste exemplo, o FreeBSD 10-STABLE foi compilado a partir do código fonte. Os diretórios da jail são criados. E então `installworld` é executado, instalando o /usr/obj do host na basejail.

   ```
   # ezjail-admin update -i -p
   ```

   O /usr/src do host é usado por padrão. Um diretório de código fonte diferente no host pode ser especificado com `-s` e um caminho ou com `ezjail_sourcetree` em /usr/local/etc/ezjail.conf.



A árvore de ports da basejail é compartilhada por outras jails. No entanto, os distfiles baixados são armazenados na jail que os baixou. Por padrão, esses arquivos são armazenados em /var/ports/distfiles dentro de cada jail. /var/ports dentro de cada jail também é usado como um diretório de trabalho ao compilar ports.



O protocolo FTP é usado por padrão para baixar pacotes para a instalação da basejail. Configurações de firewall ou proxy podem impedir ou interferir nas transferências de FTP. O protocolo HTTP funciona de maneira diferente e evita esses problemas. Ele pode ser escolhido especificando uma URL completa para um espelho de download específico no arquivo /usr/local/etc/ezjail.conf:

```
ezjail_ftphost=http://ftp.FreeBSD.org
```

Veja Sites de FTP para uma lista de sites.

14.6.3. Criando e Iniciando uma Nova Jail

Novas jails são criadas com o comando `ezjail-admin create`. Nestes exemplos, a interface de loopback `lo1` é usada conforme descrito acima.

Procedure: Crie e Inicie uma Nova Jail

1. Crie a jail, especificando um nome e as interfaces de loopback e de rede a serem usadas, junto com seus endereços IP. Neste exemplo, a jail é denominada `dnsjail`.

```
ezjail-admin create dnsjail 'lo1|127.0.1.1,em0|192.168.1.50'
```



A maioria dos serviços de rede são executados em jails sem problemas. Alguns serviços de rede, como ping(8), usam *raw network sockets*. Nas jails, raw network sockets são desativados por padrão para segurança. Serviços que exigem eles não irão funcionar.

Ocasionalmente, uma jail pode realmente precisar de raw sockets. Por exemplo, os aplicativos de monitoramento de rede geralmente usam ping(8) para verificar a disponibilidade de outros computadores. Quando raw network sockets são realmente necessários em uma jail, eles podem ser ativados editando o arquivo de configuração do ezjail para uma jail individual, /usr/local/etc/ezjail/jailname. Modifique a entrada `parameters`:

```
export jail_jailname_parameters="allow.raw_sockets=1"
```

Não habilite raw network sockets, a menos que os serviços na jail realmente precisem deles.

2. Inicie a jail:

```
# ezjail-admin start dnsjail
```

3. Use um console na jail:

```
# ezjail-admin console dnsjail
```

A jail está funcionando e configurações adicionais podem ser realizadas. Configurações típicas adicionadas neste momento incluem:

1. Defina a Senha de `root`

   Conecte-se à jail e configure a senha do usuário `root`:

```
# ezjail-admin console dnsjail
# passwd
Changing local password for root
New Password:
Retype New Password:
```

2. Configuração de Fuso Horário

   O fuso horário da jail pode ser definido com tzsetup(8). Para evitar mensagens de erro

espúrias, a entrada adjkerntz(8) em /etc/crontab pode ser comentada ou removida. Este comando tenta atualizar o relógio de hardware do computador com alterações de fuso horário, mas as jails não têm permissão para acessar esse hardware.

3. Servidores DNS

   Insira as linhas com o servidor de nomes de domínio no arquivo /etc/resolv.conf para que o DNS funcione na jail.

4. Edite o arquivo /etc/hosts

   Altere o endereço e adicione o nome da jail para as entradas `localhost` no /etc/hosts.

5. Configure o arquivo /etc/rc.conf

   Digite as definições de configuração no arquivo /etc/rc.conf. Isso é muito parecido com a configuração de um computador completo. O nome do host e o endereço IP não estão definidos aqui. Esses valores já são fornecidos pela configuração da jail.

Com a jail configurada, os aplicativos para os quais a jail foi criada podem ser instalados.



Alguns ports devem ser compilados com opções especiais para serem usados em uma jail. Por exemplo, os dois pacotes de plugin de monitoramento de rede net-mgmt/nagios-plugins e net-mgmt/monitoring-plugins possuem uma opção `JAIL` que deve ser ativada para que funcionem corretamente dentro de uma jail.

14.6.4. Atualizando as Jails

14.6.4.1. Atualizando o Sistema Operacional

Como a cópia do userland da basejail é compartilhada pelas outras jails, a atualização da basejail atualiza automaticamente todas as outras jails. Atualizações binárias ou por código fonte podem ser usadas.

Para compilar o world a partir do código fonte no host, e depois instala-lo na basejail, use:

```
# ezjail-admin update -b
```

Se o world já estiver sido compilado no host, instale-o no basejail com:

```
# ezjail-admin update -i
```

Atualizações binárias usam o freebsd-update(8). Essas atualizações têm as mesmas limitações como se o freebsd-update(8) estivesse sendo executado diretamente. O mais importante é que apenas as versões -RELEASE do FreeBSD estão disponíveis com este método.

Atualize a basejail para a última versão de patchs da versão do FreeBSD no host. Por exemplo,

atualizando de RELEASE-p1 para RELEASE-p2.

```
# ezjail-admin update -u
```

Para atualizar a basejail para uma nova versão, primeiro atualize o sistema host como descrito em Realizando Upgrades de Versão Principais e Menores. Depois que o host tiver sido atualizado e reinicializado, a basejail poderá ser atualizada. O freebsd-update(8) não tem como determinar qual versão está atualmente instalada na basejail, então a versão original deve ser especificada. Use o file(1) para determinar a versão original na basejail:

```
# file /usr/jails/basejail/bin/sh
/usr/jails/basejail/bin/sh: ELF 64-bit LSB executable, x86-64, version 1 (FreeBSD),
dynamically linked (uses shared libs), for FreeBSD 9.3, stripped
```

Agora use essas informações para executar a atualização de `9.3-RELEASE` para a versão atual do sistema host:

```
# ezjail-admin update -U -s 9.3-RELEASE
```

Depois de atualizar a basejail, o mergemaster(8) deve ser executado para atualizar os arquivos de configuração de cada jail.

Como usar o mergemaster(8) depende do propósito e da confiabilidade de uma jail. Se os serviços ou usuários de uma jail não são confiáveis, então o mergemaster(8) deve ser executado somente dentro dessa jail:

Exemplo 33. mergemaster(8) em Jail Não Confiável

Exclua o link do /usr/src da jail para a basejail e crie um novo /usr/src na jail como um ponto de montagem. Monte o /usr/src do computador host como read-only no novo ponto de montagem /usr/src da jail:

```
# rm /usr/jails/jailname/usr/src
# mkdir /usr/jails/jailname/usr/src
# mount -t nullfs -o ro /usr/src /usr/jails/jailname/usr/src
```

Execute um console na jail:

```
# ezjail-admin console jailname
```

Dentro da jail, execute `mergemaster`. Em seguida, saia do console da jail:

```
# cd /usr/src
# mergemaster -U
```

```
# exit
```

Finalmente, desmonte o /usr/src da jail:

```
# umount /usr/jails/jailname/usr/src
```

Exemplo 34. mergemaster(8) em Jail Confiável

Se os usuários e serviços em uma jail forem confiáveis, o mergemaster(8) pode ser executado a partir do host:

```
# mergemaster -U -D /usr/jails/jailname
```



Após uma atualização de versão principal, é recomendado pelo sysutils/ezjail garantir que o `pkg` seja da versão correta. Portanto, digite:

```
# pkg-static upgrade -f pkg
```

para atualizar ou fazer o downgrade para a versão apropriada.

14.6.4.2. Atualizando o Ports

A árvore de ports na basejail é compartilhada pelas outras jails. A atualização dessa cópia da árvore de ports fornece às outras jails a versão atualizada também.

A árvore de ports da basejail é atualizada com o portsnap(8):

```
# ezjail-admin update -P
```

14.6.5. Controlando as Jails

14.6.5.1. Parando e Iniciando Jails

O ezjail inicia automaticamente as jails quando o computador é iniciado. As jails podem ser manualmente paradas e reiniciadas com `stop` e `start`:

```
# ezjail-admin stop sambajail
Stopping jails: sambajail.
```

Por padrão, as jails são iniciadas automaticamente quando o computador host é iniciado. A inicialização automática pode ser desativada com `config`:

```
# ezjail-admin config -r norun seldomjail
```

Isso entrará em vigor na próxima vez em que o computador host for iniciado. Uma jail que já está em execução não será interrompida.

A ativação do início automático é muito semelhante:

```
# ezjail-admin config -r run oftenjail
```

14.6.5.2. Arquivando e Restaurando Jails

Use `archive` para criar um arquivo .tar.gz de uma jail. O nome do arquivo é composto pelo nome da jail e pela data atual. Os archives são gravados no diretório de archive, /usr/jails/ezjail_archives. Um diretório de archive diferente pode ser escolhido configurando `ezjail_archivedir` no arquivo de configuração.

O archive pode ser copiado em outro lugar como um backup, ou uma jail existente pode ser restaurada a partir dele com o `restore`. Uma nova jail pode ser criada a partir de um archive, fornecendo uma maneira conveniente de clonar as jails existentes.

Pare e arquive uma jail chamada `wwwserver`:

```
# ezjail-admin stop wwwserver
Stopping jails: wwwserver.
# ezjail-admin archive wwwserver
# ls /usr/jails/ezjail-archives/
wwwserver-201407271153.13.tar.gz
```

Crie uma nova jail chamada `wwwserver-clone` do archive criado na etapa anterior. Use a interface em1 e atribua um novo endereço IP para evitar conflito com a original:

```
# ezjail-admin create -a /usr/jails/ezjail_archives/wwwserver-201407271153.13.tar.gz
wwwserver-clone 'lo1|127.0.3.1,em1|192.168.1.51'
```

14.6.6. Exemplo Completo: BIND em uma Jail

Colocar o servidor DNSBIND em uma jail melhora a segurança ao isolá-lo. Este exemplo cria um servidor de nomes de cache simples.

- A jail será chamada de `dns1`.
- A jail usará o endereço IP `192.168.1.240` na interface `re0` do host.
- Os servidores DNS de upstream do ISP são `10.0.0.62` e `10.0.0.61`.
- A basejail já foi criada e uma árvore de ports instalada como mostrado em Configuração inicial.

Exemplo 35. Executando o BIND em uma Jail

Crie uma interface de loopback clonada adicionando uma linha ao arquivo /etc/rc.conf:

```
cloned_interfaces="lo1"
```

Imediatamente crie a nova interface de loopback:

```
# service netif cloneup
Created clone interfaces: lo1.
```

Crie a jail:

```
# ezjail-admin create dns1 'lo1|127.0.2.1,re0|192.168.1.240'
```

Inicie a jail, conecte-se a ao seu console e realize algumas configurações básicas:

```
# ezjail-admin start dns1
# ezjail-admin console dns1
# passwd
Changing local password for root
New Password:
Retype New Password:
# tzsetup
# sed -i .bak -e '/adjkerntz/ s/^/#/' /etc/crontab
# sed -i .bak -e 's/127.0.0.1/127.0.2.1/g; s/localhost.my.domain/dns1.my.domain
dns1/' /etc/hosts
```

Configure temporariamente os servidores upstream de DNS no arquivo /etc/resolv.conf para que os ports possam ser baixados:

```
nameserver 10.0.0.62
nameserver 10.0.0.61
```

Ainda usando o console da jail, instale o dns/bind99.

```
# make -C /usr/ports/dns/bind99 install clean
```

Configure o servidor de nomes editando o arquivo /usr/local/etc/namedb/named.conf.

Crie uma Access Control List (ACL) de endereços e redes que têm permissão para enviar consultas DNS para este servidor de nomes. Esta seção é adicionada logo antes da seção `options` no arquivo:

```
...
// or cause huge amounts of useless Internet traffic.

acl "trusted" {
    192.168.1.0/24;
    localhost;
    localnets;
};

options {
...
```

Use o endereço IP da jail na configuração `listen-on` para aceitar consultas DNS de outros computadores na rede:

```
    listen-on   { 192.168.1.240; };
```

Um servidor DNS de nomes para cache simples é criado alterando a seção `forwarders`. O arquivo original contém:

```
/*
    forwarders {
        127.0.0.1;
    };
*/
```

Descomente a seção removendo as linhas `/ e/`. Digite os endereços IP dos servidores DNS upstream. Logo após a seção `forwarders`, adicione referências à `trusted` ACL definida anteriormente:

```
    forwarders {
        10.0.0.62;
        10.0.0.61;
    };

    allow-query       { any; };
    allow-recursion   { trusted; };
    allow-query-cache { trusted; };
```

Ative o serviço no arquivo /etc/rc.conf:

```
named_enable="YES"
```

Inicie e teste o servidor de nomes:

```
# service named start
wrote key file "/usr/local/etc/namedb/rndc.key"
Starting named.
# /usr/local/bin/dig @192.168.1.240 freebsd.org
```

Uma resposta que inclui

```
;; Got answer;
```

mostra que o novo servidor DNS está funcionando. Um longo delay seguido por uma resposta incluindo

```
;; connection timed out; no servers could be reached
```

mostra um problema. Verifique as definições de configuração e certifique-se de que quaisquer firewalls locais permitam que o novo DNS acesse os servidores upstream de DNS.

O novo servidor DNS pode usar pra resolução de nomes seu próprio serviço, assim como outros computadores locais. Defina o endereço do servidor DNS no arquivo /etc/resolv.conf do computador-cliente:

```
nameserver 192.168.1.240
```

Um servidor DHCP local pode ser configurado para fornecer este endereço como servidor de DNS local, fornecendo configuração automática em clientes DHCP.

Capítulo 15. Controle de acesso obrigatório

15.1. Sinopse

O FreeBSD suporta extensões de segurança baseadas no projeto POSIX™.1e. Esses mecanismos de segurança incluem as Listas de Controle de Acesso do sistema de arquivos (Listas de Controle de Acesso) e o Controle de Acesso Obrigatório, (Mandatory Access Control - MAC). O MAC permite que os módulos de controle de acesso sejam carregados para implementar políticas de segurança. Alguns módulos fornecem proteções para um subconjunto restrito do sistema, fortalecendo um serviço específico. Outros fornecem segurança rotulada abrangente em todos os assuntos e objetos. A parte obrigatória da definição indica que a imposição de controles é executada pelos administradores e pelo sistema operacional. Isso está em contraste com o mecanismo de segurança padrão do Controle de Acesso Discricionário (Discretionary Access Control - DAC), onde a imposição é deixada a critério dos usuários.

Este capítulo enfoca o framework MAC e o conjunto de módulos de política de segurança plugáveis que o FreeBSD fornece para habilitar vários mecanismos de segurança.

Depois de ler este capítulo, você saberá:

- A terminologia associada ao framework MAC.
- Os recursos dos módulos de política de segurança MAC, bem como a diferença entre uma política rotulada e não rotulada.
- As considerações a se levar em conta antes de configurar um sistema para usar o framework MAC.
- Quais módulos de política de segurança MAC estão incluídos no FreeBSD e como configurá-los.
- Como implementar um ambiente mais seguro usando o framework MAC.
- Como testar a configuração para garantir que o framework MAC foi implementado corretamente.

Antes de ler este capítulo, você deve:

- Entender os fundamentos do UNIX™ e do FreeBSD (Fundamentos do FreeBSD).
- Ter alguma familiaridade com segurança e como ela está presente no FreeBSD (Segurança).



A configuração incorreta do MAC pode causar perda de acesso ao sistema, agravamento de usuários, ou incapacidade de acessar os recursos fornecidos pelo Xorg. Mais importante, o MAC não deve ser usado para proteger completamente um sistema. O framework MAC apenas aumenta uma política de segurança existente. Sem práticas de segurança sólidas e verificações regulares de segurança, o sistema nunca estará completamente seguro.

Os exemplos contidos neste capítulo são para fins de demonstração e os exemplos de configurações *não* devem ser implementadas em um sistema de produção. A implementação de qualquer política de segurança requer um bom entendimento,

> design adequado e testes completos.

Embora este capítulo abranja uma ampla gama de questões de segurança relacionadas à estrututa MAC, o desenvolvimento de novos módulos de políticas de segurança MAC não serão abrangidos. Vários módulos de política de segurança incluídos com o framework MAC possuem características específicas que são fornecidas tanto para o teste quanto para o desenvolvimento de novos módulos. Consulte mac_test(4), mac_stub(4) e mac_none(4) para obter mais informações sobre esses módulos de política de segurança e os diversos mecanismos que eles fornecem.

15.2. Termos chave

Os seguintes termos chave são usados ao se referir ao framework MAC:

- *compartment*: um conjunto de programas e dados a serem particionados ou separados, onde os usuários recebem acesso explícito ao componente específico de um sistema. Um compartimento (compartment) representa um agrupamento, como um grupo de trabalho, departamento, projeto ou tópico. Os compartimentos possibilitam a implementação de uma política de segurança baseada na necessidade de conhecimento.
- *integrity*: o nível de confiança que pode ser colocado nos dados. Como a integridade (integrity) dos dados é elevada, também aumenta a capacidade de confiar nesses dados.
- *level*: a configuração aumentada ou diminuída de um atributo de segurança. À medida que o nível (level) aumenta, sua segurança também é considerada alta.
- *label*: um atributo de segurança que pode ser aplicado a arquivos, diretórios ou outros itens no sistema. Pode ser considerado um selo de confidencialidade. Quando um rótulo (label) é colocado em um arquivo, ele descreve as propriedades de segurança desse arquivo e só permitirá acesso por arquivos, usuários e recursos com uma configuração de segurança semelhante. O significado e a interpretação dos valores do rótulo dependem da configuração da política. Algumas políticas tratam um rótulo como representando a integridade ou o sigilo de um objeto, enquanto outras políticas podem usar rótulos para manter regras de acesso.
- *multilabel*: esta propriedade é uma opção de sistema de arquivos que pode ser configurada no modo usuário único (single-user) usando o tunefs(8), durante a inicialização usando o fstab(5), ou durante a criação de um novo sistema de arquivos. Essa opção permite que um administrador aplique rótulos MAC diferentes em objetos diferentes. Essa opção aplica-se somente aos módulos de política de segurança que suportam rotulagem.
- *single label*: uma política em que o sistema de arquivos inteiro usa um rótulo para impor o controle de acesso sobre o fluxo de dados. Sempre que `multilabel` não estiver definido, todos os arquivos estarão em conformidade com a mesma configuração de rótulo.
- *object*: uma entidade através da qual a informação flui sob a direção de um *sujeito*. Isso inclui diretórios, arquivos, campos, telas, teclados, memória, armazenamento magnético, impressoras ou qualquer outro dispositivo de armazenamento ou movimentação de dados. Um objeto (object) é um contêiner de dados ou um recurso do sistema. O acesso a um objeto significa efetivamente acesso aos seus dados.
- *subject*: qualquer entidade ativa que faz com que as informações fluam entre *objetos*, como um usuário, processo do usuário ou processo do sistema. No FreeBSD, isso é quase sempre um segmento agindo em um processo em nome de um usuário.

- *policy*: uma coleção de regras que define como os objetivos devem ser alcançados. Uma política (policy) geralmente documenta como determinados itens devem ser manipulados. Este capítulo considera uma política como uma coleção de regras que controla o fluxo de dados e informações e define quem tem acesso a esses dados e informações.
- *high-watermark*: esse tipo de política permite o aumento dos níveis de segurança com o objetivo de acessar informações de nível superior. Na maioria dos casos, o nível original é restaurado depois que o processo é concluído. Atualmente, o framework MAC do FreeBSD não inclui este tipo de política.
- *low-watermark*: esse tipo de política permite reduzir os níveis de segurança com o objetivo de acessar informações menos seguras. Na maioria dos casos, o nível de segurança original do usuário é restaurado após a conclusão do processo. O único módulo de política de segurança no FreeBSD para usar isto é o mac_lomac(4).
- *sensitivity*: normalmente usado quando se discute Segurança Multinível (Multilevel Security - MLS). Um nível de sensibilidade (sensitivity) descreve o quão importante ou secreto os dados devem ser. À medida que o nível de sensibilidade aumenta, também aumenta a importância do sigilo ou confidencialidade dos dados.

15.3. Entendendo os rótulos MAC

Um rótulo MAC é um atributo de segurança que pode ser aplicado a sujeitos e objetos em todo o sistema. Ao definir um rótulo, o administrador deve entender suas implicações para evitar o comportamento inesperado ou indesejado do sistema. Os atributos disponíveis em um objeto dependem do módulo de política carregado, pois os módulos de política interpretam seus atributos de maneiras diferentes.

O rótulo de segurança em um objeto é usado como parte de uma decisão de controle de acesso de segurança por uma política. Com algumas políticas, o rótulo contém todas as informações necessárias para tomar uma decisão. Em outras políticas, os rótulos podem ser processados como parte de um conjunto de regras maior.

Existem dois tipos de políticas de rótulos: rótulo único e rótulo múltiplo. Por padrão, o sistema usará rótulo único. O administrador deve estar ciente dos prós e contras de cada um para implementar políticas que atendam aos requisitos do modelo de segurança do sistema.

Uma diretiva de segurança de rótulo único permite que apenas um rótulo seja usado para cada sujeito ou objeto. Como uma política de rótulo único impõe um conjunto de permissões de acesso em todo o sistema, ela fornece menor sobrecarga de administração, mas diminui a flexibilidade das políticas que suportam rotulagem. No entanto, em muitos ambientes, uma única diretiva de rótulo pode ser tudo o que é necessário.

Uma diretiva de segurança de rótulo único é um pouco semelhante ao DAC pois o `root` configura as políticas para que os usuários sejam colocados nas categorias e níveis de acesso apropriados. Uma diferença notável é que muitos módulos de política também podem restringir o `root`. O controle básico sobre os objetos será então liberado para o grupo, mas o `root` poderá revogar ou modificar as configurações a qualquer momento.

Quando apropriado, uma política de rótulos múltiplos pode ser configurada em um sistema de

arquivos UFS passando `multilabel` para o tunefs(8). Uma política de rótulos múltiplos permite que cada sujeito ou objeto tenha seu próprio rótulo MAC independente. A decisão de usar uma política de rótulos múltiplos ou rótulo único é necessária apenas para políticas que implementam o recurso de rotulagem, como `biba`, `lomac` e `mls`. Algumas políticas, como `seeotheruids`, `portacl` e `partition`, não usam rótulos.

Usar uma política de rótulos múltiplos em uma partição e estabelecer um modelo de segurança de rótulos múltiplos pode aumentar a sobrecarga administrativa, já que tudo nesse sistema de arquivos tem um rótulo. Isso inclui diretórios, arquivos e até mesmo nós de dispositivos.

O comando a seguir definirá a flag `multilabel` no sistema de arquivos UFS especificado . Isso só pode ser feito no modo de usuário único e não é um requisito para o sistema de arquivos de swap:

```
# tunefs -l enable /
```



Alguns usuários tiveram problemas com a configuração de flag `multilabel` na partição raiz. Se este for o caso, por favor consulte Solução de problemas do framework MAC.

Como a política de rótulos múltiplos é definida por sistema de arquivos, ela pode não ser necessária se o layout do sistema de arquivos for bem projetado. Considere um exemplo de modelo de segurança MAC para um servidor Web do FreeBSD. Esta máquina usa o rótulo único, `biba/high`, para tudo nos sistemas de arquivos padrão. Se o servidor Web precisar ser executado em `biba/low` para evitar recursos de gravação, ele poderá ser instalado em um sistema de arquivos UFS separado, /usr/local, definido com `biba/low`.

15.3.1. Configuração de rótulo

Praticamente todos os aspectos da configuração do módulo de política de rótulo serão executados usando os utilitários do sistema base. Esses comandos fornecem uma interface simples para a configuração de objeto ou sujeito ou a manipulação e verificação da configuração.

Toda a configuração pode ser feita usando `setfmac`, que é usado para definir rótulos MAC em objetos do sistema, e `setpmac`, que é usado para definir os rótulos em sujeitos do sistema. Por exemplo, para definir o rótulo MAC`biba` como `high` em test:

```
# setfmac biba/high test
```

Se a configuração for bem sucedida, o prompt será retornado sem erro. Um erro comum é `Permission denied`, que geralmente ocorre quando o rótulo está sendo definido ou modificado em um objeto restrito. Outras condições podem produzir falhas diferentes. Por exemplo, o arquivo pode não ser de propriedade do usuário que está tentando re-rotular o objeto, o objeto pode não existir ou o objeto pode ser somente de leitura. Uma política obrigatória não permitirá que o processo renomeie o arquivo, talvez devido a uma propriedade do arquivo, uma propriedade do processo ou uma propriedade do novo valor de rótulo proposto. Por exemplo, se um usuário que estiver executando com baixa integridade tentar alterar o rótulo de um arquivo de alta integridade,

ou um usuário executando com baixa integridade tentar alterar o rótulo de um arquivo de baixa integridade para um rótulo de alta integridade, essas operações falharão.

O administrador do sistema pode usar `setpmac` para substituir as configurações do módulo de política, atribuindo um rótulo diferente a chamada do processo:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Para processos atualmente em execução, como o sendmail, o `getpmac` é normalmente usado. Esse comando usa uma ID de processo (PID) no lugar de um nome de comando. Se os usuários tentarem manipular um arquivo que não esteja em seu acesso, sujeito às regras dos módulos de política carregados, o erro `Operation not permitted` será exibido.

15.3.2. Rótulos pré-definidos

Alguns módulos de política do FreeBSD que suportam o recurso de rotulagem oferecem três rótulos predefinidos: `low`, `equal` e `high`, onde:

- `low` é considerada a configuração de rótulo mais baixa que um objeto ou assunto pode ter. Definir isso em sujeitos ou objetos bloqueia o acesso a objetos ou sujeitos marcados como alto (high).
- `equal` define o sujeito ou objeto a ser desabilitado ou não afetado e deve ser colocado apenas em objetos considerados como isentos da política.
- `high` concede a um objeto ou sujeito a configuração mais alta disponível nos módulos de política Biba e MLS.

Esses módulos de política incluem mac_biba(4), mac_mls(4) e mac_lomac(4). Cada um dos rótulos predefinidos estabelece uma diretiva de fluxo de informações diferentes. Consulte a página de manual do módulo para determinar as características das configurações genéricas de rótulos.

15.3.3. Rótulos numéricos

Os módulos de políticas Biba e MLS suportam um rótulo numérico que pode ser configurado para indicar o nível exato de controle hierárquico. Esse nível numérico é usado para particionar ou classificar informações em diferentes grupos de classificação, permitindo apenas o acesso a esse grupo ou a um nível de grupo mais alto. Por exemplo:

```
biba/10:2+3+6(5:2+3-20:2+3+4+5+6)
```

pode ser interpretado como "Rótulo de Política Biba/Grau 10:Compartimentos 2, 3 e 6: (grau 5 ...")

Neste exemplo, o primeiro grau seria considerado o grau efetivo com compartimentos efetivos, o segundo grau é o grau baixo e o último é o grau alto. Na maioria das configurações, essas definições

refinadas não são necessárias, pois são consideradas configurações avançadas.

Objetos do sistema possuem apenas um grau e compartimento atuais. Os sujeitos do sistema refletem o intervalo de direitos disponíveis no sistema e as interfaces de rede, onde são usados para controle de acesso.

O grau e os compartimentos em um par de sujeito e objeto são usados para construir um relacionamento conhecido como *dominance*, em que um sujeito domina um objeto, o objeto domina o sujeito, nenhum domina o outro, ou ambos dominam cada um. O caso em que "ambos dominam" ocorre quando dois rótulos são iguais. Devido à natureza do fluxo de informações do Biba, um usuário tem direitos sobre um conjunto de compartimentos que podem corresponder aos projetos, mas os objetos também têm um conjunto de compartimentos. Os usuários podem ter que subconjuntar seus direitos usando `su` ou `setpmac` para acessar objetos em um compartimento a partir do qual eles não estão restritos.

15.3.4. Rótulos de usuários

Os usuários precisam ter rótulos para que seus arquivos e processos interajam adequadamente com a política de segurança definida no sistema. Isso é configurado no /etc/login.conf usando classes de login. Todo módulo de política que usa rótulos implementará a configuração da classe de usuário.

Para definir o rótulo padrão da classe de usuário que será imposto pelo MAC, adicione uma entrada `label`. Um exemplo de entrada `label` contendo todos os módulos de política é exibida abaixo. Observe que, em uma configuração real, o administrador nunca habilitaria todos os módulos de política. Recomenda-se que o restante deste capítulo seja revisado antes que qualquer configuração seja implementada.

```
default:\
    :copyright=/etc/COPYRIGHT:\
    :welcome=/etc/motd:\
    :setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
    :path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
    :manpath=/usr/shared/man /usr/local/man:\
    :nologin=/usr/sbin/nologin:\
    :cputime=1h30m:\
    :datasize=8M:\
    :vmemoryuse=100M:\
    :stacksize=2M:\
    :memorylocked=4M:\
    :memoryuse=8M:\
    :filesize=8M:\
    :coredumpsize=8M:\
    :openfiles=24:\
    :maxproc=32:\
    :priority=0:\
    :requirehome:\
    :passwordtime=91d:\
    :umask=022:\
```

```
    :ignoretime@:\
    :label=partition/13,mls/5,biba/10(5-15),lomac/10[2]:
```

Embora os usuários não possam modificar o valor padrão, eles podem alterar seu rótulo após o login, sujeito às restrições da política. O exemplo acima diz à política do Biba que a integridade mínima de um processo é `5`, seu máximo é `15` e o rótulo efetivo padrão é `10`. O processo será executado em `10` até que ele escolha alterar o rótulo, talvez devido ao usuário usar `setpmac`, que será restringido pelo Biba ao intervalo configurado.

Após qualquer alteração no login.conf, o banco de dados de recursos da classe de login deve ser reconstruído usando o `cap_mkdb`.

Muitos sites têm um grande número de usuários que exigem várias classes de usuário diferentes. Um planejamento detalhado é necessário, pois isso pode dificultar o gerenciamento.

15.3.5. Rótulos de interface de rede

Os rótulos podem ser definidos em interfaces de rede para ajudar a controlar o fluxo de dados através da rede. Políticas que usam rótulos de interface de rede funcionam da mesma maneira que as políticas funcionam em relação aos objetos. Usuários com configurações altas no Biba, por exemplo, não terão permissão para acessar interfaces de rede com um rótulo `low`.

Ao definir o rótulo MAC em interfaces de rede, `maclabel` pode ser passado para o `ifconfig`:

```
# ifconfig bge0 maclabel biba/equal
```

Este exemplo irá definir o rótulo MAC de `biba/equal` na interface `bge0`. Ao usar uma configuração semelhante a `biba/high(low-high)`, o rótulo inteiro deve ser citado para evitar que um erro seja retornado.

Cada módulo de política que suporta rotulagem tem um ajuste que pode ser usado para desativar o rótulo MAC em interfaces de rede. Configurar o rótulo para `equal` terá um efeito semelhante. Reveja a saída do `sysctl`, as páginas do manual de políticas e as informações no restante deste capítulo para obter mais informações sobre esses ajustes.

15.4. Planejando a configuração de segurança

Antes de implementar qualquer política de MAC, recomenda-se uma fase de planejamento. Durante as etapas de planejamento, um administrador deve considerar os requisitos e metas de implementação, como:

- Como classificar informações e recursos disponíveis nos sistemas de destino.
- Quais informações ou recursos para restringir o acesso, juntamente com o tipo de restrições que devem ser aplicadas.
- Quais módulos MAC serão necessários para atingir esse objetivo.

Um teste de sistema confiável e sua configuração deve ocorrer *antes* de uma implementação MAC

ser usada em sistemas de produção. Como diferentes ambientes têm diferentes necessidades e requisitos, estabelecer um perfil de segurança completo diminuirá a necessidade de alterações quando o sistema entrar em operação.

Considere como o framework MAC aumenta a segurança do sistema como um todo. Os vários módulos de política de segurança fornecidos pelo framework MAC podem ser usados para proteger a rede e os sistemas de arquivos ou para impedir que usuários acessem determinadas portas e soquetes. Talvez o melhor uso dos módulos de política seja carregar vários módulos de política de segurança por vez para fornecer um ambiente MLS. Essa abordagem difere de uma política rígida, que tipicamente endurece elementos de um sistema que são usados apenas para propósitos específicos. A desvantagem de MLS é o aumento da sobrecarga administrativa.

A sobrecarga é mínima quando comparada ao efeito duradouro de uma estrutura que fornece a capacidade de escolher quais políticas são necessárias para uma configuração específica e que reduzem a sobrecarga de desempenho. A redução do suporte a políticas desnecessárias pode aumentar o desempenho geral do sistema, além de oferecer flexibilidade de escolha. Uma boa implementação consideraria os requisitos gerais de segurança e implementaria efetivamente os vários módulos de política de segurança oferecidos pelo framework.

Um sistema que utiliza MAC garante que um usuário não terá permissão para alterar atributos de segurança à vontade. Todos os utilitários, programas e scripts de usuário devem funcionar dentro das restrições das regras de acesso fornecidas pelos módulos de política de segurança selecionados e o controle das regras de acesso do MAC está nas mãos do administrador do sistema.

É dever do administrador do sistema selecionar cuidadosamente os módulos de política de segurança corretos. Para um ambiente que precisa limitar o controle de acesso na rede, o mac_portacl(4), mac_ifoff(4), e os módulos de políticas mac_biba(4) são bons pontos de partida. Para um ambiente em que a confidencialidade rigorosa dos objetos do sistema de arquivos é necessária, considere mac_bsdextended(4) e os módulos de política mac_mls(4).

Decisões de políticas podem ser tomadas com base na configuração da rede. Se apenas determinados usuários tiverem permissão para acessar o ssh(1), o módulo de política mac_portacl(4) é uma boa escolha. No caso de sistemas de arquivos, o acesso a objetos pode ser considerado confidencial para alguns usuários, mas não para outros. Como um exemplo, uma grande equipe de desenvolvimento pode ser dividida em projetos menores, onde os desenvolvedores do projeto A podem não ter permissão para acessar objetos escritos por desenvolvedores do projeto B. No entanto, ambos os projetos podem precisar acessar objetos criados por desenvolvedores do projeto C. Usando os diferentes módulos de política de segurança fornecidos pelo framework MAC, os usuários poderiam ser divididos nesses grupos e então receber acesso aos objetos apropriados.

Cada módulo de política de segurança tem uma maneira exclusiva de lidar com a segurança geral de um sistema. A seleção de módulos deve se basear em uma política de segurança bem pensada, que pode exigir revisão e reimplementação. Entender os diferentes módulos da política de segurança oferecidos pelo framework MAC ajudará os administradores a escolher as melhores políticas para suas situações.

O restante deste capítulo aborda os módulos disponíveis, descreve seu uso e configuração e, em alguns casos, fornece informações sobre as situações aplicáveis.



A implementação do MAC é muito parecida com a implementação de um firewall, já que é preciso tomar cuidado para evitar que o sistema seja completamente bloqueado. A capacidade de reverter para uma configuração anterior deve ser considerada e a implementação do MAC em uma conexão remota deve ser feita com extrema cautela.

15.5. Políticas MAC Disponíveis

O kernel padrão do FreeBSD inclui a diretiva `options MAC`. Isso significa que todos os módulos incluídos no framework MAC podem ser carregados com o comando `kldload` como um módulo do kernel em tempo de execução. Depois de testar o módulo, adicione o nome do módulo ao arquivo /boot/loader.conf para que ele seja carregado durante a inicialização. Cada módulo também fornece uma opção de kernel para os administradores que escolhem compilar seu próprio kernel personalizado.

O FreeBSD inclui um grupo de políticas que cobrirá a maioria dos requisitos de segurança. Cada política é resumida abaixo. As três últimas políticas suportam configurações inteiras no lugar dos três rótulos padrão.

15.5.1. O MAC vê a Política de Outros UIDs

Nome do módulo: mac_seeotheruids.ko

Linha de configuração do kernel: `options MAC_SEEOTHERUIDS`

Opção de inicialização: `mac_seeotheruids_load="YES"`

O módulo mac_seeotheruids(4) amplia os ajustes `security.bsd.see_other_uids` e `security.bsd.see_other_gids` do `sysctl`. Esta opção não requer que nenhum rótulo seja definido antes da configuração e pode operar de forma transparente com outros módulos.

Depois de carregar o módulo, os seguintes ajustes `sysctl` podem ser usados para controlar seus recursos:

- O `security.mac.seeotheruids.enabled` ativa o módulo e implementa as configurações padrões que impedem que os usuários visualizem processos e soquetes pertencentes a outros usuários.
- `security.mac.seeotheruids.specificgid_enabled` permite que grupos especificados sejam isentos desta política. Para isentar grupos específicos, use a variável `security.mac.seeotheruids.specificgid=XXX` do `sysctl`, substituindo *XXX* pelo ID numérico do grupo a ser isento.
- `security.mac.seeotheruids.primarygroup_enabled` é usado para isentar grupos primários específicos desta política. Ao usar este ajuste, o `security.mac.seeotheruids.specificgid_enabled` não pode estar definido.

15.5.2. A Política Estendida do BSD MAC

Nome do módulo: mac_bsdextended.ko

Linha de configuração do kernel: `options MAC_BSDEXTENDED`

Opção de inicialização: `mac_bsdextended_load="YES"`

O módulo mac_bsdextended(4) aplica um firewall no sistema de arquivos. Ele fornece uma extensão para o modelo de permissões do sistema de arquivos padrão, permitindo que um administrador crie um conjunto de regras semelhante a um firewall para proteger arquivos, utilitários e diretórios na hierarquia do sistema de arquivos. Quando se tenta acessar um objeto do sistema de arquivos, a lista de regras é iterada até que uma regra correspondente seja localizada ou o final seja atingido. Esse comportamento pode ser alterado usando `security.mac.bsdextended.firstmatch_enabled`. Semelhante a outros módulos de firewall no FreeBSD, um arquivo contendo as regras de controle de acesso pode ser criado e lido pelo sistema no momento da inicialização usando uma variável do rc.conf(5).

A lista de regras pode ser inserida usando o ugidfw(8) que possui uma sintaxe similar ao ipfw(8). Mais ferramentas podem ser escritas usando as funções da biblioteca libugidfw(3).

Depois que o módulo mac_bsdextended(4) tiver sido carregado, o seguinte comando poderá ser usado para listar a configuração atual da regra:

```
# ugidfw list
0 slots, 0 rules
```

Por padrão, nenhuma regra é definida e tudo está completamente acessível. Para criar uma regra que bloqueia todo o acesso dos usuários, mas que não afeta o ` root `:

```
# ugidfw add subject not uid root new object not uid root mode n
```

Embora essa regra seja simples de implementar, é uma idéia muito ruim, pois impede que todos os usuários emitam comandos. Um exemplo mais realista bloqueia todo o acesso do `user1`, incluindo listagens de diretórios, ao diretório inicial do usuário `user2` :

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Em vez de `user1`, `not uid_user2` poderia ser usado para impor as mesmas restrições de acesso para todos os usuários. No entanto, o usuário `root` não é afetado por essas regras.



Deve-se ter extremo cuidado ao trabalhar com este módulo, pois o uso incorreto pode bloquear o acesso a certas partes do sistema de arquivos.

15.5.3. A política de silenciamento da interface MAC

Nome do módulo: mac_ifoff.ko

Linha de configuração do kernel: `options MAC_IFOFF`

Opção de inicialização: `mac_ifoff_load="YES"`

O módulo mac_ifoff(4) é usado para desabilitar as interfaces de rede e evitar que as interfaces de rede sejam ativadas durante a inicialização do sistema. Ele não usa rótulos e não depende de nenhum outro módulo MAC.

A maior parte do controle deste módulo é realizada através destes ajustes `sysctl`:

- `security.mac.ifoff.lo_enabled` ativa ou desativa todo o tráfego na interface de loopback, lo(4).
- `security.mac.ifoff.bpfrecv_enabled` ativa ou desativa todo o tráfego na interface do Filtro de Pacotes Berkeley, bpf(4).
- `security.mac.ifoff.other_enabled` ativa ou desativa o tráfego em todas as outras interfaces.

Um dos usos mais comuns do mac_ifoff(4) é o monitoramento de rede em um ambiente onde o tráfego de rede não deve ser permitido durante a sequência de inicialização. Outro uso seria escrever um script que usa um aplicativo como o security/aide para bloquear automaticamente o tráfego da rede se encontrar arquivos novos ou alterados em diretórios protegidos.

15.5.4. A política de lista de controle de acesso da porta MAC

Nome do módulo: mac_portacl.ko

Linha de configuração do kernel: `MAC_PORTACL`

Opção de inicialização: `mac_portacl_load="YES"`

O módulo mac_portacl(4) é usado para limitar a ligação a portas TCP e UDP locais , tornando possível permitir que usuários non-`root` sejam vinculados a portas privilegiadas especificadas abaixo de 1024.

Uma vez carregado, este módulo habilita a política MAC em todos os sockets. Os seguintes ajustes estão disponíveis:

- `security.mac.portacl.enabled` ativa ou desativa a política completamente.
- A `security.mac.portacl.port_high` configura o número de porta mais alto que o mac_portacl(4) protege.
- A `security.mac.portacl.suser_exempt`, quando configurada para um valor diferente de zero, isenta o usuário `root` desta política.
- A `security.mac.portacl.rules` especifica a política como uma cadeia de texto no formato `rule [, rule, ⋯]`, com tantas regras quantas forem necessárias, e onde cada regra esta na forma `idtype:id:protocol:port`. O idtype é `uid` ou `gid`. O parâmetro protocol pode ser `tcp` ou `udp`. O parâmetro port é o número da porta para permitir que o usuário ou grupo especificado se vincule. Somente valores numéricos podem ser usados para os parâmetros ID do usuário, ID do grupo e porta.

Por padrão, as portas abaixo de 1024 só podem ser usadas por processos privilegiados que são executados como `root`. Para que o mac_portacl(4) permita que processos não privilegiados se vinculem a portas abaixo de 1024, defina os seguintes ajustes da seguinte forma:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0
# sysctl net.inet.ip.portrange.reservedhigh=0
```

Para evitar que o usuário `root` seja afetado por esta política, configure `security.mac.portacl.suser_exempt` para um valor diferente de zero.

```
# sysctl security.mac.portacl.suser_exempt=1
```

Para permitir que o usuário `www` com UID 80 seja vinculado à porta 80 sem precisar do privilégio `root`:

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Este próximo exemplo permite que o usuário com o UID de 1001 se vincule às portas TCP 110 (POP3) e 995 (POP3):

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

15.5.5. A Política de Partição MAC

Nome do módulo: mac_partition.ko

Linha de configuração do kernel: `options MAC_PARTITION`

Opção de inicialização: `mac_partition_load="YES"`

A política mac_partition(4) coloca os processos em "partições" específicas com base no rótulo MAC. A maioria das configurações para esta política é feita usando setpmac(8). Uma vari[avek `sysctl` está disponível para esta política:

- A `security.mac.partition.enabled` permite a aplicação de partições de processo MAC.

Quando essa política esta ativada, os usuários só poderão ver seus processos e quaisquer outros em sua partição, mas não terão permissão para trabalhar com utilitários fora do escopo dessa partição. Por exemplo, um usuário na classe `insecure` não terá permissão para acessar `top`, bem como muitos outros comandos que devem fazer spawn de um processo.

Este exemplo adiciona o `top` ao conjunto de rótulos dos usuários na classe `insecure`. Todos os processos gerados por usuários na classe `insecure` permanecerão no rótulo `partition/13`.

```
# setpmac partition/13 top
```

Este comando exibe o rótulo da partição e a lista de processos:

```
# ps Zax
```

Esse comando exibe o rótulo da partição de processo de outro usuário e os processos atualmente em execução desse usuário:

```
# ps -ZU trhodes
```



Os usuários podem ver processos no rótulo `root`, a menos que a política mac_seeotheruids(4) esteja carregada.

15.5.6. O módulo de segurança multinível MAC

Nome do módulo: mac_mls.ko

Linha de configuração do kernel: `options MAC_MLS`

Opção de inicialização: `mac_mls_load="YES"`

A política mac_mls(4) controla o acesso entre sujeitos e objetos no sistema, aplicando uma diretiva de fluxo de informações restrita.

Em ambientes MLS, um nível de "clearance" é definido no rótulo de cada sujeito ou objeto, juntamente com os compartimentos. Como esses níveis de liberação podem atingir números maiores que vários milhares, seria uma tarefa difícil configurar completamente cada sujeito ou objeto. Para facilitar essa sobrecarga administrativa, três rótulos são incluídos nesta política: `mls/low`, `mls/equal` e `mls/high`, onde:

- Qualquer coisa rotulada com `mls/low` terá um nível de folga baixo e não será permitido acessar informações de um nível superior. Esse rótulo também evita que objetos de nível de liberação mais alto gravem ou transmitam informações para um nível inferior.
- `mls/equal` deve ser colocado em objetos que devem ser isentos da política.
- `mls/high` é o nível mais alto de permissão possível. Objetos atribuídos a esse rótulo terão domínio sobre todos os outros objetos no sistema; no entanto, eles não permitirão o vazamento de informações para objetos de classe baixa.

O MLS fornece:

- Um nível de segurança hierárquico com um conjunto de categorias não hierárquicas.
- Regras fixas de `no read up, no write down`. Isso significa que um sujeito pode ter acesso de leitura a objetos em seu próprio nível ou abaixo, mas não acima. Da mesma forma, um sujeito pode ter acesso de gravação a objetos em seu próprio nível ou acima, mas não abaixo dele.
- Sigilo, ou a prevenção de divulgação inadequada de dados.
- Uma base para o projeto de sistemas que lidam simultaneamente com dados em múltiplos níveis de sensibilidade sem vazar informações entre secretas e confidenciais.

Os seguintes ajustes `sysctl` estão disponíveis:

- `security.mac.mls.enabled` é usado para habilitar ou desabilitar a política MLS.
- `security.mac.mls.ptys_equal` todos os dispositivos pty(4) como `mls/equal` durante a criação.
- `security.mac.mls.revocation_enabled` revoga o acesso a objetos depois que seu rótulo é alterado para um rótulo de nível inferior.
- `security.mac.mls.max_compartments` define o número máximo de níveis de compartimentos permitidos em um sistema.

Para manipular os rótulos MLS, use setfmac(8). Para atribuir um rótulo a um objeto:

```
# setfmac mls/5 test
```

Para obter o rótulo MLS para o arquivo test:

```
# getfmac test
```

Outra abordagem é criar um arquivo de política mestre em /etc/, que especifica as informações de política de MLS e alimentar o `setfmac` com esse arquivo.

Ao usar o módulo de política do MLS, um administrador planeja controlar o fluxo de informações confidenciais. O padrão `block read up block write down` define tudo para um estado baixo. Tudo é acessível e um administrador aumenta lentamente a confidencialidade das informações.

Além das três opções básicas de rótulo, um administrador pode agrupar usuários e grupos conforme necessário para bloquear o fluxo de informações entre eles. Pode ser mais fácil olhar as informações em níveis de clearance usando palavras descritivas, como classificações de `Confidential`, `Secret` e `Top Secret`. Alguns administradores criam grupos diferentes com base nos níveis do projeto. Independentemente do método de classificação, um plano bem pensado deve existir antes de implementar uma política restritiva.

Alguns exemplos de situações para o módulo de política MLS incluem um servidor Web de e-commerce, um servidor de arquivos com informações críticas sobre a empresa e ambientes de instituições financeiras.

15.5.7. O Módulo MAC Biba

Nome do módulo: mac_biba.ko

Linha de configuração do kernel: `options MAC_BIBA`

Opção de inicialização: `mac_biba_load="YES"`

O módulo mac_biba(4) carrega a política MAC Biba. Essa política é semelhante à política MLS, com a exceção de que as regras para o fluxo de informações são levemente revertidas. Isso evita o fluxo descendente de informações confidenciais, enquanto a política MLS impede o fluxo ascendente de informações confidenciais.

Nos ambientes do Biba, um rótulo "integrity" é definido em cada sujeito ou objeto. Esses rótulos são compostos de classes hierárquicas e componentes não hierárquicos. Como um grau ascende, o mesmo acontece com a sua integridade.

Rótulos suportados são `biba/low`, `biba/equal` e `biba/high`, onde:

- `biba/low` é considerado a integridade mais baixa que um sujeito ou objeto pode ter. Definir isso em sujeitos ou objetos bloqueia o acesso de gravação a objetos ou sujeitos marcados como `biba/high`, mas não impede o acesso de leitura.
- `biba/equal` só deve ser colocado em objetos considerados como isentos da política.
- `biba/high` permite gravar objetos em um rótulo inferior, mas não permite a leitura desse objeto. Recomenda-se que esse rótulo seja colocado em objetos que afetam a integridade de todo o sistema.

O Biba fornece:

- Níveis de integridade hierárquica com um conjunto de categorias de integridade não hierárquicas.
- As regras fixas são `no write up, no read down`, o oposto do MLS. Um sujeito pode ter acesso de gravação a objetos em seu próprio nível ou abaixo, mas não acima. Da mesma forma, um sujeito pode ter acesso de leitura a objetos em seu próprio nível ou acima, mas não abaixo.
- Integridade, impedindo a modificação inadequada de dados.
- Níveis de integridade em vez dos níveis de sensibilidade do MLS.

Os seguintes ajustes podem ser usados para manipular a política Biba:

- `security.mac.biba.enabled` é usado para ativar ou desativar a imposição da política Biba na máquina de destino.
- O `security.mac.biba.ptys_equal` é usado para desabilitar a política Biba em dispositivos pty(4).
- `security.mac.biba.revocation_enabled` força a revogação do acesso a objetos se o rótulo for alterado para dominar o sujeito.

Para acessar a configuração de política Biba em objetos do sistema, use `setfmac` e `getfmac`:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

Integridade, que é diferente de sensibilidade, é usada para garantir que a informação não seja manipulada por partes não confiáveis. Isso inclui informações passadas entre sujeitos e objetos. Ele garante que os usuários só poderão modificar ou acessar as informações para as quais receberam acesso explícito. O módulo de política de segurança mac_biba(4) permite que um administrador configure quais arquivos e programas um usuário pode ver e invocar enquanto assegura que os programas e arquivos sejam confiáveis pelo sistema para esse usuário.

Durante a fase de planejamento inicial, um administrador deve estar preparado para particionar os

usuários em graus, níveis e áreas. O sistema terá como padrão um rótulo alto assim que esse módulo de política for ativado e cabe ao administrador configurar as diferentes classificações e níveis para os usuários. Em vez de usar níveis de liberação, um bom método de planejamento pode incluir tópicos. Por exemplo, permita apenas que os desenvolvedores modifiquem o acesso ao repositório do código-fonte, ao compilador do código-fonte e a outros utilitários de desenvolvimento. Outros usuários seriam agrupados em outras categorias, como testadores, designers ou usuários finais, e somente o acesso de leitura seria permitido.

Um sujeito de integridade inferior é incapaz de escrever para um sujeito de integridade superior e um sujeito de integridade superior não pode listar ou ler um objeto de integridade inferior. Definir um rótulo com o grau mais baixo possível pode torná-lo inacessível aos sujeitos. Alguns ambientes em potencial para esse módulo de política de segurança incluiriam um servidor Web restrito, uma máquina de desenvolvimento e teste e um repositório de código-fonte. Uma implementação menos útil seria uma estação de trabalho pessoal, uma máquina usada como roteador ou um firewall de rede.

15.5.8. O módulo MAC de marca d'água baixa

Nome do módulo: mac_lomac.ko

Linha de configuração do kernel: `options MAC_LOMAC`

Opção de inicialização: `mac_lomac_load="YES"`

Diferentemente da política do MAC Biba, a política mac_lomac(4) permite acesso a objetos de baixa integridade somente após diminuir o nível de integridade para não interromper nenhuma regra de integridade.

A política de integridade de marca d'água baixa funciona de forma quase idêntica ao Biba, com a exceção do uso de rótulos flutuantes para suportar o rebaixamento do sujeito por meio de um compartimento auxiliar de classificação. Este compartimento secundário assume o formato `[auxgrade]`. Ao atribuir uma política com um grau auxiliar, use a sintaxe `lomac/10[2]`, onde `2` é o grau auxiliar.

Essa política se baseia na rotulagem onipresente de todos os objetos do sistema com rótulos de integridade, permitindo que os sujeitos leiam objetos de baixa integridade e fazendo o downgrade do rótulo no sujeito para evitar gravações futuras em objetos de alta integridade usando `[auxgrade]` . A política pode fornecer maior compatibilidade e exigir menos configuração inicial do que o Biba.

Como as políticas Biba e MLS, `setfmac` e `setpmac` são usadas para colocar rótulos nos objetos do sistema:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Um grau auxiliar `low` é uma funcionalidade fornecida apenas pela política MACLOMAC.

15.6. Bloqueio do Usuário

Este exemplo considera um sistema de armazenamento relativamente pequeno com menos de cinquenta usuários. Os usuários terão recursos de login e terão permissão para armazenar dados e acessar recursos.

Para este cenário, os módulos de política mac_bsdextended(4) e mac_seeotheruids(4) podem coexistir e bloquear o acesso a objetos do sistema enquanto ocultam processos do usuário.

Comece adicionando a seguinte linha ao /boot/loader.conf:

```
mac_seeotheruids_load="YES"
```

O módulo de política de segurança mac_bsdextended(4) pode ser ativado adicionando esta linha ao arquivo /etc/rc.conf:

```
ugidfw_enable="YES"
```

As regras padrões armazenadas em /etc/rc.bsdextended serão carregadas na inicialização do sistema. No entanto, as entradas padrões podem precisar de modificação. Como esta máquina é destinada apenas para servir os usuários, tudo pode ser deixado comentado, exceto as duas últimas linhas, a fim de forçar o carregamento de objetos do sistema de propriedade do usuário por padrão.

Adicione os usuários necessários a esta máquina e reinicie. Para fins de teste, tente efetuar login como um usuário diferente em dois consoles. Execute `ps aux` para ver se os processos de outros usuários estão visíveis. Verifique se a execução do ls(1) no diretório inicial de outro usuário falha.

Não tente testar com o usuário `root`, a menos que o `sysctl` específico tenha sido modificado para bloquear o acesso do superusuário.



Quando um novo usuário é adicionado, sua regra mac_bsdextended(4) não estará na lista de conjuntos de regras. Para atualizar o conjunto de regras rapidamente, descarregue o módulo de política de segurança e recarregue-o novamente usando kldunload(8) e kldload(8).

15.7. Nagios em Jail MAC

Esta seção demonstra as etapas necessárias para implementar o sistema de monitoramento de rede Nagios em um ambiente MAC. Isso é um exemplo que ainda exige que o administrador teste se a política implementada atende aos requisitos de segurança da rede antes de usar em um ambiente de produção.

Este exemplo requer que o `multilabel` seja definido em cada sistema de arquivos. Ele também assume que o net-mgmt/nagios-plugins, net-mgmt/nagios e www/apache22 estão todos instalados, configurados e funcionando corretamente antes de tentar a integração na estrutura MAC.

15.7.1. Criar uma Classe de Usuário Insegura

Comece o procedimento adicionando a seguinte classe de usuário ao /etc/login.conf:

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
:manpath=/usr/shared/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=biba/10(10-10):
```

Em seguida, adicione a seguinte linha a seção de classe de usuário padrão:

```
:label=biba/high:
```

Salve as edições e rode o seguinte comando para reconstruir o banco de dados:

```
# cap_mkdb /etc/login.conf
```

15.7.2. Configurar usuários

Configure o usuário `root` para a classe padrão usando:

```
# pw usermod root -L default
```

Todas as contas de usuário que não são `root` agora exigirão uma classe de login. A classe de login é necessária, caso contrário, os usuários terão acesso recusado aos comandos comuns. O seguinte script `sh` deve resolver:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
    /etc/passwd`; do pw usermod $x -L default; done;
```

Em seguida, altere as contas `nagios` e `www` para a classe insegura:

```
# pw usermod nagios -L insecure
# pw usermod www -L insecure
```

15.7.3. Crie o arquivo de contextos

Um arquivo de contexto deve agora ser criado como /etc/policy.contexts:

```
# This is the default BIBA policy for this system.

# System:
/var/run(/.*)?          biba/equal

/dev/(/.*)?         biba/equal

/var                biba/equal
/var/spool(/.*)?        biba/equal

/var/log(/.*)?          biba/equal

/tmp(/.*)?          biba/equal
/var/tmp(/.*)?          biba/equal

/var/spool/mqueue       biba/equal
/var/spool/clientmqueue     biba/equal

# For Nagios:
/usr/local/etc/nagios(/.*)? biba/10

/var/spool/nagios(/.*)?     biba/10

# For apache
/usr/local/etc/apache(/.*)? biba/10
```

Essa política impõe segurança ao definir restrições no fluxo de informações. Nesta configuração específica, os usuários, incluindo O `root`, nunca devem ter permissão para acessar o Nagios. Arquivos de configuração e processos que fazem parte do Nagios serão completamente auto-contidos ou presos.

Este arquivo será lido depois da execução do `setfsmac` em cada sistema de arquivos. Este exemplo define a política no sistema de arquivos raiz:

```
# setfsmac -ef /etc/policy.contexts /
```

Em seguida, adicione estas edições a seção principal do /etc/mac.conf:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

15.7.4. Configuração do Inicializador

Para finalizar a configuração, adicione as seguintes linhas ao /boot/loader.conf:

```
mac_biba_load="YES"
mac_seeotheruids_load="YES"
security.mac.biba.trust_all_interfaces=1
```

E a seguinte linha para a configuração da placa de rede armazenada em /etc/rc.conf. Se a configuração de rede principal for feita via DHCP, talvez seja necessário configurá-la manualmente após cada inicialização do sistema:

```
maclabel biba/equal
```

15.7.5. Testando a Configuração

Primeiro, certifique-se de que o servidor Web e o Nagios não iniciarão na inicialização e reinicialização do sistema. Assegure-se de que o `root` não possa acessar nenhum dos arquivos no diretório de configuração do Nagios. Se o `root` puder listar o conteúdo de /var/spool/nagios, algo está errado. Em vez disso, um erro "permission denied" deve ser retornado.

Se tudo parecer bem, o Nagios, o Apache e o Sendmail agora poderão ser iniciados:

```
# cd /etc/mail && make stop && \
setpmac biba/equal make start && setpmac biba/10\(10-10\) apachectl start && \
setpmac biba/10\(10-10\) /usr/local/etc/rc.d/nagios.sh forcestart
```

Verifique novamente para garantir que tudo esteja funcionando corretamente. Caso contrário, verifique os arquivos de log em busca de mensagens de erro. Se necessário, use o sysctl(8) para desativar o módulo de política de segurança mac_biba(4) e tente iniciar tudo novamente.



O usuário `root` ainda pode alterar a aplicação de segurança e editar seus arquivos de configuração. O comando a seguir permitirá a degradação da política de segurança para um nível inferior para um shell recém executado:

```
# setpmac biba/10 csh
```

Para impedir que isso aconteça, force o usuário a um intervalo usando login.conf(5). Se o setpmac(8) tentar executar um comando fora do intervalo do compartimento, um erro será retornado e o comando não será executado. Nesse caso, defina root como `biba/high(high-high)`.

15.8. Solução de problemas do framework MAC

Esta seção discute erros de configuração comuns e como resolvê-los.

O sinalizador `multilabel` não fica habilitado na partição raiz (/)

As etapas a seguir podem resolver este erro transitório:

a. Edite /etc/fstab e defina a partição raiz como somente leitura `ro`.

b. Reinicie no modo single user.

c. Execute `tunefs -l enable` no /.

d. Reinicie o sistema.

e. Execute `mount -urw/` e mude a opção `ro` de volta para `rw` no /etc/fstab e reinicie o sistema novamente.

f. Verifique novamente a saída do `mount` para garantir que o `multilabel` tenha sido configurado corretamente no sistema de arquivos raiz.

Depois de estabelecer um ambiente seguro com o MAC, o Xorg não inicia mais

Isso pode ser causado pela política MAC `partition` ou por uma rotulagem incorreta em uma das políticas de rotulagem do MAC. Para depurar, tente o seguinte:

a. Verifique a mensagem de erro. Se o usuário estiver na classe `insecure`, a política `partition` pode ser a culpada. Tente definir a classe do usuário de volta para a classe `default` e reconstrua o banco de dados com o `cap_mkdb`. Se isso não mitigar o problema, vá para a etapa dois.

b. Verifique duas vezes se as políticas de rótulo estão definidas corretamente para o usuário, para o Xorg e para as entradas no /dev.

c. Se nenhum destes resolver o problema, envie a mensagem de erro e uma descrição do ambiente para a lista de discussão de perguntas gerais sobre o FreeBSD.

O erro `_secure_path: unable to stat .login_conf` aparece

Esse erro pode aparecer quando um usuário tenta alternar do usuário `root` para outro usuário no sistema. Essa mensagem geralmente ocorre quando o usuário possui uma qualificação mais alta do que a do usuário que ele está tentando se tornar. Por exemplo, se `joe` tiver uma classificação padrão de `biba/low` e o `root` tiver uma classificação de `biba/high`, o `root` não poderá

visualizar o diretório inicial de `joe`. Isso acontecerá independente se o `root` usou ou não o `su` para se tornar o `joe`, pois o modelo de integridade do Biba não permitirá que o `root` exiba objetos definidos em um nível de integridade mais baixo.

O sistema não reconhece mais o `root`

Quando isso ocorre, o `whoami` retorna `0` e `su` retorna `who are you?`.

Isso pode acontecer se uma política de rotulagem foi desativada por sysctl(8) ou o módulo de política foi descarregado. Se a política estiver desativada, o banco de dados de recursos de login precisará ser reconfigurado. Verifique duas vezes o /etc/login.conf para garantir que todas as opções de `label` tenham sido removidas e reconstrua o banco de dados com `cap_mkdb`.

Isso também pode acontecer se uma política restringir o acesso ao master.passwd. Isso geralmente é causado por um administrador que altera o arquivo sob um rótulo que entra em conflito com a política geral que está sendo usada pelo sistema. Nesses casos, as informações do usuário seriam lidas pelo sistema e o acesso seria bloqueado, pois o arquivo herdaria o novo rótulo. Desative a política usando o sysctl(8) e tudo deve retornar ao normal.

Capítulo 16. Auditoria de Evento de Segurança

16.1. Sinopse

O sistema operacional FreeBSD inclui suporte para auditoria de eventos de segurança. A auditoria de eventos oferece suporte a registros confiáveis, detalhados e configuráveis de diversos eventos do sistema relevantes para a segurança, incluindo logins, alterações de configuração e acesso a arquivos e rede. Esses registros de log podem ser inestimáveis para monitoramento de sistema em tempo real, detecção de intrusão e análise "post mortem". O FreeBSD implementa a Application Programming Interface (API) Basic Security Module (BSM) publicada pela Sun™ e o formato de arquivo, e é interoperável com as implementações de auditoria do Solaris™ e do Mac OS™ X.

Este capítulo se concentra na instalação e configuração da auditoria de eventos. Ele explica as políticas de auditoria e fornece um exemplo de configuração de auditoria.

Depois de ler este capítulo, você saberá:

- O que é auditoria de eventos e como funciona.
- Como configurar a auditoria de eventos no FreeBSD para usuários e processos.
- Como revisar o caminho da auditoria usando as ferramentas de auditoria para redução e revisão.

Antes de ler este capítulo, você deve:

- Entender os fundamentos do UNIX™ e do FreeBSD (Fundamentos do FreeBSD).
- Familiarize-se com os conceitos básicos de configuração/compilação do kernel (Configurando o kernel do FreeBSD).
- Ter alguma familiaridade com segurança e como ela está presente no FreeBSD (Segurança).

O recurso de auditoria possui algumas limitações conhecidas. Nem todos os eventos do sistema que são relevantes para a segurança são auditáveis, e também alguns mecanismos de login, como gerenciadores de exibição baseados em Xorg e daemons de terceiros, não configuram adequadamente a auditoria para sessões de login do usuário.



O recurso de auditoria de eventos de segurança é capaz de gerar logs muito detalhados da atividade do sistema. Em um sistema muito utilizado, os dados do arquivo de rastreamento podem ser muito grandes quando configurados para grandes detalhes, excedendo gigabytes por semana em algumas configurações. Os administradores devem levar em consideração os requisitos de espaço em disco associados a configurações de auditoria de alto volume. Por exemplo, pode ser desejável dedicar um sistema de arquivos ao /var/audit para que outros sistemas de arquivos não sejam afetados se o sistema de arquivos de auditoria ficar cheio.

16.2. Termos chave

Os termos a seguir estão relacionados a auditoria de eventos de segurança:

- *event*: um evento auditável é qualquer evento que pode ser registrado usando o subsistema de auditoria. Exemplos de eventos relevantes para a segurança incluem a criação de um arquivo, a construção de uma conexão de rede ou o logon de um usuário. Os eventos são "atribuíveis", o que significa que podem ser rastreados para um usuário autenticado, ou "não atribuível". Exemplos de eventos não atribuíveis são eventos que ocorrem antes da autenticação no processo de login, como tentativas de senha incorreta.
- *class*: um conjunto nomeado de eventos relacionados que são usados em expressões de seleção. As classes de eventos comumente usadas incluem "file creation" (fc), "exec" (ex), e "login_logout" (lo).
- *record*: uma entrada de log de auditoria que descreve um evento de segurança. Os registros contêm um tipo de evento de registro, informações sobre o assunto (usuário) executando a ação, informações de data e hora, informações sobre quaisquer objetos ou argumentos e uma condição de sucesso ou falha.
- *trail*: um arquivo de log que consiste em uma série de registros de auditoria que descrevem eventos de segurança. As trilhas estão em ordem cronológica aproximada com relação aos eventos concluídos. Apenas processos autorizados podem enviar registros para a trilha de auditoria.
- *selection expression*: uma string contendo uma lista de prefixos e nomes de classes de eventos de auditoria usados para combinar eventos.
- *preselection*: o processo pelo qual o sistema identifica quais eventos são de interesse do administrador. A configuração de pré-seleção usa uma série de expressões de seleção para identificar quais classes de eventos auditar quais usuários, bem como configurações globais que se aplicam a processos autenticados e não autenticados.
- *reduction*: o processo pelo qual os registros das trilhas de auditoria existentes são selecionados para preservação, impressão ou análise. Da mesma forma, o processo pelo qual os registros de auditoria indesejados são removidos da trilha de auditoria. Usando a redução, os administradores podem implementar políticas para a preservação de dados de auditoria. Por exemplo, trilhas de auditoria detalhadas podem ser mantidas por um mês, mas depois disso, as trilhas podem ser reduzidas para preservar apenas as informações de login para fins de arquivamento.

16.3. Configuração de Auditoria

O suporte para auditoria de eventos no espaço do usuário é instalado como parte do sistema operacional básico do FreeBSD. O suporte a kernel está disponível no kernel GENERIC por padrão, e auditd(8) pode ser ativado adicionando a seguinte linha no /etc/rc.conf:

```
auditd_enable="YES"
```

Em seguida, inicie o daemon de auditoria:

```
# service auditd start
```

Usuários que preferem compilar um kernel personalizado devem incluir a seguinte linha em seu arquivo de configuração de kernel personalizado:

```
options AUDIT
```

16.3.1. Expressões de Seleção de Eventos

Expressões de seleção são usadas em vários lugares na configuração de auditoria para determinar quais eventos devem ser auditados. Expressões contêm uma lista de classes de eventos para correspondência. As expressões de seleção são avaliadas da esquerda para a direita e duas expressões são combinadas, acrescentando uma à outra.

Classes de Eventos de Auditoria Padrão resume as classes de eventos de auditoria padrão:

Tabela 12. Classes de Eventos de Auditoria Padrão

Nome da classe	Descrição	Ação
all	all	Corresponde todas as classes de eventos.
aa	autenticação e autorização	
ad	administrativo	Ações administrativas executadas no sistema como um todo.
ap	aplicação	Ação definida pela aplicação.
cl	file close	Auditar chamadas para a chamada de sistema `close`.
ex	exec	Execução do programa de auditoria. Auditoria de argumentos de linha de comando e variáveis de ambiente são controladas via audit_control(5) usando os parâmetros `argv` e `envv` para a configuração da `política`.
fa	acesso ao atributo de arquivo	Audite o acesso de atributos de objetos como stat(1) e pathconf(2).
fc	file create	Eventos de auditoria em que um arquivo é criado.
fd	file delete	Eventos de auditoria onde ocorre a exclusão de arquivos.

Nome da classe	Descrição	Ação
fm	file attribute modify	Eventos de auditoria onde ocorre a modificação do atributo do arquivo, como chown(8), chflags(1), e flock(2).
fr	file read	Eventos de auditoria nos quais dados são lidos ou arquivos são abertos para leitura.
fw	file write	Eventos de auditoria nos quais os dados são gravados ou os arquivos são gravados ou modificados.
io	ioctl	Auditar o uso da chamada de sistema `ioctl`.
ip	ipc	Auditar várias formas de comunicação entre processos, incluindo pipes POSIX e operações IPC do System V.
lo	login_logout	Audite os eventos login(1) e logout(1).
na	não atribuível	Auditar eventos não atribuíveis.
no	classe inválida	Não coincidir com eventos de auditoria.
nt	rede (network)	Eventos de auditoria relacionados a ações de rede, como connect(2) e accept(2).
ot	outros	Auditoria de eventos diversos.
pc	processo	Auditar operações de processos, como exec(3) e exit(3).

Essas classes de eventos de auditoria podem ser personalizadas modificando os arquivos de configuração audit_class e audit_event.

Cada classe de eventos de auditoria pode ser combinada com um prefixo indicando se as operações com êxito/falha são correspondidas e se a entrada está adicionando ou removendo a correspondência para a classe e o tipo. Prefixos para Classes de Eventos de Auditoria resume os prefixos disponíveis:

Tabela 13. Prefixos para Classes de Eventos de Auditoria

Prefixo	Ação
+	Auditoria de eventos bem sucedidos nesta classe.

Prefixo	Ação
-	Auditoria de eventos com falha nesta classe.
^	Auditoria de eventos nem com sucesso e nem com falha nesta classe.
^+	Não faça auditoria de eventos bem-sucedidos nesta classe.
^-	Não audite eventos com falha nesta classe.

Se nenhum prefixo estiver presente, as instâncias com êxito e com falha do evento serão auditadas.

O seguinte exemplo de sequência de seleção seleciona eventos de login/logout bem-sucedidos e com falha, mas apenas eventos de execução bem-sucedidos:

```
lo,+ex
```

16.3.2. Arquivos de Configuração

Os seguintes arquivos de configuração para auditoria de eventos de segurança são encontrados em /etc/security:

- audit_class: contém as definições das classes de auditoria.
- audit_control: controla os aspectos do subsistema de auditoria, como as classes de auditoria padrão, o espaço em disco mínimo a ser deixado no volume do log de auditoria e o tamanho máximo da trilha de auditoria.
- audit_event: nomes e descrições textuais de eventos de auditoria do sistema e uma lista de quais classes cada evento está.
- audit_user: requisitos de auditoria específicos do usuário a serem combinados com os padrões globais no login.
- audit_warn: um script de shell personalizável usado pelo auditd(8) para gerar mensagens de aviso em situações excepcionais, como quando o espaço para registros de auditoria está baixo ou quando o arquivo de trilha de auditoria foi rotacionado.



Os arquivos de configuração de auditoria devem ser editados e mantidos com cuidado, pois erros na configuração podem resultar no registro inadequado de eventos.

Na maioria dos casos, os administradores precisarão modificar apenas audit_control e audit_user. O primeiro arquivo controla as políticas e as propriedades de auditoria de todo o sistema, e o segundo arquivo pode ser usado para ajustar a auditoria pelo usuário.

16.3.2.1. O arquivo audit_control

Vários padrões para o subsistema de auditoria são especificados em audit_control:

```
dir:/var/audit
dist:off
flags:lo,aa
minfree:5
naflags:lo,aa
policy:cnt,argv
filesz:2M
expire-after:10M
```

A entrada `dir` é usada para definir um ou mais diretórios onde os logs de auditoria serão armazenados. Se mais de uma entrada de diretório aparecer, elas serão usadas em ordem à medida que forem preenchidas. É comum configurar a auditoria para que os logs de auditoria sejam armazenados em um sistema de arquivos dedicado, para evitar a interferência entre o subsistema de auditoria e outros subsistemas, se o sistema de arquivos encher.

Se o campo `dist` estiver definido como `on` ou `yes`, os links físicos serão criados para todos os arquivos de rastreio em /var/audit/dist.

O campo `flags` define a máscara de pré-seleção padrão para todo o sistema para eventos atribuíveis. No exemplo acima, eventos de login/logout bem-sucedidos e com falha, bem como autenticação e autorização, são auditados para todos os usuários.

A entrada `minfree` define a porcentagem mínima de espaço livre para o sistema de arquivos no qual a trilha de auditoria está armazenada.

A entrada `naflags` especifica as classes de auditoria a serem auditadas para eventos não atribuídos, como o processo de login/logout e autenticação e autorização.

A entrada `policy` especifica uma lista separada por vírgula de sinalizadores de política que controla vários aspectos do comportamento de auditoria. O `cnt` indica que o sistema deve continuar em execução apesar de uma falha de auditoria (este sinalizador é altamente recomendado). O outro sinalizador, `argv`, faz com que os argumentos da linha de comando para a chamada de sistema execve(2) sejam auditados como parte de execução de comando.

A entrada `filesz` especifica o tamanho máximo para uma trilha de auditoria antes de finalizar e rotacionar automaticamente o arquivo de trilha. Um valor de `0` desabilita a rotação automática de log. Se o tamanho do arquivo solicitado estiver abaixo do mínimo de 512k, ele será ignorado e uma mensagem de log será gerada.

O campo `expire-after` especifica quando os arquivos de log de auditoria expirarão e serão removidos.

16.3.2.2. O Arquivo audit_user

O administrador pode especificar requisitos adicionais de auditoria para usuários específicos em audit_user. Cada linha configura a auditoria para um usuário através de dois campos: o campo `alwaysaudit` especifica um conjunto de eventos que devem sempre ser auditados para o usuário, e o campo `neveraudit` especifica um conjunto de eventos que nunca devem ser auditados para o usuário.

As entradas de exemplo a seguir auditam os eventos de login/logout e a execução bem-sucedida do comando para `root` e criação de arquivos e execução de comando bem-sucedida para `www`. Se usado com o audit_control, a entrada `lo` para `root` é redundante, e os eventos login/logout também serão auditados para `www`.

```
root:lo,+ex:no
www:fc,+ex:no
```

16.4. Trabalhando com Trilhas de Auditoria

Como as trilhas de auditoria são armazenadas no formato binário BSM, várias ferramentas internas estão disponíveis para modificar ou converter essas trilhas em texto. Para converter arquivos de trilha em um formato de texto simples, use o `praudit`. Para reduzir o arquivo de trilha de auditoria para fins de análise, arquivamento ou impressão, use o `auditreduce`. Esse utilitário suporta vários parâmetros de seleção, incluindo tipo de evento, classe de evento, usuário, data ou hora do evento e o caminho ou objeto do arquivo em questão.

Por exemplo, para baixar todo o conteúdo de um log de auditoria especificado em texto simples:

```
# praudit /var/audit/AUDITFILE
```

Onde *AUDITFILE* é o log de auditoria a ser descarregado.

As trilhas de auditoria consistem em uma série de registros de auditoria compostos por tokens, em que o `praudit` imprime sequencialmente, um por linha. Cada token é de um tipo específico, como `header` (um cabeçalho de registro de auditoria) ou `path` (um caminho de arquivo de uma pesquisa de nome). O seguinte é um exemplo de um evento `execve`:

```
header,133,10,execve(2),0,Mon Sep 25 15:58:03 2006, + 384 msec
exec arg,finger,doug
path,/usr/bin/finger
attribute,555,root,wheel,90,24918,104944
subject,robert,root,wheel,root,wheel,38439,38032,42086,128.232.9.100
return,success,0
trailer,133
```

Esta auditoria representa uma chamada `execve` bem-sucedida, na qual o comando `finger doug` foi executado. O token `exec arg` contém a linha de comando processada apresentada pelo shell ao kernel. O token `path` contém o caminho para o executável conforme procurado pelo kernel. O token `attribute` descreve o binário e inclui o modo de arquivo. O token `subject` armazena o ID do usuário de auditoria, ID do usuário e ID do grupo, ID do usuário real e ID do grupo, ID do processo, ID da sessão, ID da porta e endereço de login. Observe que o ID do usuário de auditoria e o ID do usuário real são diferentes quando o usuário `robert` mudou para a conta `root` antes de executar este comando, mas é auditado usando o usuário original autenticado. O token `return` indica a execução bem-sucedida e o `trailer` conclui o registro.

O formato de saída XML também é suportado e pode ser selecionado incluindo `-x`.

Como os logs de auditoria podem ser muito grandes, um subconjunto de registros pode ser selecionado usando `auditreduce`. Este exemplo seleciona todos os registros de auditoria produzidos para o usuário `trhodes` armazenados em AUDITFILE:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Os membros do grupo `audit` têm permissão para ler trilhas de auditoria em /var/audit. Por padrão, esse grupo está vazio, portanto, apenas o usuário `root` pode ler trilhas de auditoria. Os usuários podem ser adicionados ao grupo `auditoria` para delegar direitos de revisão de auditoria. Como a capacidade de rastrear o conteúdo do log de auditoria fornece informações significativas sobre o comportamento dos usuários e processos, recomenda-se que a delegação dos direitos de revisão de auditoria seja executada com cautela.

16.4.1. Monitoramento em Tempo Real Usando Pipes de Auditoria

Pipes de auditoria são pseudo-dispositivos clones que permitem que os aplicativos acessem o fluxo de registro de auditoria em tempo real. Isto é principalmente de interesse para os autores de aplicações de detecção de intrusão e monitoramento de sistemas. No entanto, o dispositivo de canal de auditoria é uma maneira conveniente para o administrador permitir o monitoramento ao vivo sem incorrer em problemas com a propriedade do arquivo de trilha de auditoria ou a rotação de log interrompendo o fluxo de eventos. Para acompanhar o fluxo de eventos de auditoria em tempo real:

```
# praudit /dev/auditpipe
```

Por padrão, os nós de dispositivo dos pipes de auditoria são acessíveis apenas para o usuário `root`. Para torná-los acessíveis aos membros do grupo `audit`, adicione uma regra `devfs` para /etc/devfs.rules:

```
add path 'auditpipe*' mode 0440 group audit
```

Veja devfs.rules(5) para mais informações sobre como configurar o sistema de arquivos devfs.



É fácil produzir ciclos de feedback de evento de auditoria, nos quais a visualização de cada evento de auditoria resulta na geração de mais eventos de auditoria. Por exemplo, se toda a rede I/O for auditada e `praudit` for executada a partir de uma sessão SSH, um fluxo contínuo de eventos de auditoria será gerada em uma taxa alta, pois cada evento sendo impresso gerará outro evento. Por esse motivo, é aconselhável executar `praudit` em um dispositivo de pipe de auditoria a partir de sessões sem auditoria de I/O de baixa granularidade.

16.4.2. Rotação e Compactação de Arquivos de Trilha de Auditoria

As trilhas de auditoria são gravadas pelo kernel e gerenciadas pelo daemon de auditoria, auditd(8). Os administradores não devem tentar usar o newsyslog.conf(5) ou outras ferramentas para rotacionar diretamente os logs de auditoria. Em vez disso, o `audit` deve ser usado para encerrar a auditoria, reconfigurar o sistema de auditoria e executar a rotação de log. O comando a seguir faz com que o daemon de auditoria crie um novo log de auditoria e sinalize ao kernel para alternar para o novo log. O log antigo será finalizado e renomeado, podendo então ser manipulado pelo administrador:

```
# audit -n
```

Se auditd(8) não estiver em execução no momento, este comando falhará e uma mensagem de erro será apresentada.

Adicionar a seguinte linha ao /etc/crontab agendará essa rotação a cada doze horas:

```
0     */12      *       *       *       root    /usr/sbin/audit -n
```

A alteração terá efeito quando o /etc/crontab for salvo.

A rotação automática do arquivo de trilha de auditoria com base no tamanho do arquivo é possível usando `filesz` em audit_control, conforme descrito em O arquivo audit_control.

Como os arquivos de trilha de auditoria podem se tornar muito grandes, geralmente é desejável compactar ou arquivar rastros depois que eles forem fechados pelo daemon de auditoria. O script audit_warn pode ser usado para executar operações personalizadas para uma variedade de eventos relacionados à auditoria, incluindo a terminação limpa de trilhas de auditoria quando elas são rotacionadas. Por exemplo, o seguinte pode ser adicionado ao /etc/security/audit_warn para compactar as trilhas de auditoria ao serem fechados:

```
#
# Compress audit trail files on close.
#
if [ "$1" = closefile ]; then
        gzip -9 $2
fi
```

Outras atividades de arquivamento podem incluir a cópia de arquivos de trilha para um servidor centralizado, a exclusão de arquivos de trilha antigos ou a redução da trilha de auditoria para remover registros desnecessários. Este script será executado somente quando os arquivos da trilha de auditoria forem finalizados de forma limpa, portanto, não serão executados em trilhas deixadas sem serem eliminadas após um desligamento incorreto.

Capítulo 17. Armazenamento

17.1. Sinopse

Este capítulo aborda o uso de discos e mídia de armazenamento no FreeBSD. Isso inclui discos SCSI e IDE, mídias de CD e DVD, discos com suporte de memória e dispositivos de armazenamento USB.

Depois de ler este capítulo, você saberá:

- Como adicionar discos rígidos adicionais a um sistema FreeBSD.
- Como aumentar o tamanho da partição de um disco no FreeBSD.
- Como configurar o FreeBSD para usar dispositivos de armazenamento USB.
- Como usar mídias de CD e DVD em um sistema FreeBSD.
- Como usar os programas de backup disponíveis no FreeBSD.
- Como configurar discos de memória.
- O que são snapshots de sistema de arquivos e como usá-los com eficiência.
- Como usar cotas para limitar o uso de espaço em disco.
- Como criptografar discos e swap para protegê-los contra invasores.
- Como configurar uma rede de armazenamento altamente disponível.

Antes de ler este capítulo, você deve:

- Saiba como configurar e instalar um novo kernel do FreeBSD.

17.2. Adicionando Discos

Esta seção descreve como adicionar um novo disco SATA a uma máquina que atualmente possui apenas uma única unidade. Primeiro, desligue o computador e instale a unidade no computador seguindo as instruções do fabricante do computador, controladora e unidade. Reinicialize o sistema e torne-se `root`.

Inspecione o arquivo /var/run/dmesg.boot para garantir que o novo disco foi encontrado. Neste exemplo, a unidade SATA recém-adicionada aparecerá como ada1.

Para este exemplo, uma única partição grande será criada no novo disco. O esquema de particionamento GPT será usado ao invés do esquema MBR, mais antigo e menos versátil.



Se o disco a ser adicionado não estiver em branco, as informações antigas da partição podem ser removidas com `gpart delete`. Veja gpart(8) para detalhes.

O esquema de partição é criado e, em seguida, uma única partição é adicionada. Para melhorar o desempenho em discos mais recentes com tamanhos maiores de blocos de hardware, a partição está alinhada a divisões de um megabyte:

```
# gpart create -s GPT ada1
# gpart add -t freebsd-ufs -a 1M ada1
```

Dependendo do uso, várias partições menores podem ser desejadas. Veja gpart(8) para opções para criar partições menores que um disco inteiro.

As informações da partição de disco podem ser visualizadas com `gpart show`:

```
% gpart show ada1
=>        34  1465146988  ada1  GPT  (699G)
          34        2014        - free -  (1.0M)
        2048  1465143296     1  freebsd-ufs  (699G)
  1465145344        1678        - free -  (839K)
```

Um sistema de arquivos é criado em uma nova partição no novo disco:

```
# newfs -U /dev/ada1p1
```

Um diretório vazio é criado como um *ponto de montagem*, um local para montar o novo disco no sistema de arquivos do disco original:

```
# mkdir /newdisk
```

Finalmente, uma entrada é adicionada ao arquivo /etc/fstab para que o novo disco seja montado automaticamente na inicialização:

```
/dev/ada1p1 /newdisk    ufs rw  2   2
```

O novo disco pode ser montado manualmente, sem reiniciar o sistema:

```
# mount /newdisk
```

17.3. Redimensionando e Ampliando Discos

A capacidade de um disco pode aumentar sem alterações nos dados já presentes. Isso acontece normalmente com máquinas virtuais, quando o disco virtual torna-se muito pequeno e é ampliado. Às vezes, uma imagem de disco é gravada em um cartão de memória USB, mas não usa toda a capacidade. Aqui nós descrevemos como redimensionar ou *ampliar* o conteúdo do disco para aproveitar a capacidade aumentada.

Determine o nome do dispositivo do disco a ser redimensionado inspecionando o arquivo /var/run/dmesg.boot. Neste exemplo, há apenas um disco SATA no sistema, portanto a unidade

aparecerá como ada0.

Liste as partições no disco para ver a configuração atual:

```
# gpart show ada0
=>      34  83886013  ada0  GPT  (48G) [CORRUPT]
        34       128     1  freebsd-boot  (64k)
       162  79691648     2  freebsd-ufs  (38G)
  79691810   4194236     3  freebsd-swap  (2G)
  83886046         1        - free -  (512B)
```



Se o disco foi formatado com o esquema de particionamento GPT, ele pode ser exibido como "corrompido" porque a tabela de partições de backup GPT não está mais no final da unidade. Corrija a tabela de partições de backup com o `gpart`:

```
# gpart recover ada0
ada0 recovered
```

Agora, o espaço adicional no disco está disponível para uso por uma nova partição ou uma partição existente pode ser expandida:

```
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   79691648     2  freebsd-ufs  (38G)
   79691810    4194236     3  freebsd-swap  (2G)
   83886046   18513921        - free -  (8.8G)
```

As partições só podem ser redimensionadas para um espaço livre contíguo. Aqui, a última partição no disco é a partição swap, mas a segunda partição é aquela que precisa ser redimensionada. As partições de Swap contêm apenas dados temporários, portanto, podem ser desmontadas, excluídas e, em seguida, recriadas a terceira partição após redimensionar a segunda partição.

Desative a partição de swap:

```
# swapoff /dev/ada0p3
```

Exclua a terceira partição, especificada pela flag `-i`, do disco *ada0*.

```
# gpart delete -i 3 ada0
ada0p3 deleted
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   79691648     2  freebsd-ufs  (38G)
```

```
   79691810   22708157        - free -  (10G)
```



Existe o risco de perda de dados ao modificar a tabela de partições de um sistema de arquivos montado. É melhor executar as etapas a seguir em um sistema de arquivos desmontado durante a execução de um dispositivo CD-ROM ou USB live. No entanto, se for absolutamente necessário, um sistema de arquivos montado pode ser redimensionado depois de desativar os recursos de segurança do GEOM:

```
# sysctl kern.geom.debugflags=16
```

Redimensione a partição, deixando espaço para recriar uma partição swap do tamanho desejado. A partição a ser redimensionada é especificada com `-i` e o novo tamanho desejado com `-s`. Opcionalmente, o alinhamento da partição é controlado com `-a`. Isso só modifica o tamanho da partição. O sistema de arquivos na partição será expandido em uma etapa separada.

```
# gpart resize -i 2 -s 47G -a 4k ada0
ada0p2 resized
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   98566144     2  freebsd-ufs  (47G)
   98566306    3833661        - free -  (1.8G)
```

Recrie a partição swap e ative-a. Se nenhum tamanho for especificado com `-s`, todo o espaço restante será usado:

```
# gpart add -t freebsd-swap -a 4k ada0
ada0p3 added
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   98566144     2  freebsd-ufs  (47G)
   98566306    3833661     3  freebsd-swap  (1.8G)
# swapon /dev/ada0p3
```

Aumente o sistema de arquivos UFS para usar a nova capacidade da partição redimensionada:

```
# growfs /dev/ada0p2
Device is mounted read-write; resizing will result in temporary write suspension for
/.
It's strongly recommended to make a backup before growing the file system.
OK to grow file system on /dev/ada0p2, mounted on /, from 38GB to 47GB? [Yes/No] Yes
super-block backups (for fsck -b #) at:
 80781312, 82063552, 83345792, 84628032, 85910272, 87192512, 88474752,
```

```
 89756992, 91039232, 92321472, 93603712, 94885952, 96168192, 97450432
```

Se o sistema de arquivos for ZFS, o redimensionamento será acionado pela execução do subcomando `online` com `-e`:

```
# zpool online -e zroot /dev/ada0p2
```

Tanto a partição quanto o sistema de arquivos foram redimensionados para usar o espaço em disco recém-disponível.

17.4. Dispositivos de Armazenamento USB

Muitas soluções de armazenamento externo, como discos rígidos, thumbdrives USB e gravadores de CD e DVD, usam o Universal Serial Bus (USB). O FreeBSD fornece suporte para dispositivos USB 1.x, 2.0 e 3.0.



O suporte a USB 3.0 não é compatível com alguns hardwares, incluindo os chipsets Haswell (Lynx Point). Se o FreeBSD inicializar com uma mensagem `falhou com erro 19`, desative xHCI/USB3 na BIOS.

O suporte para dispositivos de armazenamento USB é embutido no kernel GENERIC. Para um kernel personalizado, certifique-se de que as seguintes linhas estejam presentes no arquivo de configuração do kernel:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device pass # Passthrough device (direct ATA/SCSI access)
device uhci # provides USB 1.x support
device ohci # provides USB 1.x support
device ehci # provides USB 2.0 support
device xhci # provides USB 3.0 support
device usb  # USB Bus (required)
device umass    # Disks/Mass storage - Requires scbus and da
device cd   # needed for CD and DVD burners
```

O FreeBSD usa o driver umass(4) que usa o subsistema SCSI para acessar o armazenamento de dispositivos USB. Como qualquer dispositivo USB será visto como um dispositivo SCSI pelo sistema, se o dispositivo USB for um gravador de CD ou DVD, *não* inclua `device atapicam` em um arquivo de configuração do kernel personalizado.

O restante desta seção demonstra como verificar se um dispositivo de armazenamento USB é reconhecido pelo FreeBSD e como configurar o dispositivo para que ele possa ser usado.

17.4.1. Configuração de Dispositivo

Para testar a configuração USB, conecte o dispositivo USB. Use `dmesg` para confirmar que a unidade

aparece no buffer de mensagens do sistema. Deve parecer algo como isto:

```
umass0: <STECH Simple Drive, class 0/0, rev 2.00/1.04, addr 3> on usbus0
umass0:  SCSI over Bulk-Only; quirks = 0x0100
umass0:4:0:-1: Attached to scbus4
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
da0: <STECH Simple Drive 1.04> Fixed Direct Access SCSI-4 device
da0: Serial Number WD-WXE508CAN263
da0: 40.000MB/s transfers
da0: 152627MB (312581808 512 byte sectors: 255H 63S/T 19457C)
da0: quirks=0x2<NO_6_BYTE>
```

A marca, o nó de dispositivo (da0), a velocidade e o tamanho serão diferentes de acordo com o dispositivo.

Como o dispositivo USB é visto como um SCSI, o `camcontrol` pode ser usado para listar os dispositivos de armazenamento USB conectados ao sistema:

```
# camcontrol devlist
<STECH Simple Drive 1.04>          at scbus4 target 0 lun 0 (pass3,da0)
```

Alternativamente, o `usbconfig` pode ser usado para listar o dispositivo. Consulte o usbconfig(8) para obter mais informações sobre este comando.

```
# usbconfig
ugen0.3: <Simple Drive STECH> at usbus0, cfg=0 md=HOST spd=HIGH (480Mbps) pwr=ON (2mA)
```

Se o dispositivo não tiver sido formatado, consulte Adicionando Discos para obter instruções sobre como formatar e criar partições na unidade USB. Se a unidade vier com um sistema de arquivos, ela pode ser montada pelo `root` usando as instruções em Montando e Desmontando Sistemas de Arquivos.



Permitir que usuários não confiáveis montem mídia arbitrária, ativando `vfs.usermount` como descrito abaixo, não deve ser considerado seguro do ponto de vista da segurança. A maioria dos sistemas de arquivos não foi criada para proteger contra dispositivos maliciosos.

Para tornar o dispositivo montável como um usuário normal, uma solução é tornar todos os usuários do dispositivo membros do grupo `operator` usando pw(8). Em seguida, certifique-se de que `operator` possa ler e gravar o dispositivo adicionando estas linhas ao /etc/devfs.rules:

```
[localrules=5]
add path 'da*' mode 0660 group operator
```



Se discos internos SCSI também estiverem instalados no sistema, altere a segunda

linha da seguinte maneira:

```
add path 'da[3-9]*' mode 0660 group operator
```

Isso excluirá os três primeiros discos SCSI (da0 para da2) pertencentes ao grupo `operator`. Substitua *3* pelo número de discos SCSI internos. Consulte devfs.rules(5) para obter mais informações sobre esse arquivo.

Em seguida, ative o conjunto de regras no arquivo /etc/rc.conf:

```
devfs_system_ruleset="localrules"
```

Em seguida, instrua o sistema para permitir que usuários comuns montem sistemas de arquivos incluindo a seguinte linha no arquivo /etc/sysctl.conf:

```
vfs.usermount=1
```

Como isso só entra em vigor após a próxima reinicialização, use `sysctl` para definir essa variável agora:

```
# sysctl vfs.usermount=1
vfs.usermount: 0 -> 1
```

A etapa final é criar um diretório no qual o sistema de arquivos deve ser montado. Esse diretório precisa pertencer ao usuário que deve montar o sistema de arquivos. Uma maneira de fazer isso é para o `root` criar um subdiretório de propriedade daquele usuário como /mnt/username. No exemplo a seguir, substitua *username* pelo nome de login do usuário e *usergroup* pelo grupo principal do usuário:

```
# mkdir /mnt/username
# chown username:usergroup /mnt/username
```

Suponha que um thumbdrive USB esteja conectado e um dispositivo /dev/da0s1 apareça. Se o dispositivo estiver formatado com um sistema de arquivos FAT, o usuário poderá montá-lo usando:

```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/username
```

Antes que o dispositivo possa ser desconectado, ele *deve* ser desmontado primeiro:

```
% umount /mnt/username
```

Após a remoção do dispositivo, o buffer de mensagens do sistema mostrará mensagens semelhantes

às seguintes:

```
umass0: at uhub3, port 2, addr 3 (disconnected)
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
da0: <STECH Simple Drive 1.04> s/n WD-WXE508CAN263          detached
(da0:umass-sim0:0:0:0): Periph destroyed
```

17.4.2. Montando Automaticamente Uma Mídia Removível

Dispositivos USB podem ser montados automaticamente removendo o comentário desta linha no arquivo /etc/auto_master:

```
/media      -media      -nosuid
```

Então adicione estas linhas ao arquivo /etc/devd.conf:

```
notify 100 {
    match "system" "GEOM";
    match "subsystem" "DEV";
    action "/usr/sbin/automount -c";
};
```

Recarregue a configuração se autofs(5) e devd(8) já estiverem em execução:

```
# service automount restart
# service devd restart
```

autofs(5) pode ser configurado para iniciar no boot, adicionando esta linha ao arquivo /etc/rc.conf:

```
autofs_enable="YES"
```

autofs(5) requer que o devd(8) esteja ativado, como é por padrão.

Inicie os serviços imediatamente com:

```
# service automount start
# service automountd start
# service autounmountd start
# service devd start
```

Cada sistema de arquivos que pode ser montado automaticamente aparece como um diretório em /media/. O diretório é nomeado após o rótulo do sistema de arquivos. Se o rótulo estiver ausente, o diretório será nomeado após o nó do dispositivo.

O sistema de arquivos é montado de forma transparente no primeiro acesso e desmontado após um período de inatividade. Unidades montadas automaticamente também podem ser desmontadas manualmente:

```
# automount -fu
```

Este mecanismo é normalmente usado para cartões de memória e cartões de memória USB. Pode ser usado com qualquer dispositivo de bloco, incluindo unidades ópticas ou iSCSILUNs.

17.5. Criando e Usando Mídia em CD

A mídia em disco compacto (CD) fornece vários recursos que os diferenciam dos discos convencionais. Eles são projetados para que possam ser lidos continuamente sem atrasos para mover a cabeça entre as trilhas. Embora a mídia CD tenha faixas, elas se referem a uma seção de dados a ser lida continuamente e não a uma propriedade física do disco. O sistema de arquivos ISO 9660 foi projetado para lidar com essas diferenças.

A Coleção de Ports do FreeBSD fornece vários utilitários para gravar e duplicar áudio e dados de CDs. Este capítulo demonstra o uso de vários utilitários de linha de comando. Para o software de gravação de CD com um utilitário gráfico, considere instalar os pacotes ou ports sysutils/xcdroast ou sysutils/k3b.

17.5.1. Dispositivos Suportados

O kernel GENERIC fornece suporte para SCSI, USB, e leitores e gravadores de CDATAPI. Se um kernel personalizado for usado, as opções que precisam estar presentes no arquivo de configuração do kernel variam de acordo com o tipo de dispositivo.

Para um gravador SCSI, verifique se essas opções estão presentes:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device pass # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
```

Para um gravador de USB, verifique se essas opções estão presentes:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device pass # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
device uhci # provides USB 1.x support
device ohci # provides USB 1.x support
device ehci # provides USB 2.0 support
device xhci # provides USB 3.0 support
device usb  # USB Bus (required)
```

```
device umass    # Disks/Mass storage - Requires scbus and da
```

Para um gravador ATAPI, verifique se essas opções estão presentes:

```
device ata  # Legacy ATA/SATA controllers
device scbus    # SCSI bus (required for ATA/SCSI)
device pass # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
```



Nas versões do FreeBSD anteriores a 10.x, esta linha também é necessária no arquivo de configuração do kernel se o gravador for um dispositivo ATAPI:

```
device atapicam
```

Como alternativa, esse driver pode ser carregado no momento da inicialização adicionando a seguinte linha ao arquivo /boot/loader.conf:

```
atapicam_load="YES"
```

Isso exigirá uma reinicialização do sistema, pois esse driver só pode ser carregado no momento da inicialização.

Para verificar se o FreeBSD reconhece o dispositivo, execute o `dmesg` e procure por uma entrada para o dispositivo. Nos sistemas anteriores a 10.x, o nome do dispositivo na primeira linha da saída será acd0 em vez de cd0.

```
% dmesg | grep cd
cd0 at ahcich1 bus 0 scbus1 target 0 lun 0
cd0: <HL-DT-ST DVDRAM GU70N LT20> Removable CD-ROM SCSI-0 device
cd0: Serial Number M3OD3S34152
cd0: 150.000MB/s transfers (SATA 1.x, UDMA6, ATAPI 12bytes, PIO 8192bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

17.5.2. Gravando um CD

No FreeBSD, `cdrecord` pode ser usado para gravar CDs. Este comando é instalado com o pacote ou port sysutils/cdrtools.

Enquanto o `cdrecord` tem muitas opções, o uso básico é simples. Especifique o nome do arquivo ISO para gravar e, se o sistema tiver vários dispositivos de gravação, especifique o nome do dispositivo a ser usado:

```
# cdrecord dev=device imagefile.iso
```

Para determinar o nome do dispositivo do gravador, use `-scanbus`, que pode produzir resultados como este:

```
# cdrecord -scanbus
ProDVD-ProBD-Clone 3.00 (amd64-unknown-freebsd10.0) Copyright (C) 1995-2010 Jörg
Schilling
Using libscg version 'schily-0.9'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) 'YAMAHA  ' 'CRW4260        ' '1.0q' Removable CD-ROM
        1,6,0   106) 'ARTEC   ' 'AM12S          ' '1.06' Scanner
        1,7,0   107) *
```

Localize a entrada para o gravador de CD e use os três números separados por vírgulas como o valor para `dev`. Nesse caso, o dispositivo gravador Yamaha é `1,5,0`, portanto, a entrada apropriada para especificar esse dispositivo é `dev=1,5,0`. Consulte a página de manual do `cdrecord` para outras formas de especificar este valor e informações sobre como gravar faixas de áudio e controlar a velocidade de gravação.

Como alternativa, execute o seguinte comando para obter o endereço do dispositivo do gravador:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 (cd0,pass0)
```

Use os valores numéricos para `scbus`, `target` e `lun`. Para este exemplo, `1,0,0` é o nome do dispositivo a ser usado.

17.5.3. Escrevendo Dados em um Sistema de Arquivos ISO

Para produzir um CD de dados, os arquivos de dados que compõem as faixas no CD devem ser preparados antes que possam ser gravados no CD. No FreeBSD, sysutils/cdrtools instala o `mkisofs`, que pode ser usado para produzir um sistema de arquivos ISO 9660 que é uma imagem de uma árvore de diretórios dentro um sistema de arquivos UNIX™. O uso mais simples é especificar o nome do arquivo ISO para criar e o caminho para os arquivos a serem colocados no sistema de arquivos ISO 9660:

```
# mkisofs -o imagefile.iso /path/to/tree
```

Este comando mapeia os nomes dos arquivos no caminho especificado para os nomes que se ajustam às limitações do sistema de arquivos padrão ISO 9660 e excluirá arquivos que não atendem ao padrão para o sistemas de arquivos ISO.

Várias opções estão disponíveis para superar as restrições impostas pelo padrão. Em particular, `-R` permite que as extensões Rock Ridge comuns aos sistemas UNIX™ e `-J` ativem as extensões Joliet usadas por sistemas Microsoft™.

Para CDs que serão usados apenas em sistemas FreeBSD, `-U` pode ser usado para desabilitar todas as restrições de nome de arquivo. Quando usado com `-R`, ele produz uma imagem do sistema de arquivos que é idêntica à árvore FreeBSD especificada, mesmo se violar o padrão ISO 9660.

A última opção de uso geral é `-b`. Isso é usado para especificar a localização de uma imagem de inicialização para uso na produção de um CD inicializável "El Torito". Essa opção usa um argumento que é o caminho para uma imagem de inicialização a partir do topo da árvore que está sendo gravada no CD. Por padrão, o `mkisofs` cria uma imagem ISO no modo de "emulação de disquete" e, portanto, espera que a imagem de inicialização tenha exatamente 1200, 1440 ou 2880 KB de tamanho. Alguns gerenciadores de inicialização, como o usado pela mídia de distribuição do FreeBSD, não utilizam o modo de emulação. Nesse caso, `-no-emul-boot` deve ser usado. Então, se /tmp/myboot possuir um sistema FreeBSD inicializável com a imagem de inicialização em /tmp/myboot/boot/cdboot, este comando produziria /tmp/bootable.iso:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

A imagem ISO resultante pode ser montada como um disco de memória com:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Pode-se então verificar se /mnt e /tmp/myboot são idênticos.

Existem muitas outras opções disponíveis para `mkisofs` para ajustar seu comportamento. Consulte mkisofs(8) para obter detalhes.



É possível copiar um CD de dados para um arquivo de imagem que seja funcionalmente equivalente ao arquivo de imagem criado com `mkisofs`. Para fazer isso, use dd com o nome do dispositivo como o arquivo de entrada e o nome do ISO para criar como o arquivo de saída:

```
# dd if=/dev/cd0 of=file.iso bs=2048
```

O arquivo de imagem resultante pode ser gravado em CD, conforme descrito em Gravando um CD.

17.5.4. Usando CDs de Dados

Uma vez que uma ISO tenha sido gravada em um CD, ela pode ser montada especificando o tipo de sistema de arquivos, o nome do dispositivo que contém o CD e um ponto de montagem existente:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Como `mount` assume que um sistema de arquivos é do tipo `ufs`, um erro `Incorrect super block` ocorrerá se `-t cd9660` não está incluído ao montar um arquivo de dados CD.

Embora qualquer CD de dados possa ser montado dessa forma, discos com determinadas extensões ISO 9660 podem se comportar de maneira estranha. Por exemplo, os discos Joliet armazenam todos os nomes de arquivos em caracteres Unicode de dois bytes. Se alguns caracteres não ingleses aparecerem como pontos de interrogação, especifique o conjunto de caracteres local com `-C`. Para mais informações, consulte mount_cd9660(8).



Para fazer esta conversão de caracteres com a ajuda de `-C`, o kernel requer que o módulo cd9660_iconv.ko seja carregado. Isto pode ser feito adicionando esta linha ao arquivo loader.conf:

```
cd9660_iconv_load="YES"
```

e reinicializando a máquina, ou carregando diretamente o módulo com `kldload`.

Ocasionalmente, `Device not configured` será exibido ao tentar montar um CD de dados. Isso geralmente significa que a unidade de CD não detectou um disco na bandeja ou que a unidade não está visível no barramento. Pode levar alguns segundos para que uma unidade de CD detecte a mídia, por isso, seja paciente.

Às vezes, uma unidade de CDSCSI pode ser perdida porque não teve tempo suficiente para responder à reinicialização do barramento. Para resolver isso, um kernel personalizado pode ser criado, o que aumenta o delay SCSI padrão. Adicione a seguinte opção ao arquivo de configuração do kernel personalizado e reconstrua o kernel usando as instruções em Criando e Instalando um Kernel Customizado:

```
options SCSI_DELAY=15000
```

Isso faz com que o barramento SCSI faça uma pausa de 15 segundos durante a inicialização, para dar à unidade de CD todas as chances possíveis de responder à reinicialização do barramento.



É possível gravar um arquivo diretamente no CD, sem criar um sistema de arquivos ISO 9660. Isso é conhecido como gravação de dados brutos em CD e algumas pessoas fazem isso para fins de backup.

Este tipo de disco não pode ser montado como um CD de dados normal. Para recuperar os dados gravados em um CD, os dados devem ser lidos no nó do

dispositivo bruto. Por exemplo, este comando irá extrair um arquivo tar compactado localizado no segundo dispositivo de CD para o diretório de trabalho atual:

```
# tar xzvf /dev/cd1
```

Para montar um CD de dados, os dados devem ser escritos usando `mkisofs`.

17.5.5. Duplicando CDs de Áudio

Para duplicar um CD de áudio, extraia os dados de áudio do CD para uma série de arquivos e, em seguida, grave esses arquivos em um CD em branco.

Duplicando um CD de Áudio descreve como duplicar e gravar um CD de áudio. Se a versão do FreeBSD for menor que 10.0 e o dispositivo for ATAPI, o módulo `atapicam` deve ser carregado primeiro usando as instruções em Dispositivos Suportados.

Procedure: Duplicando um CD de Áudio

1. O pacote ou port sysutils/cdrtools instala o `cdda2wav`. Este comando pode ser usado para extrair todas as faixas de áudio, com cada faixa gravada em um arquivo WAV separado no diretório de trabalho atual:

   ```
   % cdda2wav -vall -B -Owav
   ```

   Um nome de dispositivo não precisa ser especificado se houver apenas um dispositivo de CD no sistema. Consulte a página de manual `cdda2wav` para obter instruções sobre como especificar um dispositivo e aprender mais sobre as outras opções disponíveis para este comando.

2. Use o `cdrecord` para escrever os arquivos .wav:

   ```
   % cdrecord -v dev=2,0 -dao -useinfo  *.wav
   ```

   Certifique-se de que *2,0* esteja configurado adequadamente, conforme descrito em Gravando um CD.

17.6. Criando e Usando Mídia de DVD

Comparado ao CD, o DVD é a próxima geração de tecnologia de armazenamento de mídia ótica. O DVD pode conter mais dados do que qualquer CD e é o padrão para publicação de vídeos.

Cinco formatos graváveis físicos podem ser definidos para um DVD gravável:

- DVD-R: Este foi o primeiro formato gravável disponível em DVD. O padrão DVD-R é definido

pelo DVD Forum. Este formato é escrito uma vez.

- DVD-RW: Esta é a versão regravável do padrão DVD-R. Um DVD-RW pode ser reescrito cerca de 1000 vezes.
- DVD-RAM: Este é um formato regravável que pode ser visto como um disco rígido removível. No entanto, esta mídia não é compatível com a maioria das unidades e reprodutores de DVD-Video DVD-ROM, pois apenas alguns gravadores de DVD suportam o formato DVD-RAM. Consulte Usando um DVD-RAM para mais informações sobre o uso de DVD-RAM.
- DVD+RW: Este é um formato regravável definido pelo DVD+RW Alliance. Um DVD+RW pode ser reescrito cerca de 1000 vezes.
- DVD+R: Este formato é a variação de gravação do formato DVD+RW.

Um DVD gravável de camada única pode armazenar até 4,700,000,000 bytes, o que é, na verdade, 4.38 GB ou 4485 MB, pois 1 kilobyte é 1024 bytes.



Uma distinção deve ser feita entre a mídia física e a aplicação. Por exemplo, um DVD-Vídeo é um layout de arquivo específico que pode ser gravado em qualquer mídia física DVD gravável, como DVD-R, DVD+R ou DVD-RW. Antes de escolher o tipo de mídia, verifique se o gravador e o reprodutor de DVD-Video são compatíveis com a mídia em questão.

17.6.1. Configuração

Para executar a gravação de um DVD, use growisofs(1). Este comando é parte dos utilitários sysutils/dvd+rw-tools que suportam todos os tipos de mídia DVD.

Estas ferramentas usam o subsistema SCSI para acessar os dispositivos, portanto suporte a ATAPI/CAM deve ser carregado ou estaticamente compilado no kernel. Este suporte não é necessário se o gravador usar a interface USB. Consulte Dispositivos de Armazenamento USB para mais detalhes sobre a configuração do dispositivo USB.

O acesso DMA também deve estar ativado para dispositivos ATAPI, adicionando a seguinte linha ao arquivo /boot/loader.conf:

```
hw.ata.atapi_dma="1"
```

Antes de tentar usar dvd+rw-tools, consulte o Notas de compatibilidade de hardware.



Para uma interface gráfica de usuário, considere o uso de sysutils/k3b que fornece uma interface amigável para growisofs(1) e muitas outras ferramentas de gravação.

17.6.2. Gravando DVDs de Dados

Já que growisofs(1) é um front-end para mkisofs, ele invocará mkisofs(8) para criar o layout do sistema de arquivos e executar a gravação no DVD . Isso significa que uma imagem dos dados não precisa ser criada antes do processo de gravação.

Para gravar em um DVD+R ou DVD-R os dados em /path/to/data, use o seguinte comando:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

Neste exemplo, `-J -R` é passado para mkisofs(8) para criar um sistemas de arquivos ISO 9660 com extensões Joliet e Rock Ridge. Consulte o mkisofs(8) para obter mais detalhes.

Para a gravação inicial da sessão, `-Z` é usado para sessões únicas e múltiplas. Substitua */dev/cd0*, com o nome do dispositivo de DVD. O uso de `-dvd-compat` indica que o disco será fechado e que a gravação será inaplicável. Isso também deve fornecer melhor compatibilidade de mídia com unidades DVD-ROM.

Para gravar uma imagem pré-masterizada, como *imagefile.iso*, use:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

A velocidade de gravação deve ser detectada e configurada automaticamente de acordo com a mídia e a unidade que está sendo usada. Para forçar a velocidade de gravação, use `-speed=`. Consulte o growisofs(1) para exemplos de uso.



Para suportar arquivos de trabalho maiores que 4.38GB, um sistema de arquivos híbrido UDF/ISO-9660 deve ser criado passando `-udf -iso-level 3` para mkisofs(8) e todos os programas relacionados, como growisofs(1). Isso é necessário apenas ao criar um arquivo de imagem ISO ou ao gravar arquivos diretamente em um disco. Como um disco criado dessa maneira deve ser montado como um sistema de arquivos UDF com mount_udf(8), ele será utilizável apenas em um sistema operacional com suporte a UDF. Caso contrário, parecerá que contém arquivos corrompidos.

Para criar este tipo de arquivo ISO:

```
% mkisofs -R -J -udf -iso-level 3 -o imagefile.iso /path/to/data
```

Para gravar arquivos diretamente em um disco:

```
# growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R
/path/to/data
```

Quando uma imagem ISO já contém arquivos grandes, nenhuma opção adicional é necessária para o growisofs(1) gravar a imagem em um disco.

Certifique-se de usar uma versão atualizada do port sysutils/cdrtools, que contenha o mkisofs(8), como uma versão mais antiga pode não conter suporte a arquivos grandes. Se a versão mais recente não funcionar, instale o sysutils/cdrtools-devel e leia o mkisofs(8).

17.6.3. Gravando um DVD -Video

Um DVD-Video é um layout de arquivo específico baseado nas especificações ISO 9660 e micro-UDF (M-UDF). Como o DVD-Video apresenta uma hierarquia de estrutura de dados específica, um programa específico como multimedia/dvdauthor é necessário para criar o DVD.

Se uma imagem do sistema de arquivos DVD-Video já existir, ela poderá ser gravada da mesma maneira que qualquer outra imagem. Se o `dvdauthor` foi usado para criar o DVD e o resultado está em /path/to/video, o seguinte comando deve ser usado para gravar o DVD-Vídeo:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

`-dvd-video` é passado para o mkisofs(8) para instruí-lo a criar um sistemas de arquivos com layout DVD-Video. Esta opção implica na opção `-dvd-compat` do growisofs(1).

17.6.4. Usando um DVD+RW

Ao contrário do CD-RW, um DVD+RW virgem precisa ser formatado antes do primeiro uso. É *recomendado* para permitir que growisofs(1) cuide disso automaticamente sempre que apropriado. No entanto, é possível usar `dvd+rw-format` para formatar o DVD+RW:

```
# dvd+rw-format /dev/cd0
```

Somente execute esta operação uma vez e tenha em mente que apenas mídias DVD+RW virgens precisam ser formatadas. Uma vez formatado, o DVD+RW pode ser gravado como de costume.

Para gravar um sistema de arquivos totalmente novo e não apenas acrescentar alguns dados em um DVD+RW, a mídia não precisa ser apagada primeiro. Em vez disso, escreva sobre a gravação anterior assim:

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

O formato DVD+RW suporta anexar dados a uma gravação anterior. Essa operação consiste em mesclar uma nova sessão à existente, pois ela não é considerada como gravação de várias sessões. growisofs(1) vai *ampliar* o sistema de arquivos ISO 9660 presente na mídia.

Por exemplo, para anexar dados a um DVD+RW, use o seguinte:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

As mesmas opções do mkisofs(8) usadas para gravar a sessão inicial devem ser usadas durante as próximas gravações.



Use `-dvd-compat` para melhor compatibilidade de mídia com as unidades de DVD-ROM. Ao usar DVD+RW, essa opção não impedirá a adição de dados.

Para apagar a mídia, use:

```
# growisofs -Z /dev/cd0=/dev/zero
```

17.6.5. Usando um DVD-RW

Um DVD-RW aceita dois formatos de disco: sequencial incremental e substituição restrita. Por padrão, os discos DVD-RW estão em formato sequencial.

Um DVD-RW virgem pode ser escrito diretamente sem ser formatado. No entanto, um DVD-RW não-virgem em formato sequencial precisa ser apagado antes de escrever uma nova sessão inicial.

Para apagar um DVD-RW em modo sequencial:

```
# dvd+rw-format -blank=full /dev/cd0
```



Um preenchimento completo usando `-blank=full` levará cerca de uma hora em uma mídia 1x. Um limpeza rápida pode ser executada usando `-blank`, se o DVD-RW for gravado no modo Disk-At-Once (DAO). Para gravar o DVD-RW no modo DAO, use o comando:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Como o growisofs(1) tenta automaticamente detectar a mídia rapidamente em branco e ativar a gravação do DAO, `-use-the-force -luke=dao` não deve ser requerido.

Em vez disso, deve-se usar o modo de sobrescrita restrita com qualquer DVD-RW, pois esse formato é mais flexível do que o padrão de sequencial incremental.

Para escrever dados em um DVD-RW seqüencial, use as mesmas instruções que para os outros formatos de DVD:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

Para acrescentar alguns dados a uma gravação anterior, use `-M` com o growisofs(1). No entanto, se os dados forem anexados em um DVD-RW no modo sequencial incremental, uma nova sessão será criada no disco e o resultado será um disco multi-sessão.

Um DVD-RW no formato de sobrescrita restrita não precisa ser em apagado antes de uma nova sessão inicial. Em vez disso, sobrescreva o disco com `-Z`. Também é possível aumentar um sistema de arquivos ISO 9660 existente escrito no disco com `-M`. O resultado será um DVD de uma sessão.

Para colocar um DVD-RW no formato de sobrescrita restrita, o seguinte comando deve ser usado:

```
# dvd+rw-format /dev/cd0
```

Para voltar ao formato sequencial, use:

```
# dvd+rw-format -blank=full /dev/cd0
```

17.6.6. Multi-Sessão

Poucas unidades de DVD-ROM suportam DVDs multi-sessão e na maioria das vezes apenas lêem a primeira sessão. DVD+R, DVD-R e DVD-RW em formato sequencial podem aceitar várias sessões. A noção de várias sessões não existe para os formatos de sobrescrita restrita DVD+RW e DVD-RW.

Usando o seguinte comando após uma sessão inicial não fechada em um DVD+R, DVD-R ou DVD-RW em formato sequencial, será adicionada uma nova sessão ao disco:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Usando este comando com um DVD+RW ou um DVD-RW no modo de sobrescrita restrita adicionará dados ao mesclar a nova sessão à existente. O resultado será um disco de sessão única. Use este método para adicionar dados após uma gravação inicial nesses tipos de mídia.



Como algum espaço na mídia é usado entre cada sessão para marcar o final e o início das sessões, deve-se adicionar sessões com uma grande quantidade de dados para otimizar o espaço da mídia. O número de sessões é limitado a 154 para um DVD+R, cerca de 2000 para um DVD-R e 127 para um DVD+R Double Layer.

17.6.7. Para Maiores Informações

Para obter mais informações sobre um DVD, use o `dvd+rw-mediainfo /dev/cd0` enquanto o disco estiver na unidade especificada.

Mais informações sobre dvd+rw-tools podem ser encontradas em growisofs(1), no site de dvd+rw-tools, e nos arquivos do cdwrite mailing list.



Ao criar um relatório de problemas relacionado ao uso de dvd+rw-tools, inclua sempre a saída de `dvd+rw-mediainfo`.

17.6.8. Usando um DVD-RAM

Os gravadores de DVD-RAM podem usar uma interface SCSI ou ATAPI. Para dispositivos ATAPI, o acesso DMA deve ser ativado adicionando a seguinte linha ao arquivo /boot/loader.conf:

```
hw.ata.atapi_dma="1"
```

Um DVD-RAM pode ser visto como um disco rígido removível. Como qualquer outro disco rígido, o DVD-RAM deve ser formatado antes de poder ser usado. Neste exemplo, todo o espaço em disco será formatado com um sistema de arquivos UFS2 padrão:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

O dispositivo DVD, acd0, deve ser alterado de acordo com a configuração.

Uma vez que o DVD-RAM tenha sido formatado, ele pode ser montado como um disco rígido normal:

```
# mount /dev/acd0 /mnt
```

Uma vez montado, o DVD-RAM será legível e gravável.

17.7. Criando e Usando Disquetes

Esta seção explica como formatar um disquete de 3.5 polegadas no FreeBSD.

Procedure: Etapas para Formatar um Disquete

Um disquete precisa ser formatado em baixo nível antes de poder ser usado. Isso geralmente é feito pelo fornecedor, mas a formatação é uma boa maneira de verificar a integridade da mídia. Para o formato de baixo nível do disquete no FreeBSD, use fdformat(1). Ao usar esse utilitário, anote todas as mensagens de erro, pois elas podem ajudar a determinar se o disco está bom ou ruim.

1. Para formatar o disquete, insira um novo disquete de 3.5 polegadas na primeira unidade de disquete e digite:

   ```
   # /usr/sbin/fdformat -f 1440 /dev/fd0
   ```

2. Após a formatação de baixo nível do disco, crie um rótulo de disco conforme requerido pelo sistema para determinar o tamanho do disco e sua geometria. Os valores de geometria suportados estão listados no arquivo /etc/disktab.

   Para escrever o rótulo do disco, use bsdlabel(8):

   ```
   # /sbin/bsdlabel -B -w /dev/fd0 fd1440
   ```

3. O disquete agora está pronto para ser formatado em alto nível com um sistema de arquivos. O sistema de arquivos do disquete pode ser UFS ou FAT, onde o FAT geralmente é

uma opção melhor para disquetes.

Para formatar o disquete com o FAT, digite:

```
# /sbin/newfs_msdos /dev/fd0
```

O disco está agora pronto para uso. Para usar o disquete, monte-o com mount_msdosfs(8). Também é possível instalar e usar emulators/mtools da coleção de ports.

17.8. Noções Básicas de Backup

A implementação de um plano de backup é essencial para que seja possível recuperar de uma falha de disco, exclusão acidental de arquivos, corrupção aleatória de arquivos ou destruição completa da máquina, incluindo a destruição de backups no local.

O tipo e a programação do backup variam, dependendo da importância dos dados, da granularidade necessária para as restaurações de arquivos e da quantidade de tempo de inatividade aceitável. Algumas técnicas de backup possíveis incluem:

- Arquivos de todo o sistema, protegidos por meio de backups em mídias permanentes, armazenados off-site. Isso fornece proteção contra todos os problemas listados acima, mas é lento e inconveniente para restaurar, especialmente para usuários sem privilégios.
- Snapshots do sistema de arquivos, que são úteis para restaurar arquivos excluídos ou versões anteriores de arquivos.
- Cópias de sistemas de arquivos inteiros ou discos que são sincronizados com outro sistema na rede usando um net/rsync agendado.
- RAID por hardware ou software, que minimiza ou evita paralisações quando um disco falha.

Normalmente, uma mistura de técnicas de backup é usada. Por exemplo, pode-se criar um agendamento semanal para automatizar um backup completo do sistema e armazená-lo off-site e para suplementá-lo, snapshots do ZFS tirados a cada hora. Além disso, é possível fazer um backup manual de diretórios ou arquivos individuais antes de fazer edições ou exclusões de arquivos.

Esta seção descreve alguns dos utilitários que podem ser usados para criar e gerenciar backups em um sistema FreeBSD.

17.8.1. Backups do Sistema de Arquivos

Os programas tradicionais UNIX™ para fazer backup de um sistema de arquivos são dump(8), que cria o backup, e restore(8), que restaura o backup. Esses utilitários funcionam no nível do bloco do disco, abaixo das abstrações dos arquivos, links e diretórios criados pelos sistemas de arquivos. Ao contrário de outros softwares de backup, `dump` faz backup de todo um sistema de arquivos e não é capaz de fazer backup de apenas parte de um sistema de arquivos ou de uma árvore de diretórios que abrange vários sistemas de arquivos. Em vez de gravar arquivos e diretórios, `dump` grava os blocos de dados brutos que compreendem arquivos e diretórios.



Se o dump for usado no diretório raiz, ele não fará o backup de /home, /usr ou de muitos outros diretórios, pois eles são tipicamente pontos de montagem para outros sistemas de arquivos ou links simbólicos nesses sistemas de arquivos.

Quando usado para restaurar dados, restore armazena arquivos temporários em /tmp/ por padrão. Ao usar um disco de recuperação com um pequeno /tmp, configure TMPDIR para um diretório com mais espaço livre para que a restauração seja bem-sucedida.

Ao usar dump, esteja ciente de que algumas peculiaridades permanecem desde seus primeiros dias na versão 6 do AT&T UNIX™, por volta de 1975. Os parâmetros padrão assumem um backup para uma fita de 9 faixas, em vez de para outro tipo de mídia ou para as fitas de alta densidade disponíveis atualmente. Esses padrões devem ser substituídos na linha de comando.

É possível fazer backup de um sistema de arquivos pela rede para outro sistema ou para uma unidade de fita conectada a outro computador. Enquanto os utilitários rdump(8) e rrestore(8) possam ser usado para este propósito, eles não são considerados seguros.

Em vez disso, pode-se usar dump e restore de uma maneira mais segura em uma conexão SSH. Este exemplo cria um backup completo e compactado de /usr e envia o arquivo de backup para o host especificado em uma conexão SSH.

Exemplo 36. Usando dump sobre ssh

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
          targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-l0.gz
```

Este exemplo configura RSH para gravar o backup em uma unidade de fita em um sistema remoto através de uma conexão SSH:

Exemplo 37. Usando o dump sobre ssh com o RSH configurado

```
# env RSH=/usr/bin/ssh /sbin/dump -0uan -f
targetuser@targetmachine.example.com:/dev/sa0 /usr
```

17.8.2. Backups de Diretório

Vários utilitários integrados estão disponíveis para backup e restauração de arquivos e diretórios especificados, conforme necessário.

Uma boa alternativa para fazer backup de todos os arquivos em um diretório é o tar(1). Este utilitário remonta à versão 6 do AT&T UNIX™ e, por padrão, assume um backup recursivo para um dispositivo de fita local. Redirecionadores podem ser utilizados para especificar o nome de um arquivo de backup.

Este exemplo cria um backup compactado do diretório atual e o salva no arquivo

/tmp/mybackup.tgz. Ao criar um arquivo de backup, verifique se o backup não está salvo no mesmo diretório em que está sendo feito backup.

Exemplo 38. Fazendo Backup do Diretório Atual com o `tar`

```
# tar czvf /tmp/mybackup.tgz .
```

Para restaurar o backup inteiro, `cd` no diretório para restaurar e especificar o nome do backup. Observe que isso sobrescreverá qualquer versão mais nova de arquivos no diretório de restauração. Em caso de dúvida, restaure para um diretório temporário ou especifique o nome do arquivo dentro do backup a ser restaurado.

Exemplo 39. Restaurando o Diretório Atual com o `tar`

```
# tar xzvf /tmp/mybackup.tgz
```

Existem dezenas de opções disponíveis, descritas em tar(1). Esse utilitário também suporta o uso de padrões de exclusão para especificar quais arquivos não devem ser incluídos ao fazer backup do diretório especificado ou restaurar arquivos de um backup.

Para criar um backup usando uma lista especificada de arquivos e diretórios, cpio(1) é uma boa escolha. Ao contrário do `tar`, o `cpio` não sabe como percorrer a árvore de diretórios e deve fornecer a lista de arquivos para backup.

Por exemplo, uma lista de arquivos pode ser criada usando `ls` ou `find`. Este exemplo cria uma listagem recursiva do diretório atual que é então canalizado para o `cpio` para criar um arquivo de backup de saída chamado /tmp/mybackup.cpio.

Exemplo 40. Usando `ls` e `cpio` para Criar um Backup Recursivo do Diretório Atual

```
# ls -R | cpio -ovF /tmp/mybackup.cpio
```

Um utilitário de backup que tenta conectar os recursos fornecidos pelo `tar` e `cpio` é pax(1). Ao longo dos anos, as várias versões do `tar` e do `cpio` tornaram-se ligeiramente incompatíveis. POSIX™ criou `pax` que tenta ler e escrever muitos dos vários formatos `cpio` e `tar`, além de novos formatos próprios.

O `pax` equivalente aos exemplos anteriores seria:

Exemplo 41. Fazendo Backup do Diretório Atual com `pax`

```
# pax -wf /tmp/mybackup.pax .
```

17.8.3. Usando Fitas de Dados para Backups

Embora a tecnologia de fitas tenha continuado a evoluir, os sistemas de backup modernos tendem a combinar backups externos com mídias removíveis locais. O FreeBSD suporta qualquer unidade de fita que use SCSI, como LTO ou DAT. Há suporte limitado para as unidades de fita SATA e USB.

Para dispositivos de fita SCSI, o FreeBSD usa o driver sa(4) e os dispositivos /dev/sa0, /dev/nsa0 e /dev/esa0. O nome do dispositivo físico é /dev/sa0. Quando /dev/nsa0 é usado, o aplicativo de backup não rebobina a fita depois de gravar um arquivo, o que permite gravar mais de um arquivo em uma fita. O uso de /dev/esa0 ejeta a fita após o dispositivo ser fechado.

No FreeBSD, o `mt` é usado para controlar as operações da unidade de fita, como procurar arquivos em uma fita ou gravar marcas de controle na fita. Por exemplo, os três primeiros arquivos em uma fita podem ser preservados, ignorando-os antes de gravar um novo arquivo:

```
# mt -f /dev/nsa0 fsf 3
```

Este utilitário suporta muitas operações. Consulte mt(1) para detalhes.

Para gravar um único arquivo em fita usando `tar`, especifique o nome do dispositivo de fita e o arquivo para backup:

```
# tar cvf /dev/sa0 file
```

Para recuperar arquivos de um arquivo `tar` em fita no diretório atual:

```
# tar xvf /dev/sa0
```

Para fazer backup de um sistema de arquivos UFS, use `dump`. Este exemplo faz o backup de /usr sem rebobinar a fita quando terminar:

```
# dump -0aL -b64 -f /dev/nsa0 /usr
```

Para restaurar arquivos interativamente de um arquivo `dump` em fita no diretório atual:

```
# restore -i -f /dev/nsa0
```

17.8.4. Utilitários de Backup de Terceiros

A Coleção de Ports do FreeBSD fornece muitos utilitários de terceiros que podem ser usados para agendar a criação de backups, simplificar o backup em fita e tornar os backups mais fáceis e convenientes. Muitos desses aplicativos são baseados em cliente/servidor e podem ser usados para automatizar os backups de um único sistema ou de todos os computadores em uma rede.

Os utilitários populares incluem Amanda, Bacula, rsync e duplicity.

17.8.5. Recuperação de Emergência

Além dos backups regulares, é recomendável executar as etapas a seguir como parte de um plano de preparação para emergências.

Crie uma cópia impressa da saída dos seguintes comandos:

- `gpart show`
- `more /etc/fstab`
- `dmesg`

Armazene esta saída e uma cópia da mídia de instalação em um local seguro. Se uma restauração de emergência for necessária, inicialize na mídia de instalação e selecione `Live CD` para acessar um shell de recuperação. Esse modo de recuperação pode ser usado para exibir o estado atual do sistema e, se necessário, reformatar discos e restaurar dados de backups.



A mídia de instalação do FreeBSD/i386 11.2-RELEASE não inclui um shell de recuperação. Para esta versão, baixe e grave uma imagem do Livefs CD de ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/11.2/FreeBSD-11.2-RELEASE-i386-livefs.iso.

Em seguida, teste o shell de recuperação e os backups. Faça anotações do procedimento. Armazene estas notas com a mídia, as impressões e os backups. Estas notas podem impedir a destruição inadvertida dos backups, enquanto sob o estresse de realizar uma recuperação de emergência.

Para uma medida adicional de segurança, armazene o backup mais recente em um local remoto, fisicamente separado dos computadores e das unidades de disco por uma distância significativa.

17.9. Discos de Memória

Além de discos físicos, o FreeBSD também suporta a criação e uso de discos de memória. Um uso possível para um disco de memória é acessar o conteúdo de um sistema de arquivos ISO sem a sobrecarga de primeiro gravá-lo em um CD ou DVD e, em seguida, montar a mídia CD/DVD .

No FreeBSD, o driver md(4) é usado para fornecer suporte para discos de memória. O kernel GENERIC inclui este driver. Ao usar um arquivo de configuração de kernel personalizado, certifique-se de incluir esta linha:

```
device md
```

17.9.1. Anexando e Desanexando Imagens Existentes

Para montar uma imagem do sistema de arquivos existente, use o `mdconfig` para especificar o nome do arquivo ISO e um número de unidade livre. Em seguida, consulte esse número de unidade para montá-lo em um ponto de montagem existente. Uma vez montado, os arquivos na imagem ISO

aparecerão no ponto de montagem. Este exemplo anexa o arquivo *diskimage.iso* ao dispositivo de memória /dev/md0 e monta o dispositivo de memória em /mnt:

```
# mdconfig -f diskimage.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Note que `-t cd9660` foi usado para montar uma imagem ISO. Se um número de unidade não for especificado com `-u`, o `mdconfig` alocará automaticamente um dispositivo de memória não utilizado e exibirá o nome da unidade alocada, como md4. Consulte mdconfig(8) para mais detalhes sobre este comando e suas opções.

Quando um disco de memória não está mais em uso, seus recursos devem ser liberados de volta ao sistema. Primeiro, desmonte o sistema de arquivos e use o `mdconfig` para desanexar o disco do sistema e liberar seus recursos. Para continuar este exemplo:

```
# umount /mnt
# mdconfig -d -u 0
```

Para determinar se algum disco de memória ainda está conectado ao sistema, digite `mdconfig -l`.

17.9.2. Criando um Disco Virtual Baseado em Arquivo ou Memória

O FreeBSD também suporta discos virtuais onde o armazenamento a ser utilizado é alocado a partir de um disco rígido ou de uma área de memória. O primeiro método é comumente referido como um disco virtual baseado em arquivo e o segundo como um disco virtual baseado em memória. Ambos os tipos podem ser criados usando o `mdconfig`.

Para criar um novo disco virtual baseado em memória, especifique um tipo de `swap` e o tamanho do disco de memória a ser criado. Em seguida, formate o disco de memória com um sistema de arquivos e monte como de costume. Este exemplo cria um disco de memória de 5M na unidade `1`. Esse disco de memória é formatado com o sistema de arquivos UFS antes de ser montado:

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
        with soft updates
super-block backups (for fsck -b #) at:
 160, 2752, 5344, 7936
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

Para criar um novo disco virtual baseado em arquivo, primeiro aloque a área que será usada para o disco. Esse exemplo cria um arquivo vázio de 5MB chamado newimage:

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
```

Em seguida, anexe esse arquivo a um disco de memória, rotule o disco de memória e formate-o com o sistema de arquivos UFS, monte o disco de memória e verifique o tamanho do disco com backup de arquivo:

```
# mdconfig -f newimage -u 0
# bsdlabel -w md0 auto
# newfs -U md0a
/dev/md0a: 5.0MB (10224 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
 160, 2720, 5280, 7840
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

São necessários vários comandos para criar um disco virtual baseado em arquivo ou memória usando `mdconfig`. O FreeBSD também vem com o `mdmfs` que configura automaticamente um disco de memória, formata-o com o sistema de arquivos UFS e o monta. Por exemplo, depois de criar *newimage* com `dd`, esse comando é equivalente a executar os comandos `bsdlabel`, `newfs` e `mount` mostrados acima:

```
# mdmfs -F newimage -s 5m md0 /mnt
```

Para criar um novo disco virtual baseado em memória com o `mdmfs`, use este comando:

```
# mdmfs -s 5m md1 /mnt
```

Se o número da unidade não for especificado, o `mdmfs` selecionará automaticamente um dispositivo de memória não utilizado. Para mais detalhes sobre `mdmfs`, consulte mdmfs(8).

17.10. Snapshots de Sistemas de Arquivos

O FreeBSD oferece um recurso em conjunto com Atualizações Soft: snapshots do sistema de arquivos.

Os Snapshots de UFS permitem que um usuário crie imagens de sistemas de arquivos especificados e as trate como um arquivo. Os arquivos de snapshot devem ser criados no sistema de arquivos no qual a ação é executada e um usuário pode criar no máximo 20 snapshots por sistema de arquivos. Os snapshots ativos são registradas no superbloco, de modo que são persistentes nas operações de desmontagem e remontagem, juntamente com reinicializações do sistema. Quando um snapshot

não é mais necessário, ele pode ser removido usando rm(1). Embora os snapshots possam ser removidos em qualquer ordem, todo o espaço usado pode não ser adquirido porque outro snapshot possivelmente reivindicará alguns dos blocos liberados.

A flag de arquivo `snapshot` não alterável é definida por mksnap_ffs(8) após a criação inicial de um arquivo de snapshot. O unlink(1) cria uma exceção para arquivos de snapshots, pois permite que sejam removidos.

Os snapshots são criados usando mount(8). Para colocar um snapshot de /var no arquivo /var/snapshot/snap, use o seguinte comando:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Como alternativa, use mksnap_ffs(8) para criar o snapshot:

```
# mksnap_ffs /var /var/snapshot/snap
```

É possível encontrar arquivos de snapshots em um sistema de arquivos, como /var, usando find(1):

```
# find /var -flags snapshot
```

Depois que um snapshot foi criado, ele tem vários usos:

- Alguns administradores usarão um arquivo de snapshot para fins de backup, porque o snapshot pode ser transferido para um CDs ou fita.
- O verificador de integridade do sistema de arquivos, fsck(8), pode ser executado em um snapshot. Supondo que o sistema de arquivos estava limpo quando foi montado, isso deve sempre fornecer um resultado limpo e imutável.
- Executando dump(8) em um snapshot produzirá um arquivo de dump que seja consistente com o sistema de arquivos e o registro de data e hora do snapshot. dump(8) também pode criar um snapshot, criar uma imagem de dump e remover o snapshot em um comando usando `-L`.
- O snapshot pode ser montado como uma imagem congelada do sistema de arquivos. Para montar o snapshot use mount(8) passando o nome do snapshot /var/snapshot/snap:

  ```
  # mdconfig -a -t vnode -o readonly -f /var/snapshot/snap -u 4
  # mount -r /dev/md4 /mnt
  ```

O /var congelado agora está disponível através de /mnt. Tudo estará inicialmente no mesmo estado que estava quando o snapshot foi criado. A única exceção é que os snapshots anteriores aparecerão como arquivos com comprimento zero. Para desmontar o snapshot, use:

```
# umount /mnt
# mdconfig -d -u 4
```

Para obter mais informações sobre `softupdates` e snapshots do sistema de arquivos, incluindo documentos técnicos, visite o site do Marshall Kirk McKusick em http://www.mckusick.com/.

17.11. Cotas de Disco

As cotas de disco podem ser usadas para limitar a quantidade de espaço em disco ou o número de arquivos que um usuário ou membros de um grupo podem alocar em uma base por sistema de arquivos. Isso impede que um usuário ou grupo de usuários consuma todo o espaço em disco disponível.

Esta seção descreve como configurar cotas de disco para o sistema de arquivos UFS. Para configurar cotas no sistema de arquivos ZFS, consulte Cotas para Datasets

17.11.1. Habilitando Cotas de Disco

Para determinar se o kernel do FreeBSD fornece suporte para cotas de disco:

```
% sysctl kern.features.ufs_quota
kern.features.ufs_quota: 1
```

Neste exemplo, o `1` indica suporte à cota. Se o valor for `0`, adicione a seguinte linha a um arquivo de configuração de kernel personalizado e reconstrua o kernel usando as instruções em Configurando o kernel do FreeBSD:

```
options QUOTA
```

Em seguida, habilite as cotas de disco no arquivo /etc/rc.conf:

```
quota_enable="YES"
```

Normalmente, na inicialização, a integridade da cota de cada sistema de arquivos é verificada por quotacheck(8). Esse programa garante que os dados no banco de dados de cotas reflitam adequadamente os dados no sistema de arquivos. Este é um processo demorado que afetará significativamente o tempo que o sistema leva para inicializar. Para pular este passo, adicione esta variável ao arquivo /etc/rc.conf:

```
check_quotas="NO"
```

Por fim, edite o arquivo /etc/fstab para habilitar as cotas de disco por sistema de arquivos. Para habilitar cotas por usuário em um sistema de arquivos, adicione `userquota` ao campo de opções na entrada /etc/fstab para o sistema de arquivos ativar as cotas. Por exemplo:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Para ativar cotas de grupo, use `groupquota`. Para ativar cotas de usuários e grupos, separe as opções com uma vírgula:

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

Por padrão, os arquivos de cota são armazenados no diretório raiz do sistema de arquivos como quota.user e quota.group. Consulte fstab(5) para obter mais informações. Especificar um local alternativo para os arquivos de cotas não é recomendado.

Quando a configuração estiver concluída, reinicialize o sistema e o /etc/rc executará automaticamente os comandos apropriados para criar os arquivos de cotas iniciais para todas as cotas ativadas em /etc/fstab.

No curso normal das operações, não deve haver necessidade de executar manualmente o quotacheck(8), quotaon(8), ou quotaoff(8). No entanto, deve-se ler estas páginas de manual para se familiarizar com sua operação.

17.11.2. Definindo Limites de Cota

Para verificar se as cotas estão ativadas, execute:

```
# quota -v
```

Deve haver um resumo de uma linha sobre o uso de disco e limites de cota atuais para cada sistema de arquivos em que as cotas estão ativadas.

O sistema agora está pronto para receber limites de cota com `edquota`.

Várias opções estão disponíveis para impor limites à quantidade de espaço em disco que um usuário ou grupo pode alocar e quantos arquivos eles podem criar. As alocações podem ser limitadas com base no espaço em disco (cotas de bloco), no número de arquivos (cotas de inode) ou em uma combinação de ambos. Cada limite é subdividido em duas categorias: limites rígidos e flexíveis.

Um limite rígido não pode ser excedido. Quando um usuário atinge um limite rígido, nenhuma outra alocação pode ser feita nesse sistema de arquivos por esse usuário. Por exemplo, se o usuário tiver um limite rígido de 500 kbytes em um sistema de arquivos e estiver usando atualmente 490 kbytes, o usuário poderá alocar apenas 10 kbytes adicionais. A tentativa de alocar 11 kbytes adicionais falhará.

Os limites flexíveis podem ser excedidos por um período de tempo limitado, conhecido como período de tolerância, que é uma semana por padrão. Se um usuário permanecer acima do limite por mais tempo do que o período de carência, o limite flexível se tornará um limite rígido e nenhuma outra alocação será permitida. Quando o usuário cai abaixo do limite flexível, o período de carência é zerado.

No exemplo a seguir, a cota da conta `test` está sendo editada. Quando `edquota` é invocado, o editor especificado por `EDITOR` é aberto para editar os limites de cota. O editor padrão é configurado para

vi.

```
# edquota -u test
Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

Normalmente, há duas linhas para cada sistema de arquivos com cotas ativadas. Uma linha representa os limites do bloco e a outra representa os limites do inode. Altere o valor para modificar o limite de cota. Por exemplo, para aumentar o limite de blocos em /usr para um limite flexível de `500` e um limite rígido de `600`, altere os valores nesse campo. linha da seguinte forma:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

Os novos limites de cotas entram em vigor ao sair do editor.

Às vezes, é desejável definir limites de cota em vários usuários. Isso pode ser feito primeiro atribuindo o limite de cota desejado a um usuário. Em seguida, use `-p` para duplicar essa cota para um intervalo especificado de IDs de usuário (UIDs). O comando a seguir duplicará esses limites de cota para UIDs de `10.000` até `19.999`:

```
# edquota -p test 10000-19999
```

Para mais informações, consulte edquota(8).

17.11.3. Verificando Limites de Cota e Uso de Disco

Para verificar cotas individuais de usuários ou de grupos e uso de disco, use quota(1). Um usuário só pode examinar sua própria cota e a cota de um grupo do qual é membro. Somente o superusuário pode visualizar todas as cotas de usuários e grupos. Para obter um resumo de todas as cotas e uso de disco para sistemas de arquivos com cotas ativadas, use repquota(8).

Normalmente, os sistemas de arquivos nos quais o usuário não está usando nenhum espaço em disco não serão exibidos na saída de `quota`, mesmo que o usuário tenha um limite de cota atribuído a esse sistema de arquivos. Use `-v` para exibir esses sistemas de arquivos. A seguir está a saída de amostra de `quota -v` para um usuário que possui limites de cota em dois sistemas de arquivos.

```
Disk quotas for user test (uid 1002):
     Filesystem  usage    quota   limit   grace   files   quota   limit   grace
          /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75                0      50      60
```

Neste exemplo, o usuário está atualmente 15 kbytes sobre o limite flexível de 50 kbytes em /usr e

tem 5 dias de período de carência restante. O asterisco * indica que o usuário está atualmente acima do limite de cota.

17.11.4. Quotas sobre o NFS

As cotas são impostas pelo subsistema de cotas no servidor NFS. O daemon rpc.rquotad(8) disponibiliza informações de quota para `quota` em clientes NFS, permitindo que os usuários nessas máquinas visualizem suas estatísticas de cota.

No servidor NFS, ative o `rpc.rquotad` removendo o `#` desta linha em /etc/inetd.conf:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Em seguida, reinicie o `inetd`:

```
# service inetd restart
```

17.12. Criptografando Partições de Disco

O FreeBSD oferece excelentes proteções on-line contra acesso não autorizado a dados. As permissões de arquivo e o Mandatory Access Control (MAC) ajudam a impedir que usuários não autorizados acessem dados enquanto o sistema operacional está ativo e o computador está ligado. No entanto, as permissões impostas pelo sistema operacional são irrelevantes se um invasor tiver acesso físico a um computador e puder mover o disco rígido do computador para outro sistema para copiar e analisar os dados.

Independentemente de como um invasor pode ter acesso a um disco rígido ou um computador desligado, os subsistemas criptográficos baseados em GEOM incorporados ao FreeBSD são capazes de proteger os dados nos sistemas de arquivos do computador contra atacantes super motivados com recursos significativos. Ao contrário dos métodos de criptografia que criptografam arquivos individuais, os utilitários incorporados `gbde` e `geli` podem ser usados para criptografar de forma transparente sistemas de arquivos inteiros. Nenhum dado aberto sequer toca na bandeja do disco rígido.

Este capítulo demonstra como criar um sistema de arquivos criptografado no FreeBSD. Primeiro ele demonstra o processo usando o `gbde` e depois demonstra o mesmo exemplo usando `geli`.

17.12.1. Criptografia de Disco com gbde

O objetivo do utilitário gbde(4) é fornecer um desafio formidável para que um invasor que tenha acesso ao conteúdo de um dispositivo de armazenamento *frio*. No entanto, se o computador for comprometido enquanto estiver em funcionamento e o dispositivo de armazenamento estiver ativamente conectado, ou se o invasor tiver acesso a uma frase secreta válida, ele não oferecerá proteção ao conteúdo do dispositivo de armazenamento. Portanto, é importante fornecer segurança física enquanto o sistema está em execução e proteger a frase secreta usada pelo mecanismo de criptografia.

Este recurso oferece várias barreiras para proteger os dados armazenados em cada setor de disco. Ele criptografa o conteúdo de um setor de disco usando o AES de 128 bits no modo CBC. Cada setor no disco é criptografado com uma chave AES diferente. Para obter mais informações sobre o design criptográfico, incluindo como as chaves do setor são derivadas da frase secreta fornecida pelo usuário, consulte gbde(4).

O FreeBSD fornece um módulo do kernel para gbde, que pode ser carregado com este comando:

```
# kldload geom_bde
```

Se estiver usando um arquivo de configuração de kernel personalizado, certifique-se de que ele contenha esta linha:

```
options GEOM_BDE
```

O exemplo a seguir demonstra a adição de um novo disco rígido a um sistema que conterá uma única partição criptografada que será montada como /private.

Procedure: Criptografando uma Partição com gbde

1. Adicione o Novo Disco Rígido

   Instale a nova unidade no sistema, conforme explicado em Adicionando Discos. Para propósitos deste exemplo, uma nova partição de disco rígido foi adicionada como /dev/ad4s1c e /dev/ad0s1* representa o existente partições padrão do FreeBSD.

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

2. Criar um diretório para conter os arquivos de lock do `gbde`

   ```
   # mkdir /etc/gbde
   ```

   O arquivo de lock gbde contém informações que o gbde requer para acessar partições criptografadas. Sem acesso ao arquivo de lock, o gbde não poderá descriptografar os dados contidos na partição criptografada sem intervenção manual significativa que não seja suportada pelo software. Cada partição criptografada usa um arquivo de lock separado.

3. Inicialize a Partição `gbde`

   Uma partição gbde deve ser inicializada antes de poder ser usada. Essa inicialização precisa ser executada apenas uma vez. Esse comando abrirá o editor padrão, para definir várias opções de configuração em um modelo. Para uso com o sistema de arquivos UFS, defina o sector_size como 2048:

```
# gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15
16:03:31Z dbaio $
#
# Sector size is the smallest unit of data which can be read or written.
# Making it too small decreases performance and decreases available space.
# Making it too large may prevent filesystems from working.  512 is the
# minimum and always safe.  For UFS, use the fragment size
#
sector_size =   2048
[...]
```

Depois que a edição for salva, o usuário será solicitado a digitar duas vezes a frase secreta usada para proteger os dados. A frase secreta deve ser a mesma em ambas as vezes. A capacidade de gbde de proteger os dados depende inteiramente da qualidade da frase secreta. Para obter dicas sobre como selecionar uma frase secreta que seja fácil de lembrar, consulte http://world.std.com/~reinhold/diceware.htm.

Essa inicialização cria um arquivo de lock para a partição do gbde. Neste exemplo, ele é armazenado como /etc/gbde/ad4s1c.lock. Os arquivos de lock devem terminar em ".lock" para serem corretamente detectados pelo script de inicialização do /etc/rc.d/gbde.



Arquivos de lock *devem* ter backups junto com o conteúdo de qualquer partição criptografada. Sem o arquivo de lock, o proprietário legítimo não poderá acessar os dados na partição criptografada.

4. Anexando a Partição Criptografada ao Kernel

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

Este comando solicitará a entrada da senha que foi selecionada durante a inicialização da partição criptografada. O novo dispositivo criptografado aparecerá em /dev como /dev/device_name.bde:

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Criando um Sistema de Arquivos no Dispositivo Criptografado

Uma vez que o dispositivo criptografado tenha sido anexado ao kernel, um sistema de arquivos pode ser criado no dispositivo. Este exemplo cria um sistema de arquivos UFS com atualizações soft ativadas. Certifique-se de especificar a partição que possui uma extensão *.bde:

```
# newfs -U /dev/ad4s1c.bde
```

6. Montando a Partição Criptografada

   Crie um ponto de montagem e monte o sistema de arquivos criptografados:

```
# mkdir /private
# mount /dev/ad4s1c.bde /private
```

7. Verificar se o sistema de arquivos criptografados está disponível

   O sistema de arquivos criptografados agora deve estar visível e disponível para uso:

```
% df -H
Filesystem        Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a      1037M    72M   883M     8%    /
/devfs            1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f       8.1G    55K   7.5G     0%    /home
/dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
/dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr
/dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
```

Após cada inicialização, todos os sistemas de arquivos criptografados devem ser reconectados manualmente ao kernel, verificados quanto a erros e montados antes que os sistemas de arquivos possam ser usados. Para configurar estas etapas, adicione as seguintes linhas ao arquivo /etc/rc.conf:

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

Isso requer que a frase secreta seja inserida no console no momento da inicialização. Depois de digitar a senha correta, a partição criptografada será montada automaticamente. Opções adicionais de inicialização do gbde estão disponíveis e listadas em rc.conf(5).



O sysinstall é incompatível com os dispositivos criptografados com gbde. Todos os dispositivos *.bde devem ser desanexado do kernel antes de iniciar o sysinstall ou ele irá travar durante a análise inicial dos dispositivos. Para desanexar o dispositivo criptografado usado no exemplo, use o seguinte comando:

```
# gbde detach /dev/ad4s1c
```

17.12.2. Criptografia de Disco com `geli`

Uma classe criptográfica alternativa GEOM está disponível usando `geli`. Este utilitário de controle adiciona alguns recursos e usa um esquema diferente para fazer trabalho criptográfico. Ele fornece os seguintes recursos:

- Utiliza o framework crypto(9) e usa automaticamente o hardware criptográfico quando ele está disponível.
- Suporta vários algoritmos criptográficos, como AES, Blowfish e 3DES.
- Permite que a partição raiz seja criptografada. A frase secreta usada para acessar a partição root criptografada será solicitada durante a inicialização do sistema.
- Permite o uso de duas chaves independentes.
- É rápido, pois executa criptografia simples de setor a setor.
- Permite backup e restauração de chaves mestras. Se um usuário destruir suas chaves, ainda é possível obter acesso aos dados restaurando as chaves do backup.
- Permite que um disco seja anexado com uma chave única aleatória que é útil para partições swap e sistemas de arquivos temporários.

Mais recursos e exemplos de uso podem ser encontrados em geli(8).

O exemplo a seguir descreve como gerar um arquivo de chave que será usado como parte da chave mestra para o provedor criptografado montado em /private. O arquivo chave fornecerá alguns dados aleatórios usados para criptografar a chave mestra. A chave mestra também será protegida por uma frase secreta. O tamanho do setor do provedor será de 4kB. O exemplo descreve como se conectar ao provedor `geli`, criar um sistema de arquivos, montá-lo, trabalhar com ele e, finalmente, como desanexá-lo.

Procedure: Criptografando uma Partição com `geli`

1. Carregando o suporte ao `geli`

   O suporte para `geli` está disponível como um módulo de kernel carregável. Para configurar o sistema para carregar automaticamente o módulo no momento da inicialização, adicione a seguinte linha ao arquivo /boot/loader.conf:

   ```
   geom_eli_load="YES"
   ```

   Para carregar o módulo do kernel agora:

   ```
   # kldload geom_eli
   ```

   Para um kernel customizado, assegure-se de que o arquivo de configuração do kernel contenha estas linhas:

```
options GEOM_ELI
device crypto
```

2. Gerando a Chave Mestra

   Os comandos a seguir geram uma chave mestra com a qual todos os dados serão criptografados. Esta chave nunca pode ser alterada. Em vez de usá-lo diretamente, ele é criptografado com uma ou mais chaves de usuário. As chaves do usuário são compostas por uma combinação opcional de bytes aleatórios de um arquivo, /root/da2.key e/ou uma senha. Neste caso, a fonte de dados do arquivo de chave é /dev/random. Este comando também configura o tamanho do setor do provedor (/dev/da2.eli) como 4kB, para melhor desempenho:

   ```
   # dd if=/dev/random of=/root/da2.key bs=64 count=1
   # geli init -K /root/da2.key -s 4096 /dev/da2
   Enter new passphrase:
   Reenter new passphrase:
   ```

   Não é obrigatório o uso de uma frase secreta e de um arquivo de chave, pois cada método de proteger a chave mestra pode ser usado isoladamente.

   Se o arquivo de chave é dado como "-", a entrada padrão será usada. Por exemplo, este comando gera três arquivos principais:

   ```
   # cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
   ```

3. Anexando o Provedor com a Chave Gerada

   Para anexar o provedor, especifique o arquivo de chave, o nome do disco e a frase secreta:

   ```
   # geli attach -k /root/da2.key /dev/da2
   Enter passphrase:
   ```

   Isso cria um novo dispositivo com uma extensão .eli:

   ```
   # ls /dev/da2*
   /dev/da2  /dev/da2.eli
   ```

4. Criando o Novo Sistema de Arquivos

   Em seguida, formate o dispositivo com o sistema de arquivos UFS e monte-o em um ponto de montagem existente:

   ```
   # dd if=/dev/random of=/dev/da2.eli bs=1m
   ```

```
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

O sistema de arquivos criptografado agora deve estar disponível para uso:

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
/dev/da2.eli   150G   4.1K   138G     0%    /private
```

Uma vez que o trabalho na partição criptografada é feito, e a partição /private não é mais necessária, é prudente colocar o dispositivo no armazenamento frio desmontando e desanexando a partição `geli` criptografada do kernel:

```
# umount /private
# geli detach da2.eli
```

Um script rc.d é fornecido para simplificar a montagem de dispositivos criptografados `geli` no momento da inicialização. Para este exemplo, adicione estas linhas ao arquivo /etc/rc.conf:

```
geli_devices="da2"
geli_da2_flags="-k /root/da2.key"
```

Isto configura o /dev/da2 como um provedor `geli` com uma chave mestra de /root/da2.key. O sistema irá desanexando automaticamente o provedor do kernel antes que o sistema seja desligado. Durante o processo de inicialização, o script solicitará a frase secreta antes de conectar o provedor. Outras mensagens do kernel podem ser mostradas antes e depois do prompt da frase secreta. Se o processo de inicialização parecer travar, procure cuidadosamente o prompt de senha entre as outras mensagens. Depois que a frase secreta correta é inserida, o provedor é anexado. O sistema de arquivos é então montado, normalmente por uma entrada em /etc/fstab. Consulte Montando e Desmontando Sistemas de Arquivos para obter instruções sobre como configurar um sistema de arquivos para montar no momento da inicialização.

17.13. Criptografando Swap

Como a criptografia de partições de disco, a criptografia do espaço swap é usada para proteger informações confidenciais. Considere um aplicativo que lida com senhas. Contanto que essas senhas permaneçam na memória física, elas não serão gravadas no disco e serão apagadas após a reinicialização. No entanto, se o FreeBSD iniciar a troca de páginas de memória para liberar espaço, as senhas podem ser gravadas no disco não criptografadas. O espaço de troca de criptografia pode

ser uma solução para esse cenário.

Esta seção demonstra como configurar uma partição swap criptografada usando criptografia gbde(8) ou geli(8). Ele assume que /dev/ada0s1b é a partição swap.

17.13.1. Configurando Swap Criptografada

As partições de swap não são criptografadas por padrão e devem ser limpas de quaisquer dados confidenciais antes de continuar. Para sobrescrever a partição swap atual com lixo aleatório, execute o seguinte comando:

```
# dd if=/dev/random of=/dev/ada0s1b bs=1m
```

Para criptografar a partição swap usando gbde(8), adicione o sufixo `.bde` à linha de swap no /etc/fstab:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ada0s1b.bde    none        swap    sw      0   0
```

Para criptografar a partição swap usando geli(8), use o sufixo `.eli`:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ada0s1b.eli    none        swap    sw      0   0
```

Por padrão, geli(8) usa o algoritmo AES com um comprimento de chave de 128 bits. Normalmente, as configurações padrão serão suficientes. Se desejado, estes padrões podem ser alterados no campo de opções no arquivo /etc/fstab. As possíveis flags são:

aalgo

Algoritmo de verificação de integridade de dados usado para garantir que os dados criptografados não tenham sido adulterados. Veja geli(8) para obter uma lista dos algoritmos suportados.

ealgo

Algoritmo de criptografia usado para proteger os dados. Veja geli(8) para obter uma lista dos algoritmos suportados.

keylen

O comprimento da chave usada para o algoritmo de criptografia. Veja geli(8) para os comprimentos de chave que são suportados por cada algoritmo de criptografia.

sectorsize

O tamanho em que o blocos de dados é dividido antes de ser criptografado. Tamanhos de setor maiores aumentam o desempenho ao custo de maior sobrecarga de armazenamento. O tamanho recomendado é de 4096 bytes.

Este exemplo configura uma partição swap criptografada usando o algoritmo Blowfish com um comprimento de chave de 128 bits e um setor de tamanho de 4 kilobytes:

```
# Device        Mountpoint  FStype  Options             Dump    Pass#
/dev/ada0s1b.eli    none        swap    sw,ealgo=blowfish,keylen=128,sectorsize=4096
0   0
```

17.13.2. Verificação de Swap Criptografada

Depois que o sistema for reinicializado, a operação adequada da swap criptografada poderá ser verificada usando `swapinfo`.

Se gbde(8) estiver sendo usado:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.bde   542720        0   542720     0%
```

Se geli(8) estiver sendo usado:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.eli   542720        0   542720     0%
```

17.14. Alta Disponibilidade de Armazenamento (HAST)

A alta disponibilidade é um dos principais requisitos em aplicativos de negócios sérios e o armazenamento altamente disponível é um componente-chave nesses ambientes. No FreeBSD, o framework Alta Disponiblidade de Armazenamento (HAST) permite o armazenamento transparente dos mesmos dados em várias máquinas fisicamente separadas conectadas por uma rede TCP/IP. HAST pode ser entendido como um RAID1 (mirror) baseado em rede, e é similar ao sistema de armazenamento DRBD® usado na plataforma GNU/Linux™. Em combinação com outros recursos de alta disponibilidade do FreeBSD, como o CARP, o HAST possibilita a criação de um cluster de armazenamento altamente disponível, resistente a falhas de hardware.

A seguir estão as principais características do HAST:

- Pode ser usado para mascarar erros de I/O em discos rígidos locais.
- Agnóstico a sistema de arquivos, pois funciona com qualquer sistema de arquivos suportado pelo FreeBSD.
- Ressincronização eficiente e rápida, pois somente os blocos que foram modificados durante o tempo de inatividade de um nó são sincronizados.
- Pode ser usado em um ambiente já implantado para adicionar redundância adicional.
- Juntamente com o CARP, Heartbeat, ou outras ferramentas, ele pode ser usado para construir

um sistema de armazenamento robusto e durável.

Depois de ler esta seção, você saberá:

- O que é HAST, como ele funciona e quais recursos ele fornece.
- Como configurar e usar o HAST no FreeBSD.
- Como integrar CARP e devd(8) para criar um sistema de armazenamento robusto.

Antes de ler esta seção, você deve:

- Entender os fundamentos do UNIX™ e do FreeBSD (Fundamentos do FreeBSD).
- Saber como configurar interfaces de rede e outros subsistemas principais do FreeBSD (Configuração e Ajuste).
- Ter uma boa compreensão da rede do FreeBSD (Comunicação de rede).

O projeto HAST foi patrocinado pela Fundação FreeBSD com o apoio de http://www.omc.net/ e http://www.transip.nl/.

17.14.1. Operação HAST

O HAST fornece replicação síncrona em nível de bloco entre duas máquinas físicas: o *primário*, também conhecido como o nó *master*, e o *secundário*, ou nó *slave*. Essas duas máquinas juntas são chamadas de cluster.

Como o HAST funciona em uma configuração primária-secundária, ele permite que apenas um dos nós do cluster esteja ativo a qualquer momento. O nó primário, também chamado de *active*, é aquele que irá lidar com todas as solicitações de I/O para dispositivos gerenciados por HAST. O nó secundário é automaticamente sincronizado a partir do nó primário.

Os componentes físicos do sistema HAST são o disco local no nó primário e o disco no nó secundário remoto.

O HAST opera de forma síncrona em um nível de bloco, tornando-o transparente para sistemas de arquivos e aplicativos. O HAST fornece provedores GEOM regulares em /dev/hast/ para uso por outras ferramentas ou aplicativos. Não há diferença entre o uso de dispositivos HAST e discos ou partições brutas.

Cada operação de gravação, exclusão ou liberação é enviada para o disco local e para o disco remoto sobre TCP/IP . Cada operação de leitura é fornecida a partir do disco local, a menos que o disco local não esteja atualizado ou ocorra um erro de I/O. Nesses casos, a operação de leitura é enviada para o nó secundário.

HAST tenta fornecer recuperação rápida de falhas. Por esse motivo, é importante reduzir o tempo de sincronização após a interrupção de um nó. Para fornecer sincronização rápida, o HAST gerencia um bitmap no disco de extensões sujas e sincroniza apenas aquelas durante uma sincronização regular, com exceção da sincronização inicial.

Existem muitas maneiras de lidar com a sincronização. O HAST implementa vários modos de replicação para lidar com diferentes métodos de sincronização:

- *memsync*: Este modo reporta uma operação de gravação como concluída quando a operação de gravação local é finalizada e quando o nó remoto reconhece a chegada dos dados, mas antes de realmente armazenar os dados. Os dados no nó remoto serão armazenados diretamente após o envio da confirmação. Este modo destina-se a reduzir a latência, mas ainda fornece boa confiabilidade. Este modo é o padrão.
- *fullsync*: Este modo relata uma operação de gravação como concluída quando a gravação local e a gravação remota são concluídas. Este é o modo de replicação mais seguro e mais lento.
- *async*: Este modo relata uma operação de gravação como concluída quando a gravação local é concluída. Este é o modo de replicação mais rápido e mais perigoso. Ele deve ser usado somente ao replicar para um nó distante, onde a latência é muito alta para outros modos.

17.14.2. Configuração do HAST

O framework HAST consiste em vários componentes:

- O daemon hastd(8) que fornece sincronização de dados. Quando este daemon é iniciado, ele carregará automaticamente `geom_gate.ko`.
- O utilitário de gerenciamento de usuário, hastctl(8).
- O arquivo de configuração hast.conf(5). Este arquivo deve existir antes de iniciar o hastd.

Usuários que preferem construir estaticamente o suporte a `GEOM_GATE` no kernel devem adicionar esta linha ao arquivo de configuração do kernel personalizado e reconstruir o kernel usando as instruções em Configurando o kernel do FreeBSD:

```
options GEOM_GATE
```

O exemplo a seguir descreve como configurar dois nós na operação mestre-escravo/primário-secundário usando HAST para replicar os dados entre os dois. Os nós serão chamados `hasta`, com um endereço IP `172.16.0.1`, e `hastb`, com um endereço IP `172.16.0.2`. Ambos os nós terão um disco rígido dedicado /dev/ad6 do mesmo tamanho para a operação HAST. O conjunto HAST, por vezes referido como um recurso ou o provedor GEOM em /dev/hast/, será chamado `test`.

A configuração do HAST é feita usando o arquivo /etc/hast.conf. Este arquivo deve ser idêntico nos dois nós. A configuração mais simples é:

```
resource test {
    on hasta {
        local /dev/ad6
        remote 172.16.0.2
    }
    on hastb {
        local /dev/ad6
        remote 172.16.0.1
    }
}
```

Para uma configuração mais avançada, consulte hast.conf(5).



Também é possível usar nomes de host nas instruções `remote` se os hosts forem resolvidos e definidos no arquivo /etc/hosts ou no DNS local.

Uma vez que a configuração exista em ambos os nós, o conjunto HAST pode ser criado. Execute esses comandos nos dois nós para colocar os metadados iniciais no disco local e para iniciar hastd(8):

```
# hastctl create test
# service hastd onestart
```



Não é possível usar os provedores GEOM com um sistema de arquivos existente ou converter um armazenamento existente em um pool gerenciado por HAST. Esse procedimento precisa armazenar alguns metadados no provedor e não haverá espaço suficiente disponível em um provedor existente.

Um nó HAST `primário` ou `secundário` é selecionado por um administrador, ou software como Heartbeat, usando hastctl(8). No nó primário, `hasta`, execute este comando:

```
# hastctl role primary test
```

Execute este comando no nó secundário, `hastb`:

```
# hastctl role secondary test
```

Verifique o resultado executando `hastctl` em cada nó:

```
# hastctl status test
```

Verifique a linha `status` na saída. Se disser `degraded`, algo está errado com o arquivo de configuração. Ele deve dizer `complete` em cada nó, o que significa que a sincronização entre os nós foi iniciada. A sincronização é concluída quando `hastctl status` relata 0 bytes de extensões `sujas`.

O próximo passo é criar um sistema de arquivos no provedor GEOM e montá-lo. Isso deve ser feito no nó `primário`. A criação do sistema de arquivos pode levar alguns minutos, dependendo do tamanho do disco rígido. Este exemplo cria um sistema de arquivos UFS em /dev/hast/test:

```
# newfs -U /dev/hast/test
# mkdir /hast/test
# mount /dev/hast/test /hast/test
```

Uma vez que o framework HAST esteja configurado corretamente, o passo final é garantir que o

HAST seja iniciado automaticamente durante a inicialização do sistema. Adicione esta linha ao /etc/rc.conf:

```
hastd_enable="YES"
```

17.14.2.1. Configuração de Failover

O objetivo deste exemplo é construir um sistema de armazenamento robusto que seja resistente à falha de qualquer nó. Se o nó primário falhar, o nó secundário estará lá para assumir o controle, verificar e montar o sistema de arquivos e continuar a trabalhar sem perder um único bit de dados.

Para realizar essa tarefa, o Protocolo de Redundância de Endereços Comuns (CARP) é usado para fornecer failover automático na camada IP. O CARP permite que vários hosts no mesmo segmento de rede compartilhem um endereço IP. Configure o CARP em ambos os nós do cluster de acordo com a documentação disponível em Protocolo Comum de Redundância de Endereços (CARP). Neste exemplo, cada nó terá seu próprio endereço de gerenciamento IP e um endereço IP compartilhado de *172.16.0.254*. O nó principal HAST do cluster deve ser o nó mestre CARP.

O pool HAST criado na seção anterior está agora pronto para ser exportado para os outros hosts da rede. Isso pode ser feito exportando-o através do NFS ou Samba, usando o endereço IP_172.16.0.254_ compartilhado. O único problema que permanece não resolvido é um failover automático caso o nó primário falhe.

Caso as interfaces do CARP subam ou desçam, o sistema operacional FreeBSD gera um evento devd(8), tornando possível observar mudanças de estado nas interfaces do CARP. Uma alteração de estado na interface CARP é uma indicação de que um dos nós falhou ou voltou a ficar online. Esses eventos de mudança de estado tornam possível executar um script que manipulará automaticamente o failover HAST.

Para capturar mudanças de estado nas interfaces do CARP, adicione esta configuração ao /etc/devd.conf em cada nó:

```
notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_UP";
    action "/usr/local/sbin/carp-hast-switch master";
};

notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_DOWN";
    action "/usr/local/sbin/carp-hast-switch slave";
};
```



Se os sistemas estiverem executando o FreeBSD 10 ou superior, substitua carp0

> pelo nome da interface configurada CARP.

Reinicie o devd(8) em ambos os nós para colocar a nova configuração em vigor:

```
# service devd restart
```

Quando o estado da interface especificada é alterado subindo ou descendo, o sistema gera uma notificação, permitindo que o subsistema devd(8) execute o script de failover automático especificado, /usr/local/sbin/carp-hast-switch. Para maiores esclarecimentos sobre esta configuração, consulte devd.conf(5).

Aqui está um exemplo de um script de failover automatizado:

```
#!/bin/sh

# Original script by Freddie Cash <fjwcash@gmail.com>
# Modified by Michael W. Lucas <mwlucas@BlackHelicopters.org>
# and Viktor Petersson <vpetersson@wireload.net>

# The names of the HAST resources, as listed in /etc/hast.conf
resources="test"

# delay in mounting HAST resource after becoming master
# make your best guess
delay=3

# logging
log="local0.debug"
name="carp-hast"

# end of user configurable stuff

case "$1" in
    master)
        logger -p $log -t $name "Switching to primary provider for ${resources}."
        sleep ${delay}

        # Wait for any "hastd secondary" processes to stop
        for disk in ${resources}; do
            while $( pgrep -lf "hastd: ${disk} \(secondary\)" > /dev/null 2>&1 ); do
                sleep 1
            done

            # Switch role for each disk
            hastctl role primary ${disk}
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Unable to change role to primary for resource
${disk}."
                exit 1
```

```
            fi
        done

        # Wait for the /dev/hast/* devices to appear
        for disk in ${resources}; do
            for I in $( jot 60 ); do
                [ -c "/dev/hast/${disk}" ] && break
                sleep 0.5
            done

            if [ ! -c "/dev/hast/${disk}" ]; then
                logger -p $log -t $name "GEOM provider /dev/hast/${disk} did not
appear."
                exit 1
            fi
        done

        logger -p $log -t $name "Role for HAST resources ${resources} switched to
primary."

        logger -p $log -t $name "Mounting disks."
        for disk in ${resources}; do
            mkdir -p /hast/${disk}
            fsck -p -y -t ufs /dev/hast/${disk}
            mount /dev/hast/${disk} /hast/${disk}
        done

    ;;

    slave)
        logger -p $log -t $name "Switching to secondary provider for ${resources}."

        # Switch roles for the HAST resources
        for disk in ${resources}; do
            if ! mount | grep -q "^/dev/hast/${disk} on "
            then
            else
                umount -f /hast/${disk}
            fi
            sleep $delay
            hastctl role secondary ${disk} 2>&1
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Unable to switch role to secondary for
resource ${disk}."
                exit 1
            fi
            logger -p $log -t $name "Role switched to secondary for resource ${disk}."
        done
    ;;
esac
```

Em poucas palavras, o script executa essas ações quando um nó se torna mestre:

- Promove o pool de HAST para primário no outro nó.
- Verifica o sistema de arquivos no pool HAST.
- Monta o pool.

Quando um nó se torna secundário:

- Desmonta o conjunto HAST.
- Degrada o pool HAST para secundário.



Este é apenas um script de exemplo que serve como prova de conceito. Ele não manipula todos os cenários possíveis e pode ser estendido ou alterado de qualquer forma, por exemplo, para iniciar ou interromper os serviços necessários.



Para este exemplo, foi utilizado um sistema de arquivos padrão UFS. Para reduzir o tempo necessário para a recuperação, é possível usar um sistema de arquivos UFS ou ZFS com journal ativado.

Informações mais detalhadas com exemplos adicionais podem ser encontradas em http://wiki.FreeBSD.org/HAST.

17.14.3. Solução de problemas

O HAST geralmente deve funcionar sem problemas. No entanto, como acontece com qualquer outro produto de software, pode haver momentos em que ele não funciona como deveria. As origens dos problemas podem ser diferentes, mas a regra geral é garantir que o horário esteja sincronizado entre os nós do cluster.

Quando estiver fazendo troubleshooting no HAST, o nível de depuração de hastd(8) deve ser aumentado iniciando `hastd` com `-d`. Esse argumento pode ser especificado várias vezes para aumentar ainda mais o nível de depuração. Considere também usar `-F`, que inicia o `hastd` em primeiro plano.

17.14.3.1. Recuperando-se da Condição de Split-brain

Split-brain ocorre quando os nós do cluster não conseguem se comunicar entre si e ambos são configurados como primários. Esta é uma condição perigosa porque permite que ambos os nós façam alterações incompatíveis nos dados. Esse problema deve ser corrigido manualmente pelo administrador do sistema.

O administrador deve decidir qual nó tem alterações mais importantes ou executar a mesclagem manualmente. Então, deixe o HAST executar a sincronização completa do nó que possui os dados quebrados. Para fazer isso, emita esses comandos no nó que precisa ser ressincronizado:

```
# hastctl role init test
# hastctl create test
```

```
# hastctl role secondary test
```

Capítulo 18. GEOM: Framework de Transformação de Disco Modular

18.1. Sinopse

No FreeBSD, o framework GEOM permite acesso e controle à classes, tais como Master Boot Records e labels BSD, através do uso de provedores, ou dos dispositivos de disco em /dev. Ao suportar várias configurações de RAID via software, o GEOM fornece, de forma transparente, acesso ao sistema operacional e aos utilitários do sistema operacional.

Este capítulo aborda o uso de discos sob o framework do GEOM no FreeBSD. Isso inclui os principais utilitários de controle RAID os quais usam o framework para configuração. Este capítulo não é um guia definitivo para as configurações de RAID e somente as classificações de RAID suportadas pelo GEOM são discutidas.

Depois de ler este capítulo, você saberá:

- Que tipo de suporte a RAID está disponível através do GEOM.
- Como usar os utilitários da base para configurar, manter e manipular os vários níveis de RAID.
- Como espelhar, distribuir, criptografar e conectar remotamente dispositivos de disco por meio do GEOM.
- Como solucionar problemas de discos conectados ao framework do GEOM.

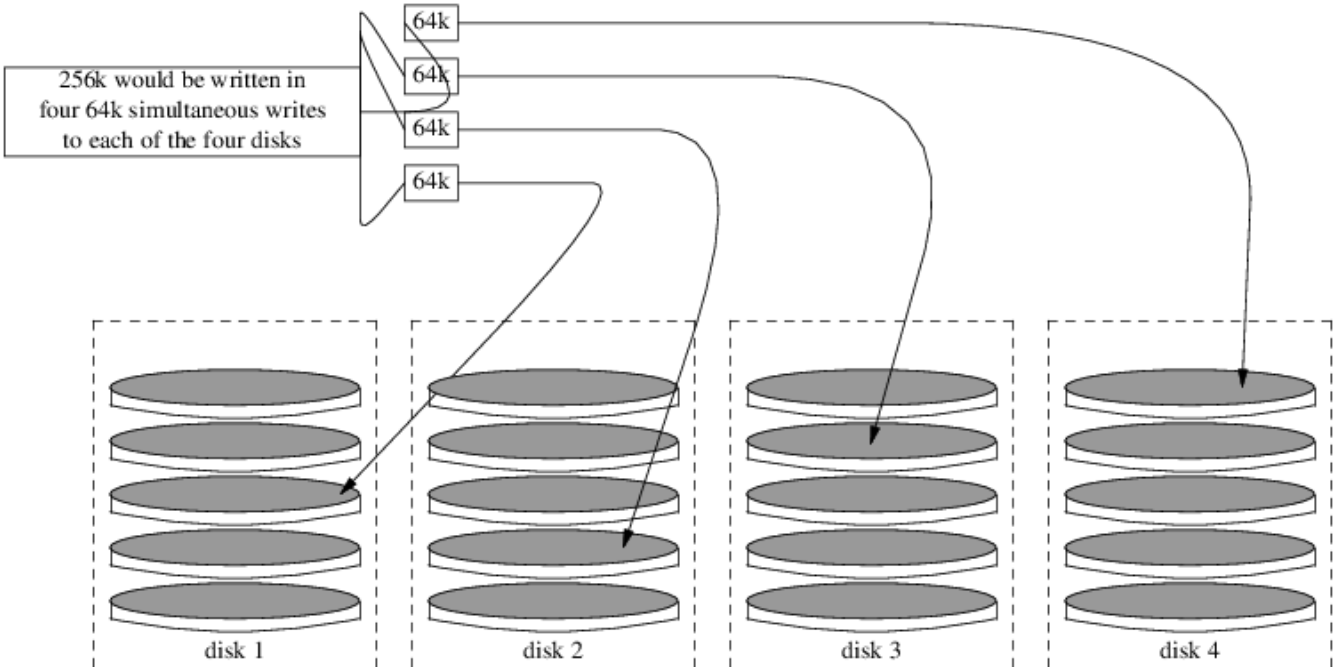
Antes de ler este capítulo, você deve:

- Entender como o FreeBSD trata os dispositivos de disco (Armazenamento).
- Saber como configurar e instalar um novo kernel (Configurando o kernel do FreeBSD).

18.2. RAID0 - Striping

O striping combina várias unidades de disco em um único volume. O striping pode ser realizado através do uso de hardwares controladores de RAID. O subsistema de disco GEOM fornece suporte de software para striping de disco, também conhecido como RAID0, sem a necessidade de um controlador RAID de disco.

No RAID0, os dados são divididos em blocos que são gravados em todas as unidades do array. Como pode ser visto na ilustração a seguir, em vez de esperar no sistema para gravar 256k em um disco, o RAID0 pode gravar 64k simultaneamente em cada um dos quatro discos do array, oferecendo um desempenho de I/O superior. Esse desempenho pode ser aprimorado ainda mais usando vários controladores de disco.



Cada disco em um stripe RAID0 deve ser do mesmo tamanho, pois as solicitações de I/O são intercaladas para ler ou gravar em vários discos em paralelo.



O RAID0 *não* fornece qualquer redundância. Isso significa que, se um disco no array falhar, todos os dados nos discos serão perdidos. Se os dados forem importantes, implemente uma estratégia de backup que salva regularmente os backups em um sistema ou dispositivo remoto.

O processo para criar um RAID0 por software, baseado no GEOM, em um sistema FreeBSD usando discos comuns é o seguinte. Uma vez que o stripe tiver sido criado, consulte gstripe(8) para obter maioress informações sobre como controlar uma stripe existente.

Procedure: Criando um Stripe de Discos ATA Não Formatados

1. Carregue o módulo geom_stripe.ko:

```
# kldload geom_stripe
```

2. Assegure-se de que exista um ponto de montagem adequado. Se esse volume se tornar uma partição root, use temporariamente outro ponto de montagem, como /mnt.
3. Determine os nomes dos dispositivos para os discos que serão striped e crie o novo dispositivo de stripe. Por exemplo, para distribuir dois discos ATA não utilizados e não particionados com nomes de dispositivos /dev/ad2 e /dev/ad3:

```
# gstripe label -v st0 /dev/ad2 /dev/ad3
Metadata value stored on /dev/ad2.
Metadata value stored on /dev/ad3.
Done.
```

4. Escreva um label padrão, também conhecido como tabela de partição, no novo volume e instale o código do bootstrap padrão:

```
# bsdlabel -wB /dev/stripe/st0
```

5. Este processo deve criar dois outros dispositivos em /dev/stripe além de st0. Esses incluem o st0a e o st0c. Neste ponto, um sistema de arquivos UFS pode ser criado no st0a usando o `newfs`:

```
# newfs -U /dev/stripe/st0a
```

   Muitos números irão deslizar pela tela e, após alguns segundos, o processo será concluído. O volume foi criado e está pronto para ser montado.

6. Para montar manualmente o stripe de disco criado:

```
# mount /dev/stripe/st0a /mnt
```

7. Para montar este sistema de arquivos distribuído automaticamente durante o processo de inicialização, coloque as informações do volume no arquivo /etc/fstab. Neste exemplo, um ponto de montagem permanente, chamado stripe, é criado:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /stripe ufs rw 2 2" \
>> /etc/fstab
```

8. O módulo geom_stripe.ko também deve ser carregado automaticamente durante a inicialização do sistema, adicionando uma linha ao arquivo /boot/loader.conf:

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

18.3. RAID1 - Espelhamento

O RAID1, ou *espelhamento*, é a técnica de gravar os mesmos dados em mais de uma unidade de disco. Os espelhos são geralmente usados para proteger contra perda de dados devido a falhas na unidade. Cada unidade espelhada contém uma cópia idêntica dos dados. Quando uma unidade individual falha, o espelhamento continua a funcionar, fornecendo dados a partir das unidades que ainda estão funcionando. O computador continua funcionando e o administrador tem tempo para substituir a unidade com falha sem impactar o usuário.

Duas situações comuns são ilustradas nesses exemplos. O primeiro cria um espelhamento de dois novos discos e usa-o como um substituto para um único disco existente. O segundo exemplo cria um espelho em um único disco novo, copia os dados do disco antigo para ele e insere o disco antigo

no espelho. Embora esse procedimento seja um pouco mais complicado, ele requer apenas um novo disco.

Tradicionalmente, os dois discos em um espelhamento são idênticos em modelo e capacidade, mas o gmirror(8) não requer isso. Os espelhamentos criados com discos diferentes terão uma capacidade igual à da menor unidade no espelhamento. O espaço extra em discos maiores não será usado. Os discos inseridos posteriormente no espelhamento devem ter pelo menos a mesma capacidade que o menor disco já existente no espelhamento.



Os procedimentos de espelhamento mostrados aqui são não-destrutivos, mas como em qualquer grande operação de disco, faça um backup completo primeiro.



Embora o dump(8) seja usado nesses procedimentos para copiar sistemas de arquivos, ele não funciona em sistemas de arquivos com Soft Updates Journaling. Consulte o tunefs(8) para obter informações sobre como detectar e desativar o Soft Updates Journaling.

18.3.1. Problemas de Metadados

Muitos sistemas de disco armazenam metadados no final de cada disco. Metadados antigos devem ser apagados antes de reutilizar o disco em um espelhamento. A maioria dos problemas é causada por dois tipos particulares de metadados residuais: tabelas de partição GPT e metadados antigos de um espelhamento anterior.

Os metadados GPT podem ser apagados com gpart(8). Este exemplo apaga as tabelas de partições primárias e de backup do GPT do disco ada8:

```
# gpart destroy -F ada8
```

Um disco pode ser removido de um espelhamento ativo e os metadados apagados em uma etapa usando gmirror(8). Aqui, o disco de exemplo ada8 é removido do espelhamento ativo gm4:

```
# gmirror remove gm4 ada8
```

Se o espelhamento não estiver em execução, mas os metadados do espelhamento antigo ainda estiverem no disco, use o comando `gmirror clear` para removê-lo:

```
# gmirror clear ada8
```

O gmirror(8) armazena um bloco de metadados no final do disco. Como os esquemas de partição GPT também armazenam metadados no final do disco, espelhar discos GPT inteiros com gmirror(8) não é recomendado. O particionamento MBR é usado aqui porque armazena apenas uma tabela de partição no início do disco e não entra em conflito com os metadados espelhados.

18.3.2. Criando um Espelhamento com Dois Discos Novos

Neste exemplo, o FreeBSD já foi instalado em um único disco, ada0. Dois novos discos, ada1 e ada2, foram conectados ao sistema. Um novo espelhamento será criado nesses dois discos e usado para substituir o antigo disco único.

O módulo do kernel geom_mirror.ko deve ser compilado no kernel ou carregado no boot ou em tempo de execução. Carregue manualmente o módulo do kernel agora:

```
# gmirror load
```

Crie o espelho com as duas novas unidades:

```
# gmirror label -v gm0 /dev/ada1 /dev/ada2
```

O gm0 é um nome de dispositivo escolhido pelo usuário atribuído ao novo espelhamento. Depois que o espelhamento for iniciado, o nome desse dispositivo aparecerá em /dev/mirror/.

As tabelas de partição MBR e bsdlabel agora podem ser criadas no mirror com o gpart(8). Este exemplo usa um layout de sistema de arquivos tradicional, com partições para /, swap, /var, /tmp e /usr. Um único / e uma partição swap também funcionarão.

As partições no espelho não precisam ser do mesmo tamanho que as do disco existente, mas devem ser grandes o suficiente para conter todos os dados já presentes no disco ada0.

```
# gpart create -s MBR mirror/gm0
# gpart add -t freebsd -a 4k mirror/gm0
# gpart show mirror/gm0
=>       63  156301423  mirror/gm0  MBR  (74G)
         63         63                  - free -  (31k)
        126  156301299                1  freebsd  (74G)
  156301425         61                  - free -  (30k)
```

```
# gpart create -s BSD mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-swap -a 4k -s 4g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 1g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k       mirror/gm0s1
# gpart show mirror/gm0s1
=>        0  156301299  mirror/gm0s1  BSD  (74G)
          0          2                     - free -  (1.0k)
          2    4194304                   1  freebsd-ufs  (2.0G)
    4194306    8388608                   2  freebsd-swap  (4.0G)
   12582914    4194304                   4  freebsd-ufs  (2.0G)
   16777218    2097152                   5  freebsd-ufs  (1.0G)
   18874370  137426928                   6  freebsd-ufs  (65G)
```

```
  156301298          1                    - free -  (512B)
```

Torne o espelhamento inicializável instalando o bootcode no MBR e no bsdlabel e definindo a slice ativa:

```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Formate os sistemas de arquivos no novo espelhamento, habilitando as atualizações simples.

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
```

Os sistemas de arquivos do disco original ada0 agora podem ser copiados para o espelho com o dump(8) e o restore(8).

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/tmp
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /tmp | (cd /mnt/tmp && restore -rf -)
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
```

Edite o arquivo /mnt/etc/fstab para apontar para os novos sistemas de arquivos espelhados:

```
# Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /tmp        ufs rw  2   2
/dev/mirror/gm0s1f  /usr        ufs rw  2   2
```

Se o módulo do kernel geom_mirror.ko não foi compilado no kernel, o /mnt/boot/loader.conf é editado para carregar o módulo na inicialização:

```
geom_mirror_load="YES"
```

Reinicialize o sistema para testar o novo espelhamento e verifique se todos os dados foram copiados. A BIOS verá o espelhamento como duas unidades individuais em vez de um

espelhamento. Como as unidades são idênticas, não importa qual seja selecionado para inicializar.

Veja Solução de problemas se houver problemas ao inicializar. Desligar e desconectar o disco original ada0 permitirá que ele seja mantido como um backup offline.

Em uso, o espelhamento se comportará exatamente como a unidade original.

18.3.3. Criando um Espelhamento com Uma Unidade Existente

Neste exemplo, o FreeBSD já foi instalado em um único disco, ada0. Um novo disco, ada1, foi conectado ao sistema. Um espelhamento de um disco será criado no novo disco, o sistema existente será copiado para ele e, em seguida, o disco antigo será inserido no espelho. Esse procedimento um pouco complexo é necessário porque o `gmirror` precisa colocar um bloco de metadados de 512 bytes no final de cada disco, e o ada0 geralmente possui todo o seu espaço já alocado.

Carregue o módulo do kernel geom_mirror.ko:

```
# gmirror load
```

Verifique o tamanho da mídia do disco original com `diskinfo`:

```
# diskinfo -v ada0 | head -n3
/dev/ada0
    512             # sectorsize
    1000204821504   # mediasize in bytes (931G)
```

Crie um espelhamento no novo disco. Para garantir que a capacidade do espelhamento não seja maior do que a unidade ada0 original, gnop(8) é usado para criar uma unidade falsa exatamente do mesmo tamanho. Esta unidade não armazena dados, mas é usada apenas para limitar o tamanho do espelhamento. Quando o gmirror(8) cria o espelhamento, ele irá restringir a capacidade ao tamanho de gzero.nop, mesmo se a nova unidade ada1 tiver mais espaço. Note que o *1000204821504* na segunda linha é igual ao tamanho de mídia do ada0 como mostrado pelo comando `diskinfo` acima.

```
# geom zero load
# gnop create -s 1000204821504 gzero
# gmirror label -v gm0 gzero.nop ada1
# gmirror forget gm0
```

Como o gzero.nop não armazena nenhum dado, o espelhamento não o vê como conectado. É dito para o espelhamento "esquecer" os componentes desconectados, removendo referências para gzero.nop. O resultado é um dispositivo espelhado contendo apenas um único disco, ada1.

Depois de criar o gm0, veja a tabela de partições em ada0. Esta saída é de uma unidade de 1 TB. Se houver algum espaço não alocado no final da unidade, o conteúdo pode ser copiado diretamente de ada0 para o novo espelho.

No entanto, se a saída mostrar que todo o espaço no disco está alocado, como na listagem a seguir, não há espaço disponível para os 512-bytes de metadados de espelhamento no final do disco.

```
# gpart show ada0
=>        63  1953525105        ada0  MBR  (931G)
          63  1953525105           1  freebsd  [active]  (931G)
```

Neste caso, a tabela de partição deve ser editada para reduzir a capacidade de um setor em mirror/gm0. O procedimento será explicado mais tarde.

Em qualquer um dos casos, as tabelas de partição no disco principal devem ser primeiro copiadas usando `gpart backup` e `gpart restore`.

```
# gpart backup ada0 > table.ada0
# gpart backup ada0s1 > table.ada0s1
```

Esses comandos criam dois arquivos, table.ada0 e table.ada0s1. Este exemplo é de uma unidade de 1 TB:

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525105   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
7  freebsd-ufs  968884224  984640881
```

Se nenhum espaço livre for exibido no final do disco, o tamanho da slice e da última partição deve ser reduzido por um setor. Edite os dois arquivos, reduzindo o tamanho da fatia e da última partição em um. Estes são os últimos números em cada listagem.

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525104   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
```

```
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
7  freebsd-ufs  968884224  984640880
```

Se pelo menos um setor não foi alocado no final do disco, esses dois arquivos podem ser usados sem modificação.

Agora restaure a tabela de partições em mirror/gm0:

```
# gpart restore mirror/gm0 < table.ada0
# gpart restore mirror/gm0s1 < table.ada0s1
```

Verifique a tabela de partições com o comando `gpart show`. Este exemplo tem gm0s1a para /, gm0s1d para /var, gm0s1e para /usr, gm0s1f para /data1 e gm0s1g para /data2.

```
# gpart show mirror/gm0
=>        63  1953525104  mirror/gm0  MBR  (931G)
          63  1953525042           1  freebsd  [active]  (931G)
  1953525105          62              - free -  (31k)

# gpart show mirror/gm0s1
=>         0  1953525042  mirror/gm0s1  BSD  (931G)
           0     2097152             1  freebsd-ufs  (1.0G)
     2097152    16777216             2  freebsd-swap  (8.0G)
    18874368    41943040             4  freebsd-ufs  (20G)
    60817408    20971520             5  freebsd-ufs  (10G)
    81788928   629145600             6  freebsd-ufs  (300G)
   710934528  1242590514             7  freebsd-ufs  (592G)
  1953525042          63                - free -  (31k)
```

Tanto a fatia quanto a última partição devem ter pelo menos um bloco livre no final do disco.

Crie sistemas de arquivos nessas novas partições. O número de partições varia de acordo com o disco original, ada0.

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
# newfs -U /dev/mirror/gm0s1g
```

Torne o espelhamento inicializável instalando o bootcode no MBR e no bsdlabel e definindo a slice ativa:

```
# gpart bootcode -b /boot/mbr mirror/gm0
```

```
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Ajuste o arquivo /etc/fstab para usar as novas partições no espelhamento.Primeiro faça o backup deste arquivo copiando ele para /etc/fstab.orig.

```
# cp /etc/fstab /etc/fstab.orig
```

Edite o arquivo /etc/fstab, substituindo /dev/ada0 por mirror/gm0.

```
# Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /usr        ufs rw  2   2
/dev/mirror/gm0s1f  /data1      ufs rw  2   2
/dev/mirror/gm0s1g  /data2      ufs rw  2   2
```

Se o módulo do kernel geom_mirror.ko não foi carregado no kernel, edite o arquivo /boot/loader.conf para carregá-lo no boot:

```
geom_mirror_load="YES"
```

Os sistemas de arquivos do disco original agora podem ser copiados para o espelhamento com o dump(8) e o restore(8). Cada sistema de arquivos copiados com o `dump -L` irá primeiro criar um snapshot, o que pode levar algum tempo.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - /    | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/usr
# mount /dev/mirror/gm0s1f /mnt/data1
# mount /dev/mirror/gm0s1g /mnt/data2
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /data1 | (cd /mnt/data1 && restore -rf -)
# dump -C16 -b64 -0aL -f - /data2 | (cd /mnt/data2 && restore -rf -)
```

Reinicie o sistema, inicializando a partir do ada1. Se tudo estiver funcionando, o sistema irá inicializar a partir de mirror/gm0, que agora contém os mesmos dados que o ada0 tinha anteriormente. Veja Solução de problemas se houver problemas ao inicializar.

Neste ponto, o espelhamento ainda consiste apenas no único disco ada1.

Após inicializar a partir de mirror/gm0 com sucesso, a etapa final é inserir ada0 no espelhamento.



Quando o ada0 for inserido no espelhamento, seu conteúdo anterior será substituído pelos dados do espelhamento. Certifique-se de que mirror/gm0 tenha o mesmo conteúdo do ada0 antes de adicionar o ada0 ao espelhamento. Se o conteúdo anteriormente copiado pelo dump(8) e restore(8) não forem idênticos ao que estava em ada0, reverta o arquivo /etc/fstab para montar os sistemas de arquivos em ada0, e reinicie todo o procedimento novamente.

```
# gmirror insert gm0 ada0
GEOM_MIRROR: Device gm0: rebuilding provider ada0
```

A sincronização entre os dois discos será iniciada imediatamente. Use `gmirror status` para visualizar o progresso.

```
# gmirror status
      Name    Status  Components
mirror/gm0  DEGRADED  ada1 (ACTIVE)
                      ada0 (SYNCHRONIZING, 64%)
```

Depois de um tempo, a sincronização será concluída.

```
GEOM_MIRROR: Device gm0: rebuilding provider ada0 finished.
# gmirror status
      Name    Status  Components
mirror/gm0  COMPLETE  ada1 (ACTIVE)
                      ada0 (ACTIVE)
```

O mirror/gm0 agora consiste de dois discos ada0 e ada1, e o conteúdo é automaticamente sincronizado entre eles. Em uso, o mirror/gm0 irá se comportar como a única unidade original.

18.3.4. Solução de problemas

Se o sistema não inicializar mais, as configurações da BIOS podem ter que ser alteradas para inicializar a partir de uma das novas unidades espelhadas. Qualquer uma das unidades espelhadas pode ser usada para inicializar, pois elas contêm dados idênticos.

Se a inicialização parar com esta mensagem, algo está errado com o dispositivo espelhado:

```
Mounting from ufs:/dev/mirror/gm0s1a failed with error 19.

Loader variables:
  vfs.root.mountfrom=ufs:/dev/mirror/gm0s1a
  vfs.root.mountfrom.options=rw

Manual root filesystem specification:
  <fstype>:<device> [options]
      Mount <device> using filesystem <fstype>
```

```
      and with the specified (optional) option list.

    eg. ufs:/dev/da0s1a
        zfs:tank
        cd9660:/dev/acd0 ro
          (which is equivalent to: mount -t cd9660 -o ro /dev/acd0 /)

  ?               List valid disk boot devices
  .               Yield 1 second (for background tasks)
  <empty line>    Abort manual input

mountroot>
```

Esquecer de carregar o módulo geom_mirror.ko no arquivo /boot/loader.conf pode causar este problema. Para consertá-lo, inicialize a partir de uma mídia de instalação do FreeBSD e escolha `Shell` no primeiro prompt. Em seguida, carregue o módulo de espelhamento e monte o dispositivo espelhado:

```
# gmirror load
# mount /dev/mirror/gm0s1a /mnt
```

Edite o arquivo /mnt/boot/loader.conf, adicionando uma linha para carregar o módulo de espelhamento:

```
geom_mirror_load="YES"
```

Salve o arquivo e reinicie.

Outros problemas que causam o `error 19` requerem mais esforço para serem corrigidos. Embora o sistema deva inicializar a partir de ada0, outro prompt para selecionar um shell aparecerá se o arquivo /etc/fstab estiver incorreto. Digite `ufs:/dev/ada0s1a` no prompt do carregador de boot e pressione `Enter`. Desfaça as edições no arquivo /etc/fstab e monte os sistemas de arquivos a partir do disco original (ada0) em vez do espelhado. Reinicialize o sistema e tente o procedimento novamente.

```
Enter full pathname of shell or RETURN for /bin/sh:
# cp /etc/fstab.orig /etc/fstab
# reboot
```

18.3.5. Recuperando de Uma Falha de Disco

O benefício do espelhamento de disco é que um disco individual pode falhar sem fazer com que o espelho perca qualquer dado. No exemplo acima, se ada0 falhar, o espelho continuará funcionando, fornecendo dados a partir do disco que continua operacional, ada1.

Para substituir a unidade com falha, desligue o sistema e substitua fisicamente a unidade com falha

por uma nova unidade com capacidade igual ou maior. Os fabricantes usam valores um tanto arbitrários ao classificar drives em gigabytes, e a única maneira de realmente ter certeza é comparar a contagem total de setores mostrados por `diskinfo -v`. Uma unidade com maior capacidade que o espelho funcionará, embora o espaço extra na nova unidade não seja usado.

Depois que o computador for ligado novamente, o espelho será executado em um modo "degradado" com apenas uma unidade. O espelho é avisado para esquecer as unidades que não estão conectadas no momento:

```
# gmirror forget gm0
```

Quaisquer metadados antigos devem ser apagados do disco de substituição usando as instruções em Problemas de Metadados. Em seguida, o disco de substituição, ada4 para este exemplo, é inserido no espelho:

```
# gmirror insert gm0 /dev/ada4
```

A ressincronização começa quando a nova unidade é inserida no espelho. Esse processo de copiar dados espelhados para uma nova unidade pode demorar um pouco. O desempenho do espelho será bastante reduzido durante a cópia, portanto, a inserção de novos discos é deve ser executada quando houver pouca demanda no computador.

O progresso pode ser monitorado com o comando `gmirror status`, que mostra as unidades que estão sendo sincronizadas e a porcentagem de conclusão. Durante a ressincronização, o status será `DEGRADED`, mudando para `COMPLETE` quando o processo for concluído.

18.4. RAID3 - Distribuição em Nível de Byte com Paridade Dedicada

O RAID3 é um método usado para combinar várias unidades de disco em um único volume com um disco de paridade dedicado. Em um sistema RAID3, os dados são divididos em vários bytes que são escritos em todas as unidades da matriz, exceto por um disco que atua como um disco de paridade dedicado. Isso significa que as leituras de disco de uma implementação de RAID3 acessam todos os discos na matriz. O desempenho pode ser aprimorado usando vários controladores de disco. O array RAID3 fornece uma tolerância a falhas de 1 unidade, enquanto fornece uma capacidade de 1 - 1/n vezes a capacidade total de todas as unidades no array, onde n é o número de unidades de disco rígido no array. Essa configuração é adequada principalmente para armazenar dados de tamanhos maiores, como arquivos multimídia.

Pelo menos 3 discos rígidos físicos são necessários para criar um array RAID3. Cada disco deve ter o mesmo tamanho, pois as solicitações de I/O são intercaladas para ler ou gravar em vários discos em paralelo. Além disso, devido à natureza do RAID3, o número de unidades deve ser igual a 3, 5, 9, 17 e assim por diante, ou 2^n + 1.

Esta seção demonstra como criar um RAID3 via software em um sistema FreeBSD.



Embora seja teoricamente possível inicializar a partir de um array RAID3 no FreeBSD, essa configuração é incomum e não é recomendada.

18.4.1. Criando uma Matriz RAID3 Dedicada

No FreeBSD, o suporte para RAID3 é implementado pela classe GEOMgraid3(8). Criar um array dedicado de RAID3 no FreeBSD requer os seguintes passos.

1. Primeiro, carregue o módulo do kernel geom_raid3.ko emitindo um dos seguintes comandos:

   ```
   # graid3 load
   ```

   ou:

   ```
   # kldload geom_raid3
   ```

2. Assegure-se de que exista um ponto de montagem adequado. Este comando cria um novo diretório para usar como ponto de montagem:

   ```
   # mkdir /multimedia
   ```

3. Determine os nomes dos dispositivos para os discos que serão adicionados à matriz e crie o novo dispositivo RAID3. O dispositivo final listado atuará como o disco de paridade dedicado. Este exemplo usa três unidades ATA não-particionadas: ada1 e ada2 para dados e ada3 para paridade.

   ```
   # graid3 label -v gr0 /dev/ada1 /dev/ada2 /dev/ada3
   Metadata value stored on /dev/ada1.
   Metadata value stored on /dev/ada2.
   Metadata value stored on /dev/ada3.
   Done.
   ```

4. Particione o dispositivo gr0 recém-criado e coloque um sistema de arquivos UFS:

   ```
   # gpart create -s GPT /dev/raid3/gr0
   # gpart add -t freebsd-ufs /dev/raid3/gr0
   # newfs -j /dev/raid3/gr0p1
   ```

   Muitos números irão ser exibios na tela e, após algum tempo, o processo será concluído. O volume foi criado e está pronto para ser montado:

```
# mount /dev/raid3/gr0p1 /multimedia/
```

A matriz RAID3 está agora pronta para uso.

Uma configuração adicional é necessária para manter essa configuração nas reinicializações do sistema.

1. O módulo geom_raid3.ko deve ser carregado antes que o array possa ser montado. Para carregar automaticamente o módulo do kernel durante a inicialização do sistema, adicione a seguinte linha ao arquivo /boot/loader.conf:

   ```
   geom_raid3_load="YES"
   ```

2. As seguintes informações de volume devem ser adicionadas ao arquivo /etc/fstab para montar automaticamente o sistema de arquivos do array durante o processo de inicialização do sistema:

   ```
   /dev/raid3/gr0p1    /multimedia ufs rw  2   2
   ```

18.5. Dispositivos RAID por Software

Algumas placas-mãe e placas de expansão adicionam um hardware simples, geralmente apenas uma ROM, que permite que o computador inicialize a partir de um array RAID. Após a inicialização, o acesso ao array RAID é feito pelo software em execução no processador principal do computador. Este "RAID via software assistido por hardware" fornece arrays RAID que não dependem de nenhum sistema operacional em particular, e que são funcionais antes mesmo de um sistema operacional ser carregado.

Vários níveis de RAID são suportados, dependendo do hardware em uso. Veja graid(8) para uma lista completa.

O graid(8) requer o módulo do kernel geom_raid.ko, que está incluído no kernel GENERIC a partir do FreeBSD 9.1. Se necessário, ele pode ser carregado manualmente com o comando `graid load`.

18.5.1. Criando um Array

Os dispositivos de RAID via software geralmente têm um menu que pode ser acessado pressionando teclas especiais quando o computador está inicializando. O menu pode ser usado para criar e excluir arrays RAID. O graid(8) também pode criar arrays diretamente a partir da linha de comando.

O `graid label` é usado para criar um novo array. A placa-mãe usada neste exemplo tem um chipset RAID da Intel, portanto, o formato de metadados da Intel é especificado. A nova matriz recebe um

rótulo de gm0, é um espelhamento (RAID1) e usa as unidades ada0 e ada1.



Algum espaço nas unidades será sobrescrito quando elas forem transformadas em um novo array. Faça o backup dos dados existentes primeiro!

```
# graid label Intel gm0 RAID1 ada0 ada1
GEOM_RAID: Intel-a29ea104: Array Intel-a29ea104 created.
GEOM_RAID: Intel-a29ea104: Disk ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:0-ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Array started.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from STARTING to OPTIMAL.
Intel-a29ea104 created
GEOM_RAID: Intel-a29ea104: Provider raid/r0 for volume gm0 created.
```

Uma verificação de status mostra que o novo espelhamento está pronto para uso:

```
# graid status
   Name   Status  Components
raid/r0  OPTIMAL  ada0 (ACTIVE (ACTIVE))
                  ada1 (ACTIVE (ACTIVE))
```

O dispositivo de array aparece em /dev/raid/. O primeiro array é chamado de r0. Arrays adicionais, se presentes, serão r1, r2 e assim por diante.

O menu da BIOS em alguns desses dispositivos pode criar arrays com caracteres especiais em seus nomes. Para evitar problemas com esses caracteres especiais, os arrays recebem nomes numerados simples como r0. Para mostrar os rótulos reais, como gm0 no exemplo acima, use o sysctl(8):

```
# sysctl kern.geom.raid.name_format=1
```

18.5.2. Múltiplos Volumes

Alguns dispositivos de RAID via software suportam mais de um *volume* em um array. Os volumes funcionam como partições, permitindo que o espaço nas unidades físicas seja dividido e usado de diferentes maneiras. Por exemplo, os dispositivos RAID via software Intel suportam dois volumes. Este exemplo cria um espelho de 40 G para armazenar com segurança o sistema operacional, seguido por um volume de 20 G RAID0 (stripe) para armazenamento temporário rápido:

```
# graid label -S 40G Intel gm0 RAID1 ada0 ada1
# graid add -S 20G gm0 RAID0
```

Os volumes aparecem como entradas adicionais rX em /dev/raid/. Um array com dois volumes mostrará r0 e r1.

Veja graid(8) para o número de volumes suportados por diferentes dispositivos RAID via software.

18.5.3. Convertendo uma Única Unidade em um Espelho

Sob certas condições específicas, é possível converter uma única unidade existente em um array graid(8) sem reformatar. Para evitar a perda de dados durante a conversão, a unidade existente deve atender a esses requisitos mínimos:

- A unidade deve ser particionada com o esquema de particionamento MBR. O GPT ou outros esquemas de particionamento com metadados no final da unidade serão sobrescritos e corrompidos pelos metadados do graid(8).
- Deve haver espaço não particionado e não utilizado o suficiente no final da unidade para conter os metadados do graid(8). Esses metadados variam em tamanho, mas o maior ocupa 64 M, então pelo menos este espaço livre é recomendado.

Se a unidade atender a esses requisitos, comece fazendo um backup completo. Em seguida, crie um espelhamento de unidade única com essa unidade:

```
# graid label Intel gm0 RAID1 ada0 NONE
```

Os metadados do graid(8) foram gravados no final da unidade no espaço não utilizado. Uma segunda unidade pode agora ser inserida no espelhamento:

```
# graid insert raid/r0 ada1
```

Os dados da unidade original começarão imediatamente a ser copiados para a segunda unidade. O espelhamento operará em status degradado até que a cópia seja concluída.

18.5.4. Inserindo Novos Discos no Array

As unidades podem ser inseridas em uma matriz como substitutos de unidades que falharam ou estão faltando. Se não houver unidades com falha ou ausentes, a nova unidade se tornará uma reserva. Por exemplo, inserir uma nova unidade em um espelhamento de duas unidades de trabalho resulta em um espelhamento de duas unidades com uma unidade sobressalente, não em um espelhamento de três unidades.

No array de espelho do exemplo, os dados começam a ser copiados imediatamente para a unidade recém-inserida. Qualquer informação existente na nova unidade será substituída.

```
# graid insert raid/r0 ada1
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to NEW.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NEW to REBUILD.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 rebuild start at 0.
```

18.5.5. Removendo Discos do Array

Discos individuais podem ser permanentemente removidos de um array e seus metadados apagados:

```
# graid remove raid/r0 ada1
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from ACTIVE to OFFLINE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-[unknown] state changed from ACTIVE to NONE.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from OPTIMAL to DEGRADED.
```

18.5.6. Parando o Array

Um array pode ser interrompido sem remover os metadados das unidades. O array será reiniciado quando o sistema for inicializado.

```
# graid stop raid/r0
```

18.5.7. Verificando o Status do Array

O status do array pode ser verificado a qualquer momento. Depois que um disco foi adicionado ao espelho no exemplo acima, os dados estarão sendo copiados do disco original para o novo disco:

```
# graid status
   Name    Status  Components
raid/r0  DEGRADED  ada0 (ACTIVE (ACTIVE))
                   ada1 (ACTIVE (REBUILD 28%))
```

Alguns tipos de arrays, como `RAID0` ou `CONCAT`, podem não ser mostrados no relatório de status se os discos falharem. Para ver esses arrays com falhas parciais, adicione `-ga`:

```
# graid status -ga
          Name  Status  Components
Intel-e2d07d9a  BROKEN  ada6 (ACTIVE (ACTIVE))
```

18.5.8. Excluindo Arrays

Arrays são destruídos, excluindo todos os volumes deles. Quando o último volume presente é excluído, o array é interrompido e os metadados são removidos dos discos:

```
# graid delete raid/r0
```

18.5.9. Excluindo Arrays Inesperados

Os discos podem conter metadados graid(8) inesperados, originados no seu uso anterior ou em testes do fabricante. O graid(8) detectará estes discos e criará um array, interferindo no acesso ao disco individual. Para remover os metadados indesejados:

1. Inicialize o sistema. No menu de inicialização, selecione `2` para o prompt do utilitário de boot. Entre:

   ```
   OK set kern.geom.raid.enable=0
   OK boot
   ```

   O sistema inicializará com o graid(8) desativado.

2. Fazer backup de todos os dados na unidade afetada.
3. Como solução alternativa, a detecção de arrays graid(8) pode ser desativada incluindo se a variável

   ```
   kern.geom.raid.enable=0
   ```

   no arquivo /boot/loader.conf.

   Para remover permanentemente os metadados graid(8) do disco afetado, inicialize uma instalação do FreeBSD usando um CD-ROM ou um memory stick e selecione a opção `Shell`. Use o comando `status` para encontrar o nome do array, normalmente `raid/r0`:

   ```
   # graid status
      Name   Status  Components
   raid/r0  OPTIMAL  ada0 (ACTIVE (ACTIVE))
                     ada1 (ACTIVE (ACTIVE))
   ```

   Exclua o volume pelo nome:

   ```
   # graid delete raid/r0
   ```

   Se houver mais de um volume exibido, repita o processo para cada volume. Após o último array ter sido excluído, o volume será destruído.

   Reinicialize e verifique os dados, restaurando a partir do backup, se necessário. Depois que os metadados forem removidos, a entrada `kern.geom.raid.enable=0` no arquivo /boot/loader.conf também pode ser removida.

18.6. GEOM Network Gate

O GEOM fornece um mecanismo simples para fornecer acesso remoto a dispositivos como discos, CDs e sistemas de arquivos através do uso do daemon GEOM Network Gate, ggated. O sistema com o dispositivo executa o daemon do servidor que manipula solicitações feitas por clientes usando o ggatec. Os dispositivos não devem conter dados confidenciais, pois a conexão entre o cliente e o servidor não é criptografada.

Semelhante ao NFS, que é discutido em Network File System (NFS), o ggated é configurado usando um arquivo de exportação. Este arquivo especifica quais sistemas têm permissão para acessar os recursos exportados e em qual nível de acesso eles são oferecidos. Por exemplo, para fornecer ao cliente `192.168.1.5` acesso de leitura e gravação à quarta slice do primeiro disco SCSI, crie o arquivo /etc/gg.exports com esta linha:

```
192.168.1.5 RW /dev/da0s4d
```

Antes de exportar o dispositivo, verifique se ele não está montado no momento. Em seguida, inicie o ggated:

```
# ggated
```

Várias opções estão disponíveis para especificar uma porta de escuta alternativa ou para alterar o local padrão do arquivo de exportação. Consulte ggated(8) para maiores detalhes.

Para acessar o dispositivo exportado na máquina cliente, primeiro use o comando `ggatec` para especificar o endereço IP do servidor e o nome do dispositivo exportado. Se bem sucedido, este comando irá exibir um nome de dispositivo `ggate` para montar. Monte esse nome de dispositivo especificado em um ponto de montagem livre. Este exemplo conecta-se à partição /dev/da0s4d no `192.168.1.1`, em seguida, monta o /dev/ggate0 em /mnt:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
ggate0
# mount /dev/ggate0 /mnt
```

O dispositivo no servidor pode agora ser acessado por meio do /mnt no cliente. Para maiores detalhes sobre o `ggatec` e alguns exemplos de uso, consulte ggatec(8).



A montagem falhará se o dispositivo estiver atualmente montado no servidor ou em qualquer outro cliente na rede. Se for necessário acesso simultâneo aos recursos de rede, use o NFS.

Quando o dispositivo não for mais necessário, desmonte-o com o `umount` para que o recurso fique disponível para outros clientes.

18.7. Rotulando Dispositivos de Disco

Durante a inicialização do sistema, o kernel do FreeBSD cria nós de dispositivos conforme os dispositivos são encontrados. Esse método de detectar dispositivos gera alguns problemas. Por exemplo, e se um novo dispositivo de disco for adicionado via USB? É provável que um dispositivo flash receba o nome do dispositivo da0 e o da0 original alterado para da1. Isso causará problemas ao montar sistemas de arquivos se eles estiverem listados no /etc/fstab, o que também pode impedir que o sistema seja inicializado.

Uma solução é encadear os dispositivos SCSI para que um novo dispositivo adicionado à placa SCSI receba números de dispositivo não utilizados. Mas e os dispositivos USB que podem substituir o disco principal SCSI? Isso acontece porque os dispositivos USB geralmente são examinados antes da placa SCSI. Uma solução é inserir esses dispositivos apenas após o sistema ter sido inicializado. Outro método é usar apenas uma única unidade ATA e nunca listar os dispositivos SCSI no arquivo /etc/fstab.

Uma solução melhor é usar o `glabel` para rotular os dispositivos de disco e usar os rótulos no arquivo /etc/fstab. Como o `glabel` armazena o rótulo no último setor de um determinado provedor, o rótulo permanecerá persistente nas reinicializações. Ao usar esse rótulo como um dispositivo, o sistema de arquivos pode sempre ser montado independentemente do nó do dispositivo pelo qual ele é acessado.



O `glabel` pode criar rótulos transitórios e permanentes. Somente rótulos permanentes são consistentes nas reinicializações. Consulte glabel(8) para obter mais informações sobre as diferenças entre os rótulos.

18.7.1. Tipos de Rótulos e Exemplos

Os rótulos permanentes podem ser um rótulo genérico ou de um sistema de arquivos. Rótulos de sistema de arquivos permanentes podem ser criados com tunefs(8) ou newfs(8). Esses tipos de rótulos são criados em um subdiretório /dev e serão nomeados de acordo com o tipo de sistema de arquivos. Por exemplo, os rótulos do sistema de arquivos UFS2 serão criados em /dev/ufs. Rótulos permanentes genéricos podem ser criados com o `glabel label`. Estes não são específicos do sistema de arquivos e serão criados em /dev/label.

Os rótulos temporários são destruídos na próxima reinicialização. Esses rótulos são criados em /dev/label e são adequados para experimentação. Um rótulo temporário pode ser criado usando `glabel create`.

Para criar um rótulo permanente para um sistema de arquivos UFS2 sem destruir nenhum dado, emita o seguinte comando:

```
# tunefs -L home /dev/da3
```

Um rótulo deve agora existir em /dev/ufs que pode ser adicionado ao arquivo /etc/fstab:

```
/dev/ufs/home       /home           ufs     rw              2       2
```



O sistema de arquivos não deve ser montado durante a tentativa de executar o `tunefs`.

Agora o sistema de arquivos pode ser montado:

```
# mount /home
```

A partir deste ponto, desde que o módulo do kernel geom_label.ko seja carregado na inicialização com o /boot/loader.conf ou com a opção do kernel `GEOM_LABEL` estando presente, o nó do dispositivo pode mudar sem qualquer efeito negativo no sistema.

Os sistemas de arquivos também podem ser criados com um rótulo padrão usando a flag `-L` com o comando `newfs`. Consulte newfs(8) para obter maiores informações.

O seguinte comando pode ser usado para destruir o rótulo:

```
# glabel destroy home
```

O exemplo a seguir mostra como rotular as partições de um disco de inicialização.

Exemplo 42. Rotulando Partições no Disco de Inicialização

Ao marcar permanentemente as partições no disco de inicialização, o sistema deve poder continuar a inicializar normalmente, mesmo se o disco for movido para outro controlador ou transferido para um sistema diferente. Para este exemplo, presume-se que um único disco ATA é usado, que é atualmente reconhecido pelo sistema como ad0. Também é assumido que o esquema de partição padrão do FreeBSD é usado, com /, /var, /usr e /tmp, bem como uma partição de swap.

Reinicialize o sistema e, no prompt do loader(8), pressione `4` para inicializar no modo de usuário único. Em seguida, insira os seguintes comandos:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs
# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var
# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
# glabel label tmp /dev/ad0s1e
GEOM_LABEL: Label for provider /dev/ad0s1e is label/tmp
# glabel label swap /dev/ad0s1b
GEOM_LABEL: Label for provider /dev/ad0s1b is label/swap
# exit
```

O sistema continuará com a inicialização multiusuário. Depois que a inicialização terminar, edite o arquivo /etc/fstab e substitua os nomes de dispositivos convencionais por seus respectivos rótulos. No final o /etc/fstab ficará assim:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/label/swap         none            swap    sw              0       0
/dev/label/rootfs       /               ufs     rw              1       1
/dev/label/tmp          /tmp            ufs     rw              2       2
/dev/label/usr          /usr            ufs     rw              2       2
/dev/label/var          /var            ufs     rw              2       2
```

O sistema agora pode ser reinicializado. Se tudo correr bem, ele aparecerá normalmente e o comando `mount` mostrará:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

A classe glabel(8) suporta um tipo de rótulo para sistemas de arquivos UFS, com base no ID do sistema de arquivos exclusivo `ufsid`. Esses rótulos podem ser encontrados em /dev/ufsid e são criados automaticamente durante a inicialização do sistema. É possível usar rótulos `ufsid` para montar partições usando o /etc/fstab. Use o `glabel status` para receber uma lista de sistemas de arquivos e seus rótulos `ufsid` correspondentes:

```
% glabel status
                  Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

No exemplo acima, ad4s1d representa /var, enquanto ad4s1f representa /usr. Usando os valores `ufsid` mostrados, essas partições podem agora ser montadas com as seguintes entradas em /etc/fstab:

```
/dev/ufsid/486b6fc38d330916        /var        ufs        rw        2      2
/dev/ufsid/486b6fc16926168e        /usr        ufs        rw        2      2
```

Quaisquer partições com rótulos `ufsid` podem ser montadas dessa forma, eliminando a necessidade de criar manualmente rótulos permanentes, enquanto ainda desfruta dos benefícios da montagem independente do nome do dispositivo.

18.8. Journaling UFS através do GEOM

Suporte para journaling em sistemas de arquivos UFS está disponível no FreeBSD. A implementação é fornecida através do subsistema GEOM e é configurada usando o comando `gjournal`. Ao contrário de outras implementações de journaling de sistemas de arquivos, o método `gjournal` é baseado em blocos e não é implementado como parte do sistema de arquivos. É uma extensão do GEOM.

O jornaling armazena um log de transações do sistema de arquivos, como alterações que compõem uma operação de gravação em disco completa, antes que os metadados e as gravações de arquivos sejam confirmados no disco. Esse log de transação pode ser repetido posteriormente para refazer as transações do sistema de arquivos, evitando inconsistências no sistema de arquivos.

Esse método fornece outro mecanismo para proteger contra perda de dados e inconsistências do sistema de arquivos. Ao contrário das Soft Updates, que rastreiam e impõem atualizações de metadados e snapshots, que criam uma imagem do sistema de arquivos, um log é armazenado no espaço em disco especificamente para essa tarefa. Para melhor desempenho, o journal pode ser armazenado em outro disco. Nessa configuração, o provedor do journal ou o dispositivo de armazenamento deve ser listado após o dispositivo para ativar o journaling.

O kernel GENERIC fornece suporte para o `gjournal`. Para carregar automaticamente o módulo do kernel geom_journal.ko no momento da inicialização, adicione a seguinte linha ao arquivo /boot/loader.conf:

```
geom_journal_load="YES"
```

Se um kernel personalizado for usado, certifique-se de que a linha a seguir esteja no arquivo de configuração do kernel:

```
options GEOM_JOURNAL
```

Depois que o módulo é carregado, um journal pode ser criado em um novo sistema de arquivos usando as etapas a seguir. Neste exemplo, da4 é um novo disco SCSI:

```
# gjournal load
# gjournal label /dev/da4
```

Isto irá carregar o módulo e criar um nó de dispositivo /dev/da4.journal em /dev/da4.

Um sistema de arquivos UFS pode agora ser criado no dispositivo journaled e depois montado em um ponto de montagem existente:

```
# newfs -O 2 -J /dev/da4.journal
# mount /dev/da4.journal /mnt
```



No caso de várias slices, será criado um journal para cada slice individual. Por exemplo, se ad4s1 e ad4s2 forem slices, o `gjournal` criará ad4s1.journal e ad4s2.journal.

O journaling também pode ser ativado nos sistemas de arquivos atuais usando o `tunefs`. No entanto, *sempre* faça um backup antes de tentar alterar um sistema de arquivos existente. Na maioria dos casos, o `gjournal` falhará se não for possível criar o registro de log, mas isso não protege contra a perda de dados incorrida como resultado do uso indevido do `tunefs`. Consulte gjournal(8) e tunefs(8) para maiores informações sobre esses comandos.

É possível fazer o journaling do disco de inicialização de um sistema FreeBSD. Consulte o artigo Implementando o journaling do UFS em um PC de mesa para obter instruções detalhadas.

Capítulo 19. O sistema de arquivos Z (ZFS)

O *Sistema de Arquivos Z*, ou ZFS, é um sistema de arquivos avançado projetado para superar muitos dos principais problemas encontrados em projetos anteriores.

Originalmente desenvolvido pela Sun™, o desenvolvimento contínuo do ZFS em código aberto foi movido para o Projeto OpenZFS.

O ZFS tem três metas principais de design:

- Integridade de dados: Todos os dados incluem um checksum dos dados. Quando os dados são gravados, o checksum é calculado e gravado junto com eles. Quando esses dados são lidos posteriormente, o checksum é calculado novamente. Se os checksum's não corresponderem, um erro de dados foi detectado. O ZFS tentará corrigir automaticamente os erros quando houver redundância de dados disponível.
- Armazenamento em pool: os dispositivos de armazenamento físico são adicionados em um pool e o espaço de armazenamento é alocado a partir desse pool compartilhado. O espaço está disponível para todos os sistemas de arquivos e pode ser aumentado pela adição de novos dispositivos de armazenamento ao pool.
- Performance: vários mecanismos de cache fornecem uma maior performance. O ARC é um avançado cache de leitura baseado em memória. Um segundo nível de cache de leitura baseado em disco pode ser adicionado com o L2ARC, e o cache síncrono de escrita baseado em disco está disponível com ZIL.

Uma lista completa de features e terminologias é mostrada em Recursos e terminologia do ZFS.

19.1. O que torna o ZFS diferente

O ZFS é significativamente diferente de qualquer outro sistema de arquivos existente, porque ele é mais do que apenas um simples sistema de arquivos. A combinação das funções tradicionalmente separadas de gerenciamento de volume e de sistema de arquivos, fornece ao ZFS vantagens exclusivas. O sistema de arquivos agora conhece a estrutura abaixo dos discos. Os sistemas de arquivos tradicionais só podem ser criados em um único disco por vez. Se houvesse dois discos, dois sistemas de arquivos separados teriam que ser criados. Em uma configuração de hardware tradicional RAID, esse problema foi contornado apresentando ao sistema operacional um único disco lógico composto pelo espaço fornecido por vários discos físicos, sobre o qual o sistema operacional colocava um sistema de arquivos. Mesmo no caso de soluções de software RAID como as fornecidas pelo GEOM, o sistema de arquivos UFS, que está no topo da transformação RAID, acreditava que estava lidando com um único dispositivo físico. A combinação feita pelo ZFS do gerenciador de volumes e do sistema de arquivos resolve isso e permite a criação de vários sistemas de arquivos, todos compartilhando um pool de armazenamento disponível. Uma das maiores vantagens do reconhecimento do layout físico dos discos pelo ZFS é que os sistemas de arquivos existentes podem ser expandidos automaticamente quando novos discos são adicionados ao pool. Esse novo espaço é disponibilizado para todos os sistemas de arquivos. O ZFS também possui várias propriedades diferentes que podem ser aplicadas a cada sistema de arquivos, oferecendo muitas vantagens para a criação de vários sistemas de arquivos e datasets diferentes, em vez de um único sistema de arquivos monolítico.

19.2. Guia de Início Rápido

Existe um mecanismo de inicialização que permite ao FreeBSD montar pools do ZFS durante a inicialização do sistema. Para habilitá-lo, adicione esta linha ao /etc/rc.conf:

```
zfs_enable="YES"
```

Então inicie o serviço:

```
# service zfs start
```

Os exemplos nesta seção assumem três discos SCSI com os seguintes nomes de dispositivo da0, da1 e da2. Usuários de hardware do tipo SATA devem usar nomes de dispositivo ada.

19.2.1. Pool de Disco Único

Para criar um pool simples e não-redundante usando um único disco:

```
# zpool create example /dev/da0
```

Para visualizar o novo pool, verifique a saída do comando `df`:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

Esta saída mostra que o pool `example` foi criado e montado e agora está acessível como um sistema de arquivos. Arquivos podem ser criados nele e os usuários podem navegar nele:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r--   1 root  wheel    0 Aug 29 23:15 testfile
```

No entanto, esse pool não está aproveitando nenhuma feature do ZFS. Para criar um dataset neste pool com a compressão ativada:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

O dataset `example/compressed` é agora um sistema de arquivos ZFS compactado. Tente copiar alguns arquivos grandes para /example/compressed.

A compactação pode ser desativada com:

```
# zfs set compression=off example/compressed
```

Para desmontar um sistema de arquivos, use `zfs umount` e, em seguida, verifique com `df`:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
example      17547008       0 17547008     0%    /example
```

Para remontar o sistema de arquivos para torná-lo acessível novamente, use `zfs mount` e verifique com o `df`:

```
# zfs mount example/compressed
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
```

O pool e o sistema de arquivos também podem ser observados visualizando a saída do comando `mount`:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

Após a criação, os datasets do ZFS podem ser usados como qualquer sistema de arquivos. No entanto, muitos outros recursos estão disponíveis, e podem ser definidos por conjunto de dados. No exemplo abaixo, um novo sistema de arquivos chamado `data` é criado. Arquivos importantes serão

armazenados nele, portanto, ele é configurado para manter duas cópias de cada bloco de dados:

```
# zfs create example/data
# zfs set copies=2 example/data
```

Agora é possível ver o sistema de arquivos `data` e o espaço utilizado através do comando `df`:

```
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
example/data        17547008       0 17547008     0%    /example/data
```

Observe que cada sistema de arquivos no pool tem a mesma quantidade de espaço disponível. Esta é a razão para usar o `df` nestes exemplos, para mostrar que os sistemas de arquivos usam apenas a quantidade de espaço de que precisam e todos utilizam o mesmo pool. O ZFS elimina conceitos como volumes e partições e permite que vários sistemas de arquivos ocupem o mesmo pool.

Para destruir os sistemas de arquivos e, em seguida, destruir o pool, se ele não for mais necessário:

```
# zfs destroy example/compressed
# zfs destroy example/data
# zpool destroy example
```

19.2.2. RAID-Z

Discos falham. Um método para evitar perda de dados devido a falhas no disco é implementar RAID. O ZFS suporta esse recurso em seu design de pool. Os pools RAID-Z exigem três ou mais discos, mas fornecem mais espaço utilizável do que os pools espelhados.

Este exemplo cria um pool RAID-Z, especificando os discos a serem adicionados ao pool:

```
# zpool create storage raidz da0 da1 da2
```



A Sun™ recomenda que o número de dispositivos usados em uma configuração RAID-Z seja entre três e nove. Para ambientes que exigem um único conjunto de 10 discos ou mais, considere dividi-lo em grupos menores de RAID-Z. Se apenas dois discos estiverem disponíveis e a redundância for um requisito, considere usar o ZFS mirror. Consulte zpool(8) para obter maiores detalhes.

O exemplo anterior criou o zpool `storage`. Este exemplo cria um novo sistema de arquivos chamado `home` neste pool:

```
# zfs create storage/home
```

A compressão e a criação de cópias extras de diretórios e arquivos podem ser ativadas:

```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```

Para tornar este o novo diretório home para usuários, copie os dados de usuários para este diretório e crie os links simbólicos apropriados:

```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```

Os dados dos usuários agora são armazenados no recém-criado diretório /storage/home. Teste adicionando um novo usuário e efetuando login como este usuário.

Tente criar um snapshot do sistema de arquivos que possa ser revertido posteriormente:

```
# zfs snapshot storage/home@08-30-08
```

Os snapshots só podem ser realizados de um sistema de arquivos completo, não de um único diretório ou arquivo.

O caractere `@` é um delimitador entre o nome do sistema de arquivos ou o nome do volume. Se um diretório importante tiver sido excluído acidentalmente, o backup do sistema de arquivos poderá ser feito e, em seguida, revertido para um snapshot anterior, quando o diretório ainda existia:

```
# zfs rollback storage/home@08-30-08
```

Para listar todos os snapshots disponíveis, execute `ls` no diretório .zfs/snapshot no sistema de arquivos. Por exemplo, para ver o snapshot obtido anteriormente:

```
# ls /storage/home/.zfs/snapshot
```

É possível escrever um script para criar snapshots frequentes dos dados do usuário. No entanto, com o tempo, os snapshots podem consumir muito espaço em disco. O snapshot anterior pode ser removido usando o comando:

```
# zfs destroy storage/home@08-30-08
```

Após o teste, /storage/home pode ser o verdadeiro /home usando este comando:

```
# zfs set mountpoint=/home storage/home
```

Execute o `df` e o `mount` para confirmar que o sistema agora trata o sistema de arquivos como o real /home:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

Isso conclui a configuração do RAID-Z. Atualizações de status diárias sobre os sistemas de arquivos criados podem ser geradas como parte das execuções noturnas doperiodic(8). Adicione esta linha ao /etc/periodic.conf:

```
daily_status_zfs_enable="YES"
```

19.2.3. Recuperando o RAID-Z

Todo software RAID tem um método de monitorar seu `status`. O status dos dispositivos RAID-Z pode ser visualizado com este comando:

```
# zpool status -x
```

Se todos os pools estiverem Online e tudo estiver normal, a mensagem mostrará:

```
all pools are healthy
```

Se houver um problema, talvez um disco que esteja no estado Offline, o status do pool será semelhante a:

```
  pool: storage
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
```

```
    Sufficient replicas exist for the pool to continue functioning in a
    degraded state.
action: Online the device using 'zpool online' or replace the device with
    'zpool replace'.
 scrub: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    storage     DEGRADED     0     0     0
      raidz1    DEGRADED     0     0     0
        da0     ONLINE       0     0     0
        da1     OFFLINE      0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Isso indica que o dispositivo foi colocado off-line anteriormente pelo administrador com este comando:

```
# zpool offline storage da1
```

Agora o sistema pode ser desligado para substituir o da1. Quando o sistema estiver novamente online, o disco com falha poderá ser substituído no pool:

```
# zpool replace storage da1
```

Agora, o status pode ser verificado novamente, desta vez sem `-x`, para que todos os pools sejam mostrados:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Neste exemplo, tudo está normal.

19.2.4. Verificação de dados

O ZFS utiliza checksums para verificar a integridade dos dados armazenados. Estes são ativados automaticamente na criação dos sistemas de arquivos.



Os checksums podem ser desabilitados, mas isto *não* é recomendado! Os checksums ocupam muito pouco espaço de armazenamento e fornecem integridade dos dados. Muitos recursos do ZFS não funcionarão adequadamente com os checksums desabilitados. Não há nenhum ganho perceptível de desempenho ao desativar os checksums.

A verificação de checksum é conhecida como *scrubbing*. Verifique a integridade dos dados do pool `storage` com este comando:

```
# zpool scrub storage
```

A duração de um scrub depende da quantidade de dados armazenados. Quantidades maiores de dados levarão proporcionalmente mais tempo para serem verificadas. Scrubs utilizam muito I/O, e apenas um scrub tem permissão para ser executado por vez. Após a conclusão do scrub, o status pode ser visualizado com `status`:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: scrub completed with 0 errors on Sat Jan 26 19:57:37 2013
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

A data de conclusão da última operação de scrub é exibida para ajudar a rastrear quando outro scrub é necessário. Uma rotina recorrente de scrubs ajuda a proteger os dados contra corrupção silenciosa e garante a integridade do pool.

Consulte zfs(8) e zpool(8) para outras opções do ZFS.

19.3. Administração `zpool`

A administração do ZFS é dividida entre dois utilitários principais. O utilitário `zpool` controla a operação do pool e trata da adição, remoção, substituição e gerenciamento de discos. O utilitário `zfs` lida com a criação, destruição e gerenciamento de datasets, tanto para sistemas de arquivos quanto

para volumes.

19.3.1. Criando e destruindo pools de armazenamento

A criação de um pool de armazenamento do ZFS (*zpool*) envolve a tomada de várias decisões que são relativamente permanentes porque a estrutura do pool não pode ser alterada depois que o pool é criado. A decisão mais importante é quais tipos de vdevs usar para agrupar os discos físicos. Consulte a lista de tipos vdev para obter detalhes sobre as opções possíveis. Após o pool ter sido criado, a maioria dos tipos de vdev não permite que discos adicionais sejam adicionados ao vdev. As exceções são os mirrors, que permitem que discos adicionais sejam adicionados ao vdev, e stripes, que podem ser atualizados para mirrors ao anexar um disco adicional ao vdev. Embora vdevs adicionais possam ser adicionados para expandir um pool, o layout do pool não pode ser alterado após a criação do pool. Em vez disso, os dados devem ser salvos em um backup e o pool destruído e recriado.

Crie um pool do tipo mirror simples:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada1    ONLINE       0     0     0
            ada2    ONLINE       0     0     0

errors: No known data errors
```

Vários vdevs podem ser criados de uma só vez. Especifique vários grupos de discos separados pela palavra-chave do tipo vdev, `mirror` neste exemplo:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2 mirror /dev/ada3 /dev/ada4
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada1    ONLINE       0     0     0
            ada2    ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
```

```
            ada3    ONLINE       0     0     0
            ada4    ONLINE       0     0     0

errors: No known data errors
```

Os pools também podem ser construídos usando partições em vez de discos inteiros. Colocar o ZFS em uma partição separada permite que o mesmo disco tenha outras partições para outras finalidades. Em particular, partições com bootcode e sistemas de arquivos necessários para a inicialização podem ser adicionadas. Isso permite inicializar a partir de discos que também são membros de um pool. Não há penalidade de desempenho no FreeBSD ao usar uma partição em vez de um disco inteiro. O uso de partições também permite ao administrador *sub-provisionar* os discos, usando menos que a capacidade total. Se um disco de substituição futuro com o mesmo tamanho nominal do original tiver uma capacidade ligeiramente menor, a partição menor ainda se ajustará e o disco de substituição ainda poderá ser usado.

Crie um pool RAID-Z2 usando partições:

```
# zpool create mypool raidz2 /dev/ada0p3 /dev/ada1p3 /dev/ada2p3 /dev/ada3p3
/dev/ada4p3 /dev/ada5p3
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

Um pool que não é mais necessário pode ser destruído para que os discos possam ser reutilizados. Destruir um pool envolve primeiro desmontar todos os datasets nesse pool. Se os datasets estiverem em uso, a operação de desmontagem falhará e o pool não será destruído. A destruição do pool pode ser forçada com `-f`, mas isso pode causar um comportamento indefinido em aplicações que tiverem arquivos abertos nesses datasets.

19.3.2. Adicionando e Removendo Dispositivos

Existem dois casos para adicionar discos a um zpool: anexar um disco a um vdev existente com `zpool attach` ou incluir vdevs ao pool com `zpool add`. Apenas alguns vdev types permitem que discos sejam adicionados ao vdev após a criação.

Um pool criado com um único disco não tem redundância. Dados corrompidos podem ser detectados, mas não reparados, porque não há outra cópia dos dados. A propriedade copies pode ser capaz de se recuperar de uma pequena falha, como um setor defeituoso, mas não fornece o mesmo nível de proteção que o mirror ou o RAID-Z. Começando com um pool de um único disco vdev, o `zpool attach` pode ser usado para adicionar um disco adicional ao vdev, criando um mirror. O `zpool attach` também pode ser usado para adicionar discos adicionais a um mirror group, aumentando a redundância e o desempenho de leitura. Se os discos usados para o pool forem particionados, replicar o layout do primeiro disco para o segundo, `gpart backup` e `gpart restore` pode ser usado para facilitar esse processo .

Atualize o disco único (stripe) vdev *ada0p3* para um mirror anexando *ada1p3*:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          ada0p3    ONLINE       0     0     0

errors: No known data errors
# zpool attach mypool ada0p3 ada1p3
Make sure to wait until resilver is done before rebooting.

If you boot from pool 'mypool', you may need to update
boot code on newly attached disk 'ada1p3'.

Assuming you use GPT partitioning and 'da0' is your new boot disk
you may use the following command:

        gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
bootcode written to ada1
# zpool status
  pool: mypool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Fri May 30 08:19:19 2014
        527M scanned out of 781M at 47.9M/s, 0h0m to go
        527M resilvered, 67.53% done
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
```

```
            ada1p3  ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Fri May 30 08:15:58 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
```

Quando adicionar discos ao vdev existente não é uma opção, como para RAID-Z, um método alternativo é adicionar outro vdev ao pool. Vdevs adicionais fornecem desempenho mais alto, distribuindo as operações de escrita nos vdevs. Cada vdev é responsável por fornecer a sua própria redundância. É possível, mas desencorajado, misturar tipos de vdev, como `mirror` e `RAID-Z`. Adicionar um vdev não-redundante a um pool que contenha um vdev mirror ou o RAID-Z arrisca os dados em todo o pool. As gravações são distribuídas, portanto, a falha do disco não-redundante resultará na perda de uma fração de cada bloco que foi gravado no pool.

Os dados são distribuídos em cada um dos vdevs. Por exemplo, com dois vdevs mirror, esse é efetivamente um RAID 10 que escreve em dois conjuntos de mirrors. O espaço é alocado de forma que cada vdev chegue a 100% de uso ao mesmo tempo. Há uma penalidade de desempenho se os vdevs tiverem quantidades diferentes de espaço livre, pois uma quantidade desproporcional dos dados é gravada no vdev menos cheio.

Ao anexar dispositivos adicionais a um pool de inicialização, lembre-se de atualizar o bootcode.

Anexe um segundo grupo de mirror's (ada2p3 and ada3p3) ao mirror existente:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Fri May 30 08:19:35 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
# zpool add mypool mirror ada2p3 ada3p3
```

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
bootcode written to ada2
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada3
bootcode written to ada3
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0

errors: No known data errors
```

Atualmente, os vdevs não podem ser removidos de um pool e os discos só podem ser removidos de um mirror se houver redundância restante suficiente. Se apenas um disco em um grupo de mirror's permanecer, ele deixará de ser um mirror e voltará a ser um srtipe, arriscando todo o pool se o disco restante falhar.

Remova um disco de um grupo de mirror's triplo:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
# zpool detach mypool ada2p3
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
```

```
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
```

19.3.3. Verificando o status de um pool

O status do pool é importante. Se uma unidade ficar off-line ou for detectado um erro de leitura, gravação ou de checksum, a contagem de erros correspondente aumentará. A saída `status` mostra a configuração e o status de cada dispositivo no pool e o status de todo o pool. Ações que precisam ser tomadas e detalhes sobre o último `scrub` também são mostrados.

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 2h25m with 0 errors on Sat Sep 14 04:25:50 2013
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

19.3.4. Limpando Erros

Quando um erro é detectado, os contadores de leitura, escrita ou checksum são incrementados. A mensagem de erro pode ser apagada e os contadores resetados com `zpool clear mypool`. Limpar o estado de erro pode ser importante para scripts automatizados que alertam o administrador quando o pool encontra um erro. Erros adicionais podem não ser relatados se os erros antigos não forem apagados.

19.3.5. Substituindo um dispositivo em funcionamento

Há várias situações em que pode ser desejável substituir um disco por um disco diferente. Ao substituir um disco em funcionamento, o processo mantém o disco antigo online durante a substituição. O pool nunca entra no estado degradado , reduzindo o risco de perda de dados. `zpool replace` copia todos os dados do disco antigo para o novo. Após a conclusão da operação, o disco antigo é desconectado do vdev. Se o novo disco for maior que o disco antigo, pode ser possível aumentar o zpool usando o novo espaço. Veja Aumentando um Pool .

Substitua um dispositivo em funcionamento no pool:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
# zpool replace mypool ada1p3 ada2p3
Make sure to wait until resilver is done before rebooting.

If you boot from pool 'zroot', you may need to update
boot code on newly attached disk 'ada2p3'.

Assuming you use GPT partitioning and 'da0' is your new boot disk
you may use the following command:

        gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
# zpool status
  pool: mypool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Mon Jun  2 14:21:35 2014
        604M scanned out of 781M at 46.5M/s, 0h0m to go
        604M resilvered, 77.39% done
config:

        NAME             STATE     READ WRITE CKSUM
        mypool           ONLINE       0     0     0
          mirror-0       ONLINE       0     0     0
            ada0p3       ONLINE       0     0     0
            replacing-1  ONLINE       0     0     0
              ada1p3     ONLINE       0     0     0
              ada2p3     ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Mon Jun  2 14:21:52 2014
```

```
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
```

19.3.6. Lidando com dispositivos com falha

Quando um disco em um pool falha, o vdev ao qual o disco pertence entra no estado degradado. Todos os dados ainda estão disponíveis, mas o desempenho pode ser reduzido porque os dados ausentes devem ser calculados a partir da redundância disponível. Para restaurar o vdev para um estado totalmente funcional, o dispositivo físico com falha deve ser substituído. O ZFS é então instruído a iniciar a operação resilver. Os dados que estavam no dispositivo com falha são recalculados da redundância disponível e gravados no dispositivo de substituição. Após a conclusão, o vdev retorna ao status online.

Se o vdev não tiver redundância, ou se vários dispositivos falharem e não houver redundância suficiente para compensar, o pool entrará no estado failed. Se um número suficiente de dispositivos não puder ser reconectado ao pool, o pool se tornará inoperante e os dados deverão ser restaurados dos backups.

Ao substituir um disco com falha, o nome do disco com falha é substituído pelo GUID do dispositivo. Um novo parâmetro de nome de dispositivo para o `zpool replace` não é necessário se o dispositivo de substituição tiver o mesmo nome de dispositivo.

Substitua um disco com falha usando o `zpool replace`:

```
# zpool status
  pool: mypool
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device and online it using 'zpool online'.
   see: http://illumos.org/msg/ZFS-8000-2Q
  scan: none requested
config:

        NAME                      STATE     READ WRITE CKSUM
        mypool                    DEGRADED     0     0     0
          mirror-0                DEGRADED     0     0     0
            ada0p3                ONLINE       0     0     0
            316502962686821739    UNAVAIL      0     0     0  was /dev/ada1p3

errors: No known data errors
# zpool replace mypool 316502962686821739 ada2p3
```

```
# zpool status
  pool: mypool
 state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Mon Jun  2 14:52:21 2014
        641M scanned out of 781M at 49.3M/s, 0h0m to go
        640M resilvered, 82.04% done
config:

        NAME                        STATE     READ WRITE CKSUM
        mypool                      DEGRADED     0     0     0
          mirror-0                  DEGRADED     0     0     0
            ada0p3                  ONLINE       0     0     0
            replacing-1             UNAVAIL      0     0     0
              15732067398082357289  UNAVAIL      0     0     0  was /dev/ada1p3/old
              ada2p3                ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Mon Jun  2 14:52:38 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
```

19.3.7. Limpeza do Pool

Recomenda-se que os pools sejam regularmente scrubbed, idealmente pelo menos uma vez por mês. A operação `scrub` requer muito disco e reduzirá o desempenho durante a execução. Evite períodos de alta demanda ao agendar o `scrub` ou use `vfs.zfs.scrub_delay` para ajustar a prioridade relativa do `scrub` para evitar que ele interfira com outras cargas de trabalho.

```
# zpool scrub mypool
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub in progress since Wed Feb 19 20:52:54 2014
        116G scanned out of 8.60T at 649M/s, 3h48m to go
        0 repaired, 1.32% done
config:
```

```
        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

No caso de uma operação de limpeza precisar ser cancelada, emita `zpool scrub -s mypool`.

19.3.8. Auto Cura (Self-Healing)

Os checksums armazenados com os blocos de dados habilitam o sistema de arquivos a se *autocorrigirem*. Esse recurso reparará automaticamente os dados cujo checksum não corresponde à registrada em outro dispositivo que faz parte do pool de armazenamento. Por exemplo, um espelho com dois discos em que uma unidade está começando a funcionar incorretamente e não pode armazenar os dados adequadamente. Isso é ainda pior quando os dados não são acessados há muito tempo, como no armazenamento de arquivos de longo prazo. Os sistemas de arquivos tradicionais precisam executar algoritmos que verificam e reparam os dados como o fsck(8). Esses comandos levam tempo e, em casos graves, um administrador precisa decidir manualmente qual operação de reparo deve ser executada. Quando o ZFS detecta um bloco de dados com um checksum que não corresponde, ele tenta ler os dados do disco de espelhamento. Se esse disco puder fornecer os dados corretos, ele não apenas fornecerá esses dados ao aplicativo que os está solicitando, mas também corrigirá os dados errados no disco que continha o checksum incorreto. Isso acontece sem qualquer interação de um administrador do sistema durante a operação normal do pool.

O próximo exemplo demonstra esse comportamento de autocura. Um conjunto espelhado de discos /dev/ada0 e /dev/ada1 é criado.

```
# zpool create healer mirror /dev/ada0 /dev/ada1
# zpool status healer
  pool: healer
 state: ONLINE
  scan: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
```

```
# zpool list
NAME     SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
healer   960M  92.5K   960M         -         -     0%    0%  1.00x  ONLINE  -
```

Alguns dados importantes que devem ser protegidos de erros de dados usando o recurso de correção automática são copiados para o pool. É criado um checksum do pool para comparação posterior.

```
# cp /some/important/data /healer
# zfs list
NAME     SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
healer   960M  67.7M   892M     7%  1.00x  ONLINE  -
# sha1 /healer > checksum.txt
# cat checksum.txt
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
```

A corrupção de dados é simulada escrevendo dados aleatórios no início de um dos discos no espelho. Para evitar que o ZFS cure os dados assim que forem detectados, o pool é exportado antes da corrupção e importado novamente depois.



Esta é uma operação perigosa que pode destruir dados vitais. Ele é mostrado aqui apenas para fins demonstrativos e não deve ser tentado durante a operação normal de um pool de armazenamento. Nem este exemplo de corrupção intencional deve ser executado em qualquer disco com um sistema de arquivos diferente. Não use outros nomes de dispositivos de disco diferentes daqueles que fazem parte do pool. Certifique-se de que os backups apropriados do pool sejam criados antes de executar o comando!

```
# zpool export healer
# dd if=/dev/random of=/dev/ada1 bs=1m count=200
200+0 records in
200+0 records out
209715200 bytes transferred in 62.992162 secs (3329227 bytes/sec)
# zpool import healer
```

O status do pool mostra que um dispositivo teve um erro. Observe que os aplicativos que leem dados do pool não receberam dados incorretos. O ZFS forneceu dados do dispositivo ada0 com os checksums corretos. O dispositivo com o checksum incorreto pode ser encontrado facilmente, pois a coluna `CKSUM` contém um valor diferente de zero.

```
# zpool status healer
    pool: healer
   state: ONLINE
  status: One or more devices has experienced an unrecoverable error.  An
          attempt was made to correct the error.  Applications are unaffected.
  action: Determine if the device needs to be replaced, and clear the errors
```

```
          using 'zpool clear' or replace the device with 'zpool replace'.
     see: http://illumos.org/msg/ZFS-8000-4J
    scan: none requested
  config:

      NAME        STATE     READ WRITE CKSUM
      healer      ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
          ada0    ONLINE       0     0     0
          ada1    ONLINE       0     0     1

errors: No known data errors
```

O erro foi detectado e tratado usando a redundância presente no disco de espelhamento ada0 não afetado. Uma comparação de checksum com o original irá revelar se o pool está consistente novamente.

```
# sha1 /healer >> checksum.txt
# cat checksum.txt
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
```

Os dois checksums que foram gerados antes e depois da adulteração intencional dos dados do conjunto ainda correspondem. Isso mostra como o ZFS é capaz de detectar e corrigir erros automaticamente quando os checksums são diferentes. Observe que isso só é possível quando há redundância suficiente presente no pool. Um pool que consiste em um único dispositivo não possui recursos de autocorreção. Essa também é a razão pela qual os cheksuma são tão importantes no ZFS e não devem ser desabilitados por nenhum motivo. Nenhum fsck(8) ou programa semelhante de verificação de consistência do sistema de arquivos é necessário para detectar e corrigir isso e o pool ainda estava disponível durante o problema. Uma operação de scrub agora é necessária para sobrescrever os dados corrompidos em ada1.

```
# zpool scrub healer
# zpool status healer
  pool: healer
 state: ONLINE
status: One or more devices has experienced an unrecoverable error.  An
            attempt was made to correct the error.  Applications are unaffected.
action: Determine if the device needs to be replaced, and clear the errors
            using 'zpool clear' or replace the device with 'zpool replace'.
   see: http://illumos.org/msg/ZFS-8000-4J
  scan: scrub in progress since Mon Dec 10 12:23:30 2012
        10.4M scanned out of 67.0M at 267K/s, 0h3m to go
        9.63M repaired, 15.56% done
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
```

```
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0   627  (repairing)

errors: No known data errors
```

A operação scrub lê os dados do ada0 e reescreve todos os dados com um checksum incorreto no ada1. Isso é indicado pela saída `(repairing)` do `zpool status`. Após a conclusão da operação, o status do conjunto é alterado para:

```
# zpool status healer
  pool: healer
 state: ONLINE
status: One or more devices has experienced an unrecoverable error.  An
        attempt was made to correct the error.  Applications are unaffected.
action: Determine if the device needs to be replaced, and clear the errors
             using 'zpool clear' or replace the device with 'zpool replace'.
   see: http://illumos.org/msg/ZFS-8000-4J
  scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0 2.72K

errors: No known data errors
```

Após a conclusão da operação scrub e todos os dados terem sido sincronizados de ada0 para ada1, as mensagens de erro podem ser Limpando Erros do status do pool executando `zpool clear`.

```
# zpool clear healer
# zpool status healer
  pool: healer
 state: ONLINE
  scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
```

O pool está agora de volta a um estado totalmente funcional e todos os erros foram apagados.

19.3.9. Crescendo um Pool

O tamanho utilizável de um pool redundante é limitado pela capacidade do menor dispositivo em cada vdev. O menor dispositivo pode ser substituído por um dispositivo maior. Depois de concluir uma operação replace ou resilver, o pool pode crescer para usar a capacidade do Novo dispositivo. Por exemplo, considere um espelho de uma unidade de 1 TB e uma unidade de 2 TB. O espaço utilizável é de 1 TB. Quando a unidade de 1 TB é substituída por outra unidade de 2 TB, o processo de resilverização copia os dados existentes para a nova unidade. Como os dois dispositivos agora têm capacidade para 2 TB, o espaço disponível do espelho pode ser aumentado para 2 TB.

A expansão é acionada usando o `zpool online -e` em cada dispositivo. Após a expansão de todos os dispositivos, o espaço adicional fica disponível para o pool.

19.3.10. Importando e exportando pools

Os pools são *exportados* antes de serem movidos para outro sistema. Todos os conjuntos de dados são desmontados e cada dispositivo é marcado como exportado, mas ainda estarão bloqueados, para que não possam ser usados por outros subsistemas de disco. Isso permite que pools sejam *importados* em outras máquinas, outros sistemas operacionais que suportem ZFS , e até mesmo arquiteturas de hardware diferentes (com algumas advertências, veja zpool(8)). Quando um conjunto de dados tem arquivos abertos, o `zpool export -f` pode ser usado para forçar a exportação de um pool. Use isso com cautela. Os conjuntos de dados são forçosamente desmontados, resultando potencialmente em um comportamento inesperado dos aplicativos que tinham arquivos abertos nesses conjuntos de dados.

Exportar um pool que não está em uso:

```
# zpool export mypool
```

Importar um pool automaticamente monta os conjuntos de dados. Este pode não ser o comportamento desejado e pode ser evitado com `zpool import -N`. O `zpool import -o` define propriedades temporárias apenas para esta importação. O `zpool import altroot=` permite importar um pool com um ponto base de montagem em vez da raiz do sistema de arquivos. Se o pool foi usado pela última vez em um sistema diferente e não foi exportado corretamente, uma importação pode ter que ser forçada com `zpool import -f`. O `zpool import -a` importa todos os pools que não parecem estar em uso por outro sistema.

Listar todos os pools disponíveis para importação:

```
# zpool import
   pool: mypool
     id: 9930174748043525076
  state: ONLINE
 action: The pool can be imported using its name or numeric identifier.
 config:

        mypool      ONLINE
```

```
          ada2p3    ONLINE
```

Importe o pool com um diretório raiz alternativo:

```
# zpool import -o altroot=/mnt mypool
# zfs list
zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
mypool               110K  47.0G    31K  /mnt/mypool
```

19.3.11. Atualizando um pool de armazenamento

Após a atualização do FreeBSD, ou se um pool foi importado de um sistema usando uma versão mais antiga do ZFS, o pool pode ser atualizado manualmente para a versão mais recente do ZFS para suportar as funcionalidades mais recentes. Considere se o pool pode precisar ser importado em um sistema antigo antes de atualizar. A atualização é um processo unidirecional. Os pools mais antigos podem ser atualizados, mas os pools com funcionalidades mais recentes não podem ser desatualizados.

Atualize um pool v28 para suportar `Feature Flags`:

```
# zpool status
  pool: mypool
 state: ONLINE
status: The pool is formatted using a legacy on-disk format.  The pool can
        still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'.  Once this is done, the
        pool will no longer be accessible on software that does not support feat
        flags.
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
# zpool upgrade
This system supports ZFS pool feature flags.

The following pools are formatted with legacy version numbers and can
be upgraded to use feature flags.  After being upgraded, these pools
will no longer be accessible by software that does not support feature
flags.

VER  POOL
```

```
---  ------------
28   mypool

Use 'zpool upgrade -v' for a list of available legacy versions.
Every feature flags pool has all supported features enabled.
# zpool upgrade mypool
This system supports ZFS pool feature flags.

Successfully upgraded 'mypool' from version 28 to feature flags.
Enabled the following features on 'mypool':
  async_destroy
  empty_bpobj
  lz4_compress
  multi_vdev_crash_dump
```

Os recursos mais recentes do ZFS não estarão disponíveis até que o `zpool upgrade` seja concluído. O `zpool upgrade -v` pode ser usado para ver quais os novos recursos que serão fornecidos pela atualização, bem como quais recursos já são suportados.

Atualize um pool para suportar feature flags adicionais:

```
# zpool status
  pool: mypool
 state: ONLINE
status: Some supported features are not enabled on the pool. The pool can
        still be used, but some features are unavailable.
action: Enable all features using 'zpool upgrade'. Once this is done,
        the pool may no longer be accessible by software that does not support
        the features. See zpool-features(7) for details.
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
# zpool upgrade
This system supports ZFS pool feature flags.

All pools are formatted using feature flags.

Some supported features are not enabled on the following pools. Once a
feature is enabled the pool may become incompatible with software
that does not support the feature. See zpool-features(7) for details.

POOL  FEATURE
---------------
```

```
zstore
      multi_vdev_crash_dump
      spacemap_histogram
      enabled_txg
      hole_birth
      extensible_dataset
      bookmarks
      filesystem_limits
# zpool upgrade mypool
This system supports ZFS pool feature flags.

Enabled the following features on 'mypool':
  spacemap_histogram
  enabled_txg
  hole_birth
  extensible_dataset
  bookmarks
  filesystem_limits
```



O boot code em sistemas que inicializam a partir de um pool deve ser atualizado para suportar a nova versão do pool. Use `gpart bootcode` na partição que contém o boot code. Existem dois tipos de bootcode disponíveis, dependendo da forma como o sistema inicializa: GPT (a opção mais comum) e EFI (para sistemas mais modernos).

Para inicialização legada usando o GPT, use o seguinte comando:

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
```

Para sistemas que usam o EFI para inicializar, execute o seguinte comando:

```
# gpart bootcode -p /boot/boot1.efifat -i 1 ada1
```

Aplique o bootcode a todos os discos inicializáveis no pool. Veja gpart(8) para obter maiores informações.

19.3.12. Exibindo o histórico gravado do pool

Comandos que modificam o pool são registrados. As ações registradas incluem a criação de conjuntos de dados, a alteração de propriedades ou a substituição de um disco. Esse histórico é útil para revisar como um pool foi criado e qual usuário executou uma ação específica e quando. O histórico não é mantido em um arquivo de log, mas faz parte do próprio pool. O comando para revisar este histórico é apropriadamente chamado de `zpool history`:

```
# zpool history
History for 'tank':
```

```
2013-02-26.23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1
2013-02-27.18:50:58 zfs set atime=off tank
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank
2013-02-27.18:51:18 zfs create tank/backup
```

A saída mostra os comandos `zpool` e `zfs` que foram executados no pool juntamente com um registro de data e hora. Somente comandos que alteram o pool de alguma forma são registrados. Comandos como `zfs list` não estão incluídos. Quando nenhum nome de pool é especificado, é exibido o histórico de todos os pools.

O `zpool history` pode mostrar ainda mais informações quando as opções `-i` ou `-l` são fornecidas. A opção `-i` exibe eventos iniciados pelo usuário, bem como eventos do ZFS registrados internamente.

```
# zpool history -i
History for 'tank':
2013-02-26.23:02:35 [internal pool create txg:5] pool spa 28; zfs spa 28; zpl 5;uts
9.1-RELEASE 901000 amd64
2013-02-27.18:50:53 [internal property set txg:50] atime=0 dataset = 21
2013-02-27.18:50:58 zfs set atime=off tank
2013-02-27.18:51:04 [internal property set txg:53] checksum=7 dataset = 21
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank
2013-02-27.18:51:13 [internal create txg:55] dataset = 39
2013-02-27.18:51:18 zfs create tank/backup
```

Mais detalhes podem ser mostrados adicionando a opção `-l`. Os registros de histórico são mostrados em um formato longo, incluindo informações como o nome do usuário que emitiu o comando e o nome do host no qual a alteração foi feita.

```
# zpool history -l
History for 'tank':
2013-02-26.23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1 [user 0 (root) on
:global]
2013-02-27.18:50:58 zfs set atime=off tank [user 0 (root) on myzfsbox:global]
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank [user 0 (root) on myzfsbox:global]
2013-02-27.18:51:18 zfs create tank/backup [user 0 (root) on myzfsbox:global]
```

A saída mostra que o usuário `root` criou o pool espelhado com os discos /dev/ada0 e /dev/ada1. O nome do host `myzfsbox` também é mostrado nos comandos após a criação do pool. A exibição do nome do host se torna importante quando o pool é exportado de um sistema e importado para outro. Os comandos que são emitidos no outro sistema podem claramente ser distinguidos pelo nome do host que é registrado para cada comando.

Ambas as opções para o `zpool history` podem ser combinadas para fornecer as informações mais detalhadas possíveis para qualquer pool. O histórico do pool fornece informações valiosas ao rastrear as ações que foram executadas ou quando é necessária uma saída mais detalhada para a depuração.

19.3.13. Monitoramento de Desempenho

Um sistema de monitoramento integrado pode exibir estatísticas de I/O do pool em tempo real. Ele mostra a quantidade de espaço livre e usado no pool, quantas operações de leitura e gravação estão sendo executadas por segundo e quanto de largura de banda de I/O está sendo utilizada no momento. Por padrão, todos os pools no sistema são monitorados e exibidos. Um nome de pool pode ser fornecido para limitar o monitoramento apenas a esse pool. Um exemplo básico:

```
# zpool iostat
               capacity     operations    bandwidth
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
data         288G  1.53T      2     11  11.3K  57.1K
```

Para monitorar continuamente a atividade de I/O, um número pode ser especificado como o último parâmetro, indicando um intervalo em segundos para aguardar entre as atualizações. A próxima linha de estatística é impressa após cada intervalo. Pressione `Ctrl` + `C` para interromper este monitoramento contínuo. Como alternativa, forneça um segundo número na linha de comando após o intervalo para especificar o número total de estatísticas a serem exibidas.

Estatísticas mais detalhadas de I/O podem ser exibidas com a opção `-v`. Cada dispositivo no pool é mostrado com uma linha de estatísticas. Isso é útil para ver quantas operações de leitura e gravação estão sendo executadas em cada dispositivo e pode ajudar a determinar se algum dispositivo individual está reduzindo a velocidade do pool. Este exemplo mostra um pool espelhado com dois dispositivos:

```
# zpool iostat -v
                           capacity     operations    bandwidth
pool                     alloc   free   read  write   read  write
-----------------------  -----  -----  -----  -----  -----  -----
data                      288G  1.53T      2     12  9.23K  61.5K
  mirror                  288G  1.53T      2     12  9.23K  61.5K
    ada1                     -      -      0      4  5.61K  61.7K
    ada2                     -      -      1      4  5.04K  61.7K
-----------------------  -----  -----  -----  -----  -----  -----
```

19.3.14. Dividindo um pool de armazenamento

Um pool que consiste em um ou mais vdevs espelhados pode ser dividido em dois conjuntos. A menos que seja especificado de outra forma, o último membro de cada espelho é desanexado e usado para criar um novo pool contendo os mesmos dados. A operação deve primeiro ser tentada com `-n`. Os detalhes da operação proposta são exibidos sem que sejam realmente executados. Isso ajuda a confirmar que a operação fará o que o usuário pretende.

19.4. Administração do `zfs`

O utilitário `zfs` é responsável por criar, destruir e gerenciar todos os conjuntos de dados ZFS existentes em um pool. O pool é gerenciado usando o `zpool`.

19.4.1. Criando e destruindo conjuntos de dados

Ao contrário dos discos tradicionais e gerenciadores de volume, o espaço no ZFS_não_ é pré-alocado. Nos sistemas de arquivos tradicionais, depois que todo o espaço é particionado e atribuído, não há como adicionar um sistema de arquivos adicional sem adicionar um novo disco. Com o ZFS, novos sistemas de arquivos podem ser criados a qualquer momento. Cada *conjunto de dados* tem propriedades incluindo recursos como compactação, deduplicação, armazenamento em cache e cotas, bem como outras propriedades úteis como somente leitura, diferenciação de maiúsculas e minúsculas , compartilhamento de arquivos de rede e um ponto de montagem. Os conjuntos de dados podem ser aninhados uns dentro dos outros e os conjuntos de dados filhos herdarão propriedades de seus pais. Cada conjunto de dados pode ser administrado, delegado, replicado, preservado por um snapshot, preso, e destruído como uma unidade. Há muitas vantagens em criar um conjunto de dados separado para cada tipo ou conjunto de arquivos diferente. A única desvantagem de ter um número extremamente grande de conjuntos de dados é que alguns comandos como `zfs list` serão mais lentos, e a montagem de centenas ou mesmo milhares de conjuntos de dados pode retardar o processo de inicialização do FreeBSD.

Crie um novo conjunto de dados e ative a compactação LZ4 nele:

```
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                781M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.20M  93.2G   608K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/tmp        152K  93.2G   152K  /var/tmp
# zfs create -o compress=lz4 mypool/usr/mydataset
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
mypool                 781M  93.2G   144K  none
mypool/ROOT            777M  93.2G   144K  none
mypool/ROOT/default    777M  93.2G   777M  /
mypool/tmp             176K  93.2G   176K  /tmp
mypool/usr             704K  93.2G   144K  /usr
mypool/usr/home        184K  93.2G   184K  /usr/home
mypool/usr/mydataset  87.5K  93.2G  87.5K  /usr/mydataset
```

```
mypool/usr/ports       144K  93.2G   144K  /usr/ports
mypool/usr/src         144K  93.2G   144K  /usr/src
mypool/var            1.20M  93.2G   610K  /var
mypool/var/crash       148K  93.2G   148K  /var/crash
mypool/var/log         178K  93.2G   178K  /var/log
mypool/var/mail        144K  93.2G   144K  /var/mail
mypool/var/tmp         152K  93.2G   152K  /var/tmp
```

A destruição de um conjunto de dados é muito mais rápida que a exclusão de todos os arquivos que residem no conjunto de dados, pois não envolve a verificação de todos os arquivos e a atualização de todos os metadados correspondentes.

Destrua o conjunto de dados criado anteriormente:

```
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
mypool                 880M  93.1G   144K  none
mypool/ROOT            777M  93.1G   144K  none
mypool/ROOT/default    777M  93.1G   777M  /
mypool/tmp             176K  93.1G   176K  /tmp
mypool/usr             101M  93.1G   144K  /usr
mypool/usr/home        184K  93.1G   184K  /usr/home
mypool/usr/mydataset   100M  93.1G   100M  /usr/mydataset
mypool/usr/ports       144K  93.1G   144K  /usr/ports
mypool/usr/src         144K  93.1G   144K  /usr/src
mypool/var            1.20M  93.1G   610K  /var
mypool/var/crash       148K  93.1G   148K  /var/crash
mypool/var/log         178K  93.1G   178K  /var/log
mypool/var/mail        144K  93.1G   144K  /var/mail
mypool/var/tmp         152K  93.1G   152K  /var/tmp
# zfs destroy mypool/usr/mydataset
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                781M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.21M  93.2G   612K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/tmp        152K  93.2G   152K  /var/tmp
```

Nas versões modernas do ZFS, o `zfs destroy` é assíncrono, e o espaço livre pode levar vários minutos para aparecer no pool. Use o `zpool get freeing poolname` para ver a propriedade `freeing`,

indicando quantos conjuntos de dados estão tendo seus blocos liberados em segundo plano. Se houver conjuntos de dados filhos, como snapshots ou outros conjuntos de dados, o pai não poderá ser destruído. Para destruir um conjunto de dados e todos os seus filhos, use `-r` para destruir recursivamente o conjunto de dados e todos os seus filhos. Use `-n -v` para listar os conjuntos de dados e snapshots que seriam destruídos por esta operação, mas na verdade não destruirão nada. O espaço que seria recuperado pela destruição dos snapshots também é mostrado.

19.4.2. Criando e Destruindo Volumes

Um volume é um tipo especial de conjunto de dados. Em vez de ser montado como um sistema de arquivos, ele é exposto como um dispositivo de bloco em /dev/zvol/poolname/dataset. Isso permite que o volume seja usado para outros sistemas de arquivos, para fazer backup dos discos de uma máquina virtual ou para ser exportado usando protocolos como iSCSI ou HAST.

Um volume pode ser formatado com qualquer sistema de arquivos ou usado sem um sistema de arquivos para armazenar dados brutos. Para o usuário, um volume parece ser um disco normal. Colocar sistemas de arquivos comuns nesses *zvols* fornece recursos que os discos comuns ou sistemas de arquivos normalmente não possuem. Por exemplo, o uso da propriedade de compactação em um volume de 250 MB permite a criação de um sistema de arquivos FAT compactado.

```
# zfs create -V 250m -o compression=on tank/fat32
# zfs list tank
NAME USED AVAIL REFER MOUNTPOINT
tank 258M  670M   31K /tank
# newfs_msdos -F32 /dev/zvol/tank/fat32
# mount -t msdosfs /dev/zvol/tank/fat32 /mnt
# df -h /mnt | grep fat32
Filesystem           Size Used Avail Capacity Mounted on
/dev/zvol/tank/fat32 249M  24k  249M     0%   /mnt
# mount | grep fat32
/dev/zvol/tank/fat32 on /mnt (msdosfs, local)
```

Destruir um volume é o mesmo que destruir um conjunto de dados regular do sistema de arquivos. A operação é quase instantânea, mas pode levar vários minutos para que o espaço livre seja recuperado em segundo plano.

19.4.3. Renomeando um Conjunto de Dados

O nome de um conjunto de dados pode ser alterado com `zfs rename`. O pai de um conjunto de dados também pode ser alterado com esse comando. A renomeação de um conjunto de dados para um conjunto de dados pai diferente alterará o valor das propriedades herdadas do conjunto de dados pai. Quando um conjunto de dados é renomeado, ele é desmontado e, em seguida, remontado no novo local (que é herdado do novo conjunto de dados pai). Esse comportamento pode ser evitado com `-u`.

Renomeie um conjunto de dados e mova-o para um conjunto de dados pai diferente:

```
# zfs list
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool                  780M  93.2G   144K  none
mypool/ROOT             777M  93.2G   144K  none
mypool/ROOT/default     777M  93.2G   777M  /
mypool/tmp              176K  93.2G   176K  /tmp
mypool/usr              704K  93.2G   144K  /usr
mypool/usr/home         184K  93.2G   184K  /usr/home
mypool/usr/mydataset   87.5K  93.2G  87.5K  /usr/mydataset
mypool/usr/ports        144K  93.2G   144K  /usr/ports
mypool/usr/src          144K  93.2G   144K  /usr/src
mypool/var             1.21M  93.2G   614K  /var
mypool/var/crash        148K  93.2G   148K  /var/crash
mypool/var/log          178K  93.2G   178K  /var/log
mypool/var/mail         144K  93.2G   144K  /var/mail
mypool/var/tmp          152K  93.2G   152K  /var/tmp
# zfs rename mypool/usr/mydataset mypool/var/newname
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                780M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.29M  93.2G   614K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/newname   87.5K  93.2G  87.5K  /var/newname
mypool/var/tmp        152K  93.2G   152K  /var/tmp
```

Os snapshots também podem ser renomeados dessa maneira. Devido à natureza dos snapshots, eles não podem ser renomeados para um conjunto de dados pai diferente. Para renomear um snapshot recursivo, especifique `-r` e todos os snapshots com o mesmo nome nos conjuntos de dados filho também serão renomeados.

```
# zfs list -t snapshot
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@first_snapshot       0      -  87.5K  -
# zfs rename mypool/var/newname@first_snapshot new_snapshot_name
# zfs list -t snapshot
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@new_snapshot_name       0      -  87.5K  -
```

19.4.4. Configurando Propriedades do Conjunto de Dados

Cada conjunto de dados do ZFS possui várias propriedades que controlam seu comportamento. A maioria das propriedades é herdada automaticamente do conjunto de dados pai, mas pode ser substituída localmente. Defina uma propriedade em um conjunto de dados com `zfs set property=value__dataset`. A maioria das propriedades tem um conjunto limitado de valores válidos, o `zfs get` exibirá cada propriedade e valor válido possível. A maioria das propriedades pode ser revertida para seus valores herdados usando `zfs inherit`.

Propriedades definidas pelo usuário também podem ser definidas. Eles se tornam parte da configuração do conjunto de dados e podem ser usados para fornecer informações adicionais sobre o conjunto de dados ou seu conteúdo. Para distinguir essas propriedades personalizadas daquelas fornecidas como parte do ZFS, dois pontos (`:`) são usados para criar um namespace personalizado para a propriedade.

```
# zfs set custom:costcenter=1234 tank
# zfs get custom:costcenter tank
NAME PROPERTY           VALUE SOURCE
tank custom:costcenter  1234  local
```

Para remover uma propriedade customizada, use o `zfs inherit` com `-r`. Se a propriedade personalizada não estiver definida em nenhum dos conjuntos de dados pai, ela será removida completamente (embora as alterações ainda sejam registradas no histórico do pool).

```
 #   zfs inherit -r  customizado :  costcenter   tanque
 #   zfs  customizado :  costcenter   tank
NAME PROPERTY VALUE SOURCE
tanque personalizado: costcenter - -
 #   zfs obtém todos  tank  | grep  personalizado :  costcenter
 #
```

19.4.4.1. Obtendo e definindo propriedades de compartilhamento

Duas propriedades de conjunto de dados comumente usadas e úteis são as opções de compartilhamento NFS e SMB. Configurar estas define se e como os conjuntos de dados do ZFS podem ser compartilhados na rede. Atualmente, apenas o compartilhamento de configurações via NFS é suportado no FreeBSD. Para obter o status atual de um compartilhamento, insira:

```
# zfs get sharenfs mypool/usr/home
NAME             PROPERTY  VALUE    SOURCE
mypool/usr/home  sharenfs  on       local
# zfs get sharesmb mypool/usr/home
NAME             PROPERTY  VALUE    SOURCE
mypool/usr/home  sharesmb  off      local
```

Para ativar o compartilhamento de um conjunto de dados, insira:

```
# zfs set sharenfs=on mypool/usr/home
```

Também é possível definir opções adicionais para compartilhar conjuntos de dados por meio do NFS, como `-alldirs`, `-maproot` e `-network`. Para definir opções adicionais para um conjunto de dados compartilhado por meio do NFS, insira:

```
# zfs set sharenfs="-alldirs,-maproot=root,-network=192.168.1.0/24" mypool/usr/home
```

19.4.5. Gerenciando Snapshots

Os snapshots são um dos recursos mais poderosos do ZFS. Um snapshot fornece uma cópia point-in-time somente leitura do conjunto de dados. Com Copy-On-Write (COW), os snapshots podem ser criados rapidamente, preservando a versão mais antiga dos dados no disco. Se não houver snapshots, o espaço será recuperado para uso futuro quando os dados forem reconfigurados ou excluídos. Os snapshots preservam o espaço em disco gravando apenas as diferenças entre o conjunto de dados atual e uma versão anterior. Os snapshots são permitidos apenas em conjuntos de dados completos, não em arquivos ou diretórios individuais. Quando um snapshot é criado a partir de um conjunto de dados, tudo contido nele é duplicado. Isso inclui as propriedades do sistema de arquivos, arquivos, diretórios, permissões e assim por diante. Os snapshots não usam espaço adicional quando são criados pela primeira vez, consumindo espaço apenas quando os blocos de referência são alterados. Snapshots recursivos obtidos com `-r` criam um instantâneo com o mesmo nome no conjunto de dados e em todos os seus filhos, fornecendo um snapshot moment-in-time de todos os sistemas de arquivos no momento. Isso pode ser importante quando um aplicativo possui arquivos em vários conjuntos de dados relacionados ou dependentes um do outro. Sem snapshots, um backup teria cópias dos arquivos de diferentes pontos no tempo.

Os snapshots no ZFS fornecem uma variedade de recursos que até mesmo outros sistemas de arquivos com a funcionalidade de snapshots não têm. Um exemplo típico de uso de snapshots é ter uma maneira rápida de fazer backup do estado atual do sistema de arquivos quando uma ação arriscada, como uma instalação de software ou uma atualização do sistema, é executada. Se a ação falhar, o snapshot poderá ser revertido e o sistema terá o mesmo estado de quando o snapshot foi criado. Se a atualização foi bem sucedida, o instantâneo pode ser excluído para liberar espaço. Sem snapshots, uma atualização com falha geralmente requer uma restauração de backup, o que é tedioso, consome tempo e pode exigir tempo de inatividade durante o qual o sistema não pode ser usado. Os snapshots podem ser revertidos rapidamente, mesmo enquanto o sistema está sendo executado em operação normal, com pouco ou nenhum tempo de inatividade. A economia de tempo é enorme com sistemas de armazenamento de vários terabytes e o tempo necessário para copiar os dados a partir do backup. Os snapshots não substituem um backup completo de um pool, mas podem ser usados de maneira rápida e fácil para armazenar uma cópia do conjunto de dados em um momento específico.

19.4.5.1. Criando Snapshots

Os snapshots são criados com `zfs snapshot dataset@snapshotname`. Adicionar a opção `-r` cria um snapshot recursivamente, com o mesmo nome em todos os conjuntos de dados filho.

Crie um Snapshot recursivo de todo o pool:

```
# zfs list -t all
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool                                780M  93.2G   144K  none
mypool/ROOT                           777M  93.2G   144K  none
mypool/ROOT/default                   777M  93.2G   777M  /
mypool/tmp                            176K  93.2G   176K  /tmp
mypool/usr                            616K  93.2G   144K  /usr
mypool/usr/home                       184K  93.2G   184K  /usr/home
mypool/usr/ports                      144K  93.2G   144K  /usr/ports
mypool/usr/src                        144K  93.2G   144K  /usr/src
mypool/var                           1.29M  93.2G   616K  /var
mypool/var/crash                      148K  93.2G   148K  /var/crash
mypool/var/log                        178K  93.2G   178K  /var/log
mypool/var/mail                       144K  93.2G   144K  /var/mail
mypool/var/newname                   87.5K  93.2G  87.5K  /var/newname
mypool/var/newname@new_snapshot_name     0      -  87.5K  -
mypool/var/tmp                        152K  93.2G   152K  /var/tmp
# zfs snapshot -r mypool@my_recursive_snapshot
# zfs list -t snapshot
NAME                                      USED  AVAIL  REFER  MOUNTPOINT
mypool@my_recursive_snapshot                 0      -   144K  -
mypool/ROOT@my_recursive_snapshot            0      -   144K  -
mypool/ROOT/default@my_recursive_snapshot     0      -   777M  -
mypool/tmp@my_recursive_snapshot             0      -   176K  -
mypool/usr@my_recursive_snapshot             0      -   144K  -
mypool/usr/home@my_recursive_snapshot        0      -   184K  -
mypool/usr/ports@my_recursive_snapshot       0      -   144K  -
mypool/usr/src@my_recursive_snapshot         0      -   144K  -
mypool/var@my_recursive_snapshot             0      -   616K  -
mypool/var/crash@my_recursive_snapshot       0      -   148K  -
mypool/var/log@my_recursive_snapshot         0      -   178K  -
mypool/var/mail@my_recursive_snapshot        0      -   144K  -
mypool/var/newname@new_snapshot_name         0      -  87.5K  -
mypool/var/newname@my_recursive_snapshot     0      -  87.5K  -
mypool/var/tmp@my_recursive_snapshot         0      -   152K  -
```

Os snapshots não são mostrados por uma operação normal do `zfs list`. Para listar snapshots , a opção `-t snapshot` é anexado ao `zfs list`. A opção `-t all` exibe os sistemas de arquivos e snapshots.

Os snapshots não são montados diretamente, portanto, nenhum caminho é mostrado na coluna `MOUNTPOINT`. Não há menção ao espaço disponível em disco na coluna `AVAIL`, já que os snapshots não podem ser gravados após serem criados. Compare o snapshot com o conjunto de dados original a partir do qual foi criado:

```
# zfs list -rt all mypool/usr/home
NAME                                USED  AVAIL  REFER  MOUNTPOINT
mypool/usr/home                     184K  93.2G   184K  /usr/home
```

```
mypool/usr/home@my_recursive_snapshot      0      -   184K  -
```

A exibição do conjunto de dados e dos snapshots juntos revela como os snapshots funcionam no modo COW. Eles salvam apenas as alterações (*deltas*) que foram feitas e não o conteúdo completo do sistema de arquivos novamente. Isso significa que os snapshots ocupam pouco espaço quando poucas alterações são feitas. O uso do espaço pode se tornar ainda mais aparente copiando um arquivo para o conjunto de dados e fazendo um segundo snapshots:

```
# cp /etc/passwd /var/tmp
# zfs snapshot mypool/var/tmp@after_cp
# zfs list -rt all mypool/var/tmp
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                        206K  93.2G   118K  /var/tmp
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp                 0      -   118K  -
```

O segundo snapshot contém apenas as alterações feitas no conjunto de dados após a operação de cópia. Isso resulta numa enorme economia de espaço. Observe que o tamanho do snapshot *mypool/var/tmp@my_recursive_snapshot* também foi alterado na coluna `USED` para indicar as alterações entre ela mesma e o snapshot obtido posteriormente.

19.4.5.2. Comparando Snapshots

O ZFS fornece um comando interno para comparar as diferenças de conteúdo entre dois snapshots. Isso é útil quando muitos snapshots foram gerados com o passar do tempo e o usuário deseja ver como o sistema de arquivos mudou ao longo do tempo. Por exemplo, o `zfs diff` permite que um usuário localize o ultimo snapshot que ainda contém um arquivo que foi acidentalmente excluído. Fazer isso para os dois snapshots criados na seção anterior produz essa saída:

```
# zfs list -rt all mypool/var/tmp
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                        206K  93.2G   118K  /var/tmp
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp                 0      -   118K  -
# zfs diff mypool/var/tmp@my_recursive_snapshot
M       /var/tmp/
+       /var/tmp/passwd
```

O comando lista as alterações entre o snapshot especificado (neste caso `mypool/var/tmp@my_recursive_snapshot`) e o sistema de arquivos ativo. A primeira coluna mostra o tipo de mudança:

+	O caminho ou arquivo foi adicionado.
-	O caminho ou arquivo foi excluído.
M	O caminho ou arquivo foi modificado.

R	O caminho ou arquivo foi renomeado.

Comparando a saída com a tabela, fica claro que o passwd foi adicionado após o snapshot `mypool/var/tmp@my_recursive_snapshot` ter sido criado. Isso também resultou em uma modificação no diretório pai montado em `/var/tmp`.

A comparação de dois snapshots é útil ao usar o recurso de replicação do ZFS para transferir um conjunto de dados para um host diferente para fins de backup.

Compare dois snapshots fornecendo o nome completo do conjunto de dados e o nome do snapshot de ambos os conjuntos de dados:

```
# cp /var/tmp/passwd /var/tmp/passwd.copy
# zfs snapshot mypool/var/tmp@diff_snapshot
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@diff_snapshot
M       /var/tmp/
+       /var/tmp/passwd
+       /var/tmp/passwd.copy
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@after_cp
M       /var/tmp/
+       /var/tmp/passwd
```

Um administrador de backup pode comparar dois snapshots recebidos do host de envio e determinar as alterações reais no conjunto de dados. Consulte a seção Replicação para obter maiores informações.

19.4.5.3. Reversão de um Snapshot

Quando pelo menos um snapshot estiver disponível, ele poderá ser revertido a qualquer momento. Na maioria das vezes, esse é o caso quando o estado atual do conjunto de dados não é mais necessário e uma versão mais antiga é preferida. Cenários em que testes de desenvolvimento local deram errado, atualizações de sistemas com falhas que dificultam o funcionamento geral do sistema ou a necessidade de restaurar arquivos ou diretórios excluídos acidentalmente são ocorrências muito comuns. Felizmente, reverter um snapshot é tão fácil quanto digitar `zfs rollback snapshotname`. Dependendo de quantas alterações estão envolvidas, a operação será concluída em um determinado período de tempo. Durante esse período, o conjunto de dados permanece sempre em um estado consistente, da mesma forma que um banco de dados em conformidade com os princípios do ACID ao realizar uma reversão. Isso está acontecendo enquanto o conjunto de dados está ativo e acessível, sem exigir um tempo de inatividade. Depois que o snapshot for revertido, o conjunto de dados terá o mesmo estado de quando o snapshot foi originalmente criado. Todos os outros dados nesse conjunto de dados que não faziam parte do snapshot são descartados. Criar um snapshot do estado atual do conjunto de dados antes de reverter para um anterior é uma boa ideia quando alguns dos dados são necessários mais tarde. Desta forma, o usuário pode alternar entre os snapshots sem perder dados que ainda são valiosos.

No primeiro exemplo, um snapshot é revertido por causa de uma operação descuidada com o comando `rm` que removeu muito mais dados do que o pretendido.

```
# zfs list -rt all mypool/var/tmp
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                        262K  93.2G   120K  /var/tmp
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp              53.5K      -   118K  -
mypool/var/tmp@diff_snapshot             0      -   120K  -
# ls /var/tmp
passwd          passwd.copy     vi.recover
# rm /var/tmp/passwd*
# ls /var/tmp
vi.recover
```

Neste ponto, o usuário percebeu que muitos arquivos foram excluídos e os quer de volta. O ZFS fornece uma maneira fácil de recuperá-los usando reversões, mas somente quando os snapshots de dados importantes são executados regularmente. Para recuperar os arquivos e recomeçar a partir do último snapshot, emita o comando:

```
# zfs rollback mypool/var/tmp@diff_snapshot
# ls /var/tmp
passwd          passwd.copy     vi.recover
```

A operação de reversão restaurou o conjunto de dados para o estado do último snapshot. Também é possível reverter para um snapshot que foi gerado muito antes e que possui outros snapshots criados após ele. Ao tentar fazer isso, o ZFS irá emitir este aviso:

```
# zfs list -rt snapshot mypool/var/tmp
AME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp              53.5K      -   118K  -
mypool/var/tmp@diff_snapshot             0      -   120K  -
# zfs rollback mypool/var/tmp@my_recursive_snapshot
cannot rollback to 'mypool/var/tmp@my_recursive_snapshot': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
mypool/var/tmp@after_cp
mypool/var/tmp@diff_snapshot
```

Esse aviso significa que existem snapshots entre o estado atual do conjunto de dados e o snapshot para o qual o usuário deseja retroceder. Para concluir a reversão, esses snapshots devem ser excluídos. O ZFS não pode rastrear todas as alterações entre estados diferentes do conjunto de dados, porque os snapshots são somente de leitura. O ZFS não excluirá os snapshots afetados, a menos que o usuário especifique a opção `-r` para indicar que essa é a ação desejada. Se essa for a intenção e as consequências da perda de todos os snapshots intermediários forem compreendidas, o comando poderá ser emitido:

```
# zfs rollback -r mypool/var/tmp@my_recursive_snapshot
# zfs list -rt snapshot mypool/var/tmp
```

```
NAME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot     8K      -   152K  -
# ls /var/tmp
vi.recover
```

A saída de `zfs list -t snapshot` confirma que os snapshots intermediários foram removidos como resultado do `zfs rollback -r`.

19.4.5.4. Restaurando arquivos individuais a partir de Snapshots

Os snapshots são montados em um diretório oculto no conjunto de dados pai: .zfs/snapshots/snapshotname. Por padrão, esses diretórios não serão exibidos mesmo quando um `ls -a` padrão for executado. Embora o diretório não seja exibido, ele está lá e pode ser acessado como qualquer diretório normal. A propriedade denominada `snapdir` controla se esses diretórios ocultos aparecem em uma listagem de diretórios. Definir a propriedade como `visible` permite que eles apareçam na saída do `ls` e de outros comandos que lidam com o conteúdo do diretório.

```
# zfs get snapdir mypool/var/tmp
NAME            PROPERTY  VALUE    SOURCE
mypool/var/tmp  snapdir   hidden   default
# ls -a /var/tmp
.               ..              passwd          vi.recover
# zfs set snapdir=visible mypool/var/tmp
# ls -a /var/tmp
.               ..              .zfs            passwd          vi.recover
```

Arquivos individuais podem ser facilmente restaurados para um estado anterior, copiando-os do snapshot de volta para o conjunto de dados pai. A estrutura de diretórios abaixo de .zfs/snapshot tem um diretório nomeado exatamente como os instantâneos criados anteriormente para facilitar sua identificação. No próximo exemplo, presume-se que um arquivo deve ser restaurado a partir do diretório .zfs oculto, copiando-o do snapshot que continha a versão mais recente do arquivo:

```
# rm /var/tmp/passwd
# ls -a /var/tmp
.               ..              .zfs            vi.recover
# ls /var/tmp/.zfs/snapshot
after_cp                my_recursive_snapshot
# ls /var/tmp/.zfs/snapshot/after_cp
passwd          vi.recover
# cp /var/tmp/.zfs/snapshot/after_cp/passwd /var/tmp
```

Quando o comando `ls .zfs/snapshot` foi emitido, a propriedade `snapdir` pode ter sido definida como oculta, mas ainda seria possível listar o conteúdo desse diretório. Cabe ao administrador decidir se esses diretórios serão exibidos. É possível exibi-los para determinados conjuntos de dados e impedi-los para outros. Copiar arquivos ou diretórios deste diretório .zfs/snapshot oculto é bastante simples. Tentar o contrário, resulta neste erro:

```
# cp /etc/rc.conf /var/tmp/.zfs/snapshot/after_cp/
cp: /var/tmp/.zfs/snapshot/after_cp/rc.conf: Read-only file system
```

O erro lembra ao usuário que os snapshots são somente de leitura e não podem ser alterados após a criação. Os arquivos não podem ser copiados para ou removidos dos diretórios de snapshot porque isso alteraria o estado do conjunto de dados que eles representam.

Os snapshots consomem espaço com base em quanto o sistema de arquivos pai foi alterado desde o momento da criação do snapshot. A propriedade `written` de um snapshot rastreia quanto espaço está sendo usado pelo snapshot.

Snapshots são destruídos e o espaço recuperado com o `zfs destroy dataset@snapshot`. Adicionar `-r` remove recursivamente todos os snapshots com o mesmo nome sob o conjunto de dados pai. Adicionar `-n -v` ao comando exibe uma lista dos snapshots que seriam excluídos e uma estimativa de quanto espaço seria recuperado sem executar a operação de destruição real.

19.4.6. Gerenciando Clones

Um clone é uma cópia de um snapshot que é tratado mais como um conjunto de dados regular. Ao contrário de um snapshot, um clone não é somente de leitura, ele pode ser montado e pode ter suas próprias propriedades. Uma vez que um clone tenha sido criado usando `zfs clone`, o snapshot do qual ele foi criado não pode ser destruído. O relacionamento filho/pai entre o clone e o snapshot pode ser revertido usando `zfs promote`. Depois que um clone é promovido, o snapshot se torna um filho do clone, em vez de filho do conjunto de dados pai original. Isso mudará a maneira como o espaço é contabilizado, mas não mudará a quantidade de espaço consumida. O clone pode ser montado em qualquer ponto dentro da hierarquia do sistema de arquivos ZFS, não apenas abaixo do local original do snapshot.

Para demonstrar o recurso de clonagem, este conjunto de dados de exemplo é usado:

```
# zfs list -rt all camino/home/joe
NAME                    USED  AVAIL  REFER  MOUNTPOINT
camino/home/joe         108K   1.3G    87K  /usr/home/joe
camino/home/joe@plans    21K      -  85.5K  -
camino/home/joe@backup    0K      -    87K  -
```

Um uso típico de clones é experimentar um conjunto de dados específico, mantendo o snapshot em volta, para o caso de algo dar errado. Como os snapshots não podem ser alterados, um clone de leitura/gravação de um snapshot é criado. Depois que o resultado desejado é alcançado no clone, o clone pode ser promovido para se tornar um conjunto de dados e o sistema de arquivos antigo é removido. Isso não é estritamente necessário, pois o clone e o conjunto de dados podem coexistir sem problemas.

```
# zfs clone camino/home/joe@backup camino/home/joenew
# ls /usr/home/joe*
/usr/home/joe:
backup.txz     plans.txt
```

```
/usr/home/joenew:
backup.txz     plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
usr/home/joe        1.3G     31k    1.3G     0%    /usr/home/joe
usr/home/joenew     1.3G     31k    1.3G     0%    /usr/home/joenew
```

Depois que um clone é criado, ele é uma cópia exata do estado em que o conjunto de dados estava quando o snapshot foi criado. O clone agora pode ser alterado independentemente de seu conjunto de dados de origem. A única conexão entre os dois é o snapshot. O ZFS registra essa conexão na propriedade `origin`. Uma vez que a dependência entre o snapshot e o clone foi removida promovendo-se o clone usando `zfs promote`, a `origem` do clone é removida, pois agora ele é um conjunto de dados independente. Este exemplo demonstra isso:

```
# zfs get origin camino/home/joenew
NAME                 PROPERTY  VALUE                    SOURCE
camino/home/joenew   origin    camino/home/joe@backup   -
# zfs promote camino/home/joenew
# zfs get origin camino/home/joenew
NAME                 PROPERTY  VALUE   SOURCE
camino/home/joenew   origin    -       -
```

Depois de fazer algumas alterações, como copiar o loader.conf para o clone promovido, por exemplo, o diretório antigo torna-se obsoleto nesse caso. Em vez disso, o clone promovido pode substituí-lo. Isso pode ser conseguido por dois comandos consecutivos: `zfs destroy` no dataset antigo e `zfs rename` no clone para nomeá-lo como o conjunto de dados antigo (ele também poderia ter um nome totalmente diferente).

```
# cp /boot/defaults/loader.conf /usr/home/joenew
# zfs destroy -f camino/home/joe
# zfs rename camino/home/joenew camino/home/joe
# ls /usr/home/joe
backup.txz     loader.conf     plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
usr/home/joe        1.3G    128k    1.3G     0%    /usr/home/joe
```

O snapshot clonado agora é tratado como um conjunto de dados comum. Ele contém todos os dados do snapshot original mais os arquivos que foram adicionados a ele como o loader.conf. Os clones podem ser usados em diferentes cenários para fornecer recursos úteis aos usuários do ZFS. Por exemplo, os jails podem ser disponibilizados como snapshots contendo diferentes conjuntos de aplicativos instalados. Os usuários podem clonar esses snapshots e adicionar seus próprios aplicativos como acharem melhor. Uma vez satisfeitos com as alterações, os clones podem ser promovidos a conjuntos de dados completos e fornecidos aos usuários finais para que trabalhem como se estivessem com um conjunto de dados real. Fornecer estes jails economiza tempo e sobrecarga administrativa.

19.4.7. Replicação

Manter os dados em um único pool e em um único local o expõe a riscos como roubo e desastres naturais ou humanos. Fazer backups regulares de todo o pool é vital. O ZFS fornece um recurso de serialização integrado que pode enviar uma representação de fluxo dos dados para a saída padrão. Usando essa técnica, é possível não apenas armazenar os dados em outro pool conectado ao sistema local, mas também enviá-los por uma rede para outro sistema. Os snapshots são a base para essa replicação (consulte a seção sobre snapshots ZFS). Os comandos usados para replicar dados são `zfs send` e `zfs receive`.

Estes exemplos demonstram a replicação do ZFS com estes dois pools:

```
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
backup  960M    77K   896M         -         -     0%    0%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%    4%  1.00x  ONLINE  -
```

O pool chamado *mypool* é o pool principal no qual os dados são gravados e lidos regularmente. Um segundo pool, *backup* é usado como standby, caso o pool principal fique indisponível. Observe que esse failover não é feito automaticamente pelo ZFS, mas deve ser feito manualmente por um administrador do sistema, quando necessário. Um snapshot é usado para fornecer uma versão consistente do sistema de arquivos a ser replicado. Depois que um snapshot de *mypool* tiver sido criado, ele poderá ser copiado para o pool *backup*. Apenas snapshots podem ser replicados. As alterações feitas desde o snapshot mais recente não serão incluídas.

```
# zfs snapshot mypool@backup1
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@backup1             0      -  43.6M  -
```

Agora que existe um snapshot, o `zfs send` pode ser usado para criar um fluxo representando o conteúdo do snapshot. Esse fluxo pode ser armazenado como um arquivo ou recebido por outro pool. O fluxo é gravado na saída padrão, mas deve ser redirecionado para um arquivo ou canal ou um erro será produzido:

```
# zfs send mypool@backup1
Error: Stream can not be written to a terminal.
You must redirect standard output.
```

Para fazer backup de um conjunto de dados com o `zfs send`, redirecione para um arquivo localizado no pool de backup montado. Assegure-se de que o pool tenha espaço livre suficiente para acomodar o tamanho do snapshot que está sendo enviado, o que significa todos os dados contidos no snapshot, não apenas as mudanças do snapshot anterior.

```
# zfs send mypool@backup1 > /backup/backup1
# zpool list
```

```
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M        -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M        -         -     0%     4%  1.00x  ONLINE  -
```

O `zfs send` transferiu todos os dados do snapshot chamado *backup1* para o pool chamado *backup*. Criar e enviar esses snapshots pode ser feito automaticamente com uma tarefa agendada do cron(8).

Em vez de armazenar os backups como arquivos compactados, o ZFS pode recebê-los como um sistema de arquivos ativo, permitindo que os dados de backup sejam acessados diretamente. Para obter os dados reais contidos nesses fluxos, o `zfs receive` é usado para transformar os fluxos novamente em arquivos e diretórios. O exemplo a seguir combina o `zfs send` e o `zfs receive` usando um canal para copiar os dados de um pool para outro. Os dados podem ser usados diretamente no pool de recebimento após a conclusão da transferência. Um conjunto de dados só pode ser replicado para um conjunto de dados vazio.

```
# zfs snapshot mypool@replica1
# zfs send -v mypool@replica1 | zfs receive backup/mypool
send from @ to mypool@replica1 estimated size is 50.1M
total estimated size is 50.1M
TIME        SENT   SNAPSHOT

# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M        -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M        -         -     0%     4%  1.00x  ONLINE  -
```

19.4.7.1. Backups Incrementais

O `zfs send` também pode determinar a diferença entre dois snapshots e enviar apenas as diferenças entre os dois. Isso economiza espaço em disco e tempo de transferência. Por exemplo:

```
# zfs snapshot mypool@replica2
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@replica1         5.72M      -  43.6M  -
mypool@replica2             0      -  44.1M  -
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
backup  960M  61.7M   898M        -         -     0%    6%  1.00x  ONLINE  -
mypool  960M  50.2M   910M        -         -     0%    5%  1.00x  ONLINE  -
```

Um segundo snapshot chamado *replica2* foi criado. Este segundo snapshot contém apenas as alterações feitas no sistema de arquivos entre o snapshot atual e o anterior, *replica1*. O uso do `zfs send -i` e a indicação do par de snapshots gera um fluxo de réplica incremental contendo apenas os dados que foram alterados. Isso só será bem-sucedido se o snapshot inicial já existir no lado do recebimento.

```
# zfs send -v -i mypool@replica1 mypool@replica2 | zfs receive /backup/mypool
send from @replica1 to mypool@replica2 estimated size is 5.02M
total estimated size is 5.02M
TIME        SENT   SNAPSHOT

# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG  CAP  DEDUP  HEALTH  ALTROOT
backup  960M  80.8M   879M         -         -     0%   8%  1.00x  ONLINE  -
mypool  960M  50.2M   910M         -         -     0%   5%  1.00x  ONLINE  -

# zfs list
NAME                         USED  AVAIL  REFER  MOUNTPOINT
backup                      55.4M   240G   152K  /backup
backup/mypool               55.3M   240G  55.2M  /backup/mypool
mypool                      55.6M  11.6G  55.0M  /mypool

# zfs list -t snapshot
NAME                                       USED  AVAIL  REFER  MOUNTPOINT
backup/mypool@replica1                     104K      -  50.2M  -
backup/mypool@replica2                        0      -  55.2M  -
mypool@replica1                            29.9K      -  50.0M  -
mypool@replica2                               0      -  55.0M  -
```

O fluxo incremental foi transferido com sucesso. Apenas os dados que foram alterados foram replicados, em vez da totalidade da *replica1*. Somente as diferenças foram enviadas, o que levou muito menos tempo para transferir e economizou espaço em disco por não copiar o pool completo novamente. Isso é útil quando se precisa confiar em redes lentas ou quando os custos por byte transferido devem ser considerados.

Um novo sistema de arquivos, *backup/mypool*, está disponível com todos os arquivos e dados do pool *mypool*. Se `-P` for especificado, as propriedades do dataset serão copiadas, incluindo configurações de compactação, cotas e pontos de montagem. Quando `-R` é especificado, todos os conjuntos de dados filho do dataset indicado serão copiados, juntamente com todas as suas propriedades. O envio e o recebimento podem ser automatizados para que backups regulares sejam criados no segundo pool.

19.4.7.2. Envio de backups criptografados pelo SSH

O envio de fluxos pela rede é uma boa maneira de manter um backup remoto, mas apresenta uma desvantagem. Os dados enviados pelo link de rede não são criptografados, permitindo que qualquer pessoa intercepte e transforme os fluxos de volta em dados sem o conhecimento do usuário remetente. Isso é indesejável, especialmente ao enviar os fluxos pela Internet para um host remoto. O SSH pode ser usado para criptografar com segurança os dados enviados por uma conexão de rede. Como o ZFS requer apenas que o fluxo seja redirecionado da saída padrão, é relativamente fácil transmiti-lo através do SSH. Para manter o conteúdo do sistema de arquivos criptografado em trânsito e no sistema remoto, considere o uso do PEFS.

Algumas configurações e precauções de segurança devem ser concluídas primeiro. Apenas as etapas necessárias para a operação do `zfs send` são mostradas aqui. Para mais informações sobre o

SSH, consulte OpenSSH.

Essa configuração é necessária:

- Acesso SSH sem senha entre o host de envio e recebimento usando chaves SSH
- Normalmente, os privilégios do usuário `root` são necessários para enviar e receber fluxos. Isso requer o login no sistema de recebimento como `root`. No entanto, o login como `root` vem desabilitado por padrão por motivos de segurança. O sistema ZFS Delegation pode ser usado para permitir que um usuário não `root` em cada sistema execute as respectivas operações de envio e recebimento.
- No sistema de envio:

```
# zfs allow -u someuser send,snapshot mypool
```

- Para montar o pool, o usuário não privilegiado deve ser o dono do diretório e os usuários regulares devem poder montar sistemas de arquivos. No sistema de recebimento:

```
# sysctl vfs.usermount=1
vfs.usermount: 0 -> 1
# echo vfs.usermount=1 >> /etc/sysctl.conf
# zfs create recvpool/backup
# zfs allow -u someuser create,mount,receive recvpool/backup
# chown someuser /recvpool/backup
```

O usuário sem privilégios agora tem a capacidade de receber e montar conjuntos de dados, e o conjunto de dados *home* pode ser replicado para o sistema remoto:

```
% zfs snapshot -r mypool/home@monday
% zfs send -R mypool/home@monday | ssh someuser@backuphost zfs recv -dvu
recvpool/backup
```

Um snapshot recursivo chamado *monday* é composto do conjunto de dados do sistema de arquivos *home* que reside no pool *mypool*. Em seguida, ele é enviado com o `zfs send -R` para incluir o conjunto de dados, todos os conjuntos de dados filho, snapshots, clones e configurações no fluxo. A saída é canalizada para o `zfs receive` em espera no host remoto *backuphost* através do SSH. Recomenda-se a utilização de um nome de domínio totalmente qualificado ou do endereço IP. A máquina receptora grava os dados no conjunto de dados *backup* no pool *recvpool*. Adicionar `-d` ao `zfs recv` sobrescreve o nome do pool no lado de recebimento com o nome do snapshot. A opção `-u` faz com que os sistemas de arquivos não sejam montados no lado do recebimento. Quando `-v` é incluído, mais detalhes sobre a transferência são mostrados, incluindo o tempo decorrido e a quantidade de dados transferidos.

19.4.8. Cotas para Datasets, Usuários e Grupos

As cotas para dataset são usadas para restringir a quantidade de espaço que pode ser consumida

por um determinado conjunto de dados. As cotas de referência funcionam basicamente da mesma maneira, mas contam apenas o espaço usado pelo próprio conjunto de dados, excluindo snapshots e conjuntos de dados filho. Da mesma forma, as cotas para usuário e para grupo podem ser usadas para impedir que usuários ou grupos usem todo o espaço do pool ou do conjunto de dados.

Os exemplos a seguir pressupõem que os usuários já existam no sistema. Antes de adicionar um usuário ao sistema, certifique-se de criar seu dataset antes e defina o seu `mountpoint` para `/home/bob`. Em seguida, crie o usuário e faça com que o diretório inicial aponte para a localização do `mountpoint` do dataset. Isso definirá corretamente as permissões de proprietário e grupo sem obscurecer nenhum caminho de diretório inicial pré-existente que possa existir.

Para impor uma cota de dataser de 10 GB para o storage/home/bob:

```
# zfs set quota=10G storage/home/bob
```

Para impor uma cota de referência de 10 GB para storage/home/bob:

```
# zfs set refquota=10G storage/home/bob
```

Para remover uma cota de 10 GB do storage/home/bob:

```
# zfs set quota=none storage/home/bob
```

O formato geral é `userquota@user=size` e o nome do usuário deve estar em um destes formatos:

- nome compatível com o POSIX, como *joe*.
- ID numérico POSIX, como *789*.
- nome SID, como *joe.bloggs@example.com*.
- ID numérico SID , como *S-1-123-456-789*.

Por exemplo, para impor uma cota de usuário de 50 GB para o usuário chamado *joe*:

```
# zfs set userquota@joe=50G
```

Para remover qualquer cota:

```
# zfs set userquota@joe=none
```



As propriedades da cota do usuário não são exibidas pelo `zfs get all`. Os usuários que não são o `root` só podem ver suas próprias cotas, a menos que tenham recebido o privilégio `userquota`. Os usuários com esse privilégio podem visualizar e definir a cota de todos.

O formato geral para definir uma cota de grupo é: `groupquota@group=size`.

Para definir a cota do grupo *firstgroup* para 50 GB, use:

```
# zfs set groupquota@firstgroup=50G
```

Para remover a cota do grupo *firstgroup* ou para certificar-se de que uma não está definida, use:

```
# zfs set groupquota@firstgroup=none
```

Assim como a propriedade de cota do usuário, os usuários que não são `root` só podem ver as cotas associadas aos grupos aos quais eles pertencem. No entanto, o `root` ou um usuário com o privilégio `groupquota` pode visualizar e definir todas as cotas para todos os grupos.

Para exibir a quantidade de espaço utilizada por cada usuário em um sistema de arquivos ou snapshot junto com quaisquer cotas, use `zfs userspace`. Para informações de grupo, use `zfs groupspace`. Para obter maiores informações sobre opções suportadas ou sobre como exibir apenas opções específicas, consulte zfs(1).

Usuários com privilégios suficientes, e o `root`, podem listar a cota para storage/home/bob usando:

```
# zfs get quota storage/home/bob
```

19.4.9. Reservas

As reservas garantem uma quantidade mínima de espaço sempre disponível em um conjunto de dados. O espaço reservado não estará disponível para nenhum outro conjunto de dados. Esse recurso pode ser especialmente útil para garantir que haja espaço livre disponível para um conjunto de dados ou arquivos de log importantes.

O formato geral da propriedade `reservation` é `reservation=size`, portanto, para definir uma reserva de 10 GB em storage/home/bob, use:

```
# zfs set reservation=10G storage/home/bob
```

Para cancelar qualquer reserva:

```
# zfs set reservation=none storage/home/bob
```

O mesmo princípio pode ser aplicado à propriedade `refreservation` para definir uma Reserva de Referência, com o formato geral `refreservation=size`.

Este comando mostra todas as reservas ou atualizações existentes no storage/home/bob:

```
# zfs get reservation storage/home/bob
# zfs get refreservation storage/home/bob
```

19.4.10. Compressão

O ZFS fornece compactação transparente. A compactação de dados no nível do bloco a medida que ele é escrito, não apenas economiza espaço, mas também pode aumentar a performance do disco. Se os dados forem compactados em 25%, mas os dados compactados forem gravados no disco na mesma taxa da versão descompactada, resulta em uma velocidade efetiva de gravação de 125%. A compactação também pode ser uma ótima alternativa para Deduplicação porque não requer memória adicional.

O ZFS oferece vários algoritmos de compactação diferentes, cada um com diferentes compensações. Com a introdução da compactação LZ4 no ZFS v5000, é possível ativar a compactação para todo o pool sem o trade-off de desempenho de outros algoritmos. A maior vantagem do LZ4 é o recurso *early abort*. Se o LZ4 não atingir pelo menos 12,5% de compactação na primeira parte dos dados, o bloco será gravado descompactado para evitar o desperdício de ciclos da CPU que tentam compactar dados já compactados ou não compactáveis. Para obter detalhes sobre os diferentes algoritmos de compactação disponíveis no ZFS, consulte a entrada Compactação na seção de terminologia.

O administrador pode monitorar a eficácia da compactação usando várias propriedades do conjunto de dados.

```
# zfs get used,compressratio,compression,logicalused mypool/compressed_dataset
NAME        PROPERTY          VALUE     SOURCE
mypool/compressed_dataset  used              449G      -
mypool/compressed_dataset  compressratio     1.11x     -
mypool/compressed_dataset  compression       lz4       local
mypool/compressed_dataset  logicalused       496G      -
```

O conjunto de dados está usando atualmente 449 GB de espaço (a propriedade used). Sem compressão, seriam necessários 496 GB de espaço (a propriedade `logicalused`). Isso resulta na taxa de compactação de 1,11: 1.

A compactação pode ter um efeito colateral inesperado quando combinada com cotas de usuário. As cotas de usuários restringem a quantidade de espaço que um usuário pode consumir em um conjunto de dados, mas as medidas são baseadas em quanto espaço é usado *após a compactação*. Portanto, se um usuário tiver uma cota de 10 GB e gravar 10 GB de dados compactáveis, eles ainda poderão armazenar dados adicionais. Se, posteriormente, atualizarem um arquivo, digamos um banco de dados, com dados mais ou menos compactáveis, a quantidade de espaço disponível para eles será alterada. Isso pode resultar na situação ímpar em que um usuário não aumentou a quantidade real de dados (a propriedade `logicalused`), mas a alteração na compactação fez com que eles atingissem seu limite de cota.

A compactação pode ter uma interação inesperada semelhante com backups. Muitas vezes, as cotas são usadas para limitar a quantidade de dados que podem ser armazenados para garantir que haja

espaço de backup suficiente disponível. No entanto, uma vez que as cotas não consideram a compactação, mais dados podem ser gravados do que caberia com os backups descompactados.

19.4.11. Desduplicação

Quando ativado, a deduplicação usa o checksum de cada bloco para detectar blocos duplicados. Quando um novo bloco é uma duplicata de um bloco existente, o ZFS grava uma referência adicional aos dados existentes, em vez de todo o bloco duplicado. Uma enorme economia de espaço é possível se os dados contiverem muitos arquivos duplicados ou informações repetidas. Esteja avisado: a desduplicação requer uma quantidade extremamente grande de memória, e a maior parte da economia de espaço pode ser obtida sem o custo extra, permitindo a compactação.

Para ativar a deduplicação, defina a propriedade `dedup` no pool de destino:

```
# zfs set dedup=on pool
```

Somente novos dados sendo gravados no pool serão desduplicados. Os dados que já foram gravados no pool não serão desduplicados simplesmente ativando essa opção. Um pool com uma propriedade de desduplicação ativada recentemente será semelhante a este exemplo:

```
# zpool list
NAME  SIZE ALLOC  FREE   CKPOINT  EXPANDSZ   FRAG   CAP   DEDUP   HEALTH   ALTROOT
pool 2.84G 2.19M 2.83G         -         -     0%    0%   1.00x   ONLINE   -
```

A coluna `DEDUP` mostra a taxa real de deduplicação para o pool. Um valor de `1.00x` mostra que os dados ainda não foram desduplicados. No próximo exemplo, a árvore de ports é copiada três vezes em diretórios diferentes no pool desduplicado criado acima.

```
# for d in dir1 dir2 dir3; do
> mkdir $d && cp -R /usr/ports $d &
> done
```

Dados redundantes são detectados e desduplicados:

```
# zpool list
NAME SIZE  ALLOC  FREE   CKPOINT  EXPANDSZ   FRAG  CAP   DEDUP   HEALTH   ALTROOT
pool 2.84G 20.9M 2.82G         -         -     0%   0%   3.00x   ONLINE   -
```

A coluna `DEDUP` mostra um fator de `3.00x`. Várias cópias dos dados da árvore de ports foram detectadas e desduplicadas, usando apenas um terço do espaço. O potencial de economia de espaço pode ser enorme, mas com o custo de ter memória suficiente para rastrear os blocos desduplicados.

A desduplicação nem sempre é benéfica, especialmente quando os dados em um pool não são redundantes. O ZFS pode mostrar uma possível economia de espaço ao simular a desduplicação em um pool existente:

```
# zdb -S pool
Simulated DDT histogram:

bucket              allocated                       referenced
______   ______________________________   ______________________________
refcnt   blocks   LSIZE   PSIZE   DSIZE   blocks   LSIZE   PSIZE   DSIZE
------   ------   -----   -----   -----   ------   -----   -----   -----
     1    2.58M    289G    264G    264G    2.58M    289G    264G    264G
     2     206K   12.6G   10.4G   10.4G     430K   26.4G   21.6G   21.6G
     4    37.6K    692M    276M    276M     170K   3.04G   1.26G   1.26G
     8    2.18K   45.2M   19.4M   19.4M    20.0K    425M    176M    176M
    16      174   2.83M   1.20M   1.20M    3.33K   48.4M   20.4M   20.4M
    32       40   2.17M    222K    222K    1.70K   97.2M   9.91M   9.91M
    64        9     56K   10.5K   10.5K      865   4.96M    948K    948K
   128        2   9.50K      2K      2K      419   2.11M    438K    438K
   256        5   61.5K     12K     12K    1.90K   23.0M   4.47M   4.47M
    1K        2      1K      1K      1K    2.98K   1.49M   1.49M   1.49M
 Total    2.82M    303G    275G    275G    3.20M    319G    287G    287G

dedup = 1.05, compress = 1.11, copies = 1.00, dedup * compress / copies = 1.16
```

Depois que o `zdb -S` termina de analisar o pool, ele mostra a taxa de redução de espaço que seria obtida ativando a deduplicação. Nesse caso, `1.16` é uma taxa de economia de espaço muito baixa e que poderia ser obtida apenas com a compactação. A ativação da deduplicação neste pool não salvaria uma quantidade significativa de espaço e não vale a quantidade de memória necessária para ativar a deduplicação. Usando a fórmula *ratio = dedup * compress / copies*, os administradores do sistema podem planejar a alocação de armazenamento, decidindo se a carga de trabalho conterá blocos duplicados suficientes para justificar os requisitos de memória. Se os dados forem razoavelmente compactáveis, a economia de espaço poderá ser muito boa. Recomenda-se ativar a compactação primeiro pois ela também pode aumentar significativamente a performance do sistema. Ative a deduplicação somente nos casos em que a economia adicional será considerável e se houver memória suficiente para o DDT.

19.4.12. ZFS e Jails

O `zfs jail` e a propriedade `jailed` correspondente são usadas para delegar um conjunto de dados ZFS para uma Jail. O `zfs jail jailid` anexa um dataset à jail especificada, e o `zfs unjail` o desanexa. Para que o conjunto de dados seja controlado de dentro de um jail, a propriedade `jailed` deve ser configurada. Depois que um conjunto de dados é anexado a um jail, ele não pode mais ser montado no host porque ele poderá ter pontos de montagem que comprometam a segurança do host.

19.5. Administração Delegada

Um sistema abrangente de delegação de permissão permite que usuários sem privilégios realizem funções de administração do ZFS. Por exemplo, se o diretório pessoal de cada usuário for um conjunto de dados, os usuários poderão receber permissão para criar e destruir snapshots de seus diretórios pessoais. Um usuário de backup pode receber permissão para usar recursos de

replicação. Um script de estatísticas de uso pode ter permissão para ser executado com acesso apenas aos dados de utilização de espaço para todos os usuários. É ainda possível delegar a capacidade de delegar permissões. A delegação de permissão é possível para cada subcomando e para a maioria das propriedades.

19.5.1. Delegando a criação de conjunto de dados

O `zfs allow someuser create mydataset` concede ao usuário especificado permissão para criar conjuntos de dados filho sob o conjunto de dados pai selecionado. Há uma ressalva: criar um novo conjunto de dados envolve montá-lo. Isso requer configurar o `vfs.usermount` sysctl(8) do FreeBSD para 1 para permitir que usuários não-root montem um sistema de arquivos. Existe outra restrição que visa impedir o abuso: os usuários que não são `root` devem ser donos do ponto de montagem onde o sistema de arquivos deve ser montado.

19.5.2. Delegando a delegação de permissão

O `zfs allow someuser allow mydataset` permite ao usuário especificado atribuir qualquer permissão que tenha no conjunto de dados de destino, ou nos seus filhos, para outros usuários . Se um usuário tiver a permissão `snapshot` e a permissão `allow`, esse usuário poderá conceder a permissão `snapshot` para outros usuários.

19.6. Tópicos Avançados

19.6.1. Otimizações

Existem vários parametros que podem ser ajustados para tornar o ZFS melhor para diferentes cargas de trabalho.

- `vfs.zfs.arc_max` - Tamanho máximo do ARC. O padrão é toda a memória RAM menos 1 GB, ou metade da RAM, o que for maior. No entanto, um valor menor deve ser usado se o sistema estiver executando quaisquer outros daemons ou processos que possam requerer memória. Este valor pode ser ajustado em tempo de execução com sysctl(8) e pode ser configurado no /boot/loader.conf ou /etc/sysctl.conf.
- `vfs.zfs.arc_meta_limit` - Limita a parte do ARC que pode ser usado para armazenar metadados. O padrão é um quarto de `vfs.zfs.arc_max`. Aumentar esse valor melhorará o desempenho se a carga de trabalho envolver operações em um grande número de arquivos e diretórios ou operações de metadados frequentes, ao custo de caber menos dados de arquivo no ARC. Este valor pode ser ajustado em tempo de execução com sysctl(8) e pode ser configurado em /boot/loader.conf ou /etc/sysctl.conf.
- `vfs.zfs.arc_min` - Tamanho mínimo do ARC. O padrão é metade de `vfs.zfs.arc_meta_limit`. Ajuste esse valor para evitar que outros aplicativos pressionem o ARC inteiro. Este valor pode ser ajustado em tempo de execução com sysctl(8) e pode ser configurado em /boot/loader.conf ou /etc/sysctl.conf.
- `vfs.zfs.vdev.cache.size` - Uma quantidade pré-alocada de memória reservada como um cache para cada dispositivo no pool. A quantidade total de memória usada será esse valor multiplicado pelo número de dispositivos. Este valor só pode ser ajustado no momento da inicialização e é definido em /boot/loader.conf.

- `vfs.zfs.min_auto_ashift` - Mínimo `ashift` (tamanho do setor) que será usado automaticamente no momento da criação do pool. O valor é uma potência de dois. O valor padrão de `9` representa `2^9 = 512`, um tamanho de setor de 512 bytes. Para evitar *amplificação de escrita* e para obter o melhor desempenho, defina esse valor para o maior tamanho de setor usado por um dispositivo no pool.

  Muitas unidades possuem setores de 4 KB. Usar o `ashift` padrão `9` com esses drives resulta em amplificação de gravação nesses dispositivos. Os dados que podem estar contidos em uma única gravação de 4 KB devem, em vez disso, ser gravados em oito gravações de 512 bytes. O ZFS tenta ler o tamanho do setor nativo de todos os dispositivos ao criar um pool, mas muitas unidades com setores de 4 KB relatam que seus setores têm 512 bytes para compatibilidade. Configure o `vfs.zfs.min_auto_ashift` para `12` (`2^12=4096`) antes de criar um pool irá forçar o ZFS a usar blocos de 4 KB para melhor desempenho nessas unidades.

  Forçar blocos de 4 KB também é útil em pools em que as atualizações de disco são planejadas. Os discos futuros provavelmente usarão setores de 4 KB, e os valores de `ashift` não poderão ser alterados depois que um pool for criado.

  Em alguns casos específicos, o menor tamanho de bloco de 512 bytes pode ser preferível. Quando usado com discos de 512 bytes para bancos de dados, ou como armazenamento para máquinas virtuais, menos dados são transferidos durante pequenas leituras aleatórias. Isso pode fornecer melhor desempenho, especialmente ao usar um tamanho de registro ZFS menor.

- `vfs.zfs.prefetch_disable` - Desabilita a pré-busca. Um valor de `0` está ativado e `1` está desativado. O padrão é `0`, a menos que o sistema tenha menos de 4 GB de RAM. A pré-busca funciona lendo blocos maiores do que os que foram solicitados no ARC na esperança de que os dados sejam necessários em breve. Se a carga de trabalho tiver um grande número de leituras aleatórias, a desativação da pré-busca poderá melhorar o desempenho reduzindo leituras desnecessárias. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.vdev.trim_on_init` - Controla se os novos dispositivos adicionados ao pool têm o comando `TRIM` executado neles. Isso garante o melhor desempenho e a longevidade dos SSDs, mas leva um tempo extra. Se o dispositivo já tiver sido apagado de forma segura, a desativação dessa configuração tornará o acréscimo do novo dispositivo mais rápido. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.vdev.max_pending` - Limita o número de solicitações de I/O pendentes por dispositivo. Um valor mais alto manterá a fila de comandos do dispositivo cheia e poderá resultar em maior rendimento. Um valor menor reduzirá a latência. Este valor pode ser ajustado a qualquer momento com o sysctl(8).
- `vfs.zfs.top_maxinflight` - Número máximo de I/Os pendentes por vdev de nível superior. Limita a profundidade da fila de comandos para evitar alta latência. O limite é por vdev de nível superior, o que significa que o limite se aplica a cada Mirror, RAID-Z, ou outro vdev independentemente. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.l2arc_write_max` - Limita a quantidade de dados gravados no L2ARC por segundo. Este ajuste foi projetado para estender a longevidade de SSDs limitando a quantidade de dados gravados no dispositivo. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.l2arc_write_boost` - O valor deste ajuste é adicionado ao `vfs.zfs.l2arc_write_max` e aumenta a velocidade de gravação para o SSD até que o primeiro bloco seja removido do L2ARC.

Esta "Turbo Warmup Phase" é projetada para reduzir a perda de desempenho de um L2ARC vazio após uma reinicialização. Este valor pode ser ajustado a qualquer momento com sysctl(8).

- `vfs.zfs.scrub_delay` - Número de ticks a serem atrasados entre cada operação de I/O durante um `scrub`. Para garantir que um `scrub` não interfira com a operação normal do pool, se qualquer outra I/O estiver acontecendo, o `scrub` será atrasado entre cada comando. Esse valor controla o limite no total de IOPS (I/Os por segundo) gerados pelo `scrub`. A granularidade da configuração é determinada pelo valor de `kern.hz`, cujo padrão é de 1.000 ticks por segundo. Essa configuração pode ser alterada, resultando em um limite efetivo de IOPS diferente. O valor padrão é `4`, resultando em um limite de: 1000 ticks/seg/4 = 250 IOPS. Usar um valor de *20* daria um limite de: 1000 ticks/seg/20 = 50 IOPS. A velocidade de `scrub` é limitada apenas quando houver atividade recente no pool, conforme determinado por `vfs.zfs.scan_idle`. Esse valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.resilver_delay` - Número de milissegundos de atraso inserido entre cada I/O durante um resilver . Para garantir que um resilver não interfira com a operação normal do pool, se qualquer outro I/O estiver acontecendo, o resilver irá atrasar entre cada comando. Esse valor controla o limite de total de IOPS (I/Os por segundo) gerados pelo resilver. A granularidade da configuração é determinada pelo valor de `kern.hz`, cujo padrão é de 1.000 marcações por segundo. Essa configuração pode ser alterada, resultando em um limite efetivo de IOPS diferente. O valor padrão é 2, resultando em um limite de: 1000 ticks / seg / 2 = 500 IOPS. Retornar o pool a um estado Online pode ser mais importante se a falha outro dispositivo levar o pool ao estado de Fault, causando perda de dados. Um valor de 0 dará à operação de resilver a mesma prioridade que outras operações, acelerando o processo de recuperação. A velocidade do resilver é limitada apenas quando houver outra atividade recente no pool, conforme determinado por `vfs.zfs.scan_idle`. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.scan_idle` - Número de milissegundos desde a última operação antes do pool ser considerado ocioso. Quando o pool estiver ocioso, a taxa limite para `scrub` e resilver fica desativada. Este valor pode ser ajustado a qualquer momento com sysctl(8).
- `vfs.zfs.txg.timeout` - Número máximo de segundos entre os grupos de transações. O grupo de transações atual será gravado no pool e um novo grupo de transações será iniciado se esse período de tempo tiver decorrido desde o grupo de transações anterior. Um grupo de transações pode ser acionado antes se dados suficientes forem gravados. O valor padrão é de 5 segundos. Um valor maior pode melhorar o desempenho de leitura atrasando gravações assíncronas, mas isso pode causar um desempenho irregular quando o grupo de transações é gravado. Este valor pode ser ajustado a qualquer momento com sysctl(8).

19.6.2. ZFS em i386

Alguns dos recursos fornecidos pelo ZFS consomem muita memória, e podem exigir ajuste para máxima eficiência em sistemas com RAM limitada.

19.6.2.1. Memória

Como mínimo, a memória total do sistema deve ter pelo menos um gigabyte. A quantidade de RAM recomendada depende do tamanho do pool e dos recursos do ZFS usados. Uma regra geral é 1 GB de RAM para cada 1 TB de armazenamento. Se o recurso de deduplicação for usado, uma regra geral é 5 GB de RAM por TB de armazenamento para ser desduplicado. Enquanto alguns usuários

usam com sucesso o ZFS com menos RAM, os sistemas sob carga pesada podem entrar em panic devido ao esgotamento da memória. Outros ajustes podem ser necessários para sistemas com uma quantia de memória RAM inferior ao recomendado.

19.6.2.2. Configuração do Kernel

Devido às limitações de espaço de endereço da plataforma i386™, os usuários do ZFS na arquitetura i386™ devem adicionar essa opção a um arquivo de configuração de kernel personalizado, reconstruir o kernel e reiniciar:

```
options         KVA_PAGES=512
```

Isso expande o espaço de endereço do kernel, permitindo que o parametro `vm.kvm_size` seja ajustado além do limite imposto atualmente de 1 GB ou o limite de 2 GB para PAE. Para encontrar o valor mais adequado para essa opção, divida o espaço de endereço desejado em megabytes por quatro. Neste exemplo, é `512` para 2 GB.

19.6.2.3. Ajustes do Carregador

O espaço de endereçamento kmem pode ser aumentado em todas as arquiteturas do FreeBSD. Em um sistema de teste com 1 GB de memória física, o sucesso foi alcançado com essas opções abaixo adicionadas ao /boot/loader.conf, e o sistema reiniciado:

```
vm.kmem_size="330M"
vm.kmem_size_max="330M"
vfs.zfs.arc_max="40M"
vfs.zfs.vdev.cache.size="5M"
```

Para obter uma lista mais detalhada de recomendações para otimizações relacionadas ao ZFS, consulte https://wiki.freebsd.org/ZFSTuningGuide.

19.7. Recursos adicionais

- OpenZFS
- FreeBSD Wiki - ZFS Tuning
- Oracle Solaris ZFS Administration Guide
- Calomel Blog - ZFS Raidz Performance, Capacity and Integrity

19.8. Recursos e terminologia do ZFS

O ZFS é um sistema de arquivos fundamentalmente diferente, porque é mais do que apenas um sistema de arquivos. O ZFS combina as funções do sistema de arquivos e do gerenciador de volume, permitindo que dispositivos de armazenamento adicionais sejam adicionados a um sistema ativo e torne o novo espaço disponível em todos os sistemas de arquivos existentes nesse pool imediatamente. Combinando os papéis tradicionalmente separados, o ZFS é capaz de superar

limitações anteriores que impediam o crescimento de grupos RAID. Cada dispositivo de nível superior em um pool é chamado de *vdev*, que pode ser um disco simples ou uma transformação RAID como um espelho ou array RAID-Z. Os sistemas de arquivos ZFS (chamados *datasets*) têm acesso ao espaço livre combinado de todo o pool. À medida que os blocos são alocados do pool, o espaço disponível para cada sistema de arquivos diminui. Essa abordagem evita a armadilha comum com o particionamento extensivo onde o espaço livre se torna fragmentado nas partições.

pool	Um *pool* de armazenamento é o bloco de construção mais básico do ZFS. Um pool é composto de um ou mais vdevs, os dispositivos subjacentes que armazenam os dados. Um pool é então usado para criar um ou mais sistemas de arquivos (datasets) ou dispositivos de bloco (volumes). Esses conjuntos de dados e volumes compartilham o espaço livre restante do pool. Cada pool é identificado exclusivamente por um nome e um GUID. Os recursos disponíveis são determinados pelo número da versão do ZFS no pool.

vdev Types	Um pool é composto de um ou mais vdevs, que podem ser um único disco ou um grupo de discos, no caso de uma transformação RAID. Quando vários vdevs são usados, o ZFS propaga dados entre os vdevs para aumentar o desempenho e maximizar o espaço utilizável. • *Disk* - O tipo mais básico de vdev é um dispositivo de bloco padrão. Isso pode ser um disco inteiro (como /dev/ada0 ou /dev/da0) ou uma partição (/dev/ada0p3). No FreeBSD, não há penalidade de desempenho por usar uma partição em vez de todo o disco. Isso difere das recomendações feitas pela documentação do Solaris.  Usar um disco inteiro como parte de um pool inicializável é altamente desencorajado, pois isso pode tornar o pool não inicializável. Da mesma forma, você não deve usar um disco inteiro como parte de um mirror ou um RAID-Z vdev. Isso ocorre porque é impossível determinar com segurança o tamanho de um disco não particionado no momento da inicialização e porque não há lugar para inserir código de inicialização. • *File*- Além dos discos, os pools do ZFS podem ser suportados por arquivos regulares, o que é especialmente útil para testes e experimentação. Use o caminho completo para o arquivo como o caminho do dispositivo no `zpool create`. Todos os vdevs devem ter pelo menos 128 MB de tamanho. • *Mirror* - Ao criar um espelho, especifique a palavra-chave `mirror` seguida pela lista de dispositivos membros para o espelho. Um espelho consiste em dois ou mais dispositivos, todos os dados serão gravados em todos os dispositivos membros. Um espelho vdev só armazenará tantos dados quanto seu menor membro. Um espelho vdev pode suportar a falha de todos, exceto um de seus membros, sem perder nenhum dado.  Um vdev de disco único regular pode ser atualizado para um vdev de espelho a qualquer momento com `zpool attach`. • *RAID-Z* - O ZFS implementa o RAID-Z, uma variação do padrão RAID-5 que oferece uma melhor distribuição de paridade e elimina o furo de escrita do "RAID-5" no qual os dados e informações de paridade tornam-se inconsistentes após um reinício inesperado. O ZFS suporta três níveis de RAID-Z, que fornecem vários níveis de redundância em troca de níveis decrescentes de armazenamento utilizável. Os tipos são nomeados de RAID-Z1 até RAID-Z3 com base no número de dispositivos de paridade na matriz e no número de discos que podem falhar enquanto o pool permanece operacional. Em uma configuração de RAID-Z1 com quatro discos, cada um com 1 TB, resultará em um volume com armazenamento utilizável de 3 TB e o pool

Transaction Group (TXG)	Grupos de transações são a forma como os blocos alterados são agrupados e eventualmente gravados no pool. Grupos de transação são a unidade atômica que o ZFS usa para garantir a consistência. Cada grupo de transações recebe um identificador consecutivo exclusivo de 64 bits. Pode haver até três grupos de transações ativos por vez, um em cada um desses três estados: * *Open* - Quando um novo grupo de transações é criado, ele está no estado aberto e aceita novas gravações. Há sempre um grupo de transações no estado aberto, no entanto, o grupo de transações pode recusar novas gravações se tiver atingido um limite. Quando o grupo de transações abertas tiver atingido um limite ou o `vfs.zfs.txg.timeout` tiver sido alcançado, o grupo de transações avança para o próximo estado. * *Quiescing* - Um estado curto que permite que qualquer operação pendente termine sem bloquear a criação de um novo grupo de transações abertas. Depois que todas as transações no grupo forem concluídas, o grupo de transações avançará para o estado final. * *Syncing* - Todos os dados no grupo de transações são gravados no armazenamento estável. Esse processo, por sua vez, modificará outros dados, como metadados e mapas de espaço, que também precisarão ser gravados no armazenamento estável. O processo de sincronização envolve vários passos. O primeiro é o maior, e trata de todos os blocos de dados alterados, seguido pelos metadados, que podem levar vários passos para serem concluídos. Como a alocação de espaço para os blocos de dados gera novos metadados, o estado de sincronização não pode ser concluído até que um passo seja concluído e não aloque espaço adicional. O estado de sincronização também é onde as *synctasks* são concluídas. As operações de sincronização são operações administrativas, como criar ou destruir snapshots e datasets, que modificam o uberblock. Quando o estado de sincronização estiver concluído, o grupo de transações no estado de quiesce é avançado para o estado de sincronização. Todas as funções administrativas, tal como `snapshot` são gravados como parte do grupo de transações. Quando uma tarefa de sincronização é criada, ela é adicionada ao grupo de transações atualmente aberto e esse grupo é avançado o mais rápido possível para o estado de sincronização para reduzir a latência de comandos administrativos.

Adaptive Replacement Cache (ARC)	O ZFS usa um Cache Adaptativo de Substituição (ARC), em vez de um mais tradicional como o Least Recently Used (LRU). Um cache LRU é uma lista simples de itens no cache, classificados por quando cada objeto foi usado mais recentemente. Novos itens são adicionados ao topo da lista. Quando o cache está cheio, os itens da parte inferior da lista são despejados para liberar espaço para mais objetos ativos. Um ARC consiste em quatro listas; os objetos Mais Recentes Utilizados (MRU) e Mais Frequentemente Usados (MFU), além de uma lista fantasma para cada um. Essas listas fantasmas rastreiam objetos recentemente despejados para evitar que sejam adicionados de volta ao cache. Isso aumenta a taxa de acertos do cache evitando objetos que têm um histórico de serem usados apenas ocasionalmente. Outra vantagem de usar um MRU e um MFU é que a verificação de um sistema de arquivos inteiro normalmente despejaria todos os dados de um MRU ou LRU do cache em favor deste conteúdo recém-acessado. Com o ZFS, há também um MFU que rastreia apenas os objetos usados com mais freqüência, e o cache dos blocos acessados com mais frequência permanece.
L2ARC	O L2ARC é o segundo nível do sistema de armazenamento em cache do ZFS. O ARC principal é armazenado em RAM. Como a quantidade de RAM disponível é limitada, o ZFS também pode usar cache vdevs. Discos de estado sólido (SSDs) geralmente são usados como esses dispositivos de cache devido à sua maior velocidade e menor latência em comparação aos discos mecânicos tradicionais. O L2ARC é totalmente opcional, mas um deles aumentará significativamente a velocidade de leitura dos arquivos armazenados em cache no SSD em vez de precisar ser lido nos discos normais. O L2ARC também pode acelerar a desduplicação porque um DDT que não cabe na RAM mas cabe no L2ARC será muito mais rápido que um DDT que deve ser lido do disco. A taxa na qual os dados são adicionados aos dispositivos de cache é limitada para evitar o desgaste prematuro dos SSDs com muitas gravações. Até que o cache esteja cheio (o primeiro bloco foi removido para liberar espaço), a gravação no L2ARC é limitada à soma do limite de gravação e do limite de aumento e depois limitada ao limite de gravação. Um par de valores sysctl(8) controla esses limites de taxa. A `vfs.zfs.l2arc_write_max` controla quantos bytes são gravados no cache por segundo, enquanto `vfs.zfs.l2arc_write_boost` adiciona a este limite durante a "Turbo Warmup Phase " (aumento de gravação).
ZIL	O ZIL acelera as transações síncronas usando dispositivos de armazenamento como SSDs mais rápidos do que os usados no pool de armazenamento principal. Quando um aplicativo solicita uma gravação síncrona (uma garantia de que os dados foram armazenados com segurança no disco, em vez de simplesmente serem gravados posteriormente), os dados são gravados no armazenamento mais rápido de ZIL e, depois, liberados aos discos regulares. Isso reduz enormemente a latência e melhora o desempenho. Apenas cargas de trabalho síncronas, como bancos de dados, serão beneficiadas com um ZIL. Gravações assíncronas regulares, como copiar arquivos, não usarão o ZIL.

Copy-On-Write	Ao contrário de um sistema de arquivos tradicional, quando os dados são sobrescritos no ZFS, os novos dados são gravados em um bloco diferente, em vez de sobrescrever os dados antigos no lugar. Somente quando essa gravação for concluída, os metadados serão atualizados para apontar para o novo local. No caso de uma gravação simplificada (uma falha do sistema ou perda de energia no meio da gravação de um arquivo), todo o conteúdo original do arquivo ainda estará disponível e a gravação incompleta será descartada. Isso também significa que o ZFS não requer um fsck(8) após um desligamento inesperado.
Dataset	*Dataset* é o termo genérico para um sistema de arquivos ZFS, volume, snapshot ou clone. Cada dataset tem um nome exclusivo no formato *poolname/path@snapshot*. A raiz do pool é tecnicamente um dataset também. Dataset filhos são nomeados hierarquicamente como diretórios. Por exemplo, *mypool/home*, o dataset inicial, é um filho de *mypool* e herda propriedades dele. Isso pode ser expandido ainda mais criando o *mypool/home/user*. Este dataset neto herdará propriedades do pai e do avô. As propriedades de um filho podem ser definidas para substituir os padrões herdados dos pais e avós. A administração de datasets e seus filhos pode ser delegada.
File system	Um dataset ZFS é mais frequentemente usado como um sistema de arquivos. Como a maioria dos outros sistemas de arquivos, um sistema de arquivos ZFS é montado em algum lugar na hierarquia de diretórios do sistema e contém arquivos e diretórios próprios com permissões, sinalizadores e outros metadados.
Volume	Além dos datasets regulares do sistema de arquivos, o ZFS também pode criar volumes, que são dispositivos de bloco. Os volumes têm muitos dos mesmos recursos, incluindo copy-on-write, snapshots, clones e checksum. Os volumes podem ser úteis para executar outros formatos de sistema de arquivos sobre o ZFS, tal como a virtualização do UFS ou a exportação de extensões iSCSI.

Snapshot	O design copy-on-write (COW) do ZFS permite snapshots quase instantâneos e consistentes com nomes arbitrários. Depois de obter um snapshot de um dataset ou um snapshot recursivo de um dataset pai que incluirá todos os datasets filho, novos dados serão gravados em novos blocos, mas os blocos antigos não serão recuperados como espaço livre. O snapshot contém a versão original do sistema de arquivos e o sistema de arquivos em tempo real contém as alterações feitas desde que o snapshot foi feito. Nenhum espaço adicional é usado. Conforme novos dados são gravados no sistema de arquivos ao vivo, novos blocos são alocados para armazenar esses dados. O tamanho aparente do snapshot aumentará à medida que os blocos não forem mais usados no sistema de arquivos ativo, mas apenas no snapshot. Estes snapshots podem ser montados somente como leitura para permitir a recuperação de versões anteriores de arquivos. Também é possível reverter um sistema de arquivos ativo para um snapshot específico, desfazendo quaisquer alterações que ocorreram depois que o snapshot foi tirado. Cada bloco no pool tem um contador de referência que registra quantos snapshots, clones, datasets ou volumes fazem uso desse bloco. À medida que arquivos e snapshots são excluídos, a contagem de referência é diminuída. Quando um bloco não é mais referenciado, ele é recuperado como espaço livre. Os snapshots também podem ser marcados com um hold. Quando um snapshot é mantido, qualquer tentativa de destruí-lo retornará um erro `EBUSY`. Cada snapshot pode ter várias retenções, cada uma com um nome exclusivo. O comando release remove a retenção para que o snapshot possa ser excluído. Snapshots podem ser obtidos de volumes, mas eles só podem ser clonados ou revertidos, não montados independentemente.
Clone	Os snapshots também podem ser clonados. Um clone é uma versão gravável de um snapshot, permitindo que o sistema de arquivos seja bifurcado como um novo dataset. Como com um snapshot, um clone inicialmente não consome espaço adicional. Conforme novos dados são gravados em um clone e novos blocos são alocados, o tamanho aparente do clone aumenta. Quando os blocos são sobrescritos no sistema de arquivos ou no volume clonado, a contagem de referência no bloco anterior é diminuída. O snapshot no qual um clone é baseado não pode ser excluído porque o clone depende dele. O snapshot é o pai e o clone é o filho. Os clones podem ser *promovidos*, invertendo essa dependência e tornando o clone o pai e o pai anterior, o filho. Esta operação não requer espaço adicional. Como a quantidade de espaço usada pelo pai e pelo filho é revertida, as cotas e reservas existentes podem ser afetadas.

Checksum	Cada bloco alocado também é verificado. O algoritmo de checksum usado é uma propriedade por dataset, consulte `set`. O checksum de cada bloco é validado de forma transparente à medida que é lido, permitindo que o ZFS detecte a corrupção silenciosa. Se os dados lidos não corresponderem à checksum esperada, o ZFS tentará recuperar os dados de qualquer redundância disponível, como espelhos ou RAID-Z). A validação de todos os checksums pode ser acionada com o `scrub`. Os algoritmos de checksum incluem: * `fletcher2` * `fletcher4` * `sha256` Os algoritmos `fletcher` são mais rápidos, mas o `sha256` é um hash criptográfico forte e tem uma chance muito menor de colisões ao custo de algum desempenho. Checksums podem ser desativados, mas isso não é recomendado.
Compression	Cada dataset tem uma propriedade de compactação, cujo padrão é off. Essa propriedade pode ser definida como um dos vários algoritmos de compactação. Isso fará com que todos os novos dados gravados no dataset sejam compactados. Além de uma redução no espaço usado, a taxa de leitura e gravação geralmente aumenta porque menos blocos são lidos ou gravados. * *LZ4* - Adicionado na versão 5000 do pool do ZFS (feature flags), o LZ4 é agora o algoritmo de compressão recomendado. O LZ4 compacta aproximadamente 50% mais rápido do que o LZJB ao operar em dados compactáveis e é três vezes mais rápido ao operar em dados não compactáveis. O LZ4 também descompacta aproximadamente 80% mais rápido que o LZJB. Nas CPUs modernas, o LZ4 pode frequentemente comprimir a mais de 500 MB/s e descompactar a mais de 1,5 GB/s (por núcleo de CPU). * *LZJB* - O algoritmo de compressão padrão. Criado por Jeff Bonwick (um dos criadores originais do ZFS). O LZJB oferece boa compactação com menos sobrecarga de CPU em comparação com o GZIP. No futuro, o algoritmo de compactação padrão provavelmente será alterado para LZ4. * *GZIP* - Um algoritmo popular de compressão de fluxo disponível no ZFS. Uma das principais vantagens de usar o GZIP é seu nível configurável de compactação. Ao definir a propriedade `compress`, o administrador pode escolher o nível de compactação, desde `gzip1`, o nível mais baixo de compactação, até `gzip9`, o maior nível de compressão. Isso dá ao administrador o controle sobre quanto tempo CPU será dedicado para economizar espaço em disco. * *ZLE* - A codificação de comprimento zero é um algoritmo de compressão especial que apenas comprime sequencias contínuas de zeros. Esse algoritmo de compactação é útil apenas quando o dataset contém grandes blocos de zeros.

Copies	Quando configurada para um valor maior que 1, a propriedade `copies` instrui o ZFS a manter várias cópias de cada bloco no sistema de arquivos ou volume. Definir essa propriedade em datasets importantes fornece redundância adicional a partir da qual recuperar um bloco que não corresponde ao seu checksum. Em pools sem redundância, o recurso de cópias é a única forma de redundância. O recurso de cópias pode se recuperar de um único setor defeituoso ou de outras formas de corrupção menor, mas não protege o pool da perda de um disco inteiro.
Deduplication	Os checksums permitem detectar blocos duplicados de dados à medida que são escritos. Com a deduplicação, a contagem de referência de um bloco existente e idêntico é aumentada, economizando espaço de armazenamento. Para detectar blocos duplicados, uma tabela de deduplicação (DDT) é mantida na memória. A tabela contém uma lista de checksums exclusivas, a localização desses blocos e uma contagem de referência. Quando novos dados são gravados, o checksum é calculado e comparado à lista. Se uma correspondência for encontrada, o bloco existente será usado. O algoritmo de checksum SHA256 é usado com deduplicação para fornecer um hash criptográfico seguro. A desduplicação é configurável. Se `dedup` for `on`, um checksum correspondente será considerado como significando que os dados são idênticos. Se `dedup` for definido como `verify`, os dados nos dois blocos serão verificados byte por byte para garantir que sejam realmente idênticos. Se os dados não forem idênticos, a colisão de hash será anotada e os dois blocos serão armazenados separadamente. Como o DDT deve armazenar o hash de cada bloco único, ele consome uma quantidade muito grande de memória. Uma regra geral é 5-6 GB de RAM por 1 TB de dados desduplicados). Em situações em que não é prático ter RAM suficiente para manter todo o DDT na memória, o desempenho sofrerá muito, pois o DDT deve ser lido do disco antes que cada novo bloco seja gravado. A desduplicação pode usar o L2ARC para armazenar o DDT, fornecendo um meio termo entre a memória rápida do sistema e os discos mais lentos. Considere a possibilidade de usar a compactação, que geralmente oferece quase a mesma economia de espaço sem o requisito de memória adicional.
Scrub	Em vez de uma verificação de consistência como o fsck(8), o ZFS tem o `scrub`. O `scrub` lê todos os blocos de dados armazenados no pool e verifica seus checksums em relação checksums bons conhecidos armazenados nos metadados. Uma verificação periódica de todos os dados armazenados no pool garante a recuperação de quaisquer blocos corrompidos antes que eles sejam necessários. Um scrub não é necessário após um desligamento inadequado do sistema, mas é recomendado pelo menos uma vez a cada três meses. O checksum de cada bloco é verificado à medida que os blocos são lidos durante o uso normal, mas um scrub garante que mesmo os blocos usados com pouca freqüência sejam verificados quanto a sua corrupção silenciosa. A segurança dos dados é aprimorada, especialmente em situações de armazenamento de arquivos. A prioridade relativa do `scrub` pode ser ajustada por meio da variável `vfs.zfs.scrub_delay` para evitar que o scrub degrade o desempenho de outras cargas de trabalho no pool.

Dataset Quota	O ZFS fornece datasets rápidos e precisos, contabilidade de espaço de usuários e grupos, além de cotas e reservas de espaço. Isso dá ao administrador um controle refinado sobre como o espaço é alocado e permite que o espaço seja reservado para sistemas de arquivos críticos. ZFS supports different types of quotas: the dataset quota, the reference quota (refquota), the user quota, and the group quota. As cotas limitam a quantidade de espaço que um dataset e todos os seus descendentes, incluindo snapshots do dataset, datasets filhos e snapshots desses datasets, podem consumir.  Cotas não podem ser definidas em volumes, pois a propriedade `volsize` atua como uma cota implícita.
Reference Quota	Uma cota de referência limita a quantidade de espaço que um dataset pode consumir impondo um limite rígido. No entanto, esse limite rígido inclui apenas o espaço ao qual o dataset faz referência e não inclui o espaço usado pelos descendentes, como sistemas de arquivos ou snapshots.
User Quota	Cotas de usuários são úteis para limitar a quantidade de espaço que pode ser usada pelo usuário especificado.
Group Quota	A cota de grupo limita a quantidade de espaço que um grupo especificado pode consumir.
Dataset Reservation	A propriedade `reservation` torna possível garantir uma quantidade mínima de espaço para um dataset específico e seus descendentes. Se uma reserva de 10 GB estiver definida em storage/home/bob, e outro dataset tentar usar todo o espaço livre, pelo menos 10 GB de espaço serão reservados para este dataset. Se um snapshot for criado de storage/home/bob, o espaço usado por esse snapshot será contabilizado contra a reserva. A propriedade `refreservation` funciona de maneira semelhante, mas *exclui* os descendentes como os snapshots. Reservas de qualquer tipo são úteis em muitas situações, como planejar e testar a adequação da alocação de espaço em disco em um novo sistema ou garantindo espaço suficiente nos sistemas de arquivos para logs de áudio ou procedimentos e arquivos de recuperação do sistema.

Reference Reservation	A propriedade `refreservation` torna possível garantir uma quantidade mínima de espaço para o uso de dataset específico *excluindo* seus descendentes. Isso significa que, se uma reserva de 10 GB estiver definida em storage/home/bob, e outro dataset tentar usar todo o espaço livre, pelo menos 10 GB de espaço serão reservados para este dataset. Em contraste com uma reserva regular, o espaço usado por snapshots e datasets descendentes não é contado contra a reserva. Por exemplo, se um snapshot for criado do storage/home/bob, deve haver espaço em disco suficiente fora da quantia de `refreservation` para que a operação seja bem-sucedida. Descendentes do dataset principal não são contados na quantia de `refreservation` e, portanto, não invadem o espaço definido.
Resilver	Quando um disco falha e é substituído, o novo disco deve ser preenchido com os dados perdidos. O processo de usar as informações de paridade distribuídas entre as unidades restantes para calcular e gravar os dados ausentes na nova unidade é chamado de *resilvering*.
Online	Um pool ou vdev no estado `Online` tem todos os seus dispositivos membros conectados e totalmente operacionais. Dispositivos individuais no estado `Online` estão funcionando normalmente.
Offline	Dispositivos individuais podem ser colocados em um estado `Offline` pelo administrador se houver redundância suficiente para evitar colocar o pool ou vdev em um estado Faulted. Um administrador pode optar por colocar um disco off-line como preparação para substituí-lo ou para facilitar sua identificação.
Degraded	Um pool ou vdev no estado `Degraded` possui um ou mais discos que foram desconectados ou falharam. O pool ainda é utilizável, mas se dispositivos adicionais falharem, o pool poderá se tornar irrecuperável. Reconectar os dispositivos ausentes ou substituir os discos com falha retornará o pool a um estado Online depois que o dispositivo reconectado ou novo tiver concluído o processo de Resilver.
Faulted	Um pool ou vdev no estado `Faulted` não está mais operacional. Os dados nele não podem mais ser acessados. Um pool ou vdev entra no estado `Faulted` quando o número de dispositivos ausentes ou com falha excede o nível de redundância no vdev. Se os dispositivos ausentes puderem ser reconectados, o pool retornará ao estado Online. Se houver redundância insuficiente para compensar o número de discos com falha, o conteúdo do pool será perdido e deverá ser restaurado a partir de um backup.

Capítulo 20. Outros Sistemas de Arquivos

20.1. Sinopse

Os sistemas de arquivos são parte integrante de qualquer sistema operacional. Eles permitem que os usuários carreguem e armazenem arquivos, forneçem acesso a dados e tornam os discos rígidos úteis. Diferentes sistemas operacionais diferem em seu sistema de arquivos nativo. Tradicionalmente, o sistema de arquivo nativo do FreeBSD tem sido o Sistema de Arquivos Unix (Unix File System) UFS o qual foi modernizado como UFS2. Desde o FreeBSD 7.0, o Sistema de Arquivos Z (ZFS) também está disponível como um sistema de arquivos nativo. Veja O sistema de arquivos Z (ZFS) para maiores informações.

Além dos seus sistemas de arquivos nativos, o FreeBSD suporta uma infinidade de outros sistemas de arquivos para que dados de outros sistemas operacionais possam ser acessados localmente, tais como dados armazenados em dispositivos de armazenamento USB conectados localmente, drives flash e discos rígidos. Isto inclui suporte para o sistema de arquivos estendidos do Linux™ (EXT).

Existem diferentes níveis de suporte do FreeBSD para os vários sistemas de arquivos. Alguns exigem que um módulo do kernel seja carregado e outros podem requerer que um conjunto de ferramentas seja instalado. O suporte a alguns dos sistemas de arquivos não nativos é completo, suportando leitura/gravação, enquanto o suporte a outros é somente de leitura.

Depois de ler este capítulo, você saberá:

- A diferença entre sistemas de arquivos nativos e suportados.
- Quais sistemas de arquivos são suportados pelo FreeBSD.
- Como ativar, configurar, acessar e usar sistemas de arquivos não nativos.

Antes de ler este capítulo, você deve:

- Compreender o UNIX™ e ter noções básicas de FreeBSD.
- Estar familiarizado com o básico da configuração e compilação do kernel.
- Sinta-se confortável instalando software no FreeBSD.
- Tenha alguma familiaridade com discos, armazenamento e nomes de dispositivos no FreeBSD.

20.2. Sistemas de arquivos do Linux™

O FreeBSD fornece suporte built-in para vários sistemas de arquivos do Linux™. Esta seção demonstra como carregar o suporte e como montar os sistemas de arquivos suportados do Linux™.

20.2.1. ext2

O suporte no kernel para sistemas de arquivos ext2 está disponível desde o FreeBSD 2.2. No FreeBSD 8.x e anterior, o código está licenciado sob a GPL. Desde o FreeBSD 9.0, o código foi reescrito e agora é licenciado sob a licença BSD.

O driver ext2fs(5) permite que o kernel do FreeBSD leia e grave em sistemas de arquivos ext2.



Esse driver também pode ser usado para acessar os sistemas de arquivos ext3 e ext4. O sistema de arquivos ext2fs(5) possui suporte completo para leitura e gravação para o ext4 a partir do FreeBSD 12.0-RELEASE. Além disso, os atributos estendidos e as ACLs também são suportados, enquanto o journalling e a criptografia não são. Começando com o FreeBSD 12.1-RELEASE, um provedor do DTrace também estará disponível. Versões anteriores do FreeBSD podem acessar o ext4 no modo de leitura e gravação usando sysutils/fusefs-ext2.

Para acessar um sistema de arquivos ext, primeiro carregue o módulo correspondente do kernel:

```
# kldload ext2fs
```

Em seguida, monte o volume ext especificando seu nome de partição no FreeBSD e um ponto de montagem existente. Este exemplo monta /dev/ad1s1 em /mnt:

```
# mount -t ext2fs /dev/ad1s1 /mnt
```

Capítulo 21. Virtualização

21.1. Sinopse

O software de virtualização permite que vários sistemas operacionais sejam executados simultaneamente no mesmo computador. Tais sistemas de software para PCs geralmente envolvem um sistema operacional host que executa o software de virtualização e suporta qualquer número de sistemas operacionais convidados.

Depois de ler este capítulo, você saberá:

- A diferença entre um sistema operacional host e um sistema operacional convidado.
- Como instalar o FreeBSD em um computador baseado em um Intel™Apple™Mac™.
- Como instalar o FreeBSD no Microsoft™Windows™ com Virtual PC.
- Como instalar o FreeBSD como um host convidado no bhyve.
- Como ajustar um sistema FreeBSD para melhor desempenho sob virtualização.

Antes de ler este capítulo, você deve:

- Entender o básico sobre sistemas UNIX™ e sobre o FreeBSD.
- Saber como instalar o FreeBSD.
- Saber como configurar uma conexão de rede.
- Saber como instalar software adicional de terceiros.

21.2. FreeBSD como Sistema Operacional Convidado no Parallels para Mac OS™ X

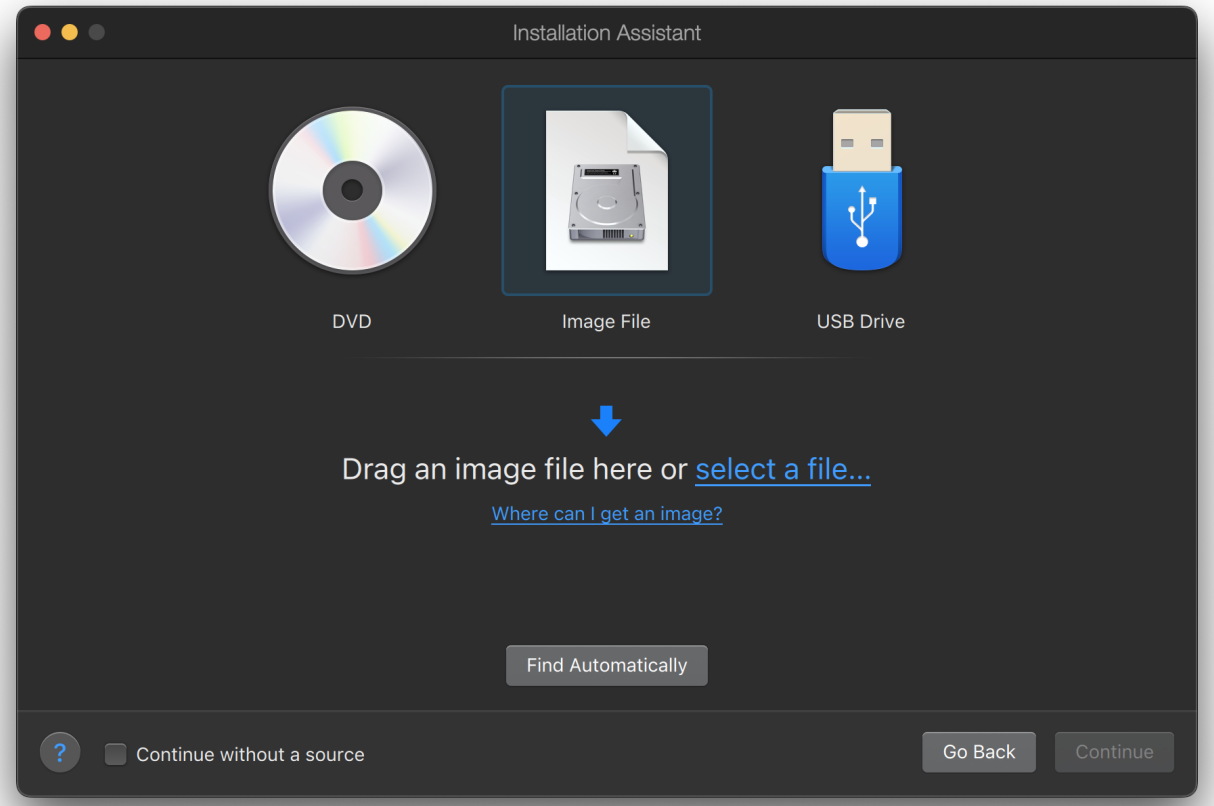
O Parallels Desktop para Mac™ é um produto de software comercial disponível para computadores baseados em Intel™Apple™Mac™ rodando Mac OS™ 10.4.6 ou superior. O FreeBSD é um sistema operacional convidado completamente suportado. Uma vez que o Parallels tiver sido instalado no Mac OS™ X, o usuário deve configurar uma maquina virtual e então instalar o sistema operacional convidado desejado.

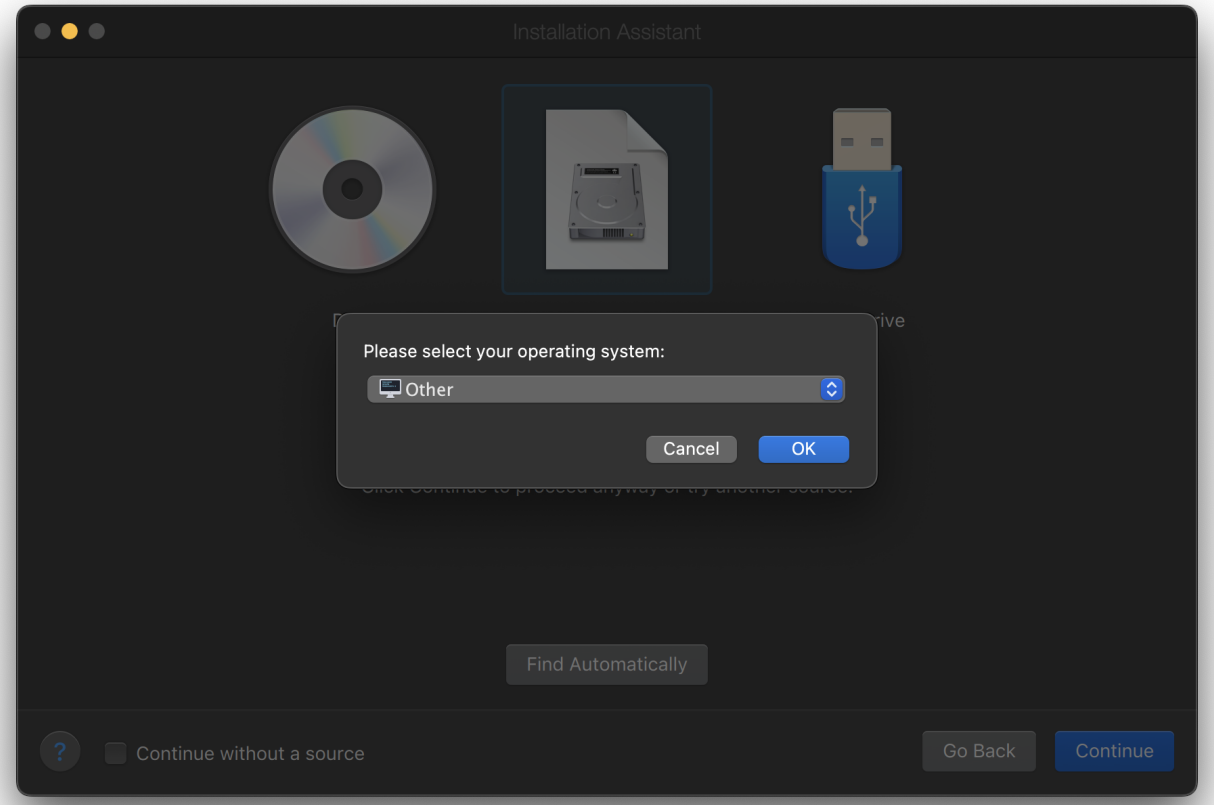
21.2.1. Instalando o FreeBSD no Parallels/Mac OS™ X

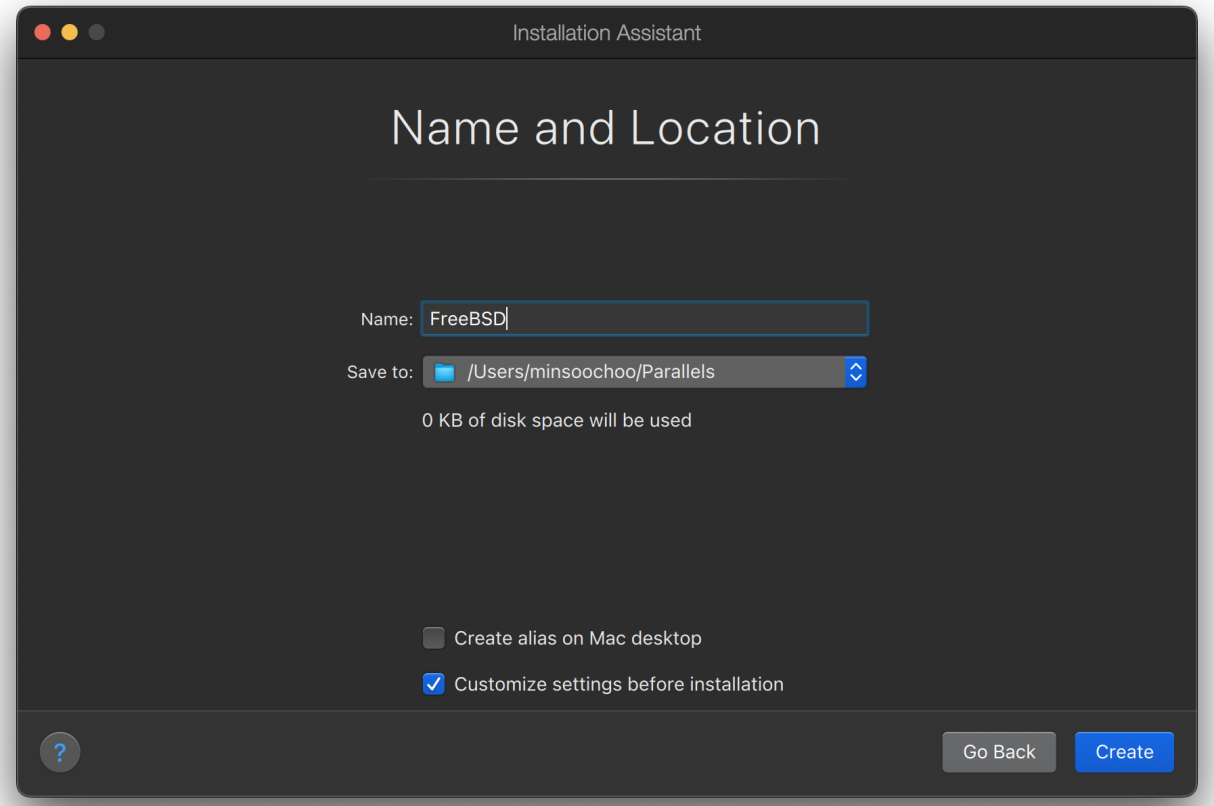
O primeiro passo para instalar o FreeBSD no Parallels é criar uma nova máquina virtual para instalar o FreeBSD. Selecione FreeBSD como o **Guest OS Type** quando solicitado:

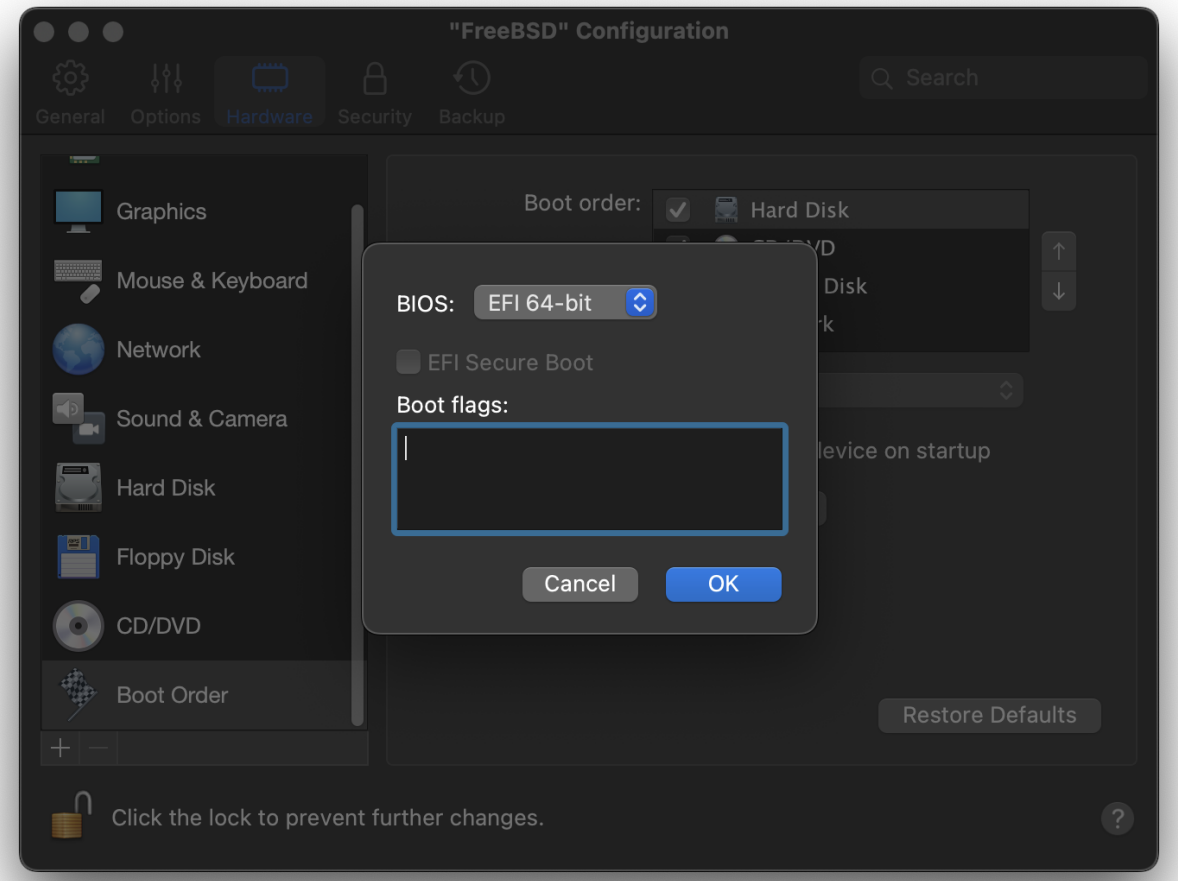


Escolha uma quantidade razoável de disco e memória, dependendo dos planos para esta instância virtual do FreeBSD. 4GB de espaço em disco e 512MB de RAM funcionam bem para a maioria dos usos do FreeBSD executando sob o Parallels:

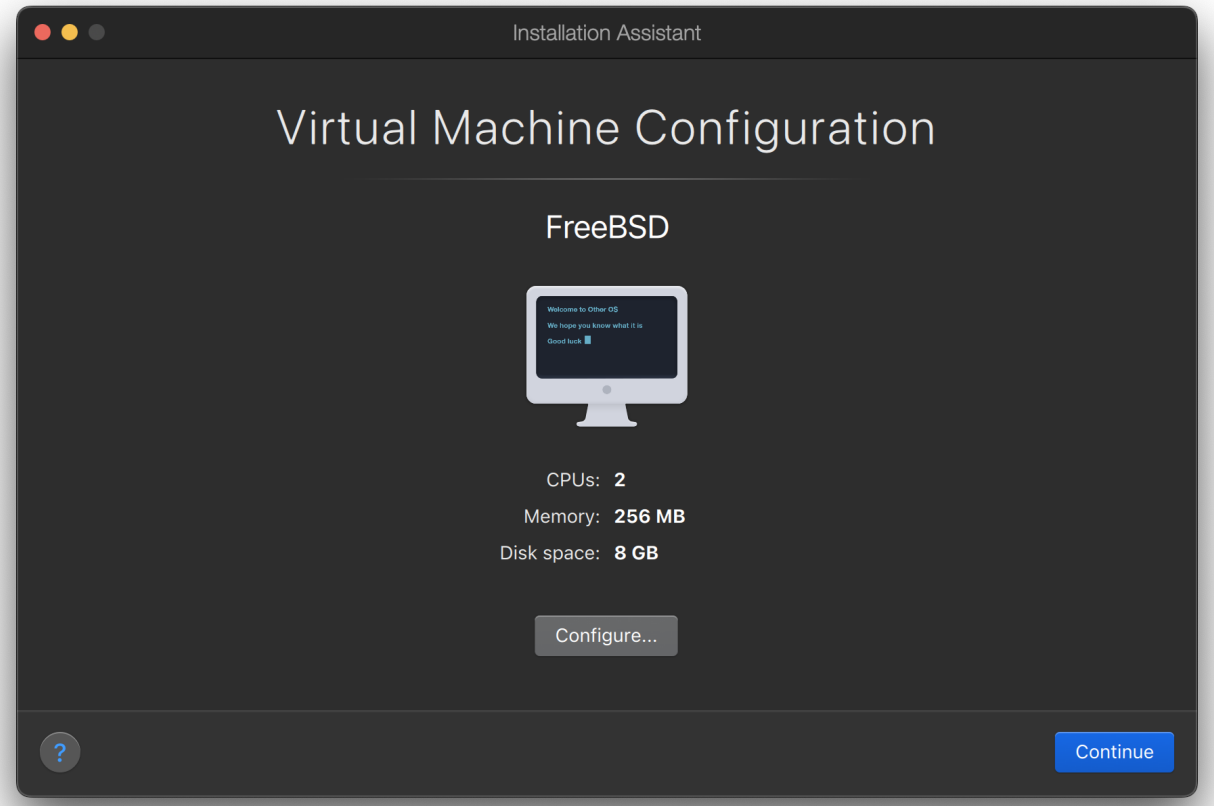

Installation Assistant
DVD
Image File
USB Drive
Drag an image file here or select a file...
Where can I get an image?
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Please select your operating system:
Other
Cancel
OK
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Name and Location
Name:
FreeBSD
Save to:
/Users/minsoochoo/Parallels
0 KB of disk space will be used
Create alias on Mac desktop
Customize settings before installation
Go Back
Create

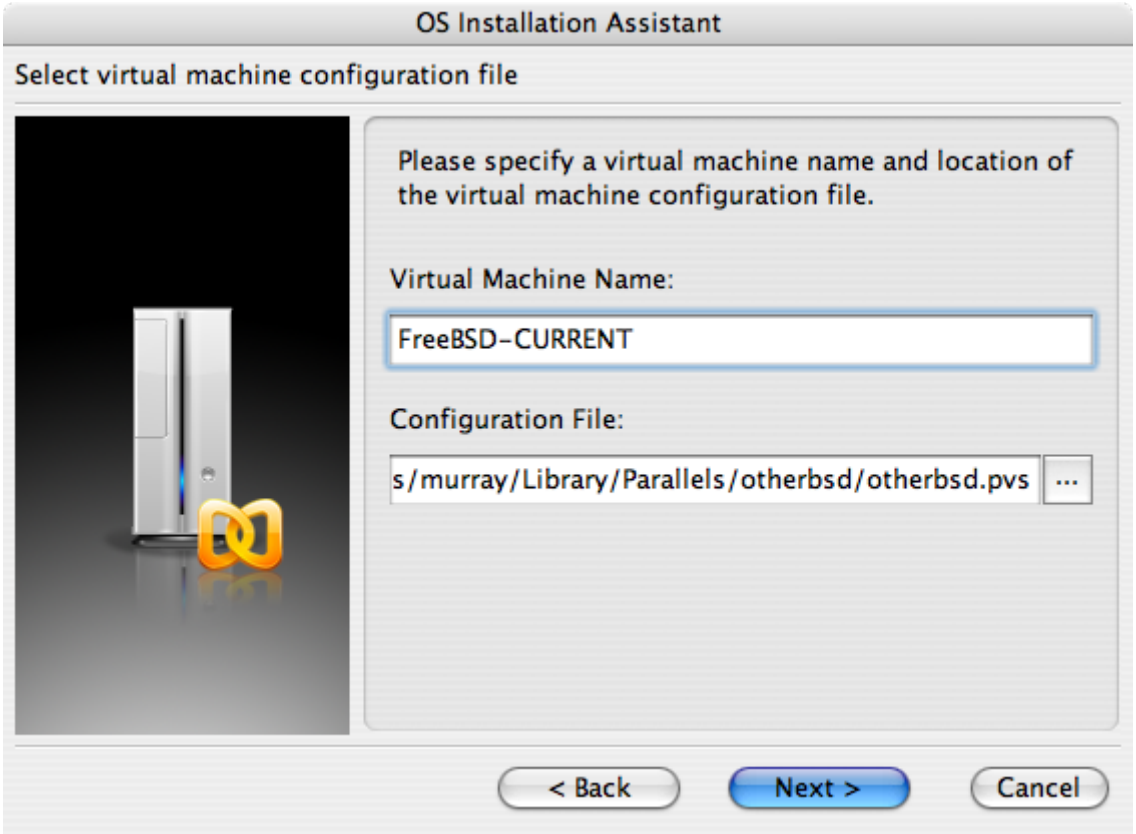


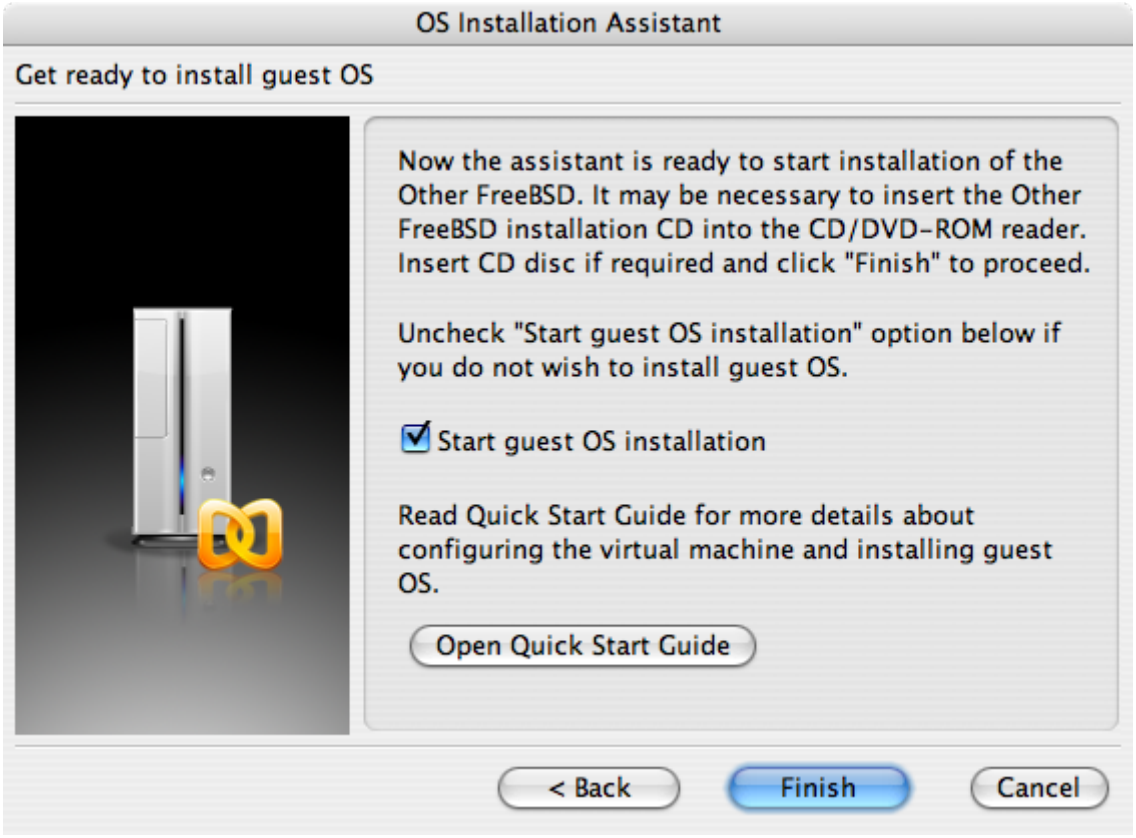
Selecione o tipo de rede e uma interface de rede:


Installation Assistant
Virtual Machine Configuration
FreeBSD
Welcome to Other OS
We hope you know what it is
Good luck
CPUs: 2
Memory: 256 MB
Disk space: 8 GB
Configure...
?
Continue

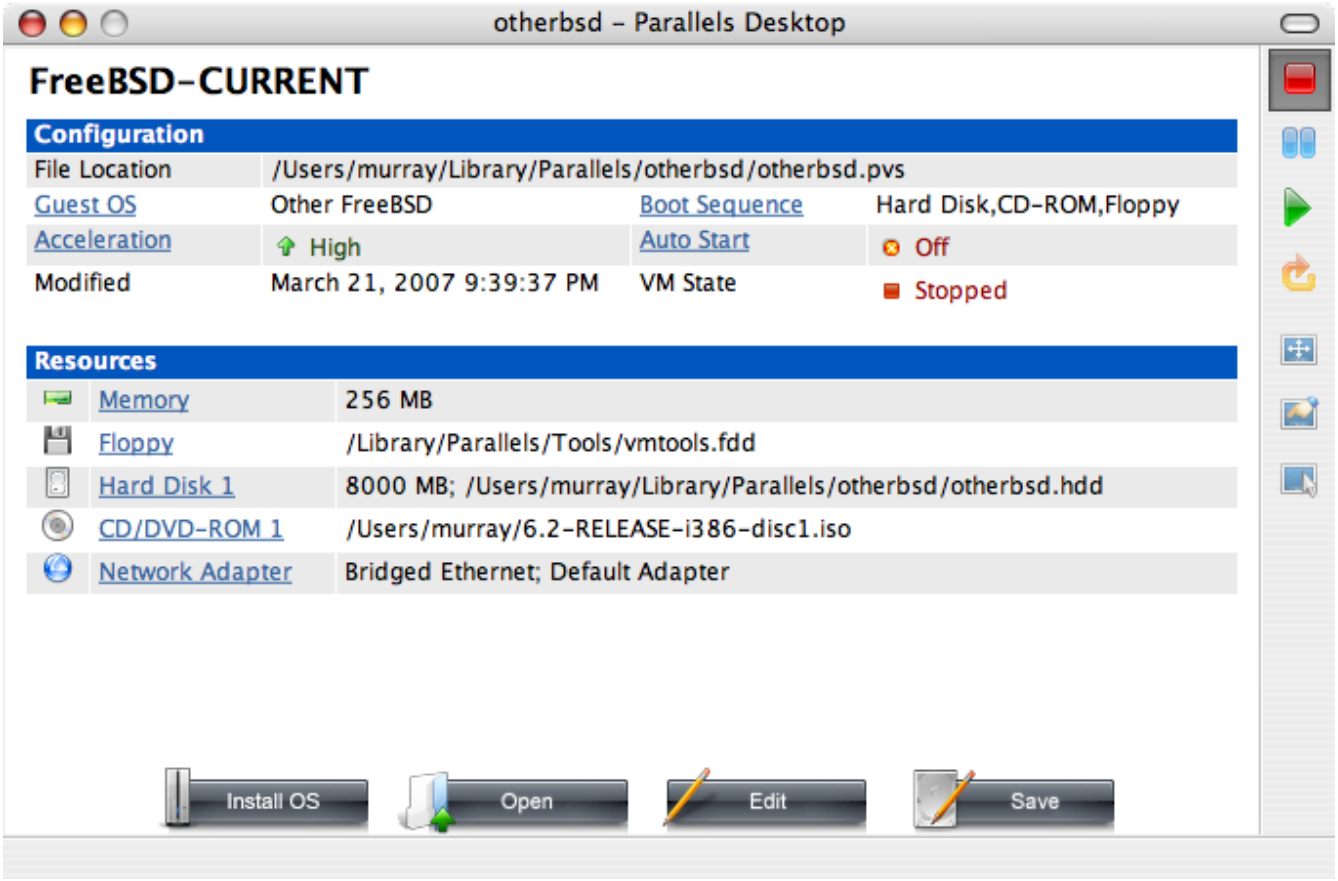


Salve e finalize a configuração:

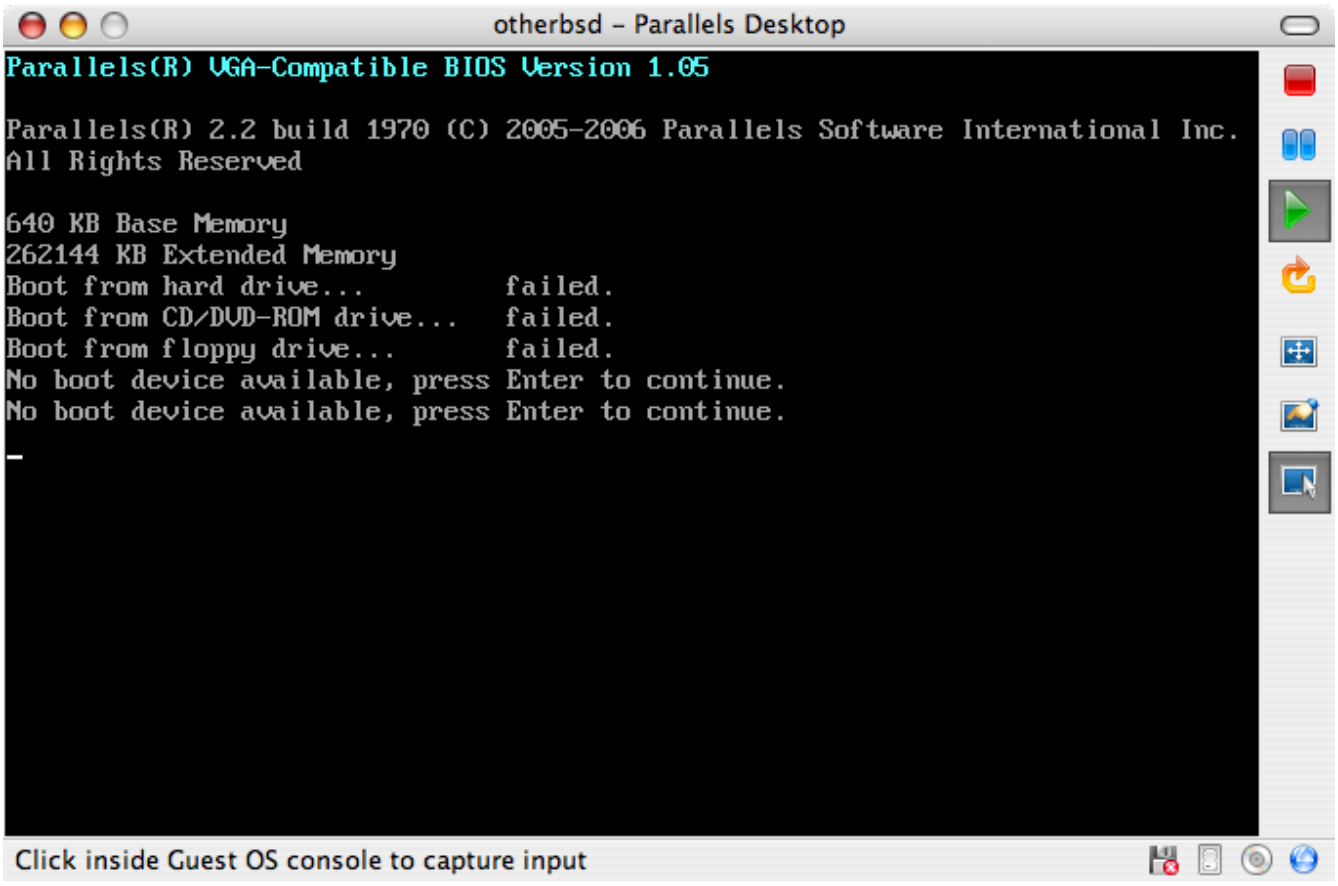




Após a criação da máquina virtual do FreeBSD, o FreeBSD pode ser instalado nela. Isto é feito melhor com um CD/DVD oficial do FreeBSD ou com uma imagem ISO baixada de um site FTP oficial. Copie a imagem ISO apropriada para o sistema de arquivos local do Mac™ ou insira um CD/DVD na unidade de CD-ROM do Mac™. Clique no ícone do disco no canto inferior direito da janela do FreeBSD no Parallels. Isso abrirá uma janela a qual pode ser usada para associar a unidade de CD-ROM na máquina virtual com o arquivo ISO no disco ou com drive CD.

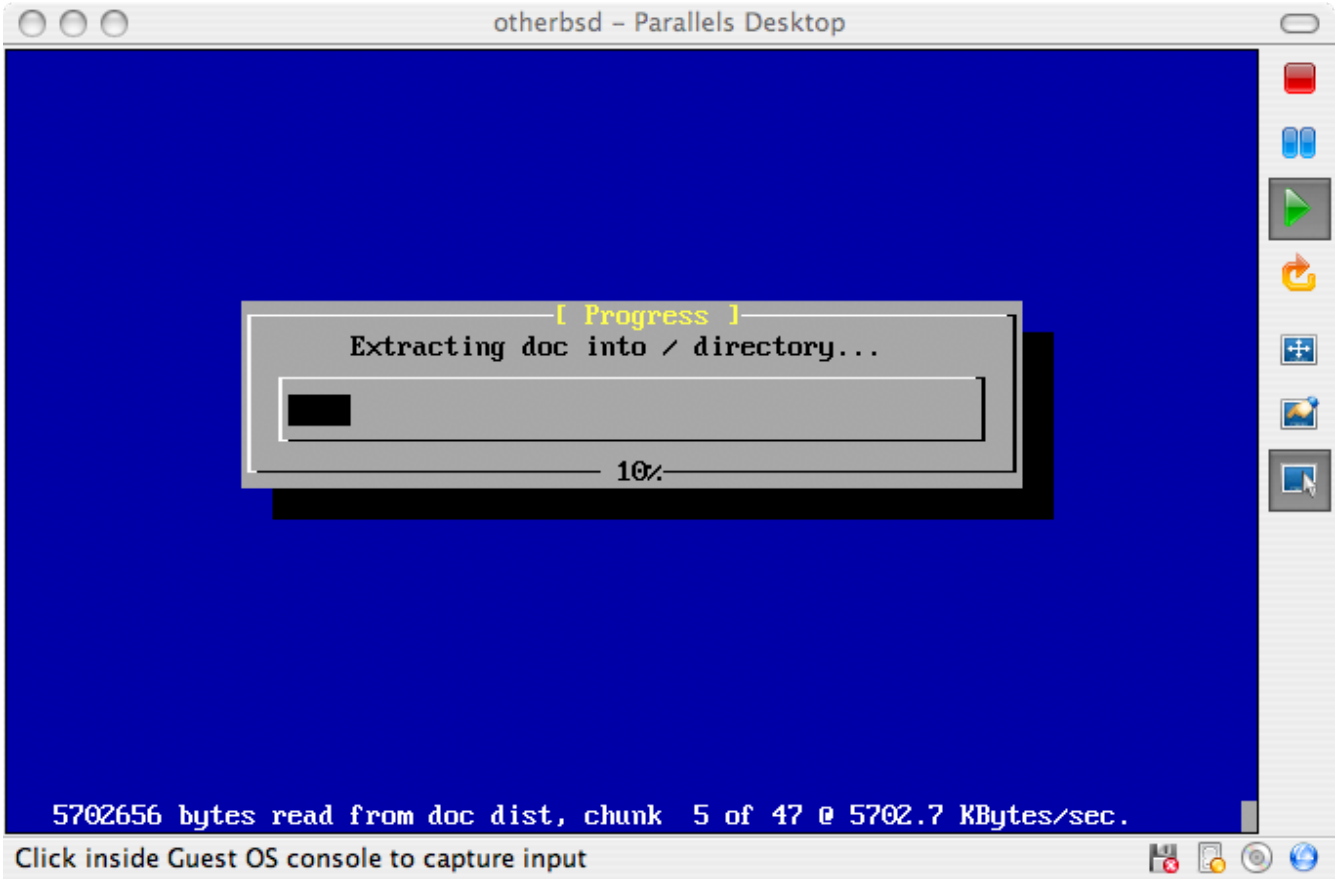


Uma vez que esta associação com a fonte do CD-ROM estiver feita, reinicialize a máquina virtual do FreeBSD clicando no ícone de reinicialização. O Parallels irá reiniciar com um BIOS especial o qual primeiro irá verificar se existe um CD-ROM.

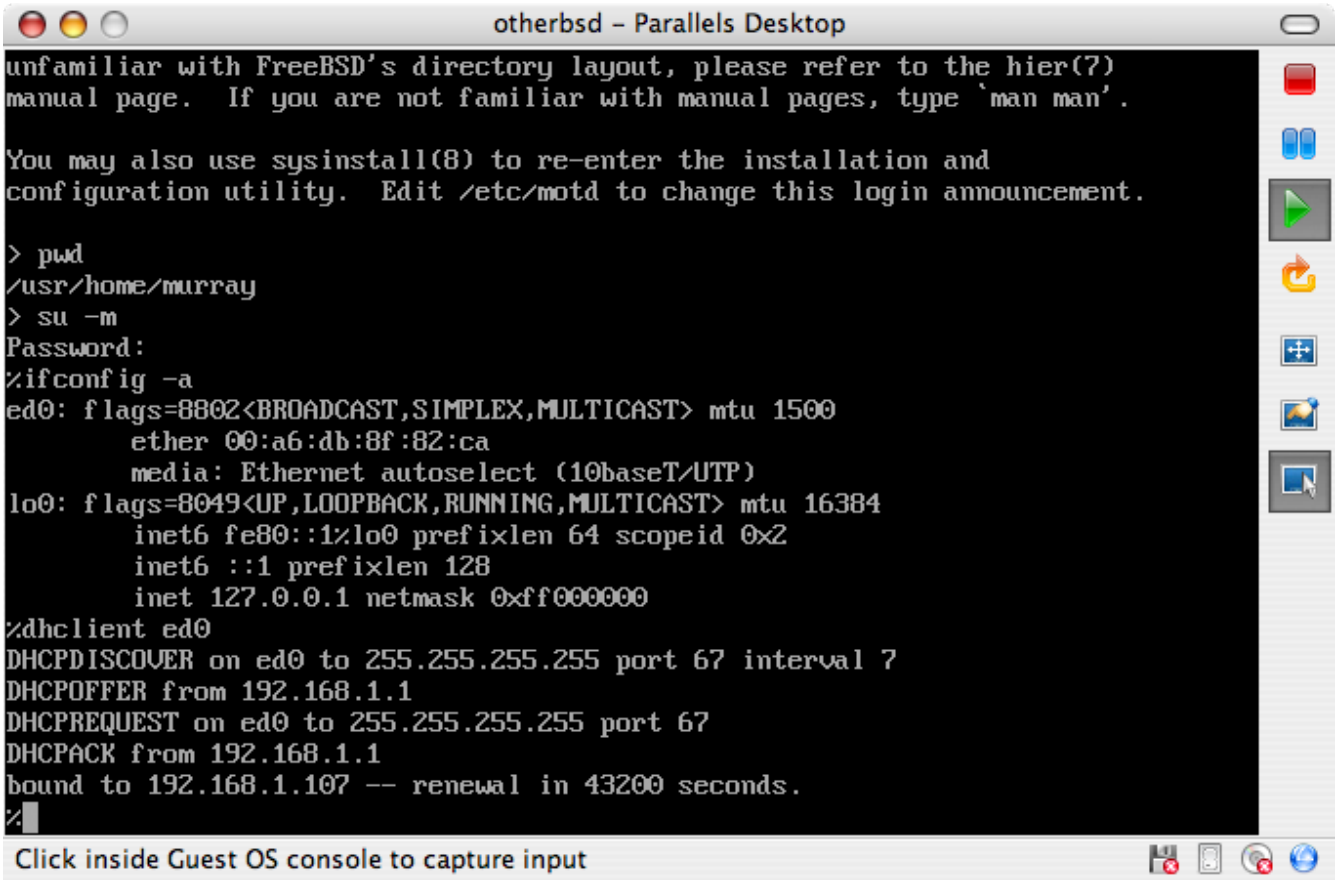


Neste caso, ele encontrará a mídia de instalação do FreeBSD e iniciará uma instalação normal do

FreeBSD. Execute a instalação, mas não tente configurar o Xorg neste momento.



Quando a instalação estiver concluída, reinicie a máquina virtual FreeBSD recém-instalada.



21.2.2. Configurando o FreeBSD no Parallels

Depois que o FreeBSD foi instalado com sucesso no Mac OS™ X com o Parallels , existem várias etapas de configuração que podem ser executadas para otimizar o sistema para operar virtualizado.

1. Definir variáveis do Boot Loader

   O passo mais importante é reduzir o `kern.hz` ajustável para reduzir a utilização de CPU no FreeBSD sob ambiente Parallels. Isso é feito adicionando a seguinte linha ao /boot/loader.conf:

   ```
   kern.hz=100
   ```

   Sem essa configuração, um sistema convidado inativo do FreeBSD no Parallels usará aproximadamente 15% da CPU de um único processador iMac™. Após essa alteração, o uso ficará mais próximo de 5%.

2. Criar um novo arquivo de configuração do kernel

   Todos os drivers de dispositivos SCSI, FireWire e USB podem ser removidos de um arquivo de configuração de kernel personalizado. O Parallels fornece um adaptador de rede virtual usado pelo driver ed(4), portanto, todos os dispositivos de rede, exceto o ed(4) e o miibus(4) podem ser removidos do kernel .

3. Configure a rede

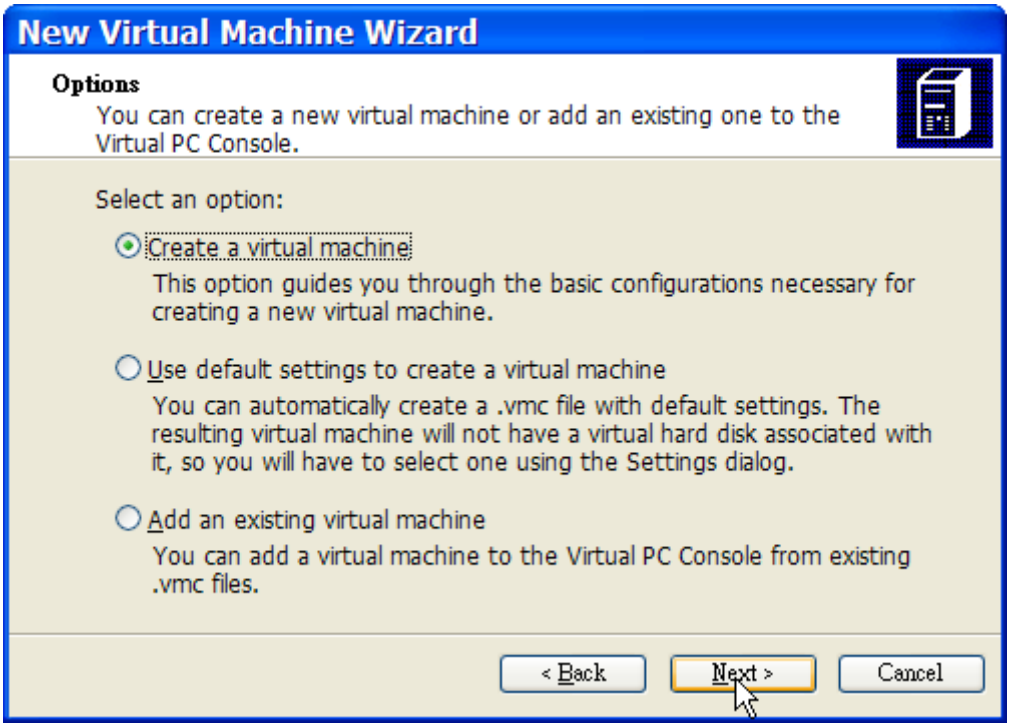
   A configuração de rede mais básica usa o DHCP para conectar a máquina virtual à mesma rede local que o host Mac™. Isso pode ser feito adicionando `ifconfig_ed0="DHCP"` ao /etc/rc.conf. Configurações de rede mais avançadas são descritas em Rede Avançada.

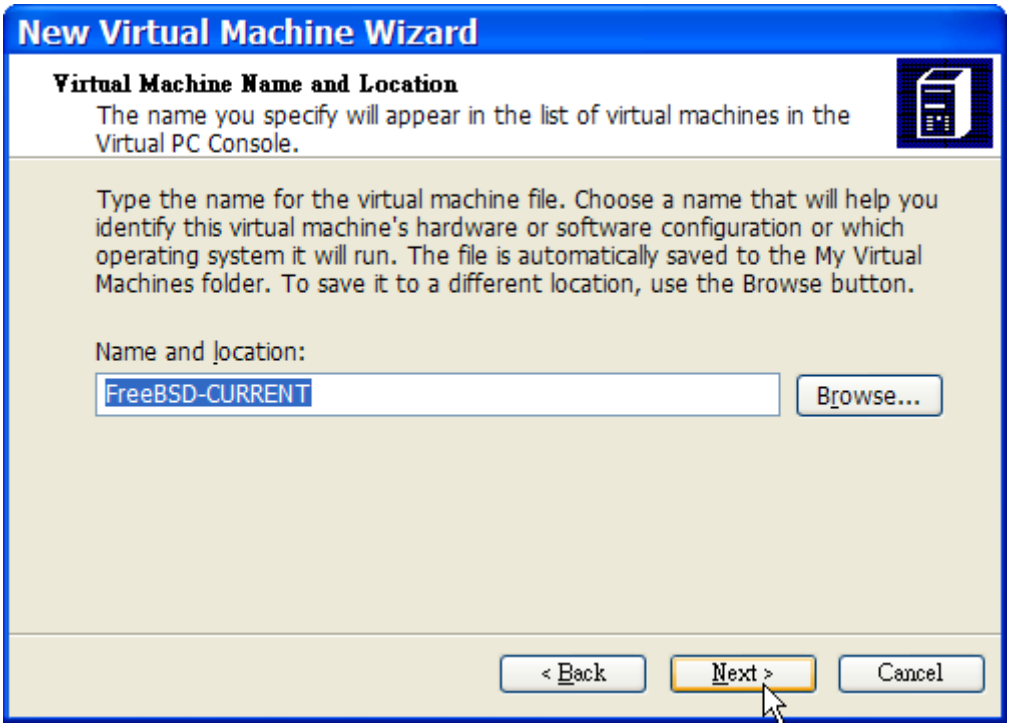
21.3. FreeBSD como sistema convidado no Virtual PC para Windows™

O Virtual PC para Windows™ é um software da Microsoft™ disponível para download gratuito. Consulte este site para os requisitos do sistema. Depois que o Virtual PC tiver sido instalado no Microsoft™Windows™, o usuário poderá configurar uma máquina virtual e depois instalar o sistema operacional convidado desejado.

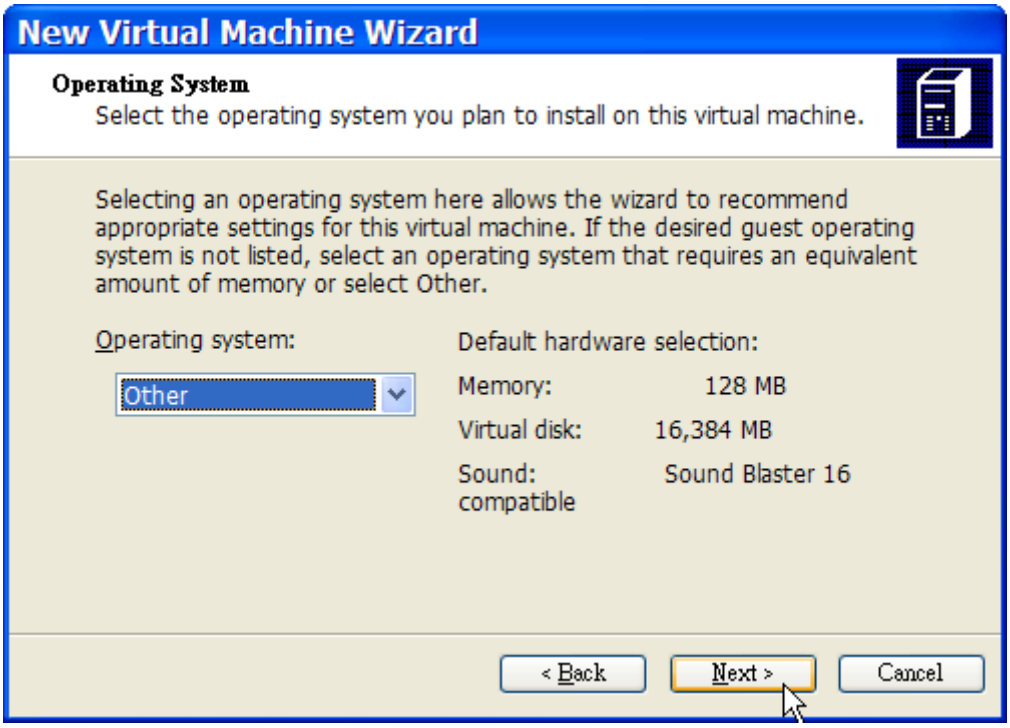
21.3.1. Instalando o FreeBSD no Virtual PC

O primeiro passo para instalar o FreeBSD no Virtual PC é criar uma nova máquina virtual para instalar o FreeBSD. Selecione Criar uma máquina virtual quando solicitado:

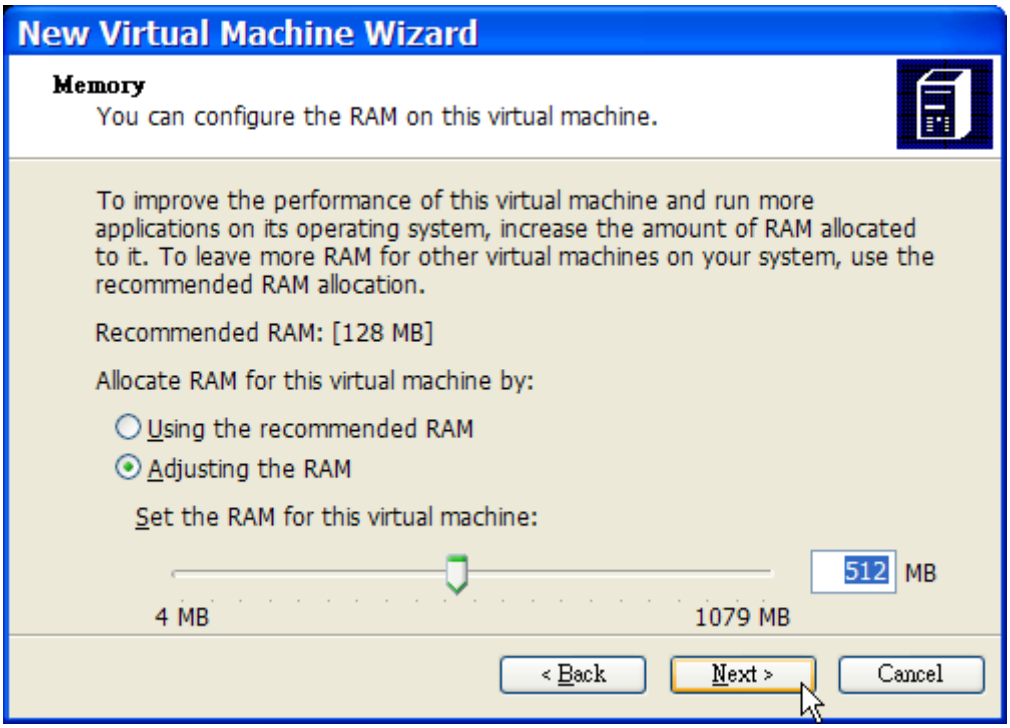


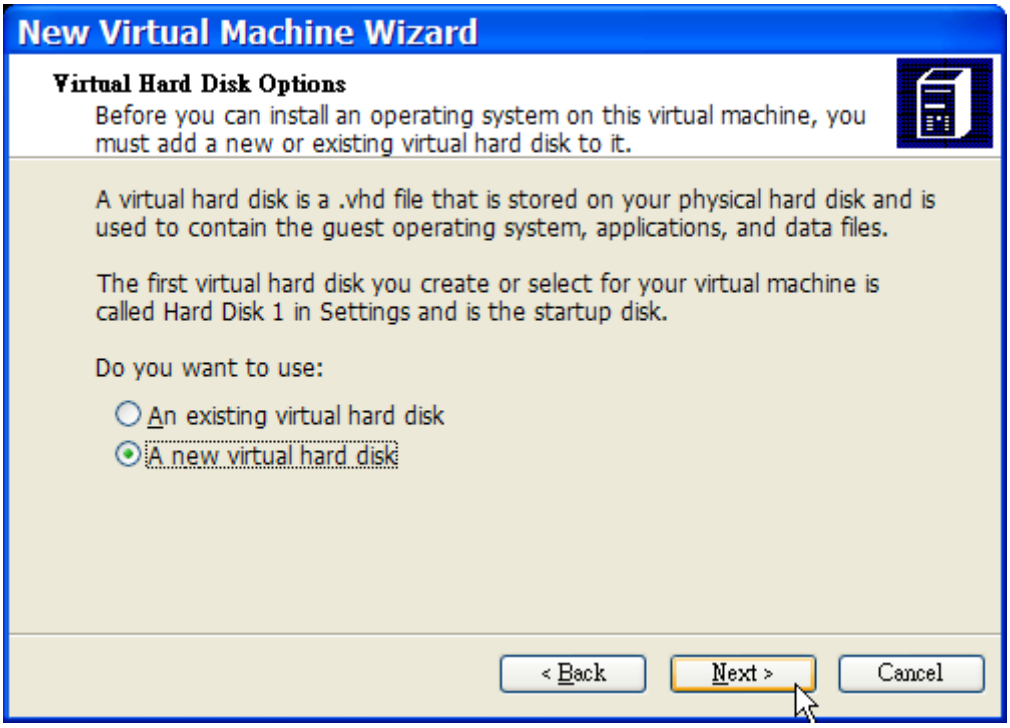


Selecione a opção Outro para o Sistema operacional quando solicitado:

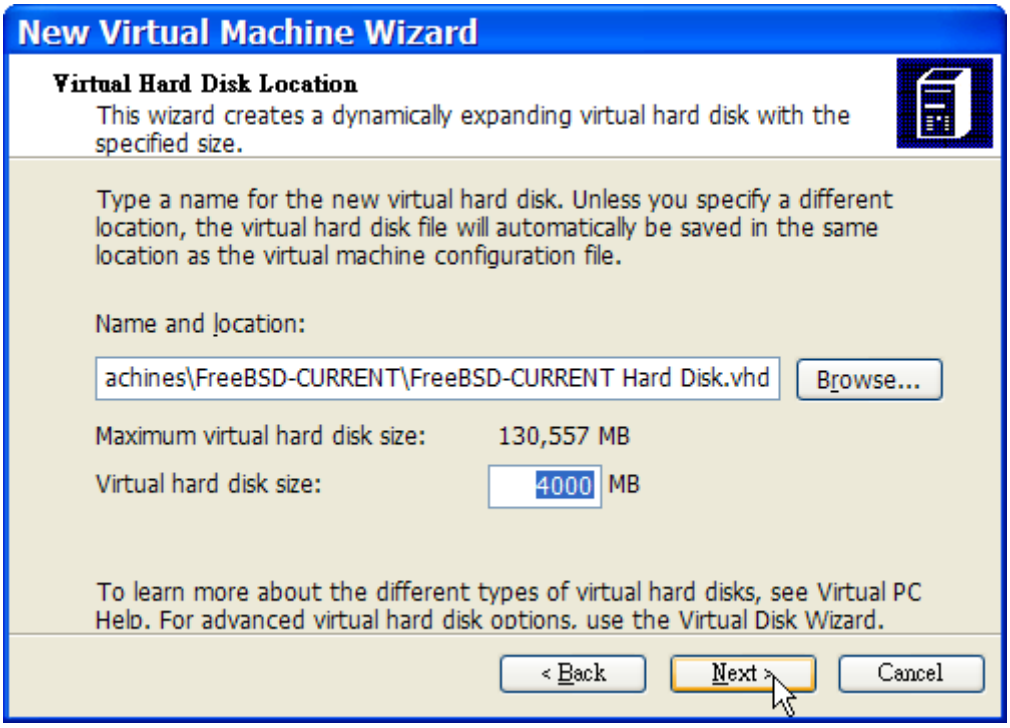


Em seguida, escolha uma quantidade razoável de disco e de memória, dependendo dos planos para esta instância virtual do FreeBSD. 4GB de espaço em disco e 512MB de RAM funcionam bem para a maioria dos usos do FreeBSD sob o Virtual PC:

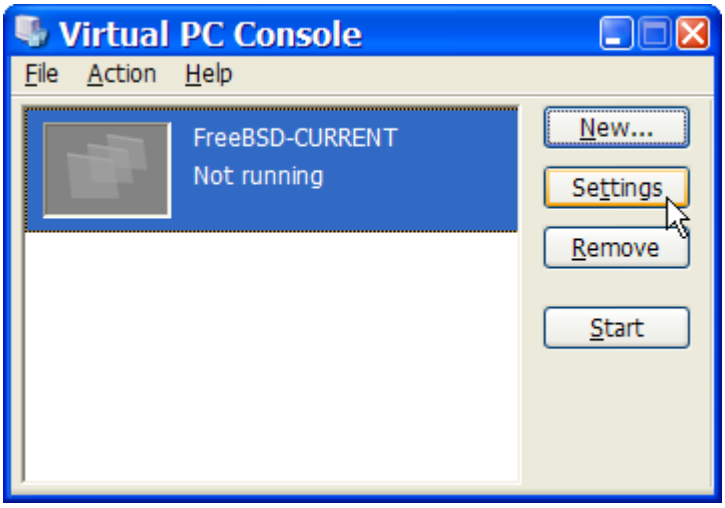


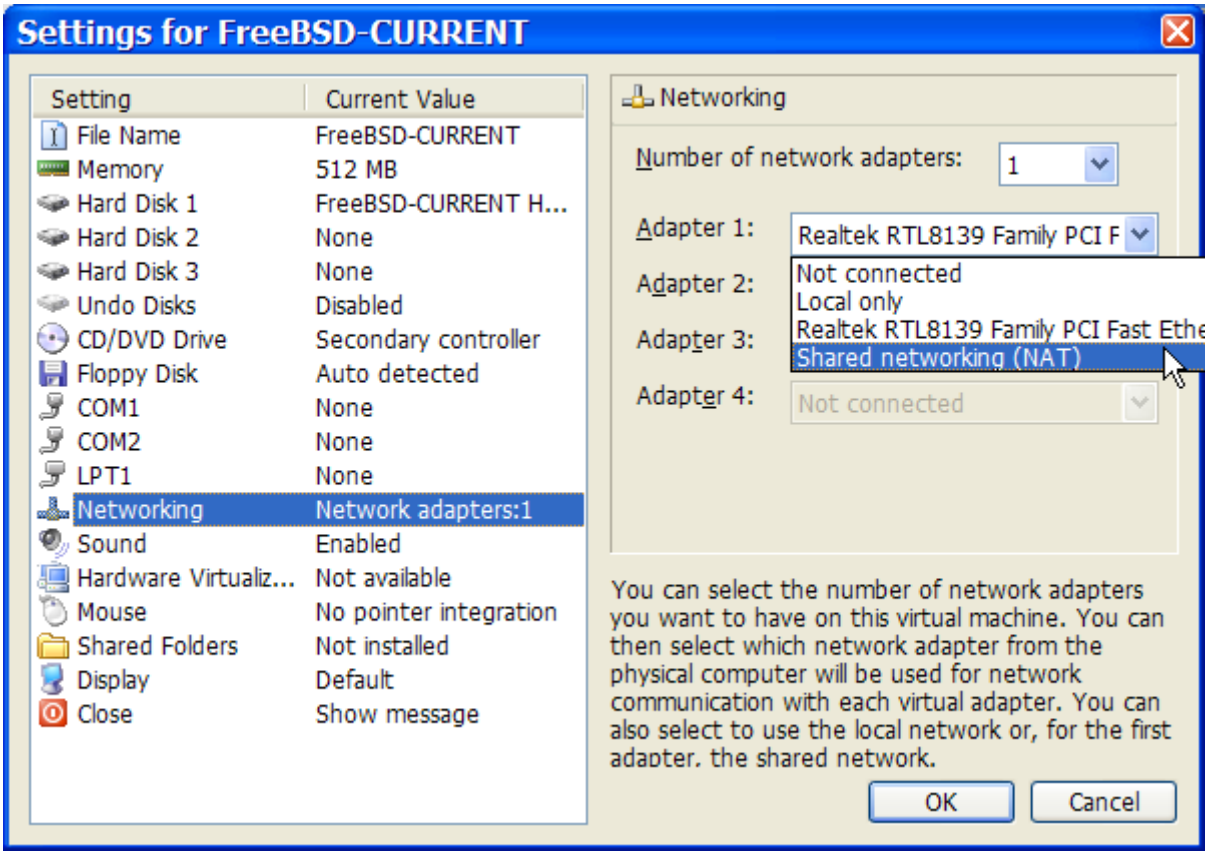


Salve e finalize a configuração:

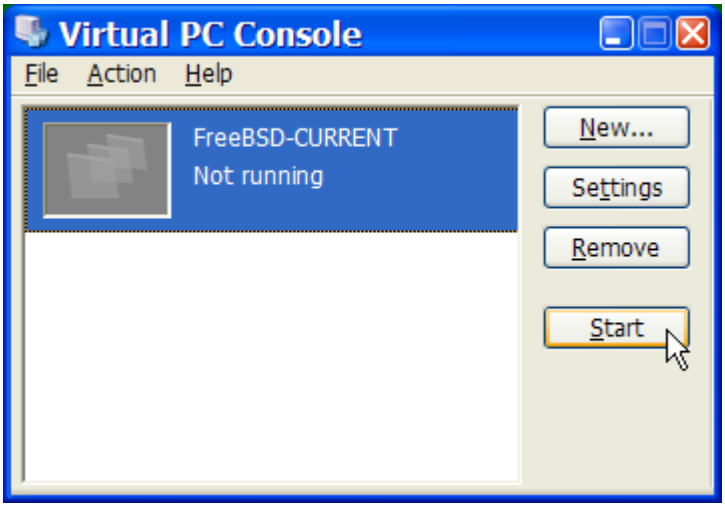


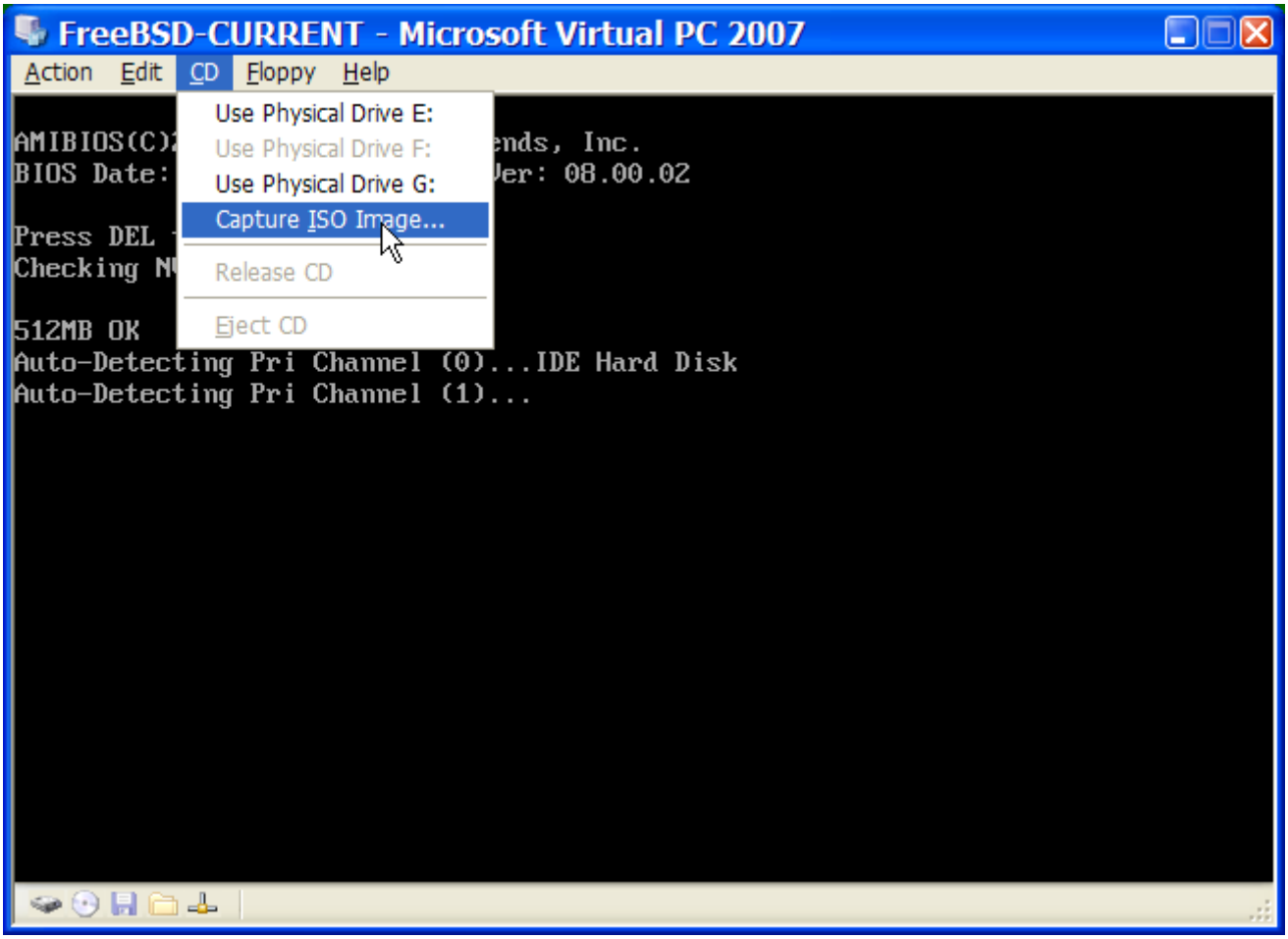
Selecione a máquina virtual do FreeBSD e clique em **Configurações**, em seguida, defina o tipo de rede e uma interface de rede:



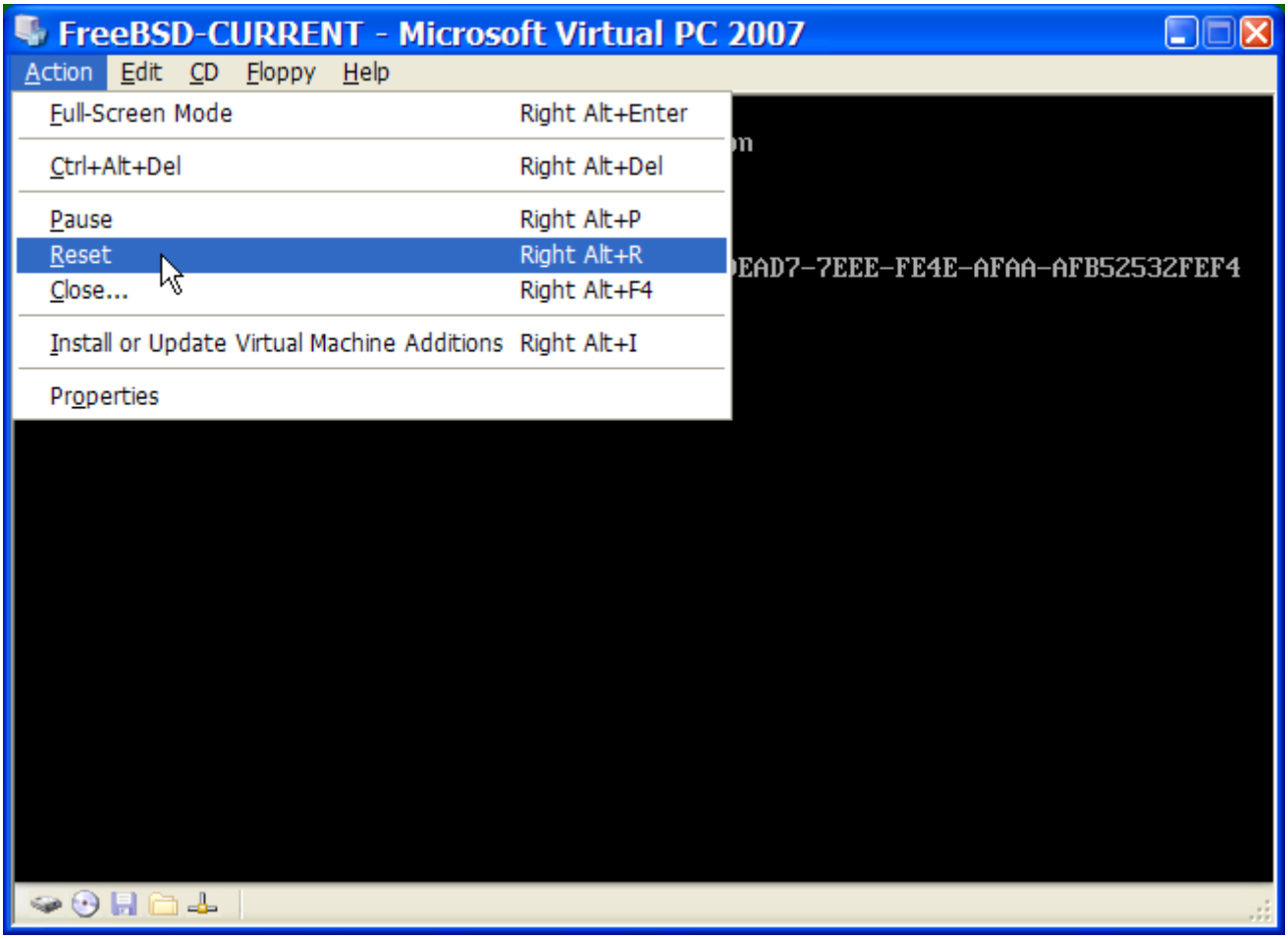


Após a criação da máquina virtual do FreeBSD, o FreeBSD pode ser instalado nela. Isso é feito da melhor maneira com um CD/DVD oficial ou com uma imagem ISO baixada de um site FTP oficial. Copie a imagem ISO apropriada para o sistema de arquivos local do Windows™ ou insira um CD/DVD na unidade de CD-ROM, então clique duas vezes na máquina virtual FreeBSD para inicializar. Em seguida, clique em **CD** e escolha **Capturar imagem ISO...** na janela do Virtual PC. Isso abrirá uma janela na qual a unidade de CD-ROM na máquina virtual poderá ser associada a um arquivo ISO no disco ou com o drive de CD-ROM real.

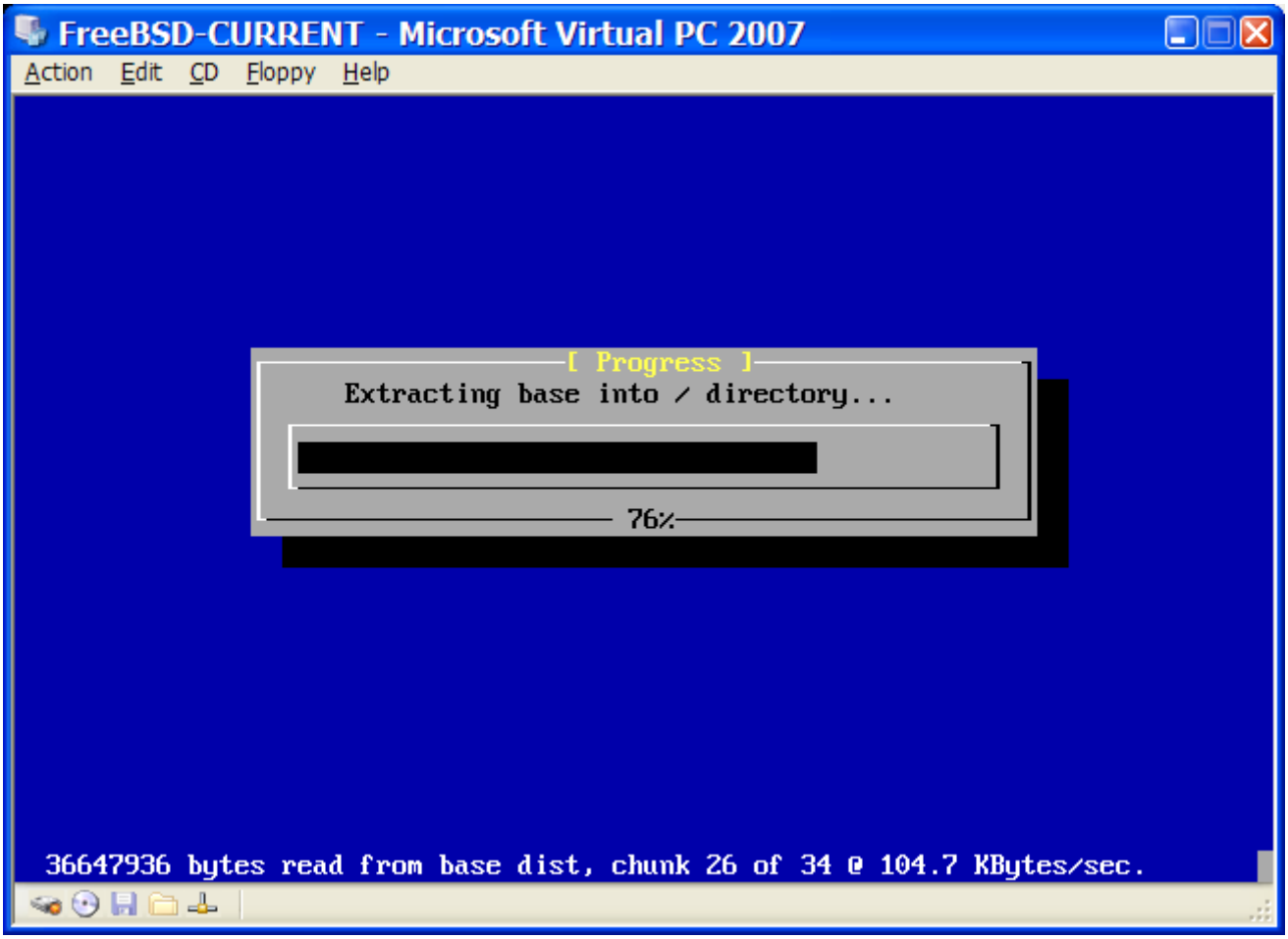




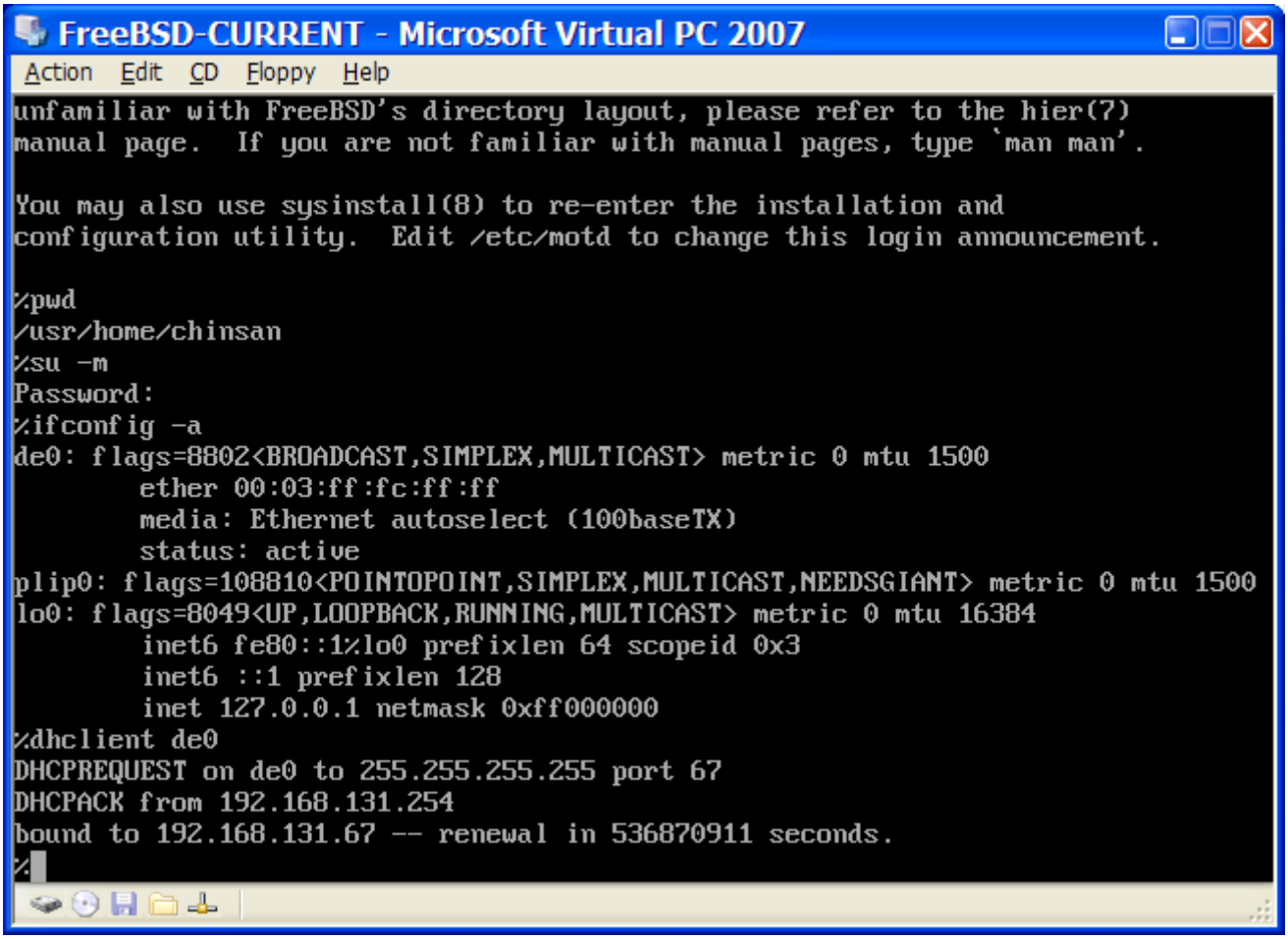
Uma vez que a associação com a fonte do CD-ROM estiver feita, reinicie a máquina virtual do FreeBSD clicando em **Action** e depois em **Reset**. O Virtual PC será reiniciado com um BIOS especial que irá procurar por um CD-ROM para inicializar.



Neste caso, ele encontrará a mídia de instalação do FreeBSD e iniciará uma instalação normal do FreeBSD. Continue com a instalação, mas não tente configurar o Xorg neste momento.



Quando a instalação estiver concluída, lembre-se de ejetar o CD/DVD ou de liberar a imagem ISO. Finalmente, reinicie a máquina virtual FreeBSD recém-instalada.



21.3.2. Configuring FreeBSD on Virtual PC

Depois que o FreeBSD tiver sido instalado com sucesso no Microsoft™Windows™ com o Virtual PC, existem várias etapas de configurações que podem ser executadas para otimizar o sistema para operação virtualizada.

1. Definir variáveis do Boot Loader

   O passo mais importante é reduzir o valor do parâmetro `kern.hz` para reduzir a utilização da CPU do FreeBSD sob o ambiente do Virtual PC. Isso é feito adicionando a seguinte linha ao /boot/loader.conf:

   ```
   kern.hz=100
   ```

   Sem esta configuração, uma VM idle do FreeBSD rodando sob o Virtual PC utilizará aproximadamente 40% da CPU de um computador com um único processador. Após essa mudança, o uso ficará mais próximo de 3%.

2. Criar um novo arquivo de configuração do kernel

   Todos os drivers de dispositivos SCSI, FireWire e USB podem ser removidos do arquivo de configuração do kernel personalizado. O Virtual PC fornece um adaptador de rede virtual usado pelo driver de(4), portanto, todos os dispositivos de rede, exceto o de(4) e o miibus(4) podem ser removidos do kernel.

3. Configure a rede

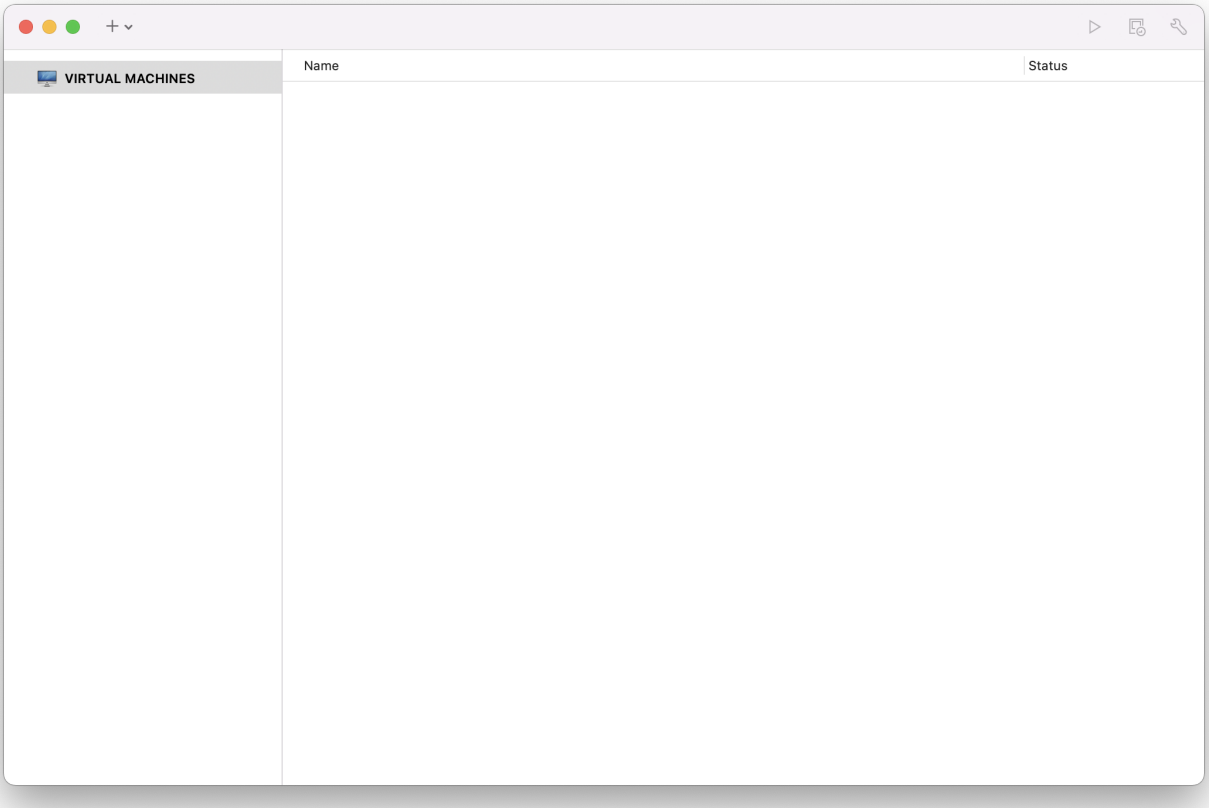
   A configuração de rede mais básica usa o DHCP para conectar a máquina virtual à mesma rede local que o host Microsoft™Windows™. Isso pode ser feito adicionando `ifconfig_de0="DHCP"` ao /etc/rc.conf. Configurações de rede mais avançadas são descritas em Rede Avançada.

21.4. FreeBSD como Sistema Operacional Convidado no VMware Fusion para Mac OS™

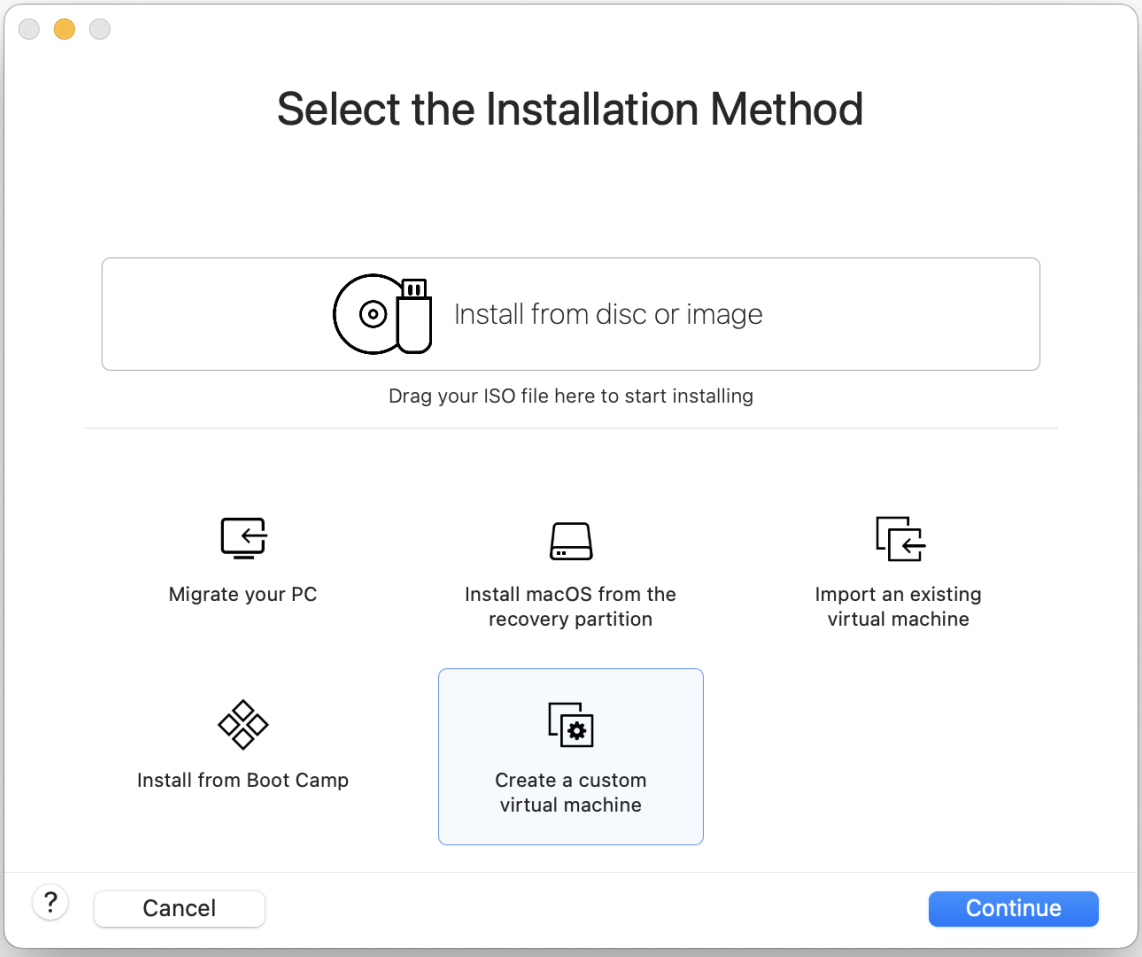
O VMware Fusion para Mac™ é um software comercial disponível para computadores Apple™Mac™ baseados em processadores Intel™ e que rodam o Mac OS™ 10.4.9 ou superior. O FreeBSD é um sistema operacional convidado totalmente suportado. Depois que o VMware Fusion for instalado no Mac OS™ X, o usuário poderá configurar uma máquina virtual e, em seguida, instalar o sistema operacional convidado desejado.

21.4.1. Instalando o FreeBSD no VMware Fusion

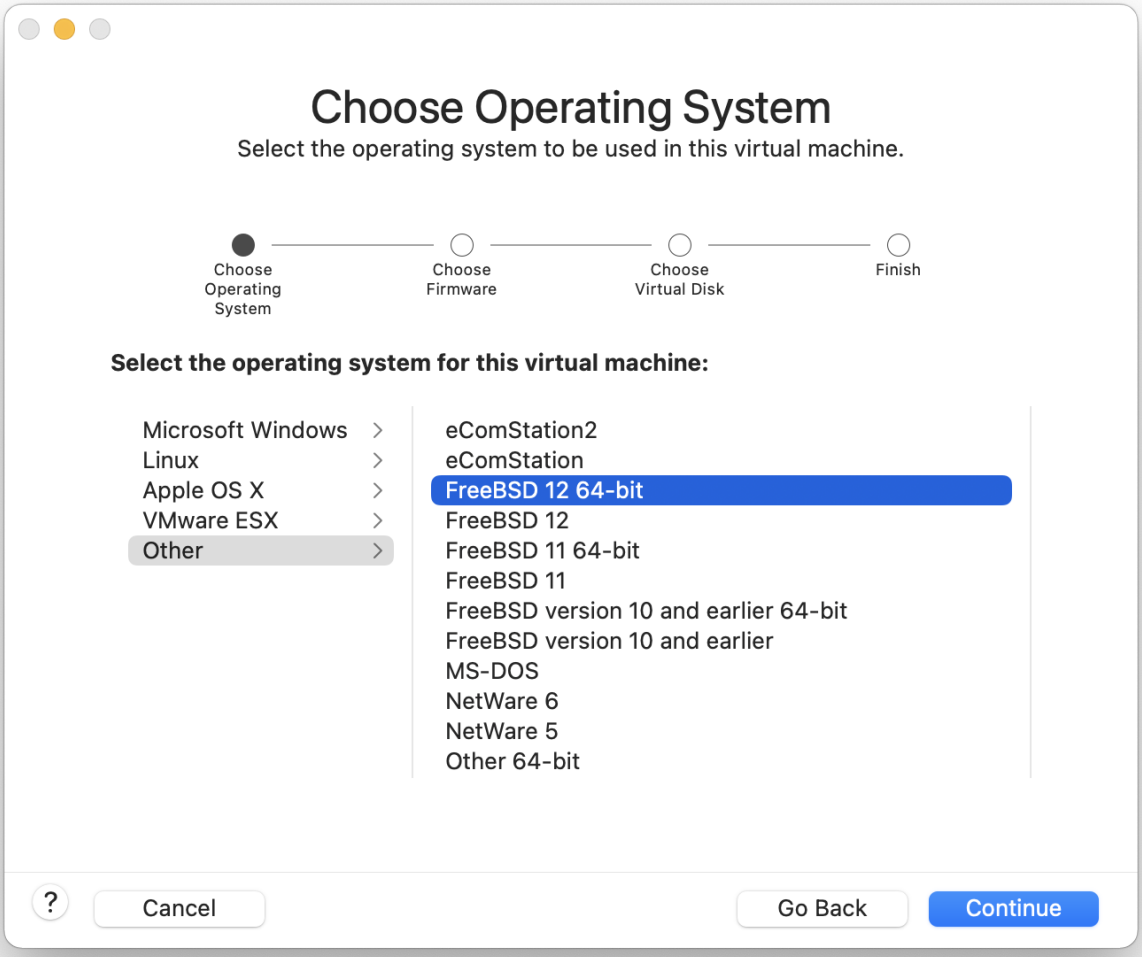
A primeira etapa é iniciar o VMware Fusion, que irá carregar a biblioteca de máquinas virtuais. Clique em Novo para criar a máquina virtual:



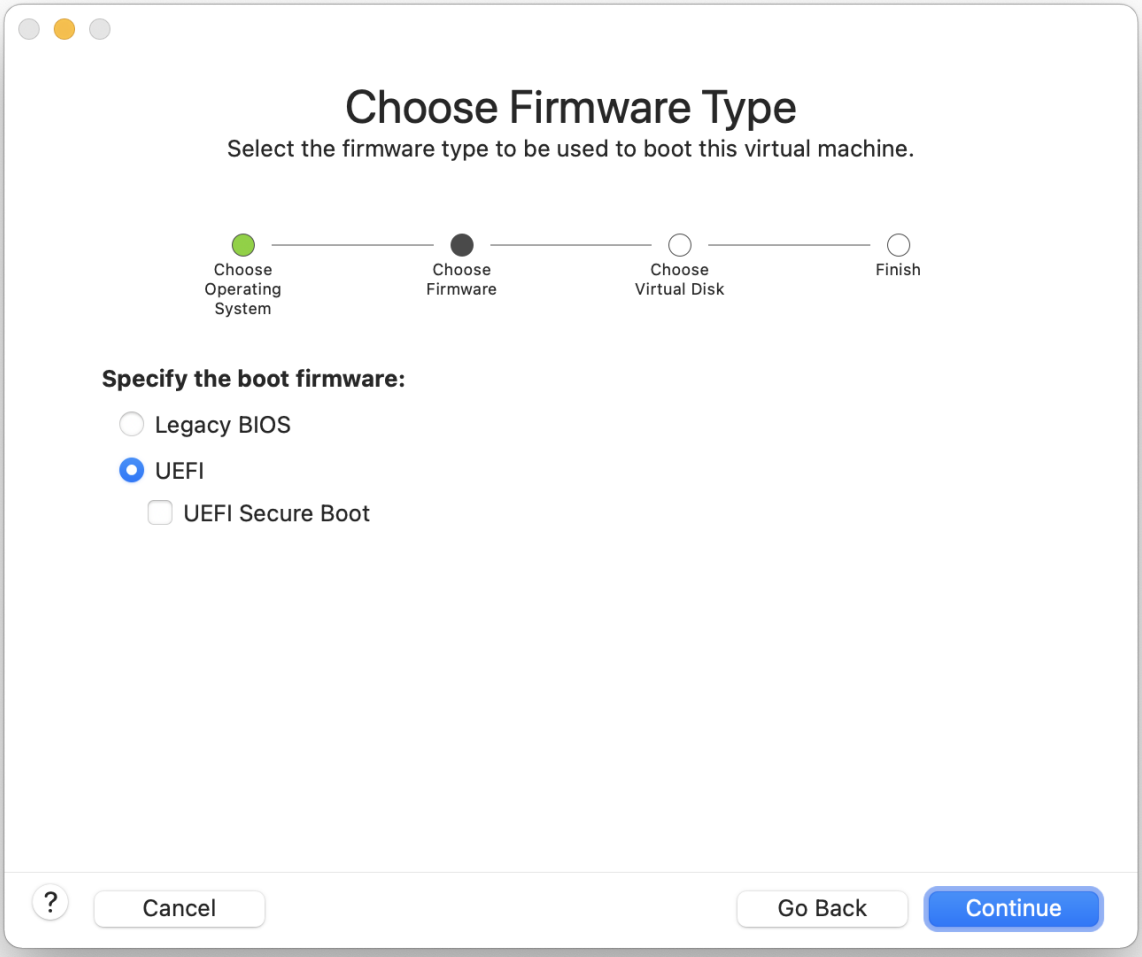
Isto irá carregar o Assistente de Nova Máquina Virtual. Clique em Continuar para prosseguir:



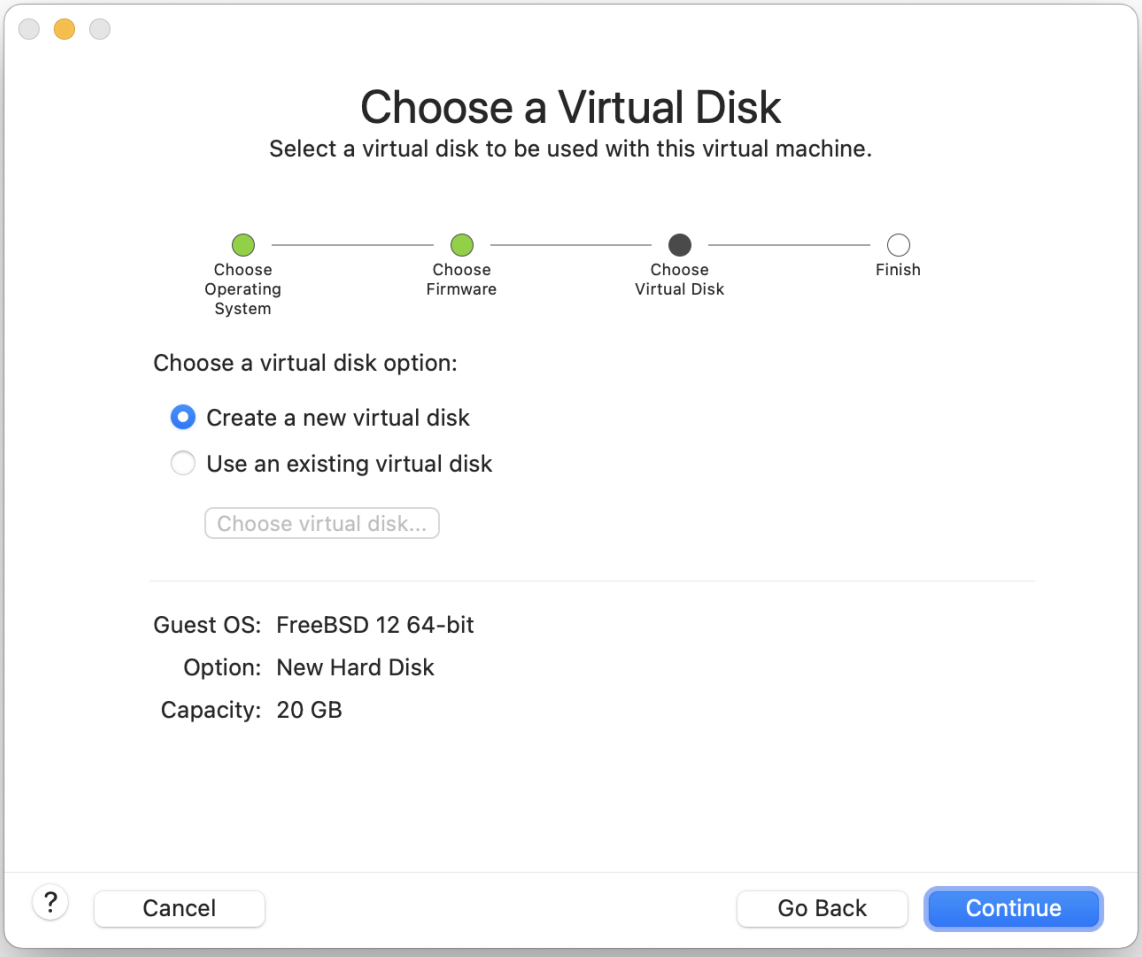
Selecione Outro como o Sistema Operacional e FreeBSD ou FreeBSD 64-bit, como **Versão** quando solicitado:



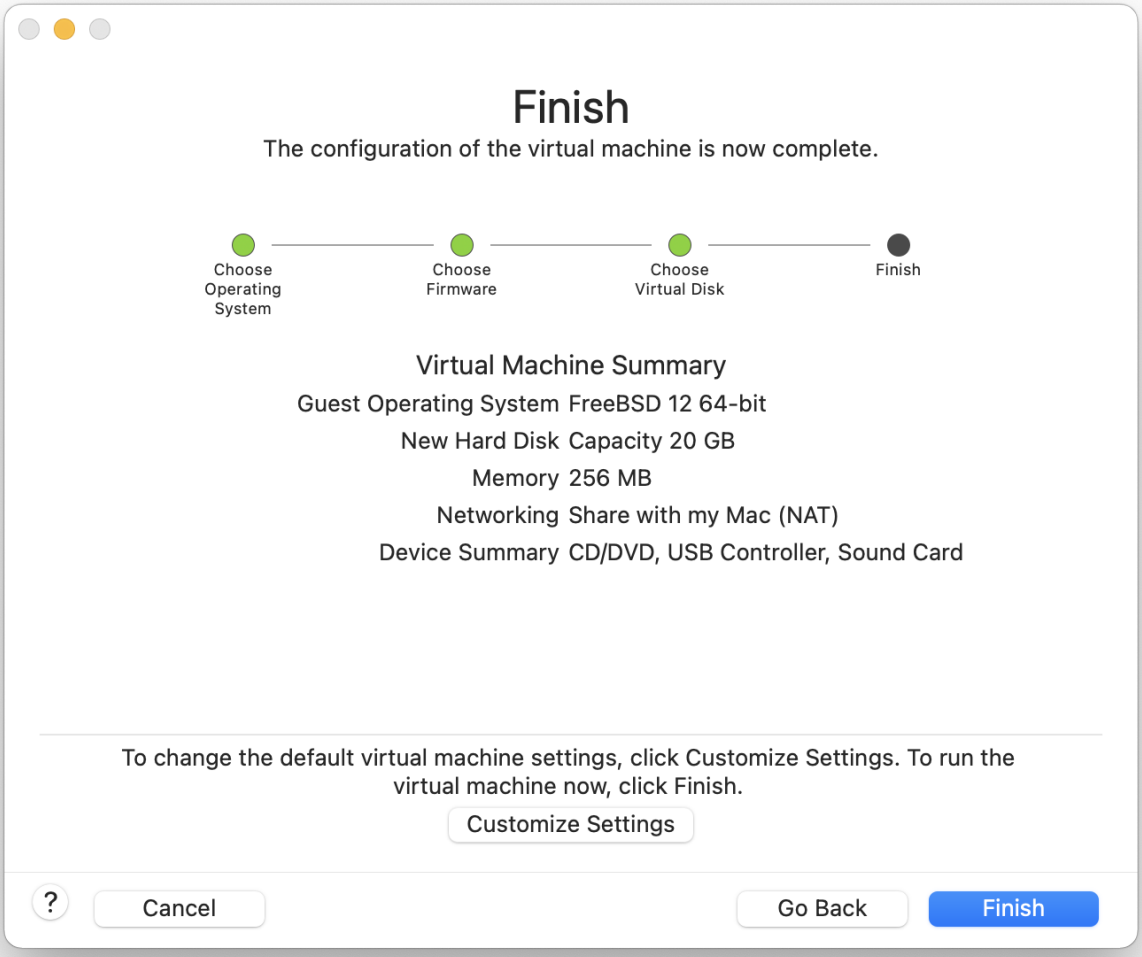
Escolha o nome da máquina virtual e o diretório onde ela deve ser salva:



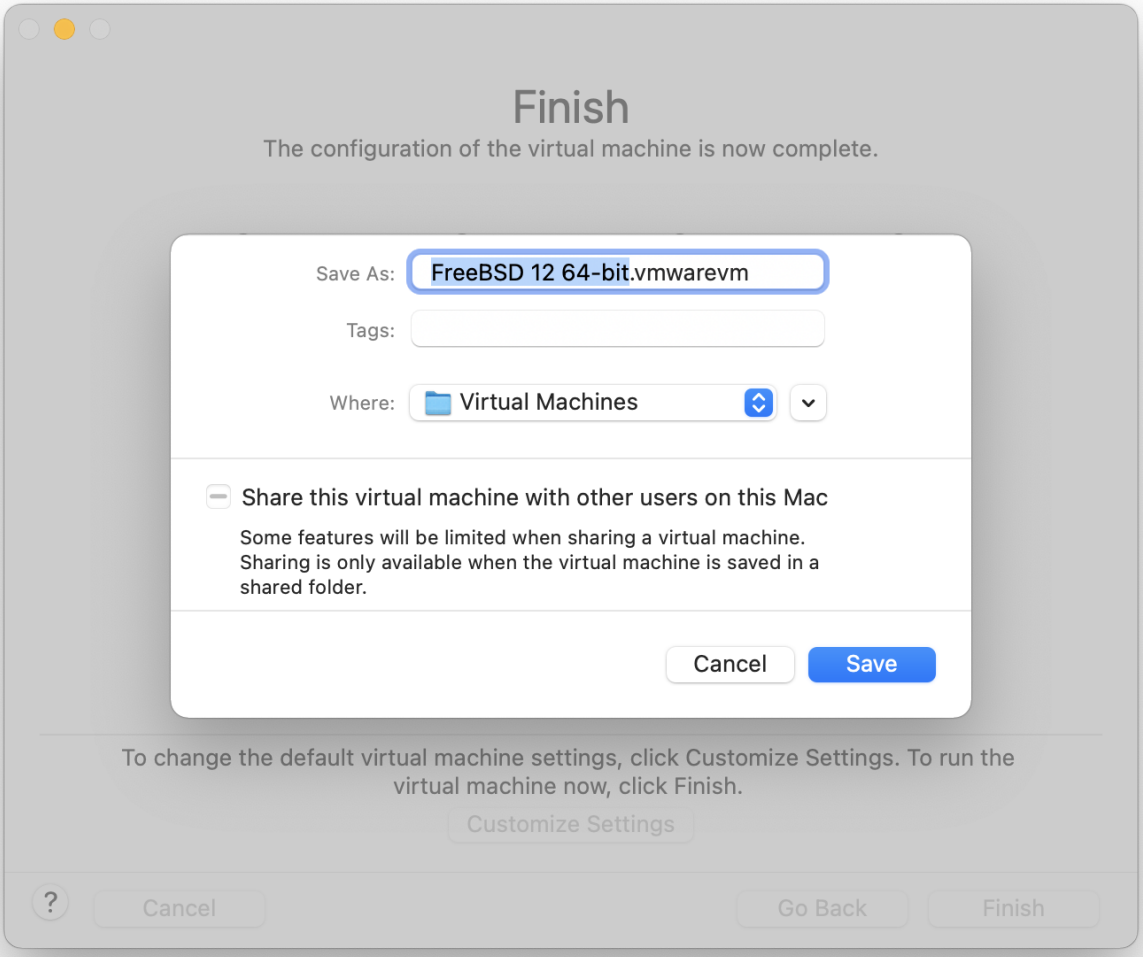
Escolha o tamanho do disco rígido virtual para a máquina virtual:



Escolha o método para instalar a máquina virtual, a partir de uma imagem ISO ou de um CD/DVD:



Clique em Concluir e a máquina virtual inicializará:



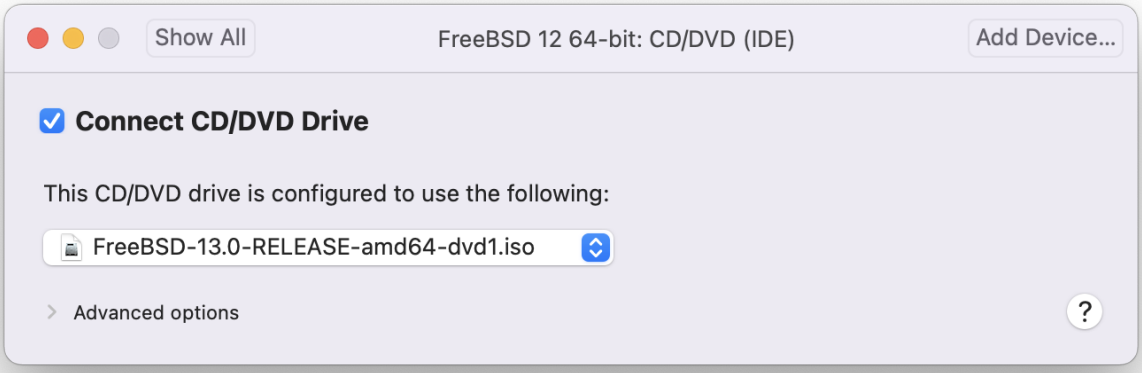
Instale o FreeBSD como de costume:



Quando a instalação estiver concluída, as configurações da máquina virtual poderão ser modificadas, como o uso de memória:



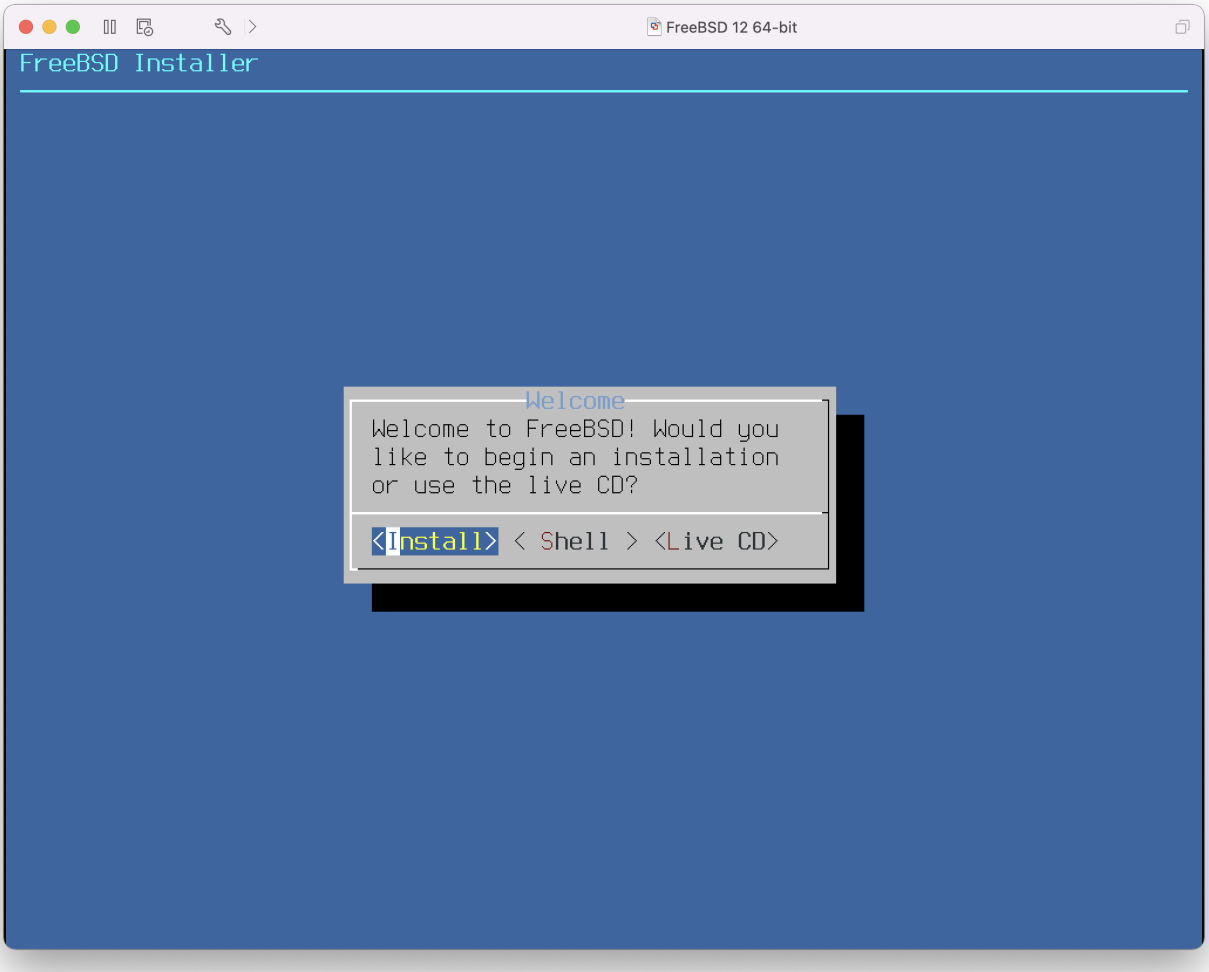
As configurações de hardware do sistema da máquina virtual não podem ser modificadas enquanto a máquina virtual estiver em execução.



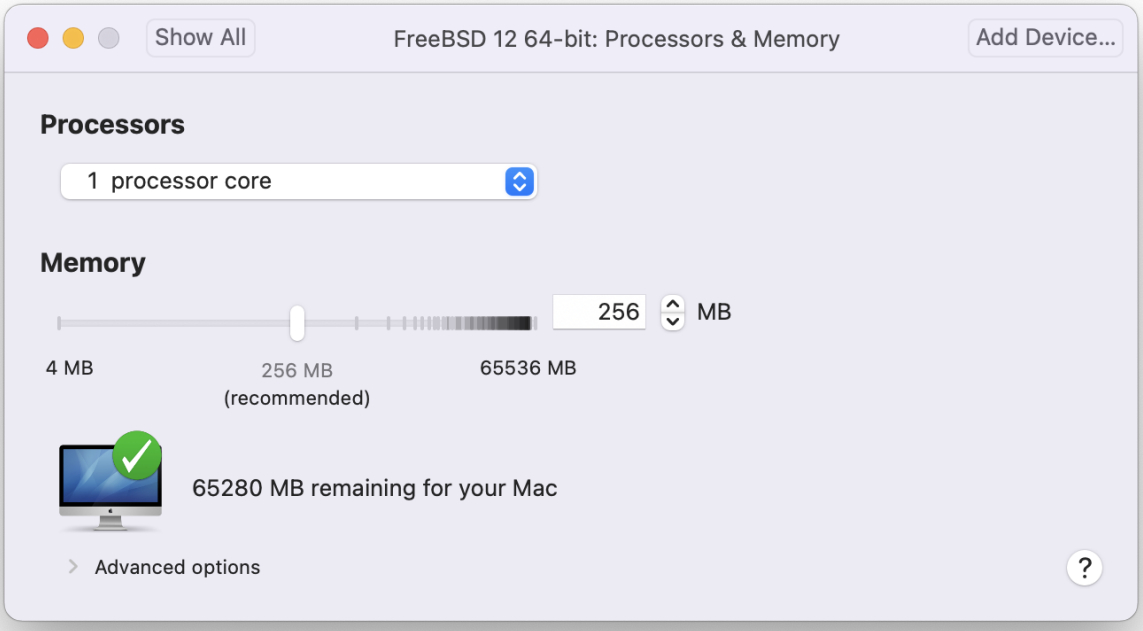
O número de CPUs a que a máquina virtual terá acesso:



O status do dispositivo CD-ROM. Normalmente, o CD/DVD/ISO é desconectado da máquina virtual quando não é mais necessário.



A última coisa a mudar é como a máquina virtual se conectará à rede. Para permitir conexões à máquina virtual de outras máquinas além do host, escolha Conectar diretamente à rede física (Bridged). Caso contrário, Compartilhar a conexão de internet do host (NAT) é preferível para que a máquina virtual possa ter acesso à Internet, porém sem que as demais maquinas da rede possam acessá-la.



Depois de modificar as configurações, inicialize a máquina virtual FreeBSD recém-instalada.

21.4.2. Configurando o FreeBSD no VMware Fusion

Depois que o FreeBSD for instalado com sucesso no Mac OS™ X rodando o VMware Fusion, existem várias etapas de configuração que podem ser executadas para otimizar o sistema para operar virtualizado.

1. Definir variáveis do Boot Loader

   O passo mais importante é reduzir o valor do parâmetro `kern.hz` para reduzir a utilização da CPU do FreeBSD sob o ambiente do VMware Fusion. Isso é feito adicionando a seguinte linha ao /boot/loader.conf:

   ```
   kern.hz=100
   ```

   Sem esta configuração, uma VM idle do FreeBSD rodando sob o VMware Fusion usará aproximadamente 15% da CPU de um único processador iMac™. Após esta mudança, o uso ficará próximo de 5%.

2. Criar um novo arquivo de configuração do kernel

   Todos os drivers de dispositivos FireWire e USB podem ser removidos do arquivo de configuração do kernel personalizado. O VMware Fusion fornece um adaptador de rede virtual usado pelo driver em(4), portanto, todos os dispositivos de rede, exceto o em(4) podem ser removidos do kernel.

3. Configure a rede

   A configuração de rede mais básica usa o DHCP para conectar a máquina virtual à mesma rede local que o host Mac™. Isso pode ser feito adicionando `ifconfig_em0="DHCP"` ao /etc/rc.conf. Configurações de rede mais avançadas estão descritas em Rede Avançada.

21.5. FreeBSD como Sistema Operacional Convidado no VirtualBox™

O FreeBSD funciona bem como um sistema operacional convidado no VirtualBox™. O software de virtualização está disponível para a maioria dos sistemas operacionais comuns, incluindo o próprio FreeBSD.

Os complementos de sistema operacional convidado do VirtualBox™ fornecem suporte para:

- Compartilhamento de área de transferência.
- Integração do ponteiro do mouse.
- Sincronização de hora com o host.
- Redimensionamento de janela.
- Modo Seamless.



Estes comandos são executados na instancia virtualizada do FreeBSD.

Primeiro, instale o pacote ou o port emulators/virtualbox-ose-additions na instancia virtualizada do FreeBSD. Isso irá instalar o port:

```
# cd /usr/ports/emulators/virtualbox-ose-additions && make install clean
```

Adicione estas linhas ao /etc/rc.conf:

```
vboxguest_enable="YES"
vboxservice_enable="YES"
```

Se o ntpd(8) ou o ntpdate(8) estiver sendo utilizado, desabilite a sincronização de horário com o host:

```
vboxservice_flags="--disable-timesync"
```

O Xorg reconhecerá automaticamente o driver `vboxvideo`. Ele também pode ser inserido manualmente no /etc/X11/xorg.conf:

```
Section "Device"
```

```
    Identifier "Card0"
    Driver "vboxvideo"
    VendorName "InnoTek Systemberatung GmbH"
    BoardName "VirtualBox Graphics Adapter"
EndSection
```

Para usar o driver vboxmouse, ajuste a seção do mouse no /etc/X11/xorg.conf:

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver "vboxmouse"
EndSection
```

Usuários do HAL devem criar o arquivo /usr/local/etc/hal/fdi/policy/90-vboxguest.fdi com o conteúdo abaixo ou copiá-lo de /usr/local/shared/hal/fdi/policy/10osvendor/90-vboxguest.fdi:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
# Sun VirtualBox
# Hal driver description for the vboxmouse driver
# $Id: chapter.xml,v 1.33 2012-03-17 04:53:52 eadler Exp $

    Copyright (C) 2008-2009 Sun Microsystems, Inc.

    This file is part of VirtualBox Open Source Edition (OSE, as
    available from http://www.virtualbox.org. This file is free software;
    you can redistribute it and/or modify it under the terms of the GNU
    General Public License (GPL) as published by the Free Software
    Foundation, in version 2 as it comes in the "COPYING" file of the
    VirtualBox OSE distribution. VirtualBox OSE is distributed in the
    hope that it will be useful, but WITHOUT ANY WARRANTY of any kind.

    Please contact Sun Microsystems, Inc., 4150 Network Circle, Santa
    Clara, CA 95054 USA or visit http://www.sun.com if you need
    additional information or have any questions.
-->
<deviceinfo version="0.2">
  <device>
    <match key="info.subsystem" string="pci">
      <match key="info.product" string="VirtualBox guest Service">
        <append key="info.capabilities" type="strlist">input</append>
    <append key="info.capabilities" type="strlist">input.mouse</append>
        <merge key="input.x11_driver" type="string">vboxmouse</merge>
    <merge key="input.device" type="string">/dev/vboxguest</merge>
      </match>
    </match>
  </device>
</deviceinfo>
```

Pastas compartilhadas para transferências de arquivos entre o host e a VM são acessíveis montando-as usando `mount_vboxvfs`. Uma pasta compartilhada pode ser criada no host usando a GUI do VirtualBox ou via `vboxmanage`. Por exemplo, para criar uma pasta compartilhada chamada *myshare* em /mnt/bsdboxshare para a VM denominada *BSDBox*, execute :

```
# vboxmanage sharedfolder add 'BSDBox' --name myshare --hostpath /mnt/bsdboxshare
```

Observe que o nome da pasta compartilhada não deve conter espaços. Monte a pasta compartilhada de dentro do sistema convidado desta forma:

```
# mount_vboxvfs -w myshare /mnt
```

21.6. FreeBSD como Host com VirtualBox™

O VirtualBox™ é um pacote de virtualização completo e ativamente desenvolvido, disponível para a maioria dos sistemas operacionais, incluindo Windows™, Mac OS™, Linux™ e FreeBSD. Ele é igualmente capaz de executar sistemas operacionais convidados como o Windows™ ou UNIX™-like. Ele é distribuído como um software de código aberto, mas com componentes de código fechado disponíveis em um pacote de extensão separado. Esses componentes incluem suporte para dispositivos USB 2.0. Maiores informações podem ser encontradas na página wiki sobre Downloads do VirtualBox. Atualmente, essas extensões não estão disponíveis para o FreeBSD.

21.6.1. Instalando o VirtualBox™

O VirtualBox™ está disponível como um pacote ou port do FreeBSD em emulators/virtualbox-ose. O port pode ser instalado usando estes comandos:

```
# cd /usr/ports/emulators/virtualbox-ose
# make install clean
```

Uma opção útil no menu de configuração do port é o conjunto de programas `GuestAdditions`. Eles fornecem vários recursos úteis em sistemas operacionais convidados, como integração de ponteiro de mouse (permitindo que o mouse seja compartilhado entre host e o sistema convidado sem a necessidade de pressionar um atalho de teclado especial para alternar) e renderização de vídeo mais rápida, especialmente em sistemas convidados Windows™. Os complementos para os sistemas convidados estão disponíveis no menu **Dispositivos**, após a conclusão da instalação do sistema convidado.

Algumas alterações de configuração são necessárias antes do VirtualBox™ ser iniciado pela primeira vez. O port instala um módulo de kernel em /boot/modules o qual deve ser carregado no kernel em execução:

```
# kldload vboxdrv
```

Para garantir que o módulo seja sempre carregado após uma reinicialização, adicione esta linha ao /boot/loader.conf:

```
vboxdrv_load="YES"
```

Para usar os módulos do kernel que permitem conexões de rede bridged ou host-only, adicione esta linha ao /etc/rc.conf e reinicie o computador:

```
vboxnet_enable="YES"
```

O grupo `vboxusers` é criado durante a instalação do VirtualBox™. Todos os usuários que precisam acessar o VirtualBox™ deverão ser adicionados como membros desse grupo. O comando `pw` pode ser usado para adicionar novos membros:

```
# pw groupmod vboxusers -m yourusername
```

As permissões padrão para o /dev/vboxnetctl são restritivas e precisam ser alteradas para redes em modo Bridged:

```
# chown root:vboxusers /dev/vboxnetctl
# chmod 0660 /dev/vboxnetctl
```

Para tornar esta permissão permanente, adicione estas linhas ao /etc/devfs.conf:

```
own     vboxnetctl root:vboxusers
perm    vboxnetctl 0660
```

Para iniciar o VirtualBox™, digite a partir de uma sessão Xorg:

```
% VirtualBox
```

Para mais informações sobre como configurar e usar o VirtualBox™, consulte o site oficial. Para obter informações específicas sobre o FreeBSD e instruções para a solução de problemas, consulte a página relevante no wiki do FreeBSD.

21.6.2. Suporte USB no VirtualBox™

O VirtualBox™ pode ser configurado para passar dispositivos USB para o sistema operacional convidado. O controlador host da versão do OSE está limitado a emular dispositivos USB 1.1 até que o pacote de extensão que suporta dispositivos USB 2.0 e 3.0 esteja disponível no FreeBSD.

Para que o VirtualBox™ esteja ciente dos dispositivos USB conectados à máquina, o usuário precisa ser um membro do grupo `operator`.

```
# pw groupmod operator -m yourusername
```

Em seguida, adicione as seguintes linhas em /etc/devfs.rules ou crie o arquivo se ele ainda não existir:

```
[system=10]
add path 'usb/*' mode 0660 group operator
```

Em seguida, adicione as seguintes linhas ao /etc/rc.conf:

```
devfs_system_ruleset="system"
```

Então reinicie o devfs:

```
# service devfs restart
```

Reinicie a sessão de login e o VirtualBox™ para que essas alterações entrem em vigor e crie os filtros USB conforme necessário.

21.6.3. Acesso ao drive de DVD/CD no Host VirtualBox™

O acesso às unidades de DVD/CD do Host a partir dos convidados é obtido através do compartilhamento das unidades físicas. Dentro do VirtualBox™, isso é configurado a partir da janela Armazenamento nas Configurações da máquina virtual. Se necessário, crie primeiro um dispositivo vazio IDECD/DVD. Em seguida, escolha a unidade do host no menu pop-up para a seleção de unidade virtual de CD/DVD. Uma caixa de seleção rotulada como `Passthrough` será exibida. Isso permitirá que a máquina virtual use o hardware diretamente. Por exemplo, CDs de áudio ou o gravador só funcionará se esta opção estiver selecionada.

O HAL precisa ser executado para que as funções de DVD/CD do VirtualBox™ funcionem, então habilite-o no /etc/rc.conf e inicie-o se ele ainda não estiver em execução:

```
hald_enable="YES"
```

```
# service hald start
```

Para que os usuários possam usar as funções de DVD/CD do VirtualBox™, eles precisam acessar /dev/xpt0, /dev/cdN, e /dev/passN. Isso geralmente é obtido tornando o usuário um membro do grupo `operator`. As permissões para esses dispositivos devem ser corrigidas adicionando estas linhas ao /etc/devfs.conf:

```
perm cd* 0660
```

```
perm xpt0 0660
perm pass* 0660
```

```
# service devfs restart
```

21.7. FreeBSD como um Host bhyve

O hypervisor bhyveBSD-licensed tornou-se parte do sistema base com o FreeBSD 10.0-RELEASE. Este hypervisor suporta uma grande variedade de sistemas operacionais convidados, incluindo FreeBSD, OpenBSD e muitas distribuições Linux™. Por padrão, o bhyve fornece acesso ao console serial e não emula um console gráfico. Os recursos de offload de virtualização das CPUs mais recentes são usados para evitar os métodos legados de tradução de instruções e de gerenciamento manual de mapeamentos de memória.

O design do bhyve requer um processador que suporte tabelas de páginas estendidas da Intel™ (EPT) ou a Indexação Rápida de Virtualização da AMD™ (RVI) ou Tabelas de Páginas Aninhadas (NPT). Hospedar sistemas operacionais convidados Linux™ ou convidados FreeBSD com mais de uma vCPU requer suporte a modo irrestrito de VMX (UG). A maioria dos processadores mais recentes, especificamente o Intel™Core™ i3/i5/i7 e o Intel™Xeon™ E3/E5/E7, suportam esses recursos. O suporte UG foi introduzido com a microarquitetura Westmere da Intel. Para obter uma lista completa dos processadores Intel™ que suportam EPT, consulte https://ark.intel.com/content/www/us/en/ark/search/featurefilter.html?productType=873&0_ExtendedPageTables=True. O RVI é encontrado na terceira geração e depois nos processadores AMD Opteron™ (Barcelona). A maneira mais fácil de saber se um processador suporta o bhyve é executar o `dmesg` ou procurar no /var/run/dmesg.boot pelo o Sinalizador de recurso do processador `POPCNT` na linha `Features2` para processadores AMD™ ou `EPT` e `UG` na linha `VT-x` para os processadores Intel™.

21.7.1. Preparando o host

O primeiro passo para criar uma máquina virtual no bhyve é configurar o sistema host. Primeiro, carregue o módulo do kernel bhyve:

```
# kldload vmm
```

Em seguida, crie uma interface tap para o dispositivo de rede na máquina virtual para anexar. Para que o dispositivo de rede participe da rede, crie também uma interface de bridge contendo a interface tap e a interface física como membros. Neste exemplo, a interface física é *igb0*:

```
# ifconfig tap0 create
# sysctl net.link.tap.up_on_open=1
net.link.tap.up_on_open: 0 -> 1
# ifconfig bridge0 create
# ifconfig bridge0 addm igb0 addm tap0
# ifconfig bridge0 up
```

21.7.2. Criando um Sistema Operacional Convidado do FreeBSD

Crie um arquivo para usar como o disco virtual da máquina convidada. Especifique o tamanho e o nome do disco virtual:

```
# truncate -s 16G guest.img
```

Baixe uma imagem de instalação do FreeBSD para instalar:

```
# fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/10.3/FreeBSD-10.3-
RELEASE-amd64-bootonly.iso
FreeBSD-10.3-RELEASE-amd64-bootonly.iso       100% of  230 MB  570 kBps 06m17s
```

O FreeBSD vem com um script de exemplo para executar uma máquina virtual com o bhyve. O script iniciará a máquina virtual e a executará em um loop, para que ela seja reiniciada automaticamente se houver falha. O script usa várias opções para controlar a configuração da máquina: `-c` controla o número de CPUs virtuais, `-m` limita a quantidade de memória disponível para o sistema operacional convidado, `-t` define qual dispositivo tap usar, `-d` indica qual imagem de disco usar, `-i` indica ao bhyve para inicializar a partir da imagem CD em vez do disco, e `-I` define qual imagem de CD deve ser usada. O último parâmetro é o nome da máquina virtual, usada para rastrear as máquinas em execução. Este exemplo inicia a máquina virtual no modo de instalação:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 1 -m 1024M -t tap0 -d guest.img -i -I
FreeBSD-10.3-RELEASE-amd64-bootonly.iso guestname
```

A máquina virtual inicializará e iniciará o instalador. Depois de instalar um sistema na máquina virtual, quando o sistema perguntar sobre a inserção em um shell no final da instalação, escolha **[Yes]**.

Reinicialize a máquina virtual. Enquanto a reinicialização da máquina virtual fará o bhyve finalizar, o script vmrun.sh executa o `bhyve` em um loop e o reiniciará automaticamente. Quando isso acontecer, escolha a opção de reinicialização no menu do carregador de inicialização para escapar do loop. Agora o convidado pode ser iniciado a partir do disco virtual:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 4 -m 1024M -t tap0 -d guest.img guestname
```

21.7.3. Criando um Sistema Operacional convidado Linux™

Para inicializar sistemas operacionais diferentes do FreeBSD, o port sysutils/grub2-bhyve deve ser instalada primeiro.

Em seguida, crie um arquivo para usar como o disco virtual da máquina convidada:

```
# truncate -s 16G linux.img
```

Iniciar uma máquina virtual com o bhyve é um processo de duas etapas. Primeiro um kernel deve ser carregado, então o sistema operacional convidado pode ser iniciado. O kernel Linux™ é carregado com o sysutils/grub2-bhyve. Crie um device.map que o grub usará para mapear os dispositivos virtuais para os arquivos no sistema host:

```
(hd0) ./linux.img
(cd0) ./somelinux.iso
```

Use o sysutils/grub2-bhyve para carregar o kernel Linux™ de uma imagem ISO:

```
# grub-bhyve -m device.map -r cd0 -M 1024M linuxguest
```

Isto irá iniciar o grub. Se o CD de instalação contiver um grub.cfg, um menu será exibido. Caso contrário, os arquivos `vmlinuz` e `initrd` devem ser localizados e carregados manualmente:

```
grub> ls
(hd0) (cd0) (cd0,msdos1) (host)
grub> ls (cd0)/isolinux
boot.cat boot.msg grub.conf initrd.img isolinux.bin isolinux.cfg memtest
splash.jpg TRANS.TBL vesamenu.c32 vmlinuz
grub> linux (cd0)/isolinux/vmlinuz
grub> initrd (cd0)/isolinux/initrd.img
grub> boot
```

Agora que o kernel Linux™ está carregado, o sistema convidado pode ser iniciado:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-
blk,./linux.img \
    -s 4:0,ahci-cd,./somelinux.iso -l com1,stdio -c 4 -m 1024M linuxguest
```

O sistema inicializará e iniciará o instalador. Depois de instalar um sistema na máquina virtual, reinicialize a máquina virtual. Isso fará com que o bhyve seja encerrado. A instância da máquina virtual precisa ser destruída antes de poder ser iniciada novamente:

```
# bhyvectl --destroy --vm=linuxguest
```

Agora, o sistema convidado pode ser iniciado diretamente do disco virtual. Carregue o kernel:

```
# grub-bhyve -m device.map -r hd0,msdos1 -M 1024M linuxguest
grub> ls
(hd0) (hd0,msdos2) (hd0,msdos1) (cd0) (cd0,msdos1) (host)
(lvm/VolGroup-lv_swap) (lvm/VolGroup-lv_root)
grub> ls (hd0,msdos1)/
lost+found/ grub/ efi/ System.map-2.6.32-431.el6.x86_64 config-2.6.32-431.el6.x
```

```
86_64 symvers-2.6.32-431.el6.x86_64.gz vmlinuz-2.6.32-431.el6.x86_64
initramfs-2.6.32-431.el6.x86_64.img
grub> linux (hd0,msdos1)/vmlinuz-2.6.32-431.el6.x86_64 root=/dev/mapper/VolGroup-
lv_root
grub> initrd (hd0,msdos1)/initramfs-2.6.32-431.el6.x86_64.img
grub> boot
```

Inicialize a máquina virtual:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 \
    -s 3:0,virtio-blk,./linux.img -l com1,stdio -c 4 -m 1024M linuxguest
```

O Linux™ iniciará agora na máquina virtual e, eventualmente, apresentará o prompt de login. Faça o login e use a máquina virtual. Quando terminar, reinicialize a máquina virtual para sair do bhyve. Destrua a instância da máquina virtual:

```
# bhyvectl --destroy --vm=linuxguest
```

21.7.4. Inicializando máquinas virtuais bhyve com Firmware UEFI

Além do bhyveload e do grub-bhyve, o hypervisor bhyve também pode inicializar máquinas virtuais usando o firmware do espaço de usuário UEFI . Esta opção pode suportar sistemas operacionais convidados que não são suportados pelos outros carregadores.

Para utilizar o suporte ao UEFI no bhyve, primeiro obtenha as imagens de firmware UEFI. Isto pode ser feito instalando o port ou pacote sysutils/bhyve-firmware.

Com o firmware no lugar, adicione os sinalizadores `-l bootrom,/path/to/firmware` à linha de comando do bhyve. A sintaxe real do bhyve pode se parecer com a seguinte:

```
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI.fd \
guest
```

O sysutils/bhyve-firmware também contém um firmware habilitado para CSM, para inicializar sistemas operacionais hóspedes sem suporte à UEFI no modo de BIOS legado:

```
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI_CSM.fd \
guest
```

21.7.5. Framebuffer UEFI Gráfico para bhyve

O suporte ao firmware UEFI é particularmente útil em sistemas operacionais convidados predominantemente gráficos, como o Microsoft Windows™.

O suporte para o framebuffer UEFI-GOP também pode ser ativado com os sinalizadores `-s 29,fbuf,tcp=0.0.0.0:5900`. A resolução do framebuffer pode ser configurada com `w=800` e `h=600` e o bhyve pode ser instruído para aguardar uma conexão VNC antes de inicializar o sistema operacional convidado adicionando `wait`. O framebuffer pode ser acessado pelo host ou pela rede através do protocolo VNC. Além disso, `-s 30,xhci,tablet` pode ser adicionado para obter a sincronização precisa do cursor do mouse com o host.

O comando bhyve resultante ficaria assim:

```
# bhyve -AHP -s 0:0,hostbridge -s 31:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-s 29,fbuf,tcp=0.0.0.0:5900,w=800,h=600,wait \
-s 30,xhci,tablet \
-l bootrom,/usr/local/share/uefi-firmware/BHYVE_UEFI.fd \
guest
```

Observe que, no modo de emulação do BIOS, o framebuffer deixará de receber atualizações quando o controle for passado do firmware para o sistema operacional convidado.

21.7.6. Usando o ZFS com os sistemas operacionais convidados no bhyve

Se o ZFS estiver disponível na máquina host, o uso de volumes ZFS em vez de arquivos de imagem de disco pode fornecer benefícios significativos de desempenho para as VMs convidadas. Um volume ZFS pode ser criado por:

```
# zfs create -V16G -o volmode=dev zroot/linuxdisk0
```

Ao iniciar a VM, especifique o volume ZFS como a unidade de disco:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s3:0,virtio
-blk,/dev/zvol/zroot/linuxdisk0 \
    -l com1,stdio -c 4 -m 1024M linuxguest
```

21.7.7. Consoles de máquinas virtuais

É vantajoso executar o console do bhyve em uma ferramenta de gerenciamento de sessão, como o sysutils/tmux ou sysutils/screen, para que possa desanexar e reanexar o console. Também é possível ter o console do bhyve como um dispositivo de modem nulo o qual pode ser acessado com o comando `cu`. Para fazer isso, carregue o módulo do kernel nmdm e substitua `-l com1,stdio` with `-l com1,/dev/nmdm0A`. Os dispositivos /dev/nmdm são criados automaticamente conforme necessário,

onde cada um é um par, correspondente às duas extremidades do cabo de modem nulo (/dev/nmdm0A e /dev/nmdm0B). Veja nmdm(4) para maiores informações.

```
# kldload nmdm
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-
blk,./linux.img \
    -l com1,/dev/nmdm0A -c 4 -m 1024M linuxguest
# cu -l /dev/nmdm0B
Connected

Ubuntu 13.10 handbook ttyS0

handbook login:
```

21.7.8. Gerenciando Máquinas Virtuais

Um nó de dispositivo é criado em /dev/vmm para cada máquina virtual. Isso permite que o administrador veja facilmente uma lista das máquinas virtuais em execução:

```
# ls -al /dev/vmm
total 1
dr-xr-xr-x   2 root  wheel    512 Mar 17 12:19 ./
dr-xr-xr-x  14 root  wheel    512 Mar 17 06:38 ../
crw-------   1 root  wheel  0x1a2 Mar 17 12:20 guestname
crw-------   1 root  wheel  0x19f Mar 17 12:19 linuxguest
crw-------   1 root  wheel  0x1a1 Mar 17 12:19 otherguest
```

Uma máquina virtual especificada pode ser destruída usando `bhyvectl`:

```
# bhyvectl --destroy --vm=guestname
```

21.7.9. Configuração Persistente

Para configurar o sistema para iniciar os sistemas operacionais convidados do bhyve no momento da inicialização, as seguintes configurações devem ser feitas nos arquivos especificados:

1. /etc/sysctl.conf

   ```
   net.link.tap.up_on_open=1
   ```

2. /etc/rc.conf

   ```
   cloned_interfaces="bridge0 tap0"
   ifconfig_bridge0="addm igb0 addm tap0"
   ```

```
kld_list="nmdm vmm"
```

21.8. FreeBSD como Host Xen™

O Xen é um hypervisor tipo 1 licenciado sob a GPLv2 para arquiteturas Intel™ e ARM™. O FreeBSD suporta domínios não privilegiados (máquina virtual) nas plataformas i386™ e AMD™ 64-Bit DomU e Amazon EC2 desde o FreeBSD 8.0 e incluiu o suporte ao domínio de controle Dom0 (host) no FreeBSD 11.0. O suporte para domínios para-virtualizados (PV) foi removido do FreeBSD 11 em favor de domínios virtualizados de hardware (HVM), o que proporciona melhor desempenho.

O Xen™ é um hypervisor bare-metal, o que significa que é o primeiro programa carregado após o BIOS. Um convidado especial privilegiado chamado Domain-0 (`Dom0` para abreviar) é então iniciado. O Dom0 usa seus privilégios especiais para acessar diretamente o hardware físico subjacente, tornando-o uma solução de alto desempenho. Ele é capaz de acessar os controladores de disco e adaptadores de rede diretamente. As ferramentas de gerenciamento do Xen™ para gerenciar e controlar o hypervisor Xen™ também são usadas pelo Dom0 para criar, listar e destruir VMs. Dom0 fornece discos virtuais e recursos de rede para domínios sem privilégios, geralmente chamados de `DomU`. O Xen™ Dom0 pode ser comparado ao console de serviço de outras soluções de hypervisor , enquanto o DomU é onde as VMs convidadas são executadas.

O Xen™ pode migrar VMs entre diferentes servidores Xen™. Quando os dois hosts xen compartilham o mesmo armazenamento subjacente, a migração pode ser feita sem a necessidade de primeiro desligar a VM. Em vez disso, a migração é executada ao vivo enquanto o DomU está em execução e não há necessidade de reiniciá-lo ou planejar um tempo de inatividade. Isso é útil em cenários de manutenção ou em janelas de atualização para garantir que os serviços fornecidos pelo DomU continuem disponiveis. Muitos outros recursos do Xen™ estão listados na página wiki com a visão global sobre o Xen. Note que ainda nem todos os recursos são suportados no FreeBSD.

21.8.1. Requisitos de hardware para o Xen™ Dom0

Para executar o hypervisor Xen™ em um host, são necessárias certas funcionalidades de hardware. Os domínios virtualizados de hardware requerem o suporte à Tabela de Páginas Estendidas (EPT) e à Unidade de Gerenciamento de Memória de Entrada / Saída (IOMMU) no processador do host.



Para executar um Xen™ Dom0 no FreeBSD, a maquina deve ser inicializada usando o boot legado (BIOS).

21.8.2. Configuração do Xen™ Dom0 Domínio de Controle

Os usuários do FreeBSD 11 devem instalar os pacotes emulators/xen-kernel47 e sysutils/xen-tools47 que são baseados no Xen versão 4.7. Sistemas rodando o FreeBSD-12.0 ou mais novo podem usar o Xen 4.11 fornecido por emulators/xen-kernel411 e sysutils/xen-tools411, respectivamente.

Os arquivos de configuração devem ser editados para preparar o host para a integração do Dom0 após a instalação dos pacotes do Xen. Uma entrada para /etc/sysctl.conf desabilita o limite de quantas páginas de memória podem ser conectadas. Caso contrário, as VMs do DomU com requisitos de memória mais altos não serão executadas.

```
# echo 'vm.max_wired=-1' >> /etc/sysctl.conf
```

Outra configuração relacionada à memória envolve a alteração do /etc/login.conf, configurando a opção `memorylocked` para `unlimited`. Caso contrário, a criação de domínios DomU poderá falhar com erros `Cannot allocate memory`. Depois de fazer a mudança no /etc/login.conf, execute o comando `cap_mkdb` para atualizar o banco de dados de recursos. Veja Limites de Recursos para detalhes.

```
# sed -i '' -e 's/memorylocked=64K/memorylocked=unlimited/' /etc/login.conf
# cap_mkdb /etc/login.conf
```

Adicione uma entrada para o console do Xen™ ao /etc/ttys:

```
# echo 'xc0     "/usr/libexec/getty Pc"         xterm   onifconsole  secure' >>
/etc/ttys
```

A seleção de um kernel Xen™ no /boot/loader.conf ativa o Dom0. O Xen™ também requer recursos como CPU e memória da máquina host para ele mesmo e para outros domínios DomU. Quanto de CPU e memória depende dos requisitos individuais e das capacidades de hardware. Neste exemplo, 8 GB de memória e 4 CPUs virtuais são disponibilizados para o Dom0. O console serial também é ativado e as opções de log são definidas.

O seguinte comando é usado para pacotes Xen 4.7:

```
# sysrc -f /boot/loader.conf hw.pci.mcfg=0
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4 dom0pvh=1
console=com1,vga com1=115200,8n1 guest_loglvl=all loglvl=all"
```

Para as versões Xen 4.11 e superiores, o seguinte comando deve ser usado:

```
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4 dom0=pvh
console=com1,vga com1=115200,8n1 guest_loglvl=all loglvl=all"
```



Os arquivos de log criados pelo Xen™ para as VMs do DomU são armazenados em /var/log/xen. Por favor, certifique-se de verificar o conteúdo do diretório em caso de problemas.

Ative o serviço xencommons durante a inicialização do sistema:

```
# sysrc xencommons_enable=yes
```

Essas configurações são suficientes para iniciar um sistema habilitado para Dom0. No entanto, falta a funcionalidade de rede para as máquinas DomU. Para corrigir isso, defina uma interface em bridge com a NIC principal do sistema que as VMs DomU poderão usar para se conectar à rede. Substitua *em0* pelo nome da interface de rede do host.

```
# sysrc cloned_interfaces="bridge0"
# sysrc ifconfig_bridge0="addm em0 SYNCDHCP"
# sysrc ifconfig_em0="up"
```

Reinicie o host para carregar o kernel Xen™ e inicie o Dom0.

```
# reboot
```

Após inicializar com sucesso o kernel Xen™ e efetuar login no sistema novamente, a ferramenta de gerenciamento do Xen™, `xl` é usada para mostrar informações sobre os domínios.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  8192     4     r-----     962.0
```

A saída confirma que o Dom0 (chamado `Domain-0`) tem o ID `0` e está em execução. Ele também possui a memória e as CPUs virtuais que foram definidas anteriormente no /boot/loader.conf. Mais informações podem ser encontradas na Documentação do Xen. Agora as VMs convidadas do DomU podem ser criadas.

21.8.3. Configuração da VM Convidada Xen™ DomU

Domínios desprivilegiados consistem em um arquivo de configuração e discos rígidos virtuais ou físicos. Os discos virtuais para armazenamento do DomU podem ser arquivos criados pelo truncate(1) ou volumes ZFS, conforme descrito em Criando e Destruindo Volumes. Neste exemplo, um volume de 20 GB é usado. Uma VM é criada com o volume ZFS, uma imagem ISO do FreeBSD, 1 GB de RAM e duas CPUs virtuais. O arquivo ISO de instalação é obtido com o fetch(1) e salvo localmente em um arquivo chamado freebsd.iso.

```
# fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/12.0/FreeBSD-12.0-
RELEASE-amd64-bootonly.iso -o freebsd.iso
```

Um volume de 20 GB do ZFS chamado xendisk0 é criado para servir como espaço em disco para a VM.

```
# zfs create -V20G -o volmode=dev zroot/xendisk0
```

A nova VM DomU convidada é definida em um arquivo. Algumas definições específicas, como nome, mapa de teclado e detalhes da conexão VNC, também são definidas. O seguinte freebsd.cfg

contém uma configuração mínima de DomU para este exemplo:

```
# cat freebsd.cfg
builder = "hvm" ①
name = "freebsd" ②
memory = 1024 ③
vcpus = 2 ④
vif = [ 'mac=00:16:3E:74:34:32,bridge=bridge0' ] ⑤
disk = [
'/dev/zvol/tank/xendisk0,raw,hda,rw', ⑥
'/root/freebsd.iso,raw,hdc:cdrom,r' ⑦
  ]
vnc = 1 ⑧
vnclisten = "0.0.0.0"
serial = "pty"
usbdevice = "tablet"
```

Estas linhas são explicadas com mais detalhes:

① Isso define que tipo de virtualização usar. `hvm` refere-se à virtualização assistida por hardware ou à máquina virtual de hardware. Os sistemas operacionais convidados podem ser executados sem modificação em CPUs com extensões de virtualização, fornecendo quase o mesmo desempenho que a execução em hardware físico. `generic` é o valor padrão e cria um domínio PV.

② Nome desta máquina virtual para distingui-la de outras executadas no mesmo Dom0. Requerido.

③ Quantidade de RAM em megabytes para disponibilizar para a VM. Esse valor é subtraído da memória total disponível do hypervisor, não da memória do Dom0.

④ Número de CPUs virtuais disponíveis para a VM convidada. Para um melhor desempenho, não crie convidados com mais CPUs virtuais do que o número de CPUs físicas no host.

⑤ Adaptador de rede virtual. Esta é a bridge conectada à interface de rede do host. O parâmetro `mac` é o endereço MAC definido na interface de rede virtual. Este parâmetro é opcional, se nenhum MAC for fornecido, o Xen™ irá gerar um aleatório.

⑥ Caminho completo para o disco, arquivo ou volume ZFS do armazenamento em disco para essa VM. As opções e as várias definições de disco são separadas por vírgulas.

⑦ Define o meio de inicialização a partir do qual o sistema operacional inicial é instalado. Neste exemplo, é a imagem ISO baixada anteriormente. Consulte a documentação do Xen™ para outros tipos de dispositivos e outras opções para configurar.

⑧ Opções que controlam a conectividade do VNC para o console serial do DomU. Em ordem, estes são: ativa suporte ao VNC, define o endereço IP no qual escutar, device node para o console serial e o método de entrada para posicionamento preciso do mouse e outros métodos de entrada. `keymap` define qual mapa de teclas usar, sendo `english` por padrão.

Após o arquivo ter sido criado com todas as opções necessárias, o DomU é criado passando-o como um parâmetro para o comando `xl create`.

```
# xl create freebsd.cfg
```



Cada vez que o Dom0 é reiniciado, o arquivo de configuração deve ser passado para `xl create` novamente para recriar o DomU. Por padrão, somente o Dom0 é criado após uma reinicialização, não as VMs individuais. As VMs podem continuar de onde pararam, pois armazenaram o sistema operacional no disco virtual. A configuração da máquina virtual pode mudar com o tempo (por exemplo, ao adicionar mais memória). Os arquivos de configuração da máquina virtual devem ter um backup e manter-se disponíveis para poder recriar a VM convidada quando necessário.

A saída de `xl list` confirma que o DomU foi criado.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  8192     4     r-----  1653.4
freebsd                                      1  1024     1     -b----   663.9
```

Para iniciar a instalação do sistema operacional base, inicie o cliente VNC, direcionando-o para o endereço de rede principal do host ou para o endereço IP definido na linha `vnclisten` do freebsd.cfg. Depois que o sistema operacional tiver sido instalado, desligue o DomU e desconecte o visualizador VNC. Edite o freebsd.cfg, removendo a linha com a definição `cdrom` ou comentando-a inserindo um caractere `#` no início da linha. Para carregar esta nova configuração, é necessário remover o DomU antigo com `xl destroy`, passando o nome ou o id como parâmetro. Depois, recrie-o usando o freebsd.cfg modificado.

```
# xl destroy freebsd
# xl create freebsd.cfg
```

A máquina pode então ser acessada novamente usando o visualizador VNC. Desta vez, ele será inicializado a partir do disco virtual em que o sistema operacional foi instalado e pode ser usado como uma máquina virtual.

21.8.4. Solução de problemas

Esta seção contém informações básicas para ajudar a solucionar problemas encontrados ao usar o FreeBSD como host ou convidado do Xen™.

21.8.4.1. Solução de problemas de inicialização do host

Observe que as dicas de solução de problemas a seguir são destinadas ao Xen™ 4.11 ou mais recente. Se você ainda estiver usando o Xen™ 4.7 e tendo problemas, considere migrar para uma versão mais recente do Xen™.

Para solucionar problemas de inicialização do host, você provavelmente precisará de um cabo

serial ou de um cabo USB de depuração. Uma saída de boot verbosa do Xen™ pode ser obtida adicionando-se parametros à opção `xen_cmdline` encontrada no loader.conf. Alguns parametros de depuração relevantes são:

- `iommu=debug`: pode ser usado para imprimir informações de diagnóstico adicionais sobre o iommu.
- `dom0=verbose`: pode ser usado para imprimir informações de diagnóstico adicionais sobre o processo de compilação dom0.
- `sync_console`: flag para forçar a saída síncrona do console. Útil para depuração para evitar a perda de mensagens devido à limitação de taxa. Nunca use essa opção em ambientes de produção, pois ela pode permitir que convidados mal-intencionados realizem ataques DoS contra o Xen™ usando o console.

O FreeBSD também deve ser inicializado no modo verbose para identificar quaisquer problemas. Para ativar a inicialização detalhada, execute este comando:

```
# sysrc -f /boot/loader.conf boot_verbose="YES"
```

Se nenhuma dessas opções ajudar a resolver o problema, envie o registro de inicialização serial para freebsd-xen@FreeBSD.org e xen-devel@lists.xenproject.org para uma análise mais aprofundada.

21.8.4.2. Solução de problemas na criação de VMs convidadas

Problemas também podem surgir ao criar convidados, as informações a seguir tentam fornecer alguma ajuda para aqueles que precisarem diagnosticar problemas de criação de convidados.

A causa mais comum de falhas na criação de convidados é o comando `xl` cuspindo algum erro e saindo com um código de retorno diferente de 0. Se o erro fornecido não for suficiente para ajudar a identificar o problema, uma saída mais detalhada pode ser obtida do comando `xl` usando-se a opção `v` repetidamente.

```
# xl -vvv create freebsd.cfg
Parsing config from freebsd.cfg
libxl: debug: libxl_create.c:1693:do_domain_create: Domain 0:ao 0x800d750a0: create:
how=0x0 callback=0x0 poller=0x800d6f0f0
libxl: debug: libxl_device.c:397:libxl__device_disk_set_backend: Disk vdev=xvda
spec.backend=unknown
libxl: debug: libxl_device.c:432:libxl__device_disk_set_backend: Disk vdev=xvda, using
backend phy
libxl: debug: libxl_create.c:1018:initiate_domain_create: Domain 1:running bootloader
libxl: debug: libxl_bootloader.c:328:libxl__bootloader_run: Domain 1:not a PV/PVH
domain, skipping bootloader
libxl: debug: libxl_event.c:689:libxl__ev_xswatch_deregister: watch w=0x800d96b98:
deregister unregistered
domainbuilder: detail: xc_dom_allocate: cmdline="", features=""
domainbuilder: detail: xc_dom_kernel_file: filename
="/usr/local/lib/xen/boot/hvmloader"
```

```
domainbuilder: detail: xc_dom_malloc_filemap    : 326 kB
libxl: debug: libxl_dom.c:988:libxl__load_hvm_firmware_module: Loading BIOS:
/usr/local/shared/seabios/bios.bin
...
```

Se a saída detalhada não ajudar a diagnosticar o problema, verifique também os logs do toolstack QEMU e do Xen™ em /var/log/xen. Observe que o nome do domínio é anexado ao nome do registro, portanto, se o domínio tiver o nome `freebsd`, você deverá encontrar um /var/log/xen/xl-freebsd.log e provavelmente um /var/log/xen/qemu-dm-freebsd.log. Ambos os arquivos de log podem conter informações úteis para a depuração. Se nada disso ajudar a resolver o problema, envie a descrição do problema que você está enfrentando e o máximo de informações possíveis para freebsd-xen@FreeBSD.org e xen-devel@lists.xenproject.org para obter ajuda.

Capítulo 22. Localização - Uso e Configuração do i18n/L10n

22.1. Sinopse

O FreeBSD é um projeto distribuído com usuários e colaboradores localizados em todo o mundo. Como tal, o FreeBSD suporta a localização em muitos idiomas, permitindo aos usuários visualizar, inserir ou processar dados em idiomas diferentes do inglês. Pode-se escolher entre a maioria dos principais idiomas, incluindo, mas não se limitando a: Chinês, Alemão, Japonês, Coreano, Francês, Russo e Vietnamita.

O termo internacionalização foi encurtado para i18n, que representa o número de letras entre a primeira e a última letra da `internacionalização`. L10n usa o mesmo esquema de nomes, mas a partir da `localização`. Os métodos, protocolos e aplicativos i18n/L10n permitem que os usuários usem os idiomas de sua escolha.

Este capítulo discute os recursos de internacionalização e localização do FreeBSD. Depois de ler este capítulo, você saberá:

- Como os nomes de localidade são construídos.
- Como definir a localidade para um login shell.
- Como configurar o console para idiomas diferentes do inglês.
- Como configurar o Xorg para diferentes idiomas.
- Como encontrar aplicativos compatíveis com i18n.
- Onde encontrar mais informações para configurar idiomas específicos.

Antes de ler este capítulo, você deve:

- Saber como instalar aplicativos adicionais de terceiros.

22.2. Usando Localização

As configurações de localização são baseadas em três componentes: o código do idioma, o código do país e a codificação. Nomes de localidade são construídos a partir dessas partes da seguinte maneira:

```
LanguageCode_CountryCode.Encoding
```

O *LanguageCode* e o *CountryCode* são usados para determinar o país e a variação de linguagem específica. A Idiomas Comum e Códigos de País apresenta alguns exemplos de *LanguageCode__CountryCode*:

Tabela 14. Idiomas Comum e Códigos de País

LanguageCode_Country Code	Descrição
en_US	Inglês, Estados Unidos
ru_RU	Russo, Rússia
zh_TW	Chinês Tradicional, Taiwan

Uma lista completa de localidades disponíveis pode ser encontrada digitando:

```
% locale -a | more
```

Para determinar a configuração atual de localidade:

```
% locale
```

Conjuntos de caracteres específicos de idioma, como ISO8859-1, ISO8859-15, KOI8-R e CP437, são descritos em multibyte(3). A lista ativa de conjuntos de caracteres pode ser encontrada no IANA Registry.

Alguns idiomas, como Chinês ou Japonês, não podem ser representados usando caracteres ASCII e requerem uma codificação de idioma estendida usando caracteres wide ou multibyte. Exemplos de codificações de wide ou multibyte incluem EUC e Big5. Aplicativos mais antigos podem confundir essas codificações com caracteres de controle, enquanto aplicativos mais novos geralmente reconhecem esses caracteres. Dependendo da implementação, os usuários podem ser obrigados a compilar um aplicativo com suporte a caracteres wide ou multibyte, ou configurá-lo corretamente.



O FreeBSD usa codificações de locale compatíveis com o Xorg.

O restante desta seção descreve os vários métodos para configurar a localidade em um sistema FreeBSD. A próxima seção discutirá as considerações para encontrar e compilar aplicativos com suporte a i18n.

22.2.1. Definindo a Localidade para o Login Shell

As configurações de localidade são configuradas no ~/.login_conf do usuário ou no arquivo de inicialização do shell do usuário: ~/.profile, ~/.bashrc, or ~/.cshrc.

Duas variáveis de ambiente devem ser definidas:

- LANG, que define o idioma *

  MM_CHARSET, que define o conjunto de caracteres MIME usado pelos aplicativos

Além da configuração do shell do usuário, essas variáveis também devem ser definidas para configurações específicas de aplicativos e configurações do Xorg.

Dois métodos estão disponíveis para fazer as atribuições de variáveis necessárias: o método classes de login, que é o método recomendado, e o método arquivo de inicialização. As próximas duas

seções demonstram como usar os dois métodos.

22.2.1.1. Método de Classes de Login

Este primeiro método é o método recomendado, pois atribui as variáveis de ambiente necessárias para o nome da localidade e os conjuntos de caracteres MIME para todos os shell possíveis. Essa configuração pode ser executada para cada usuário ou pode ser configurada para todos os usuários pelo superusuário.

Esse exemplo mínimo define as duas variáveis para a codificação Latin-1 no .login_conf do diretório inicial de um usuário individual:

```
me:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:
```

Aqui está um exemplo de ~/.login_conf de um usuário que define as variáveis para o Chinês Tradicional na codificação BIG-5. Mais variáveis são necessárias porque alguns aplicativos não respeitam corretamente variáveis de idioma para o Chinês, Japonês e Coreano:

```
#Users who do not wish to use monetary units or time formats
#of Taiwan can manually change each variable
me:\
    :lang=zh_TW.Big5:\

:setenv=LC_ALL=zh_TW.Big5,LC_COLLATE=zh_TW.Big5,LC_CTYPE=zh_TW.Big5,LC_MESSAGES=zh_TW.
Big5,LC_MONETARY=zh_TW.Big5,LC_NUMERIC=zh_TW.Big5,LC_TIME=zh_TW.Big5:\
    :charset=big5:\
    :xmodifiers="@im=gcin": #Set gcin as the XIM Input Server
```

Como alternativa, o superusuário pode configurar a localização para todos os usuários do sistema. As seguintes variáveis no /etc/login.conf são usadas para definir a localidade e o conjunto de caracteres MIME:

```
language_name|Account Type Description:\
    :charset=MIME_charset:\
    :lang=locale_name:\
    :tc=default:
```

Então, o exemplo anterior do Latin-1 ficaria assim:

```
german|German Users Accounts:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:\
    :tc=default:
```

Veja o login.conf(5) para mais detalhes sobre estas variáveis. Observe que ele já contém a classe *russian* predefinida.

Sempre que /etc/login.conf for editado, lembre-se de executar o seguinte comando para atualizar o banco de dados de recursos:

```
# cap_mkdb /etc/login.conf
```



Para um usuário final, o comando `cap_mkdb` vai precisar rodar no seu ~/.login_conf para que qualquer mudança tenha efeito.

22.2.1.1.1. Utilitários que Alteram as Classes de Login

Além de editar manualmente o /etc/login.conf, vários utilitários estão disponíveis para definir a localidade de usuários recém-criados.

Ao usar o `vipw` para adicionar novos usuários, especifique o *idioma* para definir a localidade:

```
user:password:1111:11:language:0:0:User Name:/home/user:/bin/sh
```

Ao usar o `adduser` para adicionar novos usuários, o idioma padrão pode ser pré-configurado para todos os novos usuários ou especificado para um usuário individual.

Se todos os novos usuários usarem o mesmo idioma, configure `defaultclass=language` em /etc/adduser.conf.

Para substituir essa configuração ao criar um usuário, insira a localidade necessária neste prompt:

```
Enter login class: default []:
```

ou especifique a localidade ao executar o `adduser`:

```
# adduser -class language
```

Se o `pw` for usado para adicionar novos usuários, especifique a localidade da seguinte forma:

```
# pw useradd user_name -L language
```

Para alterar a classe de login de um usuário existente, `chpass` pode ser usado. Execute-o como superusuário e forneça o nome do usuário para edição como argumento.

```
# chpass user_name
```

22.2.1.2. Método de Arquivo de Inicialização do Shell

Esse segundo método não é recomendado, pois cada shell usado requer configuração manual, e cada shell tem um arquivo de configuração diferente e uma sintaxe diferente. Como exemplo, para definir o idioma Alemão para o shell `sh`, essas linhas podem ser adicionadas ao ~/.profile para definir o shell apenas para esse usuário. Essas linhas também podem ser adicionadas ao /etc/profile ou /usr/shared/skel/dot.profile para definir esse shell para todos os usuários:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

No entanto, o nome do arquivo de configuração e a sintaxe usada são diferentes para o shell `csh`. Estas são as configurações equivalentes para o ~/.csh.login, /etc/csh.login, ou /usr/shared/skel/dot.login:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Para complicar, a sintaxe necessária para configurar o Xorg no ~/.xinitrc também depende do shell. O primeiro exemplo é para o shell `sh` e o segundo é para o shell `csh`:

```
LANG=de_DE.ISO8859-1; export LANG
```

```
setenv LANG de_DE.ISO8859-1
```

22.2.2. Configuração do Console

Várias fontes de localização estão disponíveis para o console. Para ver uma lista de fontes disponíveis, digite `ls /usr/shared/syscons/fonts`. Para configurar a fonte do console, especifique o *font_name*, sem o sufixo .fnt, em /etc/rc.conf:

```
font8x16=font_name
font8x14=font_name
font8x8=font_name
```

O keymap e o screenmap podem ser definidos adicionando o seguinte ao /etc/rc.conf:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

Para ver a lista de screenmaps disponíveis, digite `ls /usr/shared/syscons/scrnmaps`. Não inclua o sufixo .scm ao especificar *screenmap_name*. Um screenmap com uma fonte mapeada

correspondente geralmente é necessário como uma solução alternativa para expandir o bit 8 para o 9 na matriz de caracteres de fonte de um adaptador VGA para que as letras sejam movidas para fora da área de pseudo-grafia se a fonte da tela usar uma coluna de 8 bits.

Para ver a lista de mapas de teclado disponíveis, digite `ls /usr/shared/syscons/keymaps`. Ao especificar o *keymap_name*, não inclua o sufixo .kbd. Para testar os mapas de teclado sem reinicializar o sistema, use kbdmap(1).

A entrada `keychange` geralmente é necessária para programar as teclas de função para corresponder ao tipo de terminal selecionado, porque as sequências de teclas de função não podem ser definidas no mapa de teclas.

Em seguida, defina o tipo de terminal do console correto em /etc/ttys para todas as entradas do terminal virtual. Tipos de Terminal Definidos para Conjuntos de Caracteres resume os tipos de terminais disponíveis:

Tabela 15. Tipos de Terminal Definidos para Conjuntos de Caracteres

Conjunto de Caracteres	Tipo de Terminal
ISO8859-1 ou ISO8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA padrão)	`cons25`
US-ASCII	`cons25w`

Para idiomas com caracteres wide ou multibyte, instale um console para esse idioma a partir da Coleção de Ports do FreeBSD. Os ports disponíveis estão resumidos em Consoles Disponíveis pela Coleção de Ports. Uma vez instalado, consulte o pkg-message dos ports ou as páginas de manual para instruções de configuração e uso.

Tabela 16. Consoles Disponíveis pela Coleção de Ports

Idioma	Localização do Port
Chinês Tradicional (BIG-5)	chinese/big5con
Chinês/Japonês/Coreano	chinese/cce
Chinês/Japonês/Coreano	chinese/zhcon
Japonês	chinese/kon2
Japonês	japanese/kon2-14dot
Japonês	japanese/kon2-16dot

Se o moused estiver ativado no /etc/rc.conf, uma configuração adicional pode ser necessária. Por padrão, o cursor do mouse do driver syscons(4) ocupa o intervalo `0xd0-0xd3` no conjunto de caracteres. Se o idioma usar esse intervalo, mova o intervalo do cursor adicionando a seguinte

linha ao /etc/rc.conf:

```
mousechar_start=3
```

22.2.3. Configuração do Xorg

O O sistema X Window descreve como instalar e configurar o Xorg. Ao configurar localizações no Xorg, fontes adicionais e métodos de entrada estão disponíveis na Coleção de Ports do FreeBSD. Configurações específicas de i18n para aplicações como fontes e menus podem ser tunadas em ~/.Xresources e devem permitir que os usuários visualizem o idioma selecionado nos menus das aplicações gráficas.

O protocolo X Input Method (XIM) é um padrão Xorg para inserir caracteres não Ingleses. Métodos de Entrada Disponíveis resume os métodos de entrada de aplicações que estão disponíveis na Coleção de Ports do FreeBSD. Aplicativos adicionais Fcitx e Uim também estão disponíveis.

Tabela 17. Métodos de Entrada Disponíveis

Idioma	Método de Entrada
Chinês	chinese/gcin
Chinês	chinese/ibus-chewing
Chinês	chinese/ibus-pinyin
Chinês	chinese/oxim
Chinês	chinese/scim-fcitx
Chinês	chinese/scim-pinyin
Chinês	chinese/scim-tables
Japonês	japanese/ibus-anthy
Japonês	japanese/ibus-mozc
Japonês	japanese/ibus-skk
Japonês	japanese/im-ja
Japonês	japanese/kinput2
Japonês	japanese/scim-anthy
Japonês	japanese/scim-canna
Japonês	japanese/scim-honoka
Japonês	japanese/scim-honoka-plugin-romkan
Japonês	japanese/scim-honoka-plugin-wnn
Japonês	japanese/scim-prime
Japonês	japanese/scim-skk
Japonês	japanese/scim-tables

Idioma	Método de Entrada
Japonês	japanese/scim-tomoe
Japonês	japanese/scim-uim
Japonês	japanese/skkinput
Japonês	japanese/skkinput3
Japonês	japanese/uim-anthy
Coreano	korean/ibus-hangul
Coreano	korean/imhangul
Coreano	korean/nabi
Coreano	korean/scim-hangul
Coreano	korean/scim-tables
Vietnamita	vietnamese/xvnkb
Vietnamita	vietnamese/x-unikey

22.3. Encontrando Aplicações i18n

Aplicações i18n são programadas usando kits i18n em bibliotecas. Isso permite que os desenvolvedores escrevam um arquivo simples e traduzam menus e textos exibidos para cada idioma.

A Coleção de Ports do FreeBSD contém muitos aplicativos com suporte embutido para caracteres wide ou multibyte para vários idiomas. Tais aplicativos incluem `i18n` em seus nomes para fácil identificação. No entanto, eles nem sempre suportam o idioma necessário.

Alguns aplicativos podem ser compilados com o conjunto de caracteres específico. Isso geralmente é feito no Makefile do port ou passando um parâmetro para o configure. Consulte a documentação i18n no código fonte do respectivo port do FreeBSD para obter mais informações sobre como determinar o parâmetro do configure necessário ou o Makefile do port para determinar quais opções de compilação para usar ao compilar o port.

22.4. Configuração de Localização para Idiomas Específicos

Esta seção fornece exemplos de configuração para definir a localização de um sistema FreeBSD para o idioma Russo. Em seguida, ele fornece alguns recursos adicionais para definir a localização com outros idiomas.

22.4.1. Idioma Russo (Codificação KOI8-R)

Esta seção mostra as configurações específicas necessárias para definir a localização de um sistema FreeBSD para o idioma Russo. Consulte Usando Localização para obter uma descrição mais completa de cada tipo de configuração.

Para definir esta localidade para o login shell, adicione as seguintes linhas ao ~/.login_conf de cada usuário:

```
me:My Account:\
    :charset=KOI8-R:\
    :lang=ru_RU.KOI8-R:
```

Para configurar o console, adicione as seguintes linhas ao /etc/rc.conf:

```
keymap="ru.utf-8"
scrnmap="utf-82cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
mousechar_start=3
```

Para cada entrada `ttyv` em /etc/ttys, use `cons25r` como o tipo de terminal.

Para configurar a impressão, é necessário um filtro de saída especial para converter de KOI8-R para CP866, pois a maioria das impressoras com caracteres Russos vem com a página de código de hardware CP866. O FreeBSD inclui um filtro padrão para este propósito, /usr/libexec/lpr/ru/koi2alt. Para usar este filtro, adicione esta entrada ao /etc/printcap:

```
lp|Russian local line printer:\
    :sh:of=/usr/libexec/lpr/ru/koi2alt:\
    :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Consulte printcap(5) para obter uma explicação mais detalhada.

Para configurar o suporte a nomes de arquivos Russos em sistemas de arquivos montados do MS-DOS™, inclua `-L` e o nome da localidade ao adicionar uma entrada ao /etc/fstab:

```
/dev/ad0s2      /dos/c  msdos   rw,-Lru_RU.KOI8-R 0 0
```

Consulte mount_msdosfs(8) para mais detalhes.

Para configurar fontes Russas no Xorg, instale o pacote x11-fonts/xorg-fonts-cyrillic. Em seguida, verifique a seção `"Files"` em /etc/X11/xorg.conf. A seguinte linha deve ser adicionada *antes* de qualquer outra entrada `FontPath`:

```
FontPath   "/usr/local/lib/X11/fonts/cyrillic"
```

Fontes Cirílicos adicionais estão disponíveis na Coleção de Ports.

Para ativar um teclado Russo, adicione o seguinte à seção `"Keyboard"` do /etc/xorg.conf:

```
Option "XkbLayout"   "us,ru"
Option "XkbOptions"  "grp:toggle"
```

Certifique-se de que `XkbDisable` esteja comentado nesse arquivo.

Para `grp:toggle` use `Right Alt`, para `grp:ctrl_shift_toggle` use `Ctrl` + `Shift`. Para `grp:caps_toggle` use `CapsLock`. A antiga função `CapsLock` ainda está disponível no modo LAT apenas usando `Shift` + `CapsLock`. `grp:caps_toggle` não funciona no Xorg por alguma razão desconhecida.

Se o teclado tiver as teclas "Windows™" e algumas teclas não alfabéticas mapeadas incorretamente, adicione a seguinte linha ao /etc/xorg.conf:

```
Option "XkbVariant" ",winkeys"
```



O teclado Russo XKB pode não funcionar com aplicativos não localizados. Aplicativos minimamente localizados devem chamar uma função `XtSetLanguageProc (NULL, NULL, NULL);` no início do programa.

Veja http://koi8.pp.ru/xwin.html para mais instruções sobre como definir a localização em aplicações Xorg. Para mais informações gerais sobre a codificação KOI8-R, consulte http://koi8.pp.ru/.

22.4.2. Recursos Específicos de Idioma Adicionais

Esta seção lista alguns recursos adicionais para a configuração de outras localidades.

Chinês Tradicional para Taiwan

O projeto FreeBSD-Taiwan tem um HOWTO em Chinês para o FreeBSD em http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/.

Localização do Idioma Grego

Um artigo completo sobre o suporte Grego no FreeBSD está disponível aqui, somente em Grego, como parte da documentação oficial do FreeBSD em Grego.

Localização do Idioma Japonês e Coreano

Para Japonês, consulte http://www.jp.FreeBSD.org/ e, para Coreano, consulte http://www.kr.FreeBSD.org/.

Documentação do FreeBSD em Outros Idiomas

Alguns colaboradores do FreeBSD traduziram partes da documentação do FreeBSD para outros idiomas. Elas estão disponíveis através de links no site do FreeBSD ou em /usr/shared/doc.

Capítulo 23. Atualização e Upgrade do FreeBSD

23.1. Sinopse

O FreeBSD está em constante desenvolvimento entre os releases. Algumas pessoas preferem usar as versões lançadas oficialmente, enquanto outras preferem se manter em sincronia com os últimos desenvolvimentos. No entanto, até mesmo versões oficiais são atualizadas com patches de segurança e outras correções críticas. Independentemente da versão usada, o FreeBSD fornece todas as ferramentas necessárias para manter o sistema atualizado e permite atualizações fáceis entre as versões. Este capítulo descreve como acompanhar o sistema de desenvolvimento e o uso das ferramentas básicas para manter um sistema FreeBSD atualizado.

Depois de ler este capítulo, você saberá:

- Como manter um sistema FreeBSD atualizado com o freebsd-update ou com o Subversion.
- Como comparar o estado de um sistema instalado com uma cópia original.
- Como manter a documentação instalada atualizada com o Subversion ou com o port da documentação.
- A diferença entre os dois ramos de desenvolvimento: FreeBSD-STABLE e FreeBSD-CURRENT.
- Como recompilar e reinstalar todo o sistema básico.

Antes de ler este capítulo, você deve:

- Configurar corretamente a conexão de rede (Rede Avançada).
- Saber como instalar software adicional de terceiros (Instalando Aplicativos. Pacotes e Ports).



Ao longo deste capítulo, o `svnlite` é usado para obter e atualizar o código fonte do FreeBSD. Opcionalmente, o port ou pacote devel/subversion pode ser usado.

23.2. Atualização do FreeBSD

A aplicação de patches de segurança em tempo hábil e a atualização para uma versão mais recente de um sistema operacional são aspectos importantes da administração contínua do sistema. O FreeBSD inclui um utilitário chamado `freebsd-update` o qual pode ser usado para executar ambas as tarefas.

Este utilitário suporta atualizações binárias de segurança e de erratas para o FreeBSD, sem a necessidade de compilar e instalar manualmente o patch ou um novo kernel. Atualizações binárias estão disponíveis para todas as arquiteturas e versões atualmente suportadas pela equipe de segurança. A lista de versões suportadas e suas datas estimadas de fim de vida estão listadas em https://www.FreeBSD.org/security/.

Este utilitário também suporta upgrades do sistema operacional para releases menores (ponto x), bem como atualizações para outro ramo de release. Antes de atualizar para uma nova versão,

revise o seu anúncio de lançamento, pois ele contém informações importantes pertinentes ao release. Os anúncios de lançamento estão disponíveis em https://www.FreeBSD.org/releases/.



Se um `crontab` utilizando os recursos do freebsd-update(8) existir, ele deve ser desativado antes de atualizar o sistema operacional .

Esta seção descreve o arquivo de configuração usado pelo `freebsd-update`, demonstra como aplicar um patch de segurança e como atualizar para um release menor ou principal do sistema operacional e discute algumas das considerações ao atualizar o sistema operacional .

23.2.1. O Arquivo de Configuração

O arquivo de configuração padrão do `freebsd-update` funciona como está. Alguns usuários podem querer ajustar a configuração padrão no /etc/freebsd-update.conf, permitindo um melhor controle do processo. Os comentários neste arquivo explicam as opções disponíveis, mas os seguintes podem exigir um pouco mais de explicação:

```
# Componentes do sistema base que devem ser mantidos atualizados.
Components world kernel
```

Este parâmetro controla quais partes do FreeBSD serão mantidas atualizadas. O padrão é atualizar todo o sistema básico e o kernel. Componentes individuais podem ser especificados, como `src/base` ou `src/sys`. No entanto, a melhor opção é deixar isso no padrão, pois alterá-lo para incluir itens específicos requer que todos os itens necessários sejam listados. Com o tempo, isso pode ter consequências desastrosas, pois o código-fonte e os binários podem ficar fora de sincronia.

```
# Caminhos que começam com qualquer coisa que corresponda a uma entrada em uma
# declaração IgnorePaths será ignorada.
IgnorePaths /boot/kernel/linker.hints
```

Para deixar diretórios especificados, como /bin ou /sbin, intocados durante o processo de atualização, adicione seus caminhos a esta instrução. Esta opção pode ser usada para evitar que o `freebsd-update` substitua as modificações locais.

```
# Caminhos que começam com qualquer coisa que corresponda a uma entrada em uma
declaração
# UpdateIfUnmodified só será atualizada se o conteúdo do arquivo não tiver sido
# modificado pelo usuário (a menos que as alterações sejam mescladas; veja abaixo).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

Esta opção atualizará apenas os arquivos de configuração não modificados nos diretórios especificados. Quaisquer alterações feitas pelo usuário impedirão a atualização automática desses arquivos. Existe outra opção, `KeepModifiedMetadata`, que instruirá o `freebsd-update` para salvar as alterações durante a mesclagem.

```
# Ao fazer o upgrade para uma nova versão do FreeBSD, os arquivos que forem
especificados no MergeChanges
# terão quaisquer alterações locais mescladas na versão da nova release.
MergeChanges /etc/ /var/named/etc/ /boot/device.hints
```

Lista de diretórios com arquivos de configuração que o `freebsd-update` deve tentar mesclar. O processo de mesclagem de arquivos é uma série de patches diff(1) semelhantes a mergemaster(8), mas com menos opções. As mesclagens são aceitas, abrem um editor ou fazem com que o `freebsd-update` aborte. Em caso de dúvida, faça backup do /etc e apenas aceite as mesclagens. Veja mergemaster(8) para maiores informações sobre o `mergemaster`.

```
# Diretório no qual armazenar atualizações baixadas e arquivos
# temporários usados pelo FreeBSD Update.
# WorkDir /var/db/freebsd-update
```

Este diretório é onde todos os patches e arquivos temporários são colocados. Nos casos em que o usuário estiver fazendo uma atualização de versão, esse local deverá ter pelo menos um gigabyte de espaço em disco disponível.

```
# Ao atualizar entre releases, a lista de Componentes deve ser lida de forma estrita
(StrictComponents yes)
# ou meramente como uma lista de componentes que *podem* ser instalados de quais
atualizações do
# FreeBSD devem ser instaladas e atualizadas (StrictComponents no)?
# StrictComponents no
```

Quando esta opção estiver definida como `yes`, o `freebsd-update` assumirá que a lista `Componentes` está completa e não tentará fazer alterações fora da lista. Efetivamente, o `freebsd-update` tentará atualizar todos os arquivos que pertencem à lista `Componentes`.

23.2.2. Aplicando Patches de Segurança

O processo de aplicação de patches de segurança do FreeBSD foi simplificado, permitindo que um administrador mantenha um sistema totalmente corrigido usando o `freebsd-update`. Maiores informações sobre os avisos de segurança do FreeBSD podem ser encontradas em Avisos de Segurança do FreeBSD.

Patches de segurança do FreeBSD podem ser baixados e instalados usando os seguintes comandos. O primeiro comando determinará se algum patch pendente está disponível e, em caso afirmativo, listará os arquivos que serão modificados se os patches forem aplicados. O segundo comando aplicará os patches.

```
# freebsd-update fetch
# freebsd-update install
```

Se a atualização aplicar alguma correção de kernel, o sistema precisará de uma reinicialização para inicializar no kernel corrigido. Se o patch for aplicado a qualquer binário em execução, os aplicativos afetados devem ser reiniciados para que a versão corrigida do binário seja usada.



Normalmente, o usuário precisa estar preparado para reiniciar o sistema. Para saber se uma reinicialização é necessária por uma atualização do kernel, execute os comandos `freebsd-version -k` e `uname -r` e se eles forem diferentes, é necessário reiniciar.

O sistema pode ser configurado para verificar automaticamente as atualizações uma vez por dia, adicionando esta entrada ao /etc/crontab:

```
@daily                                  root    freebsd-update cron
```

Se houver patches, eles serão automaticamente baixados, mas não serão aplicados. O usuário `root` receberá um email para que os patches possam ser revisados e instalados manualmente com o `freebsd-update install`.

Se algo der errado, o `freebsd-update` terá a capacidade de reverter o último conjunto de alterações com o seguinte comando:

```
# freebsd-update rollback
Uninstalling updates... done.
```

Novamente, o sistema deve ser reiniciado se o kernel ou qualquer módulo do kernel for modificado e quaisquer binários afetados devem ser reiniciados.

Apenas o kernel GENERIC pode ser atualizado automaticamente pelo `freebsd-update`. Se um kernel personalizado estiver instalado, ele terá que ser recompilado e reinstalado depois que o `freebsd-update` terminar de instalar as atualizações. No entanto, o `freebsd-update` detectará e atualizará o kernel *GENERIC* se /boot/GENERIC existir, mesmo que não seja o kernel atual em execução no sistema. Para verificar detalhes desta instalação utilize o comando uname(1).



Sempre mantenha uma cópia do kernel GENERIC em /boot/GENERIC. Será útil no diagnóstico de vários problemas e na execução de atualizações de versão. Consulte Kernels personalizados com o FreeBSD 9.X e posteriores para obter instruções sobre como obter uma cópia do kernel GENERIC.

A menos que a configuração padrão em /etc/freebsd-update.conf tenha sido alterada, o `freebsd-update` instalará o código fonte atualizado do kernel juntamente com o restante das atualizações. O processo de recompilação e reinstalação de um novo kernel personalizado poderá ser executado da maneira usual.

As atualizações distribuídas pelo `freebsd-update` nem sempre envolvem o kernel. Não é necessário recompilar um kernel personalizado se o código fonte do kernel não tiverem sido modificado pelo `freebsd-update install`. No entanto, o `freebsd-update` sempre atualizará o /usr/src/sys/conf/newvers.sh. O nível de patch atual, conforme indicado pelo número `-p` relatado

pelo `uname -r`, é obtido desse arquivo. Recompilar um kernel personalizado, mesmo que nada mais tenha sido alterado, permite que o `uname` relate com precisão o nível de patch atual do sistema. Isso é particularmente útil ao manter vários sistemas, pois permite uma avaliação rápida das atualizações instaladas em cada um deles.

23.2.3. Realizando Upgrades de Versão Principais e Menores

Atualizações de uma versão menor do FreeBSD para outra, como do FreeBSD 9.0 para o FreeBSD 9.1, são chamadas de upgrades de *versão menor*. Atualizações de *versões principais* ocorrem quando o FreeBSD é atualizado de uma versão principal para outra, como do FreeBSD 9.X para o FreeBSD 10.X. Ambos os tipos de atualizações podem ser executados fornecendo um target de versão de release para o `freebsd-update`.



Se o sistema estiver executando um kernel personalizado, certifique-se de que uma cópia do kernel GENERIC exista em /boot/GENERIC antes de iniciar o upgrade. Consulte Kernels personalizados com o FreeBSD 9.X e posteriores para obter instruções sobre como obter uma cópia do kernel GENERIC.

O seguinte comando, quando executado em um sistema FreeBSD 9.0, irá atualizá-lo para o FreeBSD 9.1:

```
# freebsd-update -r 9.1-RELEASE upgrade
```

Depois que o comando for recebido, o `freebsd-update` avaliará o arquivo de configuração e o sistema atual na tentativa de reunir as informações necessárias para executar a atualização. Uma listagem de tela exibirá quais componentes foram e quais não foram detectados. Por exemplo:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 9.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
world/proflibs

Does this look reasonable (y/n)? y
```

Neste ponto, o `freebsd-update` tentará baixar todos os arquivos necessários para a atualização. Em alguns casos, o usuário pode ser questionado sobre o que instalar ou como proceder.

Ao usar um kernel personalizado, a etapa acima produzirá um aviso semelhante ao seguinte:

```
WARNING: This system is running a "MYKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 9.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

Este aviso pode ser ignorado com segurança neste momento. O kernel GENERIC atualizado será usado como uma etapa intermediária no processo de atualização.

Depois que todos os patches tiverem sido baixados para o sistema local, eles serão aplicados. Esse processo pode demorar um pouco, dependendo da velocidade e da carga de trabalho da máquina. Os arquivos de configuração serão então mesclados. O processo de mesclagem requer alguma intervenção do usuário, pois um arquivo pode ser mesclado ou um editor pode aparecer na tela para uma mesclagem manual. Os resultados de cada mesclagem bem-sucedida serão mostrados para o usuário enquanto o processo continua. Um merge falho ou ignorado fará com que o processo seja abortado. Os usuários podem desejar fazer um backup de /etc e mesclar manualmente os arquivos importantes, como o master.passwd ou o group posteriormente.



O sistema não está sendo alterado, já que todos os patches e merges estão acontecendo em outro diretório. Uma vez que todas as correções tenham sido aplicadas com sucesso, e todos os arquivos de configuração foram mesclados e tudo indicar que o processo ocorrerá sem problemas, as alterações poderão ser confirmadas pelo usuário usando o seguinte comando:

```
# freebsd-update install
```

O kernel e os módulos do kernel serão atualizados primeiro. Se o sistema estiver sendo executado com um kernel personalizado, use o nextboot(8) para definir que o kernel para a próxima inicialização será o /boot/GENERIC:

```
# nextboot -k GENERIC
```



Antes de reinicializar com o kernel GENERIC, verifique se ele contém todos os drivers necessários para o sistema inicializar corretamente e se conectar à rede, se a máquina que está sendo atualizada for acessada remotamente. Em particular, se o kernel customizado em execução contiver funcionalidades internas normalmente fornecidas pelos módulos do kernel, certifique-se de carregar temporariamente estes módulos no kernel GENERIC usando o /boot/loader.conf. Recomenda-se desabilitar os serviços não essenciais, bem como todas as montagens de disco e de rede, até que o processo de atualização seja concluído.

A máquina agora deve ser reiniciada com o kernel atualizado:

```
# shutdown -r now
```

Quando o sistema estiver on-line, reinicie o `freebsd-update` usando o comando a seguir. Como o estado do processo foi salvo, o `freebsd-update` não será iniciado desde o início, mas passará para a próxima fase e removerá todas as bibliotecas compartilhadas e os arquivos de objetos antigos.

```
# freebsd-update install
```



Dependendo se os números de versão de uma biblioteca foram incrementados ou não, pode haver apenas duas fases de instalação em vez de três.

A atualização está completa agora. Se esta for uma atualização de versão principal, reinstale todas os ports e pacotes conforme descrito em Atualizando pacotes após atualizar para uma versão principal (Major Release).

23.2.3.1. Kernels personalizados com o FreeBSD 9.X e posteriores

Antes de usar o `freebsd-update`, assegure-se de que uma cópia do kernel GENERIC exista em /boot/GENERIC. Se um kernel personalizado foi compilado apenas uma vez, o kernel em /boot/kernel.old é o kernel `GENERIC`. Simplesmente renomeie este diretório para /boot/GENERIC.

Se um kernel personalizado foi compilado mais de uma vez ou se é desconhecido quantas vezes o kernel personalizado foi compilado, obtenha uma cópia do kernel `GENERIC` que corresponda à versão atual do sistema operacional. Se o acesso físico ao sistema estiver disponível, uma cópia do kernel `GENERIC` pode ser instalada a partir da mídia de instalação:

```
# mount /cdrom
# cd /cdrom/usr/freebsd-dist
# tar -C/ -xvf kernel.txz boot/kernel/kernel
```

Como alternativa, o kernel `GENERIC` pode ser recriado e instalado a partir da do código fonte:

```
# cd /usr/src
# make kernel __MAKE_CONF=/dev/null SRCCONF=/dev/null
```

Para que este kernel seja identificado como o kernel `GENERIC` pelo `freebsd-update`, o arquivo de configuração GENERIC não deve ter sido modificado de forma alguma. Também é sugerido que o kernel seja compilado sem outras opções especiais.

A reinicialização no kernel GENERIC não é necessária, pois o `freebsd-update` só precisa que o /boot/GENERIC exista.

23.2.3.2. Atualizando pacotes após atualizar para uma versão principal (Major Release)

Geralmente, os aplicativos instalados continuarão funcionando sem problemas após atualizações

de versões menores. As versões principais usam diferentes interfaces binárias de aplicativos (ABIs), que quebram a maioria dos aplicativos de terceiros. Após uma atualização de versão principal, todos os pacotes e ports instalados precisam ser atualizados. Pacotes podem ser atualizados usando `pkg upgrade`. Para atualizar os ports instalados, use um utilitário como o ports-mgmt/portmaster.

Uma atualização forçada de todos os pacotes instalados substituirá os pacotes por novas versões a partir do repositório, mesmo que o número da versão não tenha aumentado. Isso é necessário por causa da alteração da versão do ABI que ocorre ao atualizar entre versões principais do FreeBSD. A atualização forçada pode ser realizada executando:

```
# pkg-static upgrade -f
```

Uma recompilação de todos os aplicativos instalados pode ser realizada com este comando:

```
# portmaster -af
```

Este comando exibirá as telas de configuração de cada aplicativo que possui opções configuráveis e aguardará que o usuário interaja com estas telas. Para evitar esse comportamento e usar apenas as opções padrões, inclua `-G` no comando acima.

Quando as atualizações de software estiverem concluídas, conclua o processo de atualização com uma chamada final para o `freebsd-update` para amarrar todas as pontas soltas no processo de atualização:

```
# freebsd-update install
```

Se o kernel GENERIC foi usado temporariamente, este é o momento de construir e instalar um novo kernel personalizado usando as instruções do Configurando o kernel do FreeBSD.

Reinicialize a máquina na nova versão do FreeBSD. O processo de atualização está concluído agora.

23.2.4. Comparação do estado do sistema

O estado da versão instalada do FreeBSD em relação a uma boa cópia conhecida pode ser testado usando o `freebsd-update IDS`. Este comando avalia a versão atual dos utilitários do sistema, bibliotecas e arquivos de configuração e pode ser usado como um Sistema de Detecção de Intrusão embutido (IDS).



Este comando não é um substituto para um IDS real como o security/snort. Como o `freebsd-update` armazena dados no disco, a possibilidade de adulteração é evidente. Embora esta possibilidade possa ser reduzida usando o `kern.securelevel` e armazenando os dados do `freebsd-update` em um sistema de arquivos read-only quando não estiver em uso, uma solução melhor seria comparar o sistema com um disco seguro, como um DVD ou dispositivo de disco externo USB armazenado em segurança. Um método alternativo para fornecer a funcionalidade de IDS usando um utilitário interno é descrito em Verificação Binária

Para começar a comparação, especifique um arquivo de saída para salvar os resultados:

```
# freebsd-update IDS >> outfile.ids
```

O sistema agora será inspecionado e uma longa lista de arquivos, junto com os valores de hash SHA256 tanto para o valor conhecido na release e como na instalação atual, será enviada para o arquivo de saída especificado.

As entradas na listagem são extremamente longas, mas o formato de saída pode ser facilmente analisado. Por exemplo, para obter uma lista de todos os arquivos que diferem daqueles na release, execute o seguinte comando:

```
# cat outfile.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

Este exemplo de saída foi truncado, pois existem muito mais arquivos. Alguns arquivos possuem modificações naturais. Por exemplo, o /etc/passwd será modificado se usuários tiverem sido adicionados ao sistema. Módulos de kernel podem diferir pois o `freebsd-update` pode tê-los atualizado. Para excluir arquivos ou diretórios específicos, adicione-os à opção `IDSIgnorePaths` em /etc/freebsd-update.conf.

23.3. Atualizando o Conjunto de Documentação

A documentação é parte integrante do sistema operacional FreeBSD. Enquanto uma versão atualizada da documentação do FreeBSD está sempre disponível no site do FreeBSD (https://www.freebsd.org/doc/), pode ser útil ter uma cópia local atualizada do site do FreeBSD, manuais, FAQ e artigos.

Esta seção descreve como usar os fontes ou a Coleção de Ports do FreeBSD para manter uma cópia local da documentação do FreeBSD atualizada.

Para obter informações sobre como editar e enviar correções para a documentação, consulte o Primer do Projeto de Documentação do FreeBSD para Novos Colaboradores (Primer do Projeto de Documentação do FreeBSD).

23.3.1. Atualizando a documentação a partir do código-fonte

Recompilar a documentação do FreeBSD a partir do código-fonte requer uma coleção de ferramentas que não fazem parte do sistema básico do FreeBSD. As ferramentas necessárias podem ser instaladas a partir do pacote textproc/docproj ou do port desenvolvido pelo Projeto de Documentação do FreeBSD.

Uma vez instalado, use o svnlite para buscar uma cópia limpa dos fontes da documentação:

```
# svnlite checkout https://svn.FreeBSD.org/doc/head /usr/doc
```

O download inicial dos fontes da documentação pode demorar um pouco. Deixe executar até completar.

Futuras atualizações dos fontes da documentação podem ser obtidas executando:

```
# svnlite update /usr/doc
```

Depois que um snapshot atualizado dos fontes da documentação for obtido e disponibilizado em /usr/doc, tudo estará pronto para uma atualização da documentação instalada.

Uma atualização completa de todos os idiomas disponíveis pode ser realizada digitando:

```
# cd /usr/doc
# make install clean
```

Se uma atualização de apenas um idioma específico for desejada, o `make` pode ser executado em um subdiretório específico de idioma do /usr/doc:

```
# cd /usr/doc/en_US.ISO8859-1
# make install clean
```

Uma maneira alternativa de atualizar a documentação é executar este comando a partir do /usr/doc ou do subdiretório específico do idioma desejado:

```
# make update
```

Os formatos de saída que serão instalados podem ser especificados definindo o parâmetro `FORMATS`:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

Várias opções estão disponíveis para facilitar o processo de atualização de apenas partes da documentação ou a construção de traduções específicas. Estas opções podem ser configuradas como opções de todo o sistema no /etc/make.conf, ou como opções de linha de comando passadas para o `make`.

As opções incluem:

`DOC_LANG`
: A lista de idiomas e codificações para compilar e instalar, como `en_US.ISO8859-1` para documentação em inglês.

`FORMATS`

Um formato único ou uma lista de formatos de saída a serem criados. Atualmente os formatos suportados são, `html`, `html-split`, `txt`, `ps`, e `pdf`.

`DOCDIR`

Onde instalar a documentação. O padrão é /usr/shared/doc.

Para mais variáveis do `make` suportadas como opções system-wide no FreeBSD, consulte make.conf(5).

23.3.2. Atualizando a documentação a partir do ports

A seção anterior apresentou um método para atualizar a documentação do FreeBSD a partir do código fonte. Esta seção descreve um método alternativo que usa a Coleção de Ports e possibilita:

- Instalar pacotes pré-compilados da documentação, sem precisar compilar nada localmente ou instalar o conjunto de ferramentas de documentação.
- Compilar o código fonte da documentação por meio do framework de ports, facilitando o check-out e as etapas de compilação.

Este método de atualização da documentação do FreeBSD é suportado por um conjunto de ports e pacotes de documentação que são atualizados mensalmente pela Equipe de Engenharia da Documentação doceng@FreeBSD.org. Eles estão listados na Coleção de Ports do FreeBSD, na categoria docs (http://www.freshports.org/docs/).

A organização dos ports de documentação é a seguinte:

- O pacote ou port misc/freebsd-doc-en instala toda a documentação em inglês.
- O meta-pacote ou port do pacote misc/freebsd-doc-all instala toda a documentação em todos os idiomas disponíveis.
- Existe um pacote e um port para cada tradução, como misc/freebsd-doc-hu para a documentação húngara.

Quando pacotes binários são usados, a documentação do FreeBSD será instalada em todos os formatos disponíveis para o idioma especificado. Por exemplo, o comando a seguir instalará o pacote mais recente da documentação em húngaro:

```
# pkg install hu-freebsd-doc
```



Os pacotes usam um formato que difere do nome do port correspondente: `lang-freebsd-doc`, onde *lang* é o formato abreviado do código de idioma, como `hu` para húngaro, ou `zh_cn` para chinês simplificado.

Para especificar o formato da documentação, compile o port em vez de instalar o pacote. Por exemplo, para compilar e instalar a documentação em inglês:

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

O port fornece um menu de configuração no qual o formato para compilar e instalar pode ser especificado. Por padrão, o HTML dividido, semelhante ao formato usado em http://www.FreeBSD.org e o PDF estão selecionados.

Alternativamente, várias opções `make` podem ser especificadas ao compilar um port de documentação, incluindo:

`WITH_HTML`
: Cria o formato HTML com um único arquivo HTML por documento. A documentação formatada é salva em um arquivo chamado article.html ou book.html.

`WITH_PDF`
: A documentação formatada é salva em um arquivo chamado article.pd ou book.pdf.

`DOCBASE`
: Especifica onde instalar a documentação. O padrão é /usr/local/shared/doc/freebsd.

Este exemplo usa variáveis para instalar a documentação húngara como um arquivo PDF no diretório especificado:

```
# cd /usr/ports/misc/freebsd-doc-hu
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

Pacotes ou ports de documentação podem ser atualizados usando as instruções em Instalando Aplicativos. Pacotes e Ports. Por exemplo, o seguinte comando atualiza a documentação húngara instalada usando ports-mgmt/portmaster através do uso apenas de pacotes:

```
# portmaster -PP hu-freebsd-doc
```

23.4. Acompanhando um ramo de desenvolvimento

O FreeBSD possui duas ramificações de desenvolvimento: FreeBSD-CURRENT e FreeBSD-STABLE.

Esta seção fornece uma explicação sobre cada ramo e seu público-alvo, bem como manter um sistema atualizado com cada ramo respectivo.

23.4.1. Usando o FreeBSD-CURRENT

O FreeBSD-CURRENT é o desenvolvimento "bleeding edge" do FreeBSD e espera-se que os usuários do FreeBSD-CURRENT tenham um alto grau de habilidade técnica. Usuários menos técnicos que desejam acompanhar um ramo de desenvolvimento devem acompanhar o FreeBSD-STABLE.

O FreeBSD-CURRENT é o código-fonte mais recente do FreeBSD e inclui trabalhos em andamento,

mudanças experimentais e mecanismos de transição que podem ou não estar presentes na próxima versão oficial. Enquanto muitos desenvolvedores do FreeBSD compilam o código-fonte do FreeBSD-CURRENT diariamente, há curtos períodos de tempo em que o código fonte pode não ser compilável. Esses problemas são resolvidos o mais rápidamente possível, mas se o FreeBSD-CURRENT traz ou não uma nova funcionalidade pode ser uma questão de quando o código-fonte foi sincronizado.

O FreeBSD-CURRENT é disponibilizado para três grupos de interesse principais:

1. Membros da comunidade do FreeBSD que estão trabalhando ativamente em alguma parte da árvore de códigos fontes.
2. Membros da comunidade FreeBSD que são testadores ativos. Eles estão dispostos a gastar tempo resolvendo problemas, fazendo sugestões sobre mudanças e sobre a direção geral do FreeBSD, e enviando correções.
3. Usuários que desejam ficar de olho nas coisas, usam o código fonte atual para fins de referência ou fazem comentários ocasionais ou contribuições de código.

O FreeBSD-CURRENT *não* deve ser considerado um fast-track para obter novos recursos antes do próximo release, já que os recursos de pré-release ainda não foram totalmente testados e provavelmente contêm bugs. Não é uma maneira rápida de obter correções de bugs, pois qualquer commit é tão provável de introduzir novos bugs quanto consertar os existentes. O FreeBSD-CURRENT não é de nenhuma maneira "oficialmente suportado".

Para acompanhar o FreeBSD-CURRENT:

1. Junte-se as listas freebsd-current e svn-src-head. Isto é *essencial* para ver os comentários que as pessoas estão fazendo sobre o estado atual do sistema e para receber importantes boletins sobre o estado atual do FreeBSD-CURRENT.

   A lista svn-src-head registra a entrada de log de commit para cada alteração assim que ela é feita, juntamente com qualquer informação pertinente sobre possíveis efeitos colaterais.

   Para juntar-se a estas listas, vá para http://lists.FreeBSD.org/mailman/listinfo, clique na lista para se inscrever e siga as instruções. A fim de rastrear mudanças em toda a árvore de código-fonte, não apenas as mudanças no FreeBSD-CURRENT, inscreva-se na lista svn-src-all.

2. Sincronize com o código-fonte do FreeBSD-CURRENT. Normalmente, o svnlite é usado para obter o código -CURRENT da ramificação `head` de um dos sites espelhos do Subversion listados em Sites Espelho do Subversion.
3. Devido ao tamanho do repositório, alguns usuários escolhem sincronizar apenas as seções do código-fonte que lhes interessam ou para as quais estão contribuindo com correções. No entanto, os usuários que planejam compilar o sistema operacional a partir do código-fonte devem baixar *tudo* do FreeBSD-CURRENT, não apenas as partes selecionadas.

   Antes de compilar o FreeBSD-CURRENT , leia o /usr/src/Makefile com muito cuidado e siga as instruções em Atualizando o FreeBSD a partir do código fonte. Leia a lista de discussão do FreeBSD-CURRENT e o /usr/src/UPDATING para manter-se atualizado sobre outros procedimentos de bootstrapping que algumas vezes se tornam necessários no caminho para a próxima versão.

4. Ser ativo! Usuários do FreeBSD-CURRENT são encorajados a enviar suas sugestões para melhorias ou correções de bugs. Sugestões acompanhadas de código são sempre bem vindas.

23.4.2. Usando o FreeBSD-STABLE

O FreeBSD-STABLE é o ramo de desenvolvimento a partir do qual as releases principais são feitas. Mudanças entram neste ramo em um ritmo mais lento e com a suposição geral de que elas foram testadas primeiro no FreeBSD-CURRENT. Ele *ainda* é um ramo de desenvolvimento e, a qualquer momento, as fontes para o FreeBSD-STABLE podem ou não ser adequadas para uso geral. É simplesmente outra trilha de desenvolvimento de engenharia, não um recurso para usuários finais. Usuários que não possuem recursos para realizar testes devem, ao invés disso, executar a release mais recente do FreeBSD.

Os interessados em acompanhar ou contribuir para o processo de desenvolvimento do FreeBSD, especialmente no que se refere à próxima versão do FreeBSD, devem considerar seguir o FreeBSD-STABLE.

Embora seja esperado que o ramo FreeBSD-STABLE compile e execute o tempo todo, isso não pode ser garantido. Uma vez que mais pessoas executam o FreeBSD-STABLE do que o FreeBSD-CURRENT, é inevitável que bugs e problemas mais raros às vezes sejam encontrados no FreeBSD-STABLE os quais não foram detectados no FreeBSD-CURRENT. Por esta razão, não se deve seguir cegamente o FreeBSD-STABLE. É particularmente importante *não* atualizar quaisquer servidores de produção para o FreeBSD-STABLE sem testar completamente o código em um ambiente de desenvolvimento ou de teste.

Para acompanhar o FreeBSD-STABLE:

1. Junte-se à lista freebsd-stable para se manter informado sobre as dependências de compilação que podem aparecer no FreeBSD-STABLE ou qualquer outro problema que requeira atenção especial. Os desenvolvedores também farão anúncios nesta lista de e-mails quando estiverem contemplando alguma correção ou atualização controversa, dando aos usuários uma chance de responder se tiverem alguma questão a ser levantada sobre a alteração proposta.

   Junte-se à lista svn relevante para o ramo que está sendo acompanhado. Por exemplo, os usuários que acompanham o ramo 9-STABLE devem se juntar a lista svn-src-stable-9. Esta lista registra a entrada do log de commit para cada alteração à medida que ela é feita, junto com qualquer informação pertinente sobre os possíveis efeitos colaterais.

   Para se juntar a estas listas, vá para http://lists.FreeBSD.org/mailman/listinfo, clique na lista para se inscrever e siga as instruções. Se desejar acompanhar as mudanças para toda a árvore de código-fonte, inscreva-se na svn-src-all .

2. Para instalar um novo sistema FreeBSD-STABLE, instale a versão mais recente do FreeBSD-STABLE a partir de um dos sites espelho do FreeBSD ou use um snapshot mensal criado a partir do FreeBSD-STABLE. Consulte www.freebsd.org/snapshots para maiores informações sobre snapshots.

   Para compilar ou atualizar um sistema FreeBSD existente para o FreeBSD-STABLE, use o svnlite para obter o código-fonte da ramificação desejada. Os nomes das ramificações, como `stable/9`, estão listados em www.freebsd.org/releng.

3. Antes de compilar ou atualizar para o FreeBSD-STABLE , leia o /usr/src/Makefile cuidadosamente e siga as instruções em Atualizando o FreeBSD a partir do código fonte. Leia a lista de discussão FreeBSD-STABLE e o /usr/src/UPDATING para manter-se atualizado sobre outros procedimentos de bootstrapping que às vezes se tornam necessários no caminho para o próximo release.

23.5. Atualizando o FreeBSD a partir do código fonte

A atualização do FreeBSD através da compilação a partir do código-fonte oferece várias vantagens sobre as atualizações binárias. O código pode ser compilado com opções para aproveitar o hardware específico. Partes do sistema base podem ser compiladas com configurações não padrões, ou deixadas de fora somente onde não são necessárias ou desejadas. O processo de compilação leva mais tempo para atualizar um sistema do que apenas instalar atualizações binárias, mas permite customização completa para produzir uma versão do FreeBSD adaptada as suas necessidades.

23.5.1. Inicio Rápido

Esta é uma referência rápida para as etapas típicas usadas para atualizar o FreeBSD compilando-o a partir do código fonte. As seções posteriores descrevem o processo com mais detalhes.

1. Atualizar e Compilar

```
# svnlite update /usr/src  ①
check /usr/src/UPDATING  ②
# cd /usr/src            ③
# make -j4 buildworld    ④
# make -j4 kernel        ⑤
# shutdown -r now        ⑥
# cd /usr/src            ⑦
# make installworld      ⑧
# mergemaster -Ui        ⑨
# shutdown -r now        ⑩
```

① Obtenha a versão mais recente do código fonte. Veja Atualizando o código fonte para maiores informações sobre como obter e atualizar o código fonte.

② Verifique o /usr/src/UPDATING para quaisquer etapas manuais necessárias antes ou depois de compilar a partir do código fonte.

③ Vá para o diretório de origem.

④ Compile o mundo, tudo exceto o kernel.

⑤ Compile e instale o kernel. Isso é equivalente a `make installkernel installkernel`.

⑥ Reinicialize o sistema com o novo kernel.

⑦ Vá para o diretório de origem.

⑧ Instale o mundo.

⑨ Atualize e mescle os arquivos de configuração em /etc/.

⑩ Reinicie o sistema para usar o mundo e o kernel recém-compilados.

23.5.2. Preparando-se para uma atualização a partir do código fonte

Leia o /usr/src/UPDATING. Quaisquer etapas manuais que devem ser executadas antes ou depois de uma atualização são descritas neste arquivo.

23.5.3. Atualizando o código fonte

O código fonte do FreeBSD está localizado em /usr/src/. O método preferido para atualizar os fontes é através do sistema de controle de versão do Subversion. Verifique se o código-fonte está sob controle de versão:

```
# svnlite info /usr/src
Path: /usr/src
Working Copy Root Path: /usr/src
...
```

Isto indica que o /usr/src/ está sob controle de versão e pode ser atualizado com o svnlite(1):

```
# svnlite update /usr/src
```

O processo de atualização pode levar algum tempo se o diretório não tiver sido atualizado recentemente. Após a conclusão, o código-fonte estará atualizado e o processo de compilação descrito na próxima seção poderá começar.



Obtendo o código fonte

Se a saída disser que `'/usr/src' is not a working copy`, estão faltando arquivos no diretório ou eles foram instalados com um método diferente. Um novo checkout da fonte é necessário.

Tabela 18. Versões do FreeBSD e Caminhos do Repositório

Saída do uname -r	Caminho do Repositório	Descrição
`X.Y-RELEASE`	`base/releng/`*X.Y*	A versão do release mais apenas correções críticas de segurança e correção de erros. Este ramo é recomendado para a maioria dos usuários.

Saída do uname -r	Caminho do Repositório	Descrição
`X.Y-STABLE`	`base/stable/`*X*	A versão de Release mais todos os desenvolvimentos adicionais nesse ramo. O *STABLE* refere-se à interface binária de aplicativos (ABI) não sendo alterada, portanto, o software compilado para versões anteriores ainda é executado. Por exemplo, o software compilado para rodar no FreeBSD 10.1 ainda rodará no FreeBSD 10-STABLE compilado posteriormente. Os ramos STABLE ocasionalmente possuem bugs ou incompatibilidades que podem afetar os usuários, embora sejam normalmente corrigidos rapidamente.
`X-CURRENT`	`base/head/`	A mais recente versão de desenvolvimento do FreeBSD. A ramificação CURRENT pode ter grandes erros ou incompatibilidades e é recomendada apenas para usuários avançados.

Determine qual versão do FreeBSD está sendo usada com uname(1):

```
# uname -r
10.3-RELEASE
```

Baseado em Versões do FreeBSD e Caminhos do Repositório, a fonte usada para atualizar `10.3-RELEASE` tem como caminho de repositório `base/releng/10.3`. Este caminho é usado ao verificar a fonte:

```
# mv /usr/src /usr/src.bak ①
```

```
# svnlite checkout https://svn.freebsd.org/base/releng/10.3 /usr/src ②
```

① Mova o diretório antigo para fora do caminho. Se não houver modificações locais nesse diretório, ele poderá ser excluído.

② O caminho da Versões do FreeBSD e Caminhos do Repositório é adicionado a URL repositório . O terceiro parâmetro é o diretório de destino do código-fonte no sistema local.

23.5.4. Compilando a partir do código-fonte

O *world*, ou todo o sistema operacional, exceto o kernel, é compilado. Isso é feito primeiro para fornecer ferramentas atualizadas para construir o kernel. Então o próprio kernel é construído:

```
# cd /usr/src
# make buildworld
# make buildkernel
```

O código compilado é escrito em /usr/obj.

Estes são os passos básicos. Opções adicionais para controlar a compilação são descritas abaixo.

23.5.4.1. Executando uma compilação limpa

Algumas versões do sistema de compilação do FreeBSD deixam o código previamente compilado no diretório de objetos temporários, /usr/obj. Isso pode acelerar as compilações posteriores, evitando recompilar o código que não foi alterado. Para forçar uma reconstrução limpa de tudo, use `cleanworld` antes de iniciar uma construção:

```
# make cleanworld
```

23.5.4.2. Definindo o Número de Jobs

Aumentar o número de jobs de compilação em processadores com vários núcleos pode melhorar a velocidade de construção. Determine o número de núcleos com `sysctl hw.ncpu`. Os processadores variam, assim como os sistemas de compilação usados com diferentes versões do FreeBSD, portanto, o teste é o único método seguro para determinar como um número diferente de tarefas afeta a velocidade de compilação. Como ponto de partida, considere valores entre metade e o dobro do número de núcleos. O número de jobs é especificado com a opção `-j`.

Exemplo 43. Aumentando o número de jobs de compilação

Compilando o mundo e o kernel com quatro jobs:

```
# make -j4 buildworld buildkernel
```

23.5.4.3. Compilando Apenas o Kernel

Um `buildworld` deve ser completado se o código-fonte for alterado. Depois disso, um `buildkernel` para compilar um kernel pode ser executado a qualquer momento. Para compilar apenas o kernel:

```
# cd /usr/src
# make buildkernel
```

23.5.4.4. Compilando um Kernel Customizado

O kernel padrão do FreeBSD é baseado em um *arquivo de configuração do kernel* chamado GENERIC. O kernel GENERIC inclui os drivers e opções de dispositivos mais comumente necessários. Às vezes, é útil ou necessário criar um kernel personalizado, adicionando ou removendo drivers de dispositivo ou opções para atender a uma necessidade específica.

Por exemplo, alguém que desenvolve um pequeno computador embarcado com RAM severamente limitada pode remover drivers de dispositivo desnecessários ou opções para tornar o kernel um pouco menor.

Os arquivos de configuração do Kernel estão localizados em /usr/src/sys/arch/conf/, onde *arch* é a saída do `uname -m`. Na maioria dos computadores, a saida será `amd64`, resultando no diretório de arquivos de configuração /usr/src/sys/amd64/conf/.



O /usr/src pode ser deletado ou recriado, então é preferível manter os arquivos de configuração do kernel em um diretório separado, como por exemplo em /root. Vincule o arquivo de configuração do kernel ao diretório conf. Se esse diretório for excluído ou sobrescrito, a configuração do kernel pode ser vinculada novamente ao novo.

Um arquivo de configuração personalizado pode ser criado copiando o arquivo de configuração GENERIC. Neste exemplo, o novo kernel personalizado é para um servidor de armazenamento, portanto, é denominado STORAGESERVER:

```
# cp /usr/src/sys/amd64/conf/GENERIC /root/STORAGESERVER
# cd /usr/src/sys/amd64/conf
# ln -s /root/STORAGESERVER .
```

O /root/STORAGESERVER é então editado, adicionando ou removendo dispositivos ou opções como mostrado em config(5).

O kernel personalizado é compilado pela configuração `KERNCONF` no arquivo de configuração do kernel na linha de comando:

```
# make buildkernel KERNCONF=STORAGESERVER
```

23.5.5. Instalando o código compilado

Depois que as etapas `buildworld` e `buildkernel` forem concluídas, o novo kernel e o restante do sistema base serão instalados:

```
# cd /usr/src
# make installkernel
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now
```

Se um kernel customizado foi compilado, `KERNCONF` também deve ser configurado para usar o novo kernel customizado:

```
# cd /usr/src
# make installkernel KERNCONF=STORAGESERVER
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now
```

23.5.6. Concluindo a atualização

Algumas tarefas finais completam a atualização. Quaisquer arquivos de configuração que tenham sido modificados serão mesclado com as novas versões, as bibliotecas desatualizadas são localizadas e removidas e, em seguida, o sistema é reiniciado.

23.5.6.1. Mesclando arquivos de configuração com o mergemaster(8)

O mergemaster(8) fornece uma maneira fácil de mesclar as alterações feitas nos arquivos de configuração do sistema com novas versões desses arquivos.

Com a opção `-Ui`, o mergemaster(8) atualizará automaticamente os arquivos que não foram modificados pelo usuário e instalará os novos arquivos que ainda não estiverem presentes:

```
# mergemaster -Ui
```

Se um arquivo precisar ser mesclado manualmente, uma exibição interativa permitirá que o usuário escolha quais partes dos arquivos serão mantidas. Veja mergemaster(8) para maiores informações.

23.5.6.2. Verificando Arquivos e Bibliotecas Desatualizados

Alguns arquivos ou diretórios obsoletos podem permanecer após uma atualização. Esses arquivos podem ser localizados:

```
# make check-old
```

e excluído:

```
# make delete-old
```

Algumas bibliotecas obsoletas também podem permanecer. Estes podem ser detectados com:

```
# make check-old-libs
```

e deletado com

```
# make delete-old-libs
```

Os programas que ainda estavam usando estas bibliotecas antigas deixarão de funcionar quando a biblioteca for excluída. Estes programas devem ser recompilados ou substituídos após a exclusão das bibliotecas antigas.



Quando todos os arquivos ou diretórios antigos forem considerados seguros para serem excluídos, a ação de pressionar `y` e `Enter` para excluir cada arquivo poderá ser evitada configurando a variável `BATCH_DELETE_OLD_FILES` no comando. Por exemplo:

```
# make BATCH_DELETE_OLD_FILES=yes delete-old-libs
```

23.5.6.3. Reiniciando após a atualização

A última etapa após a atualização é reiniciar o computador para que todas as alterações entrem em vigor:

```
# shutdown -r now
```

23.6. Atualização de várias máquinas

Quando várias máquinas precisam rastrear a mesma árvore de código-fonte, é um desperdício de espaço em disco, largura de banda de rede e ciclos de CPU se cada sistema tiver que baixar o código-fonte e recompilar tudo. A solução é ter uma máquina fazendo a maior parte do trabalho, enquanto o resto das máquinas montam esse trabalho via NFS. Esta seção descreve um método para fazer isso. Para maiores informações sobre o uso de NFS, consulte Network File System (NFS).

Primeiro, identifique um conjunto de máquinas que executará o mesmo conjunto de binários, conhecido como *conjunto de compilação*. Cada máquina pode ter um kernel customizado, mas

executará os mesmos binários do userland. A partir desse conjunto, escolha uma máquina para ser a *máquina de compilação* em que o sistema base e o kernel serão compilados. Idealmente, esta deverá ser uma máquina rápida que tenha CPU suficiente disponível para executar o `make buildworld` e o `make buildkernel`.

Selecione uma máquina para ser a *máquina de teste*, que testará as atualizações de software antes de serem colocadas em produção. Esta *deve* ser uma máquina que você possa se dar ao luxo de ficar inativa por um longo período de tempo. Pode ser a máquina de compilação, mas não precisa ser.

Todas as máquinas neste conjunto de compilação precisam montar o /usr/obj e o /usr/src da máquina de compilação através do NFS. Para vários conjuntos de compilação, o /usr/src deve ser um sistema de arquivos local na máquina de compilação e um sistema montado por NFS nas demais.

Certifique-se de que o /etc/make.conf e o /etc/src.conf em todas as máquinas no conjunto de compilação concordem com a máquina de compilação. Isso significa que a máquina de compilação deve compilar todas as partes do sistema base que qualquer máquina no conjunto de compilação irá instalar. Além disso, cada máquina de compilação deve ter seu nome de kernel definido com `KERNCONF` em /etc/make.conf, e a máquina de compilação deve listá-los todos em seu `KERNCONF`, listando seu próprio kernel primeiro. A máquina de compilação deve ter os arquivos de configuração do kernel para cada máquina em seu /usr/src/sys/arch/conf.

Na máquina de compilação, compile o kernel e o mundo conforme descrito em Atualizando o FreeBSD a partir do código fonte, mas não instale nada na máquina de compilação. Em vez disso, instale o kernel compilado na máquina de teste. Na máquina de teste, monte /usr/src e o /usr/obj via NFS. Em seguida, execute `shutdown now` para ir para o modo de usuário único para instalar o novo kernel e o restante do sistema base e execute o `mergemaster` como de costume. Quando terminar, reinicialize para retornar às operações multiusuário normais.

Depois de verificar se tudo na máquina de teste está funcionando corretamente, use o mesmo procedimento para instalar o novo software em cada uma das outras máquinas no conjunto de compilação.

A mesma metodologia pode ser usada para a árvore de ports. O primeiro passo é compartilhar o /usr/ports via NFS para todas as máquinas no conjunto de compilação. Para configurar o /etc/make.conf para compartilhar os distfiles, configure o `DISTDIR` para um diretório compartilhado que possa ser escrito por qualquer usuário `root` mapeado pela montagem NFS. Cada máquina deve definir o `WRKDIRPREFIX` para um diretório de compilação local, se os ports precisarem ser compilados localmente. Como alternativa, se o sistema de compilação tiver que compilar e distribuir pacotes para as máquinas no conjunto de compilação, configure o `PACKAGES` no sistema de compilação para um diretório semelhante ao `DISTDIR`.

Capítulo 24. DTrace

24.1. Sinopse

O DTrace, também conhecido como Dynamic Tracing, foi desenvolvido pela Sun™ como uma ferramenta para localizar gargalos de desempenho em sistemas de produção e pré-produção. Além de diagnosticar problemas de desempenho, o DTrace pode ser usado para ajudar a investigar e depurar comportamentos inesperados no kernel do FreeBSD e em programas da userland.

O DTrace é uma ferramenta de criação de perfil notável, com uma impressionante variedade de recursos para diagnosticar problemas do sistema. Ele também pode ser usado para executar scripts pré-escritos para aproveitar seus recursos. Os usuários podem criar seus próprios utilitários usando a DTrace D Language, permitindo que eles personalizem seus perfis com base em necessidades específicas.

A implementação do FreeBSD fornece suporte completo para o DTrace do kernel e suporte experimental para o DTrace da userland. O Userland DTrace permite que os usuários executem o rastreio de limite de função para programas de área de trabalho usando o provedor `pid` e insiram investigações estáticas em programas da userland para rastreamento posterior. Alguns ports, como databases/postgresql12-server e lang/php74, possuem uma opção do DTrace para ativar testes estáticos.

O guia oficial do DTrace é mantido pelo projeto Illumos no Guia do DTrace.

Depois de ler este capítulo, você saberá:

- O que é o DTrace e quais recursos ele fornece.
- Diferenças entre a implementação do DTrace Solaris™ e a fornecida pelo FreeBSD.
- Como ativar e usar o DTrace no FreeBSD.

Antes de ler este capítulo, você deve:

- Entender os fundamentos do UNIX™ e do FreeBSD (Fundamentos do FreeBSD).
- Ter alguma familiaridade com segurança e como ela está presente no FreeBSD (Segurança).

24.2. Diferenças de Implementação

Embora o DTrace no FreeBSD seja semelhante ao encontrado no Solaris™, existem diferenças. A principal diferença é que no FreeBSD, o DTrace é implementado como um conjunto de módulos do kernel e o DTrace não pode ser usado até que os módulos sejam carregados. Para carregar todos os módulos necessários:

```
# kldload dtraceall
```

Começando com o FreeBSD 10.0-RELEASE, os módulos são carregados automaticamente quando o `dtrace` é executado.

O FreeBSD usa a opção do kernel `DDB_CTF` para ativar o suporte para carregar dados CTF dos módulos do kernel e do próprio kernel. O CTF é o Solaris™ Compact C Type Format, que encapsula uma forma reduzida de informações de depuração semelhantes ao DWARF e aos veneráveis stabs. Os dados do CTF são adicionados aos binários pelas ferramentas de compilação `ctfconvert` e `ctfmerge`. O utilitário `ctfconvert` analisa as seções de depuração do DWARFELF criadas pelo compilador e o `ctfmerge` mescla as seções do ELF do CTF dos objetos em executáveis ou bibliotecas compartilhadas.

Alguns provedores diferentes existem para o FreeBSD não para o Solaris™. O mais notável é o provedor `dtmalloc`, que permite rastrear `malloc()` por tipo no kernel do FreeBSD. Alguns dos provedores encontrados no Solaris™, como `cpc` e `mib`, não estão presentes no FreeBSD. Estes podem aparecer em futuras versões do FreeBSD. Além disso, alguns dos provedores disponíveis em ambos os sistemas operacionais não são compatíveis, no sentido de que suas sondas têm tipos de argumentos diferentes. Assim, os scripts D escritos em Solaris™ podem ou não funcionar sem modificações no FreeBSD, e vice-versa.

Devido as diferenças de segurança, somente o `root` pode usar o DTrace no FreeBSD. O Solaris™ possui algumas verificações de segurança de baixo nível que ainda não existem no FreeBSD. Como tal, o /dev/dtrace/dtrace é estritamente limitado ao `root`.

O DTrace se enquadra na licença Common Development and Distribution License (CDDL). Para ver esta licença no FreeBSD, consulte /usr/src/cddl/contrib/opensolaris/OPENSOLARIS.LICENSE ou acesse on-line em http://opensource.org/licenses/CDDL-1.0. Enquanto um kernel do FreeBSD com suporte a DTrace é licenciado sob BSD, o CDDL é usado quando os módulos são distribuídos em formato binário ou quando os binários são carregados.

24.3. Ativando o Suporte do DTrace

No FreeBSD 9.2 e 10.0, o suporte do DTrace está embutido no kernel GENERIC. Usuários de versões anteriores do FreeBSD ou que preferem compilar estaticamente o suporte do DTrace devem adicionar as seguintes linhas a um arquivo de configuração de kernel personalizado e recompilar o kernel usando as instruções em Configurando o kernel do FreeBSD:

```
options         KDTRACE_HOOKS
options         DDB_CTF
makeoptions DEBUG=-g
makeoptions WITH_CTF=1
```

Os usuários da arquitetura AMD64 também devem adicionar esta linha:

```
options         KDTRACE_FRAME
```

Esta opção fornece suporte para FBT. Embora o DTrace funcione sem essa opção, haverá suporte limitado para o rastreamento de limite de função.

Uma vez que o sistema FreeBSD foi reinicializado no novo kernel, ou os módulos de kernel do DTrace foram carregados usando `kldload dtraceall`, o sistema precisará de suporte para o shell

Korn, pois o DTrace Toolkit possui vários utilitários escritos em ksh. Certifique-se de que o pacote ou port shells/ksh93 esteja instalado. Também é possível rodar estas ferramentas com shells/pdksh ou shells/mksh.

Por fim, instale o DTrace Toolkit atual, uma coleção de scripts prontos para coletar informações do sistema. Existem scripts para verificar arquivos abertos, memória, uso de CPU e muito mais. O FreeBSD 10 instala alguns desses scripts em /usr/shared/dtrace. Em outras versões do FreeBSD, ou para instalar o DTrace Toolkit completo, use o pacote ou port sysutils/dtrace-toolkit.



Os scripts encontrados em /usr/shared/dtrace foram especificamente portados para o FreeBSD. Nem todos os scripts encontrados no DTrace Toolkit funcionarão no FreeBSD e alguns scripts podem exigir algum esforço para que funcionem no FreeBSD.

O DTrace Toolkit inclui muitos scripts no idioma especial do DTrace. Esta linguagem é chamada de linguagem D e é muito semelhante ao C++. Uma discussão aprofundada da linguagem está além do escopo deste documento. Ele é abordado extensivamente no Illumos Dynamic Tracing Guide.

24.4. Usando o DTrace

Os scripts do DTrace consistem em uma lista de um ou mais testes *probes*, ou pontos de instrumentação, em que cada teste é associado a uma ação. Sempre que a condição de uma sonda é atendida, a ação associada é executada. Por exemplo, uma ação pode ocorrer quando um arquivo é aberto, um processo é iniciado ou uma linha de código é executada. A ação pode ser registrar algumas informações ou modificar variáveis de contexto. A leitura e a escrita de variáveis de contexto permitem que os probes compartilhem informações e analisem cooperativa-mente a correlação de diferentes eventos.

Para ver todos os probes, o administrador pode executar o seguinte comando:

```
# dtrace -l | more
```

Cada probe possui um ID, um PROVIDER (dtrace ou fbt), um MODULE e um FUNCTION NAME. Consulte dtrace(1) para obter maiores informações sobre este comando.

Os exemplos nesta seção fornecem uma visão geral de como usar dois dos scripts totalmente suportados dos scripts do DTrace Toolkit: o hotkernel e procsystime.

O script hotkernel é projetado para identificar qual função está usando a maior parte do tempo do kernel. Ele produzirá uma saída semelhante à seguinte:

```
# cd /usr/local/share/dtrace-toolkit
# ./hotkernel
Sampling... Hit Ctrl-C to end.
```

Conforme instruído, use a combinação de teclas Ctrl + C para interromper o processo. Após o término, o script exibirá uma lista de funções do kernel e informações de tempo, classificando a

saída em ordem crescente de tempo:

```
kernel`_thread_lock_flags                                2   0.0%
0xc1097063                                               2   0.0%
kernel`sched_userret                                     2   0.0%
kernel`kern_select                                       2   0.0%
kernel`generic_copyin                                    3   0.0%
kernel`_mtx_assert                                       3   0.0%
kernel`vm_fault                                          3   0.0%
kernel`sopoll_generic                                    3   0.0%
kernel`fixup_filename                                    4   0.0%
kernel`_isitmyx                                          4   0.0%
kernel`find_instance                                     4   0.0%
kernel`_mtx_unlock_flags                                 5   0.0%
kernel`syscall                                           5   0.0%
kernel`DELAY                                             5   0.0%
0xc108a253                                               6   0.0%
kernel`witness_lock                                      7   0.0%
kernel`read_aux_data_no_wait                             7   0.0%
kernel`Xint0x80_syscall                                  7   0.0%
kernel`witness_checkorder                                7   0.0%
kernel`sse2_pagezero                                     8   0.0%
kernel`strncmp                                           9   0.0%
kernel`spinlock_exit                                    10   0.0%
kernel`_mtx_lock_flags                                  11   0.0%
kernel`witness_unlock                                   15   0.0%
kernel`sched_idletd                                    137   0.3%
0xc10981a5                                           42139  99.3%
```

Este script também funcionará com módulos do kernel. Para usar este recurso, execute o script com `-m`:

```
# ./hotkernel -m
Sampling... Hit Ctrl-C to end.
^C
MODULE                                                COUNT   PCNT
0xc107882e                                                1   0.0%
0xc10e6aa4                                                1   0.0%
0xc1076983                                                1   0.0%
0xc109708a                                                1   0.0%
0xc1075a5d                                                1   0.0%
0xc1077325                                                1   0.0%
0xc108a245                                                1   0.0%
0xc107730d                                                1   0.0%
0xc1097063                                                2   0.0%
0xc108a253                                               73   0.0%
kernel                                                  874   0.4%
0xc10981a5                                           213781  99.6%
```

O script procsystime captura e imprime o uso do tempo de chamada do sistema para um determinado processo ID (PID) ou nome do processo. No exemplo a seguir, uma nova instância de /bin/csh foi gerada. Então, procsystime foi executado e permaneceu aguardando enquanto alguns comandos foram digitados na outra instância do `csh`. Estes são os resultados deste teste:

```
# ./procsystime -n csh
Tracing... Hit Ctrl-C to end...
^C

Elapsed Times for processes csh,

         SYSCALL          TIME (ns)
          getpid               6131
       sigreturn               8121
           close              19127
           fcntl              19959
             dup              26955
         setpgid              28070
            stat              31899
       setitimer              40938
           wait4              62717
       sigaction              67372
     sigprocmask             119091
    gettimeofday             183710
           write             263242
          execve             492547
           ioctl             770073
           vfork            3258923
      sigsuspend            6985124
            read         3988049784
```

Como mostrado, a chamada de sistema `read()` usou a maior parte do tempo em nanossegundos enquanto a chamada de sistema `getpid()` usou a menor quantidade de tempo.

Capítulo 25. Modo de dispositivo USB/USB OTG

25.1. Sinopse

Este capítulo aborda o uso do Modo de Dispositivo USB e USB On The Go (USB OTG) no FreeBSD. Isso inclui consoles seriais virtuais, interfaces de rede virtual e drives USB virtuais.

Quando rodando em hardware que suporta o modo de dispositivo USB ou USB OTG, como aquele embutido em muitas placas embarcadas, a stack USB do FreeBSD pode ser executada em *modo de dispositivo*. O modo de dispositivo possibilita que o computador apresente-se como diferentes tipos de classes de dispositivos USB, incluindo portas seriais, adaptadores de rede e armazenamento em massa, ou uma combinação dos mesmos. Um host USB como um laptop ou computador desktop pode acessá-los assim como faria com dispositivos USB físicos. O modo de dispositivo é algumas vezes chamado de "modo USB gadget".

Existem duas maneiras básicas pelas quais o hardware pode fornecer a funcionalidade do modo de dispositivo: com uma "porta de cliente" separada, que suporta apenas o modo de dispositivo, e com uma porta USB OTG, que pode fornecer o modo de dispositivo e o modo de host. Para portas USB OTG, a stack USB alterna automaticamente entre o lado do host e o lado do dispositivo, dependendo do que estiver conectado à porta. Conectar um dispositivo USB como um cartão de memória à porta faz com que o FreeBSD mude para o modo de host. Conectar um host USB como um computador faz com que o FreeBSD mude para o modo de dispositivo. As "portas do cliente" de finalidade única sempre funcionam no modo de dispositivo.

O que o FreeBSD apresenta para o host USB depende do sysctl `hw.usb.template`. Alguns modelos fornecem um único dispositivo, como um terminal serial; outros fornecem vários, que podem ser todos usados ao mesmo tempo. Um exemplo é o template 10, que fornece um dispositivo de armazenamento em massa, um console serial e uma interface de rede. Veja o usb_template(4) para obter a lista de valores disponíveis.

Observe que, em alguns casos, dependendo do hardware e do sistema operacional do host, para o host notar a alteração da configuração, ele deve ser fisicamente desconectado e reconectado ou forçado a verificar novamente o barramento USB de uma maneira específica do sistema. Quando o FreeBSD é executado no host, o usbconfig(8)`reset` pode ser usado. Isto também deve ser feito após carregar o usb_template.ko se o host USB já estiver conectado ao soquete USB OTG .

Depois de ler este capítulo, você saberá:

- Como configurar a funcionalidade do modo de dispositivo USB no FreeBSD.
- Como configurar a porta serial virtual no FreeBSD.
- Como se conectar à porta serial virtual de vários sistemas operacionais.
- Como configurar o FreeBSD para fornecer uma interface de rede virtual USB.
- Como configurar o FreeBSD para fornecer um dispositivo virtual de armazenamento USB.

25.2. Portas Seriais Virtuais USB

25.2.1. Configurando Portas Seriais do Modo de Dispositivo USB

O suporte para porta serial virtual é fornecido pelos templates número 3, 8 e 10. Observe que o template 3 funciona com o Microsoft Windows 10 sem a necessidade de drivers especiais e de arquivos INF. Outros sistemas operacionais host funcionam com todos os três modelos. Os módulos do kernel usb_template(4) e umodem(4) devem ser carregados.

Para ativar as portas seriais do modo de dispositivo USB, adicione essas linhas ao /etc/ttys:

```
ttyU0   "/usr/libexec/getty 3wire"  vt100   onifconsole secure
ttyU1   "/usr/libexec/getty 3wire"  vt100   onifconsole secure
```

Então adicione estas linhas ao arquivo /etc/devd.conf:

```
notify 100 {
    match "system"      "DEVFS";
    match "subsystem"   "CDEV";
    match "type"        "CREATE";
    match "cdev"        "ttyU[0-9]+";
    action "/sbin/init q";
};
```

Recarregue a configuração se o devd(8) já estiver em execução:

```
# service devd restart
```

Certifique-se de que os módulos necessários estejam carregados e que o template correto esteja configurado na inicialização, adicionando estas linhas ao /boot/loader.conf, criando-o se ele ainda não existir:

```
umodem_load="YES"
hw.usb.template=3
```

Para carregar o módulo e definir o modelo sem reiniciar, use:

```
# kldload umodem
# sysctl hw.usb.template=3
```

25.2.2. Conectando-se às portas seriais do modo de dispositivo USB a partir do FreeBSD

Para conectar-se a uma placa configurada para fornecer portas seriais de um dispositivo USB, conecte o host USB, como um laptop, às placas USB OTG ou porta de cliente USB. Use `pstat -t` no host para listar as linhas de terminal. Perto do final da lista, você deve ver uma porta serial USB, por exemplo, "ttyU0". Para abrir a conexão, use:

```
# cu -l /dev/ttyU0
```

Depois de pressionar a tecla `Enter` algumas vezes, você verá um prompt de login.

25.2.3. Conectando-se às Portas Seriais do Modo de Dispositivo USB a partir do Mac OS®

Para conectar-se a uma placa configurada para fornecer portas seriais de modo de dispositivo USB, conecte o host USB, como um laptop, às placas USB OTG ou porta de cliente USB. Para abrir a conexão, use:

```
# cu -l /dev/cu.usbmodemFreeBSD1
```

25.2.4. Conectando-se às portas seriais do modo de dispositivo USB a partir do Linux

Para conectar-se a uma placa configurada para fornecer portas seriais de modo de dispositivo USB, conecte o host USB, como um laptop, às placas USB OTG ou porta de cliente USB. Para abrir a conexão, use:

```
# minicom -D /dev/ttyACM0
```

25.2.5. Conectando-se às portas seriais do modo de dispositivo USB a partir do Microsoft Windows 10

Para conectar-se a uma placa configurada para fornecer portas seriais de modo de dispositivo USB, conecte o host USB, como um laptop, às placas USB OTG ou porta de cliente USB. Para abrir uma conexão, você precisará de um programa de terminal serial, como PuTTY. Para verificar o nome da porta COM usada pelo Windows, execute o Gerenciador de dispositivos, expanda. "Ports (COM & LPT)".Você verá um nome semelhante a "USB Serial Device (COM4)". Execute o programa do terminal serial de sua escolha, por exemplo, PuTTY. Na caixa de diálogo PuTTY defina "Connection type" como "Serial", digite o COMx obtido no Gerenciador de Dispositivos na caixa de diálogo "Serial line" e clique em Abrir.

25.3. Interfaces de rede do modo de dispositivo USB

O suporte a interfaces de rede virtuais é fornecido pelos templates número 1, 8 e 10. Observe que nenhum deles funciona com o Microsoft Windows. Outros sistemas operacionais host funcionam com todos os três modelos. Os módulos de kernel usb_template(4) e if_cdce(4) devem ser carregados.

Certifique-se de que os módulos necessários estejam carregados e que o template correto esteja configurado na inicialização, adicionando estas linhas ao /boot/loader.conf, criando-o se ele ainda não existir:

```
if_cdce_load="YES"
hw.usb.template=1
```

Para carregar o módulo e definir o modelo sem reiniciar, use:

```
# kldload if_cdce
# sysctl hw.usb.template=1
```

25.4. Dispositivo de armazenamento virtual USB



O driver cfumass(4) é um driver de modo de dispositivo USB disponibilizado pela primeira vez no FreeBSD 12.0.

O target de armazenamento em massa é fornecido pelos templates 0 e 10. Os módulos de kernel usb_template(4) e cfumass(4) devem ser carregados. O cfumass(4) faz interface com o subsistema CTL, o mesmo usado para os targets iSCSI ou Fibre Channel. No lado do host, os inicializadores do armazenamento em massa USB só podem acessar um único LUN, o LUN 0.

25.4.1. Configurando o target de armazenamento em massa USB usando o script de inicialização cfumass

A maneira mais simples de configurar um target de armazenamento USB somente de leitura é usar o script rc cfumass. Para configurá-lo dessa maneira, copie os arquivos a serem apresentados para a máquina host USB no diretório `/var/cfumass` e inclua esta linha no /etc/rc.conf :

```
cfumass_enable="YES"
```

Para fazer valer a configuração sem reiniciar, execute este comando:

```
# service cfumass start
```

Diferentemente da funcionalidade serial e de rede, o modelo não deve ser definido como 0 ou 10 no /boot/loader.conf. Isso ocorre porque o LUN deve ser configurado antes de definir o template. O

script de inicialização cfumass define o número do modelo correto automaticamente quando iniciado.

25.4.2. Configurando o armazenamento em massa USB usando outros meios

O restante deste capítulo fornece uma descrição detalhada da configuração do target sem usar o arquivo rc cfumass. Isso é necessário se, por exemplo, alguém quiser fornecer um LUN gravável.

O armazenamento em massaUSB não exige que o daemon ctld(8) esteja em execução, embora ele possa ser usado se desejado. Isso é diferente do iSCSI. Portanto, existem duas maneiras de configurar o target: o ctladm(8) ou o ctld(8). Ambos exigem que o módulo do kernel cfumass.ko seja carregado. O módulo pode ser carregado manualmente:

```
# kldload cfumass
```

Se o cfumass.ko não foi compilado estaticamente no kernel, o /boot/loader.conf pode ser configurado para carregar o módulo na inicialização:

```
cfumass_load="YES"
```

Um LUN pode ser criado sem o daemon ctld(8):

```
# ctladm create -b block -o file=/data/target0
```

Isto apresenta o conteúdo do arquivo de imagem /data/target0 como um LUN para o host USB. O arquivo deve existir antes de executar o comando. Para configurar o LUN na inicialização do sistema, adicione o comando ao /etc/rc.local.

O ctld(8) também pode ser usado para gerenciar LUNs. Crie /etc/ctl.conf, adicione uma linha ao /etc/rc.conf para certificar-se ctld(8) é iniciado automaticamente na inicialização e, em seguida, inicie o daemon.

Este é um exemplo de um arquivo de configuração /etc/ctl.conf simple. Consulte ctl.conf(5) para obter uma descrição mais completa das opções.

```
target naa.50015178f369f092 {
    lun 0 {
        path /data/target0
        size 4G
    }
}
```

O exemplo cria um único target com um único LUN. O `naa.50015178f369f092` é um identificador de dispositivo composto por 32 dígitos hexadecimais aleatórios. A linha `path` define o caminho completo para o arquivo ou zvol que suporta o LUN. Esse arquivo deve existir antes do ctld(8) ser

iniciado. A segunda linha é opcional e especifica o tamanho do LUN.

Para ter certeza que o daemon ctld(8) foi iniciado na inicialização, adicione esta linha ao /etc/rc.conf:

```
ctld_enable="YES"
```

Para iniciar o ctld(8) agora, execute este comando:

```
# service ctld start
```

Quando o daemon ctld(8) é iniciado, ele lê o /etc/ctl.conf. Se esse arquivo for editado depois que o daemon iniciar, recarregue as alterações para que elas entrem em vigor imediatamente:

```
# service ctld reload
```

Parte IV: Comunicação de rede

O FreeBSD é um dos sistemas operacionais mais amplamente implantados para servidores de rede de alto desempenho. Os capítulos desta parte cobrem:

- Comunicação Serial
- PPP e PPP sobre Ethernet
- Correio Eletrônico
- Executando Servidores de Rede
- Firewalls
- Outros tópicos avançados de rede

Esses capítulos são projetados para serem lidos quando a informação for necessária. Eles não precisam ser lidos em qualquer ordem específica, nem é necessário ler todos eles antes de usar o FreeBSD em um ambiente de rede.

Capítulo 26. Comunicações Seriais

26.1. Sinopse

O UNIX™ sempre teve suporte para comunicação serial, pois as primeiras máquinas UNIX™ dependiam de linhas seriais para entrada e saída do usuário. As coisas mudaram muito desde os dias em que o terminal médio consistia de uma impressora serial de 10 caracteres por segundo e um teclado. Este capítulo aborda algumas das maneiras pelas quais as comunicações seriais podem ser usadas no FreeBSD.

Depois de ler este capítulo, você saberá:

- Como conectar terminais a um sistema FreeBSD.
- Como usar um modem para discar para hosts remotos.
- Como permitir que usuários remotos efetuem login em um sistema FreeBSD com um modem.
- Como inicializar um sistema FreeBSD a partir de uma console serial.

Antes de ler este capítulo, você deve:

- Saiba como configurar e instalar um kernel personalizado.
- Entenda permissões e processos do FreeBSD.
- Tenha acesso ao manual técnico para o hardware serial a ser usado com o FreeBSD.

26.2. Terminologia serial e hardware

Os termos a seguir são frequentemente usados em comunicações seriais:

bps
: Bits por Segundo (bps) é a taxa na qual os dados são transmitidos.

DTE
: Equipamento Terminal de Dados (DTE) é um dos dois terminais em uma comunicação serial. Um exemplo seria um computador.

DCE
: Equipamento de Comunicações de Dados (DTE) é o outro terminal em uma comunicação serial. Normalmente, é um modem ou terminal serial.

RS-232
: O padrão original que definiu as comunicações seriais de hardware. Desde então, foi renomeado para TIA-232.

Ao se referir a taxas de dados de comunicação, esta seção não usa o termo *baud*. Baud refere-se ao número de transições de estado elétrico feitas em um período de tempo, enquanto bps é o termo correto a ser usado.

Para conectar um terminal serial a um sistema FreeBSD, são necessárias uma porta serial no computador e o cabo adequado para conectar ao dispositivo serial. Os usuários que já estão familiarizados com hardware serial e cabeamento podem pular esta seção com segurança.

26.2.1. Cabos Serial e Portas

Existem vários tipos diferentes de cabos seriais. Os dois tipos mais comuns são cabo null-modem e cabo padrão RS-232. A documentação do hardware deve descrever o tipo de cabo necessário.

Estes dois tipos de cabos diferem em como os fios são conectados ao conector. Cada fio representa um sinal, com os sinais definidos resumidos em RS-232C Nomes dos Sinais. Um cabo serial padrão passa todos os sinais RS-232C diretamente. Por exemplo, o pino "Transmitted Data" em uma extremidade do cabo vai para o pino "Transmitted Data" na outra extremidade. Este é o tipo de cabo usado para conectar um modem ao sistema FreeBSD e também é apropriado para alguns terminais.

Um cabo de modem nulo alterna o pino "Transmitted Data" do conector em uma extremidade com o pino "Received Data" na outra extremidade. O conector pode ser um DB-25 ou um DB-9.

Um cabo de modem nulo pode ser construído usando as conexões de pinos resumidas em Cabo Null-Modem DB-25 para DB-25, Cabo DB-9 para DB-9 Null-Modem e Cabo DB-9 para DB-25 Null-Modem. Enquanto o padrão exige um pino direto 1 para fixar uma linha "Protective Ground", ele é freqüentemente omitido. Alguns terminais funcionam usando apenas os pinos 2, 3 e 7, enquanto outros exigem configurações diferentes. Em caso de dúvida, consulte a documentação do hardware.

Tabela 19. RS-232C Nomes dos Sinais

Siglas	Nomes
RD	Received Data
TD	Transmitted Data
DTR	Data Terminal Ready
DSR	Data Set Ready
DCD	Data Carrier Detect
SG	Signal Ground
RTS	Request to Send
CTS	Clear to Send

Tabela 20. Cabo Null-Modem DB-25 para DB-25

Sinal	Pin #		Pin #	Sinal
SG	7	conecta-se a	7	SG
TD	2	conecta-se a	3	RD
RD	3	conecta-se a	2	TD
RTS	4	conecta-se a	5	CTS
CTS	5	conecta-se a	4	RTS

Sinal	Pin #		Pin #	Sinal
DTR	20	conecta-se a	6	DSR
DTR	20	conecta-se a	8	DCD
DSR	6	conecta-se a	20	DTR
DCD	8	conecta-se a	20	DTR

Tabela 21. Cabo DB-9 para DB-9 Null-Modem

Sinal	Pin #		Pin #	Sinal
RD	2	conecta-se a	3	TD
TD	3	conecta-se a	2	RD
DTR	4	conecta-se a	6	DSR
DTR	4	conecta-se a	1	DCD
SG	5	conecta-se a	5	SG
DSR	6	conecta-se a	4	DTR
DCD	1	conecta-se a	4	DTR
RTS	7	conecta-se a	8	CTS
CTS	8	conecta-se a	7	RTS

Tabela 22. Cabo DB-9 para DB-25 Null-Modem

Sinal	Pin #		Pin #	Sinal
RD	2	conecta-se a	2	TD
TD	3	conecta-se a	3	RD
DTR	4	conecta-se a	6	DSR
DTR	4	conecta-se a	8	DCD
SG	5	conecta-se a	7	SG
DSR	6	conecta-se a	20	DTR
DCD	1	conecta-se a	20	DTR
RTS	7	conecta-se a	5	CTS
CTS	8	conecta-se a	4	RTS



Quando um pino em uma extremidade se conecta a um par de pinos na outra extremidade, geralmente é implementado com um fio curto entre o par de pinos em seu conector e um fio longo no outro pino único.

As portas seriais são os dispositivos através dos quais os dados são transferidos entre o computador host do FreeBSD e o terminal. Vários tipos de portas seriais existem. Antes de comprar ou construir um cabo, verifique se ele irá se encaixar nas portas do terminal e no sistema FreeBSD.

A maioria dos terminais tem portas DB-25. Os computadores pessoais podem ter portas DB-25 ou DB-9. Um cartão serial multiportas pode ter portas RJ-12 ou RJ-45/. Consulte a documentação que acompanha o hardware para especificações sobre o tipo de porta ou verifique visualmente o tipo de porta.

No FreeBSD, cada porta serial é acessada através de uma entrada em /dev. Existem dois tipos diferentes de entradas:

- As portas de chamada são nomeadas /dev/ttyuN onde *N* é o número da porta, começando do zero. Se um terminal estiver conectado a primeira porta serial (COM1), use /dev/ttyu0 para se referir ao terminal. Se o terminal estiver na segunda porta serial (COM2), use /dev/ttyu1 e assim por diante. Geralmente, a porta de chamada é usada para terminais. As portas de chamada requerem que a linha serial declare o sinal "Data Carrier Detect" para funcionar corretamente.
- As portas de chamadas de saida são nomeadas /dev/cuauN nas versões 8.X e superiores do FreeBSD e /dev/cuadN nas versões 7.X e inferiores do FreeBSD. As portas de chamada de saida geralmente não são usadas para terminais, mas são usadas para modems. A porta de evocação pode ser usada se o cabo serial ou o terminal não suportar o sinal "Data Carrier Detect".

O FreeBSD também fornece dispositivos de inicialização (/dev/ttyuN.init e /dev/cuauN.init ou /dev/cuadN.init) e dispositivos de bloqueio (/dev/ttyuN.lock e /dev/cuauN.lock ou /dev/cuadN.lock). Os dispositivos de inicialização são utilizados para inicializar os parâmetros da porta de comunicações de cada vez que uma porta é aberta, tal como o `crtscts` para modems que usam `RTS/CTS` sinalização para controle de fluxo. Os dispositivos de bloqueio são usados para bloquear sinalizadores nas portas para impedir que usuários ou programas alterem determinados parâmetros. Consulte termios(4), sio(4), e stty(1) para obter informações sobre configurações de terminal, bloqueio e inicialização de dispositivos e configuração de opções de terminal, respectivamente.

26.2.2. Configuração de Porta Serial

Por padrão, o FreeBSD suporta quatro portas seriais que são comumente conhecidas como COM1, COM2, COM3 e COM4. O FreeBSD também suporta placas de interfaces seriais multi-port, como o BocaBoard 1008 e 2016, bem como placas multi-port mais inteligentes, como as feitas pela Digiboard. No entanto, o kernel padrão procura apenas as portas padrão COM.

Para ver se o sistema reconhece as portas seriais, procure por mensagens de inicialização do sistema que começam com `uart`:

```
# grep uart /var/run/dmesg.boot
```

Se o sistema não reconhecer todas as portas seriais necessárias, entradas adicionais podem ser adicionadas ao /boot/device.hints. Este arquivo já contém entradas `hint.uart.0.*` para entradas COM1 e `hint.uart.1.*` para COM2. Ao adicionar uma entrada de porta para COM3 use `0x3E8` e para COM4 use `0x2E8`. Endereços comuns de IRQ são `5` para COM3 e `9` para COM4.

Para determinar o conjunto padrão de configurações de terminal E/S usadas pela porta, especifique o nome do dispositivo. Este exemplo determina as configurações para a porta de chamada em COM2:

```
# stty -a -f /dev/ttyu1
```

A inicialização de dispositivos seriais em todo o sistema é controlada por /etc/rc.d/serial. Este arquivo afeta as configurações padrão dos dispositivos seriais. Para alterar as configurações de um dispositivo, use `stty`. Por padrão, as configurações alteradas estão em vigor até que o dispositivo seja fechado e, quando o dispositivo for reaberto, volte para o conjunto padrão. Para alterar permanentemente o conjunto padrão, abra e ajuste as configurações do dispositivo de inicialização. Por exemplo, para ativar o modo `CLOCAL`, comunicação de 8 bits e controle de fluxo `XON/XOFF` para ttyu5, digite:

```
# stty -f /dev/ttyu5.init clocal cs8 ixon ixoff
```

Para impedir que determinadas configurações sejam alteradas por um aplicativo, faça ajustes no dispositivo de bloqueio. Por exemplo, para bloquear a velocidade de ttyu5 para 57600 bps, digite:

```
# stty -f /dev/ttyu5.lock 57600
```

Agora, qualquer aplicativo que abra ttyu5 e tente alterar a velocidade da porta será bloqueado com 57600 bps.

26.3. Terminais

Os terminais fornecem uma maneira conveniente e barata de acessar um sistema FreeBSD quando não estão no console do computador ou em uma rede conectada. Esta seção descreve como usar terminais com o FreeBSD.

Os sistemas originais UNIX™ não tinham consoles. Em vez disso, os usuários efetuaram login e executaram programas por meio de terminais conectados as portas seriais do computador.

A capacidade de estabelecer uma sessão de login em uma porta serial ainda existe em quase todos os sistemas operacionais do tipo UNIX™ hoje, incluindo o FreeBSD. Usando um terminal conectado a uma porta serial não usada, um usuário pode efetuar login e executar qualquer programa de texto que possa ser executado normalmente no console ou em uma janela `xterm`.

Muitos terminais podem ser conectados a um sistema FreeBSD. Um computador sobressalente mais antigo pode ser usado como um terminal conectado a um computador mais potente executando o FreeBSD. Isso pode transformar o que poderia ser um computador de usuário único em um poderoso sistema de múltiplos usuários.

O FreeBSD suporta três tipos de terminais:

Terminais Burros

Terminais burro são um hardware especializado que se conecta a computadores através de linhas seriais. Eles são chamados de "dumb" porque eles possuem apenas poder computacional suficiente para exibir, enviar e receber texto. Nenhum programa pode ser executado nesses

dispositivos. Em vez disso, os terminais burros se conectam a um computador que executa os programas necessários.

Existem centenas de tipos de terminais burro feitos por muitos fabricantes, e praticamente qualquer tipo funciona com o FreeBSD. Alguns terminais high-end podem até exibir gráficos, mas apenas determinados pacotes de software podem aproveitar esses recursos avançados.

Terminais burro são populares em ambientes de trabalho onde os trabalhadores não precisam de acesso a aplicativos gráficos.

Computadores Atuando como Terminais

Como um terminal burro tem capacidade suficiente para exibir, enviar e receber texto, qualquer computador de reserva pode ser um terminal burro. Tudo o que é necessário é o cabo adequado e algum software de *terminal emulation* para ser executado no computador.

Esta configuração pode ser útil. Por exemplo, se um usuário está ocupado trabalhando no console do sistema FreeBSD, outro usuário pode fazer algum trabalho somente de texto ao mesmo tempo de um computador pessoal menos potente ligado como um terminal ao sistema FreeBSD.

Existem pelo menos dois utilitários no sistema base do FreeBSD que podem ser usados para trabalhar através de uma conexão serial: cu(1) e tip(1).

Por exemplo, para conectar-se de um sistema cliente que executa o FreeBSD para a conexão serial de outro sistema:

```
# cu -l /dev/cuauN
```

Portas são numeradas a partir de zero. Isso significa que COM1 é /dev/cuau0.

Programas adicionais estão disponíveis através da coleção de ports, como comms/minicom.

Terminais X

Os terminais X são o tipo de terminal mais sofisticado disponível. Em vez de se conectar a uma porta serial, eles geralmente se conectam a uma rede como a Ethernet. Em vez de serem relegados a aplicativos somente de texto, eles podem exibir qualquer aplicativo Xorg.

Este capítulo não cobre a configuração ou uso de terminais X.

26.3.1. Configuração do Terminal

Esta seção descreve como configurar um sistema FreeBSD para ativar uma sessão de login em um terminal serial. Assume-se que o sistema reconhece a porta serial a qual o terminal está conectado e que o terminal está conectado com o cabo correto.

No FreeBSD, o `init` lê o /etc/ttys e inicia um processo `getty` nos terminais disponíveis. O processo `getty` é responsável por ler um nome de login e iniciar o programa `login`. As portas no sistema FreeBSD que permitem logins estão listadas em /etc/ttys. Por exemplo, o primeiro console virtual, ttyv0, possui uma entrada nesse arquivo, permitindo logins no console. Este arquivo também

contém entradas para os outros consoles virtuais, portas seriais e pseudo-ttys. Para um terminal com fio, a entrada /dev da porta serial é listada sem a parte `/dev`. Por exemplo, /dev/ttyv0 está listado como `ttyv0`.

Por padrão o /etc/ttys configura o suporte para as quatro primeiras portas seriais, ttyu0 até ttyu3:

```
ttyu0   "/usr/libexec/getty std.9600"   dialup  off secure
ttyu1   "/usr/libexec/getty std.9600"   dialup  off secure
ttyu2   "/usr/libexec/getty std.9600"   dialup  off secure
ttyu3   "/usr/libexec/getty std.9600"   dialup  off secure
```

Ao conectar um terminal a uma destas portas, modifique a entrada padrão para definir a velocidade e o tipo de terminal necessários, para ligar o dispositivo `on` e, se necessário, para alterar o `secure` da porta. Se o terminal estiver conectado a outra porta, adicione uma entrada para a porta.

Configurando Entradas de Terminal configura dois terminais em /etc/ttys. A primeira entrada configura um Wyse-50 conectado ao COM2. A segunda entrada configura um computador antigo executando o software do terminal Procomm emulando um terminal VT-100. O computador está conectado à sexta porta serial em uma placa serial com várias portas.

Exemplo 44. Configurando Entradas de Terminal

```
ttyu1  "/usr/libexec/getty std.38400"  wy50  on  insecure
ttyu5   "/usr/libexec/getty std.19200"  vt100  on insecure
```

- O primeiro campo especifica o nome do dispositivo do terminal serial.
- O segundo campo informa ao `getty` para inicializar e abrir a linha, definir a velocidade da linha, solicitar um nome de usuário e, em seguida, executar o programa `login`. O tipo de *getty type* configura características na linha do terminal, como taxa e paridade bps. Os tipos de getty disponíveis estão listados em /etc/gettytab. Em quase todos os casos, os tipos de getty que começam com `std` funcionarão para terminais conectados, já que essas entradas ignoram a paridade. Há uma entrada `std` para cada taxa de bps de 110 a 115200. Consulte gettytab(5) para mais informações.Ao definir o tipo de getty, certifique-se de coincidir com as configurações de comunicação usadas pelo terminal. Para este exemplo, o Wyse-50 não usa paridade e se conecta a 38400 bps. O computador não usa paridade e se conecta a 19200 bps.
- O terceiro campo é o tipo de terminal. Para portas dial-up, `unknown` ou `dialup` é normalmente usado, pois os usuários podem discar praticamente com qualquer tipo de terminal ou software. Como o tipo de terminal não muda para terminais com fio, um tipo de terminal real de /etc/termcap pode ser especificado. Para este exemplo, o Wyse-50 usa o tipo de terminal real enquanto o computador executando o Procomm está configurado para emular um VT-100.
- O quarto campo especifica se a porta deve estar ativada. Para ativar logins nessa porta, este campo deve ser definido como `on`.
- O campo final é usado para especificar se a porta é segura. Marcar uma porta como `secure`

> significa que ela é confiável o suficiente para permitir que `root` faça login a partir dessa porta. As portas inseguras não permitem logins `root`. Em uma porta insegura, os usuários devem efetuar login de contas não privilegiadas e, em seguida, usar o `su` ou um mecanismo semelhante para obter privilégios de superusuário, conforme descrito em A conta de superusuário. Por razões de segurança, recomenda-se alterar esta configuração para `insecure`.

Depois de fazer qualquer alteração em /etc/ttys, envie um sinal SIGHUP (hangup) para o processo `init` para forçá-lo a reler seu arquivo de configuração:

```
# kill -HUP 1
```

Como o `init` é sempre o primeiro processo executado em um sistema, ele sempre tem um processo ID de `1`.

Se tudo estiver configurado corretamente, todos os cabos estiverem no lugar e os terminais ligados, um processo `getty` deverá estar em execução em cada terminal e as solicitações de login deverão estar disponíveis em cada terminal.

26.3.2. Solução de Problemas da Conexão

Mesmo com a mais meticulosa atenção aos detalhes, algo poderia dar errado ao configurar um terminal. Aqui está uma lista de sintomas comuns e algumas correções sugeridas.

Se nenhum prompt de login aparecer, verifique se o terminal está conectado e ligado. Se for um computador pessoal atuando como um terminal, verifique se ele está executando o software de emulação de terminal na porta serial correta.

Certifique-se de que o cabo esteja conectado firmemente ao terminal e ao computador do FreeBSD. Certifique-se de que é o tipo certo de cabo.

Certifique-se de que o terminal e o FreeBSD concordem com as configurações de taxa e paridade de bps. Para um terminal de exibição de vídeo, verifique se os controles de contraste e brilho estão ativados. Se for um terminal de impressão, verifique se o papel e a tinta estão em bom estado.

Use `ps` para certificar-se de que um processo `getty` esteja em execução e atendendo ao terminal. Por exemplo, a listagem a seguir mostra que um `getty` está sendo executado na segunda porta serial, ttyu1, e está usando a entrada `std.38400` em /etc/gettytab:

```
# ps -axww|grep ttyu
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyu1
```

Se nenhum processo `getty` estiver em execução, certifique-se de que a porta esteja ativada em /etc/ttys. Lembre-se de executar `kill -HUP 1` após modificar /etc/ttys.

Se o processo `getty` estiver em execução, mas o terminal ainda não exibir um prompt de login ou se exibir um prompt, mas não aceitar entrada digitada, o terminal ou cabo poderá não suportar

handshaking de hardware. Tente alterar a entrada em /etc/ttys de `std.38400` para `3wire.38400` e, em seguida, execute `kill -HUP 1` depois de modificar o /etc/ttys. A entrada `3wire` é semelhante a `std`, mas ignora handshaking de hardware. Pode ser necessário reduzir a taxa de transmissão ou ativar o controle de fluxo de software ao usar `3wire` para evitar buffer overflows.

Se aparecer lixo em vez de um prompt de login, certifique-se de que o terminal e o FreeBSD concordem com as configurações de taxa e paridade de bps. Verifique os processos `getty` para certificar-se de que o tipo correto *getty* esteja em uso. Se não, edite /etc/ttys e execute `kill -HUP 1`.

Se os caracteres aparecerem duplicados e a senha aparecer quando digitada, alterne o terminal ou o software de emulação de terminal de "half duplex" ou "local echo" para "full duplex."

26.4. Serviço Dial-in

Configurar um sistema FreeBSD para serviço de discagem é semelhante a configurar terminais, exceto que os modens são usados em vez de dispositivos terminais. O FreeBSD suporta modens externos e internos.

Os modems externos são mais convenientes, pois geralmente podem ser configurados por meio de parâmetros armazenados em RAM não voláteis e geralmente fornecem indicadores luminosos que exibem o estado dos sinais RS-232 importantes, indicando se o modem está funcionando corretamente.

Normalmente, os modems internos não possuem RAM não volátil, portanto, sua configuração pode ser limitada à configuração de switches DIP. Se o modem interno tiver luzes indicadoras de sinal, elas serão difíceis de visualizar quando a tampa do sistema estiver no lugar.

Ao usar um modem externo, é necessário um cabo adequado. Um cabo serial padrão de RS-232C deve ser suficiente.

O FreeBSD precisa dos sinais RTS e CTS para controle de fluxo em velocidades acima de 2400 bps, o sinal CD para detectar quando uma chamada foi atendida ou a linha foi desligada e o sinal DTR para redefinir o modem após a conclusão de uma sessão. Alguns cabos são conectados sem todos os sinais necessários, portanto, se uma sessão de login não desaparecer quando a linha for desligada, pode haver um problema com o cabo. Consulte Cabos Serial e Portas para mais informações sobre esses sinais.

Como outros sistemas operacionais similares ao UNIX™-like, o FreeBSD usa os sinais de hardware para descobrir quando uma chamada foi atendida ou uma linha foi desconectada e para desligar e reinicializar o modem após uma chamada. O FreeBSD evita enviar comandos para o modem ou observar relatórios de status do modem.

O FreeBSD suporta interfaces de comunicação NS8250, NS16450, NS16550 e NS16550A baseado em RS-232C (CCITT V.24). Os dispositivos 8250 e 16450 possuem buffers de caractere único. O dispositivo 16550 fornece um buffer de 16 caracteres, o que permite um melhor desempenho do sistema. Bugs em dispositivos simples 16550 impedem o uso do buffer de 16 caracteres, portanto, use dispositivos 16550A, se possível. Como os dispositivos de buffer de caractere único requerem mais trabalho pelo sistema operacional do que os dispositivos de buffer de 16 caracteres, as placas de interface serial baseadas no 16550A são preferidas. Se o sistema tiver muitas portas seriais

ativas ou tiver uma carga pesada, as placas baseadas em 16550A são melhores para comunicações com baixa taxa de erro.

O restante desta seção demonstra como configurar um modem para receber conexões de entrada, como se comunicar com o modem e oferece algumas dicas de solução de problemas.

26.4.1. Configuração de Modem

Como nos terminais, o `init` gera um processo `getty` para cada porta serial configurada usada para conexões de dial-in. Quando um usuário disca a linha do modem e os modems se conectam, o sinal "Carrier Detect" é informado pelo modem. O kernel percebe que a portadora foi detectada e instrui o `getty` a abrir a porta e exibir um prompt `login:` na velocidade da linha inicial especificada. Em uma configuração típica, se caracteres de lixo forem recebidos, geralmente devido à velocidade de conexão do modem ser diferente da velocidade configurada, o `getty` tenta ajustar as velocidades de linha até receber caracteres razoáveis. Depois que o usuário digita seu nome de login, o `getty` executa o `login`, que conclui o processo de login solicitando a senha do usuário e iniciando o shell do usuário.

Existem duas escolas de pensamento sobre modems dial-up. Um método de configuração é definir os modems e sistemas de modo que, independentemente da velocidade em que um usuário remoto disca, a interface de discagem RS-232 seja executada em uma velocidade travada. O benefício dessa configuração é que o usuário remoto sempre vê um prompt de login do sistema imediatamente. A desvantagem é que o sistema não sabe qual é a verdadeira taxa de dados do usuário, portanto, programas em tela cheia como o Emacs não ajustam seus métodos de tela para melhorar a resposta para conexões mais lentas.

O segundo método é configurar a interface RS-232 para variar sua velocidade com base na velocidade de conexão do usuário remoto. Como o `getty` não compreende o relatório de velocidade de conexão de nenhum modem em particular, ele fornece uma mensagem `login:` em uma velocidade inicial e observa os caracteres que retornam em resposta. Se o usuário vê lixo, eles devem pressionar `Enter` até que um prompt reconhecível seja exibido. Se as taxas de dados não corresponderem, `getty` verá qualquer coisa que o usuário digita como lixo, e tentará a próxima velocidade e informará novamente o prompt `login:`. Esse procedimento normalmente leva apenas um pressionamento de tecla ou dois antes que o usuário veja um bom prompt. Essa seqüência de login não parece tão limpa quanto o método de velocidade travada, mas um usuário em uma conexão de baixa velocidade deve receber uma melhor resposta interativa de programas em tela cheia.

Ao travar a taxa de comunicação de dados de um modem a uma velocidade específica, nenhuma alteração em /etc/gettytab deve ser necessária. No entanto, para uma configuração de velocidade compatível, entradas adicionais podem ser necessárias para definir as velocidades a serem usadas para o modem. Este exemplo configura um modem de 14,4 Kbps com uma velocidade de interface superior de 19,2 Kbps usando conexões de 8 bits sem paridade. Ele configura o `getty` para iniciar a taxa de comunicação para uma conexão V.32bis a 19,2 Kbps, passando por 9600 bps, 2400 bps, 1200 bps, 300 bps e de volta para 19,2 Kbps. O ciclo de taxa de comunicação é implementado com o recurso `nx=` (proxima tabela). Cada linha usa uma entrada `tc=` (continuação de tabela) para selecionar o restante das configurações para uma taxa de dados específica.

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

Para um modem de 28,8 Kbps ou para aproveitar a compactação em um modem de 14,4 Kbps, use uma taxa de comunicação mais alta, conforme mostrado neste exemplo:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

Para uma CPU lenta ou um sistema altamente carregado sem portas seriais baseadas no 16550A, esta configuração pode produzir erros `sio` "silo" a 57,6 Kbps.

A configuração do /etc/ttys é similar a Configurando Entradas de Terminal, mas um argumento diferente é passado para o `getty` e `dialup` é usado para o tipo de terminal. Substitua *xxx* pelo processo `init` que será executado no dispositivo:

```
ttyu0   "/usr/libexec/getty xxx"   dialup on
```

O tipo de terminal `dialup` pode ser alterado. Por exemplo, definir `vt102` como o tipo de terminal padrão permite que os usuários usem a emulação VT102 em seus sistemas remotos.

Para uma configuração de velocidade travada, especifique a velocidade com um tipo válido listado em /etc/gettytab. Este exemplo é para um modem cuja velocidade de porta está travada em 19,2 Kbps:

```
ttyu0   "/usr/libexec/getty std.19200"   dialup on
```

Em uma configuração de velocidade correspondente, a entrada precisa referenciar a entrada inicial apropriada "auto-baud" em /etc/gettytab. Para continuar o exemplo de um modem com velocidade correspondente que começa em 19,2 Kbps, use esta entrada:

```
ttyu0   "/usr/libexec/getty V19200"   dialup on
```

Depois de editar o /etc/ttys, espere até que o modem esteja devidamente configurado e conectado antes de sinalizar o `init`:

```
# kill -HUP 1
```

Modems de alta velocidade, como os modems V.32, V.32bis e V.34, usam hardware (`RTS/CTS`) para controle de fluxo. Use o `stty` para definir a flag de controle de fluxo de hardware para a porta do modem. Este exemplo define a flag `crtscts` na inicialização dos dispositivos COM2 de dial-in e de dial-out:

```
# stty -f /dev/ttyu1.init crtscts
# stty -f /dev/cuau1.init crtscts
```

26.4.2. Solução de problemas

Esta seção fornece algumas dicas para solucionar problemas de um modem dial-up que não se conecta há um sistema FreeBSD.

Conecte o modem ao sistema FreeBSD e inicialize o sistema. Se o modem tiver luzes de indicação de status, observe se o indicador DTR do modem acende quando o prompt `login:` é exibido no console do sistema. Se acender, isso deve significar que o FreeBSD iniciou um processo `getty` na porta de comunicação apropriada e está aguardando o modem aceitar uma chamada.

Se o indicador DTR não acender, faça o login no sistema FreeBSD através do console e digite `ps ax` para ver se o FreeBSD está executando um processo `getty` na porta correta:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200`ttyu0`
```

Se a segunda coluna contiver um `d0` em vez de um `??` e o modem ainda não aceitou uma chamada, isso significa que o `getty` completou sua chamada na porta de comunicações. Isso pode indicar um problema com o cabeamento ou com um modem configurado incorretamente porque o `getty` não deve conseguir abrir a porta de comunicação até que o sinal de detecção da portadora tenha sido declarado pelo modem.

Se nenhum processo `getty` estiver aguardando para abrir a porta, verifique se a entrada da porta está correta no /etc/ttys. Além disso, verifique o /var/log/messages para ver se há alguma mensagem

de log do `init` ou do `getty`.

Em seguida, tente discar para o sistema. Certifique-se de usar 8 bits, sem paridade e 1 bit de stop no sistema remoto. Se um prompt não aparecer imediatamente ou o prompt mostrar lixo, tente pressionar `Enter` uma vez por segundo durante alguns segundos. Se ainda não houver nenhum prompt de `login:`, tente enviar um `BREAK`. Ao usar um modem de alta velocidade, tente discar novamente após travar a velocidade da interface do modem de discagem.

Se ainda não houver o prompt `login:`, verifique novamente o /etc/gettytab e faça um double-check:

- O nome do recurso inicial especificado na entrada em /etc/ttys corresponde ao nome de um recurso em /etc/gettytab.
- Cada entrada `nx=` corresponde a outro nome de recurso gettytab.
- Cada entrada `tc=` corresponde a outro nome de recurso gettytab.

Se o modem no sistema FreeBSD não responder, verifique se o modem está configurado para atender o telefone quando o DTR é ativado. Se o modem parece estar configurado corretamente, verifique se a linha DTR é ativada, verificando as luzes indicadoras do modem.

Se ainda assim não funcionar, tente enviar um e-mail para a lista de discussão de perguntas gerais do FreeBSD descrevendo o modem e o problema.

26.5. Serviço de Dial-in

A seguir, dicas para fazer com que o host conecte-se através do modem a outro computador. Isto é apropriado para estabelecer uma sessão de terminal com um host remoto.

Esse tipo de conexão pode ser útil para obter um arquivo na Internet, caso haja problemas no uso do PPP. Se o PPP não estiver funcionando, use a sessão do terminal para enviar por FTP o arquivo necessário. Em seguida, use o zmodem para transferi-lo para a máquina.

26.5.1. Usando um Modem Stock Hayes

Um dialer Hayes genérico está incorporado no `tip`. Use `at=hayes` em /etc/remote.

O driver Hayes não é inteligente o suficiente para reconhecer alguns dos recursos avançados de mensagens de modems mais recentes como `BUSY`, `NO DIALTONE` ou `CONNECT 115200`. Desative essas mensagens ao usar o `tip` com o `ATX0&W`.

O tempo limite de discagem para o `tip` é de 60 segundos. O modem deve usar algo menor, ou então o `tip` irá achar que existe um problema de comunicação. Tente usar `ATS7=45&W`.

26.5.2. Usando comandos `AT`

Crie uma entrada "direct" em /etc/remote. Por exemplo, se o modem estiver conectado à primeira porta serial, /dev/cuau0, use a seguinte linha:

```
cuau0:dv=/dev/cuau0:br#19200:pa=none
```

Use a taxa mais alta de bps que o modem suporta no recurso `br`. Em seguida, digite `tip cuau0` para conectar-se ao modem.

Ou use `cu` como `root` com o seguinte comando:

```
# cu -lline -sspeed
```

line é a porta serial, tal como /dev/cuau0, e *speed* é a velocidade, tal como `57600` . Quando terminar de digitar os comandos AT, digite `~.` para sair.

26.5.3. O Sinal `@` Não Funciona

O `@` na capability do número de telefone diz ao `tip` para procurar em /etc/phones um número de telefone. Mas, o sinal. `@` também é um caractere especial em arquivos de capablity como o /etc/remote, então ele precisa ser escapado com uma barra invertida:

```
pn=\@
```

26.5.4. Discando a Partir da Linha de Comando

Coloque uma entrada "genérica" em /etc/remote. Por exemplo:

```
tip115200|Dial any phone number at 115200 bps:\
        :dv=/dev/cuau0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
        :dv=/dev/cuau0:br#57600:at=hayes:pa=none:du:
```

Isto deve funcionar agora:

```
# tip -115200 5551234
```

Usuários que preferem comando `cu` ao `tip`, podem usar uma entrada `cu` genérica:

```
cu115200|Use cu to dial any number at 115200bps:\
        :dv=/dev/cuau1:br#57600:at=hayes:pa=none:du:
```

e digite:

```
# cu 5551234 -s 115200
```

26.5.5. Definindo a Taxa de bps

Coloque uma entrada para `tip1200` ou `cu1200`, mas vá em frente e use qualquer taxa bps apropriada

com o capability `br`. O `tip` acha que um bom padrão é de 1200 bps, e é por isso que ele procura por uma entrada `tip1200`. No entanto, 1200 bps não precisa ser usado.

26.5.6. Acessando um Conjunto de Hosts por Meio de um Servidor de Terminal

Em vez de esperar até conectar-se e digitar `CONNECT_host_` a cada vez, use o recurso `cm` do `tip`. Por exemplo, estas entradas no /etc/remote permitirão que você digite `tip pain` ou `tip muffin` para conectar-se aos hosts `pain` ou `muffin` e `tip deep13` para conectar ao servidor de terminal.

```
pain|pain.deep13.com|Forrester's machine:\
        :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        :dv=/dev/cuau2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

26.5.7. Usando Mais de Uma Linha com `tip`

Isto geralmente é um problema em que uma universidade tem várias linhas de modems e vários milhares de estudantes tentando usá-las.

Faça uma entrada em /etc/remote e use `@` para o recurso `pn`:

```
big-university:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuau3:br#9600:at=courier:du:pa=none:
```

Em seguida, liste os números de telefone em /etc/phones:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

O `tip` tentará cada número na ordem listada, depois desistirá. Para continuar tentando, execute o `tip` em um loop `while`.

26.5.8. Usando o Caractere de Force

O `Ctrl` + `P` é o caracter "force" padrão, usado para dizer ao `tip` que o próximo caractere é um dado literal. O caractere force pode ser definido para qualquer outro caractere com o escape `~s`, o que significa "definir uma variável."

Digite `~sforce=single-char` seguido por uma nova linha. Onde *single-char* é qualquer caractere

único. Se o *single-char* for omitido, o caractere force será o caractere nulo, que é acessado digitando-se `Ctrl` + `2` ou `Ctrl` + `Espace`. Um valor muito bom para *single-char* é o `Shift` + `Ctrl` + `6`, que é usado apenas em alguns servidores de terminal.

Para alterar o caractere force, especifique o seguinte em ~/.tiprc:

```
force=single-char
```

26.5.9. Caracteres Maiúsculos

Isso acontece quando o `Ctrl` + `A` é pressionado, o qual corresponde ao `tip` "raise character", especialmente concebido para pessoas coma tecla de caps-lock quebrada. Use `~s` para definir `raisechar` para algo razoável. Ele pode ser configurado para ser o mesmo que o caractere de force, se nenhum recurso for usado.

Aqui está um exemplo do ~/.tiprc para os usuários do Emacs que precisam digitar `Ctrl` + `2` e `Ctrl` + `A`:

```
force=^^
raisechar=^^
```

O `^^` é `Shift` + `Ctrl` + `6`.

26.5.10. Transferências de Arquivos com `tip`

Ao falar com outro sistema operacional semelhante ao UNIX™, os arquivos podem ser enviados e recebidos usando `~p` (put) e `~t` (take). Esses comandos executam `cat` e `echo` no sistema remoto para aceitar e enviar arquivos. A sintaxe é:

`~p` local-file [remote-file]

`~t` remote-file [local-file]

Não há verificação de erros, então outro protocolo, como zmodem, provavelmente deveria ser usado.

26.5.11. Usando o zmodem com o `tip`?

Para receber arquivos, inicie o programa de envio no terminal remoto. Em seguida, digite `~C rz` para começar a recebê-los localmente.

Para enviar arquivos, inicie o programa de recebimento no terminal remoto. Em seguida, digite `~C sz files` para enviá-los ao sistema remoto.

26.6. Configurando o Console Serial

O FreeBSD tem a capacidade de inicializar um sistema com um terminal burro em uma porta serial como um console. Esta configuração é útil para administradores de sistemas que desejam instalar o

FreeBSD em máquinas que não possuem teclado ou monitor conectados, e desenvolvedores que desejam depurar o kernel ou drivers de dispositivos.

Como descrito em O processo de inicialização do FreeBSD, o FreeBSD emprega um bootstrap de três estágios. Os dois primeiros estágios estão no código do bloco de inicialização que é armazenado no início da slice do FreeBSD no disco de inicialização. O bloco de inicialização, em seguida, carrega e executa o carregador de boot como o código do terceiro estágio.

Para configurar a inicialização a partir de um console serial, o código do bloco de inicialização, o código do carregador de inicialização e o kernel precisam ser configurados.

26.6.1. Configuração Rápida do Console Serial

Esta seção fornece uma visão geral rápida da configuração do console serial. Este procedimento pode ser usado quando o terminal burro é conectado ao COM1.

Procedure: Configurando um Console Serial no COM1

1. Conecte o cabo serial ao COM1 e ao terminal de controle.
2. Para configurar mensagens de inicialização para exibição no console serial, emita o seguinte comando como o superusuário:

```
# echo 'console="comconsole"' >> /boot/loader.conf
```

3. Edite /etc/ttys e mude `off` para `on` e `dialup` para `vt100` para a entrada ttyu0. Caso contrário, uma senha não será necessária para conectar-se através do console serial, resultando em uma potencial brecha de segurança.
4. Reinicialize o sistema para ver se as alterações entraram em vigor.

Se uma configuração diferente for necessária, consulte a próxima seção para obter uma explicação de configuração mais detalhada.

26.6.2. Configuração do console serial em profundidade

Esta seção fornece uma explicação mais detalhada das etapas necessárias para configurar um console serial no FreeBSD.

Procedure: Configurando um Console Serial

1. Prepare um cabo serial.

   Use um cabo de null-modem ou um cabo serial padrão e um adaptador de null-modem. Veja Cabos Serial e Portas para uma discussão sobre cabos seriais.

2. Desconecte o teclado.

Muitos sistemas detectam o teclado durante o Power-On Self-Test (POST) e geram um erro se o teclado não for detectado. Algumas máquinas recusarão a inicialização até que o teclado esteja conectado.

Se o computador reclamar do erro, mas inicializar de qualquer maneira, nenhuma outra configuração será necessária.

Se o computador se recusar a inicializar sem um teclado conectado, configure o BIOS para que ele ignore este erro. Consulte o manual da placa-mãe para obter detalhes sobre como fazer isso.



Tente configurar o teclado para "Not installed" no BIOS. Esta configuração diz ao BIOS para não detectar um teclado ao ligar, então ele não deve reclamar se o teclado estiver ausente. Se essa opção não estiver presente no BIOS, procure uma opção "Halt on Error". Configurando isto para "All but Keyboard" ou para "No Errors" terá o mesmo efeito.

Se o sistema tiver um mouse PS/2™, desconecte-o também. Os mouses PS/2™ compartilham algum hardware com o teclado e, deixar o mouse conectado, pode enganar o sistema e faze-lo pensar que o teclado ainda está lá.



Embora a maioria dos sistemas inicialize sem um teclado, alguns não inicializarão sem um adaptador gráfico. Alguns sistemas podem ser configurados para inicializar sem nenhum adaptador gráfico alterando a configuração do "graphics adapter" na configuração BIOS para "Not installed". Outros sistemas não suportam esta opção e recusarão a inicialização se não houver hardware de exibição no sistema. Com estas máquinas, deixe algum tipo de placa gráfica ligada, mesmo que seja apenas uma placa mono lixo. Um monitor não precisa ser conectado.

3. Conecte um terminal burro, um computador antigo com um programa de modem ou a porta serial de outra máquina UNIX™ na porta serial da máquina freebsd.
4. Adicione as entradas `hint.sio.*` apropriadas para o /boot/device.hints para a porta serial. Algumas placas com várias portas também exigem opções de configuração do kernel. Consulte sio(4) para obter as opções necessárias e os device hints para cada porta serial suportada.
5. Crie o boot.config no diretório raiz da partição `a` na unidade de inicialização.

   Este arquivo instrui o código do bloco de inicialização como inicializar o sistema. Para ativar o console serial, uma ou mais das seguintes opções são necessárias. Ao usar várias opções, inclua todas elas na mesma linha:

   `-h`

   Alterna entre os consoles interno e serial. Use isso para alternar dispositivos do console. Por exemplo, para inicializar a partir do console (vídeo) interno, use `-h` para direcionar o carregador de boot e o kernel para usar a porta serial como seu dispositivo de console. Alternativamente, para inicializar a partir da porta serial, use `-h` para dizer ao gerenciador de inicialização e ao kernel para usar a exibição de vídeo como o console.

`-D`

Alterna entre as configurações de console única e dupla. Na configuração única, o console será o console interno (exibição de vídeo) ou a porta serial, dependendo do estado de `-h`. Na configuração do console duplo, a exibição de vídeo e a porta serial se tornarão o console ao mesmo tempo, independentemente do estado de `-h`. No entanto, a configuração do console duplo entrará em vigor somente enquanto o bloco de inicialização estiver em execução. Depois que o gerenciador de boot obtiver controle, o console especificado por `-h` se tornará o único console.

`-P`

Faz com que o bloco de inicialização avalie o teclado. Se nenhum teclado for encontrado, as opções `-D` e `-h` serão automaticamente definidas.



Devido a restrições de espaço na versão atual dos blocos de inicialização, `-P` é capaz de detectar somente teclados estendidos. Teclados com menos de 101 teclas e sem as teclas F11 e F12 podem não ser detectados. Teclados em alguns laptops podem não ser encontrados corretamente devido a essa limitação. Se este for o caso, não use `-P`.

Use `-P` para selecionar o console automaticamente ou `-h` para ativar o console serial. Consulte boot(8) e boot.config(5) para maiores detalhes.

As opções, exceto para `-P`, são passadas para o carregador de boot. O gerenciador de boot determinará se o vídeo interno ou a porta serial deve se tornar o console examinando o estado de `-h`. Isto significa que se `-D` for especificado mas `-h` não estiver especificado no /boot.config, a porta serial pode ser usada como console somente durante o bloco de inicialização, pois o gerenciador de inicialização usará a exibição de vídeo interna como o console.

6. Inicialize a máquina.

Quando o FreeBSD inicia, os blocos de inicialização mostram o conteúdo do /boot.config para o console. Por exemplo:

```
/boot.config: -P
Keyboard: no
```

A segunda linha aparece somente se `-P` aparecer no /boot.config e indica a presença ou ausência do teclado. Estas mensagens vão para o console serial ou interno, ou ambos, dependendo da opção em /boot.config:

Opções	Mensagem vai para
nenhum	console interno
`-h`	console serial
`-D`	consoles seriais e internos
`-Dh`	consoles seriais e internos

Opções	Mensagem vai para
`-P`, teclado presente	console interno
`-P`, teclado ausente	console serial

Após a mensagem, haverá uma pequena pausa antes que os blocos de inicialização continuem carregando o carregador de boot e antes que qualquer outra mensagem seja impressa no console. Em circunstâncias normais, não há necessidade de interromper os blocos de inicialização, mas pode-se fazê-lo para garantir que as coisas sejam configuradas corretamente.

Pressione qualquer tecla, exceto `Enter`, no console para interromper o processo de inicialização. Os blocos de inicialização então solicitarão mais ações:

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Verifique se a mensagem acima aparece no console serial ou interno, ou em /boot.config. Se a mensagem aparecer no console correto, pressione `Enter` para continuar o processo de inicialização.

Se não houver nenhum prompt no terminal serial, algo está errado com as configurações. Digite `-h` e depois `Enter` ou `Return` para informar o bloco de inicialização (e depois o carregador de inicialização e o kernel) para escolher a porta serial para o console. Quando o sistema estiver ativo, volte e verifique o que deu errado.

Durante o terceiro estágio do processo de inicialização, ainda é possível alternar entre o console interno e o console serial definindo as variáveis de ambiente apropriadas no carregador de inicialização. Veja loader(8) para obter maiores informações.

Esta linha no /boot/loader.conf ou /boot/loader.conf.local configura o carregador de inicialização e o kernel para enviar suas mensagens de inicialização para o console serial, independentemente das opções no /boot.config:

```
console="comconsole"
```



Esta linha deve ser a primeira linha do /boot/loader.conf para que as mensagens de boot sejam exibidas no console serial o mais cedo possível.

Se essa linha não existir, ou se estiver definida como `console="vidconsole"`, o carregador de inicialização e o kernel usarão qualquer console indicado por `-h` no bloco de inicialização. Veja loader.conf(5) para maiores informações.

No momento, o carregador de boot não tem nenhuma opção equivalente a `-P` no bloco de inicialização, e não há provisão para selecionar automaticamente o

console interno e o console serial com base na presença do teclado.



Embora não seja obrigatório, é possível fornecer um prompt `login` na linha serial. Para configurar isto, edite a entrada para a porta serial em /etc/ttys usando as instruções em Configuração do Terminal. Se a velocidade da porta serial tiver sido alterada, altere `std.9600` para corresponder à nova configuração.

26.6.3. Defina uma velocidade de porta serial mais rápida

Por padrão, as configurações da porta serial são 9600 baud, 8 bits, sem paridade e 1 bit de parada. Para alterar a velocidade do console padrão, use uma das seguintes opções:

- Edite o /etc/make.conf e configure o `BOOT_COMCONSOLE_SPEED` para a nova velocidade do console. Em seguida, recompile e instale os blocos de inicialização e o carregador de boot:

```
# cd /sys/boot
# make clean
# make
# make install
```

  Se o console serial estiver configurado de alguma outra maneira que não seja inicializando com `-h`, ou se o console serial usado pelo kernel for diferente daquele usado pelos blocos de inicialização, adicione a seguinte opção, com a velocidade desejada, em um arquivo de configuração de kernel personalizado e compile um novo kernel:

```
options CONSPEED=19200
```

- Acrescente a opção de inicialização `-S_19200_` ao /boot.config, substituindo *19200* pela velocidade a ser utilizada.
- Adicione as seguintes opções ao /boot/loader.conf. Substitua *115200* pela velocidade de uso.

```
boot_multicons="YES"
boot_serial="YES"
comconsole_speed="115200"
console="comconsole,vidconsole"
```

26.6.4. Entrando no Depurador DDB da Linha Serial

Para configurar a capacidade de inserir o depurador de kernel no console serial, inclua as seguintes opções em um arquivo de configuração de kernel personalizado e compile o kernel usando as instruções em Configurando o kernel do FreeBSD. Observe que, embora isso seja útil para diagnósticos remotos, também é perigoso se um BREAK espúrio for gerado na porta serial. Consulte ddb(4) e ddb(8) para mais informações sobre o depurador do kernel.

```
options BREAK_TO_DEBUGGER
options DDB
```

Capítulo 27. PPP

27.1. Sinopse

O FreeBSD suporta o protocolo Point-to-Point (PPP) que pode ser usado para estabelecer uma conexão de rede ou Internet usando um modem dial-up. Este capítulo descreve como configurar serviços de comunicação baseados em modem no FreeBSD.

Depois de ler este capítulo, você saberá:

- Como configurar, usar e solucionar problemas de uma conexão PPP.
- Como configurar o PPP sobre Ethernet (PPPoE).
- Como configurar o PPP sobre ATM (PPPoA).

Antes de ler este capítulo, você deve:

- Estar familiarizado com a terminologia básica de rede.
- Entender os conceitos básicos e o propósito de uma conexão dial-up e PPP.

27.2. Configurando o PPP

O FreeBSD fornece suporte nativo para gerenciamento conexões dial-up PPP usando ppp(8). O kernel padrão do FreeBSD fornece suporte para o tun, que é usado para interagir com um hardware de modem. A configuração é executada editando pelo menos um arquivo de configuração, e exemplos destes arquivos de configuração são fornecidos com o sistema. Finalmente, o `ppp` é usado para iniciar e gerenciar conexões.

Para usar uma conexão PPP, os seguintes itens são necessários:

- Uma conta dial-up com um provedor de serviços de Internet (ISP).
- Um modem dial-up.
- O número de discagem para o ISP.
- O nome de usuário e a senha atribuídos pelo ISP.
- O endereço IP de um ou mais servidores de DNS. Normalmente, o ISP fornece estes endereços. Caso contrário, o FreeBSD pode ser configurado para usar a negociação de DNS.

Se alguma das informações necessárias estiver faltando, entre em contato com o ISP.

As seguintes informações podem ser fornecidas pelo ISP, mas não são necessárias:

- O endereço IP do gateway padrão. Se esta informação for desconhecida, o ISP fornecerá automaticamente o valor correto durante a configuração da conexão. Ao configurar o PPP no FreeBSD, este endereço é chamado de `HISADDR`.
- A máscara de sub-rede. Se o ISP não tiver fornecido um, `255.255.255.255` será usado no arquivo de configuração do ppp(8). *

Se o ISP tiver atribuído um endereço IP estático e um nome de host, ele deverá ser inserido no arquivo de configuração. Caso contrário, essas informações serão fornecidas automaticamente durante a configuração da conexão.

O restante desta seção demonstra como configurar o FreeBSD para cenários de conexão PPP comuns. O arquivo de configuração requerido é o /etc/ppp/ppp.conf e arquivos de exemplos adicionais estão disponíveis em /usr/shared/examples/ppp/.



Ao longo desta seção, muitos dos exemplos de arquivos exibem números de linha. Esses números de linha foram adicionados para facilitar o acompanhamento da discussão e não devem ser colocados no arquivo real.

Ao editar um arquivo de configuração, o recuo adequado é importante. Linhas que terminam em um `:` iniciam na primeira coluna (início da linha) enquanto todas as outras linhas devem ser recuadas como mostrado usando espaços ou tabulações.

27.2.1. Configuração básica

Para configurar uma conexão PPP, primeiro edite o /etc/ppp/ppp.conf com as informações de discagem do ISP. Este arquivo é descrito da seguinte maneira:

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION
4       set device /dev/cuau0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set timeout 300
16      set ifaddr x.x.x.x/0 y.y.y.y/0 255.255.255.255 0.0.0.0
17      add default HISADDR
```

Linha 1

Identifica a entrada `default`. Os comandos nesta entrada (linhas 2 a 9) são executados automaticamente quando o `ppp` é executado.

Linha 2

Ativa os parâmetros de log detalhado para testar a conexão. Uma vez que a configuração esteja funcionando satisfatoriamente, esta linha deve ser reduzida para:

```
set log phase tun
```

Linha 3

Exibe a versão do ppp(8) para o software PPP em execução no outro lado da conexão.

Linha 4

Identifica o dispositivo ao qual o modem está conectado, onde COM1 é /dev/cuau0 e COM2 é /dev/cuau1.

Linha 5

Define a velocidade de conexão. Se `115200` não funcionar em um modem mais antigo, tente `38400` em seu lugar.

Linhas 6 & 7

A string de discagem escrita como na sintaxe de envio e espera. Consulte chat(8) para obter maiores informações.

Observe que esse comando continua na próxima linha para facilitar a leitura. Qualquer comando no ppp.conf pode fazer isso se o último caractere na linha for `\`.

Linha 8

Define o tempo ocioso limite do link em segundos.

Linha 9

Instrui o peer para confirmar as configurações de DNS. Se a rede local estiver executando seu próprio servidor DNS, essa linha deve ser comentada, adicionando um `#` no início da linha ou removendo-a.

Linha 10

Uma linha em branco para facilitar a leitura. Linhas em branco são ignoradas pelo ppp(8).

Linha 11

Identifica uma entrada chamada `provider`. Isto pode ser alterado para o nome do ISP, para que `load ISP` possa ser usado para iniciar a conexão.

Linha 12

Use o número de telefone para o ISP. Vários números de telefone podem ser especificados usando os dois-pontos (`:`) ou o caractere pipe (`|`) como um separador. Para rotacionar entre os números, use dois pontos. Para sempre tentar discar o primeiro número primeiro e usar os outros números apenas se o primeiro número falhar, use o caractere pipe. Sempre coloque todo o conjunto de números de telefone entre aspas (`"`) para evitar falhas de discagem.

Linhas 13 & 14

Use o nome de usuário e senha para o ISP.

Linha 15

Define o tempo ocioso limite padrão em segundos para a conexão. Neste exemplo, a conexão

será fechada automaticamente após 300 segundos de inatividade. Para evitar um tempo limite, defina esse valor como zero.

Linha 16

Define os endereços da interface. Os valores usados dependem de se um endereço IP estático foi obtido do ISP ou se ele negocia um endereço IP dinâmico durante a conexão.

Se o ISP tiver alocado um endereço IP estático e um gateway padrão, substitua *xxxx* pelo endereço IP estático e substitua *yyyy* com o endereço IP do gateway padrão. Se o ISP tiver fornecido apenas um endereço IP estático sem um endereço de gateway, substitua *yyyy* por `10.0.0.2/0`.

Se o endereço IP mudar sempre que uma conexão for feita, altere essa linha para o seguinte valor. Isso diz ao ppp(8)para usar o IP Configuration Protocol (IPCP) para negociar um endereço IP dinâmico:

```
set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255 0.0.0.0
```

Linha 17

Mantenha esta linha como está, pois ela adiciona uma rota padrão ao gateway. O `HISADDR` será automaticamente substituído pelo endereço do gateway especificado na linha 16. É importante que esta linha apareça depois da linha 16.

Dependendo se o ppp(8) for iniciado manualmente ou automaticamente, um arquivo /etc/ppp/ppp.linkup também pode precisar ser criado, contendo as seguintes linhas. Este arquivo é requerido ao executar o `ppp` no modo `-auto`. Este arquivo é usado após a conexão ter sido estabelecida. Neste ponto, o endereço IP será atribuído e agora será possível adicionar as entradas da tabela de roteamento. Ao criar este arquivo, certifique-se de que o *provider* corresponda ao valor demonstrado na linha 11 do ppp.conf.

```
provider:
      add default HISADDR
```

Este arquivo também é necessário quando o endereço do gateway padrão é "adivinhado" em uma configuração de endereço IP estático. Neste caso, remova a linha 17 do ppp.conf e crie o /etc/ppp/ppp.linkup com as duas linhas acima. Outros exemplos para este arquivo podem ser encontrados em /usr/shared/examples/ppp/.

Por padrão, o `ppp` deve ser executado como `root`. Para alterar esse padrão, adicione a conta do usuário que deve executar o `ppp` ao grupo `network` em /etc/group .

Em seguida, conceda ao usuário acesso a uma ou mais entradas em /etc/ppp/ppp.conf com `allow`. Por exemplo, para dar a permissão para os usuários `fred` e `mary` somente à entrada `provider:`, inclua esta linha para a seção `provider:`:

```
allow users fred mary
```

Para fornecer aos usuários especificados acesso a todas as entradas, coloque essa linha na seção `default`.

27.2.2. Configuração Avançada

É possível configurar o PPP para fornecer endereços de servidores DNS e NetBIOS sob demanda.

Para habilitar estas extensões com o PPP versão 1.x, as seguintes linhas podem ser adicionadas à seção relevante do /etc/ppp/ppp.conf.

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

E para o PPP versão 2 e acima:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Isso informará aos clientes os endereços do servidor de nomes primário e secundário e um host do servidor de nomes NetBIOS.

Na versão 2 e acima, se a linha `set dns` for omitida, o PPP usará os valores encontrados em /etc/resolv.conf.

27.2.2.1. Autenticação PAP e CHAP

Alguns ISPs configuram seu sistema para que a parte de autenticação da conexão seja feita usando um dos mecanismos de autenticação PAP ou CHAP. Se este for o caso, o ISP não exibirá um prompt `login:` na conexão, mas começará a falar PPP imediatamente.

O PAP é menos seguro que o CHAP, mas a segurança normalmente não é um problema aqui, pois as senhas, embora sejam enviadas como texto simples com o PAP, estão sendo transmitidas apenas por uma linha serial. Não há muito espaço para crackers "escutarem".

As seguintes alterações devem ser feitas:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

Linha 13

Esta linha especifica o nome de usuário do PAP/CHAP. Insira o valor correto para *MyUserName*.

Linha 14

Esta linha especifica a senha PAP/CHAP . Insira o valor correto para *MyPassword*. Você pode

querer adicionar uma linha adicional, como:

```
16      accept PAP
```

ou

```
16      accept CHAP
```

para tornar óbvio que essa é a intenção, mas o PAP e o CHAP são aceitos por padrão.

Linha 15

O ISP normalmente não exigirá um login no servidor ao usar o PAP ou o CHAP. Portanto, desabilite a string "set login".

27.2.2.2. Usando a funcionalidade de conversão de endereços de rede (NAT) do PPP

O PPP tem a capacidade de usar o NAT interno sem recursos de diverting do kernel. Esta funcionalidade pode ser ativada pela seguinte linha no /etc/ppp/ppp.conf:

```
nat enable yes
```

Como alternativa, o NAT pode ser ativado pela opção de linha de comando `-nat`. Há também uma opção no /etc/rc.conf chamada `ppp_nat`, que é ativada por padrão.

Ao usar este recurso, pode ser útil incluir as seguintes opções no /etc/ppp/ppp.conf para habilitar o encaminhamento de conexões de entrada:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

ou para não confiar em nenhuma conexão de entrada

```
nat deny_incoming yes
```

27.2.3. Configuração final do sistema

Embora o `ppp` agora esteja configurado, algumas edições ainda precisam ser feitas no /etc/rc.conf.

Trabalhando de cima para baixo neste arquivo, certifique-se de que a linha `hostname=` esteja configurada:

```
hostname="foo.example.com"
```

Se o ISP tiver fornecido um nome de host e um endereço IP estático, use este nome como o nome do host.

Procure pela variável `network_interfaces`. Para configurar o sistema para discar para o ISP sob demanda, certifique-se de que o dispositivo tun0 esteja adicionado à lista, caso contrário, remova-o.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```



A variável `ifconfig_tun0` deve estar vazia, e um arquivo chamado /etc/start_if.tun0 deve ser criado. Este arquivo deve conter a linha:

```
ppp -auto mysystem
```

Este script é executado no momento da configuração da rede, iniciando o daemon do ppp no modo automático. Se esta máquina funcionar como um gateway, considere incluir a opção `-alias`. Consulte a página de manual para maiores detalhes.

Certifique-se de que o programa roteador está configurado para `NO` com a seguinte linha em /etc/rc.conf:

```
router_enable="NO"
```

É importante que o daemon `routed` não seja iniciado, pois o `routed` tende a excluir as entradas da tabela de roteamento padrão criadas pelo `ppp`.

É provavelmente uma boa idéia garantir que a linha `sendmail_flags` não inclua a opção `-q`, caso contrário o `sendmail` tentará fazer uma pesquisa de rede de vez em quando, possivelmente fazendo com que sua máquina disque. Você pode tentar:

```
sendmail_flags="-bd"
```

A desvantagem é que o `sendmail` é forçado a reexaminar a fila de mensagens sempre que o link ppp subir. Para automatizar isso, inclua `!Bg` no ppp.linkup:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

Uma alternativa é configurar um "dfilter" para bloquear o tráfego SMTP. Consulte os arquivos de exemplo para maiores detalhes.

27.2.4. Usando o ppp

Tudo o que resta é reiniciar a máquina. Após a reinicialização, digite:

```
# ppp
```

e, em seguida, o `dial provider` para iniciar a sessão PPP ou para configurar o `ppp` para estabelecer sessões automaticamente quando houver tráfego de saída e o start_if .tun0 não existir, digite:

```
# ppp -auto provider
```

É possível falar com o programa `ppp` enquanto ele está sendo executado em segundo plano, mas somente se uma porta de diagnóstico adequada tiver sido configurada. Para fazer isso, adicione a seguinte linha à configuração:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

Isso fará com que o PPP escute no soquete de domínio UNIX™ especificado, solicitando aos clientes a senha especificada antes de permitir o acesso. O `%d` no nome é substituído pelo número do dispositivo tun que está em uso.

Uma vez que um socket tenha sido configurado, o programa pppctl(8) pode ser usado em scripts que desejam manipular o programa em execução.

27.2.5. Configurando serviços de discagem

A Serviço Dial-in fornece uma boa descrição sobre como ativar serviços dial-up usando o getty(8).

Uma alternativa para o `getty` é o port comms/mgetty+sendfax), uma versão mais inteligente do `getty` projetada com as linhas dial-up em mente.

As vantagens de usar o `mgetty` é que ele *fala* ativamente com os modems, o que significa que se a porta estiver desligada no /etc/ttys então o modem não irá atender o telefone.

Versões posteriores do `mgetty` (da 0.99beta em diante) também suportam a detecção automática de fluxos PPP, permitindo acesso ao servidor de clientes sem script.

Consulte a URL http://mgetty.greenie.net/doc/mgetty_toc.html para maiores informações sobre o `mgetty`.

Por padrão, o port comms/mgetty+sendfax vem com a opção `AUTO_PPP` ativada permitindo que o `mgetty` detecte a fase LCP das conexões PPP e crie automaticamente um shell ppp. No entanto, como a sequência de login/senha padrão não ocorre, é necessário autenticar os usuários usando o PAP ou o CHAP.

Esta seção assume que o usuário compilou com sucesso e instalou o port comms/mgetty+sendfax em seu sistema.

Assegure-se de que o /usr/local/etc/mgetty+sendfax/login.config tenha o seguinte:

```
/AutoPPP/ -     - /etc/ppp/ppp-pap-dialup
```

Isto diz ao `mgetty` para executar o ppp-pap-dialup para conexões PPP detectadas.

Crie um arquivo executável chamado /etc/ppp/ppp-pap-dialup contendo o seguinte:

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Para cada linha dial-up ativada em /etc/ttys, crie uma entrada correspondente em /etc/ppp/ppp.conf. Isso irá coexistir com as definições que criamos acima.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Cada usuário que fizer login com este método precisará ter um nome de usuário/senha em /etc/ppp/ppp.secret ou, como alternativa, adicione a seguinte opção para autenticar os usuários via PAP a partir de /etc/passwd.

```
enable passwdauth
```

Para atribuir à alguns usuários um endereço de IP estático , especifique o endereço como o terceiro argumento em /etc/ppp/ppp.secret. Consulte o /usr/shared/examples/ppp/ppp.secret.sample para exemplos.

27.3. Solução de problemas de conexões PPP

Esta seção aborda alguns problemas que podem surgir ao usar PPP em uma conexão de modem. Alguns ISPs apresentam o prompt `ssword` enquanto outros apresentam `password`. Se o script `ppp` não for escrito de acordo, a tentativa de login falhará. A maneira mais comum de depurar as conexões `ppp` é conectando manualmente conforme descrito nesta seção.

27.3.1. Verifique os Device Nodes

Ao usar um kernel personalizado, certifique-se de incluir a seguinte linha no arquivo de configuração do kernel:

```
device   uart
```

O dispositivo uart já está incluído no kernel `GENERIC`, portanto, nenhuma etapa adicional é

necessária neste caso. Basta verificar a saída do dmesg para o dispositivo do modem com:

```
# dmesg | grep uart
```

Isso deve exibir alguma saída pertinente sobre os dispositivos uart. Estas são as portas COM que precisamos. Se o modem funcionar como uma porta serial padrão, ele deve estar listado em uart1 ou COM2. Nesse caso, uma recompilação do kernel não é necessária. Ao fazer a verificação, se o modem estiver em uart1, o dispositivo do modem será /dev/cuau1.

27.3.2. Conectando Manualmente

Conectar-se à Internet controlando manualmente o ppp é rápido, fácil e uma ótima maneira de depurar uma conexão ou simplesmente obter informações sobre como o ISP trata as conexões ppp do cliente. Vamos iniciar o PPP na linha de comando. Note que em todos os nossos exemplos nós usaremos *example* como o nome do host da máquina rodando o PPP. Para iniciar o ppp:

```
# ppp
```

```
ppp ON example> set device /dev/cuau1
```

Este segundo comando define o dispositivo do modem como cuau1.

```
ppp ON example> set speed 115200
```

Isso define a velocidade de conexão para 115.200 kbps.

```
ppp ON example> enable dns
```

Isto diz ao ppp para configurar o resolver e adicionar as linhas do servidor de nomes ao /etc/resolv.conf. Se o ppp não puder determinar o nome do host, ele poderá ser configurado manualmente mais tarde.

```
ppp ON example> term
```

Isso alterna para o modo de "terminal" para controlar manualmente o modem.

```
deflink: Entering terminal mode on /dev/cuau1
type '~h' for help
```

```
 at
OK
```

```
 atdt123456789
```

Use o comando `at` para inicializar o modem, então use o comando `atdt` e o número o ISP para iniciar o processo de discagem.

```
CONNECT
```

Confirmação da conexão, se tivermos problemas de conexão, não relacionados ao hardware, aqui é onde tentaremos resolvê-los.

```
ISP Login:myusername
```

Nesse prompt, responda com o nome de usuário fornecido pelo ISP.

```
ISP Pass:mypassword
```

Nesse prompt, responda com a senha fornecida pelo ISP. Assim como ocorre ao se logar no FreeBSD, a senha não será exibida quando você a digitar.

```
Shell or PPP:ppp
```

Dependendo do ISP, este aviso pode não aparecer. Em caso afirmativo, ele está perguntando se deve usar um shell no provedor ou iniciar o `ppp`. Neste exemplo, o `ppp` foi selecionado para estabelecer uma conexão com a Internet.

```
Ppp ON example>
```

Observe que neste exemplo o primeiro `p` foi capitalizado. Isso mostra que nós nos conectamos com sucesso ao ISP.

```
PPp ON example>
```

Nós nos autenticamos com sucesso com nosso ISP e estamos aguardando que o endereço IP seja atribuído.

```
PPP ON example>
```

Fizemos a negociação de um endereço IP e concluímos nossa conexão com êxito.

```
PPP ON example>add default HISADDR
```

Aqui nós adicionamos nossa rota padrão, precisamos fazer isso antes de podermos conversar com o mundo externo, já que atualmente a única conexão estabelecida é com o peer. Se isso falhar devido a rotas existentes, coloque o caractere `!` na frente do `add`. Alternativamente, defina isso antes de fazer a conexão real e ele negociará uma nova rota de acordo.

Se tudo correu bem, agora deveríamos ter uma conexão ativa com a Internet, que poderia ser colocada em segundo plano usando `CTRL` + `z` Se o `PPP` retornar para `ppp`, a conexão será perdida. É bom saber isso porque mostra o status da conexão. Os P maiúsculos representam uma conexão com o ISP e os p minúsculos mostram que a conexão foi perdida.

27.3.3. Depuração

Se uma conexão não puder ser estabelecida, desligue o fluxo de hardware CTS/RTS usando `set ctsrts off`. Normalmente este é o problema quando nos conectamos há alguns servidores de terminal com PPP, onde o PPP trava quando tenta gravar dados no link de comunicação e aguarda um Clear To Send (CTS), sinal que pode nunca vir. Ao usar esta opção, inclua `set accmap`, pois isso pode ser necessário para evitar que o hardware dependa de passar certos caracteres de ponta a ponta, na maioria das vezes XON/XOFF. Consulte ppp(8) para obter maiores informações sobre essa opção e como ela é usada.

Um modem mais antigo pode precisar de `set parity even`. A paridade é definida como none por padrão, mas é usada para verificação de erros com um grande aumento no tráfego, em modems mais antigos.

O PPP pode não retornar ao modo de comando, que geralmente é um erro de negociação em que o ISP está aguardando a negociação começar. Neste ponto, usando `~p` forçará o ppp a começar a enviar as informações de configuração.

Se um prompt de login nunca aparecer, a autenticação PAP ou CHAP provavelmente será necessária. Para usar PAP ou CHAP, adicione as seguintes opções ao PPP antes de entrar no modo terminal:

```
ppp ON example> set authname myusername
```

Onde *myusername* deve ser substituído pelo nome de usuário que foi atribuído pelo ISP.

```
ppp ON example> set authkey mypassword
```

Onde *mypassword* deve ser substituído pela senha que foi atribuída pelo ISP.

Se uma conexão for estabelecida, mas não conseguir encontrar nenhum nome de domínio, tente utilizar o ping(8) em um endereço IP. Se houver 100 por cento (100%) de perda de pacotes, é provável que uma rota padrão não tenha sido atribuída. Verifique novamente se `add default HISADDR` foi definido durante a conexão. Se uma conexão puder ser feita para um endereço IP remoto, é possível que um endereço de resolvedor não tenha sido adicionado ao /etc/resolv.conf. Este arquivo deve se parecer com:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

Onde *x.x.x.x* e *y.y.y.y* deve ser substituído pelo endereço IP dos servidores DNS do ISP.

Para configurar syslog(3) para fornecer o registro para a conexão PPP, verifique se essa linha existe no /etc/syslog.conf:

```
!ppp
*.*     /var/log/ppp.log
```

27.4. Usando o PPP sobre Ethernet (PPPoE)

Esta seção descreve como configurar o PPP sobre Ethernet (PPPoE).

Aqui está um exemplo de ppp.conf funcional:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your Ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
  set login
  add default HISADDR
```

Como `root`, execute:

```
# ppp -ddial name_of_service_provider
```

Adicione o seguinte ao /etc/rc.conf:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"   # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

27.4.1. Usando um nome de perfil PPPoE

Às vezes, será necessário usar nome de perfil para estabelecer a conexão. Nomes de perfil são

usados para distinguir entre diferentes servidores PPPoE conectados a uma determinada rede.

Qualquer informação do nome do perfil necessário deve estar na documentação fornecida pelo ISP.

Como último recurso, pode-se tentar instalar o pacote ou port net/rr-pppoe. Lembre-se, no entanto, que isso pode desprogramar o seu modem e torná-lo inútil, então pense duas vezes antes de fazê-lo. Basta instalar o programa enviado com o modem. Em seguida, acesse o menu **System** do programa. O nome do perfil deve estar listado lá. Geralmente é *ISP*.

O nome do perfil (service tag) será usado na entrada de configuração PPPoE em ppp.conf como a parte do provedor para o `set device`. Consulte ppp(8) para detalhes completos. Deve ficar assim:

```
set device PPPoE:xl1:ISP
```

Não se esqueça de alterar o *xl1* para o dispositivo adequado para a placa Ethernet.

Não se esqueça de alterar o *ISP* para o nome de perfil.

Para informações adicionais, consulte Banda larga mais barata com o FreeBSD em DSL por Renaud Waldura.

27.4.2. PPPoE com um 3Com™HomeConnect™ ADSL Modem Dual Link

Este modem não segue a especificação PPPoE definida em RFC 2516.

Para tornar o FreeBSD capaz de se comunicar com este dispositivo, um sysctl deve ser configurado. Isso pode ser feito automaticamente no momento da inicialização, atualizando o /etc/sysctl.conf:

```
net.graph.nonstandard_pppoe=1
```

ou pode ser feito imediatamente com o comando:

```
# sysctl net.graph.nonstandard_pppoe=1
```

Infelizmente, como essa é uma configuração valida para todo o sistema, não é possível falar com um cliente ou servidor PPPoE normal e um 3Com™HomeConnect™ Modem ADSL ao mesmo tempo.

27.5. Usando PPP sobre ATM (PPPoA)

Esta sessão descreve como configurar o PPP sobre ATM (PPPoA). O PPPoA é uma escolha popular entre os provedores europeus de DSL.

27.5.1. Usando o mpd

O aplicativo mpd pode ser usado para conectar-se a uma variedade de serviços, em particular serviços PPTP. Ele pode ser instalado usando o pacote ou port net/mpd5. Muitos modems ADSL

exigem que um túnel PPTP seja criado entre o modem e o computador.

Uma vez instalado, configure o mpd para adequar-se às configurações do provedor. O port coloca um conjunto de arquivos de configuração de exemplos os quais são bem documentados em /usr/local/etc/mpd/. Um guia completo para configurar o mpd está disponível no formato HTML em /usr/ports/shared/doc/mpd/. Aqui está uma configuração de exemplo para conectar-se a um serviço ADSL com o mpd. A configuração está espalhada em dois arquivos, primeiro o mpd.conf:



Este exemplo de mpd.conf só funciona com o mpd 4.x.

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ①
    set bundle password password ②
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

① O nome de usuário usado para autenticar com seu ISP.

② A senha usada para autenticar com seu ISP.

Informações sobre o link, ou links, a estabelecer são encontradas em mpd.links. Um exemplo do mpd.links para acompanhar o exemplo acima é dado abaixo:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ①
    set pptp peer 10.0.0.138 ②
```

① O endereço IP do computador FreeBSD executando o mpd.

② O endereço IP do modem ADSL. O padrão do Alcatel SpeedTouch™ padrão é `10.0.0.138`.

É possível inicializar a conexão facilmente, emitindo o seguinte comando como `root`:

```
# mpd -b adsl
```

Para ver o status da conexão:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Usar o mpd é a maneira recomendada de se conectar a um serviço ADSL com o FreeBSD.

27.5.2. Usando o pptpclient

Também é possível usar o FreeBSD para conectar-se a outros serviços PPPoA usando o net/pptpclient.

Para usar o net/pptpclient para conectar-se a um serviço DSL, instale o port ou o pacote e edite o /etc/ppp/ppp.conf. Uma seção de exemplo do ppp.conf é dada abaixo. Para maiores informações sobre as opções do ppp.conf consulte ppp(8).

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname username ①
 set authkey password ②
 set ifaddr 0 0
 add default HISADDR
```

① O nome de usuário no provedor de DSL.

② A senha da sua conta.



Como a senha da conta é adicionada ao ppp.conf em forma de texto simples, certifique-se de que ninguém possa ler o conteúdo deste arquivo:

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Isso abrirá um túnel para uma sessão PPP para o roteador DSL. Os modems Ethernet DSL têm um endereço IP LAN pré-configurado para conexão. No caso do Alcatel SpeedTouch™ Home, este endereço é `10.0.0.138`. A documentação do roteador deve listar o endereço que o dispositivo usa. Para abrir o túnel e iniciar uma sessão PPP:

```
# pptp address adsl
```



Se um E comercial ("&") for adicionado ao final desse comando, o pptp retornará ao prompt.

Um dispositivo de túnel virtual tun será criado para interação entre os processos do pptp e do ppp. Quando o prompt for retornado ou o processo do pptp confirmar uma conexão, examine o túnel:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
    Opened by PID 918
```

Se a conexão falhar, verifique a configuração do roteador, que geralmente é acessível usando um navegador da web. Além disso, examine a saída do `pptp` e o conteúdo do arquivo de log /var/log/ppp.log para pistas.

Capítulo 28. Correio Eletrônico

28.1. Sinopse

O "Electronic Mail", mais conhecido como email, é uma das formas de comunicação mais utilizadas atualmente. Este capítulo fornece uma introdução básica à execução de um servidor de email no FreeBSD, bem como uma introdução ao envio e recebimento de email usando o FreeBSD. Para uma cobertura mais completa deste assunto, consulte os livros listados em Bibliografia.

Depois de ler este capítulo, você saberá:

- Quais softwares estão envolvidos no envio e recebimento de mensagens de email.
- Onde os arquivos básicos de configuração do Sendmail estão localizados no FreeBSD.
- A diferença entre caixas de correio remotas e locais.
- Como bloquear spammers de utilizar ilegalmente um servidor de email como relay.
- Como instalar e configurar um Mail Transfer Agent, substituindo o Sendmail.
- Como solucionar problemas comuns de servidor de email.
- Como configurar o sistema para apenas enviar email.
- Como usar email com uma conexão discada.
- Como configurar a autenticação SMTP para segurança adicional.
- Como instalar e usar um Mail User Agent, como o mutt, para enviar e receber email.
- Como baixar emails de um servidor remoto utilizando POP ou IMAP.
- Como aplicar automaticamente filtros e regras ao email recebido.

Antes de ler este capítulo, você deve:

- Configurar corretamente uma conexão de rede (Rede Avançada).
- Configure corretamente as informações de DNS para um host de email (Servidores de Rede).
- Saber como instalar software adicional de terceiros (Instalando Aplicativos. Pacotes e Ports).

28.2. Componentes de Email

Há cinco partes principais envolvidas em uma troca de email: o Mail User Agent (MUA), o Mail Transfer Agent (MTA), um host de email, uma caixa de correio remota ou local e DNS. Esta seção fornece uma visão geral desses componentes.

Mail User Agent (MUA)

O Mail User Agent (MUA) é um aplicativo que é usado para redigir, enviar e receber emails. Este aplicativo pode ser um programa de linha de comando, como o utilitário `mail` interno ou um aplicativo de terceiros da Coleção de Ports, como mutt, alpine ou elm. Dezenas de programas gráficos também estão disponíveis na Coleção de Ports, incluindo o Claws Mail, Evolution e Thunderbird. Algumas organizações fornecem um programa web de email que pode ser

acessado por meio de um navegador. Mais informações sobre como instalar e usar um MUA no FreeBSD podem ser encontradas em Mail User Agents.

Mail Transfer Agent (MTA)

O Mail Transfer Agent (MTA) é responsável por receber emails de entrada e entregar emails de saída. O FreeBSD vem com o Sendmail como o MTA padrão, mas também suporta vários outros daemons de servidor de email, incluindo Exim, Postfix e qmail. A configuração do Sendmail é descrita em Arquivos de Configuração do Sendmail. Se outro MTA estiver instalado usando a Coleção de Ports, consulte sua mensagem de pós-instalação para detalhes de configuração específicos do FreeBSD e o site do aplicativo para obter instruções de configuração mais completas.

Servidor de Email e Caixas de Correio

O servidor de email é um servidor responsável por entregar e receber emails para um host ou uma rede. O servidor de email coleta todas as mensagens enviadas para o domínio e as armazena no mbox padrão ou no formato alternativo Maildir, dependendo da configuração. Uma vez que o email foi armazenado, ele pode ser lido localmente usando um MUA ou acessado e coletado remotamente usando protocolos como POP ou IMAP. Se o email for lido localmente, não é necessário instalar um servidor POP ou IMAP.

Para acessar as caixas de email remotamente, é necessário um servidor POP ou IMAP, pois esses protocolos permitem que os usuários se conectem a suas caixas de correio de locais remotos. O IMAP oferece várias vantagens sobre o POP. Isso inclui a capacidade de armazenar uma cópia de mensagens em um servidor remoto após o download e atualizações simultâneas. O IMAP pode ser útil em links de baixa velocidade, pois permite aos usuários buscar a estrutura das mensagens sem baixá-las. Ele também pode executar tarefas como pesquisas no servidor para minimizar a transferência de dados entre clientes e servidores.

Vários servidores POP e IMAP estão disponíveis na Coleção de Ports. Estes incluem o mail/qpopper, mail/imap-uw, mail/courier-imap e mail/dovecot2 .



Deve-se notar que o POP e o IMAP transmitem informações, incluindo nome de usuário e senha, em texto não criptografado. Para garantir a segurança na transmissão de informações entre esses protocolos, considere a utilização de túneis seguros com ssh(1) (Tunelamento SSH) ou utilize SSL (OpenSSL).

Sistema de Nomes de Domínio (DNS)

O Sistema de Nomes de Domínio (DNS) e seu daemon `named` desempenham um grande papel na entrega de email. Para enviar emails de um site para outro, o MTA procurará o site remoto por DNS para determinar qual host receberá os emails para o destino. Esse processo também ocorre quando o email é enviado de um host remoto para o MTA.

Além de mapear nomes de hosts para endereços de IP, o DNS é responsável por armazenar informações específicas da entrega de emails, conhecidas como Mail eXchanger MX. O registro MX especifica quais hosts receberão mensagens de um domínio em particular.

Para visualizar os registros MX de um domínio, especifique o tipo de registro. Consulte host(1), para mais detalhes sobre este comando:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled by 10 mx1.FreeBSD.org
```

Consulte Sistema de Nomes de Domínio (DNS) para mais informações sobre DNS e sua configuração.

28.3. Arquivos de Configuração do Sendmail

Sendmail é o MTA padrão instalado com o FreeBSD. Ele aceita emails de MUAs e os entrega ao host de email apropriado, conforme definido por sua configuração. O Sendmail também pode aceitar conexões de rede e enviar mensagens para caixas de correio locais ou para outro programa.

Os arquivos de configuração do Sendmail estão localizados em /etc/mail. Esta seção descreve esses arquivos em mais detalhes.

/etc/mail/access

Este arquivo de acesso define quais hosts ou endereços de IP têm acesso ao servidor de email local e que tipo de acesso eles possuem. Os hosts listados como `OK`, que é a opção padrão, têm permissão para enviar emails para esse host, desde que o destino final do email seja a máquina local. Os hosts listados como `REJECT` são rejeitados para todas as conexões de email. Os hosts listados como `RELAY` têm permissão para enviar emails para qualquer destino usando este servidor de email. Os hosts listados como `ERROR` terão seus emails retornados com o erro de email especificado. Se um host estiver listado como `SKIP`, o Sendmail interromperá a pesquisa atual por esta entrada sem aceitar ou rejeitar o email. Os hosts listados como `QUARANTINE` terão suas mensagens retidas e receberão o texto especificado como o motivo da retenção.

Exemplos de uso destas opções para endereços IPv4 e IPv6 podem ser encontrados na configuração de exemplo do FreeBSD, /etc/mail/access.sample:

```
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
dbaio $
#
# Mail relay access control list.  Default is to reject mail unless the
# destination is local, or listed in /etc/mail/local-host-names
#
## Examples (commented out for safety)
#From:cyberspammer.com          ERROR:"550 We don't accept mail from spammers"
#From:okay.cyberspammer.com     OK
#Connect:sendmail.org           RELAY
#To:sendmail.org                RELAY
#Connect:128.32                 RELAY
#Connect:128.32.2               SKIP
#Connect:IPv6:1:2:3:4:5:6:7     RELAY
#Connect:suspicious.example.com QUARANTINE:Mail from suspicious host
#Connect:[127.0.0.3]            OK
#Connect:[IPv6:1:2:3:4:5:6:7:8] OK
```

Para configurar o arquivo de acesso, use o formato mostrado no exemplo para adicionar entradas em /etc/mail/access, mas não coloque um símbolo de comentário (#) na frente das entradas. Crie uma entrada para cada host ou rede cujo acesso deve ser configurado. Os remetentes de email que correspondem ao lado esquerdo da tabela são afetados pela ação no lado direito da tabela.

Sempre que este arquivo for atualizado, atualize seu banco de dados e reinicie o Sendmail:

```
# makemap hash /etc/mail/access < /etc/mail/access
# service sendmail restart
```

/etc/mail/aliases

Este arquivo aliases contém uma lista de caixas de correio virtuais que são expandidas para usuários, arquivos, programas ou outros aliases. Aqui estão algumas entradas para ilustrar o formato do arquivo:

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

O nome da caixa de correio no lado esquerdo dos dois pontos é expandido para o(s) alvo(s) à direita. A primeira entrada expande a caixa de correio `root` para a caixa de correio `localuser`, que é então pesquisada no /etc/mail/aliases. Se nenhuma correspondência for encontrada, a mensagem será entregue para `localuser`. A segunda entrada mostra uma lista de email. Um email para `ftp-bugs` é expandido para as três caixas de correio locais `joe`, `eric` e `paul`. Uma caixa de correio remota pode ser especificada como *user@example.com*. A terceira entrada mostra como escrever mensagens em um arquivo, neste caso, /dev/null. A última entrada demonstra como enviar email para um programa, /usr/local/bin/procmail, através de um pipe UNIX™. Consulte aliases(5) para obter mais informações sobre o formato desse arquivo.

Sempre que este arquivo for atualizado, execute `newaliases` para atualizar e inicializar o banco de dados de aliases.

/etc/mail/sendmail.cf

Este é o arquivo de configuração principal do Sendmail. Ele controla o comportamento geral do Sendmail, incluindo tudo desde a tradução de endereços de email até a impressão de mensagens de rejeição para servidores de email remotos. Assim, este arquivo de configuração é bastante complexo. Felizmente, esse arquivo raramente precisa ser alterado para servidores de email padrão.

O arquivo de configuração master do Sendmail pode ser criado a partir de macros m4(1) que definem os recursos e o comportamento do Sendmail. Consulte /usr/src/contrib/sendmail/cf/README para mais detalhes.

Sempre que alterações nesse arquivo são feitas, o Sendmail precisa ser reiniciado para que as alterações entrem em vigor.

/etc/mail/virtusertable

Esse arquivo mapeia endereços de email de domínios virtuais para caixas de correio usuários reais. Essas caixas de correio podem ser locais, remotas, aliases definidas em /etc/mail/aliases ou arquivos. Isso permite que vários domínios virtuais sejam hospedados em uma máquina.

O FreeBSD fornece um exemplo de arquivo de configuração em /etc/mail/virtusertable.sample para demonstrar ainda mais seu formato. O exemplo a seguir demonstra como criar entradas personalizadas usando esse formato:

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

Este arquivo é processado pela primeira entrada que for correspondida. Quando um endereço de email corresponde ao endereço à esquerda, ele é mapeado para a caixa de correio local listada à direita. O formato da primeira entrada neste exemplo mapeia um endereço de email específico para uma caixa de correio local, enquanto o formato da segunda entrada mapeia um endereço de email específico para uma caixa de correio remota. Por fim, qualquer endereço de email de `example.com` que não correspondeu a nenhuma das entradas anteriores corresponderá ao último mapeamento e será enviado para a caixa de correio local `joe`. Ao criar entradas personalizadas, use este formato e adicione-as ao /etc/mail/virtusertable. Sempre que este arquivo for editado, atualize seu banco de dados e reinicie o Sendmail:

```
# makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
# service sendmail restart
```

/etc/mail/relay-domains

Em uma instalação padrão do FreeBSD, o Sendmail é configurado para enviar apenas mensagens provenientes do host em que está sendo executado. Por exemplo, se um servidor POP estiver disponível, os usuários poderão verificar os emails de locais remotos, mas não poderão enviar emails de domínios externos. Normalmente, após alguns momentos da tentativa, um email será enviado de `MAILER-DAEMON` com uma mensagem `5.7 Relaying Denied`.

A solução mais simples é adicionar o FQDN do ISP ao /etc/mail/relay-domains. Se vários endereços forem necessários, adicione-os um por linha:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Depois de criar ou editar este arquivo, reinicie o Sendmail com `service sendmail restart`.

Agora, qualquer mensagem enviada pelo sistema por qualquer domínio dessa lista, desde que o usuário tenha uma conta no sistema, será aceita. Isso permite que os usuários enviem emails de domínios remotos do sistema sem precisar liberar acesso externo ao sistema, evitando SPAM da

Internet.

28.4. Alterando o Mail Transfer Agent

O FreeBSD vem com o Sendmail já instalado como MTA, que é responsável pelos emails enviados e recebidos. No entanto, o administrador do sistema pode alterar o MTA do sistema. Uma ampla lista de alternativas de MTAs está disponível na categoria `mail` da Coleção de Ports do FreeBSD.

Uma vez que um novo MTA esteja instalado, configure e teste o novo software antes de substituir o Sendmail. Consulte a documentação do novo MTA para obter informações sobre como configurar o software.

Uma vez que o novo MTA estiver funcionando, use as instruções nesta seção para desativar o Sendmail e configurar o FreeBSD para usar o MTA substituto.

28.4.1. Desativar o Sendmail



Se o serviço de email de saída do Sendmail estiver desabilitado, é importante que ele seja substituído por um sistema de entrega de email alternativo. Caso contrário, as funções do sistema, como periodic(8), não poderão entregar seus resultados por email. Muitas partes do sistema esperam um MTA funcional. Se os aplicativos continuarem a usar os binários do Sendmail para tentar enviar emails depois que eles forem desativados, o email poderá entrar em uma fila inativa do Sendmail e nunca será entregue.

Para desabilitar completamente o Sendmail, adicione ou edite as seguintes linhas no /etc/rc.conf:

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

Para desabilitar somente o serviço de email de entrada do Sendmail, use apenas esta entrada no /etc/rc.conf:

```
sendmail_enable="NO"
```

Mais informações sobre as opções de inicialização do Sendmail estão disponíveis em rc.sendmail(8).

28.4.2. Substitua o MTA Padrão

Quando um novo MTA é instalado usando a Coleção de Ports, seu script de inicialização também é instalado e as instruções de inicialização são mencionadas em sua mensagem de pacote. Antes de iniciar o novo MTA, pare os processos do Sendmail em execução. Este exemplo interrompe todos esses serviços e em seguida, inicia o serviço Postfix:

```
# service sendmail stop
# service postfix start
```

Para configurar a substituição MTA na inicialização do sistema, adicione sua linha de configuração ao /etc/rc.conf. Esta entrada habilita o MTA Postfix:

```
postfix_enable="YES"
```

Algumas configurações adicionais são necessárias, pois o Sendmail é tão onipresente que alguns softwares assumem que ele já está instalado e configurado. Verifique o /etc/periodic.conf e certifique-se de que esses valores estejam configurados como NO. Se este arquivo não existir, crie-o com estas entradas:

```
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"
```

Alguns MTAs alternativos fornecem suas próprias implementações compatíveis de linha de comando do Sendmail para facilitar o uso delas como substitutos para o Sendmail. No entanto, alguns MUAs podem tentar executar binários padrão do Sendmail em vez dos binários do novo MTA. O FreeBSD usa o /etc/mail/mailer.conf para mapear os binários esperados do Sendmail para o local dos novos binários. Mais informações sobre esse mapeamento podem ser encontradas em mailwrapper(8).

O /etc/mail/mailer.conf padrão se parece com isto:

```
# $FreeBSD: head/pt_BR.ISO8859-1/books/handbook/book.xml 53984 2020-03-15 16:03:31Z
dbaio $
#
# Execute the "real" sendmail program, named /usr/libexec/sendmail/sendmail
#
sendmail        /usr/libexec/sendmail/sendmail
send-mail       /usr/libexec/sendmail/sendmail
mailq           /usr/libexec/sendmail/sendmail
newaliases      /usr/libexec/sendmail/sendmail
hoststat        /usr/libexec/sendmail/sendmail
purgestat       /usr/libexec/sendmail/sendmail
```

Quando qualquer um dos comandos listados à esquerda é executado, o sistema na verdade executa o comando associado mostrado à direita. Esse sistema facilita a alteração de quais binários são executados quando esses binários padrões são chamados.

Alguns MTAs, quando instalados usando a Coleção de Ports, solicitarão a atualização deste arquivo para os novos binários. Por exemplo, o Postfix atualizará o arquivo da seguinte forma:

```
#
# Execute the Postfix sendmail program, named /usr/local/sbin/sendmail
#
sendmail        /usr/local/sbin/sendmail
send-mail       /usr/local/sbin/sendmail
mailq           /usr/local/sbin/sendmail
newaliases      /usr/local/sbin/sendmail
```

Se a instalação do MTA não atualizar automaticamente o /etc/mail/mailer.conf, edite esse arquivo em um editor de texto para que ele aponte para os novos binários. Este exemplo aponta para os binários instalados pelo mail/ssmtp:

```
sendmail        /usr/local/sbin/ssmtp
send-mail       /usr/local/sbin/ssmtp
mailq           /usr/local/sbin/ssmtp
newaliases      /usr/local/sbin/ssmtp
hoststat        /usr/bin/true
purgestat       /usr/bin/true
```

Depois que tudo estiver configurado, é recomendável reinicializar o sistema. A reinicialização oferece a oportunidade de garantir que o sistema esteja configurado corretamente para iniciar o novo MTA automaticamente no boot.

28.5. Solução de problemas

28.5.1. Por que preciso usar o FQDN para hosts no meu site?

O host pode, na verdade, estar em um domínio diferente. Por exemplo, para um host em `foo.bar.edu` se conectar a um host chamado `mumble` no domínio `bar.edu`, faça a referência pelo Nome de Domínio Totalmente Qualificado (Fully-Qualified Domain Name) FQDN, `mumble.bar.edu`, em vez de apenas `mumble`.

Isso ocorre porque a versão do BIND que vem com o FreeBSD não fornece mais abreviações padrão para não-FQDNs que não sejam o domínio local. Um host não qualificado como `mumble` deve ser encontrado como `mumble.foo.bar.edu`, ou ele será procurado no domínio raiz.

Nas versões mais antigas do BIND, a pesquisa continuava em `mumble.bar.edu` e `mumble.edu`. A RFC 1535 detalha por que isso é considerado uma má prática ou até mesmo uma falha de segurança.

Como uma boa solução, coloque a linha:

```
search foo.bar.edu bar.edu
```

em vez do anterior:

```
domain foo.bar.edu
```

no /etc/resolv.conf. No entanto, certifique-se de que a ordem de pesquisa não ultrapasse o limite "entre administração local e pública", como a RFC 1535 a chama.

28.5.2. Como posso executar um servidor de email em um host PPP dial-up?

Conecte-se a um gateway de email FreeBSD na LAN. A conexão PPP não é dedicada.

Uma maneira de fazer isso é obter um servidor de Internet em tempo integral para fornecer serviços MX secundários para o domínio. Neste exemplo, o domínio é `example.com` e o ISP configurou `example.net` para fornecer o serviço de MX secundário para o domínio:

```
example.com.          MX        10      example.com.
                      MX        20      example.net.
```

Apenas um host deve ser especificado como o destinatário final. Para Sendmail, adicione `Cw example.com` em /etc/mail/sendmail.cf em `example.com`.

Quando o MTA de envio tentar entregar o email, ele tentará conectar ao sistema, `example.com`, através do link PPP. Isso expirará se o destino estiver offline. O MTA irá entregá-lo automaticamente ao site MX secundário no Provedor de Serviços de Internet (ISP), `example.net`. O site secundário de MX tentará conectar-se periodicamente ao host primário MX, `example.com`.

Use algo assim como um script de login:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Ao criar um script de login separado para usuários, use `sendmail -qRexample.com` no script acima. Isso forçará todos os emails na fila para que `example.com` sejam processados imediatamente.

Um refinamento adicional da situação pode ser visto neste exemplo na lista de discussão Lista de discussão de provedor de serviços de Internet do FreeBSD:

```
> we provide the secondary MX for a customer. The customer connects to
> our services several times a day automatically to get the mails to
> his primary MX (We do not call his site when a mail for his domains
> arrived). Our sendmail sends the mailqueue every 30 minutes. At the
> moment he has to stay 30 minutes online to be sure that all mail is
> gone to the primary MX.
>
> Is there a command that would initiate sendmail to send all the mails
> now? The user has not root-privileges on our machine of course.
```

```
In the privacy flags section of sendmail.cf, there is a
definition Opgoaway,restrictqrun

Remove restrictqrun to allow non-root users to start the queue processing.
You might also like to rearrange the MXs. We are the 1st MX for our
customers like this, and we have defined:

# If we are the best MX for a host, try directly instead of generating
# local config error.
OwTrue

That way a remote site will deliver straight to you, without trying
the customer connection.  You then send to your customer.  Only works for
hosts, so you need to get your customer to name their mail
machine customer.com as well as
hostname.customer.com in the DNS.  Just put an A record in
the DNS for customer.com.
```

28.6. Tópicos Avançados

Esta seção aborda tópicos mais envolvidos, como configuração de email e configuração de email para um domínio inteiro.

28.6.1. Configuração básica

Fora da caixa, pode-se enviar email para hosts externos desde que /etc/resolv.conf esteja configurado ou a rede tenha acesso a um servidor DNS. Para ter um email entregue ao MTA em um host FreeBSD, siga um destes procedimentos:

- Execute um servidor DNS para o domínio.
- Tenha o email entregue diretamente para o FQDN para a máquina.

Para que o email seja entregue diretamente a um host, ele deve ter um endereço IP estático permanente, não um endereço IP dinâmico. Se o sistema estiver protegido por um firewall, ele deverá ser configurado para permitir o tráfego SMTP. Para receber mensagens diretamente em um host, um desses dois deve ser configurado:

- Certifique-se de que o registro MX de menor numeração no DNS aponte para o endereço IP estático do host.
- Certifique-se de que não exista nenhuma entrada MX no DNS para o host.

Qualquer um dos itens acima permitirá que o correio seja recebido diretamente no host.

Tente isto:

```
# hostname
example.FreeBSD.org
```

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

Neste exemplo, as mensagens enviadas diretamente para yourlogin@exemplo.FreeBSD.org devem funcionar sem problemas, supondo que o Sendmail esteja sendo executado corretamente em `example.FreeBSD.org`.

Para este exemplo:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by nevdull.FreeBSD.org
```

Todas as mensagens enviadas para `exemple.FreeBSD.org` serão coletadas no `hub` sob o mesmo nome de usuário, em vez de serem enviadas diretamente para o seu host.

As informações acima são tratadas pelo servidor DNS. O registro DNS que possui as informações de roteamento de email é a entrada MX. Se não existir nenhum registro MX, os emails serão entregues diretamente ao host por meio de seu endereço IP.

A entrada MX de `freefall.FreeBSD.org` uma vez foi assim:

```
freefall        MX  30  mail.crl.net
freefall        MX  40  agora.rdrop.com
freefall        MX  10  freefall.FreeBSD.org
freefall        MX  20  who.cdrom.com
```

`freefall` teve muitas entradas MX. O menor número MX é o host que recebe email diretamente, se disponível. Se não for acessível por algum motivo, o próximo host de número mais baixo aceitará as mensagens temporariamente e as transmitirá quando um host de número inferior for disponibilizado.

Sites alternativos de MX devem ter conexões de Internet separadas para serem mais úteis. Seu ISP pode fornecer este serviço.

28.6.2. Email para um Domínio

Ao configurar um MTA para uma rede, qualquer mensagem enviada para hosts em seu domínio deve ser desviada para o MTA para que os usuários possam receber seus emails no servidor de email principal.

Para tornar a vida mais fácil, uma conta de usuário com o mesmo *username* deve existir tanto no MTA como no sistema com o MUA. Use adduser(8) para criar as contas de usuário.

O MTA deve ser o servidor de mensagens designado para cada estação de trabalho na rede. Isso é feito na configuração DNS com um registro MX:

```
example.FreeBSD.org A   204.216.27.XX       ; Workstation
            MX  10 nevdull.FreeBSD.org  ; Mailhost
```

Isso redirecionará o email para a estação de trabalho para o MTA, não importa onde o registro A aponta. O email é enviado para o host MX.

Isso deve ser configurado em um servidor DNS. Se a rede não executar seu próprio servidor DNS, fale com o ISP ou provedor DNS.

A seguir, um exemplo de hospedagem de email virtual. Considere um cliente com o domínio `customer1.org`, onde todas as mensagens para `customer1.org` devem ser enviadas para `mail.myhost.com`. A entrada DNS deve ficar assim:

```
customer1.org       MX  10  mail.myhost.com
```

Um registro `A`> *não* é necessário em `customer1.org` para que seja enviado emails para esse domínio. No entanto, um `ping` em `customer1.org` não funcionará, a menos que exista um registro `A` para ele.

Diga ao MTA quais domínios e/ou nomes de host que ele deve aceitar emails. Qualquer um dos itens a seguir funcionará para o Sendmail:

- Adicione os hosts ao /etc/mail/local-host-names ao usar `FEATURE (use_cw_file)`.
- Adicione uma linha `Cwyour.host.com` em /etc/sendmail.cf.

28.7. Configurando Apenas Envio

Há muitos casos em que muitas instâncias podem querer enviar email através de um relay. Alguns exemplos são:

- O computador é uma máquina desktop que precisa usar programas como mail(1), usando o relay de email do ISP.
- O computador é um servidor que não manipula emails localmente, mas precisa passar todos os emails para um relay para processamento.

Embora qualquer MTA seja capaz de preencher esse nicho específico, pode ser difícil configurar adequadamente um MTA com todos os recursos apenas para lidar com o descarregamento de email. Programas como Sendmail e Postfix são um exagero para esse uso.

Além disso, um acordo típico de serviço de acesso à Internet pode proibir a execução de um "servidor de email".

A maneira mais fácil de atender a essas necessidades é instalar o port mail/ssmtp:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Uma vez instalado, o mail/ssmtp pode ser configurado em /usr/local/etc/ssmtp/ssmtp.conf:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Use o endereço de email real para `root`. Insira a retransmissão de mensagens de saída do ISP no lugar de `mail.example.com`. Alguns ISPs chamam isso de "servidor de email de saída" ou "servidor SMTP".

Certifique-se de desativar o Sendmail, incluindo o serviço de envio de mensagens. Veja Desativar o Sendmail para detalhes.

mail/ssmtp tem algumas outras opções disponíveis. Consulte os exemplos em /usr/local/etc/ssmtp ou na página de manual do ssmtp para obter mais informações.

A configuração do ssmtp dessa maneira permite que qualquer software no computador que precise enviar mensagens funcione corretamente, sem violar a política de uso dos ISPs ou permitindo que o computador seja sequestrado para envio de spam.

28.8. Usando Email com uma Conexão Dialup

Ao usar um endereço IP estático, não é necessário ajustar a configuração padrão. Configure o nome do host para o nome da Internet designado e o Sendmail fará o resto.

Ao usar um endereço IP atribuído dinamicamente e uma conexão PPP de discagem à Internet, geralmente há uma caixa de correio no servidor de email do ISP. Neste exemplo, o domínio do ISP é `example.net`, o nome de usuário é `user`, o nome do host é `bsd.home`, e o ISP permitiu `relay.example.net` como um relay de email.

Para baixar emails da caixa de correio do ISP, instale um agente pela coleção de ports. O mail/fetchmail é uma boa escolha, pois suporta muitos protocolos diferentes. Normalmente, o ISP fornecerá POP. Ao usar o usuário PPP, o email pode ser baixado automaticamente quando uma conexão com a Internet é estabelecida com a seguinte entrada em /etc/ppp/ppp.linkup:

```
MYADDR:
!bg su user -c fetchmail
```

Ao usar o Sendmail para entregar emails em contas não locais, configure o Sendmail para processar a fila de mensagens assim que a conexão com a Internet for estabelecida. Para fazer isso, adicione esta linha após a entrada `fetchmail` acima em /etc/ppp/ppp.linkup:

```
  !bg su user -c "sendmail -q"
```

Neste exemplo, há uma conta para `user` em `bsd.home`. No diretório home de `user` em `bsd.home`, crie

um .fetchmailrc que contenha esta linha :

```
poll example.net protocol pop3 fetchall pass MySecret
```

Este arquivo não deve ter permissão de leitura para ninguém, exceto pelo `user`, pois contém a senha `MySecret`.

Para enviar emails com o cabeçalho correto `from:`, configure o Sendmail para usar user@example.net em vez de user@bsd.home e para enviar todos os emails através de `relay.example.net`, permitindo uma transmissão de email mais rápida.

O seguinte .mc deve ser suficiente:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME',`bsd.home')dnl
define(`confDELIVERY_MODE',`deferred')dnl
```

Consulte a seção anterior para obter detalhes sobre como converter esse arquivo no formato sendmail.cf. Não esqueça de reiniciar o Sendmail após atualizar o sendmail.cf.

28.9. Autenticação SMTP

Configurar a autenticação SMTP no MTA oferece vários benefícios. A autenticação SMTP adiciona uma camada de segurança ao Sendmail e fornece aos usuários móveis que alternam os hosts a capacidade de usar o mesmo MTA sem a necessidade de reconfigurar as configurações de seus clientes de email a cada vez.

1. Instale o security/cyrus-sasl2 da Coleção de Ports. Este port suporta várias opções de tempo de compilação. Para o método de autenticação SMTP demonstrado neste exemplo, certifique-se de que `LOGIN` não esteja desabilitado.
2. Depois de instalar o security/cyrus-sasl2, edite o /usr/local/lib/sasl2/Sendmail.conf, ou crie-o se ele não existir, e adicione a seguinte linha :

```
pwcheck_method: saslauthd
```

3. Em seguida, instale o security/cyrus-sasl2-saslauthd e adicione a seguinte linha ao /etc/rc.conf:

   ```
   saslauthd_enable="YES"
   ```

   Finalmente, inicie o daemon saslauthd:

   ```
   # service saslauthd start
   ```

   Este daemon serve como um intermediário para o Sendmail autenticar no banco de dados do FreeBSD o passwd(5). Isso evita o trabalho de criar um novo conjunto de nomes de usuário e senhas para cada usuário que precise usar a autenticação SMTP e mantém a senha de login e email igual.

4. Em seguida, edite o /etc/make.conf e adicione as seguintes linhas:

   ```
   SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
   SENDMAIL_LDADD=/usr/local/lib/libsasl2.so
   ```

   Essas linhas fornecem ao Sendmail as opções de configuração apropriadas para vincular ao cyrus-sasl2 em tempo de compilação. Certifique-se de que o cyrus-sasl2 tenha sido instalado antes de recompilar o Sendmail.

5. Recompile o Sendmail executando os seguintes comandos:

   ```
   # cd /usr/src/lib/libsmutil
   # make cleandir && make obj && make
   # cd /usr/src/lib/libsm
   # make cleandir && make obj && make
   # cd /usr/src/usr.sbin/sendmail
   # make cleandir && make obj && make && make install
   ```

   Esta compilação não deve ter nenhum problema se o /usr/src não foi alterado extensivamente e as bibliotecas compartilhadas necessárias estiverem disponíveis.

6. Depois que o Sendmail tenha sido compilado e reinstalado, edite o /etc/mail/freebsd.mc ou o arquivo local .mc. Muitos administradores optam por usar a saída de hostname(1) como o nome de .mc para exclusividade. Adicione estas linhas:

   ```
   dnl set SASL options
   TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
   ```

```
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
```

Essas opções configuram os diferentes métodos disponíveis para que o Sendmail autentique usuários. Para usar um método diferente de pwcheck, consulte a documentação do Sendmail.

7. Finalmente, execute make(1) enquanto estiver em /etc/mail. Isso executará o novo .mc e criará um .cf chamado freebsd.cf ou o nome usado para o arquivo local .mc. Em seguida, execute `make install restart`, que copiará o arquivo para o sendmail.cf, e reinicie corretamente o Sendmail. Para mais informações sobre este processo, consulte /etc/mail/Makefile.

Para testar a configuração, use um MUA para enviar uma mensagem de teste. Para investigações posteriores, defina o `LogLevel` do Sendmail como `13` e verifique o /var/log/maillog para quaisquer erros.

Para mais informações, consulte autenticação SMTP.

28.10. Mail User Agents

Um MUA é um aplicativo usado para enviar e receber emails. À medida que o email "evolui" e se torna mais complexo, os MUAs estão se tornando cada vez mais poderosos e fornecem aos usuários maior funcionalidade e flexibilidade. A categoria `mail` da Coleção de Ports do FreeBSD contém numerosos MUAs. Eles incluem clientes de email gráficos como Evolution ou Balsa e clientes baseados em console, como mutt ou alpine.

28.10.1. `mail`

mail(1) é o MUA padrão instalado com o FreeBSD. É um MUA baseado em console que oferece a funcionalidade básica necessária para enviar e receber email em texto. Ele fornece suporte limitado a anexos e só pode acessar caixas de correio locais.

Embora o `mail` não suporte nativamente a interação com os servidores POP ou IMAP, essas caixas de correio podem ser baixadas para um arquivo mbox local usando um aplicativo como fetchmail.

Para enviar e receber email, execute `mail`:

```
% mail
```

O conteúdo da caixa de correio do usuário em /var/mail é lido automaticamente pelo `mail`. Se a caixa de correio estiver vazia, o utilitário sairá com uma mensagem indicando que nenhum email foi encontrado. Se o email existir, a interface do aplicativo será iniciada e uma lista de mensagens será exibida. As mensagens são numeradas automaticamente, como pode ser visto no exemplo a seguir:

```
Mail version 8.1 6/6/93.  Type ? for help.
```

```
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Agora as mensagens podem ser lidas digitando `t` seguido pelo número da mensagem. Este exemplo lê o primeiro email:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

This is a test message, please reply if you receive it.
```

Como visto neste exemplo, a mensagem será exibida com cabeçalhos completos. Para exibir novamente a lista de mensagens, pressione `h`.

Se o email exigir uma resposta, pressione as teclas `R` ou `r` no `mail`. `R` instrui o `mail` a responder apenas ao remetente do email, enquanto `r` responde a todos os outros destinatários da mensagem. Esses comandos podem ser sufixados com o número do email. Depois de digitar a resposta, o final da mensagem deve ser marcado por um único `.` em sua própria linha. Um exemplo pode ser visto abaixo:

```
& R 1
To: root@localhost
Subject: Re: test

Thank you, I did get your email.
.
EOT
```

Para enviar um novo email, pressione `m`, seguido pelo endereço de email do destinatário. Vários destinatários podem ser especificados separando cada endereço com o delimitador `,`. O assunto da mensagem pode então ser inserido, seguido pelo conteúdo da mensagem. O final da mensagem deve ser especificado colocando um único `.` em sua própria linha.

```
& mail root@localhost
Subject: I mastered mail

Now I can send and receive email using mail ... :)
.
```

```
EOT
```

Enquanto estiver usando o `mail`, pressione `?` para exibir a ajuda a qualquer momento. Consulte mail(1) para obter mais detalhes sobre como usar o `mail`.



O mail(1) não foi projetado para manipular anexos e, portanto, lida mal com eles. Novos MUAs lidam com anexos de uma maneira mais inteligente. Usuários que preferem usar `mail` podem preferir o port converters/mpack.

28.10.2. mutt

O mutt é um poderoso MUA, com muitos recursos, incluindo:

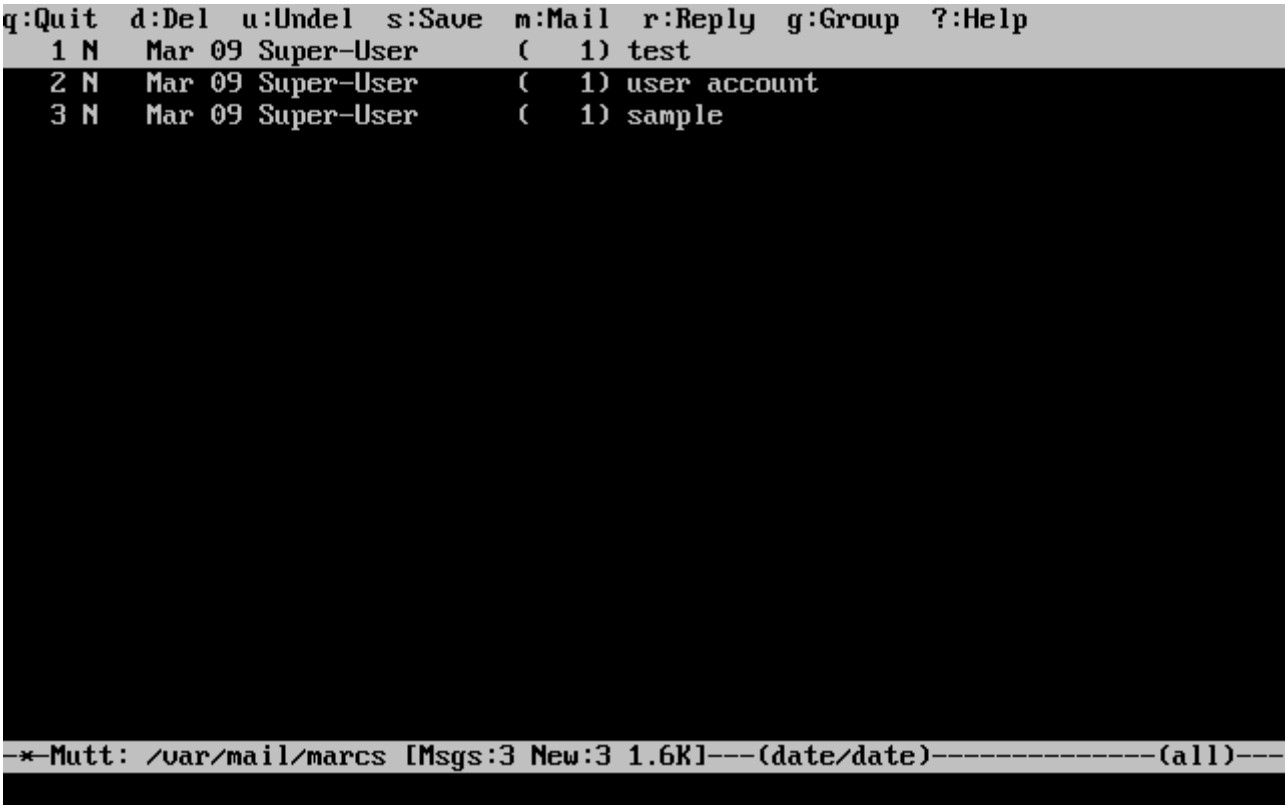
- A capacidade de enviar mensagens.
- Suporte PGP para assinatura digital e criptografia de email.
- Suporte MIME.
- Suporte Maildir.
- Altamente personalizável.

Consulte http://www.mutt.org para mais informações sobre o mutt.

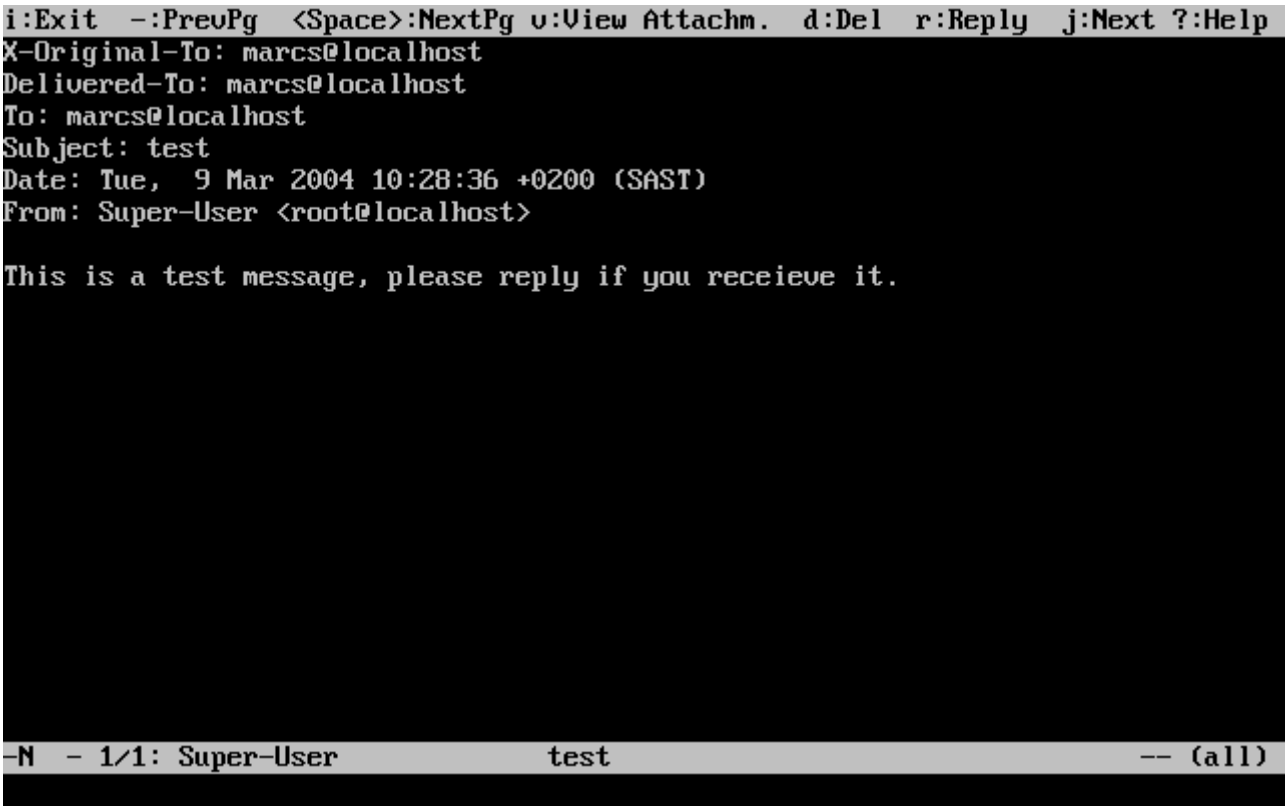
O mutt pode ser instalado usando o port mail/mutt. Após o port ter sido instalado, o mutt pode ser iniciado com o seguinte comando:

```
% mutt
```

O mutt irá automaticamente ler o conteúdo da caixa de correio do usuário em /var/mail. Se nenhum email for encontrado, o mutt aguardará os comandos do usuário. O exemplo abaixo mostra o mutt exibindo uma lista de mensagens:



Para ler um email, selecione-o usando as teclas de cursor e pressione `Enter`. Um exemplo de email exibido pelo mutt pode ser visto abaixo:



Semelhante ao mail(1), o mutt pode ser usado para responder apenas ao remetente da mensagem, bem como para todos os destinatários. Para responder apenas ao remetente do email, pressione `r`. Para enviar uma resposta de grupo ao remetente original e a todos os destinatários da mensagem, pressione `g`.



Por padrão, o mutt usa o editor vi(1) para criar e responder emails. Cada usuário pode personalizar isso criando ou editando o .muttrc em seu diretório home e configurando a variável `editor` ou definindo a variável de ambiente `EDITOR`. Consulte http://www.mutt.org/ para obter mais informações sobre como configurar o mutt.

Para escrever uma nova mensagem de email, pressione `m`. Depois que um assunto válido foi dado, mutt iniciará o vi(1) para que o email possa ser escrito. Quando o conteúdo do email estiver completo, salve e saia do `vi`. O mutt será retomado, exibindo uma tela de resumo do email que será enviado. Para enviar o email, pressione `y`. Um exemplo da tela de resumo pode ser visto abaixo:

```
y:Send  q:Abort  t:To  c:CC  s:Subj  a:Attach file  d:Descrip  ?:Help
    From: Marc Silver <marcs@localhost>
      To: Super-User <root@localhost>
      Cc:
     Bcc:
 Subject: Re: test
Reply-To:
     Fcc:
Security: Clear

-- Attachments
- I     1 /tmp/mutt-bsd-c0hobscQ              [text/plain, 7bit, us-ascii, 1.1K]

-- Mutt: Compose  [Approx. msg size: 1.1K   Atts: 1]----------------------------
```

O mutt contém manuais extensos que podem ser acessados pela maioria dos menus pressionando `?`. A linha superior também exibe os atalhos de teclado, quando apropriado.

28.10.3. alpine

O alpine é destinado a um usuário iniciante, mas também inclui alguns recursos avançados.

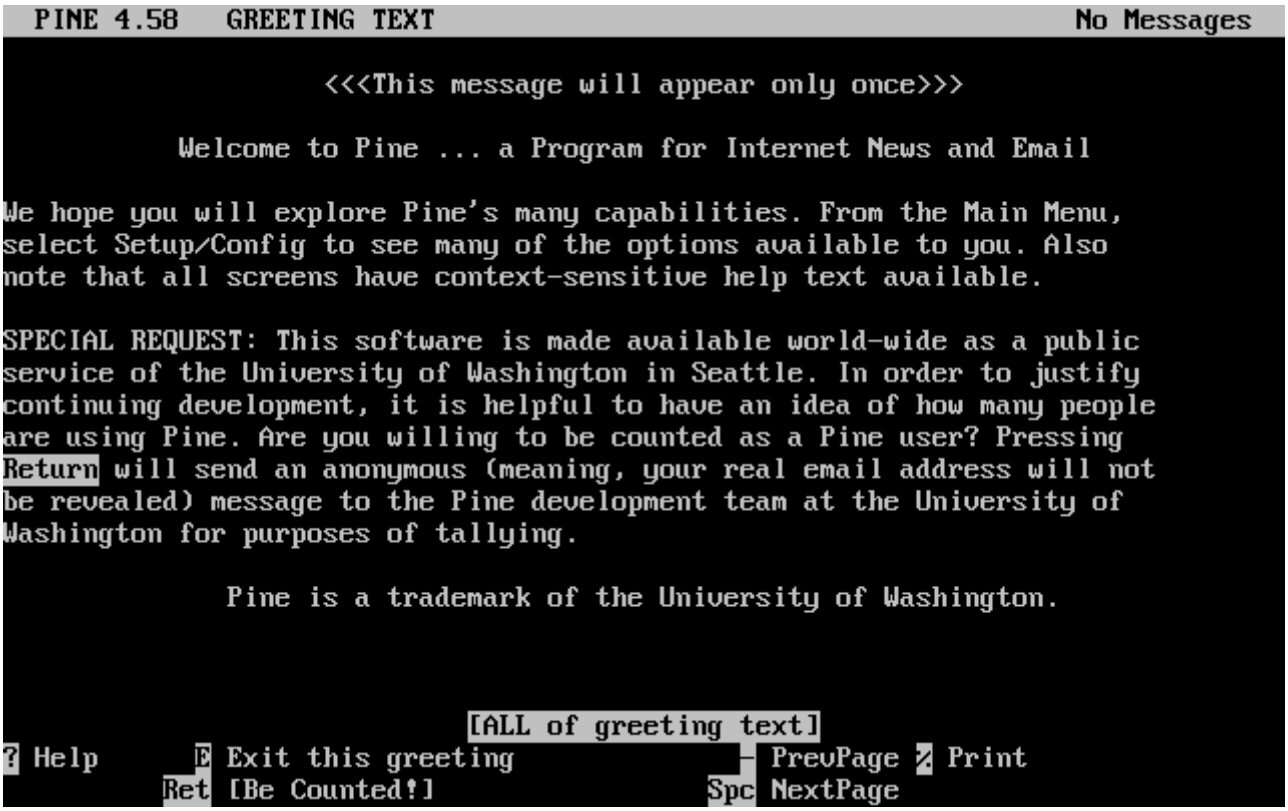


O alpine teve várias vulnerabilidades remotas descobertas no passado, que permitiam que atacantes remotos executassem código arbitrário como usuários no sistema local, pela ação de enviar um email especialmente preparado. Enquanto problemas *conhecidos* foram corrigidos, o código alpine foi escrito em um estilo inseguro e o FreeBSD Security Officer acredita que provavelmente há outras vulnerabilidades não descobertas. Os usuários instalam o alpine por sua conta e risco.

A versão atual do alpine pode ser instalada usando o port mail/alpine. Após a instalação do port, o alpine pode ser iniciado executando o seguinte comando:

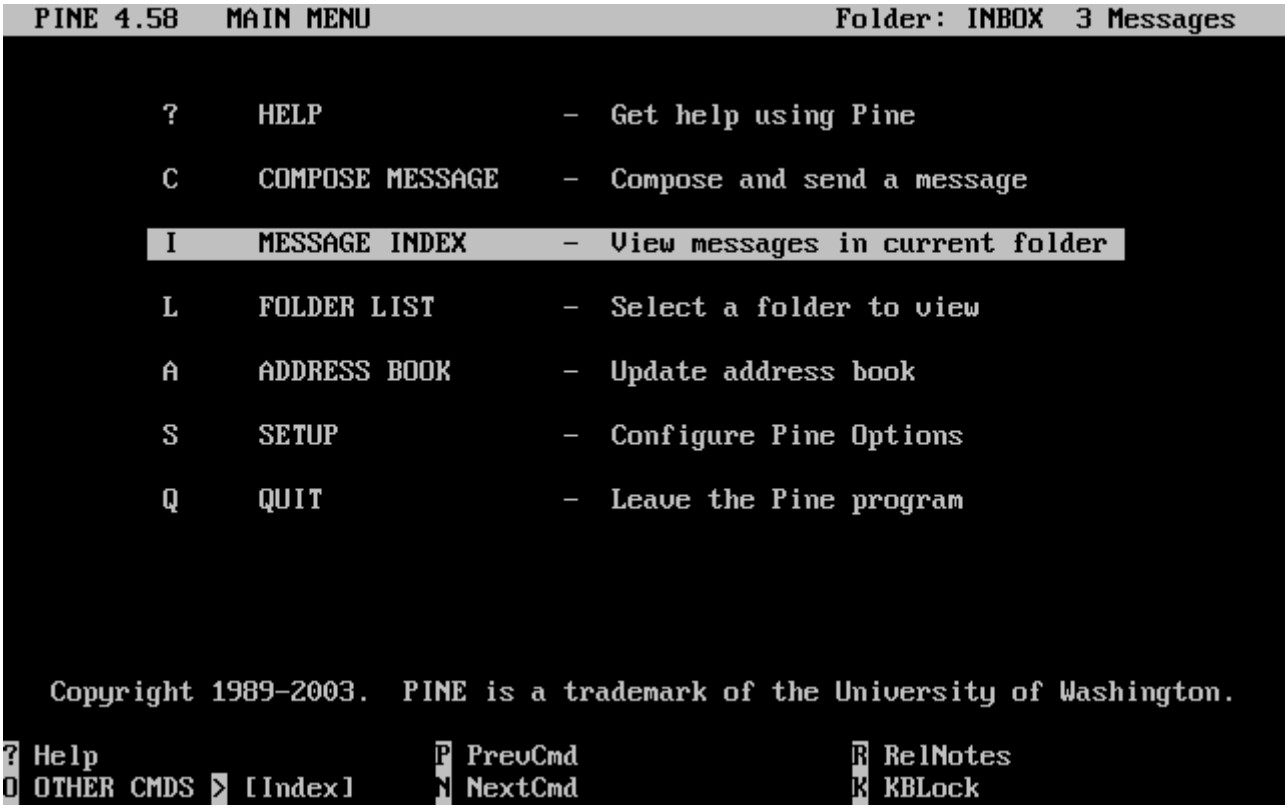
```
% alpine
```

A primeira vez que o alpine é executado, ele exibe uma página de saudação com uma breve introdução, bem como uma solicitação da equipe de desenvolvimento do alpine para enviar uma mensagem de email anônima para que eles saibam quantos usuários estão usando o seu cliente. Para enviar esta mensagem anônima, pressione `Enter`. Como alternativa, pressione `E` para sair da saudação sem enviar uma mensagem anônima. Um exemplo da página de saudação é mostrado abaixo:



O menu principal é então apresentado, o qual pode ser navegado usando as teclas de cursor. Esse menu principal fornece atalhos para a composição de novos emails, navegação em diretórios de email e administração de entradas do catálogo de endereços. Abaixo do menu principal, são mostrados os atalhos de teclado relevantes para executar funções específicas da tarefa em questão.

O diretório padrão aberto pelo alpine é o inbox. Para visualizar o índice da mensagem, pressione `I` ou selecione a opção MESSAGE INDEX mostrada abaixo:



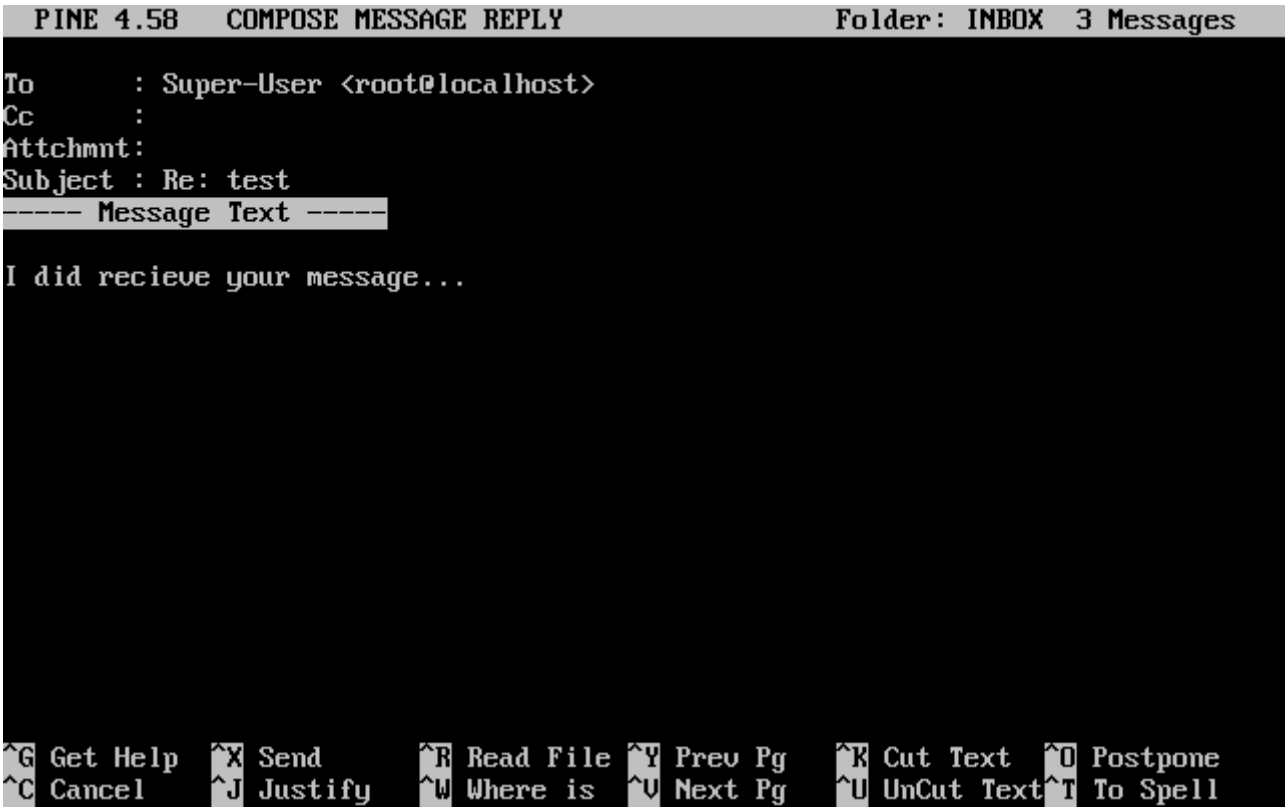
O índice de mensagens mostra mensagens no diretório atual e pode ser navegado usando as teclas de cursor. As mensagens destacadas podem ser lidas pressionando `Enter`.



Na captura de tela abaixo, uma mensagem de exemplo é exibida pelo alpine. Atalhos de teclado contextuais são exibidos na parte inferior da tela. Um exemplo de um atalho é `r`, que diz ao MUA para responder à mensagem atual sendo exibida.



A resposta de um email pelo alpine é feita usando o editor pico, que é instalado por padrão com o alpine. O pico facilita a navegação na mensagem e é mais fácil de ser usado por usuários iniciantes do que o vi(1) ou mail(1). Quando a resposta estiver completa, a mensagem pode ser enviada pressionando `Ctrl` + `X`. O alpine solicitará confirmação antes de enviar a mensagem.



O alpine pode ser personalizado usando a opção SETUP no menu principal. Consulte http://www.washington.edu/alpine/ para mais informações.

28.11. Usando o fetchmail

O fetchmail é um cliente IMAP e POP completo. Ele permite que os usuários baixem automaticamente emails de servidores IMAP e POP remotos e os salvem em caixas de correio locais, onde podem ser acessados mais facilmente. O fetchmail pode ser instalado usando o port mail/fetchmail e oferece vários recursos, incluindo:

- Suporte para os protocolos POP3, o APOP, o KPOP, o IMAP, o ETRN e o ODMR.
- Capacidade de encaminhar emails usando SMTP, que permite que a filtragem, o encaminhamento e aliases funcionem normalmente.
- Pode ser executado no modo daemon para verificar periodicamente novas mensagens.
- Pode buscar várias caixas de correio e encaminhá-las, com base na configuração, para diferentes usuários locais.

Esta seção explica alguns dos recursos básicos do fetchmail. Este utilitário requer uma configuração .fetchmailrc no diretório pessoal do usuário para que seja executado corretamente. Este arquivo inclui informações do servidor, bem como credenciais de login. Devido à natureza sensível do conteúdo deste arquivo, é aconselhável torná-lo legível apenas pelo usuário, com o seguinte comando:

```
% chmod 600 .fetchmailrc
```

O seguinte .fetchmailrc serve como um exemplo para fazer o download de uma única caixa de correio de usuário usando POP. Ele diz ao fetchmail para se conectar ao `example.com` usando um nome de usuário `joesoap` e uma senha de `XXX`. Este exemplo pressupõe que o usuário `joesoap` exista no sistema local.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

O próximo exemplo conecta-se a vários servidores POP e IMAP e redireciona para diferentes nomes de usuários locais quando aplicável:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

fetchmail pode ser executado no modo daemon executando-o com `-d`, seguido pelo intervalo (em segundos) que o fetchmail deve pesquisar servidores listados em .fetchmailrc. O exemplo a seguir configura o fetchmail para pesquisar a cada 600 segundos:

```
% fetchmail -d 600
```

Mais informações sobre o fetchmail podem ser encontradas em http://www.fetchmail.info/.

28.12. Usando o procmail

O procmail é um poderoso aplicativo usado para filtrar mensagens recebidas. Ele permite que os usuários definam "regras" que podem ser correspondidas aos emails recebidos para executar funções específicas ou para redirecionar o email para caixas de correio alternativas ou endereços de email. O procmail pode ser instalado usando o port mail/procmail. Uma vez instalado, ele pode ser diretamente integrado na maioria dos MTAs. Consulte a documentação do MTA para mais informações. Alternativamente, procmail pode ser integrado adicionando a seguinte linha a um .forward no diretório pessoal do usuário:

```
"|exec /usr/local/bin/procmail || exit 75"
```

A seção a seguir exibe algumas regras básicas do procmail, além de breves descrições do que elas fazem. As regras devem ser inseridas em um .procmailrc, que deve residir no diretório pessoal do usuário.

A maioria dessas regras pode ser encontrada em procmailex(5).

Para encaminhar todos os emails de user@example.com para um endereço externo de goodmail@example2.com:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Para encaminhar todos os emails com menos de 1000 bytes para um endereço externo de goodmail@example2.com:

```
:0
* < 1000
! goodmail@example2.com
```

Para enviar todas as mensagens enviadas para alternate@example.com para uma caixa de correio chamada alternate:

```
:0
* ^TOalternate@example.com
alternate
```

Para enviar todas as mensagens com um assunto de "Spam" para /dev/null:

```
:0
```

```
^Subject:.*Spam
/dev/null
```

Uma receita útil que analisa listas de discussão `do FreeBSD.org` e coloca cada lista em sua própria caixa de correio:

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
    LISTNAME=${MATCH}
    :0
    * LISTNAME??^\/[^@]+
    FreeBSD-${MATCH}
}
```

Capítulo 29. Servidores de Rede

29.1. Sinopse

Este capítulo aborda alguns dos serviços de rede usados com mais frequência em sistemas UNIX™. Isso inclui instalar, configurar, testar e manter muitos tipos diferentes de serviços de rede. Exemplos de arquivos de configuração estão incluídos neste capítulo para referência.

No final deste capítulo, os leitores saberão:

- Como gerenciar o daemon inetd.
- Como configurar o Network File System (NFS).
- Como configurar o Network Information Server (NIS) para centralizar e compartilhar contas de usuários.
- Como configurar o FreeBSD para funcionar como um servidor ou cliente LDAP
- Como configurar configurações de rede automáticas usando o DHCP.
- Como configurar um Domain Name Server (DNS).
- Como configurar o servidor ApacheHTTP.
- Como Configurar um Servidor de File Transfer Protocol (FTP).
- Como configurar um servidor de arquivo e de impressão para clientes Windows™ usando o Samba.
- Como sincronizar a hora e a data e configurar um servidor de horário usando o Network Time Protocol (NTP).
- Como configurar o iSCSI.

Este capítulo pressupõe um conhecimento básico de:

- scripts /etc/rc.
- Terminologia de rede.
- Instalação de software adicional de terceiros (Instalando Aplicativos. Pacotes e Ports).

29.2. O super-servidor inetd

O daemon inetd(8) é algumas vezes chamado de Super-Servidor porque gerencia conexões para muitos serviços. Em vez de iniciar vários aplicativos, apenas o serviço inetd precisa ser iniciado. Quando uma conexão é recebida para um serviço gerenciado pelo inetd, ele determina para qual programa a conexão está destinada, gera um processo para esse programa e delega ao programa um socket. O uso de inetd para serviços que não são muito usados pode reduzir a carga do sistema, quando comparado à execução de cada daemon individualmente no modo independente.

Primeiramente, inetd é usado para gerar outros daemons, mas vários protocolos triviais são tratados internamente, como chargen, auth, time, echo, discard e daytime.

Esta seção aborda os conceitos básicos da configuração do inetd.

29.2.1. Arquivo de Configuração

A configuração do inetd é feita editando o /etc/inetd.conf. Cada linha deste arquivo de configuração representa um aplicativo que pode ser iniciado pelo inetd. Por padrão, cada linha começa com um comentário (`#`), o que significa que inetd não está atendendo a nenhum aplicativo. Para configurar o inetd para escutar as conexões de um aplicativo, remova o `#` no início da linha desse aplicativo.

Depois de salvar suas edições, configure o inetd para iniciar na inicialização do sistema editando o arquivo /etc/rc.conf:

```
inetd_enable="YES"
```

Para iniciar o inetd agora, para que ele ouça o serviço que você configurou, digite:

```
# service inetd start
```

Uma vez iniciado o inetd, ele precisa ser notificado sempre que uma modificação for feita no arquivo /etc/inetd.conf:

Exemplo 45. Recarregando o Arquivo de Configuração do inetd

```
# service inetd reload
```

Normalmente, a entrada padrão de um aplicativo não precisa ser editada além da remoção do `#`. Em algumas situações, pode ser apropriado editar a entrada padrão.

Como exemplo, esta é a entrada padrão para ftpd(8) sobre o IPv4:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

As sete colunas em uma entrada são as seguintes:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][/login-class]
server-program
server-program-arguments
```

Onde:

service-name

O nome do serviço do daemon para iniciar. Deve corresponder a um serviço listado no arquivo /etc/services. Isso determina qual porta inetd atende para conexões de entrada para esse serviço. Ao usar um serviço personalizado, ele deve primeiro ser adicionado ao arquivo /etc/services.

socket-type

Ou `stream`, `dgram`, `raw`, ou `seqpacket`. Use `stream` para conexões TCP e `dgram` para serviços UDP.

protocol

Use um dos seguintes nomes de protocolo:

Protocol Name	Explicação
tcp ou tcp4	TCP IPv4
udp ou udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	Ambos TCP IPv4 e IPv6
udp46	Ambos UDP IPv4 e IPv6

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]

Neste campo, `wait` ou `nowait` deve ser especificado. `max-child`, `max-connections-per-ip-por-minute` e `max-child-per-ip` são opcionais.

`wait|nowait` indica se o serviço pode ou não manipular seu próprio socket. Os tipos de socket `dgram` devem usar `wait` enquanto os daemons `stream`, que geralmente são multi-threaded, devem usar `nowait`. `wait` geralmente passa vários sockets para um único daemon, enquanto `nowait` gera um daemon filho para cada novo socket.

O número máximo de daemons inetd que podem aparecer é definido por `max-child`. Por exemplo, para limitar dez instâncias do daemon, coloque um `/10` após o `nowait`. Especificar `/0` permite um número ilimitado de filhos.

`max-connections-per-ip-per-minute` limita o número de conexões de qualquer endereço específico de IP por minuto. Quando o limite for atingido, outras conexões desse endereço IP serão descartadas até o final do minuto. Por exemplo, um valor de `/10` limitaria qualquer endereço IP específico a dez tentativas de conexão por minuto. `max-child-per-ip` limita o número de processos-filhos que podem ser iniciados em nome de um único endereço IP a qualquer momento. Essas opções podem limitar o consumo excessivo de recursos e ajudar a impedir ataques de negação de serviço (DoS (Denial Of Service)).

Um exemplo pode ser visto nas configurações padrão para fingerd(8):

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -k -s
```

usuário

O nome de usuário que o daemon será executado como. Daemons geralmente são executados como `root`, `daemon`, ou `nobody`.

programa servidor

O caminho completo para o daemon. Se o daemon for um serviço fornecido pelo inetd internamente, use `internal`.

argumentos do programa servidor

Usado para especificar qualquer argumento de comando a ser transmitido ao daemon na chamada. Se o daemon for um serviço interno, use `internal`.

29.2.2. Opções de linha de comando

Como a maioria dos daemons de servidor, o inetd tem várias opções que podem ser usadas para modificar seu comportamento. Por padrão, inetd é iniciado com `-wW -C 60`. Essas opções ativam TCP wrappers para todos os serviços, incluindo serviços internos, e impedem que qualquer endereço de IP solicite qualquer serviço mais de 60 vezes por minuto.

Para alterar as opções padrão que são passadas para inetd, adicione uma entrada para `inetd_flags` no arquivo /etc/rc.conf. Se o inetd já estiver em execução, reinicie-o com `service inetd restart`.

As opções disponíveis de limitação de taxa são:

-c máximo

Especifique o número máximo padrão de chamadas simultâneas de cada serviço, em que o padrão é ilimitado. Pode ser sobrescrito com base no serviço usando `max-child` em /etc/inetd.conf.

-C taxa

Especifique o número máximo padrão de vezes por minuto que um serviço pode ser chamado a partir de um único endereço de IP. Pode ser substituído com base no serviço usando `max-connections-per-ip-por-minute` em /etc/inetd.conf.

-R taxa

Especifique o número máximo de vezes que um serviço pode ser chamado em um minuto, em que o padrão é `256`. Uma taxa de `0` permite um número ilimitado.

-s máximo

Especifique o número máximo de vezes que um serviço pode ser chamado a partir de um único endereço IP a qualquer momento, em que o padrão é ilimitado. Pode ser sobrescrito com base no serviço usando `max-child-per-ip` no arquivo /etc/inetd.conf.

Opções adicionais estão disponíveis. Consulte inetd(8) para a lista completa de opções.

29.2.3. Considerações de segurança

Muitos dos daemons que podem ser gerenciados pelo inetd não são conscientes da segurança. Alguns daemons, como fingerd, podem fornecer informações que podem ser úteis para um invasor.

Ative apenas os serviços necessários e monitore o sistema para tentativas excessivas de conexão. `max-connections-per-ip-por-minute`, `max-child` e `max-child-per-ip` podem ser usados para limitar tais ataques.

Por padrão, TCP wrappers estão ativados. Consulte hosts_access(5) para obter mais informações sobre como colocar restrições TCP em vários daemons chamados pelo inetd.

29.3. Network File System (NFS)

O FreeBSD suporta o Network File System (NFS), que permite que um servidor compartilhe diretórios e arquivos com clientes através de uma rede. Com o NFS, os usuários e programas podem acessar arquivos em sistemas remotos como se estivessem armazenados localmente.

NFS tem muitos usos práticos. Alguns dos usos mais comuns incluem:

- Os dados que seriam duplicados em cada cliente podem ser mantidos em um único local e acessados por clientes na rede.
- Vários clientes podem precisar de acesso ao diretório /usr/ports/distfiles. Compartilhar esse diretório permite acesso rápido aos arquivos fonte sem precisar baixá-los para cada cliente.
- Em grandes redes, geralmente é mais conveniente configurar um servidor central NFS no qual todos os diretórios home dos usuários são armazenados. Os usuários podem logar em um cliente em qualquer lugar da rede e ter acesso aos seus diretórios home.
- A administração de exports do NFS é simplificada. Por exemplo, há apenas um sistema de arquivos no qual as políticas de segurança ou de backup devem ser definidas.
- Dispositivos removíveis de armazenamento de mídia podem ser usados por outras máquinas na rede. Isso reduz o número de dispositivos em toda a rede e fornece um local centralizado para gerenciar sua segurança. Geralmente, é mais conveniente instalar software em várias máquinas a partir de uma mídia de instalação centralizada.

O NFS consiste em um servidor e um ou mais clientes. O cliente acessa remotamente os dados armazenados na máquina do servidor. Para que isso funcione corretamente, alguns processos precisam ser configurados e executados.

Esses daemons devem estar em execução no servidor:

Daemon	Descrição
nfsd	O daemon NFS que atende a solicitações de clientes NFS.
mountd	O daemon de montagem do NFS que realiza solicitações recebidas do nfsd.
rpcbind	Este daemon permite que clientes NF descubram qual porta o servidor NFS está usando.

A execução de nfsiod(8) no cliente pode melhorar o desempenho, mas não é necessária.

29.3.1. Configurando o Servidor

Os sistemas de arquivos que o servidor NFS irá compartilhar são especificados no arquivo /etc/exports. Cada linha neste arquivo especifica um sistema de arquivos a ser exportado, quais clientes têm acesso a esse sistema de arquivos e quaisquer opções de acesso. Ao adicionar entradas a este arquivo, cada sistema de arquivos exportado, suas propriedades e hosts permitidos devem ocorrer em uma única linha. Se nenhum cliente estiver listado na entrada, qualquer cliente na rede poderá montar esse sistema de arquivos.

As seguintes entradas no arquivo /etc/exports demonstram como exportar sistemas de arquivos. Os exemplos podem ser modificados para corresponder aos sistemas de arquivos e nomes de clientes na rede do leitor. Existem muitas opções que podem ser usadas neste arquivo, mas apenas algumas serão mencionadas aqui. Veja exports(5) para a lista completa de opções.

Este exemplo mostra como exportar /cdrom para três hosts chamados *alpha*, *bravo* e *charlie*:

```
/cdrom -ro alpha bravo charlie
```

A flag `-ro` torna o sistema de arquivos somente leitura, impedindo que os clientes façam alterações no sistema de arquivos exportado. Este exemplo assume que os nomes de host estão no DNS ou no arquivo /etc/hosts. Consulte hosts(5) se a rede não tiver um servidor de DNS.

O próximo exemplo exporta /home para três clientes pelo endereço IP. Isso pode ser útil para redes sem DNS ou /etc/hosts. A flag `-alldirs` permite que os subdiretórios sejam pontos de montagem. Em outras palavras, ele não montará automaticamente os subdiretórios, mas permitirá que o cliente monte os diretórios necessários conforme necessário.

```
/usr/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

Este próximo exemplo exporta /a para que dois clientes de domínios diferentes possam acessar esse sistema de arquivos. `-maproot=root` permite que o usuário `root` no sistema remoto grave os dados no sistema de arquivos exportado como `root`. Se `-maproot=root` não for especificado, o usuário `root` do cliente será mapeado para a conta `nobody` do servidor e estará sujeito às limitações de acesso definidas para `nobody`.

```
/a  -maproot=root  host.example.com box.example.org
```

Um cliente só pode ser especificado uma vez por sistema de arquivos. Por exemplo, se /usr for um único sistema de arquivos, essas entradas serão inválidas, já que ambas as entradas especificam o mesmo host:

```
# Invalid when /usr is one file system
/usr/src   client
/usr/ports client
```

O formato correto para essa situação é usar uma entrada:

```
/usr/src /usr/ports  client
```

A seguir, um exemplo de uma lista de exportação válida, em que /usr e /exports são sistemas de arquivos locais:

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports               client02
# The client machines have root and can mount anywhere
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

Para habilitar os processos requeridos pelo servidor NFS no momento da inicialização, adicione estas opções ao arquivo /etc/rc.conf:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_enable="YES
```

O servidor pode ser iniciado agora executando este comando:

```
# service nfsd start
```

Sempre que o servidor NFS for iniciado, o mountd também é iniciado automaticamente. No entanto, mountd lê apenas /etc/exports quando é iniciado. Para fazer as edições subsequentes de /etc/exports entrarem em vigor imediatamente, force mountd para ler novamente:

```
# service mountd reload
```

29.3.2. Configurando o Cliente

Para ativar clientes NFS, defina essa opção no arquivo /etc/rc.conf de cada cliente:

```
nfs_client_enable="YES"
```

Em seguida, execute este comando em cada cliente NFS:

```
# service nfsclient start
```

O cliente agora tem tudo de que precisa para montar um sistema de arquivos remoto. Nestes exemplos, o nome do servidor é `server` e o nome do cliente é `client`. Para montar /home no `server` para o ponto de montagem /mnt no `client`:

```
# mount server:/home /mnt
```

Os arquivos e diretórios em /home agora estarão disponíveis no `client`, no diretório /mnt.

Para montar um sistema de arquivos remoto toda vez que o cliente for inicializado, adicione-o ao arquivo /etc/fstab:

```
server:/home    /mnt    nfs rw  0   0
```

Consulte fstab(5) para obter uma descrição de todas as opções disponíveis.

29.3.3. Bloqueando

Alguns aplicativos exigem o bloqueio de arquivos para funcionar corretamente. Para ativar o bloqueio, adicione estas linhas ao arquivo /etc/rc.conf no cliente e no servidor:

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Então inicie as aplicações:

```
# service lockd start
# service statd start
```

Se o bloqueio não for necessário no servidor, o cliente NFS pode ser configurado para bloquear localmente incluindo `-L` ao executar o mount. Consulte mount_nfs(8) para mais detalhes.

29.3.4. Automatizando Montagens com autofs(5)



O recurso de montagem automática autofs(5) é suportado a partir do FreeBSD 10.1-RELEASE. Para usar a funcionalidade automounter em versões mais antigas do FreeBSD, use amd(8). Este capítulo descreve apenas o montador automático autofs(5).

O recurso autofs(5) é um nome comum para vários componentes que, juntos, permitem a montagem automática de sistemas de arquivos locais e remotos sempre que um arquivo ou diretório dentro desse sistema de arquivos é acessado. Ele consiste no componente do kernel, autofs(5) e vários aplicativos no espaço do usuário: automount(8), automountd(8) e autounmountd(8). Ele serve como uma alternativa para amd(8) de versões anteriores do FreeBSD. Amd ainda é fornecido para fins de compatibilidade com versões anteriores, já que os dois usam formato de mapeamento diferentes; o usado pelo autofs é o mesmo que com outros

automontadores do SVR4, como os do Solaris, MacOS X e Linux.

O sistema de arquivos virtual autofs(5) é montado em pontos de montagem especificados por automount(8), geralmente chamado durante a inicialização.

Sempre que um processo tentar acessar o arquivo dentro do ponto de montagem autofs(), o kernel notificará o daemon automountd(8) e irá pausar o processo de disparo. O daemon automountd(8) processará as solicitações do kernel localizando o mapeamento apropriado e irá montar o sistema de arquivos de acordo com ele, então sinaliza ao kernel para liberar o processo bloqueado . O daemon autounmountd(8) desmonta automaticamente os sistemas de arquivos montados automaticamente após algum tempo, a menos que eles ainda estejam sendo usados.

O arquivo de configuração principal do autofs é o /etc/auto_master. Atribui mapeamentos individuais a montagens de nível superior. Para uma explicação do auto_master e da sintaxe do mapeamento, consulte auto_master(5).

Existe um mapeamento especial montado automaticamente em /net. Quando um arquivo é acessado dentro desse diretório, o autofs(5) procura a montagem remota correspondente e monta-a automaticamente. Por exemplo, uma tentativa de acessar um arquivo dentro de /net/foobar/usr informaria automountd(8) para montar a exportação /usr do host `foobar`.

Exemplo 46. Montando uma Exportação com autofs(5)

Neste exemplo, `showmount -e` mostra os sistemas de arquivos exportados que podem ser montados a partir do servidor NFS, `foobar`:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /net/foobar/usr
```

A saída de `showmount` mostra /usr como uma exportação. Ao alterar os diretórios para /host/foobar/usr, o automountd(8) intercepta o pedido e tenta resolver o nome do host `foobar`. Se for bem-sucedido, automountd(8) montará automaticamente a exportação de origem.

Para habilitar autofs(5) no momento da inicialização, adicione esta linha ao arquivo /etc/rc.conf:

```
autofs_enable="YES"
```

Em seguida, autofs(5) pode ser iniciado executando:

```
# service automount start
# service automountd start
# service autounmountd start
```

O formato de mapeamento de autofs(5) é o mesmo que em outros sistemas operacionais. Informações sobre este formato de outras fontes podem ser úteis, como o documento do Mac OS X.

Consulte as páginas de manuais automount(8), automountd(8), autounmountd(8) e auto_master(5) para maiores informações.

29.4. Sistema de Informação de Rede (NIS)

O Network Information System (NIS) foi projetado para centralizar a administração de sistemas UNIX™ como Solaris™, HP-UX, AIX™, Linux, NetBSD, OpenBSD e FreeBSD. O NIS era originalmente conhecido como Yellow Pages, mas o nome foi alterado devido a problemas de marca registrada. Esta é a razão pela qual os comandos do NIS começam com `yp`.

O NIS é um sistema cliente/servidor baseado em Remote Procedure Call (RPC) que permite que um grupo de máquinas dentro de um domínio NIS compartilhe um conjunto de arquivos de configuração. Isso permite que um administrador do sistema configure sistemas clientes NIS com apenas dados mínimos de configuração e adicione, remova ou modifique dados de configuração de um único local.

O FreeBSD usa a versão 2 do protocolo NIS.

29.4.1. Termos do NIS e Processos

A Tabela 28.1 resume os termos e processos importantes usados pelo NIS:

Tabela 23. Terminologia do NIS

Termo	Descrição
nome de domínio NIS	Os servidores e clientes do NIS compartilham um nome de domínio NIS. Normalmente, esse nome não tem nada a ver com DNS.
rpcbind(8)	Este serviço habilita o RPC e deve estar rodando para rodar um servidor NIS ou atuar como um cliente NIS.
ypbind(8)	Este serviço liga um cliente NIS ao seu servidor NIS. Ele levará o nome de domínio NIS e usará RPC para se conectar ao servidor. É o núcleo da comunicação cliente/servidor em um ambiente NIS. Se este serviço não estiver sendo executado em uma máquina cliente, ele não poderá acessar o servidor NIS.

Termo	Descrição
ypserv(8)	Este é o processo para o servidor NIS. Se este serviço parar de funcionar, o servidor não poderá mais responder aos pedidos do NIS, portanto, esperamos que exista um servidor slave para assumir o controle. Alguns clientes não-FreeBSD não tentarão se reconectar usando um servidor slave e o processo ypbind pode precisar ser reiniciado nesses clientes.
rpc.yppasswdd(8)	Este processo só é executado em servidores principais de NIS. Este daemon permite que clientes NIS alterem suas senhas do NIS. Se este daemon não estiver rodando, os usuários terão que acessar o servidor principal do NIS e alterar suas senhas lá.

29.4.2. Tipos de Máquinas

Existem três tipos de hosts em um ambiente NIS:

- Servidor NIS master

  Esse servidor atua como um repositório central para as informações de configuração do host e mantém a cópia autoritativa dos arquivos usados por todos os clientes do NIS. O passwd, o group e outros arquivos usados pelos clientes do NIS são armazenados no servidor master. Embora seja possível que uma máquina seja um servidor NIS master para mais de um domínio NIS, esse tipo de configuração não será abordado neste capítulo, pois pressupõe ambiente NIS de pequena escala.

- Servidores NIS slave

  Os servidores slaves do NIS mantêm cópias dos arquivos de dados do master do NIS para fornecer redundância. Os servidores slaves também ajudam a balancear a carga do servidor master, pois os clientes do NIS sempre se conectam ao servidor do NIS que responde primeiro.

- Clientes NIS

  Os clientes do NIS autenticam-se contra o servidor NIS durante o logon.

Informações em muitos arquivos podem ser compartilhadas usando o NIS . Os arquivos master.passwd, group e hosts são comumente compartilhados via NIS. Sempre que um processo em um cliente precisa de informações que normalmente seriam encontradas nesses arquivos localmente, ele faz uma consulta ao servidor NIS ao qual está vinculado.

29.4.3. Considerações de Planejamento

Esta seção descreve um ambiente NIS de exemplo que consiste em 15 máquinas FreeBSD sem ponto de administração centralizado. Cada máquina tem seu próprio /etc/passwd e /etc/master.passwd.

Esses arquivos são mantidos em sincronia entre si somente por meio de intervenção manual. Atualmente, quando um usuário é adicionado ao laboratório, o processo deve ser repetido em todas as 15 máquinas.

A configuração do laboratório será a seguinte:

Nome da maquina	Endereço IP	Role da máquina
`ellington`	`10.0.0.2`	NIS master
`coltrane`	`10.0.0.3`	NIS slave
`basie`	`10.0.0.4`	Estação de Trabalho da Facultativa
`bird`	`10.0.0.5`	Máquina Cliente
`cli[1-11]`	`10.0.0.[6-17]`	Outras Máquinas Clientes

Se esta é a primeira vez que um esquema de NIS está sendo desenvolvido, ele deve ser cuidadosamente planejado através do tempo. Independentemente do tamanho da rede, várias decisões precisam ser tomadas como parte do processo de planejamento.

29.4.3.1. Escolhendo um Nome de Domínio NIS

Quando um cliente transmite suas solicitações de informações, ele inclui o nome do domínio NIS do qual faz parte. É assim que vários servidores em uma rede podem informar qual servidor deve responder a qual solicitação. Pense no nome de domínio NIS como o nome de um grupo de hosts.

Algumas organizações optam por usar o nome de domínio da Internet para o nome de domínio NIS. Isso não é recomendado, pois pode causar confusão ao tentar depurar problemas de rede. O nome de domínio NIS deve ser único dentro da rede e é útil se ele descrever o grupo de máquinas que representa. Por exemplo, o departamento de Arte da Acme Inc. pode estar no domínio NIS"acme-art". Este exemplo usará o nome de domínio `test-domain`.

No entanto, alguns sistemas operacionais não-FreeBSD exigem que o nome de domínio NIS seja o mesmo que o nome de domínio da Internet. Se uma ou mais máquinas na rede tiverem essa restrição, o nome de domínio da Internet *deve* ser usado como o nome de domínio NIS.

29.4.3.2. Requisitos Físicos do Servidor

Há várias coisas que você deve ter em mente ao escolher uma máquina para usar como um servidor NIS. Como os clientes do NIS dependem da disponibilidade do servidor, escolha uma máquina que não seja reinicializada com freqüência. O servidor do NIS deve idealmente ser uma máquina autônoma cujo único propósito seja ser um servidor NIS. Se a rede não for muito usada, é aceitável colocar o servidor NIS em uma máquina que executa outros serviços. No entanto, se o servidor NIS ficar indisponível, isso afetará negativamente todos os clientes NIS.

29.4.4. Configurando o Servidor NIS Master

As cópias canônicas de todos os arquivos NIS são armazenadas no servidor master. Os bancos de dados usados para armazenar as informações são chamados de mapas de NIS. No FreeBSD, estes

mapas são armazenados em /var/yp/[nome_do_domínio] onde [nome_do_dominio] é o nome do domínio NIS. Como vários domínios são suportados, é possível ter vários diretórios, um para cada domínio. Cada domínio terá seu próprio conjunto independente de mapas.

Os servidores master e slave do NIS lidam com todas as requisições NIS através do ypserv(8). Esse daemon é responsável por receber solicitações de entrada de clientes NIS, traduzindo o domínio e o nome do mapa solicitados para um caminho para o arquivo de banco de dados correspondente e transmitindo dados do banco de dados de volta ao cliente.

Configurar um servidor NIS master pode ser relativamente simples, dependendo das necessidades ambientais. Como o FreeBSD oferece suporte a NIS embutido, ele só precisa ser ativado adicionando as seguintes linhas ao arquivo /etc/rc.conf:

```
nisdomainname="test-domain" ①
nis_server_enable="YES"     ②
nis_yppasswdd_enable="YES"  ③
```

① Esta linha define o nome de domínio NIS para `test-domain`.

② Isto automatiza o início dos processos do servidor NIS quando o sistema é inicializado.

③ Isso habilita o daemon rpc.yppasswdd(8) para que os usuários possam alterar sua senha NIS de uma máquina cliente.

É preciso ter cuidado em um domínio com vários servidores, no qual as máquinas do servidor também são clientes NIS. Geralmente, é uma boa ideia forçar os servidores a fazerem bind em si mesmos, em vez de permitir que eles transmitam solicitações de bind e, possivelmente, fiquem vinculados um ao outro. Modos de falha estranhos podem ocorrer se um servidor cair e outros dependerem dele. Eventualmente, todos os clientes terão tempo limite e tentarão fazer bind em outros servidores, mas o atraso envolvido poderá ser considerável e o modo de falha ainda estará presente, uma vez que os servidores podem ligar-se entre si novamente.

Um servidor que também é um cliente pode ser forçado fazer bind em um servidor em particular adicionando estas linhas adicionais ao arquivo /etc/rc.conf:

```
nis_client_enable="YES"             ①
nis_client_flags="-S test-domain,server"    ②
```

① Isso permite rodar coisas do cliente também.

② Esta linha define o nome de domínio NIS para test-domain e vincula para si mesmo.

Depois de salvar as edições, digite `/etc/netstart` para reiniciar a rede e aplicar os valores definidos no arquivo /etc/rc.conf. Antes de inicializar os mapas de NIS, inicie ypserv(8):

```
# service ypserv start
```

29.4.4.1. Inicializando os mapas do NIS

Os mapeamentos NIS são gerados a partir dos arquivos de configuração no diretório /etc no NIS master, com uma exceção: /etc/master.passwd. Isso evita a propagação de senhas para todos os servidores no domínio NIS. Portanto, antes de inicializar os mapas do NIS, configure os arquivos de senha primários:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

É aconselhável remover todas as entradas de contas do sistema, bem como quaisquer contas de usuário que não precisem ser propagadas para os clientes do NIS, como o `root` e quaisquer outras contas administrativas.



Assegure-se de que o arquivo /var/yp/master.passwd não seja de grupo nem de mundo legível, definindo suas permissões para `600`.

Depois de concluir esta tarefa, inicialize os mapas do NIS. O FreeBSD inclui o script ypinit(8) para fazer isso. Ao gerar mapas para o servidor master, inclua `-m` e especifique o nome de domínio NIS:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If not, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..output from map generation..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

Isto irá criar /var/yp/Makefile a partir de /var/yp/Makefile.dist. Por padrão, este arquivo assume que o ambiente tem um único servidor NIS com apenas clientes FreeBSD. Como `test-domain` tem um servidor slave, edite esta linha no arquivo /var/yp/Makefile para que comece com um comentário (

#) :

```
NOPUSH = "True"
```

29.4.4.2. Adicionando novos usuários

Toda vez que um novo usuário é criado, a conta de usuário deve ser adicionada ao servidor mestre NIS e aos mapeamentos do NIS reconstruídos. Até que isso ocorra, o novo usuário não poderá efetuar login em nenhum lugar, exceto no NIS master. Por exemplo, para adicionar o novo usuário `jsmith` ao domínio `test-domain`, execute estes comandos no servidor master:

```
# pw useradd jsmith
# cd /var/yp
# make test-domain
```

O usuário também pode ser adicionado usando `adduser jsmith` em vez de `pw useradd smith`.

29.4.5. Configurando um Servidor NIS Slave

Para configurar um servidor NIS slave, faça o logon no servidor slave e edite o arquivo /etc/rc.conf assim como para o servidor master. Não gere nenhum mapa de NIS, pois estes já existem no servidor master. Ao executar `ypinit` no servidor slave, use `-s` (para slave) ao invés de `-m` (para master). Esta opção requer o nome do NIS master, além do nome do domínio, como visto neste exemplo:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If not, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
```

```
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Remember to update map ypservers on ellington.
```

Isto irá gerar um diretório no servidor slave chamado /var/yp/test-domain que contém cópias dos mapas do servidor principal do NIS. Adicionar estas entradas ao arquivo /etc/crontab em cada servidor slave forçará os slaves a sincronizar seus mapas com os mapas no servidor master:

```
20      *       *       *       *       root    /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root    /usr/libexec/ypxfr passwd.byuid
```

Essas entradas não são obrigatórias porque o servidor master tenta enviar automaticamente quaisquer alterações no mapa para seus escravos. No entanto, como os clientes podem depender do servidor escravo para fornecer informações corretas de senha, recomenda-se forçar atualizações frequentes de mapas de senha. Isso é especialmente importante em redes ocupadas nas quais as atualizações de mapas nem sempre são concluídas.

Para finalizar a configuração, execute `/etc/netstart` no servidor slave para iniciar os serviços do

NIS.

29.4.6. Configurando um cliente NIS

Um cliente NIS é vinculado a um servidor NIS usando ypbind(8). Esse daemon transmite solicitações de RPC na rede local. Essas solicitações especificam o nome do domínio configurado no cliente. Se um servidor NIS no mesmo domínio receber uma das transmissões, ele responderá a ypbind, que registrará o endereço do servidor. Se houver vários servidores disponíveis, o cliente usará o endereço do primeiro servidor para responder e direcionará todas as suas solicitações de NIS para esse servidor. O cliente irá automaticamente pingar o servidor regularmente para garantir que ainda esteja disponível. Se ele não receber uma resposta dentro de um período de tempo razoável, o ypbind marcará o domínio como não acoplado e começará a transmitir novamente na esperança de localizar outro servidor.

Para configurar uma máquina FreeBSD para ser um cliente NIS:

1. Edite o /etc/rc.conf e adicione as seguintes linhas para definir o nome de domínio NIS e inicie ypbind(8) durante a inicialização da rede:

   ```
   nisdomainname="test-domain"
   nis_client_enable="YES"
   ```

2. Para importar todas as possíveis entradas de senha do servidor NIS, use `vipw` para remover todas as contas de usuário, exceto uma do arquivo /etc/master.passwd. Ao remover as contas, lembre-se de que pelo menos uma conta local deve permanecer e essa conta deve ser membro do grupo `wheel`. Se houver um problema com o NIS, essa conta local poderá ser usada para efetuar login remotamente, tornar-se o superusuário e corrigir o problema. Antes de salvar as edições, adicione a seguinte linha ao final do arquivo:

   ```
   +::::::::::
   ```

   Esta linha configura o cliente para fornecer qualquer pessoa com uma conta válida na senha do servidor do NIS mapeia uma conta no cliente. Existem várias maneiras de configurar o cliente NIS modificando essa linha. Um método é descrito em Usando Netgroups. Para uma leitura mais detalhada, consulte o livro `Managing NFS and NIS`, publicado pela O'Reilly Media.

3. Para importar todas as entradas de grupo possíveis do servidor NIS, adicione esta linha ao /etc/group:

   ```
   +:*::
   ```

Para iniciar imediatamente o cliente NIS, execute os seguintes comandos como superusuário:

```
# /etc/netstart
# service ypbind start
```

Depois de concluir estas etapas, a execução do `ypcat passwd` no cliente deve mostrar o mapa passwd do servidor.

29.4.7. Segurança NIS

Como o RPC é um serviço baseado em broadcast, qualquer sistema executando o ypbind dentro do mesmo domínio pode recuperar o conteúdo dos mapas do NIS. Para evitar transações não autorizadas, ypserv(8) suporta um recurso chamado "securenets" que pode ser usado para restringir o acesso a um dado conjunto de hosts. Por padrão, essas informações são armazenadas no arquivo /var/yp/securenets, a menos que ypserv(8) seja iniciado com `-p` e um caminho alternativo. Este arquivo contém entradas que consistem em uma especificação de rede e uma máscara de rede separadas por espaço em branco. Linhas iniciando com `#` são consideradas comentários. Um exemplo de securenets pode ser assim:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

Se ypserv(8) receber uma solicitação de um endereço que corresponda a uma dessas regras, ela processará a solicitação normalmente. Se o endereço não corresponder a uma regra, a solicitação será ignorada e uma mensagem de aviso será registrada. Se o securenets não existir, o `ypserv` permitirá conexões de qualquer host.

TCP Wrapper é um mecanismo alternativo para fornecer controle de acesso em vez de securenets. Embora o mecanismo de controle de acesso acrescente alguma segurança, ambos são vulneráveis a ataques como "IP spoofing". Todo o tráfego relacionado a NIS deve ser bloqueado no firewall.

Servidores que usam securenets podem não servir clientes legítimos de NIS com implementações arcaicas de TCP/IP. Algumas dessas implementações definem todos os bits do host como zero ao fazer transmissões ou não observam a máscara de sub-rede ao calcular o endereço de transmissão. Embora alguns desses problemas possam ser corrigidos alterando a configuração do cliente, outros problemas podem forçar a desativação desses sistemas clientes ou o abandono do securenets.

O uso de TCP Wrapper aumenta a latência do servidor NIS. O atraso adicional pode ser longo o suficiente para causar timeouts em programas clientes, especialmente em redes ocupadas com servidores NIS lentos. Se um ou mais clientes sofrerem de latência, converta esses clientes em servidores de NIS slaves e force-os a se ligarem a eles mesmos.

29.4.7.1. Barrando alguns usuários

Neste exemplo, o sistema `basie` é uma estação de trabalho da dentro do domínio NIS facultativo. O mapa passwd no servidor NIS master contém contas para professores e alunos. Esta seção demonstra como permitir o login do corpo docente neste sistema e, ao mesmo tempo, recusar logins de alunos.

Para previnir usuários específicos de logar em um sistema, desde que eles estejam presentes no banco de dados do NIS, use `vipw` para adicionar `-username` com o numero correto de virgulas em direção ao fim do arquivo /etc/master.passwd no cliente, onde *username* é o nome de usuário a impedir de logar. A linha com o usuário bloqueado deve estar antes da linha `+` que permite usuários do NIS. Neste exemplo, `bill` está impedido de logar no `basie`:

```
basie# cat /etc/master.passwd
root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5::0:0:System &:/:/usr/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/usr/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/usr/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/shared/man:/usr/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/usr/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/usr/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/usr/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/usr/sbin/nologin
-bill:::::::::
+:::::::::

basie#
```

29.4.8. Usando Netgroups

A exclusão de usuários especificados do logon em sistemas individuais torna-se imprestável em redes maiores e perde rapidamente o principal benefício do NIS: administração *centralizada*.

Os netgroups foram desenvolvidos para lidar com redes grandes e complexas com centenas de usuários e máquinas. Seu uso é comparável aos grupos UNIX™, onde a principal diferença é a falta de um ID numérico e a capacidade de definir um netgroup incluindo contas de usuário e outros netgroups.

Para expandir o exemplo usado neste capítulo, o domínio NIS será estendido para adicionar os usuários e sistemas mostrados nas Tabelas 28.2 e 28.3:

Tabela 24. Usuários Adicionais

Nome(s) de usuário	Descrição
alpha, beta	Funcionários do departamento de TI
charlie, delta	Aprendizes do departamento de TI
echo, foxtrott, golf, ...	funcionários
able, baker, ...	estagiários

Tabela 25. Sistemas Adicionais

Nome(s) de máquina	Descrição
war, death, famine, pollution	Somente funcionários de TI podem fazer logon nesses servidores.
pride, greed, envy, wrath, lust, sloth	Todos os membros do departamento de TI podem fazer login nesses servidores.
one, two, three, four, ...	Estações de trabalho comuns usadas pelos funcionários.
trashcan	Uma máquina muito antiga sem dados críticos. Até os estagiários podem usar este sistema.

Ao usar netgroups para configurar esse cenário, cada usuário é atribuído a um ou mais netgroups e os logins são permitidos ou proibidos para todos os membros do netgroup. Ao adicionar uma nova máquina, as restrições de login devem ser definidas para todos os netgroups. Quando um novo usuário é adicionado, a conta deve ser adicionada a um ou mais netgroups. Se a configuração do NIS for planejada com cuidado, somente um arquivo de configuração central precisará ser modificado para conceder ou negar acesso a máquinas.

O primeiro passo é a inicialização do mapa do NIS `netgroup`. No FreeBSD, este mapa não é criado por padrão. No servidor NIS master, use um editor para criar um mapa chamado /var/yp/netgroup.

Este exemplo cria quatro grupos de rede para representar funcionários de TI, aprendizes de TI, funcionários e estagiários:

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
        (,golf,test-domain)
INTERNS (,able,test-domain)     (,baker,test-domain)
```

Cada entrada configura um netgroup. A primeira coluna em uma entrada é o nome do netgroup. Cada conjunto de colchetes representa um grupo de um ou mais usuários ou o nome de outro grupo de rede. Ao especificar um usuário, os três campos delimitados por vírgula dentro de cada grupo representam:

1. O nome do(s) host(s) onde os outros campos que representam o usuário são válidos. Se um nome de host não for especificado, a entrada será válida em todos os hosts.
2. O nome da conta que pertence a este netgroup.

3. O domínio NIS da conta. As contas podem ser importadas de outros domínios do NIS para um netgroup.

Se um grupo contiver vários usuários, separe cada usuário com espaço em branco. Além disso, cada campo pode conter curingas. Veja netgroup(5) para detalhes.

Nomes de grupos maiores que 8 caracteres não devem ser usados. Os nomes diferenciam maiúsculas de minúsculas e usar letras maiúsculas para nomes de grupos de rede é uma maneira fácil de distinguir entre nomes de usuários, máquinas e grupos de rede.

Alguns clientes não-FreeBSD NIS não podem lidar com netgroups contendo mais de 15 entradas. Esse limite pode ser contornado criando vários grupos de sub-redes com 15 usuários ou menos e um grupo de rede real consistindo dos grupos de sub-redes, como visto neste exemplo:

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
```

Repita este processo se mais de 225 (15 vezes 15) usuários existirem dentro de um único netgroup.

Para ativar e distribuir o novo mapa do NIS:

```
ellington# cd /var/yp
ellington# make
```

Isso gerará os três mapas NIS, netgroup, netgroup.byhost e netgroup.byuser. Use a opção de chave de mapa ypcat(1) para verificar se os novos mapas de NIS estão disponíveis:

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

A saída do primeiro comando deve lembrar o conteúdo de /var/yp/netgroup. O segundo comando só produz saída se os netgroups específicos do host foram criados. O terceiro comando é usado para obter a lista de netgroups de um usuário.

Para configurar um cliente, use vipw(8) para especificar o nome do netgroup. Por exemplo, no servidor chamado `war`, substitua esta linha:

```
+:::::::::
```

com

```
+@IT_EMP:::::::::
```

Isso especifica que apenas os usuários definidos no netgroup `IT_EMP` serão importados para o banco de dados de senhas deste sistema e somente esses usuários terão permissão para efetuar login nesse sistema.

Essa configuração também se aplica à função `~` do shell e a todas as rotinas que convertem entre nomes de usuário e IDs de usuário numérico. Em outras palavras, `cd ~user` não funcionará, `ls -l` mostrará o ID numérico em vez do nome de usuário e `find . -user joe -print` falhará com a mensagem `No such user`. Para corrigir isso, importe todas as entradas do usuário sem permitir que elas efetuem login nos servidores. Isto pode ser conseguido adicionando uma linha extra:

```
+::::::::::/usr/sbin/nologin
```

Esta linha configura o cliente para importar todas as entradas, mas para substituir o shell nessas entradas com /usr/sbin/nologin.

Certifique-se que a linha extra é colocada *após* `+@IT_EMP:::::::::`. Caso contrário, todas as contas de usuário importadas do NIS terão /usr/sbin/nologin como seu shell de login e ninguém poderá efetuar o login no sistema.

Para configurar os servidores menos importantes, substitua o antigo `+::::::::::` nos servidores com estas linhas:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+::::::::::/usr/sbin/nologin
```

As linhas correspondentes para as estações de trabalho seriam:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+::::::::::/usr/sbin/nologin
```

O NIS suporta a criação de grupos de rede de outros grupos de rede, o que pode ser útil se a política relacionada ao acesso do usuário for alterada. Uma possibilidade é a criação de netgroups baseados em funções. Por exemplo, pode-se criar um netgroup chamado `BIGSRV` para definir as restrições de login para os servidores importantes, outro grupo de rede chamado `SMALLSRV` para os servidores menos importantes e um terceiro netgroup chamado `USERBOX` para as estações de trabalho. Cada um desses netgroups contém os netgroups com permissão para efetuar login nessas máquinas. As novas entradas para o mapa do NIS `netgroup` seriam assim:

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS
```

Esse método de definir restrições de login funciona razoavelmente bem quando é possível definir

grupos de máquinas com restrições idênticas. Infelizmente, esta é a exceção e não a regra. Na maioria das vezes, é necessária a capacidade de definir restrições de login por máquina.

As definições de netgroup específicas da máquina são outra possibilidade para lidar com as mudanças na política. Neste cenário, o /etc/master.passwd de cada sistema contém duas linhas que começam com "+". A primeira linha adiciona um netgroup com as contas permitidas para entrar nesta máquina e a segunda linha adiciona todas as outras contas com /usr/sbin/nologin como shell. Recomenda-se usar a versão "ALL-CAPS" do nome do host como o nome do netgroup:

```
+@BOXNAME:::::::::
+:::::::::/usr/sbin/nologin
```

Quando esta tarefa estiver completa em todas as máquinas, não haverá mais a necessidade de modificar as versões locais de /etc/master.passwd novamente. Todas as alterações posteriores podem ser manipuladas, modificando o mapa do NIS. Aqui está um exemplo de um possível mapa `netgroup` para este cenário:

```
# Define groups of users first
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Now, define some groups based on roles
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# And a groups for a special tasks
# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)
#
# This one is really important and needs more access restrictions
DEATH     IT_EMP
#
# The anti-virus-machine mentioned above
ONE       SECURITY
#
```

```
# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]
```

Pode não ser sempre aconselhável usar netgroups baseados em máquina. Ao implantar algumas dúzias ou centenas de sistemas, grupos de rede baseados em funções em vez de grupos de rede baseados em máquina podem ser usados para manter o tamanho do mapa do NIS dentro de limites razoáveis.

29.4.9. Formatos de Senha

O NIS requer que todos os hosts em um domínio NIS usem o mesmo formato para criptografar senhas. Se os usuários tiverem problemas para autenticar em um cliente NIS, pode ser devido a um formato de senha diferente. Em uma rede heterogênea, o formato deve ser suportado por todos os sistemas operacionais, onde DES é o padrão comum mais baixo.

Para verificar qual formato um servidor ou cliente está usando, veja esta seção do /etc/login.conf:

```
default:\
    :passwd_format=des:\
    :copyright=/etc/COPYRIGHT:\
    [Further entries elided]
```

Neste exemplo, o sistema está usando o formato DES. Outros valores possíveis são `blf` para Blowfish e `md5` para senhas criptografadas com MD5.

Se o formato em um host precisar ser editado para corresponder ao que está sendo usado no domínio NIS, o banco de dados de recursos de login deve ser reconstruído após salvar a alteração:

```
# cap_mkdb /etc/login.conf
```



O formato das senhas das contas de usuários existentes não será atualizado até que cada usuário mude sua senha *após* o banco de dados de recursos de login ser reconstruído.

29.5. Protocolo leve de acesso de diretório (LDAP)

O protocolo LDAP (LDAP) é um protocolo da camada de aplicação usado para acessar, modificar e autenticar objetos usando um serviço de informações de diretório distribuído. Pense nisso como um telefone ou livro de registro que armazena vários níveis de informações hierárquicas e homogêneas. Ele é usado nas redes do Active Directory e do OpenLDAP e permite que os usuários acessem vários níveis de informações internas utilizando uma única conta. Por exemplo, a autenticação de email, a obtenção de informações de contato dos funcionários e a autenticação interna de sites podem usar uma única conta de usuário na base de registros do servidor LDAP.

Esta seção fornece um guia de início rápido para configurar um servidor LDAP em um sistema

FreeBSD. Ele pressupõe que o administrador já tenha um plano de design que inclua o tipo de informação a ser armazenada, para que essas informações sejam usadas, quais usuários devem ter acesso a essas informações e como proteger essas informações contra acesso não autorizado.

29.5.1. Terminologia e Estrutura do LDAP

O LDAP usa vários termos que devem ser entendidos antes de iniciar a configuração. Todas as entradas de diretório consistem em um grupo de *attributes*. Cada um desses conjuntos de atributos contém um identificador exclusivo conhecido como *Distinguished Name* (DN) que é normalmente criado a partir de vários outros atributos, como Common ou *Relative Distinguished Name* (RDN). Semelhante a como os diretórios têm caminhos absolutos e relativos, considere um DN como um caminho absoluto e o RDN como o caminho relativo.

Um exemplo de entrada LDAP é semelhante ao seguinte. Este exemplo procura a entrada para a conta de usuário especificada (`uid`), unidade organizacional (`ou`) e organização (`o`):

```
% ldapsearch -xb "uid=trhodes,ou=users,o=example.com"
# extended LDIF
#
# LDAPv3
# base <uid=trhodes,ou=users,o=example.com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# trhodes, users, example.com
dn: uid=trhodes,ou=users,o=example.com
mail: trhodes@example.com
cn: Tom Rhodes
uid: trhodes
telephoneNumber: (123) 456-7890

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Esta entrada de exemplo mostra os valores para os atributos `dn`, `mail`, `cn`, `uid` e `telephoneNumber`. O atributo do cn é o RDN.

Maiores informações sobre o LDAP e sua terminologia podem ser encontradas em http://www.openldap.org/doc/admin24/intro.html.

29.5.2. Configurando um servidor LDAP

O FreeBSD não provê um servidor LDAP embutido. Comece a configuração instalando o pacote ou port net/openldap-server:

```
# pkg install openldap-server
```

Aqui está um largo conjunto de opções habilitadas no pacote. Reveja-os rodando o comando `pkg info openldap-server`. Se não for suficiente (por exemplo se o suporte a SQL for necessário), por favor considere recompilar o port usando o framework apropriado.

A instalação cria o diretório /var/db/openldap-data para conter os dados. O diretório para armazenar os certificados deve ser criado:

```
# mkdir /usr/local/etc/openldap/private
```

A próxima fase é configurar a autoridade de certificação. Os seguintes comandos devem ser executados em /usr/local/etc/openldap/private. Isso é importante, pois as permissões de arquivo precisam ser restritivas e os usuários não devem ter acesso a esses arquivos. Informações mais detalhadas sobre certificados e seus parâmetros podem ser encontradas em OpenSSL. Para criar a Autoridade de Certificação, comece com este comando e siga os prompts:

```
# openssl req -days 365 -nodes -new -x509 -keyout ca.key -out ../ca.crt
```

As entradas para os prompts podem ser genéricas *exceto* para o `Common Name`. Esta entrada deve ser *diferente* do nome do host do sistema. Se este será um certificado auto-assinado, prefixe o nome do host com `CA` para a Autoridade de Certificação.

A próxima tarefa é criar uma solicitação de assinatura de certificado e uma chave privada. Insira este comando e siga os prompts:

```
# openssl req -days 365 -nodes -new -keyout server.key -out server.csr
```

Durante o processo de geração de certificados, certifique-se de configurar corretamente o atributo `Common Name`. A Solicitação de Assinatura de Certificado deve ser assinada com a Autoridade de Certificação para ser usada como um certificado válido:

```
# openssl x509 -req -days 365 -in server.csr -out ../server.crt -CA ../ca.crt -CAkey
ca.key -CAcreateserial
```

A parte final do processo de geração de certificados é gerar e assinar os certificados do cliente:

```
# openssl req -days 365 -nodes -new -keyout client.key -out client.csr
# openssl x509 -req -days 3650 -in client.csr -out ../client.crt -CA ../ca.crt -CAkey
ca.key
```

Lembre-se de usar o mesmo atributo `Common Name` quando solicitado. Quando terminar, assegure-se de que um total de oito (8) novos arquivos tenham sido gerado através dos comandos procedentes.

O daemon que executa o servidor OpenLDAP é o slapd. Sua configuração é executada através do slapd.ldif: o antigo slapd.conf foi descontinuado pelo OpenLDAP.

Exemplos de configuração para o slapd.ldif estão disponíveis e também podem ser encontrados em /usr/local/etc/openldap/slapd.ldif.sample. As opções estão documentadas em slapd-config(5). Cada seção do slapd.ldif, como todos os outros conjuntos de atributos LDAP, é identificada exclusivamente por meio de um DN. Certifique-se de que nenhuma linha em branco seja deixada entre a instrução `dn:` e o final desejado da seção. No exemplo a seguir, o TLS será usado para implementar um canal seguro. A primeira seção representa a configuração global:

```
#
# See slapd-config(5) for details on configuration options.
# This file should NOT be world readable.
#
dn: cn=config
objectClass: olcGlobal
cn: config
#
#
# Define global ACLs to disable default read access.
#
olcArgsFile: /var/run/openldap/slapd.args
olcPidFile: /var/run/openldap/slapd.pid
olcTLSCertificateFile: /usr/local/etc/openldap/server.crt
olcTLSCertificateKeyFile: /usr/local/etc/openldap/private/server.key
olcTLSCACertificateFile: /usr/local/etc/openldap/ca.crt
#olcTLSCipherSuite: HIGH
olcTLSProtocolMin: 3.1
olcTLSVerifyClient: never
```

A Autoridade de Certificação, o certificado do servidor e os arquivos de chave privada do servidor devem ser especificados aqui. Recomenda-se que os clientes escolham a opção de criptografia de segurança e omitam `olcTLSCipherSuite` (incompatível com clientes TLS diferentes de openssl). A opção `olcTLSProtocolMin` permite que o servidor exija um nível mínimo de segurança: é recomendado. Enquanto a verificação é obrigatória para o servidor, não é para o cliente: `olcTLSVerifyClient: never`.

A segunda seção é sobre os módulos de backend e pode ser configurada da seguinte maneira:

```
#
# Load dynamic backend modules:
#
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath:  /usr/local/libexec/openldap
olcModuleload:  back_mdb.la
#olcModuleload: back_bdb.la
#olcModuleload: back_hdb.la
```

```
#olcModuleload: back_ldap.la
#olcModuleload: back_passwd.la
#olcModuleload: back_shell.la
```

A terceira seção é dedicada a carregar os esquemas `ldif` necessários para serem usados pelos bancos de dados: eles são essenciais.

```
dn: cn=schema,cn=config
objectClass: olcSchemaConfig
cn: schema

include: file:///usr/local/etc/openldap/schema/core.ldif
include: file:///usr/local/etc/openldap/schema/cosine.ldif
include: file:///usr/local/etc/openldap/schema/inetorgperson.ldif
include: file:///usr/local/etc/openldap/schema/nis.ldif
```

Em seguida, a seção de configuração do frontend:

```
# Frontend settings
#
dn: olcDatabase={-1}frontend,cn=config
objectClass: olcDatabaseConfig
objectClass: olcFrontendConfig
olcDatabase: {-1}frontend
olcAccess: to * by * read
#
# Sample global access control policy:
#   Root DSE: allow anyone to read it
#   Subschema (sub)entry DSE: allow anyone to read it
#   Other DSEs:
#       Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
#
#olcAccess: to dn.base="" by * read
#olcAccess: to dn.base="cn=Subschema" by * read
#olcAccess: to *
#   by self write
#   by users read
#   by anonymous auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!
#
olcPasswordHash: {SSHA}
```

```
# {SSHA} is already the default for olcPasswordHash
```

Outra seção é dedicada ao *backend de configuração*, a única maneira de acessar posteriormente a configuração do servidor OpenLDAP é como um superusuário global.

```
dn: olcDatabase={0}config,cn=config
objectClass: olcDatabaseConfig
olcDatabase: {0}config
olcAccess: to * by * none
olcRootPW: {SSHA}iae+lrQZILpiUdf16Z9KmDmSwT77Dj4U
```

O nome de usuário administrador padrão é `cn=config`. Digite slappasswd em um shell, escolha a senha e use sua hash `olcRootPW`. Se essa opção não for especificada agora, antes do arquivo slapd.ldif ser importado, ninguém poderá modificar a seção de *configuração global*.

A última seção é sobre o back-end do banco de dados:

```
#######################################################################
# LMDB database definitions
#######################################################################
#
dn: olcDatabase=mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: mdb
olcDbMaxSize: 1073741824
olcSuffix: dc=domain,dc=example
olcRootDN: cn=mdbadmin,dc=domain,dc=example
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd-config(5) for details.
# Use of strong authentication encouraged.
olcRootPW: {SSHA}X2wHvIWDk6G76CQyCMS1vDCvtICWgn0+
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
olcDbDirectory: /var/db/openldap-data
# Indices to maintain
olcDbIndex: objectClass eq
```

Esse banco de dados hospeda os *conteudos atuais* do diretório LDAP. Outros tipos diferentes de `mdb` estão disponiveis. Esse é super-usuário, não confundir com um global, é configurado aqui: um usuário (possivelmente customizado) em `olcRootDN` e a hash da senha em `olcRootPW`; slappasswd pode ser usado como antes.

Esse repositorio contém quatro exemplos do arquivo slapd.ldif. Para converter um arquivo slapd.conf existente dentro de slapd.ldif, referencie a essa página (por favor, note que isso pode introduzir algumas opções inuteis).

Quando a configuração estiver concluída, o slapd.ldif deve ser colocado em um diretório vazio. Recomenda-se criá-lo como:

```
# mkdir /usr/local/etc/openldap/slapd.d/
```

Importe o banco de dados de configuração:

```
# /usr/local/sbin/slapadd -n0 -F /usr/local/etc/openldap/slapd.d/ -l
/usr/local/etc/openldap/slapd.ldif
```

Inicie o daemon slapd:

```
# /usr/local/libexec/slapd -F /usr/local/etc/openldap/slapd.d/
```

A opção `-d` pode ser usada para depuração, conforme especificado em slapd(8). Para verificar se o servidor está em execução e funcionando:

```
# ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts
#

#
dn:
namingContexts: dc=domain,dc=example

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

O servidor ainda deve ser confiável. Se isso nunca foi feito antes, siga estas instruções. Instale o pacote ou o port OpenSSL:

```
# pkg install openssl
```

No diretório onde o ca.crt está armazenado (neste exemplo, /usr/local/etc/openldap), execute:

```
# c_rehash .
```

Tanto a CA quanto o certificado do servidor agora são reconhecidos corretamente em suas respectivas funções. Para verificar isso, execute este comando no diretório server.crt:

```
# openssl verify -verbose -CApath . server.crt
```

Se o slapd estiver em execução, reinicie-o. Como declarado em /usr/local/etc/rc.d/slapd, para executar corretamente o slapd na inicialização, as seguintes linhas devem ser adicionadas ao /etc/rc.conf:

```
lapd_enable="YES"
slapd_flags='-h "ldapi://%2fvar%2frun%2fopenldap%2fldapi/
ldap://0.0.0.0/"'
slapd_sockets="/var/run/openldap/ldapi"
slapd_cn_config="YES"
```

O slapd não fornece depuração na inicialização. Verifique o /var/log/debug.log, o dmesg -a e o /var/log/messages para este propósito.

O exemplo a seguir adiciona o grupo `team` e o usuário `john` ao banco de dados LDAP de `domain.example`, que ainda está vazio. Primeiro, crie o arquivo domain.ldif:

```
# cat domain.ldif
dn: dc=domain,dc=example
objectClass: dcObject
objectClass: organization
o: domain.example
dc: domain

dn: ou=groups,dc=domain,dc=example
objectClass: top
objectClass: organizationalunit
ou: groups

dn: ou=users,dc=domain,dc=example
objectClass: top
objectClass: organizationalunit
ou: users

dn: cn=team,ou=groups,dc=domain,dc=example
objectClass: top
objectClass: posixGroup
cn: team
gidNumber: 10001
```

```
dn: uid=john,ou=users,dc=domain,dc=example
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: John McUser
uid: john
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/john/
loginShell: /usr/bin/bash
userPassword: secret
```

Veja a documentação do OpenLDAP para mais detalhes. Use slappasswd para substituir a senha `secret` em texto puro com um hash no `userPassword`. O caminho especificado como `loginShell` deve existir em todos sistemas onde `john` pode se logar. Finalmente, use o administrador `mdb` para modificar o banco de dados:

```
# ldapadd -W -D "cn=mdbadmin,dc=domain,dc=example" -f domain.ldif
```

Modificações para a seção *configurações globais* podem ser feitas apenas pelo super-usuário global. Por exemplo, assume que a opção `olcTLSCipherSuite: HIGH:MEDIUM:SSLv3` foi inicialmente especificada e deve agora ser deletada. Primeiro, crie um arquivo que contenha o seguinte:

```
# cat global_mod
dn: cn=config
changetype: modify
delete: olcTLSCipherSuite
```

Em seguida, aplique as modificações:

```
# ldapmodify -f global_mod -x -D "cn=config" -W
```

Quando solicitado, forneça a senha escolhida na seção *configuração backend*. O nome de usuário não é necessário: aqui, `cn=config` representa o DN da seção do banco de dados a ser modificada. Como alternativa, use `ldapmodify` para excluir uma única linha do banco de dados, `ldapdelete` para excluir uma entrada inteira.

Se algo der errado ou se o superusuário global não puder acessar o backend de configuração, é possível excluir e reescrever toda a configuração:

```
# rm -rf /usr/local/etc/openldap/slapd.d/
```

O slapd.ldif pode então ser editado e importado novamente. Por favor, siga este procedimento somente quando nenhuma outra solução estiver disponível.

Esta é a configuração do servidor apenas. A mesma máquina também pode hospedar um cliente LDAP, com sua própria configuração separada.

29.6. Protocolo de configuração dinâmica de hosts (DHCP)

O protocolo de configuração dinâmica de hosts (DHCP) permite que um sistema se conecte a uma rede para receber as informações de endereçamento necessárias para a comunicação nessa rede. O FreeBSD inclui a versão do `dhclient` do OpenBSD que é usada pelo cliente para obter as informações de endereçamento. O FreeBSD não instala um servidor DHCP, mas vários servidores estão disponíveis na coleção de Ports do FreeBSD. O protocolo DHCP é totalmente descrito em RFC 2131. Recursos informativos também estão disponíveis em isc.org/downloads/dhcp/.

Esta seção descreve como usar o cliente DHCP integrado. Em seguida, descreve como instalar e configurar um servidor DHCP.



No FreeBSD, o dispositivo bpf(4) é necessário tanto pelo servidor DHCP como pelo DHCP > cliente. Este dispositivo está incluído no kernel GENERIC que é instalado com o FreeBSD. Usuários que preferem criar um kernel personalizado precisam manter este dispositivo se o DHCP for usado.

Deve-se notar que o bpf também permite que usuários privilegiados executem sniffers de pacotes de rede naquele sistema.

29.6.1. Configurando um cliente DHCP

O suporte ao cliente DHCP está incluído no instalador do FreeBSD, facilitando a configuração de um sistema recém-instalado para receber automaticamente as informações de endereçamento de rede de um servidor DHCP existente. Consulte Pós-instalação para exemplos de configuração de rede.

Quando o `dhclient` é executado na máquina cliente, ele inicia as solicitações de transmissão das informações de configuração. Por padrão, esses pedidos usam a porta UDP 68. O servidor responde na porta UDP 67 , fornecendo ao cliente um endereço IP e outras informações de rede relevantes como uma máscara de sub-rede, gateway padrão e endereços de servidor DNS. Esta informação está na forma de uma "concessão" de DHCP e é válida por um tempo configurável. Isso permite que endereços IP obsoletos para clientes que não estejam mais conectados à rede sejam reutilizados automaticamente. Clientes DHCP podem obter uma grande quantidade de informações do servidor. Uma lista exaustiva pode ser encontrada em dhcp-options(5).

Por padrão, quando um sistema FreeBSD inicializa, seu cliente DHCP é executado em segundo plano, ou *asynchronously*. Outros scripts de inicialização continuam sendo executados enquanto o processo DHCP é concluído, o que acelera a inicialização do sistema.

O DHCP em segundo plano funciona bem quando o servidor DHCP responde rapidamente às solicitações do cliente. No entanto, o DHCP pode levar muito tempo para ser concluído em alguns sistemas. Se os serviços de rede tentarem executar antes que o DHCP tenha atribuído as informações de endereçamento de rede, eles falharão. O uso do DHCP no modo *synchronous* impede esse problema, pois ele pausa a inicialização até que a configuração DHCP seja concluída.

Esta linha no /etc/rc.conf é usada para configurar o modo background ou assíncrono:

```
ifconfig_fxp0="DHCP"
```

Esta linha pode já existir se o sistema foi configurado para usar o DHCP durante a instalação. Substitua o *fxp0* mostrado nesses exemplos pelo nome da interface a ser configurada dinamicamente, conforme descrito em Configurando Placas de Interface de Rede.

Para configurar o sistema para usar o modo síncrono e pausar durante a inicialização enquanto o DHCP é concluído, use "SYNCDHCP":

```
ifconfig_fxp0="SYNCDHCP"
```

Opções adicionais do cliente estão disponíveis. Procure por `dhclient` in rc.conf(5) para detalhes.

O cliente DHCP usa os seguintes arquivos:

- /etc/dhclient.conf

  O arquivo de configuração usado pelo `dhclient`. Normalmente, esse arquivo contém apenas comentários, pois os padrões são adequados para a maioria dos clientes. Este arquivo de configuração é descrito em dhclient.conf(5).

- /sbin/dhclient

  Maiores informações sobre o comando em si podem ser encontradas em dhclient(8).

- /sbin/dhclient-script

  O script de configuração do cliente DHCP específico do FreeBSD. Ele é descrito em dhclient-script(8), mas não deve precisar de nenhuma modificação do usuário para funcionar corretamente.

- /var/db/dhclient.leases.interface

  O cliente DHCP mantém um banco de dados de concessões válidas neste arquivo, que é escrito como um log e é descrito em dhclient.leases(5).

29.6.2. Instalando e configurando um servidor DHCP

Esta seção demonstra como configurar um sistema FreeBSD para atuar como um servidor DHCP usando a implementação do servidor DHCP do Internet Systems Consortium (ISC). Esta implementação e a sua documentação podem ser instaladas usando o pacote ou port net/isc-dhcp44-server.

A instalação do net/isc-dhcp44-server instala um arquivo de configuração de exemplo. Copie o /usr/local/etc/dhcpd.conf.example para /usr/local/etc/dhcpd.conf e faça as alterações neste novo arquivo.

O arquivo de configuração é composto de declarações para sub-redes e hosts que definem as informações que são fornecidas aos clientes DHCP. Por exemplo, essas linhas configuram o seguinte:

```
option domain-name "example.org";①
option domain-name-servers ns1.example.org;②
option subnet-mask 255.255.255.0;③

default-lease-time 600;④
max-lease-time 72400;⑤
ddns-update-style none;⑥

subnet 10.254.239.0 netmask 255.255.255.224 {
  range 10.254.239.10 10.254.239.20;⑦
  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;⑧
}

host fantasia {
  hardware ethernet 08:00:07:26:c0:a5;⑨
  fixed-address fantasia.fugue.com;⑩
}
```

① Esta opção especifica o domínio de pesquisa padrão que será fornecido aos clientes. Consulte resolv.conf(5) para obter maiores informações.

② Esta opção especifica uma lista separada por vírgula de servidores DNS que o cliente deve usar. Eles podem ser listados por seus nomes de domínio totalmente qualificados (FQDN), como visto no exemplo, ou por seus endereços de IP.

③ A máscara de sub-rede que será fornecida aos clientes.

④ O tempo de expiração da concessão padrão em segundos. Um cliente pode ser configurado para substituir esse valor.

⑤ O período máximo permitido de tempo, em segundos, para uma concessão. Se um cliente solicitar uma concessão mais longa, uma concessão ainda será emitida, mas será válida apenas para o tempo especificado em `max-lease-time`.

⑥ O padrão `none` desabilita as atualizações de DNS dinâmicas. Alterar isso para `interim` configura o servidor DHCP para atualizar um servidor DNS sempre que for concedido um contrato para que o servidor de DNS saiba quais endereços de IP estão associados a quais computadores na rede. Não altere a configuração padrão, a menos que o servidor de DNS tenha sido configurado para suportar DNS dinâmico.

⑦ Esta linha cria um conjunto de endereços IP disponíveis que são reservados para alocação a clientes DHCP. O intervalo de endereços deve ser válido para a rede ou sub-rede especificada na linha anterior.

⑧ Declara o gateway padrão que é válido para a rede ou sub-rede especificada antes do colchete de abertura `{`.

⑨ Especifica o endereço de hardware MAC de um cliente para que o servidor DHCP possa reconhecer o cliente quando ele fizer uma solicitação.

⑩ Especifica que este host deve sempre receber o mesmo endereço IP. A utilização do nome do host está correta, pois o servidor DHCP resolverá o nome do host antes de retornar as informações de concessão.

Este arquivo de configuração suporta muito mais opções. Consulte o dhcpd.conf(5), instalado com o servidor, para obter detalhes e exemplos.

Uma vez que a configuração do dhcpd.conf estiver completa, habilite o servidor DHCP em /etc/rc.conf:

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Substitua o `dc0` pela interface (ou interfaces, separadas por espaço em branco) que o servidor DHCP deverá escutar por solicitações de clientes DHCP.

Inicie o servidor executando o seguinte comando:

```
# service isc-dhcpd start
```

Quaisquer mudanças futuras na configuração do servidor exigirão que o serviço dhcpd seja interrompido e, em seguida, iniciado usando service(8).

O servidor DHCP usa os seguintes arquivos. Observe que as páginas de manual são instaladas com o software do servidor.

- /usr/local/sbin/dhcpd

  Maiores informações sobre o servidor dhcpd podem ser encontradas em dhcpd(8).

- /usr/local/etc/dhcpd.conf

  O arquivo de configuração do servidor precisa conter todas as informações que devem ser fornecidas aos clientes, juntamente com informações sobre a operação do servidor. Este arquivo de configuração é descrito no dhcpd.conf(5).

- /var/db/dhcpd.leases

  O servidor DHCP mantém um banco de dados das concessões que ele emitiu neste arquivo, que é gravado como um log. Consulte dhcpd.leases(5), o qual fornece uma descrição um pouco mais longa.

- /usr/local/sbin/dhcrelay

  Esse daemon é usado em ambientes avançados, onde um servidor DHCP encaminha uma solicitação de um cliente para outro servidor DHCP em uma rede separada. Se esta funcionalidade for necessária, instale o pacote ou port net/isc-dhcp44-relay. A instalação inclui o dhcrelay(8), que fornece maiores detalhes.

29.7. Sistema de Nomes de Domínio (DNS)

O Sistema de Nomes de Domínio (DNS) é o protocolo através do qual os nomes de domínio são mapeados para endereços de IP e vice-versa. O DNS é coordenado pela Internet através de um sistema complexo de raiz de autoridade, Top Level Domain (TLD) e outros servidores de nomes de menor escala, que hospedam e armazenam em cache domínios individuais. Não é necessário executar um servidor de nomes para executar pesquisas de DNS em um sistema.

A tabela a seguir descreve alguns dos termos associados ao DNS:

Tabela 26. Terminologia DNS

Termo	Definição
Encaminhamento de DNS	Mapeamento de nomes de hosts para endereços de IP.
Origem	Refere-se ao domínio coberto em um arquivo de zona específico.
Resolver	Um processo do sistema através do qual uma máquina consulta um servidor de nomes para informações de zona.
DNS Reverso	Mapeamento de endereços IP para hostnames.
Root zone	O início da hierarquia da zona da Internet. Todas as zonas se enquadram na zona de raiz, semelhante a como todos os arquivos em um sistema de arquivos se enquadram no diretório raiz.
Zona	Um domínio individual, subdomínio ou parte do DNS administrado pela mesma autoridade.

Exemplos de zonas:

- `.` é como a zona root é geralmente referida na documentação.
- `org.` é um domínio de nível superior (TLD) sob a zona raiz.
- `example.org.` é uma zona sob o TLD `org.`.
- `1.168.192.in-addr.arpa` é uma zona que faz referência a todos os endereços IP que se enquadram no espaço de endereçamento IP `192.168.1.*` .

Como se pode ver, a parte mais específica de um nome de host aparece à esquerda. Por exemplo, `example.org.` é mais específico que `org.`, como `org.` é mais específico que a zona raiz . O layout de cada parte de um nome de host é muito parecido com um sistema de arquivos: o diretório /dev está dentro da raiz e assim por diante.

29.7.1. Razões para executar um servidor de nomes

Os servidores de nomes geralmente vêm em duas formas: servidores de nomes autoritativos e servidores de nomes de armazenamento em cache (também conhecidos como servidores de

resolução).

Um servidor de nomes autoritativo é necessário quando:

- Alguém quer servir ao mundo informações de DNS, respondendo autoritariamente a consultas.
- Um domínio, como `example.org`, está registrado e os endereços IP precisam ser atribuídos a nomes de host sob ele.
- Um bloco de endereços IP requer entradas reversas de DNS (IP para hostname).
- Um servidor de nomes de backup ou secundário, chamado de escravo, responderá às consultas.

Um servidor de nomes em cache é necessário quando:

- Um servidor DNS local pode armazenar em cache e responder mais rapidamente do que consultar um servidor de nomes externo.

Quando alguém pergunta por `www.FreeBSD.org`, o resolvedor geralmente consulta o servidor de nomes do ISP e recupera a resposta. Com um servidor local, de cache DNS, a consulta só precisa ser feita uma vez para o mundo externo pelo servidor de Cache DNS. Consultas adicionais não precisarão sair da rede local, pois as informações estão armazenadas em um cache local.

29.7.2. Configuração do servidor de DNS

O Unbound é fornecido no sistema básico do FreeBSD. Por padrão, ele fornecerá a resolução de DNS apenas para a máquina local. Embora o pacote básico do sistema possa ser configurado para fornecer serviços de resolução além da máquina local, é recomendável que esses requisitos sejam resolvidos instalando o Unbound da coleção de ports do FreeBSD.

Para ativar o Unbound, adicione o seguinte ao /etc/rc.conf:

```
local_unbound_enable="YES"
```

Quaisquer servidores de nomes existentes em /etc/resolv.conf serão configurados como forwarders na nova configuração do Unbound.



Se algum dos servidores de nomes listados não suportar o DNSSEC, a resolução local DNS falhará. Certifique-se de testar cada servidor de nomes e remover qualquer um que falhe no teste. O seguinte comando mostrará a árvore de confiança ou uma falha para um servidor de nomes em execução em `192.168.1.1`:

```
% drill -S FreeBSD.org @192.168.1.1
```

Quando cada servidor de nomes for confirmado para suportar DNSSEC, inicie o Unbound:

```
# service local_unbound onestart
```

Isso cuidará da atualização do arquivo /etc/resolv.conf para que as consultas para domínios seguros DNSSEC funcionem agora. Por exemplo, execute o seguinte DNSSEC para validar a árvore confiável do FreeBSD.org :

```
% drill -S FreeBSD.org
;; Number of trusted keys: 1
;; Chasing: freebsd.org. A

DNSSEC Trust tree:
freebsd.org. (A)
|---freebsd.org. (DNSKEY keytag: 36786 alg: 8 flags: 256)
    |---freebsd.org. (DNSKEY keytag: 32659 alg: 8 flags: 257)
    |---freebsd.org. (DS keytag: 32659 digest type: 2)
        |---org. (DNSKEY keytag: 49587 alg: 7 flags: 256)
            |---org. (DNSKEY keytag: 9795 alg: 7 flags: 257)
            |---org. (DNSKEY keytag: 21366 alg: 7 flags: 257)
            |---org. (DS keytag: 21366 digest type: 1)
            |   |---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
            |       |---. (DNSKEY keytag: 19036 alg: 8 flags: 257)
            |---org. (DS keytag: 21366 digest type: 2)
                |---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
                    |---. (DNSKEY keytag: 19036 alg: 8 flags: 257)
;; Chase successful
```

29.8. Servidor HTTP Apache

O open source Apache HTTP Server é o servidor Web mais utilizado. O FreeBSD não instala este servidor web por padrão, mas ele pode ser instalado a partir do pacote ou Port www/apache24.

Esta seção resume como configurar e iniciar a versão 2.*x* do Servidor HTTP Apache no FreeBSD. Para informações mais detalhadas sobre o Apache2.X e suas diretivas de configuração, consulte httpd.apache.org.

29.8.1. Configurando e Iniciando o Apache

No FreeBSD, o arquivo de configuração principal do Apache HTTP Server é instalado como /usr/local/etc/apache2x/httpd.conf, onde *x* representa o número da versão. Este arquivo ASCII de texto inicia as linhas de comentário com um #. As diretivas modificadas com mais freqüência são:

`ServerRoot "/usr/local"`

Especifica a hierarquia de diretório padrão para a instalação do Apache. Os binários são armazenados nos subdiretórios bin e sbin da raiz do servidor e os arquivos de configuração são armazenados no subdiretório etc/apache2x.

`ServerAdmin you@example.com`

Altere isso para seu endereço de e-mail para receber problemas com o servidor. Esse endereço também aparece em algumas páginas geradas pelo servidor, como documentos de erro.

`ServerName www.example.com:80`

Permite que um administrador defina um nome de host que é enviado de volta aos clientes pelo servidor. Por exemplo, `www` pode ser usado em vez do nome do host real. Se o sistema não tiver um nome registrado no DNS, insira seu endereço IP. Se o servidor irá escutar em um relatório alternativo, altere a porta `80` para o número de porta alternativa.

`DocumentRoot "/usr/local/www/apache2x/data"`

O diretório no qual os documentos serão exibidos. Por padrão, todas as solicitações são obtidas desse diretório, mas os links e aliases simbólicos podem ser usados para apontar para outros locais.

É sempre uma boa ideia fazer uma cópia de backup do arquivo de configuração do Apache padrão antes de fazer alterações. Quando a configuração do Apache estiver concluída, salve o arquivo e verifique a configuração usando o `apachectl`. A execução do `apachectl configtest` deve retornar `Syntax OK`.

Para iniciar o Apache na inicialização do sistema, adicione a seguinte linha ao /etc/rc.conf:

```
apache24_enable="YES"
```

Se o Apache deve ser iniciado com opções não-padrão, a seguinte linha pode ser adicionada ao /etc/rc.conf para especificar os flags necessários:

```
apache24_flags=""
```

Se o apachectl não relatar erros de configuração, inicie o `httpd` agora:

```
# service apache24 start
```

O serviço `httpd` pode ser testado inserindo `http://localhost` em um navegador da Web, substituindo *localhost* pelo nome de domínio totalmente qualificado da máquina que está executando o `httpd`. A página padrão da Web exibida é /usr/local/www/apache24/data/index.html.

A configuração do Apache pode ser testada quanto a erros depois de fazer alterações subsequentes de configuração enquanto o `httpd` está em execução usando o seguinte comando:

```
# service apache24 configtest
```



É importante notar que o `configtest` não é um padrão rc(8) e não se espera que funcione para todos os scripts de inicialização.

29.8.2. Hospedagem Virtual

A hospedagem virtual permite que vários sites sejam executados em um servidor Apache. Os hosts

virtuais podem ser *baseados em IP* ou *baseados em nome*. A hospedagem virtual baseada em IP usa um endereço IP diferente para cada site. A hospedagem virtual baseada em nome usa os cabeçalhos HTTP/1.1 do cliente para descobrir o nome do host, o que permite que os sites compartilhem o mesmo endereço de IP.

Para configurar o Apache para usar hospedagem virtual baseada em nome, adicione um bloco `VirtualHost` para cada site. Por exemplo, para o servidor Web denominado `www.domain.tld` com um domínio virtual de `www.someotherdomain.tld`, adicione as seguintes entradas ao arquivo httpd.conf:

```
<VirtualHost *>
    ServerName www.domain.tld
    DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
    ServerName www.someotherdomain.tld
    DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Para cada host virtual, substitua os valores de `ServerName` e `DocumentRoot` pelos valores a serem usados.

Para obter mais informações sobre como configurar hosts virtuais, consulte a documentação oficial do Apache em: http://httpd.apache.org/docs/vhosts/.

29.8.3. Módulos Apache

O Apache usa módulos para aumentar a funcionalidade fornecida pelo servidor básico. Consulte o http://httpd.apache.org/docs/current/mod/ para uma lista completa e detalhes de configuração para os módulos disponíveis.

No FreeBSD, alguns módulos podem ser compilados com o port www/apache24. Digite `make config` dentro do diretório /usr/ports/www/apache24 para ver quais módulos estão disponíveis e quais estão ativados por padrão. Se o módulo não é compilado com o port, a Coleção de Ports do FreeBSD fornece uma maneira fácil de instalar vários módulos. Esta seção descreve três dos módulos mais usados.

29.8.3.1. Suporte SSL

Em algum momento, o suporte para o SSL dentro do Apache requer um modulo secundário chamado mod_ssl. Esse não é mais o casoe a instalação padrão do Apache vem com SSL embutido no servidor web. Um exemplo de como habilitar o suporte para paginas com SSL está disponível no arquivo http-ssl.conf instalado dentro do diretório /usr/local/etc/apache24/extra. Dentro desse diretório também esta um exemplo do arquivo chamado ssl.conf-sample. É recomendado que ambos arquivos sejam avaliados para configurar apropriadamente páginas seguras no servidor web Apache.

Depois da configuração do SSL estiver completa, deve ser removido o comentário da linha seguinte no arquivo http.conf principal para ativar as mudanças no próximo restart ou reload do Apache:

```
#Include etc/apache24/extra/httpd-ssl.conf
```



Versão dois do SSL e a versão três tem problemas de vulnerabilidades conhecidas. É altamente recomendado a versão 1.2 do TLS e 1.3 deve ser habilitada no lugar das velhas opções do SSL. Isso pode ser realizado configurando as seguintes opções no arquivo ssl.conf:

```
SSLProtocol all -SSLv3 -SSLv2 +TLSv1.2 +TLSv1.3
SSLProxyProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
```

Para completar a configuração do SSL no servidor web, remova os comentários da seguinte linha para garantir que a configuração irá ser enviada para dentro do Apache durante o restart ou reload:

```
# Secure (SSL/TLS) connections
Include etc/apache24/extra/httpd-ssl.conf
```

As linhas a seguir também devem ser descomentadas no httpd.conf para suportar totalmente o SSL no Apache:

```
LoadModule authn_socache_module libexec/apache24/mod_authn_socache.so
LoadModule socache_shmcb_module libexec/apache24/mod_socache_shmcb.so
LoadModule ssl_module libexec/apache24/mod_ssl.so
```

O próximo passo é trabalhar com uma autoridade certificadora para ter certificados apropriados instalados no sistema. Isso vai configurar um cadeia de confiança para a pagina e prever alguns avisos de certificados auto assinados.

29.8.3.2. mod_perl

O módulo mod_perl torna possível escrever módulos Apache em Perl. Além disso, o intérprete persistente embutido no servidor evita a sobrecarga de iniciar um intérprete externo e a penalidade do tempo de inicialização do Perl.

O mod_perl pode ser instalado usando o pacote ou port www/mod_perl2. A documentação para usar este módulo pode ser encontrada em http://perl.apache.org/docs/2.0/index .html.

29.8.3.3. mod_php

PHP: Pré-processador de hipertexto (PHP) é uma linguagem de script de propósito geral que é especialmente adequada para desenvolvimento web. Capaz de ser incorporada em HTML, sua sintaxe se baseia em C, Java™ e Perl com a intenção de permitir desenvolvedores web para escrever rapidamente páginas da web geradas dinamicamente.

Suporte para PHP para o Apache e alguma outra parte escrita na linguagem, pode ser adicionada

instalando o port apropriado.

Para todas versões suportadas, procure os dados do pacote usando o comando `pkg`:

```
# pkg search php
```

Uma lista vai ser disponibilizada incluindo as versões e partes adicionais que elas proverem. Os componentes são completamente modulares, significando que as partes especificas são habilitadas instalando o port apropriado. Para instalar o PHP na versão 7.4 para o Apache, use o seguinte comando:

```
# pkg install mod_php74
```

Se algum pacote dependente precisar ser instalado, ele irá ser instalado também.

Por padrão, o PHP não estará habilitado. As seguintes linhas precisam ser adicionadas no arquivo de configuração do Apache localizado em /usr/local/etc/apache24 para ativa-lo:

```
<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
```

Em adição, a opção `DirectoryIndex` no arquivo de configuração irá precisar ser atualizada também e o Apache irá precisar ser reiniciado ou feito um relaoad também para as mudanças surtirem efeito.

Suporte para muitas partes do PHP podem ser instalado também usando o comando `pkg`. Por exemplo, para instalar suporte para o XML ou para SSL, instale os seguintes ports:

```
# pkg install php74-xml php74-openssl
```

Como antes, a configuração do Apache irá precisar ser recarregada para as mudanças surtirem efeito, mesmo em casos onde foi feita apenas a instalação de um modulo.

Para realizar uma reinicialização normal para recarregar a configuração, digite o seguinte comando:

```
# apachectl graceful
```

Uma vez que a instalação esteja completa, há dois métodos para obter o suporte para os modulos do PHP e a informação do ambiente dessa instalação. A primeira é instalar o binário completo do PHP e rodar o seguinte comando para obter a informação:

```
# pkg install php74
```

```
# php -i |less
```

Isso é necessário para passar a saída paga um paginador, como o comando `more` ou `less` para visualizar melhor a saída.

Finalmente, para fazer alguma mudança na configuração global do PHP há um arquivo bem documentado instalado dentro de /usr/local/etc/php.ini. No momento da instalação, esse arquivo não irá existir porque há duas versões para escolher, uma é o arquivo php.ini-development e outra o php.ini-production. Esses são pontos iniciais para ajudar os administradores na implementação.

29.8.3.4. Suporte a HTTP2

Suporte do Apache para o protocolo HTTP está incluido por padrão quando instala o port com o comando `pkg`. A nova versão do HTTP inclui muitas melhorias em relação a versão anterior, incluindo utilizar uma conexão singular para uma página, reduzindo as idas e vindas de conexões TCP. Também, os dados no cabeçalho do pacote é comprimido e o HTTP2 requer encriptação por padrão.

Quando o Apache estiver configurado para usar HTTP2 apenas, os navegadores web irão requisitar conexões seguras, encriptadas com HTTPS. Quando o Apache estiver configurado para usar ambas versões, o HTTP1.1 irá ser considerado uma opção substituta se algum problema surgir durante a conexão.

Embora essa mudança exija que os administradores façam alterações, elas são positivas e equivalem a uma Internet mais segura para todos. As mudanças são requeridas apenas para paginas não implementada corretamente com SSL e TLS.



Essa configuração depende das seções anteriores, incluindo suporte a TLS. É recomendado que essas instruções seja seguidas antes de continuar com essa configuração.

Comece o processo habilitando o modulo http2 removendo o comentário da linha no arquivo /usr/local/etc/apache24/httpd.conf e trocando o modulo mpm_prefork pelo mpm_event pois o anterior não suporta o http2.

```
LoadModule http2_module libexec/apache24/mod_http2.so
LoadModule mpm_event_module libexec/apache24/mod_mpm_event.so
```



Aqui há um port mod_http1 distinto que está disponível. Ele existe pra entregar segurança e correção de bugs mais rápido que o modulo instalado por padrão com o port apache24. Ele não é requisitado para o suporte do HTTP2 mas está disponível. Quando instalado, o mod_h2.so deve ser usado no lugar do mod_http2.so na configuração do Apache.

Aqui há dois métodos para implementar o HTTP2 no Apache; um caminho é de forma global para todos os sites e cada VirtualHost rodando no sistema. Para habilitar o HTTP2 globalmente, adicione a seguinte linha abaixo da diretiva ServerName:

```
Protocolos h2 http/1.1
```



Para habilitar HTTP2 sobre texto simples, use h2h2chttp/1.1 no arquivo httpd.conf.

Tendo o h2c aqui irá permitir que o dado em texto simples do HTTP2 passar pelo sistema mas isso não é recomendado. Em adição a isso, usando o http/1.1 aqui irá permitir retornar para a versão do protocolo HTTP1.1 caso sejá necessário pelo sistema.

Para habilitar HTTP2 para VirtualHosts individuais, adicione a mesma linha com a diretiva VirtualHost no arquivo httpd.conf ou httpd-ssl.conf.

Recarregue a configuração usando o comando `apachectl` reload e teste a configuração seguindo um dos métodos após visitar uma das paginas hosteadas:

```
# grep "HTTP/2.0" /var/log/httpd-access.log
```

A saída deve ser semelhante à seguinte:

```
192.168.1.205 - - [18/Oct/2020:18:34:36 -0400] "GET / HTTP/2.0" 304 -
192.0.2.205 - - [18/Oct/2020:19:19:57 -0400] "GET / HTTP/2.0" 304 -
192.0.0.205 - - [18/Oct/2020:19:20:52 -0400] "GET / HTTP/2.0" 304 -
192.0.2.205 - - [18/Oct/2020:19:23:10 -0400] "GET / HTTP/2.0" 304 -
```

O outro metodo é usar o navegador web padrão no debugger do site ou o comando `tcpdump`; contanto, o uso de qualquer método está além do escopo desse documento.

Suporte para conexões do proxy reverso HTTP2 usando o modulo mod_proxy_http2.so. Quando declarado na configuração o ProxyPass ou RewriteRules [P], eles devem usar h2:// para a conexão.

29.8.4. Websites Dinâmicos

Além do mod_perl e do mod_php, outras linguagens estão disponíveis para a criação de conteúdo dinâmico da web. Estes incluem o Django e o Ruby on Rails.

29.8.4.1. Django

O Django é um framework de licença BSD projetado para permitir que desenvolvedores escrevam aplicações web elegantes e de alto desempenho rapidamente. Ele fornece um mapeador relacional de objeto para que os tipos de dados sejam desenvolvidos como objetos Python. Uma API rica e dinâmica de acesso ao banco de dados é fornecida para os objetos sem que o desenvolvedor tenha que escrever SQL. Ele também fornece um sistema de template extensível para que a lógica do aplicativo seja separada da apresentação HTML.

Django depende de mod_python, e um mecanismo de banco de dados SQL. No FreeBSD, o port www/py-django instala automaticamente o mod_python e suporta os banco de dados PostgreSQL, MySQL, ou SQLite, com o padrão sendo o SQLite. Para trocar o mecanismo de banco de dados, digite `make config` dentro do diretório /usr/ports/www/py-django, então instale o port.

Uma vez instalado o Django, a aplicação precisará de um diretório de projeto junto com a configuração Apache para usar o interpretador Python incorporado. Este intérprete é usado para chamar o aplicativo para URLs específicas no site.

Para configurar o Apache para que passe a fazer solicitações para determinadas URLs para a aplicação Web, adicione o seguinte ao httpd.conf, especificando o caminho completo para o diretório do projeto:

```
<Location "/">
    SetHandler python-program
    PythonPath "['/dir/to/the/django/packages/'] + sys.path"
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE mysite.settings
    PythonAutoReload On
    PythonDebug On
</Location>
```

Consulte https://docs.djangoproject.com para maiores informações sobre como usar o Django.

29.8.4.2. Ruby on Rails

O Ruby on Rails é outro framework de software livre da Web que fornece uma stack de desenvolvimento completa. Ele é otimizado para tornar os desenvolvedores da Web mais produtivos e capazes de criar rapidamente aplicativos poderosos. No FreeBSD, ele pode ser instalado usando o pacote ou port www/rubygem-rails.

Consulte http://guides.rubyonrails.org para maiores informações sobre como usar o Ruby on Rails .

29.9. Protocolo de Transferência de Arquivos (FTP)

O Protocolo de Transferência de Arquivos (FTP) fornece aos usuários uma maneira simples de transferir arquivos para um servidor FTP. O FreeBSD inclui o software do servidor FTP, ftpd, no sistema base.

O FreeBSD fornece vários arquivos de configuração para controlar o acesso ao servidor FTP. Esta seção resume esses arquivos. Consulte ftpd(8) para obter mais detalhes sobre o servidor FTP incorporado.

29.9.1. Configuração

A etapa de configuração mais importante é decidir quais contas terão permissão para acessar o servidor FTP. Um sistema FreeBSD possui várias contas do sistema que não devem ter acesso ao FTP. A lista de usuários que não permitem acesso FTP pode ser encontrada em /etc/ftpusers. Por

padrão, inclui contas do sistema. Usuários adicionais que não devem ter acesso a FTP podem ser adicionados.

Em alguns casos, pode ser desejável restringir o acesso de alguns usuários sem impedi-los completamente de usar o FTP. Isso pode ser feito criando /etc/ftpchroot como descrito em ftpchroot(5). Este arquivo lista usuários e grupos sujeitos a restrições de acesso a FTP.

Para permitir acesso anônimo ao servidor FTP, crie um usuário chamado `ftp` no sistema FreeBSD. Os usuários poderão então fazer logon no servidor FTP com um nome de usuário `ftp` ou `anonymous` . Quando for solicitada a senha, qualquer entrada será aceita, mas por convenção, um endereço de e-mail deverá ser usado como a senha. O servidor FTP chamará chroot(2) quando um usuário anônimo efetuar login para restringir o acesso somente ao diretório home do usuário `ftp`.

Existem dois arquivos de texto que podem ser criados para especificar mensagens de boas-vindas a serem exibidas para clientes FTP. O conteúdo de /etc/ftpwelcome será exibido aos usuários antes que eles atinjam o prompt de login. Após um login bem sucedido, o conteúdo de /etc/ftpmotd será exibido. Observe que o caminho para esse arquivo é relativo ao ambiente de login, portanto, o conteúdo de ~ftp/etc/ftpmotd seria exibido para usuários anônimos.

Uma vez configurado o servidor FTP, defina a variável apropriada em /etc/rc.conf para iniciar o serviço durante a inicialização:

```
ftpd_enable="YES"
```

Para iniciar o serviço agora:

```
# service ftpd start
```

Teste a conexão com o servidor FTP digitando:

```
% ftp localhost
```

O daemon ftpd usa o syslog(3) para registrar mensagens. Por padrão, o daemon de log do sistema gravará mensagens relacionadas a FTP em /var/log/xferlog. A localização do log do FTP pode ser modificada alterando a seguinte linha no /etc/syslog.conf:

```
ftp.info      /var/log/xferlog
```



Esteja ciente dos possíveis problemas envolvidos na execução de um servidor FTP anônimo. Em particular, pense duas vezes antes de permitir que usuários anônimos façam upload de arquivos. Pode acontecer que o site FTP se torne um fórum para o comércio de software comercial não licenciado ou pior. Se uploads anônimos de FTP forem necessários, verifique as permissões para que esses arquivos não possam ser lidos por outros usuários anônimos até que sejam revisados por um administrador.

29.10. Serviços de arquivos e impressão para clientes Microsoft™Windows™ Clients (Samba)

Samba é um popular pacote de software de código aberto que fornece serviços de arquivo e impressão usando o protocolo SMB/CIFS. Este protocolo está incorporado nos sistemas Microsoft™Windows™. Ele pode ser adicionado a sistemas não Microsoft™Windows™ instalando as bibliotecas-cliente Samba. O protocolo permite que os clientes acessem dados e impressoras compartilhadas. Esses compartilhamentos podem ser mapeados como uma unidade de disco local e as impressoras compartilhadas podem ser usadas como se fossem impressoras locais.

No FreeBSD, as bibliotecas cliente do Samba podem ser instaladas usando o port ou pacote net/samba410. O cliente fornece a capacidade de um sistema FreeBSD acessar compartilhamentos de SMB/CIFS em uma rede Microsoft™Windows™.

Um sistema FreeBSD também pode ser configurado para atuar como um servidor Samba instalando o port ou pacote net/samba410. Isso permite que o administrador crie compartilhamentos de SMB/CIFS no sistema FreeBSD que podem ser acessados por clientes executando Microsoft™Windows™ ou as bibliotecas do cliente Samba.

29.10.1. Configuração do Servidor

O Samba é configurado em /usr/local/etc/smb4.conf. Este arquivo deve ser criado antes que o Samba possa ser usado.

Um simples smb4.conf para compartilhar diretórios e impressoras com clientes Windows™ em um grupo de trabalho é mostrado aqui. Para configurações mais complexas envolvendo LDAP ou Active Directory, é mais fácil usar o samba-tool(8) para criar o smb4.conf.

```
[global]
workgroup = WORKGROUP
server string = Samba Server Version %v
netbios name = ExampleMachine
wins support = Yes
security = user
passdb backend = tdbsam

# Example: share /usr/src accessible only to 'developer' user
[src]
path = /usr/src
valid users = developer
writable  = yes
browsable = yes
read only = no
guest ok = no
public = no
create mask = 0666
directory mask = 0755
```

29.10.1.1. Configurações Globais

As configurações que descrevem a rede são adicionadas em /usr/local/etc/smb4.conf:

`workgroup`

O nome do grupo de trabalho a ser servido.

`netbios name`

O nome NetBIOS pelo qual um servidor Samba é conhecido. Por padrão, é o mesmo que o primeiro componente do nome do DNS do host.

`server string`

A string que será exibida na saída de `net view` e algumas outras ferramentas de rede que buscam exibir texto descritivo sobre o servidor.

`wins support`

Se o Samba funcionará como um servidor WINS. Não habilite o suporte para WINS em mais de um servidor na rede.

29.10.1.2. Configurações de Segurança

As configurações mais importantes em /usr/local/etc/smb4.conf são o modelo de segurança e o formato de senha de backend. Essas diretivas controlam as opções:

`security`

As configurações mais comuns são `security=share` e `security=user`. Se os clientes usarem nomes de usuários que sejam os mesmos nomes de usuários na máquina do FreeBSD, a segurança no nível do usuário deve ser usada. Essa é a política de segurança padrão e exige que os clientes façam logon pela primeira vez antes de poderem acessar recursos compartilhados.

Na segurança em nível de compartilhamento, os clientes não precisam efetuar logon no servidor com um nome de usuário e senha válidos antes de tentar se conectar a um recurso compartilhado. Este era o modelo de segurança padrão para versões mais antigas do Samba.

`passdb backend`

O Samba possui vários modelos de autenticação de backend diferentes. Os clientes podem ser autenticados com LDAP, NIS+, um banco de dados SQL ou um arquivo de senha modificado. O método de autenticação recomendado, `tdbsam`, é ideal para redes simples e é abordado aqui. Para redes maiores ou mais complexas, o `ldapsam` é recomendado. `smbpasswd` foi o padrão anterior e agora está obsoleto.

29.10.1.3. Usuários do Samba

As contas de usuário do FreeBSD devem ser mapeadas para o banco de dados `SambaSAMAccount` para que os clientes Windows™ acessem o compartilhamento. Mapear contas de usuários existentes do FreeBSD usando pdbedit(8):

```
# pdbedit -a username
```

Esta seção mencionou apenas as configurações mais usadas. Consulte a Wiki Oficial do Samba para obter informações adicionais sobre as opções de configuração disponíveis.

29.10.2. Iniciando o Samba

Para habilitar o Samba no momento da inicialização, adicione a seguinte linha ao /etc/rc.conf:

```
samba_server_enable="YES"
```

Para iniciar o Samba agora:

```
# service samba_server start
Performing sanity check on Samba configuration: OK
Starting nmbd.
Starting smbd.
```

O Samba consiste em três daemons separados. Os daemons nmbd e smbd são iniciados por `samba_enable`. Se resolução de nomes winbind também é necessária, defina:

```
winbindd_enable="YES"
```

O Samba pode ser interrompido a qualquer momento digitando:

```
# service samba_server stop
```

O Samba é um conjunto de software complexo com funcionalidade que permite ampla integração com as redes Microsoft™Windows™. Para mais informações sobre a funcionalidade além da configuração básica descrita aqui, consulte https://www.samba.org.

29.11. Sincronização de Relógio com NTP

Com o tempo, o relógio de um computador está propenso a se desviar. Isso é problemático, pois muitos serviços de rede exigem que os computadores em uma rede compartilhem o mesmo tempo exato. Tempo preciso também é necessário para garantir que os registros de data e hora dos arquivos permaneçam consistentes. O protocolo de horário da rede (NTP) é uma maneira de fornecer precisão de relógio em uma rede.

O FreeBSD inclui o ntpd(8) o qual pode ser configurado para consultar outros servidores NTP para sincronizar o relógio nessa máquina ou para fornecer serviços de horário para outros computadores na rede.

Esta seção descreve como configurar o ntpd no FreeBSD. Mais documentação pode ser encontrada em /usr/shared/doc/ntp/ no formato HTML.

29.11.1. Configuração de NTP

No FreeBSD, o ntpd nativo pode ser usado para sincronizar o relógio do sistema. O Ntpd é configurado usando variáveis no rc.conf(5) e no /etc/ntp.conf, conforme detalhado nas seções a seguir.

O Ntpd se comunica com seus network peers usando pacotes UDP. Quaisquer firewalls entre sua máquina e seus NTP peers devem ser configurados para permitir a entrada e saída de pacotes UDP na porta 123.

29.11.1.1. O arquivo /etc/ntp.conf

O Ntpd faz a leitura do /etc/ntp.conf para determinar quais servidores NTP que ele deve consultar. É recomendável escolher vários servidores NTP, caso um dos servidores se torne inacessível ou seu relógio torne-se não confiável. Como o ntpd recebe respostas, ele favorece servidores confiáveis em vez dos menos confiáveis. Os servidores consultados podem ser locais na rede, fornecidos por um ISP ou selecionados a partir de uma lista online de servidores NTP publicamente acessíveis. Ao escolher um servidor NTP público, selecione um servidor geograficamente próximo e revise sua política de uso. A palavra-chave `pool` de configuração seleciona um ou mais servidores de um pool de servidores. Está disponível uma lista online de pools NTP publicamente acessíveis, organizada por área geográfica. Além disso, o FreeBSD fornece um pool patrocinado pelo projeto, `0.freebsd.pool.ntp.org`.

Exemplo 47. Exemplo de /etc/ntp.conf

Este é um exemplo simples de um arquivo ntp.conf. Ele pode ser usado com segurança como está; ele contém as opções `restrict` recomendadas para operação em uma conexão de rede pública.

```
# Disallow ntpq control/query access.  Allow peers to be added only
# based on pool and server statements in this file.
restrict default limited kod nomodify notrap noquery nopeer
restrict source  limited kod nomodify notrap noquery

# Allow unrestricted access from localhost for queries and control.
restrict 127.0.0.1
restrict ::1

# Add a specific server.
server ntplocal.example.com iburst

# Add FreeBSD pool servers until 3-6 good servers are available.
tos minclock 3 maxclock 6
pool 0.freebsd.pool.ntp.org iburst

# Use a local leap-seconds file.
leapfile "/var/db/ntpd.leap-seconds.list"
```

O formato deste arquivo é descrito em ntp.conf(5). As descrições abaixo fornecem uma visão geral rápida apenas das palavras-chave usadas no arquivo de exemplo acima.

Por padrão, um servidor NTP pode ser acessado de qualquer host da rede. A palavra-chave `restrict` controla quais sistemas podem acessar o servidor. Múltiplas entradas `restrict` são suportadas, cada uma refinando as restrições fornecidas nas instruções anteriores. Os valores mostrados no exemplo concedem ao sistema local o acesso completo à consulta e controle, enquanto permitem aos sistemas remotos apenas a capacidade de consultar o horário. Para obter mais detalhes, consulte a subseção `Access Control Support` de ntp.conf(5).

A palavra-chave `server` especifica um único servidor para consulta. O arquivo pode conter várias palavras-chave server, com um servidor listado em cada linha. A palavra-chave `pool` especifica um pool de servidores. O Ntpd adicionará um ou mais servidores desse pool, conforme necessário, para atingir o número de peers especificado usando o valor `tos minclock`. A palavra-chave `iburst` direciona o ntpd para executar um burst de oito trocas rápidas de pacotes com um servidor quando o contato é estabelecido pela primeira vez, para ajudar a sincronizar rapidamente a hora do sistema.

A palavra-chave `leapfile` especifica o local de um arquivo que contém informações sobre segundos bissextos. O arquivo é atualizado automaticamente pelo periodic(8). O local do arquivo especificado por esta palavra-chave deve corresponder ao local definido na variável `ntp_db_leapfile` em /etc/rc.conf.

29.11.1.2. Entradas NTP no /etc/rc.conf

Defina `ntpd_enable=YES` para iniciar o ntpd no momento do boot do sistema. Depois que o `ntpd_enable=YES` for adicionado ao /etc/rc.conf, o ntpd poderá ser iniciado imediatamente sem reiniciar o sistema, digitando:

```
# service ntpd start
```

Somente `ntpd_enable` deve ser configurado para usar o ntpd. As variáveis rc.conf listadas abaixo também podem ser definidas conforme necessário.

Defina `ntpd_sync_on_start=YES` para permitir que o ntpd adiante o relógio, uma vez na inicialização. Normalmente, o ntpd registra uma mensagem de erro e se finaliza se o relógio estiver dessincronizado por mais de 1000 segundos. Essa opção é especialmente útil em sistemas sem um relógio em tempo real com bateria.

Defina `ntpd_oomprotect=YES` para proteger o serviço ntpd de ser finalizado pelo sistema quando ele tentar se recuperar de uma condição de Falta de Nemória (OOM).

Defina `ntpd_config=` para o local de um arquivo ntp.conf alternativo.

Defina `ntpd_flags=` para conter outras flags ntpd conforme necessário, mas evite usar as flags gerenciadas internamente pelo /etc/rc.d/ntpd:

- `-p` (local do arquivo pid)
- `-c` (configure `ntpd_config=` como alternativa)

29.11.1.3. O Ntpd e o usuário não privilegiado `ntpd`

O Ntpd no FreeBSD pode ser iniciado e executado como um usuário não privilegiado. Para isso, é necessário o módulo de política mac_ntpd(4). O script de inicialização /etc/rc.d/ntpd examina primeiro a configuração do NTP. Se possível, ele carrega o módulo `mac_ntpd` e inicia o ntpd como um usuário não vinculado `ntpd` (user id 123). Para evitar problemas com o acesso a arquivos e diretórios, o script de inicialização não iniciará automaticamente o ntpd como `ntpd` quando a configuração contiver quaisquer opções relacionadas a arquivos.

A presença de qualquer um dos itens a seguir em `ntpd_flags` requer configuração manual, conforme descrito abaixo, para ser executada como o usuário `ntpd` user:

- -f or --driftfile
- -i or --jaildir
- -k or --keyfile
- -l or --logfile
- -s or --statsdir

A presença de qualquer uma das seguintes palavras-chave no ntp.conf requer configuração manual, conforme descrito abaixo, para ser executado como usuário `ntpd`:

- crypto
- driftfile
- key
- logdir
- statsdir

Para configurar manualmente o ntpd para ser executado como usuário `ntpd`, você deve:

- Certifique-se de que o usuário `ntpd` tenha acesso a todos os arquivos e diretórios especificados na configuração.
- Se certifique para que o módulo `mac_ntpd` seja carregado ou compilado no kernel. Consulte mac_ntpd(4) para obter detalhes.
- Defina `ntpd_user="ntpd"` no /etc/rc.conf

29.11.2. Usando NTP com uma Conexão PPP

O ntpd não precisa de uma conexão permanente com a Internet para funcionar corretamente. No entanto, se uma conexão PPP estiver configurada para discar sob demanda, o tráfego de NTP deverá ser impedido de disparar uma discagem ou manter a conexão ativa. Isso pode ser configurado com as diretivas `filter` em /etc/ppp/ppp.conf. Por exemplo:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
```

```
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Para mais detalhes, consulte a seção PACKET FILTERING em ppp(8) e os exemplos em /usr/shared/examples/ppp/.



Alguns provedores de acesso à Internet bloqueiam portas de números baixos, impedindo o funcionamento do NTP, pois as respostas nunca chegam à máquina.

29.12. Inicializador iSCSI e Configuração Alvo

iSCSI é uma maneira de compartilhar o armazenamento em uma rede. Ao contrário do NFS, que funciona no nível do sistema de arquivos, o iSCSI funciona no nível do dispositivo de bloco.

Na terminologia iSCSI, o sistema que compartilha o armazenamento é conhecido como *alvo*. O armazenamento pode ser um disco físico ou uma área representando vários discos ou uma parte de um disco físico. Por exemplo, se os discos estiverem formatados com ZFS, um zvol poderá ser criado para ser usado como armazenamento iSCSI.

Os clientes que acessam o armazenamento do iSCSI são chamados de *iniciadores*. Para os iniciadores, o armazenamento disponível por meio do iSCSI aparece como um disco bruto, não formatado, conhecido como LUN. Nós de dispositivo para o disco aparecem em /dev/ e o dispositivo deve ser formatado e montado separadamente.

O FreeBSD fornece um alvo e iniciador nativo, baseado em kernel iSCSI. Esta seção descreve como configurar um sistema FreeBSD como um alvo ou um iniciador.

29.12.1. Configurando um Alvo iSCSI

Para configurar um alvo iSCSI, crie o arquivo de configuração /etc/ctl.conf, adicione uma linha ao arquivo /etc/rc.conf para certificar-se de que o daemon ctld(8) seja iniciado automaticamente na inicialização e, em seguida, inicie-o.

A seguir, um exemplo de um arquivo de configuração simples /etc/ctl.conf. Consulte ctl.conf(5) para obter uma descrição mais completa das opções disponíveis deste arquivo.

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}

target iqn.2012-06.com.example:target0 {
    auth-group no-authentication
    portal-group pg0
```

```
    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

A primeira entrada define o grupo de portais `pg0`. Grupos de portal definem quais endereços de rede o daemon ctld(8) irá escutar. A entrada `discovery-auth-group no-authentication` indica que qualquer iniciador tem permissão para executar descoberta de alvo iSCSI sem autenticação. As linhas três e quatro configuram ctld(8) para escutar em todos os endereços IPv4 (`listen 0.0.0.0`) e IPv6 (`listen [::]`) na porta padrão 3260.

Não é necessário definir um grupo de portais, pois há um grupo de portais interno chamado `default`. Nesse caso, a diferença entre `default` e `pg0` é que com `default`, a descoberta de alvo é sempre negada, enquanto com `pg0`, é sempre permitido.

A segunda entrada define um único alvo. O alvo tem dois significados possíveis: uma máquina que atende iSCSI ou um grupo nomeado de LUNs. Este exemplo usa o último significado, onde `iqn.2012-06.com.example:target0` é o nome do alvo. Este nome de alvo é adequado para fins de teste. Para uso real, altere `com.example` para o nome de domínio real, invertido. O `2012-06` representa o ano e o mês de aquisição do controle desse nome de domínio, e `target0` pode ser qualquer valor. Qualquer número de alvos pode ser definido neste arquivo de configuração.

A linha `auth-group no-authentication` permite que todos os iniciadores se conectem ao alvo especificado e `portal-group pg0` torna o alvo acessível através do grupo do portal `pg0`.

A próxima seção define o LUN. Para o iniciador, cada LUN será visível como um dispositivo de disco separado. Múltiplos LUNs podem ser definidos para cada destino. Cada LUN é identificado por um número, onde LUN 0 é obrigatório. A linha `path/data/target0-0` define o caminho completo para um arquivo ou zvol que suporta o LUN. Esse caminho deve existir antes de iniciar ctld(8). A segunda linha é opcional e especifica o tamanho do LUN.

Em seguida, para ter certeza que o daemon ctld(8) foi iniciado no boot, adicione esta linha ao arquivo /etc/rc.conf:

```
ctld_enable="YES"
```

Para iniciar o ctld(8) agora, execute este comando:

```
# service ctld start
```

Quando o daemon ctld(8) é iniciado, ele lê o arquivo /etc/ctl.conf. Se este arquivo for editado depois que o daemon iniciar, use este comando para que as mudanças entrem em vigor imediatamente:

```
# service ctld reload
```

29.12.1.1. Autenticação

O exemplo anterior é inerentemente inseguro, pois não usa autenticação, concedendo a qualquer um acesso total a todos os alvos. Para exigir um nome de usuário e senha para acessar os alvos, modifique a configuração da seguinte maneira:

```
auth-group ag0 {
    chap username1 secretsecret
    chap username2 anothersecret
}

portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}

target iqn.2012-06.com.example:target0 {
    auth-group ag0
    portal-group pg0
    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

A seção `auth-group` define os pares de nome de usuário e senha. Um inicializador tentando se conectar a `iqn.2012-06.com.example:target0` deve primeiro especificar um nome de usuário e senha definidos. No entanto, a descoberta do alvo ainda é permitida sem autenticação. Para exigir autenticação de descoberta de alvo, defina `discovery-auth-group` como um nome `auth-group` definido em vez de `no-authentication`.

É comum definir um único alvo exportado para cada inicializador. Como um atalho para a sintaxe acima, o nome de usuário e a senha podem ser especificados diretamente na entrada do alvo:

```
target iqn.2012-06.com.example:target0 {
    portal-group pg0
    chap username1 secretsecret

    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

29.12.2. Configurando um Inicializador iSCSI



O inicializador iSCSI descrito nesta seção é suportado a partir do FreeBSD 10.0-

> RELEASE. Para usar o inicializador iSCSI disponível em versões mais antigas, consulte iscontrol(8).

O inicializador iSCSI requer que o daemon iscsid(8) esteja em execução. Este daemon não usa um arquivo de configuração. Para iniciá-lo automaticamente na inicialização, adicione esta linha ao arquivo /etc/rc.conf:

```
iscsid_enable="YES"
```

Para iniciar iscsid(8) agora, execute este comando:

```
# service iscsid start
```

Conectar-se a um alvo pode ser feito com ou sem um arquivo /etc/iscsi.conf de configuração. Esta seção demonstra os dois tipos de conexões.

29.12.2.1. Conectando-se a um Alvo sem um Arquivo de Configuração

Para conectar um inicializador a um único alvo, especifique o endereço IP do portal e o nome do alvo:

```
# iscsictl -A -p 10.10.10.10 -t iqn.2012-06.com.example:target0
```

Para verificar se a conexão foi bem sucedida, execute `iscsictl` sem nenhum argumento. A saída deve ser semelhante a esta:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.10     Connected: da0
```

Neste exemplo, a sessão iSCSI foi estabelecida com sucesso, com /dev/da0 representando o LUN anexado. Se o destino `iqn.2012-06.com.example:target0` exportar mais de um LUN, vários nós de dispositivos serão mostrados nessa seção da saída:

```
Connected: da0 da1 da2.
```

Quaisquer erros serão relatados na saída, assim como os logs do sistema. Por exemplo, esta mensagem normalmente significa que o daemon iscsid(8) não está em execução:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.10     Waiting for iscsid(8)
```

A mensagem a seguir sugere um problema de rede, como uma porta ou endereço IP incorreto:

```
Target name                                     Target portal   State
iqn.2012-06.com.example:target0                 10.10.10.11     Connection refused
```

Esta mensagem significa que o nome do alvo especificado está errado:

```
Target name                                     Target portal   State
iqn.2012-06.com.example:target0                 10.10.10.10     Not found
```

Esta mensagem significa que o alvo requer autenticação:

```
Target name                                     Target portal   State
iqn.2012-06.com.example:target0                 10.10.10.10     Authentication failed
```

Para especificar um nome de usuário e uma senha de CHAP, use esta sintaxe:

```
# iscsictl -A -p 10.10.10.10 -t iqn.2012-06.com.example:target0 -u user -s
secretsecret
```

29.12.2.2. Conectando-se a um Alvo com um Arquivo de Configuração

Para se conectar usando um arquivo de configuração, crie o /etc/iscsi.conf com o seguinte conteúdo:

```
t0 {
    TargetAddress   = 10.10.10.10
    TargetName      = iqn.2012-06.com.example:target0
    AuthMethod      = CHAP
    chapIName       = user
    chapSecret      = secretsecret
}
```

O `t0` especifica um nickname para a seção do arquivo de configuração. Ele será usado pelo iniciador para especificar qual configuração usar. As outras linhas especificam os parâmetros a serem usados durante a conexão. O `TargetAddress` e `TargetName` são obrigatórios, enquanto as outras opções são opcionais. Neste exemplo, o nome de usuário e a senha do CHAP são mostrados.

Para se conectar ao alvo definido, especifique o apelido:

```
# iscsictl -An t0
```

Como alternativa, para conectar-se a todos os alvos definidos no arquivo de configuração, use:

```
# iscsictl -Aa
```

Para fazer com que o inicializador se conecte automaticamente a todos os alvos no arquivo /etc/iscsi.conf, adicione o seguinte ao arquivo /etc/rc.conf:

```
iscsictl_enable="YES"
iscsictl_flags="-Aa"
```

Capítulo 30. Firewalls

30.1. Sinopse

Os firewalls permitem filtrar o tráfego de entrada e saída que flui através de um sistema. Um firewall pode usar um ou mais conjuntos de "regras" para inspecionar os pacotes de rede à medida que eles entram ou saem das conexões de rede e assim permitir ou bloquear o tráfego. As regras de um firewall podem inspecionar uma ou mais características dos pacotes, como o tipo de protocolo, o endereço do host de origem ou de destino e a porta de origem ou de destino.

Os firewalls podem melhorar a segurança de um host ou de uma rede. Eles podem ser usados para fazer um ou mais dos seguintes procedimentos:

- Proteger e isolar as aplicações, serviços e máquinas de uma rede interna contra tráfego indesejado da Internet pública.
- Limitar ou desabilitar o acesso de hosts da rede interna para serviços da Internet pública.
- Suportar a tradução de endereços de rede (NAT), que possibilita que uma rede interna use endereços IP privados e compartilhe uma única conexão com a Internet pública usando um único endereço IP ou um pool compartilhado de endereços públicos atribuídos automaticamente.

O FreeBSD possui três firewalls embutidos no sistema base: PF, IPFW e IPFILTER, também conhecido como IPF. O FreeBSD também fornece dois traffic shapers para controlar o uso da largura de banda: altq(4) e dummynet(4). O ALTQ tem sido tradicionalmente vinculado ao PF e o dummynet ao IPFW. Cada firewall usa regras para controlar o acesso de pacotes provenientes e com destino a um sistema FreeBSD, embora eles façam isso de maneiras diferentes e cada um com uma sintaxe de regra diferente.

O FreeBSD fornece vários firewalls para atender aos diferentes requisitos e preferências para uma ampla variedade de usuários. Cada usuário deve avaliar qual firewall atende melhor às suas necessidades.

Depois de ler este capítulo, você saberá:

- Como definir regras de filtragem de pacotes.
- As diferenças entre os firewalls embutidos no FreeBSD.
- Como usar e configurar o firewall PF.
- Como usar e configurar o firewall IPFW.
- Como usar e configurar o firewall IPFILTER.

Antes de ler este capítulo, você deve:

- Entender os conceitos básicos do FreeBSD e de Internet.



Como todos os firewalls são baseados em inspecionar os valores dos campos de controle de pacotes selecionados, o criador do conjunto de regras do firewall deve

ter uma compreensão de como funciona o TCP/IP, quais são os diferentes valores nos campos de controle de pacotes e como esses valores são usados em uma conversa de sessão normal. Para uma boa introdução, consulte Daryl's TCP/IP Primer.

30.2. Conceitos de Firewall

Um conjunto de regras contém um grupo de regras que liberam ou bloqueiam pacotes com base nos valores contidos no pacote. A troca bidirecional de pacotes entre hosts compreende uma conversa de sessão. O conjunto de regras do firewall processa os pacotes que chegam da Internet pública, bem como os pacotes produzidos pelo sistema como uma resposta aos que chegaram. Cada serviço TCP/IP é pré-definido pelo seu protocolo e porta de escuta. Os pacotes destinados a um serviço específico são originados do endereço de origem usando uma porta não privilegiada e têm como destino a porta do serviço específica no endereço de destino. Todos os parâmetros acima podem ser usados como critérios de seleção para criar regras que irão liberar ou bloquear serviços.

Para procurar números de porta desconhecidos, consulte o arquivo /etc/services. Alternativamente, visite http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers e faça uma pesquisa de número de porta para encontrar a finalidade de um determinado número de porta.

Confira este link para ver os números de porta usados por Trojans.

O FTP possui dois modos: modo ativo e modo passivo. A diferença está em como o canal de dados é adquirido. O modo passivo é mais seguro, pois o canal de dados é adquirido pelo solicitante de sessão ftp. Para obter uma boa explicação sobre o FTP e seus diferentes modos, consulte http://www.slacksite.com/other/ftp.html.

Um conjunto de regras de firewall pode ser "exclusivo" ou "inclusivo". Um firewall exclusivo libera todo o tráfego, exceto o tráfego correspondente ao conjunto de regras. Um firewall inclusivo faz o inverso, liberando o tráfego que corresponde as regras e bloqueia todo o resto.

Um firewall inclusivo oferece melhor controle do tráfego de saída, tornando-o uma melhor escolha para sistemas que oferecem serviços à Internet pública. Também controla o tipo de tráfego originado da Internet pública que pode obter acesso a uma rede privada. Todo o tráfego que não corresponde às regras é bloqueado e registrado. Os firewalls inclusivos são geralmente mais seguros do que os firewalls exclusivos, pois reduzem significativamente o risco de permitir tráfego indesejado.



Salvo indicação contrária, todos os conjuntos de regras de configuração e exemplo neste capítulo criam conjuntos de regras de firewall inclusivos.

A segurança pode ser reforçada usando um "firewall stateful". Esse tipo de firewall registra e acompanha as conexões abertas e libera apenas o tráfego que corresponde a uma conexão existente ou libera e abre uma nova conexão.

A filtragem stateful trata o tráfego como uma troca bidirecional de pacotes compondo uma sessão. Quando um estado é especificado em uma regra de correspondência, o firewall gera dinamicamente regras internas para cada pacote antecipado sendo trocado durante a sessão. Ele

possui recursos de correspondência suficientes para determinar se um pacote é válido para uma sessão. Quaisquer pacotes que não se encaixem corretamente no modelo de sessão serão automaticamente rejeitados.

Quando a sessão é concluída, ela é removida da tabela de estados dinâmicos.

A filtragem stateful permite dar foco no bloqueio/liberação de novas sessões. Se a nova sessão for passada, todos os seus pacotes subsequentes serão permitidos automaticamente e todos os pacotes de um impostor serão automaticamente rejeitados. Se uma nova sessão for bloqueada, nenhum dos seus pacotes subsequentes serão permitidos. A filtragem stateful fornece habilidades de correspondência avançadas capazes de se defender contra o flood de diferentes métodos de ataque empregados pelos invasores.

NAT significa *Tradução de Endereços de Rede*. A função NAT permite que a LAN privada por trás do firewall compartilhe um único endereço IP atribuído pelo ISP, mesmo que esse endereço seja atribuído dinamicamente. O NAT permite que cada computador na LAN tenha acesso à Internet, sem ter que pagar ao ISP por várias contas de Internet ou endereços IP.

O NAT traduzirá automaticamente o endereço IP da LAN privada de cada sistema na LAN para o único endereço IP público, à medida que os pacotes saem do firewall vinculado à Internet pública. Também executa a conversão inversa para devolver os pacotes.

De acordo com a RFC 1918, os seguintes intervalos de endereços IP são reservados para redes privadas que nunca serão roteadas diretamente para a Internet pública e, portanto, estão disponíveis para uso com o NAT:

- `10.0.0.0/8`.
- `172.16.0.0/12`.
- `192.168.0.0/16`.



Ao trabalhar com regras de firewall, seja *muito cuidadoso*. Algumas configurações *podem bloquear o administrador* do servidor. Para estar seguro, considere executar a configuração inicial do firewall a partir do console local, em vez de fazê-lo remotamente por ssh.

30.3. PF

Desde o FreeBSD 5.3, uma versão portada do firewall PF do OpenBSD foi incluída como uma parte integrada do sistema base. O PF é um firewall completo, cheio de recursos que possui suporte opcional para ALTQ (Alternate Queuing), que fornece Qualidade de Serviço (QoS).

O Projeto OpenBSD mantém a referência definitiva para PF no FAQ do PF. Peter Hansteen mantém um tutorial completo do PF em http://home.nuug.no/~peter/pf/.



Ao ler o FAQ do PF, tenha em mente que a versão do PF do FreeBSD divergiu substancialmente da versão inicial do OpenBSD ao longo dos anos. Nem todos os recursos funcionam da mesma maneira no FreeBSD como no OpenBSD e vice-versa.

A lista de emails do packet filter do FreeBSD é um bom lugar para perguntar questões relacionadas a configuração e execução do firewall PF. Verifique os arquivos da lista de email antes de perguntar alguma questão, pois ela já pode ter sido respondida.

Esta seção do Handbook foca no PF no que se refere ao FreeBSD. Ele demonstra como ativar o PF e ALTQ. Em seguida, ele fornece vários exemplos para criar conjuntos de regras em um sistema FreeBSD.

30.3.1. Ativando o PF

Para usar o PF, seu módulo do kernel deve ser carregado primeiro. Esta seção descreve as entradas que podem ser adicionadas ao /etc/rc.conf para habilitar o PF.

Comece adicionando a seguinte linha `pf_enable=yes` ao arquivo /etc/rc.conf:

```
# sysrc pf_enable=yes
```

Opções adicionais, descritas em pfctl(8), podem ser passadas para o PF quando ele é iniciado. Adicione esta entrada ao arquivo /etc/rc.conf e especifique quaisquer flags necessárias entre duas aspas (""):

```
pf_flags=""                   # additional flags for pfctl startup
```

O PF não será iniciado se não puder localizar o arquivo de configuração do conjunto de regras. Por padrão, o FreeBSD não vem com um conjunto de regras e não há um /etc/pf.conf. Exemplos de regras podem ser encontrados em /usr/shared/examples/pf/. Se um conjunto de regras personalizado foi salvo em algum outro lugar, adicione uma linha ao arquivo /etc/rc.conf que especifica o caminho completo para o arquivo:

```
pf_rules="/path/to/pf.conf"
```

O suporte de log para o PF é fornecido pelo pflog(4). Para ativar o suporte aos logs, adicione esta linha ao /etc/rc.conf:

```
# sysrc pflog_enable=yes
```

As seguintes linhas também podem ser adicionadas para alterar a localização padrão do arquivo de log ou para especificar quaisquer flags adicionais na inicialização do pflog(4):

```
pflog_logfile="/var/log/pflog"  # where pflogd should store the logfile
pflog_flags=""                  # additional flags for pflogd startup
```

Finalmente, se houver uma LAN atrás do firewall e os pacotes precisarem ser encaminhados para os computadores na LAN, ou se NAT for necessário, adicione a seguinte opção:

```
gateway_enable="YES"            # Enable as LAN gateway
```

Depois de salvar as edições necessárias, o PF pode ser iniciado com o suporte de log, digitando:

```
# service pf start
# service pflog start
```

Por padrão, o PF lê suas regras de configuração do arquivo /etc/pf.conf e modifica, descarta ou libera pacotes de acordo com as regras ou definições especificadas neste arquivo. A instalação do FreeBSD inclui vários arquivos de exemplo localizados em /usr/shared/examples/pf/. Consulte o FAQ do PF para obter uma cobertura completa dos conjuntos de regras do PF.

Para controlar o PF, use o `pfctl`. Opções Úteis do `pfctl` resume algumas opções úteis para este comando. Consulte pfctl(8) para obter uma descrição de todas as opções disponíveis:

Tabela 27. Opções Úteis do `pfctl`

Comando	Propósito
`pfctl -e`	Ativa o PF.
`pfctl -d`	Desabilita o PF.
`pfctl -F all -f /etc/pf.conf`	Limpa todas as regras de NAT, filtro, estado e tabela e recarrega o /etc/pf.conf.
`pfctl -s [ rules | nat | states ]`	Informa as regras de filtragem, de NAT ou a tabela de estados.
`pfctl -vnf /etc/pf.conf`	Verifica se tem erros no arquivo /etc/pf.conf, mas não carrega o conjunto de regras.



security/sudo é útil para executar comandos como `pfctl` que exigem privilégios elevados. Ele pode ser instalado a partir da Coleção de Ports.

Para ficar de olho no tráfego que passa pelo firewall PF, considere instalar o pacote ou port sysutils/pftop. Uma vez instalado, o pftop pode ser executado para exibir um snapshot do estado atual do tráfego em um formato semelhante ao top(1).

30.3.2. Conjuntos de Regras do PF

Esta seção demonstra como criar um conjunto de regras personalizado. Ele começa com o mais simples dos conjuntos de regras e baseia-se em seus conceitos usando vários exemplos para demonstrar o uso real dos diversos recursos do PF.

O conjunto de regras mais simples possível é para uma única máquina que não executa nenhum serviço e que precisa de acesso a uma rede, que pode ser a Internet. Para criar este conjunto de regras mínimo, edite o arquivo /etc/pf.conf para que fique assim:

```
block in all
```

```
pass out all keep state
```

A primeira regra nega todo o tráfego de entrada por padrão. A segunda regra permite que as conexões originadas por este sistema sejam liberadas, mantendo as informações de estado nessas conexões. Essas informações de estado permitem que o tráfego de retorno para essas conexões seja liberado e só deve ser usado em máquinas confiáveis. O conjunto de regras pode ser carregado com:

```
# pfctl -e ; pfctl -f /etc/pf.conf
```

Além de manter estados, o PF fornece *listas* e *macros* que podem ser definidas para uso ao criar regras. As macros podem incluir listas e precisam ser definidas antes de serem usadas. Como exemplo, insira essas linhas no topo do conjunto de regras:

```
tcp_services = "{ ssh, smtp, domain, www, pop3, auth, pop3s }"
udp_services = "{ domain }"
```

O PF entende os nomes das portas, assim como os números das portas, desde que os nomes estejam listados em /etc/services. Este exemplo cria duas macros. A primeira é uma lista de sete nomes de portas TCP e a segunda é um nome de porta UDP. Uma vez definidas, as macros podem ser usadas em regras. Neste exemplo, todo o tráfego é bloqueado, exceto pelas conexões originadas por este sistema para os sete serviços TCP especificados e para o serviço UDP especificado:

```
tcp_services = "{ ssh, smtp, domain, www, pop3, auth, pop3s }"
udp_services = "{ domain }"
block all
pass out proto tcp to any port $tcp_services keep state
pass proto udp to any port $udp_services keep state
```

Embora o UDP seja considerado um protocolo sem estado, o PF é capaz de rastrear algumas informações de estado. Por exemplo, quando uma solicitação UDP é liberada perguntando a um servidor de nomes sobre um nome de domínio, o PF irá procurar pela resposta para libera-la.

Sempre que uma edição é feita em um conjunto de regras, as novas regras devem ser carregadas para que possam ser usadas:

```
# pfctl -f /etc/pf.conf
```

Se não houver erros de sintaxe, o `pfctl` não exibirá nenhuma mensagem durante o carregamento da regra. As regras também podem ser testadas antes de tentar carregá-las:

```
# pfctl -nf /etc/pf.conf
```

A inclusão de `-n` faz com que as regras sejam interpretadas apenas, mas não carregadas. Isso fornece uma oportunidade para corrigir quaisquer erros. Em todos os momentos, o último conjunto de regras válido carregado será imposto até que o PF seja desativado ou um novo conjunto de regras seja carregado.



Adicionando `-v` ao comando `pfctl` no carregamento ou checagem de conjuntos de regras, será exibido as regras exatamente da maneira como elas serão carregadas. Isso é extremamente útil ao depurar regras.

30.3.2.1. Um Gateway Simples com NAT

Esta seção demonstra como configurar um sistema FreeBSD executando PF para atuar como um gateway para pelo menos uma outra máquina. O gateway precisa de pelo menos duas interfaces de rede, cada uma conectada a uma rede separada. Neste exemplo, xl1 está conectada à Internet e a xl0 está conectada à rede interna.

Primeiro, ative o gateway para permitir que a máquina encaminhe o tráfego de rede que recebe em uma interface para outra interface. Esta configuração do sysctl encaminhará pacotes IPv4:

```
# sysctl net.inet.ip.forwarding=1
```

Para encaminhar tráfego IPv6, use:

```
# sysctl net.inet6.ip6.forwarding=1
```

Para ativar essas configurações na inicialização do sistema, use o sysrc(8) para adicioná-las ao /etc/rc.conf:

```
# sysrc gateway_enable=yes
# sysrc ipv6_gateway_enable=yes
```

Verifique com o `ifconfig` se ambas as interfaces estão ativadas e em execução.

Em seguida, crie as regras PF para permitir que o gateway transmita tráfego. Embora a regra a seguir permita que o tráfego stateful de hosts da rede interna passe para o gateway, a palavra-chave `to` não garante a passagem da origem até o destino:

```
pass in on xl1 from xl1:network to xl0:network port $ports keep state
```

Essa regra só permite que o tráfego passe para o gateway na interface interna. Para deixar os pacotes irem mais longe, é necessária uma regra de correspondência:

```
pass out on xl0 from xl1:network to xl0:network port $ports keep state
```

Embora essas duas regras funcionem, regras especificadas dessa forma raramente são necessárias. Para um administrador de rede ocupado, um conjunto de regras legível é um conjunto de regras mais seguro. O restante desta seção demonstra como manter as regras o mais simples possível para facilitar a leitura. Por exemplo, essas duas regras podem ser substituídas por uma regra:

```
pass from xl1:network to any port $ports keep state
```

A notação `interface:network` pode ser substituída por uma macro para tornar o conjunto de regras ainda mais legível. Por exemplo, uma macro `$localnet` pode ser definida como a rede diretamente conectada à interface interna (`$xl1:network`). Alternativamente, a definição de `$localnet` poderia ser alterada para uma notação *IP address/netmask* para denotar uma rede, como `192.168.100.1/24` para uma sub-rede de endereços privados.

Se necessário, `$localnet` pode ser definido como uma lista de redes. Quaisquer que sejam as necessidades específicas, uma definição sensata de `$localnet` poderia ser usada em uma regra típica de liberação da seguinte maneira:

```
pass from $localnet to any port $ports keep state
```

O conjunto de regras de exemplo a seguir libera todo o tráfego originado por máquinas na rede interna. Primeiro define duas macros para representar as interfaces externas e internas 3COM do gateway.



Para usuários dial-up, a interface externa será tun0. Para uma conexão ADSL, especificamente aquelas que usam PPP over Ethernet (PPPoE), a interface externa correta é tun0, não a interface física Ethernet.

```
ext_if = "xl0"  # macro for external interface - use tun0 for PPPoE
int_if = "xl1"  # macro for internal interface
localnet = $int_if:network
# ext_if IP address could be dynamic, hence ($ext_if)
nat on $ext_if from $localnet to any -> ($ext_if)
block all
pass from { lo0, $localnet } to any keep state
```

Este conjunto de regras introduz a regra `nat` que é usada para tratar a tradução de endereços de rede dos endereços não roteáveis dentro da rede interna para o endereço IP atribuído à interface externa. Os parênteses em torno da última parte da regra nat `($ext_if)` são incluídos quando o endereço IP da interface externa é atribuído dinamicamente. Ele garante que o tráfego de rede seja executado sem interrupções graves, mesmo se o endereço IP externo for alterado.

Observe que esse conjunto de regras provavelmente permite que mais tráfego seja transmitido para fora da rede do que o necessário. Uma configuração razoável poderia criar essa macro:

```
client_out = "{ ftp-data, ftp, ssh, domain, pop3, auth, nntp, http, \
```

```
    https, cvspserver, 2628, 5999, 8000, 8080 }"
```

para usar na regra principal de liberação:

```
pass inet proto tcp from $localnet to any port $client_out \
    flags S/SA keep state
```

Algumas outras regras de aprovação podem ser necessárias. Esta permite ativar o SSH na interface externa:

```
pass in inet proto tcp to $ext_if port ssh
```

Esta definição de macro e regra permite DNS e NTP para clientes internos:

```
udp_services = "{ domain, ntp }"
pass quick inet proto { tcp, udp } to any port $udp_services keep state
```

Observe a palavra-chave `quick` nesta regra. Como o conjunto de regras consiste em várias regras, é importante entender as relações entre as regras em um conjunto de regras. As regras são avaliadas de cima para baixo, na sequência em que são escritas. Para cada pacote ou conexão avaliado pelo PF, *a última regra correspondente* no conjunto de regras é aquela que é aplicada. No entanto, quando um pacote corresponde a uma regra que contém a palavra-chave `quick`, o processamento da regra é interrompido e o pacote é tratado de acordo com essa regra. Isso é muito útil quando é necessária uma exceção às regras gerais.

30.3.2.2. Criando um Proxy FTP

Configurar regras funcionais de FTP pode ser problemático devido à natureza do protocolo FTP. O FTP pré-data os firewalls por várias décadas e é inseguro em seu design. Os pontos mais comuns contra o uso do FTP incluem:

- As senhas são transferidas em texto puro.
- O protocolo exige o uso de pelo menos duas conexões TCP (controle e dados) em portas separadas.
- Quando uma sessão é estabelecida, os dados são transmitidos usando portas selecionadas aleatoriamente.

Todos esses pontos apresentam desafios de segurança, mesmo antes de considerar possíveis pontos fracos de segurança no software cliente ou servidor. Há alternativas mais seguras para a transferência de arquivos, como sftp(1) ou scp(1), que apresentam autenticação e transferência de dados através de conexões criptografadas.

Para as situações em que o FTP é necessário, o PF fornece o redirecionamento do tráfego FTP para um pequeno programa proxy chamado ftp-proxy(8), que está incluído no sistema base do FreeBSD. O papel do proxy é inserir dinamicamente e excluir regras no conjunto de regras, usando um

conjunto de âncoras, para lidar corretamente com o tráfego de FTP.

Para habilitar o proxy FTP, adicione esta linha ao /etc/rc.conf:

```
ftpproxy_enable="YES"
```

Em seguida, inicie o proxy executando `service ftp-proxy start`.

Para uma configuração básica, três elementos precisam ser adicionados ao arquivo /etc/pf.conf. Primeiro, as âncoras que o proxy usará para inserir as regras que ele gera para as sessões de FTP:

```
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"
```

Em segundo, é necessária uma regra de liberação para permitir o tráfego de FTP para o proxy.

Terceiro, as regras de redirecionamento e NAT precisam ser definidas antes das regras de filtragem. Insira esta regra `rdr` imediatamente após a regra `nat`:

```
rdr pass on $int_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021
```

Finalmente, permita que o tráfego redirecionado passe:

```
pass out proto tcp from $proxy to any port ftp
```

onde `$proxy` se expande para o endereço ao qual o daemon proxy está vinculado.

Salve o arquivo /etc/pf.conf, carregue as novas regras e verifique a partir de um cliente se as conexões FTP estão funcionando:

```
# pfctl -f /etc/pf.conf
```

Este exemplo cobre uma configuração básica em que os clientes na rede local precisam entrar em contato com servidores FTP em outro lugar. Essa configuração básica deve funcionar bem com a maioria das combinações de clientes e servidores FTP. Como mostrado em ftp-proxy(8), o comportamento do proxy pode ser alterado de várias maneiras adicionando opções na linha `ftpproxy_flags=`. Alguns clientes ou servidores podem ter peculiaridades específicas que devem ser compensadas na configuração ou pode ser necessário integrar o proxy de maneiras específicas, como atribuir tráfego FTP a uma fila específica.

Para formas de executar um servidor FTP protegido por PF e ftp-proxy(8), configure um `ftp-proxy` separado em modo reverso, usando `-R`, em uma porta separada com sua própria regra de redirecionamento de passagem.

30.3.2.3. Gerenciando ICMP

Muitas das ferramentas usadas para depurar ou solucionar problemas de uma rede TCP/IP dependem do Internet Control Message Protocol (ICMP), o qual foi projetado especificamente para depuração.

O protocolo ICMP envia e recebe *mensagens de controle* entre hosts e gateways, principalmente para fornecer feedback a um remetente sobre quaisquer condições incomuns ou difíceis na rota para o host de destino. Os roteadores usam ICMP para negociar tamanhos de pacote e outros parâmetros de transmissão em um processo geralmente chamado de descoberta de *path MTU*.

Do ponto de vista do firewall, algumas mensagens de controle ICMP são vulneráveis a vetores de ataque conhecidos. Além disso, deixar todo o tráfego de diagnóstico passar incondicionalmente torna a depuração mais fácil, mas também torna mais fácil para os outros extraírem informações sobre a rede. Por esses motivos, a regra a seguir pode não ser a ideal:

```
pass inet proto icmp from any to any
```

Uma solução é permitir todo o tráfego de ICMP originado na rede local e bloquear as chamadas provenientes de fora da rede:

```
pass inet proto icmp from $localnet to any keep state
pass inet proto icmp from any to $ext_if keep state
```

Opções adicionais estão disponíveis, o que demonstra algumas das flexibilidades do PF. Por exemplo, em vez de liberar todas as mensagens ICMP, pode-se especificar as mensagens usadas pelo ping(8) e traceroute(8). Comece definindo uma macro para esse tipo de mensagem:

```
icmp_types = "echoreq"
```

e uma regra que usa a macro:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

Se outros tipos de pacotes ICMP forem necessários, expanda `icmp_types` para uma lista desses tipos de pacotes. Digite `more /usr/src/sbin/pfctl/pfctl_parser.c` para ver a lista de tipos de mensagem ICMP suportados pelo PF. Consulte http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml para uma explicação de cada tipo de mensagem.

Como o Unix `traceroute` usa UDP por padrão, outra regra é necessária para permitir o comando `traceroute` do Unix:

```
# allow out the default range for traceroute(8):
pass out on $ext_if inet proto udp from any to any port 33433 >< 33626 keep state
```

Como o TRACERT.EXE em sistemas Microsoft Windows usa ICMP echo request messages, somente a primeira regra é necessária para permitir rastreamentos de rede desses sistemas. O Unix traceroute também pode ser instruído a usar outros protocolos e usará ICMP echo request messages se -I for usado. Verifique a página de manual traceroute(8) para detalhes.

30.3.2.3.1. Descoberta de Path MTU

Os protocolos de Internet são projetados para serem independentes do dispositivo, e uma consequência da independência do dispositivo é que o tamanho ideal do pacote para uma determinada conexão nem sempre pode ser previsto com segurança. A principal restrição no tamanho do pacote é a *Maximum Transmission Unit* (MTU), que define o limite superior do tamanho do pacote para uma interface. Digite ifconfig para exibir os MTUs para as interfaces de rede do sistema.

O TCP/IP usa um processo conhecido como descoberta de path MTU para determinar o tamanho correto do pacote para uma conexão. Este processo envia pacotes de tamanhos variados com o conjunto de flag "Não fragmentar", esperando um pacote de retorno ICMP de "tipo 3, código 4" quando o limite for alcançado. O tipo 3 significa "destino inacessível", e o código 4 é uma abreviação para "fragmentação necessária, mas a flag para não fragmentar está definida". Para permitir que a descoberta de path MTU suporte conexões com outros MTUs, adicione o tipo destination unreachable à macro icmp_types:

```
icmp_types = "{ echoreq, unreach }"
```

Como a regra de liberação já usa essa macro, ela não precisa ser modificada para suportar o novo tipo de ICMP:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

O PF permite filtrar todas as variações dos tipos e códigos de ICMP. A lista de tipos e códigos possíveis está documentada em icmp(4) and icmp6(4).

30.3.2.4. Usando Tabelas

Alguns tipos de dados são relevantes para filtragem e redirecionamento em um determinado momento, mas sua definição é muito longa para ser incluída no arquivo do conjunto de regras. O PF suporta o uso de tabelas, que são listas definidas que podem ser manipuladas sem a necessidade de recarregar todo o conjunto de regras e que podem fornecer pesquisas rápidas. Nomes de tabelas são sempre colocados dentro de < >, assim:

```
table <clients> { 192.168.2.0/24, !192.168.2.5 }
```

Neste exemplo, a rede 192.168.2.0/24 faz parte da tabela, exceto pelo endereço 192.168.2.5, que é excluído pelo operador !. Também é possível carregar tabelas de arquivos onde cada entrada está em uma linha separada. como neste exemplo /etc/clients:

```
192.168.2.0/24
!192.168.2.5
```

Para se referir ao arquivo, defina a tabela da seguinte forma:

```
table <clients> persist file "/etc/clients"
```

Depois que a tabela é definida, ela pode ser referenciada por uma regra:

```
pass inet proto tcp from <clients> to any port $client_out flags S/SA keep state
```

O conteúdo de uma tabela pode ser manipulado ao vivo, usando `pfctl`. Este exemplo adiciona outra rede a tabela:

```
# pfctl -t clients -T add 192.168.1.0/16
```

Observe que quaisquer alterações feitas dessa maneira terão efeito imediato, tornando-as ideais para testes, mas não sobreviverão a uma falha de energia ou reinicialização. Para tornar as alterações permanentes, modifique a definição da tabela no conjunto de regras ou edite o arquivo a que a tabela se refere. É possível manter a cópia em disco da tabela usando uma tarefa cron(8) que copia o conteúdo da tabela para o disco em intervalos de tempo, usando um comando como `pfctl -t clients -T show >/etc/clients`. Alternativamente, o /etc/clients pode ser atualizado com o conteúdo da tabela na memória:

```
# pfctl -t clients -T replace -f /etc/clients
```

30.3.2.5. Usando Tabelas de Sobrecarga para Proteger o SSH

Aqueles que executam o SSH em uma interface externa provavelmente já viram algo assim nos logs de autenticação:

```
Sep 26 03:12:34 skapet sshd[25771]: Failed password for root from 200.72.41.31 port
40992 ssh2
Sep 26 03:12:34 skapet sshd[5279]: Failed password for root from 200.72.41.31 port
40992 ssh2
Sep 26 03:12:35 skapet sshd[5279]: Received disconnect from 200.72.41.31: 11: Bye Bye
Sep 26 03:12:44 skapet sshd[29635]: Invalid user admin from 200.72.41.31
Sep 26 03:12:44 skapet sshd[24703]: input_userauth_request: invalid user admin
Sep 26 03:12:44 skapet sshd[24703]: Failed password for invalid user admin from
200.72.41.31 port 41484 ssh2
```

Isso indica um ataque de força bruta em que alguém ou algum programa está tentando descobrir o nome de usuário e senha que os permitirá entrar no sistema.

Se o acesso externo ao SSH for necessário para usuários legítimos, a alteração da porta padrão usada pelo SSH pode oferecer alguma proteção. No entanto, o PF fornece uma solução mais elegante. As regras de liberação podem conter limites sobre o que os hosts de conexão podem fazer e os violadores podem ser banidos para uma tabela de endereços aos quais é negado algum ou todo o acesso. É até possível descartar todas as conexões existentes de máquinas que excedem os limites.

Para configurar isso, crie esta tabela na seção de tabelas do conjunto de regras:

```
table <bruteforce> persist
```

Então, em algum lugar no início do conjunto de regras, adicione regras para bloquear o acesso bruto, permitindo acesso legítimo:

```
block quick from <bruteforce>
pass inet proto tcp from any to $localnet port $tcp_services \
    flags S/SA keep state \
    (max-src-conn 100, max-src-conn-rate 15/5, \
    overload <bruteforce> flush global)
```

A parte entre parênteses define os limites e os valores devem ser alterados para atender aos requisitos locais. Isso pode ser lido da \seguinte forma:

`max-src-conn` é o número de conexões simultâneas permitidas de um host.

`max-src-conn-rate` é a taxa de novas conexões permitidas de qualquer host único (*15*) por número de segundos (*5*).

`overload <bruteforce>` significa que qualquer host que excede esses limites obtém seu endereço adicionado à tabela `bruteforce`. O conjunto de regras bloqueia todo o tráfego de endereços na tabela `bruteforce`.

Finalmente, `flush global` diz que quando um host atinge o limite, todo (`global`) das conexões desse host será finalizado (`flush`).



Estas regras *não* irão bloquear bruteforcers lentos, como descrito em http://home.nuug.no/~peter/hailmary2013/.

Este conjunto de regras de exemplo é projetado principalmente como uma ilustração. Por exemplo, se um número grande de conexões em geral é desejado, mas o desejo é ser mais restritivo quando se trata de ssh, complemente a regra acima com algo como o abaixo, no início do conjunto de regras:

```
pass quick proto { tcp, udp } from any to any port ssh \
    flags S/SA keep state \
    (max-src-conn 15, max-src-conn-rate 5/3, \
    overload <bruteforce> flush global)
```



Pode Não ser Necessário Bloquear Todos os Overloaders

É importante notar que o mecanismo de sobrecarga é uma técnica geral que não se aplica exclusivamente ao SSH, e nem sempre é ideal bloquear totalmente todo o tráfego dos infratores.

Por exemplo, uma regra de sobrecarga pode ser usada para proteger um serviço de email ou um serviço Web e a tabela de sobrecarga pode ser usada em uma regra para atribuir infratores a uma fila com uma alocação de largura de banda mínima ou redirecionar para uma página Web específica.

Com o tempo, as tabelas serão preenchidas por regras de sobrecarga e seu tamanho crescerá incrementalmente, ocupando mais memória. Às vezes, um endereço de IP que é bloqueado é atribuído dinamicamente, que já foi atribuído a um host que tem um motivo legítimo para se comunicar com hosts na rede local.

Para situações como essas, o pfctl fornece a capacidade de expirar as entradas da tabela. Por exemplo, este comando removerá entradas de tabela `<bruteforce>` que não foram referenciadas por `86400` segundos:

```
# pfctl -t bruteforce -T expire 86400
```

Funcionalidade semelhante é fornecida por security/expiretable, que remove entradas de tabela que não foram acessadas por um período de tempo especificado.

Uma vez instalado, o expiretable pode ser executado para remover entradas de tabela `<bruteforce>` mais antigas que uma tempo especifico. Este exemplo remove todas as entradas com mais de 24 horas:

```
/usr/local/sbin/expiretable -v -d -t 24h bruteforce
```

30.3.2.6. Protegendo Contra SPAM

Não deve ser confundido com o daemon spamd que vem junto com spamassassin, mail/spamd pode ser configurado com o PF para fornecer uma defesa externa contra SPAM. Esse spamd conecta-se à configuração do PF usando um conjunto de redirecionamentos.

Os spammers tendem a enviar um grande número de mensagens, e o SPAM é enviado principalmente de algumas redes amigáveis de spammers e um grande número de máquinas sequestradas, sendo que ambas são reportadas a *blacklists* bem rápido.

Quando uma conexão SMTP de um endereço que está em uma blacklist é recebido, o spamd apresenta seu banner e imediatamente muda para um modo em que ele responde o trágefo SMTP um byte de cada vez. Esta técnica, que pretende desperdiçar tanto tempo quanto possível do spammer, é chamada de *tarpitting*. A implementação específica que usa respostas de um byte SMTP é muitas vezes referenciada como *stuttering*.

Este exemplo demonstra o procedimento básico para configurar o spamd com blacklists atualizadas

automaticamente. Consulte as páginas de manual que são instaladas com o mail/spamd para mais informações.

Procedure: Configurando o spamd

1. Instale o pacote ou port mail/spamd. Para usar os recursos de greylist do spamd, fdescfs(5) deve ser montado em /dev/fd. Adicione a seguinte linha ao arquivo /etc/fstab:

   ```
   fdescfs /dev/fd fdescfs rw 0 0
   ```

   Em seguida, monte o sistema de arquivos:

   ```
   # mount fdescfs
   ```

2. Em seguida, edite o conjunto de regras do PF para incluir:

   ```
   table <spamd> persist
   table <spamd-white> persist
   rdr pass on $ext_if inet proto tcp from <spamd> to \
       { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025
   rdr pass on $ext_if inet proto tcp from !<spamd-white> to \
       { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025
   ```

   As duas tabelas `<spamd>` e `<spamd-white>` são essenciais. O trafego SMTP de um endereço listado em `<spamd>` mas não em `<spamd-white>` é redirecionado para o daemon spamd ouvindo a porta 8025.

3. O próximo passo é configurar o spamd no arquivo /usr/local/etc/spamd.conf e adicionar alguns parâmetros no arquivo rc.conf.

   A instalação do mail/spamd inclui um arquivo de configuração de exemplo (/usr/local/etc/spamd.conf.sample) e uma página de manual para o spamd.conf. Refira-se a estes para opções adicionais de configuração além daquelas mostradas neste exemplo.

   Uma das primeiras linhas no arquivo de configuração que não começa com um sinal de comentário `#` contém o bloco que define a lista `all`, que especifica as listas a serem usadas:

   ```
   all:\
       :traplist:whitelist:
   ```

   Esta entrada adiciona as blacklists desejadas, separadas por dois pontos (`:`). Para usar uma whitelist para subtrair endereços de uma blacklist, adicione o nome da whitelist *imediatamente* após o nome dessa blacklist. Por exemplo: `:blacklist:whitelist:`.

   Isto é seguido pela definição da blacklist especificada:

```
traplist:\
    :black:\
    :msg="SPAM. Your address %A has sent spam within the last 24 hours":\
    :method=http:\
    :file=www.openbsd.org/spamd/traplist.gz
```

onde a primeira linha é o nome da blacklist e a segunda linha especifica o tipo da lista. O campo `msg` contém a mensagem a ser exibida aos remetentes da blacklist durante a comunicação SMTP. O campo `method` especifica como o spamd-setup busca os dados da lista; os métodos suportados são `http`, `ftp`, de um `arquivo` em um sistema de arquivos montado e via `exec` de um programa externo. Finalmente, o campo `file` especifica o nome do arquivo que o spamd espera receber.

A definição da whitelist especificada é semelhante, mas omite o campo `msg` porque uma mensagem não é necessária:

```
whitelist:\
    :white:\
    :method=file:\
    :file=/var/mail/whitelist.txt
```



Escolha Fontes de Dados com Cuidado

Usar todas as blacklists do arquivo de exemplo spamd.conf irá colocar na blacklist grandes blocos da Internet. Os administradores precisam editar o arquivo para criar uma configuração ideal que use fontes de dados aplicáveis e, quando necessário, use listas personalizadas.

Em seguida, adicione esta entrada ao arquivo /etc/rc.conf. Flags adicionais são descritas na página de manual especificada pelo comentário:

```
spamd_flags="-v" # use "" and see spamd-setup(8) for flags
```

Quando terminar, recarregue o conjunto de regras, inicio o spamd digitando `service obspamd start`, e complete a configuração usando `spamd-setup`. Finalemente, crie uma tarefa cron(8) que chame `spamd-setup` para atualizar as tabelas razoáveis.

Em um gateway típico na frente de um servidor de email, os hosts logo começam a ficar presos dentro de segundos ou alguns minutos.

PF também suporta *greylist*, que rejeita temporariamente mensagens de hosts desconhecidos com códigos *45n*. Conexões de hosts que estão na greylist e que tentam novamente dentro de um tempo razoável de tempo são liberados. O tráfego de remetentes que estão configurados para se comportarem dentro dos limites estabelecidos pela RFC 1123 e pela RFC 2821 é imediatamente permitido.

Mais informações sobre técnicas de greylist podem ser encontradas no site greylisting.org. A coisa mais surpreendente sobre greylist, além de sua simplicidade, é que ainda funciona. Os spammers e os criadores de malware têm sido muito lentos para se adaptar, a fim de contornar essa técnica.

O procedimento básico para configurar o greylist é o seguinte:

Procedure: Configurando Greylist

1. Certifique-se de que fdescfs(5) esteja montado conforme descrito na Etapa 1 do Procedimento anterior.
2. Para executar spamd no modo greylist, adicione esta linha ao /etc/rc.conf:

   ```
   spamd_grey="YES"  # use spamd greylisting if YES
   ```

   Consulte a página de manual do spamd para obter descrições de parâmetros relacionados adicionais.
3. Para concluir a configuração da greylist:

   ```
   # service obspamd restart
   # service obspamlogd start
   ```

Nos bastidores, a ferramenta de banco de dados spamdb e o atualizador de whitelist spamlogd executam funções essenciais para o recurso de greylist. O spamdb é a interface principal do administrador para gerenciar as greylists, blacklists e whitelists por meio do conteúdo do banco de dados /var/db/spamdb.

30.3.2.7. Higiene de Rede

Esta seção descreve como o `block-policy`, `scrub`, e `antispoof` pode ser usado para fazer o conjunto de regras se comportar corretamente.

O `block-policy` é uma opção que pode ser definida na parte de `opções` do conjunto de regras, que precede as regras de redirecionamento e filtragem. Essa opção determina qual feedback, se houver, que o PF envia para hosts que são bloqueados por uma regra. A opção tem dois valores possíveis: `drop` descarta pacotes bloqueados sem feedback, e `return` retorna um código de status como `Connection refused`.

Se não definido, a política padrão é `drop`. Para alterar o `block-policy`, especifique o valor desejado:

```
set block-policy return
```

No PF, `scrub` é uma palavra-chave que permite a normalização do pacote de rede. Esse processo remonta pacotes fragmentados e descarta pacotes TCP que possuem combinações de sinalizadores inválidos. Ativar `scrub` fornece uma medida de proteção contra certos tipos de ataques com base no

manuseio incorreto de fragmentos de pacotes. Várias opções estão disponíveis, mas a forma mais simples é adequada para a maioria das configurações:

```
scrub in all
```

Alguns serviços, como o NFS, exigem opções específicas de manipulação de fragmentos. Consulte https://home.nuug.no/~peter/pf/en/scrub.html para mais informações.

Este exemplo remonta fragmentos, limpa o bit "não fragmentar" e define o tamanho máximo do segmento para 1440 bytes:

```
scrub in all fragment reassemble no-df max-mss 1440
```

O mecanismo `antispoof` protege contra a atividade de endereços IP falsos ou forjados, principalmente bloqueando pacotes que aparecem em interfaces e em direções que logicamente não são possíveis.

Essas regras eliminam tráfego falsificado do resto do mundo, bem como qualquer pacote falsificado originado na rede local:

```
antispoof for $ext_if
antispoof for $int_if
```

30.3.2.8. Manipulando Endereços Não-Roteados

Mesmo com um gateway configurado adequadamente para lidar com a tradução de endereços de rede, pode ser necessário compensar as configurações incorretas de outras pessoas. Uma configuração incorreta comum é permitir o tráfego com endereços não roteáveis para a Internet. Como o tráfego de endereços não roteados pode desempenhar um papel em várias técnicas de ataque de DoS, considere bloquear explicitamente o tráfego de endereços não roteáveis de entrar na rede por meio da interface externa.

Neste exemplo, uma macro contendo endereços não roteáveis é definida e usada em regras de bloqueio. O tráfego de origem e destino para esses endereços é silenciosamente descartado na interface externa do gateway.

```
martians = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, \
          10.0.0.0/8, 169.254.0.0/16, 192.0.2.0/24, \
          0.0.0.0/8, 240.0.0.0/4 }"

block drop in quick on $ext_if from $martians to any
block drop out quick on $ext_if from any to $martians
```

30.3.3. Ativando o ALTQ

No FreeBSD, o ALTQ pode ser usado com PF para fornecer Qualidade de Serviço (QOS). Depois que o ALTQ é ativado, as filas podem ser definidas no conjunto de regras que determina a prioridade de processamento dos pacotes de saída.

Antes de ativar o ALTQ, consulte altq(4) para determinar se os drivers das placas de rede instaladas no sistema suportam isto.

ALTQ não está disponível como um módulo de kernel carregável. Se as interfaces do sistema suportarem ALTQ, crie um kernel personalizado usando as instruções em Configurando o kernel do FreeBSD. As seguintes opções do kernel estão disponíveis. O primeira é necessária para ativar o ALTQ. Pelo menos uma das outras opções é necessária para especificar o algoritmo do scheduler de enfileiramento:

```
options         ALTQ
options         ALTQ_CBQ        # Class Based Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Scheduler (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
```

Os seguintes algoritmos de agendamento estão disponíveis:

CBQ

Class Based Queuing (CBQ) é usado para dividir a largura de banda de uma conexão em diferentes classes ou filas para priorizar o tráfego com base nas regras de filtragem.

RED

Random Early Detection (RED) é usado para evitar o congestionamento da rede, medindo o comprimento da fila e comparando-a com os limites mínimo e máximo da fila. Quando a fila está acima do máximo, todos os novos pacotes são descartados aleatoriamente.

RIO

No modo Random Early Detection In and Out (RIO), RED mantém vários comprimentos médios de fila e vários valores limite, um para cada nível QOS.

HFSC

Hierarchical Fair Service Curve Packet Scheduler (HFSC) é descrito em http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html.

PRIQ

Priority Queuing (PRIQ) sempre passa primeiro o tráfego que está em uma fila mais alta.

Maiores informações sobre os algoritmos de agendamento e os conjuntos de regras de exemplo estão disponíveis no arquivo web do OpenBSD.

30.4. IPFW

O IPFW é um firewall stateful para o FreeBSD, que suporta tanto o IPv4 como o IPv6. Ele é composto de vários componentes: o processador de regras de filtro de firewall do kernel e seu recurso integrado de contabilidade de pacotes, o recurso de registro em log, NAT, o dummynet(4) traffic shaper, um recurso de forward, um recurso de bridge e uma habilidade ipstealth.

O FreeBSD fornece um conjunto de regras de exemplo em /etc/rc.firewall que define vários tipos de firewall para cenários comuns para ajudar usuários iniciantes a gerar um conjunto de regras apropriado. O IPFW fornece uma poderosa sintaxe que os usuários avançados podem usar para criar conjuntos de regras personalizados que atendam aos requisitos de segurança de um determinado ambiente.

Esta seção descreve como ativar o IPFW, fornece uma visão geral de sua sintaxe de regra e demonstra vários conjuntos de regras para cenários comuns de configuração.

30.4.1. Ativando o IPFW

O IPFW está incluído na instalação base do FreeBSD como um módulo carregável do kernel, o que significa que um kernel customizado não é necessário para ativar o IPFW.

Para aqueles usuários que desejam compilar estaticamente o suporte ao IPFW em um kernel personalizado, veja Opções do Kerne para o IPFW.

Para configurar o sistema para ativar o IPFW no momento da inicialização, adicione `firewall_enable="YES"` ao /etc/rc.conf:

```
# sysrc firewall_enable="YES"
```

Para usar um dos tipos de firewall padrão fornecidos pelo FreeBSD, adicione outra linha que especifique o tipo:

```
# sysrc firewall_type="open"
```

Os tipos disponíveis são:

- `open`: passa todo o tráfego.
- `client`: protege apenas esta máquina.
- `simple`: protege toda a rede.
- `closed`: desativa completamente o tráfego IP, exceto na interface de loopback.
- `workstation`: protege apenas esta máquina usando regras stateful.
- `UNKNOWN`: desativa o carregamento de regras de firewall.
- filename: caminho completo do arquivo que contém o conjunto de regras do firewall.

Se `firewall_type` estiver definido como `client` ou `simple`, modifique as regras padrão encontradas

em /etc/rc.firewall para se adequar a configuração do sistema.

Observe que o tipo `filename` é usado para carregar um conjunto de regras customizado.

Uma maneira alternativa de carregar um conjunto de regras personalizado é definir a variável `firewall_script` para o caminho absoluto de um *script executável* que inclui comandos IPFW. Os exemplos usados nesta seção assumem que o `firewall_script` está definido como /etc/ipfw.rules:

```
# sysrc firewall_script="/etc/ipfw.rules"
```

Para habilitar o registro em log por meio do syslogd(8), inclua esta linha:

```
# sysrc firewall_logging="YES"
```



Somente regras de firewall com opção de `log` vão ser registradas. As regras padrão não contém essa opção e deve ser adicionada manualmente. Por isso é avisado que o conjunto de regras padrão é editado para logar. Em adição a isso, rotacionamento de log é desejado se os logs estiverem em um arquivo separado.

Não existe uma variável em /etc/rc.conf para definir os limites de log. Para limitar o número de vezes que uma regra é registrada por tentativa de conexão, especifique o número usando esta linha no /etc/sysctl.conf:

```
# echo "net.inet.ip.fw.verbose_limit=5" >> /etc/sysctl.conf
```

Para habilitar o registro através de uma interface dedicada chamada `ipfw0`, adicione esta linha ao /etc/rc.conf em vez disso:

```
# sysrc firewall_logif="YES"
```

Em seguida, use o tcpdump para ver o que está sendo registrado:

```
# tcpdump -t -n -i ipfw0
```



Não há sobrecarga devido ao log, a menos que o tcpdump esteja anexado.

Depois de salvar as edições necessárias, inicie o firewall. Para ativar os limites de log agora, defina também o valor `sysctl` especificado acima:

```
# service ipfw start
# sysctl net.inet.ip.fw.verbose_limit=5
```

30.4.2. Sintaxe de Regras IPFW

Quando um pacote entra no firewall IPFW, ele é comparado com a primeira regra no conjunto de regras e avança uma regra por vez, movendo-se de cima para baixo em sequência. Quando o pacote corresponde aos parâmetros de seleção de uma regra, a ação da regra é executada e a pesquisa do conjunto de regras termina para esse pacote. Isto é conhecido como "primeira combinação vence". Se o pacote não corresponder a nenhuma das regras, ele será pego pela regra padrão obrigatória IPFW de número 65535, que bloqueia todos os pacotes e os descarta silenciosamente. No entanto, se o pacote corresponder a uma regra que contenha as palavras-chave `count`, `skipto` ou `tee`, a pesquisa continuará. Consulte ipfw(8) para obter detalhes sobre como essas palavras-chave afetam o processamento de regras.

Ao criar uma regra IPFW, as palavras-chave devem ser escritas na seguinte ordem. Algumas palavras-chave são obrigatórias, enquanto outras são opcionais. As palavras mostradas em maiúsculas representam uma variável e as palavras mostradas em minúsculas devem preceder a variável que a segue. O símbolo `#` é usado para marcar o início de um comentário e pode aparecer no final de uma regra ou em sua própria linha. Linhas em branco são ignoradas.

CMD RULE_NUMBER set SET_NUMBER ACTION log LOG_AMOUNT PROTO from SRC SRC_PORT to DST DST_PORT OPTIONS

Esta seção fornece uma visão geral dessas palavras-chave e suas opções. Não é uma lista exaustiva de todas as opções possíveis. Consulte ipfw(8) para obter uma descrição completa da sintaxe de regra que pode ser usada ao criar regras IPFW.

CMD

Toda regra deve começar com ipfw add.

RULE_NUMBER

Cada regra é associada a um número de `1` até `65534`. O número é usado para indicar a ordem do processamento da regra. Várias regras podem ter o mesmo número e, nesse caso, elas são aplicadas de acordo com a ordem em que foram adicionadas.

SET_NUMBER

Cada regra é associada a um número definido de `0` até `31`. Os conjuntos podem ser desativados ou ativados individualmente, possibilitando adicionar ou excluir rapidamente um conjunto de regras. Se um SET_NUMBER não for especificado, a regra será adicionada no conjunto `0`.

ACTION

Uma regra pode ser associada a uma das ações a seguir. A ação especificada será executada quando o pacote corresponder ao critério de seleção da regra.

allow | accept | pass | permit: essas palavras-chave são equivalentes e permitem pacotes que correspondem à regra.

check-state: verifica o pacote em relação à tabela de estados dinâmicos. Se uma correspondência for encontrada, execute a ação associada à regra que gerou essa regra dinâmica, caso contrário, vá para a próxima regra. Uma regra `check-state` não possui critério de seleção. Se nenhuma regra `check-state` estiver presente no conjunto de regras, a tabela de regras dinâmicas será

verificada na primeira regra `keep-state` ou `limit`.

count: atualiza os contadores de todos os pacotes que correspondem à regra. A pesquisa continua com a próxima regra.

deny | drop: qualquer das duas palavras descarta silenciosamente os pacotes que correspondem a essa regra.

Ações adicionais estão disponíveis. Consulte ipfw(8) para detalhes.

LOG_AMOUNT

Quando um pacote corresponde a uma regra com a palavra-chave `log`, uma mensagem será registrada no syslogd(8) com nome `SECURITY`. O registro somente ocorre se o número de pacotes registrados para essa regra específica não exceder um LOG_AMOUNT especificado. Se nenhum LOG_AMOUNT for especificado, o limite será retirado do valor de `net.inet.ip.fw.verbose_limit`. Um valor de zero remove o limite de registro. Quando o limite for atingido, o registro em log poderá ser reativado, limpando o contador de registro ou o contador de pacotes para essa regra, usando `ipfw resetlog`.



O registro é feito depois que todas as outras condições de correspondência de pacote foram atendidas e antes de executar a ação final no pacote. O administrador decide quais regras habilitar o log.

PROTO

Este valor opcional pode ser usado para especificar qualquer nome ou número de protocolo encontrado no arquivo /etc/protocols.

SRC

A palavra-chave `from` deve ser seguida pelo endereço de origem ou por uma palavra-chave que represente o endereço de origem. Um endereço pode ser representado por `any`, `me` (qualquer endereço configurado em uma interface neste sistema), `me6`, (qualquer endereço IPv6 configurado em uma interface neste sistema), ou `table` seguido pelo número de uma tabela de consulta que contém uma lista de endereços. Ao especificar um endereço IP, ele pode ser seguido opcionalmente pela máscara ou pela máscara de sub-rede do CIDR. Por exemplo, `1.2.3.4/25` ou `1.2.3.4:255.255.255.128`.

SRC_PORT

Uma porta de origem opcional pode ser especificada usando o número da porta ou um nome de /etc/services.

DST

A palavra-chave `to` deve ser seguida pelo endereço de destino ou por uma palavra-chave que represente o endereço de destino. As mesmas palavras-chave e endereços descritos na seção SRC podem ser usados para descrever o destino.

DST_PORT

Uma porta de destino opcional pode ser especificada usando o número da porta ou um nome de /etc/services.

OPTIONS

Várias palavras-chave podem seguir a origem e o destino. Como o nome sugere, OPTIONS são opcionais. As opções comumente usadas incluem `in` ou `out`, que especificam a direção do fluxo de pacotes, `icmptypes` seguido pelo tipo de mensagem ICMP e `keep-state`.

Quando uma regra keep-state é correspondida, o firewall criará uma regra dinâmica que corresponda ao tráfego bidirecional entre os endereços e portas de origem e destino usando o mesmo protocolo.

O recurso de regras dinâmicas é vulnerável ao esgotamento de recursos de um ataque SYN-flood, o que abriria um grande número de regras dinâmicas. Para combater esse tipo de ataque com IPFW, use `limit`. Esta opção limita o número de sessões simultâneas verificando as regras dinâmicas abertas, contando o número de vezes que esta regra e a combinação de endereços IP ocorreram. Se essa contagem for maior que o valor especificado por `limit`, o pacote será descartado.

Dezenas de OPTIONS estão disponíveis. Consulte ipfw(8) para obter uma descrição de cada opção disponível.

30.4.3. Exemplo de Conjunto de Regras

Esta seção demonstra como criar um exemplo de script de conjunto de regras de firewall stateful chamado /etc/ipfw.rules. Neste exemplo, todas as regras de conexão usam `in` ou `out` para esclarecer a direção. Eles também usam `via` *nome-da-interface* para especificar a interface que o pacote está percorrendo.



Ao criar ou testar um conjunto de regras de firewall, considere esta configuração temporária:

```
net.inet.ip.fw.default_to_accept="1"
```

Isso define a política padrão do ipfw(8) para ser mais permissiva do que o padrão `deny ip from any to any`, tornando um pouco mais difícil ficar bloqueado fora do sistema logo após a reinicialização.

O script de firewall começa indicando que é um script Bourne shell e limpa quaisquer regras existentes. Em seguida, ele cria a variável `cmd` para que `ipfw add` não precise ser digitado no início de cada regra. Ele também define a variável `pif` que representa o nome da interface que está conectada à Internet.

```
#!/bin/sh
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
pif="dc0"     # interface name of NIC attached to Internet
```

As duas primeiras regras permitem todo o tráfego na interface interna e na interface de loopback:

```
# Change xl0 to LAN NIC interface name
$cmd 00005 allow all from any to any via xl0

# No restrictions on Loopback Interface
$cmd 00010 allow all from any to any via lo0
```

A próxima regra permite que o pacote passe se corresponder a uma entrada existente na tabela de regras dinâmicas:

```
$cmd 00101 check-state
```

O próximo conjunto de regras define quais conexões stateful os sistemas internos podem criar para hosts na Internet:

```
# Allow access to public DNS
# Replace x.x.x.x with the IP address of a public DNS server
# and repeat for each DNS server in /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Allow access to ISP's DHCP server for cable/DSL configurations.
# Use the first rule and check log for IP address.
# Then, uncomment the second rule, input the IP address, and delete the first rule
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Allow outbound HTTP and HTTPS connections
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state

# Allow outbound email connections
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state

# Allow outbound ping
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Allow outbound NTP
$cmd 00260 allow udp from any to any 123 out via $pif keep-state

# Allow outbound SSH
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state

# deny and log all other outbound connections
$cmd 00299 deny log all from any to any out via $pif
```

O próximo conjunto de regras controla conexões de hosts da Internet para a rede interna. Ele começa negando pacotes tipicamente associados a ataques e, em seguida, permite explicitamente tipos específicos de conexões. Todos os serviços autorizados originados da Internet usam `limit` para evitar ataques de flood.

```
# Deny all inbound traffic from non-routable reserved address spaces
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif     #RFC 1918 private IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif      #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif         #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif          #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif     #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif    #Sun cluster
interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif        #Class D & E multicast

# Deny public pings
$cmd 00310 deny icmp from any to any in via $pif

# Deny ident
$cmd 00315 deny tcp from any to any 113 in via $pif

# Deny all Netbios services.
$cmd 00320 deny tcp from any to any 137 in via $pif
$cmd 00321 deny tcp from any to any 138 in via $pif
$cmd 00322 deny tcp from any to any 139 in via $pif
$cmd 00323 deny tcp from any to any 81 in via $pif

# Deny fragments
$cmd 00330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
$cmd 00332 deny tcp from any to any established in via $pif

# Allow traffic from ISP's DHCP server.
# Replace x.x.x.x with the same IP address used in rule 00120.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Allow HTTP connections to internal web server
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow inbound SSH connections
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Reject and log all other incoming connections
$cmd 00499 deny log all from any to any in via $pif
```

A última regra registra todos os pacotes que não correspondem a nenhuma das regras do conjunto

de regras:

```
# Everything else is denied and logged
$cmd 00999 deny log all from any to any
```

30.4.4. NAT no Kernel

O firewall IPFW do FreeBSD possui duas implementações de NAT: a implementação do sistema base natd(8) e a implementação de NAT interno do IPFW. Ambos trabalham em conjunto com o IPFW para fornecer tradução de endereço de rede. Isso pode ser usado para fornecer uma solução de compartilhamento de conexão com a Internet, para que vários computadores internos possam se conectar à Internet usando um único endereço IP público.

Para isso, a maquina FreeBSD conectada na internet deve atuar como um gateway. Esse sistema deve ter duas NICs, onde uma é conectada a internet e a outra conectada a LAN interna. Cada maquina conectada com a LAN deve estar associada a um endereço IP no espaço de rede privado, como definido pela RFC 1918.

Algumas configuração adicionais são necessárias para ativar a funcionalidade in-kernel NAT do IPFW. Para ativar o suporte ao in-kernel NAT no momento da inicialização do sistema, o seguinte deve ser definido em /etc/rc.conf:

```
gateway_enable="YES"
firewall_enable="YES"
firewall_nat_enable="YES"
```



Quando `firewall_nat_enable` estiver definido, mas `firewall_enable` não estiver, ele não terá efeito e não fará nada. Isso ocorre porque a implementação do in-kernel NAT é compatível apenas com o IPFW.

Quando o conjunto de regras contém regras stateful, o posicionamento da regra NAT é crítico e a ação `skipto` é usada. A ação `skipto` requer um número de regra para que ele saiba para qual regra saltar. O exemplo abaixo se baseia no conjunto de regras do firewall mostrado na seção anterior. Ele adiciona algumas entradas adicionais e modifica algumas regras existentes para configurar o firewall com in-kernel NAT. Ele começa adicionando algumas variáveis adicionais que representam o número da regra para pular para, a opção `keep-state` e uma lista de portas TCP que serão usadas para reduzir o número de regras.

```
#!/bin/sh
ipfw -q -f flush
cmd="ipfw -q add"
skip="skipto 1000"
pif=dc0
ks="keep-state"
good_tcpo="22,25,37,53,80,443,110"
```

Com o in-kernel NAT é necessário desativar o descarregamento da segmentação TCP (TSO) devido à arquitetura do libalias(3), uma biblioteca implementada como um módulo do kernel para fornecer o in-kernel NAT do IPFW. O TSO pode ser desativado em uma interface de rede usando ifconfig(8) ou em todo o sistema usando sysctl(8). Para desativar o TSO em todo o sistema, deve-se definir o seguinte em /etc/sysctl.conf:

```
net.inet.tcp.tso="0"
```

Uma instância NAT também será configurada. É possível ter várias instâncias de NAT, cada uma com sua própria configuração. Para este exemplo, apenas uma instância NAT é necessária; Instância NAT número 1. A configuração pode receber algumas opções, como: `if`, que indica a interface pública, `same_ports`, que cuida para que as portas mapeadas e o números das portas locais sejam mapeados da mesma maneira, `unreg_only` resultará em apenas espaços de endereço não registrados (privados) a serem processados pela instância NAT e `reset`, que ajudará a manter uma instância NAT em funcionamento, mesmo quando o endereço de IP público da máquina IPFW for alterado. Para todas as opções possíveis que podem ser passadas para uma única configuração de instância NAT, consulte ipfw(8). Ao configurar um firewall NAT stateful, é necessário permitir que pacotes traduzidos sejam reinjetados no firewall para processamento subsequente. Isso pode ser obtido desativando o comportamento `one_pass` no início do script do firewall.

```
ipfw disable one_pass
ipfw -q nat 1 config if $pif same_ports unreg_only reset
```

A regra NAT de entrada é inserida *após* as duas regras que permitem todo o tráfego nas interfaces interna e de loopback e após a regra de remontagem, mas *antes* da regra `check-state` . É importante que o número da regra selecionada para esta regra NAT, neste exemplo `100`, seja maior que as três primeiras regras e menor que a regra `check-state`. Além disso, devido ao comportamento do in-kernel NAT, é recomendável colocar uma regra de remontagem pouco antes da primeira regra NAT e depois das regras que permitem tráfego nas interfaces. Normalmente, a fragmentação IP não deve ocorrer, mas ao lidar com o tráfego de tunelamento com IPSEC/ESP/GRE, isso pode ocorrer e a recomposição de fragmentos é necessária antes de entregar o pacote completo para o mecanismo de in-kernel NAT.



A regra de remontagem não era necessária com o natd(8) do sistema base porque o recurso interno de `divert` no IPFW já cuida disso, remontando os pacotes antes da entrega no socket, também informado em ipfw(8).

A instância NAT e o número da regra usados neste exemplo não coincidem com a instância NAT e o número da regra padrão criados por rc.firewall. rc.firewall é um script que configura as regras de firewall padrão presentes no FreeBSD.

```
$cmd 005 allow all from any to any via xl0  # exclude LAN traffic
$cmd 010 allow all from any to any via lo0  # exclude loopback traffic
$cmd 099 reass all from any to any in       # reassemble inbound packets
$cmd 100 nat 1 ip from any to any in via $pif # NAT any inbound packets
# Allow the packet through if it has an existing entry in the dynamic rules table
```

```
$cmd 101 check-state
```

As regras de saída são modificadas para substituir a ação `allow` com a variável `$skip`, indicando que o processamento da regra continuará na regra `1000`. As sete regras `tcp` foram substituídas pela regra `125` porque a variável `$good_tcpo` contém as sete portas de saída permitidas.



Lembre-se de que o desempenho do IPFW é amplamente determinado pelo número de regras presentes no conjunto de regras.

```
# Authorized outbound packets
$cmd 120 $skip udp from any to x.x.x.x 53 out via $pif $ks
$cmd 121 $skip udp from any to x.x.x.x 67 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
```

As regras de entrada permanecem as mesmas, exceto a ultima regra que remove `via $pif` com intenção de casar com ambas regras de entrada e saida. A regra de NAT deve seguir essa ultima regra de saida, deve ter um numero maior que a ultima regra, e o numero da regra deve referenciar a ação `skipto`. Nesse conjunto de regras, o numero de regra `1000` lida com a passagem de todos os pacotes para nossa instância configurada para processamento NAT. A próxima regra permite que qualquer pacote submetido ao processamento NAT seja liberado.

```
$cmd 999 deny log all from any to any
$cmd 1000 nat 1 ip from any to any out via $pif # skipto location for outbound
stateful rules
$cmd 1001 allow ip from any to any
```

Neste exemplo, as regras `100`, `101`, `125`, `1000` e `1001` controlam a tradução de endereços dos pacotes de saída e de entrada para que as entradas na tabela de estado dinâmico sempre registrem o endereço de IP privado da LAN .

Considere um navegador Web interno que inicialize uma nova sessão HTTP pela porta 80. Quando o primeiro pacote de saída entra no firewall, ele não corresponde à regra `100` porque ele está saindo e não entrando. Ele pula a regra `101` porque este é o primeiro pacote e ainda não foi inserido na tabela de estados dinâmicos. O pacote finalmente corresponde à regra `125` pois é uma conexão de saída em uma porta permitida e tem um endereço IP de origem da LAN interna. Ao combinar essa regra, duas ações ocorrem. Primeiro, a ação `keep-state` adiciona uma entrada à tabela de estados dinâmicos e a ação especificada, `skipto rule 1000`, é executada. Em seguida, o pacote passa pelo NAT e é enviado para a Internet. Este pacote faz o seu caminho para o servidor web de destino, onde um pacote de resposta é gerado e enviado de volta. Este novo pacote entra no topo do conjunto de regras. Ele corresponde à regra `100` e tem seu endereço de destino IP mapeado de volta para o endereço interno original. Em seguida, ele é processado pela regra `check-state`, é encontrado na tabela como uma sessão existente e é liberado para a LAN.

No lado da entrada, o conjunto de regras deve negar pacotes inválidos e permitir apenas serviços autorizados. Um pacote que corresponde a uma regra de entrada é postado na tabela de estados

dinâmicos e o pacote é liberado para a LAN. O pacote gerado como resposta é reconhecido pela regra `check-state` como pertencente a uma sessão existente. Em seguida, ele é enviado para a regra `1000` para passar pelo NAT antes de ser liberado para a interface de saída.



A transição do natd(8) do sistema base para o in-kernel NAT pode parecer fácil no início, mas há algumas particularidades. Ao usar o kernel GENERIC, o IPFW carregará o módulo libalias.ko do kernel, quando o `firewall_nat_enable` estiver ativado no rc.conf. O módulo do kernel libalias.ko fornece apenas a funcionalidade básica de NAT, enquanto a implementação do natd(8) do sistema base possui todas as funcionalidades de NAT disponível na userland sem nenhuma configuração extra. Toda funcionalidade refere-se aos seguintes módulos do kernel que podem ser carregados adicionalmente quando necessário, além do módulo do kernel padrão libalias.ko: alias_cuseeme.ko, alias_ftp.ko, alias_bbt.ko, skinny.ko, irc.ko, alias_pptp.ko and alias_smedia.ko usando a diretiva `kld_list` em rc.conf. Se um kernel personalizado for usado, a funcionalidade completa do sistema base poderá ser compilada no kernel, usando a opção `options LIBALIAS`.

30.4.4.1. Redirecionamento de Portas

A desvantagem com NAT em geral é que os clientes da LAN não estão acessíveis na Internet. Os clientes na LAN podem fazer conexões de saída para o mundo, mas não podem receber conexões diretas. Isso é um problema ao tentar executar serviços de Internet em uma das máquinas clientes da LAN. Uma forma simples de contornar isso é redirecionar as portas selecionadas da Internet na máquina NAT para um cliente da LAN.

Por exemplo, um servidor IRC é executado no cliente `A` e um servidor Web é executado no cliente `B`. Para que isso funcione corretamente, as conexões recebidas nas portas 6667 (IRC) e 80 (HTTP) devem ser redirecionadas para as respectivas máquinas.

Com o in-kernel NAT, toda a configuração é feita na configuração da instância NAT. Para obter uma lista completa de opções que uma instância in-kernel NAT pode usar, consulte ipfw(8). A sintaxe IPFW segue a sintaxe do natd. A sintaxe para `redirect_port` é a seguinte:

```
redirect_port proto targetIP:targetPORT[-targetPORT]
  [aliasIP:]aliasPORT[-aliasPORT]
  [remoteIP[:remotePORT[-remotePORT]]]
```

Para configurar o exemplo de instalação acima, os argumentos devem ser:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

Depois de adicionar esses argumentos à configuração da instância 1 de NAT no conjunto de regras acima, as portas TCP serão encaminhadas para as máquinas clientes da LAN que rodam os serviços IRC e HTTP.

```
ipfw -q nat 1 config if $pif same_ports unreg_only reset \
  redirect_port tcp 192.168.0.2:6667 6667 \
  redirect_port tcp 192.168.0.3:80 80
```

Intervalos de portas podem ser indicados com `redirect_port`. Por exemplo, *tcp 192.168.0.2:2000-3000 2000-3000* redirecionaria todas as conexões recebidas entre as portas 2000 e 3000 para as portas 2000 a 3000 no cliente `A`.

30.4.4.2. Redirecionamento de Endereços

Redirecionamento de endereços é útil se mais de um endereço IP estiver disponível. Cada cliente da LAN pode receber seu próprio endereço IP externo pelo ipfw(8), que reescreverá os pacotes de saída dos clientes da LAN com o endereço IP externo apropriado e redirecionará todo o tráfego recebido naquele endereço IP específico de volta para o cliente da LAN específico. Isso também é conhecido como NAT estático. Por exemplo, se o endereço IP `128.1.1.1`, `128.1.1.2`, e `128.1.1.3` estiverem disponíveis, `128.1.1.1` pode ser usado pelo ipfw(8) como o endereço IP de saída externa, enquanto `128.1.1.2` e `128.1.1.3` são encaminhados de volta para os clientes da LAN `A` e `B`.

A sintaxe `redirect_address` é a seguinte, onde `localIP` é o endereço IP interno do cliente da LAN e `publicIP` é o endereço IP externo que corresponde ao cliente da LAN .

```
redirect_address localIP publicIP
```

No exemplo, os argumentos seriam:

```
redirect_address 192.168.0.2 128.1.1.2
redirect_address 192.168.0.3 128.1.1.3
```

Como o `redirect_port`, esses argumentos são inseridos na configuração da instância NAT. Com o redirecionamento de endereço, não há necessidade de redirecionamento de porta, pois todos os dados recebidos em um determinado endereço IP são redirecionados.

Os endereços IP externos na máquina ipfw(8) devem estar ativos e com alias na interface externa. Consulte rc.conf(5) para mais informações.

30.4.4.3. NAT do espaço do usuário

Vamos começar com uma declaração: a implementação de NAT do sistema base: natd(8), tem mais sobrecarga do que no in-kernel NAT. Para que o natd(8) traduza pacotes, os pacotes precisam ser copiados do kernel para o espaço do usuário e vice-versa, o que gera uma sobrecarga extra que não está presente com o in-kernel NAT.

Para ativar o daemon de NAT do sistema base , natd(8), no momento da inicialização do sistema, é necessário a seguinte configuração mínima em /etc/rc.conf. Onde `natd_interface` é definido com o nome da interface NIC conectada à Internet. O script rc(8) do natd(8) verifica automaticamente se um endereço IP dinâmico é usado e configura-se para lidar com isso.

```
gateway_enable="YES"
natd_enable="YES"
natd_interface="rl0"
```

Em geral, o conjunto de regras acima, conforme explicado para o in-kernel NAT, também pode ser usado junto com natd(8). As exceções são a configuração da instância in-kernel NAT (`ipfw -q nat 1 config ⋯`) que não é necessária junto com a regra de remontagem 99 porque sua funcionalidade é incluída na ação `divert`. As regras número 100 e 1000 terão que mudar ligeiramente, como mostrado abaixo.

```
$cmd 100 divert natd ip from any to any in via $pif
$cmd 1000 divert natd ip from any to any out via $pif
```

Para configurar o redirecionamento de porta ou endereço, é usada uma sintaxe semelhante à do in-kernel NAT. Embora agora, em vez de especificar a configuração em nosso script de conjunto de regras, como no in-kernel NAT, a configuração do natd(8) é melhor realizada em um arquivo de configuração. Para fazer isso, uma flag extra deve ser passado através do /etc/rc.conf, que especifica o caminho do arquivo de configuração.

```
natd_flags="-f /etc/natd.conf"
```



O arquivo especificado deve conter uma lista de opções de configuração, uma por linha. Para obter mais informações sobre esse arquivo de configuração e possíveis variáveis, consulte natd(8). Abaixo estão dois exemplos de valores, um por linha:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_address 192.168.0.3 128.1.1.3
```

30.4.5. O Comando IPFW

O `ipfw` pode ser usado para adicionar ou excluir regras únicas e manuais ao firewall ativo enquanto ele estiver em execução. O problema com o uso desse método é que todas as alterações são perdidas quando o sistema é reinicializado. Recomenda-se, em vez disso, gravar todas as regras em um arquivo e usar esse arquivo para carregar as regras no momento da inicialização e substituir as regras de firewall em execução no momento em que o arquivo for alterado.

O `ipfw` é uma maneira útil para se exibir as regras de firewall em execução na tela do console. O recurso de contabilidade IPFW cria dinamicamente um contador para cada regra que case com cada pacote que corresponde à regra. Durante o processo de teste de uma regra, listar a regra com seu contador é uma maneira de determinar se a regra está funcionando conforme o esperado.

Para listar todas as regras em execução em sequência:

```
# ipfw list
```

Para listar todas as regras em execução com um registro de data e hora de quando a última vez em que a regra foi utilizada:

```
# ipfw -t list
```

O próximo exemplo lista as informações contábeis e a contagem de pacotes das regras correspondentes, junto com as próprias regras. A primeira coluna é o número da regra, seguido pelo número de pacotes e bytes correspondidos, seguidos pela própria regra.

```
# ipfw -a list
```

Para listar regras dinâmicas além das regras estáticas:

```
# ipfw -d list
```

Para mostrar também as regras dinâmicas expiradas:

```
# ipfw -d -e list
```

Para zerar os contadores:

```
# ipfw zero
```

Para zerar os contadores apenas para a regra com o número *NUM*:

```
# ipfw zero NUM
```

30.4.5.1. Mensagens de Log do Firewall

Mesmo com o recurso de geração de log ativado, o IPFW não irá gerar nenhum log de regras por conta própria. O administrador do firewall decide quais regras no conjunto de regras serão logadas e adiciona a palavra-chave `log` a essas regras. Normalmente, apenas as regras de bloqueio são logadas. É costume duplicar a regra "ipfw default deny everything" com a palavra-chave `log` incluída como a última regra no conjunto de regras. Dessa forma, é possível ver todos os pacotes que não correspondem a nenhuma das regras do conjunto de regras.

O log é uma espada de dois gumes. Se não houver cuidado, uma abundância de dados de log ou um ataque DoS pode encher o disco com arquivos de log. As mensagens de log não são gravadas apenas no syslogd, mas também são exibidas na tela do console do root e logo se tornam irritantes.

A opção do kernel `IPFIREWALL_VERBOSE_LIMIT=5` limita o número de mensagens consecutivas enviadas para o syslogd(8), referente à correspondência de pacotes de uma regra dada. Quando esta opção está ativada no kernel, o número de mensagens consecutivas relativas a uma regra específica é limitado ao número especificado. Não há nada a ganhar com 200 mensagens de log idênticas. Com essa opção definida como cinco, cinco mensagens consecutivas referentes a uma regra específica seriam registradas no syslogd e as mensagens consecutivas idênticas restantes seriam contadas e postadas no syslogd com uma frase assim:

```
last message repeated 45 times
```

Todas os pacotes logados são escritos por padrão no arquivo /var/log/security, que é definido no /etc/syslog.conf.

30.4.5.2. Criando um Script de Regras

Os usuários mais experientes do IPFW criam um arquivo contendo as regras e as codificam de maneira compatível com sua execução como um script. A principal vantagem de fazer isso é que as regras de firewall podem ser atualizadas em massa sem a necessidade de reinicializar o sistema para ativá-las. Este método é conveniente para testar novas regras, pois o procedimento pode ser executado quantas vezes forem necessárias. Sendo um script, a substituição simbólica pode ser usada para valores usados frequentemente para serem substituídos em várias regras.

Este script de exemplo tem a sintaxe compatível com shells sh(1), csh(1), e tcsh(1). Campos de substituição simbólicos são prefixados com um sinal de dólar ($). Campos simbólicos não possuem o prefixo $. O valor para preencher o campo simbólico deve ser colocado entre aspas duplas ("").

Inicie o arquivo de regras assim:

```
############### start of example ipfw rules script #############
#
ipfw -q -f flush       # Delete all rules
# Set defaults
oif="tun0"             # out interface
odns="192.0.2.11"      # ISP's DNS server IP address
cmd="ipfw -q add "     # build rule prefix
ks="keep-state"        # just too lazy to key this each time
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
################### End of example ipfw rules script ############
```

As regras não são importantes, pois o foco deste exemplo é como os campos de substituição simbólica são preenchidos.

Se o exemplo acima estiver no arquivo /etc/ipfw.rules, as regras podem ser recarregadas pelo

seguinte comando:

```
# sh /etc/ipfw.rules
```

/etc/ipfw.rules pode estar localizado em qualquer lugar e o arquivo pode ter qualquer nome.

A mesma coisa pode ser realizada executando esses comandos manualmente:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

30.4.6. Opções do Kerne para o IPFW

Para compilar estaticamente o suporte ao IPFW em um kernel personalizado, consulte as instruções em Configurando o kernel do FreeBSD. As seguintes opções estão disponíveis para o arquivo de configuração do kernel personalizado:

```
options    IPFIREWALL           # enables IPFW
options    IPFIREWALL_VERBOSE       # enables logging for rules with log keyword to
syslogd(8)
options    IPFIREWALL_VERBOSE_LIMIT=5   # limits number of logged packets per-entry
options    IPFIREWALL_DEFAULT_TO_ACCEPT # sets default policy to pass what is not
explicitly denied
options    IPFIREWALL_NAT       # enables basic in-kernel NAT support
options    LIBALIAS         # enables full in-kernel NAT support
options    IPFIREWALL_NAT64     # enables in-kernel NAT64 support
options    IPFIREWALL_NPTV6     # enables in-kernel IPv6 NPT support
options    IPFIREWALL_PMOD      # enables protocols modification module support
options    IPDIVERT         # enables NAT through natd(8)
```



O IPFW pode ser carregado como um módulo do kernel: as opções acima são compiladas por padrão como módulos ou podem ser configuradas em tempo de execução usando parâmetros configuráveis.

30.5. IPFILTER (IPF)

O IPFILTER, também conhecido como IPF, é um firewall cross-platform de código aberto que foi portado para vários sistemas operacionais, incluindo FreeBSD, NetBSD, OpenBSD e Solaris™.

O IPFILTER é um firewall kernel-side e um mecanismo NAT que pode ser controlado e monitorado por programas da área de usuário. As regras de firewall podem ser definidas ou excluídas usando

ipf, as regras NAT podem ser definidas ou excluídas usando ipnat, estatísticas em tempo de execução para as partes do kernel IPFILTER podem ser informadas usando ipfstat, e ipmon pode ser usado para logar ações do IPFILTER nos arquivos de log do sistema.

O IPF foi originalmente escrito usando uma lógica de processamento de regra de que "a última regra que corresponder, ganha" e era utilizado apenas regras stateless. Desde então, IPF foi aprimorado para incluir as opções `quick` e `keep state`.

O FAQ IPF está em http://www.phildev.net/ipf/index.html. Um arquivo liberado para buscas da lista de discussão IPFilter está disponível em http://marc.info/?l=ipfilter.

Esta seção do Handbook foca no IPF no que se refere ao FreeBSD. Ele fornece exemplos de regras que contêm as opções `quick` e `keep state`.

30.5.1. Ativando o IPF

O IPF está incluído na instalação base do FreeBSD como um módulo carregável do kernel, o que significa que um kernel personalizado não é necessário para habilitar o IPF.

Para usuários que preferem compilar estaticamente o suporte ao IPF em um kernel personalizado, consulte as instruções em Configurando o kernel do FreeBSD. As seguintes opções do kernel estão disponíveis:

```
options IPFILTER
options IPFILTER_LOG
options IPFILTER_LOOKUP
options IPFILTER_DEFAULT_BLOCK
```

onde `options IPFILTER` ativa o suporte para o IPFILTER, `options IPFILTER_LOG` ativa o log do IPF usando o pseudo-dispositivo de log ipl para cada regra que tenha a palavra-chave `log`, `IPFILTER_LOOKUP` ativa as pools IP para acelerar IP lookups, e `options IPFILTER_DEFAULT_BLOCK` altera o comportamento padrão para que qualquer pacote que não corresponda a uma regra `pass` do firewall seja bloqueado.

Para configurar o sistema para ativar o IPF no momento da inicialização, adicione as seguintes entradas ao /etc/rc.conf. Essas entradas também ativarão o log e o `default pass all`. Para alterar a política padrão para `block all` sem compilar um kernel personalizado, lembre-se de adicionar uma regra `block all` no final do conjunto de regras.

```
ipfilter_enable="YES"             # Start ipf firewall
ipfilter_rules="/etc/ipf.rules"   # loads rules definition text file
ipv6_ipfilter_rules="/etc/ipf6.rules" # loads rules definition text file for IPv6
ipmon_enable="YES"                # Start IP monitor log
ipmon_flags="-Ds"                 # D = start as daemon
                                  # s = log to syslog
                                  # v = log tcp window, ack, seq
                                  # n = map IP & port to names
```

Se a funcionalidade NAT for necessária, adicione também estas linhas:

```
gateway_enable="YES"              # Enable as LAN gateway
ipnat_enable="YES"                # Start ipnat function
ipnat_rules="/etc/ipnat.rules"    # rules definition file for ipnat
```

Então, inicie o IPF:

```
# service ipfilter start
```

Para carregar as regras de firewall, especifique o nome do arquivo do conjunto de regras usando `ipf`. O comando a seguir pode ser usado para substituir as regras de firewall que está em execução:

```
# ipf -Fa -f /etc/ipf.rules
```

onde `-Fa` limpa todas as tabelas de regras internas e `-f` especifica o arquivo que contém as regras a serem carregadas.

Isso fornece a capacidade de fazer alterações em um conjunto de regras personalizado e atualizar o firewall em execução com uma nova cópia das regras sem precisar reinicializar o sistema. Esse método é conveniente para testar novas regras, pois o procedimento pode ser executado quantas vezes forem necessárias.

Consulte ipf(8) para detalhes sobre as outras flags disponíveis com este comando.

30.5.2. Sintaxe de Regras IPF

Esta seção descreve a sintaxe de regras IPF usada para criar regras stateful. Ao criar regras, lembre-se de que, a menos que a palavra-chave `quick` apareça em uma regra, todas as regras são lidas em ordem, com a *última regra correspondente* sendo a aplicada. Isso significa que, mesmo que a primeira regra que corresponder a um pacote seja `pass`, se houver uma regra de correspondência posterior que seja `block`, o pacote será descartado. Os conjuntos de regras de exemplo podem ser encontrados em /usr/shared/examples/ipfilter.

Ao criar regras, um caractere `#` é usado para marcar o início de um comentário e pode aparecer no final de uma regra, para explicar a função dessa regra ou em sua própria linha. Todas as linhas em branco são ignoradas.

As palavras-chave usadas nas regras devem ser escritas em uma ordem específica, da esquerda para a direita. Algumas palavras-chave são obrigatórias, enquanto outras são opcionais. Algumas palavras-chave têm sub-opções que podem ser palavras-chave e também incluem mais sub-opções. A ordem das palavras-chave é a seguinte, em que as palavras mostradas em maiúsculas representam uma variável e as palavras mostradas em minúsculas devem preceder a variável que a segue:

ACTION DIRECTION OPTIONS proto PROTO_TYPE from SRC_ADDR SRC_PORT to DST_ADDR

DST_PORT TCP_FLAG|ICMP_TYPE keep state STATE

Esta seção descreve cada uma dessas palavras-chave e suas opções. Não é uma lista exaustiva de todas as opções possíveis. Consulte ipf(5) para obter uma descrição completa da sintaxe de regra que pode ser usada ao criar regras IPF e exemplos para usar de cada palavra-chave.

ACTION

A palavra-chave action indica o que fazer com o pacote se corresponder a essa regra. Toda regra *deve* ter uma ação. As seguintes ações são reconhecidas:

`block`: descarta o pacote.

`pass`: permite o pacote.

`log`: gera um registro de log.

`count`: conta o número de pacotes e bytes que podem fornecer uma indicação da frequência com que uma regra é usada.

`auth`: enfileira o pacote para processamento adicional por outro programa.

`call`: fornece acesso a funções embutidas no IPF que permitem ações mais complexas.

`decapsulate`: remove quaisquer cabeçalhos para processar o conteúdo do pacote.

DIRECTION

Em seguida, cada regra deve indicar explicitamente a direção do tráfego usando uma dessas palavras-chave:

`in`: a regra é aplicada em um pacote de entrada.

`out`: a regra é aplicada em um pacote de saída.

`all`: a regra se aplica em qualquer direção.

Se o sistema tiver várias interfaces, a interface pode ser especificada junto com a direção. Um exemplo seria `in on fxp0`.

OPTIONS

Opções são opcionais. No entanto, se várias opções forem especificadas, elas deverão ser usadas na ordem apresentada aqui.

`log`: ao executar a ACTION especificada, o conteúdo dos cabeçalhos do pacote será gravado no pseudo-dispositivo de log ipl(4).

`quick`: se um pacote corresponder a essa regra, a ACTION especificada pela regra ocorrerá e nenhum processamento adicional das regras a seguir ocorrerá para este pacote.

`on`: deve ser seguido pelo nome da interface conforme exibido pelo ifconfig(8). A regra corresponderá somente se o pacote estiver passando pela interface especificada na direção especificada.

Ao usar a palavra-chave `log`, os seguintes qualificadores podem ser usados nesta ordem:

`body`: indica que os primeiros 128 bytes do conteúdo do pacote serão registrados após os cabeçalhos.

`first`: se a palavra-chave `log` estiver sendo usada em conjunto com uma opção `keep state`, esta opção é recomendada para que somente o pacote acionador seja logado e não todos os pacotes que corresponde à conexão stateful.

Opções adicionais estão disponíveis para especificar mensagens de retorno de erro. Consulte ipf(5) para mais detalhes.

PROTO_TYPE

O tipo de protocolo é opcional. No entanto, é obrigatório se a regra precisar especificar um SRC_PORT ou um DST_PORT, uma vez que isso requer o tipo de protocolo. Ao especificar o tipo de protocolo, use a palavra-chave `proto` seguida de um número de protocolo ou nome de /etc/protocols. Exemplos de nomes de protocolos incluem `tcp`, `udp` ou `icmp`. Se PROTO_TYPE for especificado, mas nenhum SRC_PORT ou DST_PORT for especificado, todos os números de porta desse protocolo corresponderão a essa regra.

SRC_ADDR

A palavra-chave `from` é obrigatória e é seguida por uma palavra-chave que representa a origem do pacote. A origem pode ser um nome de host, um endereço IP seguido pela máscara CIDR, um pool de endereços ou a palavra-chave `all`. Consulte ipf(5) para exemplos.

Não há como definir intervalos de endereços de IP que não se expressam facilmente usando a notação de formato numérico com ponto / máscara. O pacote ou port net-mgmt/ipcalc pode ser usado para facilitar o cálculo da máscara CIDR. Informações adicionais estão disponíveis na página web da ferramenta: http://jodies.de/ipcalc.

SRC_PORT

O número da porta da origem é opcional. No entanto, se for usado, ela exige que o PROTO_TYPE seja definido primeiramente na regra. O número da porta também deve ser precedido pela palavra-chave `proto`.

Diferentes operadores de comparação são suportados: `=` (igual a), `!=` (diferente de), `<` (menor que), `>` (maior que), ⇐ (menor ou igual a) e `>=` (maior que ou igual a).

Para especificar intervalos de porta, coloque os dois números de porta entre `<>` (menor que e maior que), `><` (maior que e menor que) ou `:` (maior que ou igual a e menor que ou igual a).

DST_ADDR

A palavra-chave `to` é obrigatória e é seguida por uma palavra-chave que representa o destino do pacote. Semelhante ao SRC_ADDR, ela pode ser um nome de host, um endereço IP seguido pela máscara CIDR, um pool de endereços ou a palavra-chave `all`.

DST_PORT

Semelhante ao SRC_PORT, o número da porta do destino é opcional. No entanto, se for usada, ela exige que o PROTO_TYPE seja definido primeiramente na regra. O número da porta também

deve ser precedido pela palavra-chave `proto`.

TCP_FLAG|ICMP_TYPE

Se `tcp` for especificado como o PROTO_TYPE, flags poderão ser especificadas como letras, onde cada letra representa uma das possíveis flags TCP utilizadas para determinar o estado de uma conexão. Os valores possíveis são: `S` (SYN), `A` (ACK), `P` (PSH), `F` (FIN), `U` (URG), `R` (RST), `C` (CWN), e `E` (ECN).

Se o `icmp` for especificado como o PROTO_TYPE, o tipo ICMP para correspondência pode ser especificado. Consulte o ipf(5) para os tipos permitidos.

STATE

Se uma regra `pass` contiver `keep state`, o IPF incluirá uma entrada em sua tabela de estados dinâmicos e permitirá o tráfego os pacotes subsequentes que correspondam à conexão. O IPF pode rastrear o estado das sessões TCP, UDP e ICMP. Qualquer pacote que o IPF tenha certeza de que faz parte de uma sessão ativa, mesmo que seja um protocolo diferente, será liberado.

No IPF, os pacotes destinados a sair pela interface conectada à Internet pública são verificados primeiro na tabela de estados dinâmicos. Se o pacote corresponder ao próximo pacote esperado, compreendendo uma sessão ativa, ele sairá do firewall e o estado do fluxo da sessão será atualizado na tabela de estados dinâmicos. Os pacotes que não pertencem a uma sessão já ativa são verificados no conjunto de regras de saída. Os pacotes vindos da interface conectada à Internet pública são verificados primeiro na tabela de estados dinâmicos. Se o pacote corresponder ao próximo pacote esperado que compreende uma sessão ativa, ele sairá do firewall e o estado do fluxo da sessão será atualizado na tabela de estados dinâmicos. Os pacotes que não pertencem a uma sessão já ativa são verificados no conjunto de regras de entrada.

Várias palavras-chave podem ser adicionadas depois de `keep state`. Se usadas, essas palavras-chave definem várias opções que controlam a filtragem stateful, como a configuração de limites de conexão ou o tempo de vida da conexão. Consulte ipf(5) para obter a lista de opções disponíveis e suas descrições.

30.5.3. Exemplo de Conjunto de Regras

Esta seção demonstra como criar um conjunto de regras de exemplo que permite apenas serviços que correspondam às regras `pass` e bloqueie todo o resto.

O FreeBSD usa a interface de loopback (lo0) e o endereço IP `127.0.0.1` para comunicação interna. O conjunto de regras do firewall deve conter regras para permitir o livre movimento desses pacotes usados internamente:

```
# no restrictions on loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
```

A interface pública conectada à Internet é usada para autorizar e controlar o acesso de todas as conexões de entrada e saída. Se uma ou mais interfaces forem cabeadas para redes privadas, essas interfaces internas poderão exigir regras para permitir que os pacotes originados da LAN fluam

entre as redes internas ou para a interface conectada à Internet. O conjunto de regras deve ser organizado em três seções principais: quaisquer interfaces internas confiáveis, conexões de saída por meio da interface pública e conexões de entrada por meio da interface pública.

Essas duas regras permitem que todo o tráfego passe por uma interface confiável LAN chamada xl0:

```
# no restrictions on inside LAN interface for private network
pass out quick on xl0 all
pass in quick on xl0 all
```

As regras para as seções de saída e entrada da interface pública devem ter as regras correspondidas com mais frequência antes das regras menos comuns, com a última regra na seção bloqueando e registrando todos os pacotes para essa interface e direção.

Este conjunto de regras define a seção de saída da interface pública denominada dc0. Essas regras mantêm o estado e identificam os serviços específicos que os sistemas internos estão autorizados para acesso público à Internet. Todas as regras usam `quick` e especificam os números de porta apropriados e, quando aplicável, os endereços de destino.

```
# interface facing Internet (outbound)
# Matches session start requests originating from or behind the
# firewall, destined for the Internet.

# Allow outbound access to public DNS servers.
# Replace x.x.x. with address listed in /etc/resolv.conf.
# Repeat for each DNS server.
pass out quick on dc0 proto tcp from any to x.x.x. port = 53 flags S keep state
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Allow access to ISP's specified DHCP server for cable or DSL networks.
# Use the first rule, then check log for the IP address of DHCP server.
# Then, uncomment the second rule, replace z.z.z.z with the IP address,
# and comment out the first rule
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state

# Allow HTTP and HTTPS
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Allow email
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Allow NTP
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Allow FTP
```

```
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Allow SSH
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Allow ping
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Block and log everything else
block out log first quick on dc0 all
```

Neste exemplo de regras na seção de entrada da interface pública todos os pacotes indesejáveis são bloqueados primeiro. Isso reduz o número de pacotes registrados pela última regra.

```
# interface facing Internet (inbound)
# Block all inbound traffic from non-routable or reserved address spaces
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 private IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 private IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 private IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #reserved for docs
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Class D & E multicast

# Block fragments and too short tcp packets
block in quick on dc0 all with frags
block in quick on dc0 proto tcp all with short

# block source routed packets
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr

# Block OS fingerprint attempts and log first occurrence
block in log first quick on dc0 proto tcp from any to any flags FUP

# Block anything with special options
block in quick on dc0 all with ipopts

# Block public pings and ident
block in quick on dc0 proto icmp all icmp-type 8
block in quick on dc0 proto tcp from any to any port = 113

# Block incoming Netbios services
block in log first quick on dc0 proto tcp/udp from any to any port = 137
block in log first quick on dc0 proto tcp/udp from any to any port = 138
block in log first quick on dc0 proto tcp/udp from any to any port = 139
block in log first quick on dc0 proto tcp/udp from any to any port = 81
```

Sempre que houver mensagens de log em uma regra com a opção `log first`, execute `ipfstat -hio` para saber quantas vezes a regra foi correspondida. Um grande número de correspondências pode indicar que o sistema está sob ataque.

O restante das regras na seção de entrada define quais conexões podem ser iniciadas a partir da Internet. A última regra nega todas as conexões que não foram explicitamente permitidas pelas regras anteriores desta seção.

```
# Allow traffic in from ISP's DHCP server. Replace z.z.z.z with
# the same IP address used in the outbound section.
pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Allow public connections to specified internal web server
pass in quick on dc0 proto tcp from any to x.x.x.x port = 80 flags S keep state

# Block and log only first occurrence of all remaining traffic.
block in log first quick on dc0 all
```

30.5.4. Configurando o NAT

Para ativar o NAT, adicione estas instruções ao arquivo /etc/rc.conf e especifique o nome do arquivo que contém as regras de NAT:

```
gateway_enable="YES"
ipnat_enable="YES"
ipnat_rules="/etc/ipnat.rules"
```

As regras de NAT são flexíveis e podem realizar muitas coisas diferentes para atender às necessidades dos usuários comerciais e domésticos. A sintaxe da regra apresentada aqui foi simplificada para demonstrar um uso comum. Para obter uma descrição completa da sintaxe da regra, consulte ipnat(5).

A sintaxe básica para uma regra NAT é a seguinte, onde `map` inicia a regra e *IF* deve ser substituído pelo nome da interface externa:

```
map IF LAN_IP_RANGE -> PUBLIC_ADDRESS
```

O *LAN_IP_RANGE* é o intervalo de endereços IP usados pelos clientes internos. Geralmente, é um intervalo de endereços privados, como `192.168.1.0/24`. O *PUBLIC_ADDRESS* pode ser o endereço IP externo estático ou a palavra-chave `0/32` que representa o endereço IP atribuído para *IF*.

No IPF, quando um pacote chega ao firewall a partir da LAN com um destino público, ele primeiro passa pelas regras de saída do conjunto de regras do firewall. Em seguida, o pacote é passado para o conjunto de regras NAT, o qual é lido de cima para baixo, onde a primeira regra correspondente ganha. O IPF testa cada regra de NAT em relação ao nome da interface e ao endereço IP de origem do pacote. Quando o nome da interface de um pacote corresponde a uma regra NAT, o endereço IP

de origem do pacote na LAN privada é verificado para ver se ele está dentro do intervalo de endereços IP especificado em *LAN_IP_RANGE*. Se corresponder, o pacote tem seu endereço IP de origem reescrito com o endereço IP público especificado por *PUBLIC_ADDRESS*. O IPF adiciona uma entrada em sua tabela NAT interna para que, quando o pacote retornar da Internet, possa ser mapeado de volta para seu endereço IP privado original antes de ser passado para as regras de firewall para processamento adicional.

Para redes que possuem um grande número de sistemas internos ou várias sub-redes, o processo de afunilar todo endereço IP em um único endereço IP público se torna um problema de recursos. Dois métodos estão disponíveis para aliviar esse problema.

O primeiro método é atribuir um intervalo de portas para usar como portas de origem. Adicionando a palavra-chave `portmap`, o NAT pode ser direcionado para usar apenas portas de origem no intervalo especificado:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000
```

Como alternativa, use a palavra-chave `auto` que informa ao NAT para determinar as portas que estão disponíveis para uso:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto
```

O segundo método é usar um pool de endereços públicos. Isso é útil quando existem muitos clientes na LAN para para usar um único endereço público e um bloco de endereços públicos de IP está disponível. Esses endereços públicos podem ser usados como um pool do qual o NAT seleciona um endereço IP à medida que o endereço de um pacote é mapeado ao sair.

O intervalo de endereços IP públicos pode ser especificado usando uma notação de netmask ou CIDR. Essas duas regras são equivalentes:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/255.255.255.0
map dc0 192.168.1.0/24 -> 204.134.75.0/24
```

Uma prática comum é ter um servidor web ou servidor de email publicamente acessível isolado a um segmento de rede interno. O tráfego desses servidores ainda precisa passar por NAT, mas o redirecionamento de porta é necessário para direcionar o tráfego de entrada para o servidor correto. Por exemplo, para mapear um servidor web usando o endereço interno `10.0.10.25` para seu endereço IP público `20.20.20.5`, use esta regra:

```
rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

Se for o único servidor web, essa regra também funcionará, pois redirecionará todas as solicitações HTTP externas para `10.0.10.25`:

```
rdr dc0 0.0.0.0/0 port 80 -> 10.0.10.25 port 80
```

O IPF possui um proxy FTP embutido que pode ser usado com o NAT. Ele monitora todo o tráfego de saída de conexões ativa ou passiva de FTP e cria dinamicamente regras de filtro temporário contendo o número de porta usado pelo canal de dados FTP. Isso elimina a necessidade de abrir grandes intervalos de portas altas para conexões de FTP.

Neste exemplo, a primeira regra chama o proxy no tráfego de saída FTP da LAN interna. A segunda regra passa o tráfego de FTP do firewall para a Internet, e a terceira regra lida com todo o tráfego não FTP da LAN interna:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
map dc0 10.0.10.0/29 -> 0/32
```

As regras `map` de FTP vem antes da regra NAT, de modo que quando um pacote corresponder a uma regra FTP, o proxy FTP crie regras temporárias de filtragem para permitir que os pacotes da sessão FTP sejam liberados e que passem pelo NAT. Todos os pacotes de rede local que não sejam FTP não corresponderão às regras de FTP, mas serão liberados pelo NAT se corresponderem à terceira regra.

Sem o proxy FTPem, as seguintes regras de firewall seriam necessárias. Note que sem o proxy, todas as portas acima de `1024` precisam ser permitidas:

```
# Allow out LAN PC client FTP to public Internet
# Active and passive modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state

# Allow out passive mode data channel high order port numbers
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state

# Active mode let data channel in from FTP server
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

Sempre que o arquivo contendo as regras de NAT for editado, execute `ipnat` com `-CF` para excluir as regras atuais de NAT e liberar o conteúdo da tabela de tradução dinâmica. Inclua `-f` e especifique o nome do conjunto de regras NAT para carregar:

```
# ipnat -CF -f /etc/ipnat.rules
```

Para exibir as estatísticas de NAT:

```
# ipnat -s
```

Para listar os mapeamentos atuais da tabela NAT:

```
# ipnat -l
```

Para ativar o modo verbose e exibir informações relacionadas ao processamento de regras, regras ativas e registros nas tabelas:

```
# ipnat -v
```

30.5.5. Visualizando Estatísticas do IPF

O IPF inclui o ipfstat(8) que pode ser usado para recuperar e exibir estatísticas das regras sendo utilizadas enquanto os pacotes passam pelo firewall. As estatísticas são acumuladas desde que o firewall foi iniciado pela última vez ou desde a última vez que foram redefinidas para zero usando `ipf -Z`.

A saída padrão do `ipfstat` é semelhante a esta:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
 output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
 input packets logged: blocked 99286 passed 0
 output packets logged: blocked 0 passed 0
 packets logged: input 0 output 0
 log failures: input 3898 output 0
 fragment state(in): kept 0 lost 0
 fragment state(out): kept 0 lost 0
 packet state(in): kept 169364 lost 0
 packet state(out): kept 431395 lost 0
 ICMP replies: 0 TCP RSTs sent: 0
 Result cache hits(in): 1215208 (out): 1098963
 IN Pullups succeeded: 2 failed: 0
 OUT Pullups succeeded: 0 failed: 0
 Fastroute successes: 0 failures: 0
 TCP cksum fails(in): 0 (out): 0
 Packet log flags set: (0)
```

Várias opções estão disponíveis. Quando executado com `-i` para entrada ou `-o` para saída, o comando recuperará e exibirá a lista apropriada de regras de filtro atualmente instaladas e em uso pelo kernel. Para também ver os números das regras, inclua `-n`. Por exemplo, `ipfstat -on` exibe a tabela de regras de saída com os números de regra:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Inclua `-h` para prefixar cada regra com uma contagem de quantas vezes a regra foi utilizada. Por exemplo, `ipfstat -oh` exibe a tabela de regras internas de saída, prefixando cada regra com sua

contagem de uso:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Para exibir a tabela de estados em um formato similar ao top(1), use `ipfstat -t`. Quando o firewall está sob ataque, essa opção fornece a capacidade de identificar e ver os pacotes de ataque. As sub-flags opcionais dão a possibilidade de selecionar o IP destino ou origem, porta ou protocolo a ser monitorado em tempo real. Consulte ipfstat(8) para detalhes.

30.5.6. Log do IPF

O IPF fornece o `ipmon`, que pode ser usado para gravar as informações de log do firewall em um formato legível por humanos. Isso requer que as opções `IPFILTER_LOG` sejam primeiramente adicionadas a um kernel personalizado usando as instruções em Configurando o kernel do FreeBSD.

Esse comando geralmente é executado no modo daemon para fornecer um arquivo de log contínuo do sistema para que o registro de eventos passados possa ser revisado. Como o FreeBSD possui um recurso syslogd(8) integrado para rotacionar automaticamente os logs do sistema, a instrução `ipmon_flags` no arquivo rc.conf por padrão utiliza `-Ds`:

```
ipmon_flags="-Ds" # D = start as daemon
                  # s = log to syslog
                  # v = log tcp window, ack, seq
                  # n = map IP & port to names
```

O registro em log fornece a capacidade de revisar, após o fato, informações como quais pacotes foram descartados, de que endereços eles vieram e para onde estavam indo. Esta informação é útil para rastrear invasores.

Uma vez que o recurso de criação de log esteja ativado no arquivo rc.conf e iniciado com o serviço `ipmon start`, o IPF irá registrar apenas as regras que contêm a palavra-chave `log`. O administrador do firewall decide quais regras no conjunto de regras devem ser logadas e normalmente apenas as regras de negação são registradas. É costume incluir a palavra-chave `log` na última regra do conjunto de regras. Isso possibilita ver todos os pacotes que não correspondem a nenhuma das regras do conjunto de regras.

Por padrão, o modo `ipmon -Ds` usa `local0` como o recurso de log. Os níveis de registro a seguir podem ser usados para separar ainda mais os dados logados:

```
LOG_INFO - pacotes logados usando a palavra-chave "log" ao invés da ação  pass ou
block.
LOG_NOTICE - pacotes logados que também são liberados
LOG_WARNING - pacotes logados que também são bloqueados
LOG_ERR - pacotes que foram logados e que podem ser considerados insuficientes devido
```

```
a um cabeçalho incompleto
```

Para configurar o IPF para logar todos os dados em /var/log/ipfilter.log, primeiro crie o arquivo vazio:

```
# touch /var/log/ipfilter.log
```

Em seguida, para gravar todas as mensagens de log no arquivo especificado, inclua a seguinte instrução no arquivo /etc/syslog.conf:

```
local0.* /var/log/ipfilter.log
```

Para ativar as alterações e instruir o syslogd(8) para ler o arquivo modificado /etc/syslog.conf, execute `service syslogd reload`.

Não se esqueça de editar o /etc/newsyslog.conf para rotacionar o novo arquivo de log.

As mensagens geradas pelo `ipmon` consistem em campos de dados separados por espaços em branco. Campos comuns a todas as mensagens são:

1. A data do recebimento do pacote.
2. O horário do recebimento do pacote. Isto está no formato HH:MM:SS.F, para horas, minutos, segundos e frações de segundo.
3. O nome da interface que processou o pacote.
4. O grupo e o número da regra no formato `@0:17`.
5. A ação: `p` para liberado (pass), `b` para bloqueado, `S` para um pacote com problema (short), `n` não corresponde a nenhuma regra e `L` para uma regra de log.
6. Os endereços escritos em três campos: o endereço de origem e porta separados por uma vírgula, o símbolo → , e o endereço e porta de destino. Por exemplo: `209.53.17.22,80 → 198.73.220.17,1722`.
7. `PR` seguido pelo nome ou número do protocolo: por exemplo, `PR tcp`.
8. `len` seguido pelo tamanho do cabeçalho e comprimento total do pacote: por exemplo, `len 20 40`.

Se o pacote for um pacote TCP, haverá um campo adicional começando com um hífen seguido por letras correspondentes a quaisquer flags que foram configuradas. Consulte ipf(5) para obter uma lista de letras e suas flags.

Se o pacote for um pacote ICMP, haverá dois campos no final: o primeiro sempre sendo "icmp" e o próximo sendo a mensagem ICMP e sub-tipo de mensagem, separados por uma barra. Por exemplo: `icmp 3/3` para uma mensagem port unreachable.

30.6. Blacklistd

O Blacklistd é um daemon que escuta sockets para receber notificações de outros daemons sobre

tentativas de conexão que falharam ou foram bem-sucedidas. É mais amplamente utilizado no bloqueio de muitas tentativas de conexão em portas abertas. Um exemplo excelente é o SSH, executado na Internet, recebendo muitas solicitações de conexão de bots ou scripts tentando adivinhar senhas e obter acesso. Utilizando blacklistd, o daemon pode notificar o firewall para criar uma regra de filtro para bloquear tentativas excessivas de conexão de uma única origem após várias tentativas. O Blacklistd foi desenvolvido pela primeira vez no NetBSD e apareceu na versão 7. O FreeBSD 11 importou o blacklistd do NetBSD.

Este capítulo descreve como instalar o blacklistd, configurá-lo e fornece exemplos de como usá-la. Os leitores devem estar familiarizados com os conceitos básicos de firewall, como regras. Para detalhes, consulte o capítulo sobre firewall. O PF é usado nos exemplos, mas outros firewalls disponíveis no FreeBSD também devem funcionar com o blacklistd.

30.6.1. Habilitando a Blacklistd

A configuração principal do blacklistd é armazenada em blacklistd.conf(5). Várias opções de linha de comando também estão disponíveis para alterar o comportamento em tempo de execução do blacklistd. Para persistir as configurações em uma reinicialização do sistema, deve se armazenar as opções em /etc/blacklistd.conf. Para ativar o daemon durante a inicialização do sistema, adicione a linha `blacklistd_enable` no /etc/rc.conf assim:

```
# sysrc blacklistd_enable=yes
```

Para iniciar o serviço manualmente, execute este comando:

```
# service blacklistd start
```

30.6.2. Criando um conjunto de regras no Blacklistd

As regras do blacklistd são configuradas em blacklistd.conf(5) com uma opção por linha. Cada regra contém uma tupla separada por espaços ou tabulações. As regras pertencem a um `local` ou a um `remote`, que se aplica à máquina em que o blacklistd está sendo executado ou a uma origem externa, respectivamente.

30.6.2.1. Regras Locais

Um exemplo de entrada blacklistd.conf para uma regra local se parece com isso:

```
[local]
ssh             stream  *       *               *       3       24h
```

Todas as regras que seguem a seção `[local]` são tratadas como regras locais (que é o padrão), aplicadas à máquina local. Quando uma seção `[remote]` é encontrada, todas as regras a seguir são tratadas como regras de máquina remota.

Sete campos definem uma regra separada por tabulações ou espaços. Os quatro primeiros campos

identificam o tráfego que deve estar na lista negra. Os três campos a seguir definem o comportamento do backlistd. Os curingas são indicados como asteriscos (*), correspondendo a qualquer coisa nesse campo. O primeiro campo define a localização. Nas regras locais, essas são as portas de rede. A sintaxe para o campo local é a seguinte:

```
[address|interface][/mask][:port]
```

Os endereços podem ser especificados como IPv4 no formato numérico ou IPv6 entre colchetes. Um nome de interface como `em0` também pode ser usado.

O tipo de socket é definido pelo segundo campo. Os socket TCP são do tipo `stream`, enquanto UDP é indicado como `dgram`. O exemplo acima usa TCP, pois o SSH está usando esse protocolo.

Um protocolo pode ser usado no terceiro campo de uma regra de lista negra. Os seguintes protocolos podem ser usados: `tcp`, `udp`, `tcp6`, `udp6` ou numérico. Um curinga, como no exemplo, geralmente é usado para corresponder a todos os protocolos, a menos que haja um motivo para distinguir o tráfego por um determinado protocolo.

No quarto campo, o usuário ou proprietário efetivo do processo daemon que está reportando o evento é definido. O nome de usuário ou o UID pode ser usado aqui, bem como um curinga (veja a regra de exemplo acima).

O nome da regra do packet filter é declarado pelo quinto campo, que inicia a parte de comportamento da regra. Por padrão, blacklistd coloca todos os blocos sob uma âncora pf chamada `blacklistd` em pf.conf assim:

```
anchor "blacklistd/*" in on $ext_if
block in
pass out
```

Para blacklists separadas, um nome de âncora pode ser usado neste campo. Em outros casos, o curinga será suficiente. Quando um nome começa com um hífen (`-`), significa que uma âncora com o nome de regra padrão precedido deve ser usada. Uma modificação do exemplo acima usando o hífen ficaria assim:

```
ssh             stream  *       *               -ssh       3       24h
```

Com essa regra, quaisquer novas regras de blacklist são adicionadas a uma âncora chamada `blacklistd-ssh`.

Para bloquear sub-redes inteiras para uma única violação de regra, um `/` no nome da regra pode ser usado. Isso faz com que a parte restante do nome seja interpretada como a máscara a ser aplicada ao endereço especificado na regra. Por exemplo, esta regra bloquearia todos os endereços adjacentes a `/24`.

```
22              stream  tcp       *               */24    3       24h
```



É importante especificar o protocolo apropriado aqui. O IPv4 e o IPv6 tratam o /24 de maneira diferente, é por isso que * não pode ser usado no terceiro campo para esta regra.

Esta regra define que, se qualquer host dessa rede estiver se comportando mal, todo o resto da rede também será bloqueado.

O sexto campo, chamado `nfail`, define o número de falhas de login necessárias para colocar na blacklist o IP remoto em questão. Quando um curinga é usado nessa posição, isso significa que o bloqueio nunca irá acontecer. Na regra de exemplo acima, um limite de três é definido, o que significa que, após três tentativas de logon no SSH em uma conexão, o IP é bloqueado.

O último campo em uma definição de regra do blacklistd especifica por quanto tempo um host ficará na lista negra. A unidade padrão é segundos, mas sufixos como `m`, `h` e `d` também podem ser especificados por minutos, horas e dias, respectivamente.

A regra de exemplo na íntegra significa que, após três vezes a autenticação no SSH, resultará em uma nova regra de bloqueio de PF para esse host. As correspondências de regras são realizadas verificando primeiro as regras locais, uma após a outra, da mais específica à menos específica. Quando ocorre uma correspondência, as regras `remote` são aplicadas e o nome `nfail` e os campos de desativação são alterados pela regra `remote` correspondente.

30.6.2.2. Regras Remotas

As regras remotas são usadas para especificar como o blacklistd muda seu comportamento, dependendo do host remoto que está sendo avaliado no momento. Cada campo em uma regra remota é o mesmo que em uma regra local. A única diferença está na maneira como o blacklistd os usa. Para explicar, esta regra de exemplo é usada:

```
[remote]
203.0.113.128/25 *      *       *               =/25    =       48h
```

O campo de endereço pode ser um endereço IP (v4 ou v6), uma porta ou ambas. Isso permite definir regras especiais para um intervalo de endereços remotos específico, como neste exemplo. Os campos para tipo, protocolo e proprietário são identicamente interpretados como na regra local.

Porém, os campos de nome são diferentes: o sinal de igual (`=`) em uma regra remota diz ao blacklistd para usar o valor da regra local correspondente. Isso significa que a entrada da regra de firewall é obtida e o prefixo `/25` (uma máscara de rede `255.255.255.128`) é adicionada. Quando uma conexão desse intervalo de endereços é colocada na lista negra, toda a sub-rede é afetada. Um nome de âncora PF também pode ser usado aqui; nesse caso, o blacklisted adicionará regras para esse bloco de endereços à âncora desse nome. A tabela padrão é usada quando um curinga é especificado.

Um número personalizado de falhas na coluna `nfail` pode ser definido para um endereço. Isso é

útil para exceções a uma regra específica, talvez para permitir a alguém uma aplicação menos rigorosa de regras ou um pouco mais de clemência nas tentativas de login. O bloqueio é desativado quando um asterisco é usado neste sexto campo.

As regras remotas permitem uma aplicação mais rigorosa dos limites das tentativas de logon, em comparação com as tentativas provenientes de uma rede local como um escritório.

30.6.3. Configuração do cliente no Blacklistd

Existem alguns pacotes de software no FreeBSD que podem utilizar a funcionalidade do blacklistd. Os dois mais proeminentes são ftpd(8) e sshd(8) para bloquear tentativas excessivas de conexão. Para ativar o blacklistd no daemon SSH, adicione a seguinte linha ao /etc/ssh/sshd_config:

```
UseBlacklist yes
```

Reinicie o sshd posteriormente para que essas alterações entrem em vigor.

A lista negra do ftpd(8) é ativada usando `-B`, em /etc/inetd.conf ou como uma flag no /etc/rc.conf assim:

```
ftpd_flags="-B"
```

Isso é tudo o que é necessário para que esses programas conversem com o blacklistd.

30.6.4. Gerenciamento do Blacklistd

O Blacklistd fornece ao usuário um utilitário de gerenciamento chamado blacklistctl(8). Ele exibe endereços e redes bloqueados que estão na lista negra pelas regras definidas em blacklistd.conf(5). Para ver a lista de hosts atualmente bloqueados, use `dump` combinado com `-b` assim.

```
# blacklistctl dump -b
      address/ma:port id      nfail   last access
213.0.123.128/25:22   OK      6/3     2019/06/08 14:30:19
```

Este exemplo mostra que houve 6 de três tentativas permitidas na porta 22 provenientes do intervalo de endereços `213.0.123.128/25`. Há mais tentativas listadas do que são permitidas porque o SSH permite que um cliente tente vários logins em uma única conexão TCP. Uma conexão que está em andamento no momento não é interrompida pelo blacklistd. A última tentativa de conexão está listada na coluna `last access` da saída.

Para ver o tempo restante em que esse host estará na lista negra, adicione `-r` ao comando anterior.

```
# blacklistctl dump -br
      address/ma:port id      nfail   remaining time
213.0.123.128/25:22   OK      6/3     36s
```

Neste exemplo, restam 36 segundos para que este host não seja mais bloqueado.

30.6.5. Removendo hosts da lista de bloqueios

Às vezes, é necessário remover um host da lista de bloqueios antes que o tempo restante expire. Infelizmente, não há funcionalidade no blacklistd para fazer isso. No entanto, é possível remover o endereço da tabela PF usando pfctl. Para cada porta bloqueada, existe uma âncora filha dentro da âncora do blacklistd definida em /etc/pf.conf. Por exemplo, se houver uma âncora filha para bloquear a porta 22, ela será chamada `blacklistd/22`. Há uma tabela dentro dessa âncora filha que contém os endereços bloqueados. Essa tabela é chamada de port seguida pelo número da porta. Neste exemplo, ele seria chamada de `port22`. Com essas informações em mãos, agora é possível usar o pfctl(8) para exibir todos os endereços listados desta maneira:

```
# pfctl -a blacklistd/22 -t port22 -T show
...
213.0.123.128/25
...
```

Depois de identificar o endereço a ser desbloqueado da lista, o seguinte comando o remove da lista:

```
# pfctl -a blacklistd/22 -t port22 -T delete 213.0.123.128/25
```

O endereço agora foi removido do PF, mas ainda será exibido no blacklistctl, pois ele não conhece nenhuma alteração feita no PF. A entrada no banco de dados do blacklistd expirará e será removida de sua saída eventualmente. A entrada será adicionada novamente se o host estiver correspondendo a uma das regras de bloqueio no blacklistd novamente.

Capítulo 31. Rede Avançada

31.1. Sinopse

Este capítulo aborda vários tópicos avançados de rede.

Depois de ler este capítulo, você saberá:

- O básico de gateways e rotas.
- Como configurar o USB tethering.
- Como configurar os dispositivos IEEE™ 802.11 e Bluetooth™.
- Como fazer o FreeBSD atuar como uma Bridge.
- Como configurar a inicialização via PXE na rede.
- Como configurar o IPv6 em uma máquina FreeBSD.
- Como habilitar e utilizar os recursos do Protocolo CARP (Common Address Redundancy Protocol) no FreeBSD.
- Como configurar múltiplas VLANs no FreeBSD.
- Como configurar um fone de ouvido bluetooth.

Antes de ler este capítulo, você deve:

- Entender os fundamentos dos scripts /etc/rc.
- Estar familiarizado com a terminologia básica de rede.
- Saber como configurar e instalar um novo kernel do FreeBSD (Configurando o kernel do FreeBSD).
- Saber como instalar software adicional de terceiros (Instalando Aplicativos. Pacotes e Ports).

31.2. Gateways e Rotas

O *roteamento* é o mecanismo que permite que um sistema encontre o caminho da rede para outro sistema. Uma *rota* é um par definido de endereços que representam o "destino" e um "gateway". A rota indica que, ao tentar chegar ao destino especificado, você deverá enviar os pacotes pelo gateway especificado. Existem três tipos de destinos: hosts individuais, sub-redes e "padrão". A "rota padrão" é usada se nenhuma outra rota for aplicada. Existem também três tipos de gateways: hosts individuais, interfaces, também chamados de links, e endereços de hardware Ethernet (MAC). Rotas conhecidas são armazenadas em uma tabela de roteamento.

Esta seção fornece uma visão geral dos fundamentos de roteamento. Em seguida, ele demonstra como configurar um sistema FreeBSD como um roteador e oferece algumas dicas de solução de problemas.

31.2.1. Fundamentos de roteamento

Para ver a tabela de roteamento de um sistema FreeBSD, use netstat(1):

```
% netstat -r
Routing tables

Internet:
Destination      Gateway            Flags     Refs     Use     Netif Expire
default          outside-gw         UGS         37     418       em0
localhost        localhost          UH           0     181       lo0
test0            0:e0:b5:36:cf:4f   UHLW         5   63288       re0     77
10.20.30.255     link#1             UHLW         1    2421
example.com      link#1             UC           0       0
host1            0:e0:a8:37:8:1e    UHLW         3    4601       lo0
host2            0:e0:a8:37:8:1e    UHLW         0       5       lo0 =>
host2.example.com link#1            UC           0       0
224              link#1             UC           0       0
```

As entradas neste exemplo são as seguintes:

padrão

A primeira rota nesta tabela especifica a rota `padrão`. Quando o sistema local precisa estabelecer uma conexão com um host remoto, ele verifica a tabela de roteamento para determinar se existe um caminho conhecido. Se o host remoto corresponder a uma entrada na tabela, o sistema verificará se pode se conectar usando a interface especificada nessa entrada.

Se o destino não corresponder a uma entrada ou se todos os caminhos conhecidos falharem, o sistema usará a entrada para a rota padrão. Para hosts em uma rede local, o campo `Gateway` na rota padrão é definido para o sistema que possui uma conexão direta com a internet. Ao ler esta entrada, verifique se a coluna `Flags` indica que o gateway é utilizável (`UG`).

A rota padrão para uma máquina que está funcionando como gateway para o mundo externo será a máquina de gateway no provedor de serviços de Internet (ISP).

localhost

A segunda rota é a `localhost`. A interface especificada na coluna `Netif` para `localhost` é lo0, também conhecido como o dispositivo de loopback. Isso indica que todo o tráfego para esse destino deve ser interno, em vez de enviá-lo pela rede.

Endereço MAC

Os endereços que começam com `0:e0:` são endereços de MAC. O FreeBSD irá identificar automaticamente quaisquer hosts, `test0` no exemplo, na Ethernet local e adicionará uma rota para aquele host através da interface Ethernet, re0. Esse tipo de rota tem um tempo limite, visto na coluna `Expire`, que é usada se o host não responder em um período de tempo específico. Quando isso acontecer, a rota para esse host será automaticamente excluída. Esses hosts são identificados usando o protocolo de informações de roteamento (RIP), que calcula rotas para hosts locais com base em uma determinação de caminho mais curto.

sub-rede

O FreeBSD irá adicionar automaticamente rotas de sub-rede para a sub-rede local. Neste exemplo, `10.20.30.255` é o endereço de broadcast da sub-rede `10.20.30` e `example.com` é o nome de domínio associado a essa sub-rede. A designação `link#1` refere-se à primeira placa Ethernet na máquina.

Hosts de rede local e sub-redes locais têm suas rotas configuradas automaticamente por um daemon chamado routed(8). Se ele não estiver em execução, somente as rotas definidas estaticamente pelo administrador existirão.

host

A linha `host1` refere-se ao host pelo seu endereço Ethernet. Como é o host de envio, o FreeBSD sabe usar a interface de loopback (lo0) em vez da interface Ethernet.

As duas linhas `host2` representam os aliases que foram criados usando ifconfig(8). O símbolo ⇒ após a interface lo0 diz que um alias foi definido além do endereço de loopback. Tais rotas só aparecem no host que suporta o alias e todos os outros hosts na rede local terão uma linha `link#1` para tais rotas.

224

A linha final (destino subnet `224`) lida com multicasting.

Vários atributos de cada rota podem ser vistos na coluna `Flags`. A Flags da Tabela de Roteamento Frequentemente Observados resume algumas destas flags e seus significados:

Tabela 28. Flags da Tabela de Roteamento Frequentemente Observados

Comando	Propósito
U	A rota está ativa (up).
H	O destino da rota é um único host.
G	Envie qualquer coisa para este destino por este gateway, que ele irá descobrir a partir daí para onde enviá-lo.
S	Esta rota foi configurada estaticamente.
C	Clona uma nova rota baseada nessa rota para as máquinas se conectarem. Esse tipo de rota é normalmente usado para redes locais.
W	A rota foi configurada automaticamente com base em uma rota de rede local (clone).
L	A rota envolve referências a um hardware Ethernet (link).

Em um sistema FreeBSD, a rota padrão pode ser definida no /etc/rc.conf especificando o endereço IP do gateway padrão:

```
defaultrouter="10.20.30.1"
```

Também é possível adicionar manualmente a rota usando o comando `route`:

```
# route add default 10.20.30.1
```

Observe que as rotas adicionadas manualmente não sobreviverão a uma reinicialização. Para obter mais informações sobre a manipulação manual das tabelas de roteamento de rede, consulte route(8).

31.2.2. Configurando um roteador com rotas estáticas

Um sistema FreeBSD pode ser configurado como o gateway padrão, ou roteador, para uma rede se for um sistema dual-homed. Um sistema dual-homed é um host que reside em pelo menos duas redes diferentes. Normalmente, cada rede é conectada a uma interface de rede separada, embora o aliasing IP possa ser usado para vincular vários endereços, cada um em uma sub-rede diferente, a uma interface física.

Para que o sistema encaminhe os pacotes entre as interfaces, o FreeBSD deve ser configurado como um roteador. Padrões da Internet e boas práticas de engenharia impedem o Projeto FreeBSD de habilitar esse recurso por padrão, mas ele pode ser configurado para iniciar na inicialização adicionando esta linha ao /etc/rc.conf:

```
gateway_enable="YES"          # Set to YES if this host will be a gateway
```

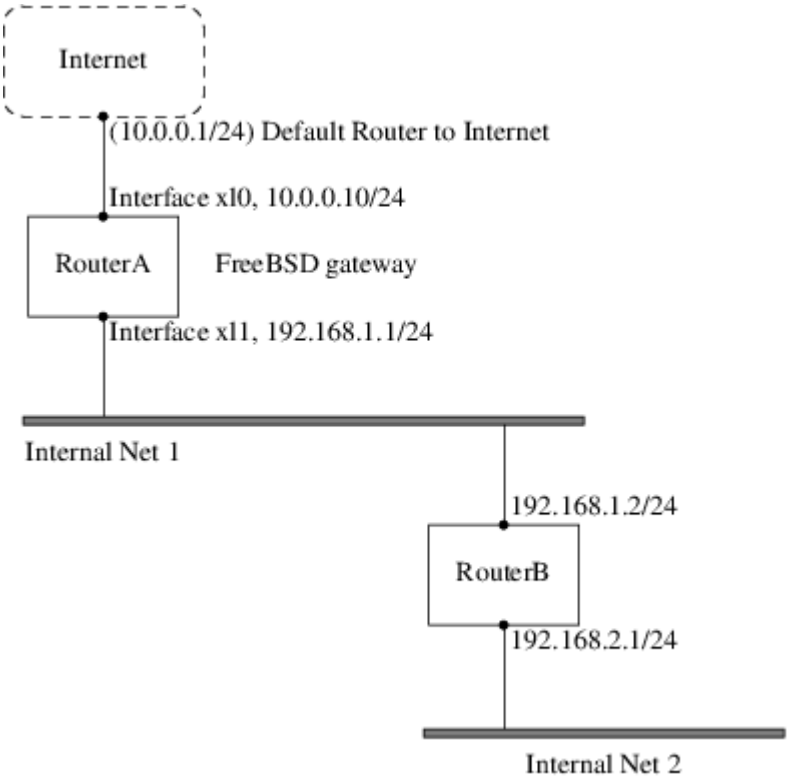
Para habilitar o roteamento agora, defina a variável sysctl(8)`net.inet.ip.forwarding` para `1`. Para parar o roteamento, redefina essa variável para `0`.

A tabela de roteamento de um roteador precisa de rotas adicionais para saber como acessar outras redes. Rotas podem ser adicionadas manualmente usando rotas estáticas ou rotas podem ser aprendidas automaticamente usando um protocolo de roteamento. As rotas estáticas são apropriadas para redes pequenas e esta seção descreve como adicionar uma entrada de roteamento estático para uma rede pequena.



Para grandes redes, as rotas estáticas se tornam não escaláveis rapidamente. O FreeBSD vem com o daemon de roteamento BSD padrão routed(8), que fornece os protocolos de roteamento RIP, versões 1 e 2 e IRDP. O suporte para os protocolos de roteamento BGP e OSPF pode ser instalado usando o pacote ou port net/zebra.

Considere a seguinte rede:



Neste cenário, o `RouterA` é uma máquina FreeBSD que está agindo como um roteador para o resto da Internet. Ele tem uma rota padrão definida como `10.0.0.1`, que permite a conexão com o mundo externo. O `RouterB` já está configurado para usar `192.168.1.1` como seu gateway padrão.

Antes de adicionar rotas estáticas, a tabela de roteamento no `RouterA` se parece com:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags     Refs      Use  Netif  Expire
default            10.0.0.1           UGS          0    49378    xl0
127.0.0.1          127.0.0.1          UH           0        6    lo0
10.0.0.0/24        link#1             UC           0        0    xl0
192.168.1.0/24     link#2             UC           0        0    xl1
```

Com a tabela de roteamento atual, o `RouterA` não tem uma rota para a rede `192.168.2.0/24`. O comando a seguir adiciona a rede `Internal Net 2` à tabela de roteamento do `RouterA` usando `192.168.1.2` como o próximo salto:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Agora, o `RouterA` pode alcançar qualquer host na rede `192.168.2.0/24`. No entanto, as informações de roteamento não persistirão se o sistema FreeBSD for reinicializado. Se uma rota estática precisar ser persistente, adicione-a ao /etc/rc.conf:

```
# Add Internal Net 2 as a persistent static route
```

```
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

A variável de configuração `static_routes` é uma lista de strings separadas por um espaço, onde cada string faz referência a um nome de rota. A variável `route_internalnet2` contém a rota estática para esse nome de rota.

Usar mais de uma string em `static_routes` cria várias rotas estáticas. A seguir, é mostrado um exemplo de adição de rotas estáticas para as redes `192.168.0.0/24` e `192.168.1.0/24`:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

31.2.3. Solução de problemas

Quando um espaço de endereçamento é atribuído a uma rede, o provedor de serviços configura suas tabelas de roteamento para que todo o tráfego da rede seja enviado para o link do site. Mas como os sites externos sabem enviar seus pacotes para a rede do ISP?

Existe um sistema que rastreia todos os espaços de endereçamento e define seu ponto de conexão com o backbone da Internet, ou as principais linhas que transportam o tráfego da Internet pelo país e pelo mundo. Cada máquina de backbone possui uma cópia de um conjunto mestre de tabelas, que direciona o tráfego de uma rede específica para uma portadora de backbone específica e, a partir daí, desce a cadeia de provedores de serviços até alcançar uma determinada rede.

É tarefa do provedor de serviços anunciar aos sites de backbone que eles são o ponto de conexão e, assim, o caminho para dentro de um site. Isso é conhecido como propagação de rota.

Às vezes, há um problema com a propagação de rotas e alguns sites não conseguem se conectar. Talvez o comando mais útil para tentar descobrir onde o roteamento está quebrando seja o `traceroute`. Ele é útil quando o `ping` falha.

Ao usar o `traceroute`, inclua o endereço do host remoto para se conectar. A saída mostrará os gateway ao longo do caminho da tentativa, eventualmente atingindo o host de destino ou encerrando devido à falta de conexão. Para mais informações, consulte traceroute(8).

31.2.4. Considerações sobre Multicast

O FreeBSD suporta nativamente tanto aplicativos multicast e quanto roteamento multicast. Os aplicativos multicast não exigem nenhuma configuração especial para serem executados no FreeBSD. O suporte ao roteamento multicast requer que a seguinte opção seja compilada em um kernel personalizado:

```
options MROUTING
```

O daemon de roteamento multicast, mrouted, pode ser instalado usando o pacote ou port

net/mrouted. Este daemon implementa o protocolo de roteamento multicast DVMRP e é configurado editando o /usr/local/etc/mrouted.conf para configurar os túneis e o DVMRP. A instalação do mrouted também instala o map-mbone e o mrinfo, bem como suas páginas de manual associadas. Consulte estes documentos para exemplos de configuração.



O DVMRP foi amplamente substituído pelo protocolo PIM em muitas instalações multicast. Consulte pim(4) para obter maiores informações.

31.3. Rede sem fio

31.3.1. Noções básicas sobre redes sem fio

A maioria das redes sem fio é baseada nos padrões IEEE™802.11. Uma rede sem fio básica consiste em várias estações que se comunicam com rádios que transmitem na banda de 2,4 GHz ou 5 GHz, embora isso varie de acordo com a localidade e também esteja mudando para permitir a comunicação nas faixas de 2,3 GHz e 4,9 GHz.

As redes 802.11 são organizadas de duas maneiras. No *modo de infra-estrutura*, uma estação atua como mestre para todas as outras estações que se associam a ela, a rede é conhecida como BSS e a estação mestre é denominada ponto de acesso. (AP). Em um BSS, toda a comunicação passa pelo AP; mesmo quando uma estação deseja se comunicar com outra estação sem fio, as mensagens devem passar pelo AP. Na segunda forma de rede, não há mestre e as estações se comunicam diretamente. Esta forma de rede é denominada IBSS e é comumente conhecida como uma *rede ad-hoc*.

As redes 802.11 foram implantadas pela primeira vez na banda de 2,4 GHz usando protocolos definidos pelo padrão 802.11 e 802.11b da IEEE™. Essas especificações incluem as frequências operacionais e as características da camada MAC, incluindo as taxas de enquadramento e transmissão, pois a comunicação pode ocorrer em várias taxas. Posteriormente, o padrão 802.11a definiu a operação na faixa de 5GHz, incluindo diferentes mecanismos de sinalização e taxas de transmissão mais altas. Mais tarde, o padrão 802.11g definiu o uso de mecanismos de sinalização e transmissão 802.11a na banda de 2,4 GHz de modo a ser compatível com redes 802.11b.

Separadas das técnicas de transmissão básicas, as redes 802.11 possuem uma variedade de mecanismos de segurança. As especificações originais do 802.11 definiam um protocolo de segurança simples chamado WEP. Este protocolo usa uma chave pré-compartilhada fixa e a criptografia criptográfica RC4 para codificar dados transmitidos em uma rede. Todas as estações devem concordar com a chave fixa para se comunicar. Esse esquema mostrou-se de fácil quebra e agora raramente é usado, exceto para desencorajar usuários transitórios a se juntarem a uma rede. A prática atual de segurança é dada pela especificação 802.11i do IEEE™ que define novas cifras criptográficas e um protocolo adicional para autenticar estações para um ponto de acesso e para trocar chaves para comunicação de dados. As chaves criptográficas são atualizadas periodicamente e existem mecanismos para detectar e combater tentativas de invasão. Outra especificação de protocolo de segurança comumente usada em redes sem fio é denominada WPA, que foi um precursor do 802.11i. O WPA especifica um subconjunto dos requisitos encontrados no 802.11i e foi projetado para implementação em hardware legado. Especificamente, o WPA requer apenas a codificação TKIP derivada da codificação original WEP. O 802.11i permite o uso do TKIP, mas também requer suporte para uma criptografia mais forte, o AES-CCM, para criptografar os dados. A codificação AES não era exigida no WPA porque foi considerada demasiadamente cara

computacionalmente para ser executada em hardware legado.

Um outro padrão a se ter em conta é o 802.11e. Ele define protocolos para a implantação de aplicativos multimídia, como streaming de vídeo e voz sobre IP (VoIP), em uma rede 802.11. Como o 802.11i, o 802.11e também tem uma especificação de precursor denominada WME (posteriormente renomeada como WMM) que foi definida por um grupo industrial como um subconjunto do 802.11e que pode ser implantado agora para habilitar aplicativos multimídia enquanto aguarda a ratificação final do 802.11e. O mais importante a saber sobre o 802.11e e o WME/WMM é que ele permite o tráfego prioritário através de uma rede sem fio através de protocolos de Qualidade de Serviço (QoS) e protocolos de acesso de mídia aprimorados. A implementação adequada desses protocolos permite o aumento rápido de dados e o fluxo de tráfego priorizado.

O FreeBSD suporta redes que operam usando 802.11a, 802.11b e 802.11g. Os protocolos de segurança WPA e 802.11i também são suportados (em conjunto com qualquer um dos 11a, 11b e 11g) e o QoS e priorização de tráfego exigidos pelo protocolo WME/WMM são suportados por um conjunto limitado de dispositivos sem fio.

31.3.2. Inicio Rápido

Conectar um computador a uma rede sem fio existente é uma situação muito comum. Este procedimento mostra as etapas necessárias.

1. Obtenha o SSID (identificador de conjunto de serviços) e PSK (chave pré-compartilhada) para a rede sem fio do administrador da rede.
2. Identifique o adaptador sem fio. O kernel GENERIC do FreeBSD inclui drivers para muitos adaptadores sem fio comuns. Se o adaptador sem fio for um desses modelos, ele será mostrado na saída do ifconfig(8):

   ```
   % ifconfig | grep -B3 -i wireless
   ```

   No FreeBSD 11 ou superior, use este comando:

   ```
   % sysctl net.wlan.devices
   ```

   Se um adaptador sem fio não estiver listado, um módulo adicional do kernel pode ser necessário, ou pode ser um modelo não suportado pelo FreeBSD.

   Este exemplo mostra o adaptador wireless Atheros `ath0`.

3. Adicione uma entrada para esta rede ao /etc/wpa_supplicant.conf. Se o arquivo não existir, crie-o. Substitua *myssid* e *mypsk* pelo SSID e PSK fornecidos pelo administrador da rede.

   ```
   network={
       ssid="myssid"
       psk="mypsk"
   ```

```
}
```

4. Adicione entradas ao /etc/rc.conf para configurar a rede na inicialização:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA SYNCDHCP"
```

5. Reinicie o computador ou reinicie o serviço de rede para conectar-se à rede:

```
# service netif restart
```

31.3.3. Configuração básica

31.3.3.1. Configuração do Kernel

Para usar a rede sem fio, uma placa de rede sem fio é necessária e o kernel precisa ser configurado com o suporte de rede sem fio apropriado. O kernel é separado em vários módulos para que apenas o suporte necessário precise ser configurado.

Os dispositivos sem fio mais comumente usados são aqueles que usam peças fabricadas pela Atheros. Estes dispositivos são suportados pelo ath(4) e requerem que a seguinte linha seja adicionada ao /boot/loader.conf :

```
if_ath_load="YES"
```

O driver Atheros é dividido em três partes separadas: o driver (ath(4)), a camada de suporte de hardware que lida com funções específicas do chip (ath_hal(4)) e um algoritmo para selecionar a taxa de transmissão de quadros. Quando este suporte é carregado como módulo do kernel, quaisquer dependências são tratadas automaticamente. Para carregar o suporte para um tipo diferente de dispositivo sem fio, especifique o módulo para esse dispositivo. Este exemplo é para dispositivos baseados no driver Intersil Prism parts (wi(4)):

```
if_wi_load="YES"
```



Os exemplos nesta seção usam um dispositivoath(4) e o nome do dispositivo nos exemplos deve ser alterado de acordo com a configuração. Uma lista de drivers sem fio disponíveis e adaptadores suportados pode ser encontrada nas Notas de Hardware do FreeBSD, disponíveis nas Informações de Release da página do site do FreeBSD. Se um driver nativo do FreeBSD para o dispositivo sem fio não existir, pode ser possível usar o driver Windows™ com a ajuda do wrapper de driver NDIS.

Além disso, os módulos que implementam o suporte criptográfico para os protocolos de segurança

devem ser carregados. Estes destinam-se a ser dinamicamente carregados sob demanda pelo módulo wlan(4), mas por enquanto eles devem ser configurados manualmente. Os seguintes módulos estão disponíveis: wlan_wep(4), wlan_ccmp(4), e wlan_tkip(4). Os drivers wlan_ccmp(4) e wlan_tkip(4) são necessário apenas ao usar os protocolos de segurança WPA ou 802.11i. Se a rede não usar criptografia, o suporte a wlan_wep(4) não será necessário. Para carregar estes módulos no momento da inicialização, adicione as seguintes linhas ao /boot/loader.conf:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```

Uma vez que esta informação tenha sido adicionada ao /boot/loader.conf, reinicie a caixa FreeBSD. Como alternativa, carregue os módulos manualmente usando kldload(8).



Para usuários que não querem usar módulos, é possível compilar esses drivers no kernel adicionando as seguintes linhas a um arquivo de configuração de kernel personalizado:

```
device wlan              # 802.11 support
device wlan_wep          # 802.11 WEP support
device wlan_ccmp         # 802.11 CCMP support
device wlan_tkip         # 802.11 TKIP support
device wlan_amrr         # AMRR transmit rate control algorithm
device ath               # Atheros pci/cardbus NIC's
device ath_hal           # pci/cardbus chip support
options AH_SUPPORT_AR5416 # enable AR5416 tx/rx descriptors
device ath_rate_sample   # SampleRate tx rate control for ath
```

Com esta informação no arquivo de configuração do kernel, recompile o kernel e reinicie a máquina do FreeBSD.

Informações sobre o dispositivo sem fio devem aparecer nas mensagens de inicialização, assim:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```

31.3.3.2. Definindo a Região Correta

Como a situação regulatória é diferente em várias partes do mundo, é necessário definir corretamente os domínios que se aplicam à sua localização para obter as informações corretas sobre quais canais podem ser usados.

As definições de região disponíveis podem ser encontradas em /etc/regdomain.xml. Para definir os dados em tempo de execução, use o `ifconfig`:

```
# ifconfig wlan0 regdomain ETSI country AT
```

Para persistir as configurações, adicione-o ao /etc/rc.conf:

```
# sysrc create_args_wlan0="country AT regdomain ETSI"
```

31.3.4. Modo de Infraestrutura

O modo de infra-estrutura (BSS) é o modo normalmente usado. Neste modo, vários pontos de acesso sem fio são conectados a uma rede com fio. Cada rede sem fio tem seu próprio nome, chamado de SSID. Os clientes sem fio se conectam aos pontos de acesso sem fio.

31.3.4.1. Clientes do FreeBSD

31.3.4.1.1. Como encontrar pontos de acesso

Para procurar redes disponíveis, use ifconfig(8). Essa solicitação pode demorar alguns instantes para ser concluída, pois exige que o sistema alterne para cada frequência sem fio disponível e sonde os pontos de acesso disponíveis. Apenas o superusuário pode iniciar uma varredura:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
dlinkap         00:13:46:49:41:76   11   54M -90:96   100 EPS  WPA WME
freebsdap       00:11:95:c3:0d:ac    1   54M -83:96   100 EPS  WPA
```



A interface deve estar `up` antes de poder efetuar a busca. Pedidos de varredura subsequentes não exigem que a interface seja marcada como up novamente.

A saída de uma solicitação de varredura lista cada rede BSS/IBSS encontrada. Além de listar o nome da rede, o `SSID`, a saída também mostra o `BSSID`, que é o endereço MAC do ponto de acesso. O campo `CAPS` identifica o tipo de cada rede e os recursos das estações que operam lá:

Tabela 29. Códigos de capacidade da estação

Código de capacidade	Significado
`E`	Conjunto de serviços estendidos (ESS). Indica que a estação faz parte de uma rede de infraestrutura em vez de uma rede IBSS/ad-hoc.
`I`	Rede IBSS/ad-hoc. Indica que a estação faz parte de uma rede ad-hoc em vez de uma rede ESS.
`P`	Privacidade. A criptografia é necessária para todos os quadros de dados trocados dentro do BSS usando meios criptográficos como o WEP, o TKIP ou o AES-CCMP.

Código de capacidade	Significado
S	Preâmbulo Curto. Indica que a rede está usando preâmbulos curtos, definidos em 802.11b de Alta Taxa/DSSS PHYS, e utiliza um campo de sincronização de 56 bits em vez do campo de 128 bits usado no modo de preâmbulo longo.
s	Tempo de slot curto. Indica que a rede 802.11g está usando um tempo de slot curto porque não há estações legadas (802.11b) presentes.

Pode-se também exibir a lista atual de redes conhecidas com:

```
# ifconfig wlan0 list scan
```

Essas informações podem ser atualizadas automaticamente pelo adaptador ou manualmente com uma solicitação de `scan`. Dados antigos são automaticamente removidos do cache, então com o tempo essa lista pode diminuir a menos que mais varreduras sejam feitas.

31.3.4.1.2. Configurações básicas

Esta seção fornece um exemplo simples de como fazer com que o adaptador de rede sem fio funcione no FreeBSD sem criptografia. Uma vez familiarizado com esses conceitos, é altamente recomendável usar o WPA para configurar a rede sem fio.

Existem três etapas básicas para configurar uma rede sem fio: selecionar um ponto de acesso, autenticar a estação e configurar um endereço IP. As seções a seguir discutem cada etapa.

31.3.4.1.2.1. Selecionando um ponto de acesso

Na maioria das vezes, é suficiente deixar o sistema escolher um ponto de acesso usando a heurística integrada. Este é o comportamento padrão quando uma interface é marcada como up ou está listada em /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Se houver vários pontos de acesso, um específico pode ser selecionado pelo seu SSID:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid your_ssid_here DHCP"
```

Em um ambiente em que há vários pontos de acesso com o mesmo SSID, o que geralmente é feito para simplificar o roaming, talvez seja necessário associá-lo a um dispositivo específico. Neste caso, o BSSID do ponto de acesso pode ser especificado, com ou sem o SSID:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid your_ssid_here bssid xx:xx:xx:xx:xx:xx DHCP"
```

Existem outras maneiras de restringir a escolha de um ponto de acesso, como limitar o conjunto de freqüências que o sistema fará a varredura. Isso pode ser útil para uma placa sem fio de banda múltipla, pois a varredura de todos os canais possíveis pode consumir muito tempo. Para limitar a operação a uma banda específica, use o parâmetro `mode`:

```
wlans_ath0="wlan0"
ifconfig_wlan0="mode 11g ssid your_ssid_here DHCP"
```

Este exemplo forçará a placa a operar em 802.11g, que é definido apenas para freqüências de 2.4GHz, portanto, qualquer canal de 5GHz não será considerado. Isso também pode ser obtido com o parâmetro `channel`, que bloqueia a operação para uma frequência específica, e o parâmetro `chanlist`, para especificar uma lista de canais para varredura. Maiores informações sobre esses parâmetros podem ser encontradas em ifconfig(8).

31.3.4.1.2.2. Autenticação

Quando um ponto de acesso é selecionado, a estação precisa se autenticar antes de poder transmitir dados. A autenticação pode acontecer de várias maneiras. O esquema mais comum, autenticação aberta, permite que qualquer estação entre na rede e se comunique. Essa é a autenticação a ser usada para fins de teste na primeira vez em que uma rede sem fio é configurada. Outros esquemas exigem que os handshakes criptográficos sejam concluídos antes que o tráfego de dados possa fluir, usando chaves ou segredos pré-compartilhados ou esquemas mais complexos que envolvam serviços de back-end, como o RADIUS. Autenticação aberta é a configuração padrão. A próxima configuração mais comum é o WPA-PSK, também conhecido como WPA Pessoal, que é descrito em WPA-PSK.



Se estiver usando uma estação base Extreme AirPort™ da Apple™ para um ponto de acesso, a autenticação de chave compartilhada juntamente com um WEP chave precisa ser configurada. Isto pode ser configurado em /etc/rc.conf ou usando wpa_supplicant(8). Para uma única estação base AirPort™, o acesso pode ser configurado com:

```
wlans_ath0="wlan0"
ifconfig_wlan0="authmode shared wepmode on weptxkey 1 wepkey 01234567
DHCP"
```

Em geral, a autenticação de chave compartilhada deve ser evitada porque ela usa o material de chave WEP de uma maneira altamente restrita, facilitando ainda mais a quebra da chave. Se o WEP deve ser usado para compatibilidade com dispositivos legados, é melhor usar o WEP com a autenticação `open`. Mais informações sobre o WEP podem ser encontradas em WEP.

31.3.4.1.2.3. Obtendo um endereço IP com DHCP

Quando um ponto de acesso é selecionado e os parâmetros de autenticação são definidos, um endereço IP deve ser obtido para se comunicar. Na maioria das vezes, o endereço IP é obtido através do DHCP. Para isso, edite o /etc/rc.conf e adicione o `DHCP` à configuração do dispositivo:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

A interface sem fio está agora pronta para subir:

```
# service netif start
```

Quando a interface estiver rodando, use o ifconfig(8) para ver o status da interface ath0:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
        media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
        status: associated
        ssid dlinkap channel 11 (2462 Mhz 11g) bssid 00:13:46:49:41:76
        country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
        scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
        roam:rate 5 protmode CTS wme burst
```

A linha `status: associated` significa que está conectada à rede sem fio. O `bssid 00:13:46:49:41:76` é o endereço MAC do ponto de acesso e o `authmode OPEN` indica que a comunicação é não criptografada.

31.3.4.1.2.4. Endereço IP estático

Se um endereço IP não puder ser obtido de um servidor DHCP, defina um endereço de IP fixo. Substitua a palavra-chave `DHCP` mostrada acima pelas informações do endereço. Certifique-se de reter quaisquer outros parâmetros para selecionar o ponto de acesso:

```
wlans_ath0="wlan0"
ifconfig_wlan0="inet 192.168.1.100 netmask 255.255.255.0 ssid your_ssid_here"
```

31.3.4.1.3. WPA

O Wi-Fi Protected Access (WPA) é um protocolo de segurança usado em conjunto com redes 802.11 para resolver a falta de autenticação adequada e a fraqueza do WEP. O WPA utiliza o protocolo de autenticação 802.1X e usa uma das várias codificações disponíveis em vez do WEP para integridade de dados. A única codificação exigida pelo WPA é o protocolo de integridade de chave temporária (TKIP). O TKIP é uma codificação que estende a codificação básica RC4 usada pelo WEP,

adicionando verificação de integridade, detecção de adulteração e medidas para responder a intrusões detectadas. O TKIP foi projetado para funcionar em hardware legado apenas com uma modificação de software. Ele representa um compromisso que melhora a segurança, mas ainda não é totalmente imune a ataques. O WPA também especifica a codificação AES-CCMP como uma alternativa para o TKIP, e é preferível quando possível. Para esta especificação, o termo WPA2 ou RSN é comumente usado.

O WPA define protocolos de autenticação e criptografia. A autenticação é mais comumente feita usando uma de duas técnicas: por 802.1X e um serviço de autenticação backend, como o RADIUS, ou por um handshake mínimo entre a estação e o ponto de acesso usando um segredo pré-compartilhado. O primeiro é comumente chamado de WPA Enterprise e o último é conhecido como WPA Pessoal. Como a maioria das pessoas não configurará um servidor backend RADIUS para sua rede sem fio, o WPA-PSK é de longe a configuração mais comumente encontrada para o WPA .

O controle da conexão sem fio e a negociação ou autenticação de chave com um servidor é feito usando o wpa_supplicant(8). Este programa requer um arquivo de configuração, o /etc/wpa_supplicant.conf, para ser executado. Maiores informações sobre este arquivo podem ser encontradas em wpa_supplicant.conf(5).

31.3.4.1.3.1. WPA-PSK

O WPA-PSK, também conhecido como WPA Pessoal, é baseado em uma chave pré-compartilhada (PSK) que é gerada a partir de uma determinada senha e usado como chave mestra na rede sem fio. Isso significa que todos os usuários sem fio compartilharão a mesma chave. O WPA-PSK destina-se a redes pequenas em que o uso de um servidor de autenticação não é possível ou desejado.



Sempre use senhas fortes que sejam suficientemente longas e feitas de um alfabeto rico para que elas não sejam facilmente adivinhadas ou atacadas.

O primeiro passo é a configuração do /etc/wpa_supplicant.conf com o SSID e a chave pré-compartilhada da rede:

```
network={
  ssid="freebsdap"
  psk="freebsdmall"
}
```

Então, em /etc/rc.conf, indique que a configuração do dispositivo sem fio será feita com o WPA e o endereço IP será obtido com o DHCP:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Então, suba a interface:

```
# service netif start
Starting wpa_supplicant.
```

```
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Ou, tente configurar a interface manualmente usando as informações em /etc/wpa_supplicant.conf:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)
Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to 00:11:95:c3:0d:ac completed (auth) [id=0 id_str=]
```

A próxima operação é iniciar o dhclient(8) para obter o endereço IP do servidor DHCP:

```
# dhclient wlan0
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```



Se o /etc/rc.conf tiver uma entrada `ifconfig_wlan0="DHCP"`, dhclient(8) será iniciado automaticamente após o wpa_supplicant(8) associar-se ao ponto de acesso.

Se o DHCP não for possível ou desejado, defina um endereço IP estático após o wpa_supplicant(8)

autenticar a estação:

```
# ifconfig wlan0 inet 192.168.0.100 netmask 255.255.255.0
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Quando o DHCP não é usado, o gateway padrão e o servidor de nomes também precisam ser definidos manualmente:

```
# route add default your_default_router
# echo "nameserver your_DNS_server" >> /etc/resolv.conf
```

31.3.4.1.3.2. WPA com EAP-TLS

A segunda maneira de usar o WPA é com um servidor de autenticação de backend 802.1X. Neste caso, o WPA é chamado de WPA Enterprise para diferenciá-lo do WPA Pessoal menos seguro. A autenticação no WPA Enterprise é baseada no protocolo de autenticação extensível (EAP).

O EAP não vem com um método de criptografia. Em vez disso, o EAP é incorporado dentro de um túnel criptografado. Existem muitos métodos de autenticação EAP, mas o EAP-TLS, o EAP-TTLS e o EAP-PEAP são os mais comum.

O EAP com Segurança da Camada de Transporte (EAP-TLS) é um protocolo de autenticação sem fio bem suportado, já que foi o primeiro método EAP a ser certificado pela WiFi Alliance. O EAP-TLS requer três certificados para executar: o certificado da Autoridade de Certificação (CA) instalado em todas as máquinas, o certificado do servidor para o servidor de autenticação e um certificado de cliente para cada cliente sem fio. Nesse método EAP, o servidor de autenticação e o cliente sem fio autenticam um ao outro apresentando seus respectivos certificados e, em seguida, verificam se esses certificados foram assinados pela CA da organização.

Como anteriormente, a configuração é feita através do /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap" ①
  proto=RSN  ②
  key_mgmt=WPA-EAP ③
  eap=TLS ④
  identity="loader" ⑤
  ca_cert="/etc/certs/cacert.pem" ⑥
```

```
  client_cert="/etc/certs/clientcert.pem" ⑦
  private_key="/etc/certs/clientkey.pem" ⑧
  private_key_passwd="freebsdmallclient" ⑨
}
```

① Este campo indica o nome da rede (SSID).

② Este exemplo usa o protocolo 802.11i RSN IEEE™, também conhecido como WPA2.

③ A linha `key_mgmt` refere-se ao protocolo de gerenciamento de chaves a ser usado. Neste exemplo, é o WPA usando a autenticação EAP.

④ Este campo indica o método EAP para a conexão.

⑤ O campo `identity` contém a sequência de identidade para EAP.

⑥ O campo `ca_cert` indica o nome do caminho do arquivo de certificado CA. Este arquivo é necessário para verificar o certificado do servidor.

⑦ A linha `client_cert` fornece o nome do caminho para o arquivo de certificado do cliente. Este certificado é exclusivo para cada cliente sem fio da rede.

⑧ O campo `private_key` é o nome do caminho para o arquivo de chave privada do certificado do cliente.

⑨ O campo `private_key_passwd` contém a frase secreta para a chave privada.

Em seguida, adicione as seguintes linhas ao /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

O próximo passo é subir a interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Também é possível subir a interface manualmente usando wpa_supplicant(8) e ifconfig(8).

31.3.4.1.3.3. WPA com EAP-TTLS

Com o EAP-TLS, o servidor de autenticação e o cliente precisam de um certificado. Com o EAP-TTLS, um certificado de cliente é opcional. Esse método é semelhante a um servidor da Web que cria um túnel seguro SSL, mesmo se os visitantes não tiverem certificados do lado do cliente. O EAP-TTLS usa um túnel TLS criptografado para o transporte seguro dos dados de autenticação.

A configuração necessária pode ser adicionada ao /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS ①
  identity="test" ②
  password="test" ③
  ca_cert="/etc/certs/cacert.pem" ④
  phase2="auth=MD5" ⑤
}
```

① Este campo especifica o método EAP para a conexão.

② O campo `identity` contém a sequência de identidade para a autenticação EAP dentro do túnel TLS criptografado.

③ O campo `password` contém a senha para a autenticação EAP.

④ O campo `ca_cert` indica o nome do caminho do arquivo de certificado CA. Este arquivo é necessário para verificar o certificado do servidor.

⑤ Este campo especifica o método de autenticação usado no túnel TLS criptografado. Neste exemplo, o EAP com desafio MD5 é usado. A fase de "inner authentication" é freqüentemente chamada de "phase2".

Em seguida, adicione as seguintes linhas ao /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

O próximo passo é subir a interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
```

```
        media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
        status: associated
        ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
        country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
        AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
        bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
        wme burst roaming MANUAL
```

31.3.4.1.3.4. WPA com EAP-PEAP



O PEAPv0/EAP-MSCHAPv2 é o método PEAP mais comum. Neste capítulo, o termo PEAP é usado para se referir a esse método.

O EAP protegido (PEAP) foi criado como uma alternativa ao EAP-TTLS e é o padrão mais usado do EAP após o EAP-TLS. Em uma rede com sistemas operacionais mistos, o PEAP deve ser o padrão mais suportado após o EAP-TLS.

O PEAP é semelhante ao EAP-TTLS, pois usa um certificado do lado do servidor para autenticar clientes criando um túnel TLS criptografado entre o cliente e o servidor de autenticação, que protege a troca subsequente das informações de autenticação. A autenticação PEAP difere do EAP-TTLS, pois transmite o nome de usuário em texto aberto e somente a senha é enviada no túnel TLS criptografado. O EAP-TTLS usará o túnel TLS para o nome de usuário e para a senha.

Adicione as seguintes linhas ao /etc/wpa_supplicant.conf para ajustar as configurações relacionadas ao EAP-PEAP:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP ①
  identity="test" ②
  password="test" ③
  ca_cert="/etc/certs/cacert.pem" ④
  phase1="peaplabel=0" ⑤
  phase2="auth=MSCHAPV2" ⑥
}
```

① Este campo especifica o método EAP para a conexão.

② O campo `identity` contém a sequência de identidade para a autenticação EAP dentro do túnel TLS criptografado.

③ O campo `password` contém a senha para a autenticação EAP.

④ O campo `ca_cert` indica o nome do caminho do arquivo de certificado CA. Este arquivo é necessário para verificar o certificado do servidor.

⑤ Este campo contém os parâmetros para a primeira fase de autenticação, o túnel TLS. De acordo com o servidor de autenticação usado, especifique um label específico para autenticação. Na maioria das vezes, o label será "client EAP encryption" que é definido usando `peaplabel=0`.

Maiores informações podem ser encontradas em wpa_supplicant.conf(5).

⑥ Este campo especifica o protocolo de autenticação usado no túnel TLS criptografado. No caso do PEAP, é `auth=MSCHAPV2`.

Adicione o seguinte ao /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Então, suba a interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

31.3.4.1.4. WEP

A privacidade equivalente com fio (WEP) faz parte do padrão 802.11 original. Não há mecanismo de autenticação, apenas uma forma fraca de controle de acesso que é facilmente quebrada.

O WEP pode ser configurado usando o ifconfig(8):

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 inet 192.168.1.100 netmask 255.255.255.0 \
        ssid my_net wepmode on weptxkey 3 wepkey 3:0x3456789012
```

- O `weptxkey` especifica qual chave WEP será usada na transmissão. Este exemplo usa a terceira chave. Isso deve corresponder à configuração no ponto de acesso. Quando não tiver certeza de qual chave é usada pelo ponto de acesso, tente `1` (a primeira chave) para esse valor.
- O `wepkey` seleciona uma das chaves WEP. Deve estar no formato *index:key*. A chave `1` é usada por padrão; o índice só precisa ser definido ao usar uma chave diferente da primeira.



Substitua o `0x3456789012` com a chave configurada para uso no ponto de acesso.

Consulte o ifconfig(8) para obter maiores informações.

O recurso wpa_supplicant(8) pode ser usado para configurar uma interface sem fio com o WEP. O exemplo acima pode ser configurado adicionando as seguintes linhas ao /etc/wpa_supplicant.conf:

```
network={
  ssid="my_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Então:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:49:41:76 (SSID='dlinkap' freq=2437 MHz)
Associated with 00:13:46:49:41:76
```

31.3.5. Modo Ad-hoc

O modo IBSS, também chamado de modo ad-hoc, é projetado para conexões ponto a ponto. Por exemplo, para estabelecer uma rede ad-hoc entre as máquinas `A` e `B`, escolha dois endereços IP e um SSID.

Em `A`:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

O parâmetro `adhoc` indica que a interface está sendo executada no modo IBSS.

`B` deve ser capaz de detectar `A`:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 up scan
```

```
  SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
  freebsdap       02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

O I na saída confirma que A está no modo ad-hoc. Agora, configure B com um endereço IP diferente:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Ambos A e B agora estão prontos para trocar informações.

31.3.6. Pontos de Acesso com um host FreeBSD

O FreeBSD pode atuar como um Access Point (AP), o que elimina a necessidade de comprar um hardware AP ou executar uma rede ad-hoc. Isso pode ser particularmente útil quando uma máquina FreeBSD está atuando como um gateway para outra rede, como a Internet.

31.3.6.1. Configurações básicas

Antes de configurar uma máquina FreeBSD como um AP, o kernel deve ser configurado com o suporte de rede apropriado para a placa wireless assim como os protocolos de segurança que estão sendo usados. Para maiores detalhes, veja Configuração básica.



O wrapper do driver NDIS para os drivers Windows™ não suporta atualmente a operação AP. Somente os drivers nativos de rede sem fio do FreeBSD suportam o modo AP.

Quando o suporte à rede sem fio estiver carregado, verifique se o dispositivo sem fio oferece suporte ao modo de ponto de acesso baseado em host, também conhecido como modo hostap:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 list caps
drivercaps=6f85edc1<STA,FF,TURBOP,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,
MBSS,WPA1,WPA2,BURST,WME,WDS,BGSCAN,TXFRAG>
cryptocaps=1f<WEP,TKIP,AES,AES_CCM,TKIPMIC>
```

Esta saída exibe os recursos da placa. A palavra HOSTAP confirma que esta placa sem fio pode atuar como um AP. Diversas cifras suportadas também são listadas: WEP, TKIP e AES. Esta informação indica quais protocolos de segurança podem ser usados no AP.

O dispositivo sem fio só pode ser colocado no modo hostap durante a criação do pseudo-dispositivo de rede, portanto, um dispositivo criado anteriormente deve ser destruído primeiro:

```
# ifconfig wlan0 destroy
```

e então regenerado com a opção correta antes de configurar os outros parâmetros:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g
channel 1
```

Use o ifconfig(8) novamente para ver o status da interface wlan0:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst dtimperiod 1 -dfs
```

O parâmetro `hostap` indica que a interface está sendo executada no modo de ponto de acesso baseado em host.

A configuração da interface pode ser feita automaticamente no momento da inicialização, adicionando as seguintes linhas ao /etc/rc.conf:

```
wlans_ath0="wlan0"
create_args_wlan0="wlanmode hostap"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel
1"
```

31.3.6.2. Ponto de acesso baseado em host sem autenticação ou criptografia

Embora não seja recomendado executar um AP sem nenhuma autenticação ou criptografia, esta é uma maneira simples de verificar se o AP está funcionando. Essa configuração também é importante para depurar problemas do cliente.

Quando o AP estiver configurado, inicie uma verificação de outra máquina sem fio para encontrar o AP:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
```

```
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M -66:-96  100 ES   WME
```

A máquina cliente encontrou o AP e pode ser associado a ele:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
      scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
      roam:rate 5 protmode CTS wme burst
```

31.3.6.3. Ponto de acesso baseado em host com WPA2

Esta seção se concentra na configuração de um ponto de acesso do FreeBSD usando o protocolo de segurança WPA2. Maiores detalhes sobre WPA e a configuração de clientes sem fio baseados em WPA podem ser encontrados em WPA.

O daemon hostapd(8) é usado para lidar com a autenticação de clientes e o gerenciamento de chaves no AP com WPA2 habilitado.

As seguintes operações de configuração são executadas na máquina FreeBSD atuando como o AP. Uma vez que o AP esteja funcionando corretamente, o hostapd(8) pode ser iniciado automaticamente na inicialização com essa linha em /etc/rc.conf:

```
hostapd_enable="YES"
```

Antes de tentar configurar o hostapd(8), primeiro defina as configurações básicas introduzidas em Configurações básicas .

31.3.6.3.1. WPA2-PSK

O WPA2-PSK destina-se a redes pequenas em que o uso de um servidor de autenticação backend não é possível ou desejado.

A configuração é feita em /etc/hostapd.conf:

```
interface=wlan0                 ①
debug=1                         ②
ctrl_interface=/var/run/hostapd ③
ctrl_interface_group=wheel      ④
ssid=freebsdap                  ⑤
```

```
wpa=2                           ⑥
wpa_passphrase=freebsdmall      ⑦
wpa_key_mgmt=WPA-PSK            ⑧
wpa_pairwise=CCMP               ⑨
```

① Interface sem fio usada para o ponto de acesso.

② Nível de detalhamento usado durante a execução de hostapd(8). Um valor de 1 representa o nível mínimo.

③ Nome do caminho de diretório usado pelo hostapd(8) para armazenar arquivos de soquete de domínio para comunicação com programas externos, como hostapd_cli(8). O valor padrão é usado neste exemplo.

④ O grupo permitiu acessar os arquivos da interface de controle.

⑤ O nome da rede sem fio, ou SSID, que aparecerá nas varreduras sem fio.

⑥ Ative o WPA e especifique qual protocolo de autenticação WPA será necessário. Um valor de 2 configura o AP para WPA2 e é recomendado. Defina como 1 apenas se o WPA obsoleto for necessário.

⑦ Senha ASCII para autenticação WPA.

⑧ O protocolo de gerenciamento de chaves a ser usado. Este exemplo define o WPA-PSK. Algoritmos de criptografia aceitos pelo ponto de acesso. Neste exemplo, apenas a codificação CCMP (AES) é aceita. O CCMP é uma alternativa ao TKIP e é fortemente preferido quando possível. O TKIP só deve ser permitido quando houver estações incapazes de usar o CCMP.

O próximo passo é iniciar hostapd(8):

```
# service hostapd forcestart
```

```
# ifconfig wlan0
wlan0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 04:f0:21:16:8e:10
    inet6 fe80::6f0:21ff:fe16:8e10%wlan0 prefixlen 64 scopeid 0x9
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    media: IEEE 802.11 Wireless Ethernet autoselect mode 11na <hostap>
    status: running
    ssid No5ignal channel 36 (5180 MHz 11a ht/40+) bssid 04:f0:21:16:8e:10
    country US ecm authmode WPA2/802.11i privacy MIXED deftxkey 2
    AES-CCM 2:128-bit AES-CCM 3:128-bit txpower 17 mcastrate 6 mgmtrate 6
    scanvalid 60 ampdulimit 64k ampdudensity 8 shortgi wme burst
    dtimperiod 1 -dfs
    groups: wlan
```

Quando o AP está em execução, os clientes podem associar-se a ele. Veja WPA para maiores detalhes. É possível ver as estações associadas ao AP usando o `ifconfig wlan0 list sta`.

31.3.6.4. Ponto de acesso baseado em host WEP

Não é recomendado o uso do WEP para configurar um AP, já que não há mecanismo de autenticação e a criptografia é facilmente quebrada. Algumas placas sem fio legadas suportam apenas o WEP e essas placas suportarão apenas um AP sem autenticação ou criptografia.

O dispositivo sem fio agora pode ser colocado no modo hostap e configurado com o endereço SSID e IP corretos:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 \
    ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g
```

- O `weptxkey` indica qual a chave WEP será usada na transmissão. Este exemplo usa a terceira chave, pois a numeração de chaves começa com `1`. Esse parâmetro deve ser especificado para criptografar os dados.
- O `wepkey` define a chave WEP selecionada. Ela deve estar no formato *index:key*. Se o índice não for fornecido, a chave `1` será configurada. O índice precisa ser definido ao usar chaves diferentes da primeira chave.

Use o ifconfig(8) para ver o status da interface wlan0:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 4 (2427 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit
      txpower 21.5 scanvalid 60 protmode CTS wme burst dtimperiod 1 -dfs
```

De uma outra máquina sem fio, agora é possível iniciar uma varredura para encontrar o AP:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 EPS
```

Neste exemplo, a máquina cliente encontrou o AP e pode associá-lo usando os parâmetros corretos. Veja WEP para maiores detalhes.

31.3.7. Usando conexões com fio e sem fio

Uma conexão com fio oferece melhor desempenho e confiabilidade, enquanto uma conexão sem fio fornece flexibilidade e mobilidade. Os usuários de laptop normalmente querem se movimentar perfeitamente entre os dois tipos de conexão.

No FreeBSD, é possível combinar duas ou mais interfaces de rede em um "failover". Esse tipo de configuração usa a conexão mais prioritária e disponível de um grupo de interfaces de rede, e o sistema operacional alterna automaticamente quando o estado do link é alterado.

A agregação de links e o failover são cobertos em Agregação de links e failover e um exemplo para usar conexões com e sem fio é fornecido em Modo de failover entre interfaces Ethernet e sem fio.

31.3.8. Solução de problemas

Esta seção descreve várias etapas para ajudar a solucionar problemas comuns de rede sem fio.

- Se o ponto de acesso não estiver listado durante a verificação, verifique se a configuração não limitou o dispositivo sem fio a um conjunto limitado de canais.
- Se o dispositivo não puder se associar a um ponto de acesso, verifique se a configuração corresponde às configurações no ponto de acesso. Isso inclui o esquema de autenticação e qualquer protocolo de segurança. Simplifique a configuração tanto quanto possível. Se estiver usando um protocolo de segurança, como o WPA ou o WEP, configure o ponto de acesso para autenticação aberta e nenhuma segurança para ver se o tráfego irá passar.

  O suporte a depuração é fornecido pelo wpa_supplicant(8). Tente executar este utilitário manualmente com a opção `-dd` e examine os logs do sistema.

- Uma vez que o sistema possa se associar com o ponto de acesso, diagnostique a configuração da rede usando ferramentas como o ping(8).
- Existem muitas ferramentas de depuração de nível inferior. As mensagens de depuração podem ser ativadas na camada de suporte do protocolo 802.11 usando o wlandebug(8). Por exemplo, para habilitar mensagens do console relacionadas à varredura de pontos de acesso e aos handshakes do protocolo 802.11 necessários para organizar a comunicação:

  ```
  # wlandebug -i wlan0 +scan+auth+debug+assoc
    net.wlan.0.debug: 0 => 0xc80000<assoc,auth,scan>
  ```

  Muitas estatísticas úteis são mantidas pela camada 802.11 e o `wlanstats`, encontrado em /usr/src/tools/tools/net80211, vai despejar esta informação. Essas estatísticas devem exibir todos os erros identificados pela camada 802.11. No entanto, alguns erros são identificados nos drivers de dispositivo que estão abaixo da camada 802.11, portanto eles podem não aparecer. Para diagnosticar problemas específicos do dispositivo, consulte a documentação do driver.

Se as informações acima não ajudarem a esclarecer o problema, envie um relatório de problemas e inclua a saída das ferramentas acima.

31.4. USB Tethering

Muitos telefones celulares oferecem a opção de compartilhar sua conexão de dados sobre o USB (muitas vezes chamado de "tethering"). Este recurso usa o RNDIS, CDC ou um protocolo personalizado Apple™iPhone™/iPad™.

- Os dispositivos Android™ geralmente utilizam o driver urndis(4).
- Os dispositivos Apple™ utilizam o driver ipheth(4).
- Dispositivos mais antigos geralmente utilizam o driver cdce(4).

Antes de conectar um dispositivo, carregue o driver apropriado no kernel:

```
# kldload if_urndis
# kldload if_cdce
# kldload if_ipheth
```

Uma vez que o dispositivo esteja conectado, ue*0* estará disponível para uso como um dispositivo de rede normal. Certifique-se de que a opção "USB Tethering" esteja ativada no dispositivo.

Para tornar essa alteração permanente e carregar o driver como um módulo no momento da inicialização, coloque a linha apropriada abaixo em /boot/loader.conf:

```
 if_urndis_load="YES"
if_cdce_load="YES"
if_ipheth_load="YES"
```

31.5. Bluetooth

O bluetooth é uma tecnologia sem fio para a criação de redes pessoais que operam na faixa não licenciada de 2,4 GHz, com um alcance de 10 metros. As redes geralmente são formadas em modo ad-hoc a partir de dispositivos portáteis, como telefones celulares, computadores de mão e laptops. Ao contrário da tecnologia sem fio Wi-Fi, o Bluetooth oferece perfis de serviços de nível superior, como servidores de arquivos semelhantes ao FTP, envio de arquivos, transporte de voz, emulação de linha serial e muito mais.

Esta seção descreve o uso de um dongle Bluetooth USB em um sistema FreeBSD. Em seguida, descreve os vários protocolos e utilitários Bluetooth.

31.5.1. Carregando o Suporte Bluetooth

A pilha Bluetooth no FreeBSD é implementada usando o framework netgraph(4). Uma ampla variedade de dongles Bluetooth USB é suportada pelo ng_ubt(4). Os dispositivos Bluetooth baseados no Broadcom BCM2033 são suportados pelos drivers ubtbcmfw(4) e ng_ubt(4). A placa 3Com Bluetooth PC Card 3CRWB60-A é suportada pelo driver ng_bt3c(4). Dispositivos Bluetooth baseados em Portas Seriais e UART são suportados por sio(4), ng_h4(4), e hcseriald(8).

Antes de conectar um dispositivo, determine qual dos drivers acima ele usa e, em seguida, carregue o driver. Por exemplo, se o dispositivo usar o driver ng_ubt(4):

```
# kldload ng_ubt
```

Se o dispositivo Bluetooth for conectado ao sistema durante a inicialização do sistema, o sistema pode ser configurado para carregar o módulo no momento da inicialização, adicionando o driver ao /boot/loader.conf:

```
ng_ubt_load="YES"
```

Quando o driver estiver carregado, conecte o dongle USB. Se a carga do driver tiver sido bem-sucedida, uma saída semelhante à seguinte deve aparecer no console e em /var/log/messages:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Para iniciar e parar a stack Bluetooth, use seu script de inicialização. É uma boa ideia parar a stack antes de desconectar o dispositivo. Iniciar a stack bluetooth pode exigir que o hcsecd(8) seja iniciado. Ao iniciar a stack, a saída deve ser semelhante à seguinte:

```
# service bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

31.5.2. Encontrando outros dispositivos Bluetooth

A Interface do Controlador do Host (HCI) fornece um método uniforme para acessar os recursos de banda básica do Bluetooth. No FreeBSD, um nó netgraph HCI é criado para cada dispositivo Bluetooth. Para mais detalhes, consulte ng_hci(4).

Uma das tarefas mais comuns é a descoberta de dispositivos Bluetooth dentro da proximidade RF. Esta operação é chamada *inquiry*. Investigação e outras operações relacionadas a HCI são feitas usando hccontrol(8). O exemplo abaixo mostra como descobrir quais dispositivos Bluetooth estão ao alcance. A lista de dispositivos deve ser exibida em alguns segundos. Note que um dispositivo remoto só irá responder a pergunta se estiver configurado para o modo *detectável*.

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
```

```
        BD_ADDR: 00:80:37:29:19:a4
        Page Scan Rep. Mode: 0x1
        Page Scan Period Mode: 00
        Page Scan Mode: 00
        Class: 52:02:04
        Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

O BD_ADDR é o endereço exclusivo de um dispositivo Bluetooth, semelhante ao endereço MAC de uma placa de rede. Este endereço é necessário para uma comunicação posterior com um dispositivo e é possível atribuir um nome legível a um BD_ADDR. Informações sobre os hosts Bluetooth conhecidos estão contidas em /etc/bluetooth/hosts. O exemplo a seguir mostra como obter o nome legível que foi atribuído ao dispositivo remoto:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

Se uma consulta for realizada em um dispositivo Bluetooth remoto, ele encontrará o computador como "your.host.name (ubt0)". O nome atribuído ao dispositivo local pode ser alterado a qualquer momento.

Dispositivos remotos podem receber aliases em /etc/bluetooth/hosts. Maiores informações sobre o arquivo /etc/bluetooth/hosts podem ser encontradas em bluetooth.hosts(5).

O sistema Bluetooth fornece uma conexão ponta-a-ponto entre duas unidades Bluetooth ou uma conexão ponto-a-multiponto que é compartilhada entre vários dispositivos Bluetooth. O exemplo a seguir mostra como criar uma conexão a um dispositivo remoto:

```
% hccontrol -n ubt0hci create_connection BT_ADDR
```

O create_connection aceita BT_ADDR, bem como aliases de host em /etc/bluetooth/hosts.

O exemplo a seguir mostra como obter a lista de conexões de banda base ativas para o dispositivo local:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

Um *identificador de conexão* é útil quando a finalização da conexão de banda base é necessária, embora normalmente não seja necessário fazer isso manualmente. A stack terminará automaticamente as conexões de banda básica inativas.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
```

```
Reason: Connection terminated by local host [0x16]
```

Digite `hccontrol help` para obter uma lista completa dos comandos HCI disponíveis. A maioria dos comandos HCI não requer privilégios de superusuário.

31.5.3. Emparelhamento de dispositivos

Por padrão, a comunicação Bluetooth não é autenticada e qualquer dispositivo pode conversar com qualquer outro dispositivo. Um dispositivo Bluetooth, como um telefone celular, pode optar por exigir autenticação para fornecer um serviço específico. A autenticação Bluetooth é normalmente feita com um *PIN code*, uma string ASCII com até 16 caracteres de comprimento. O usuário é obrigado a digitar o mesmo código de PIN em ambos os dispositivos. Depois que o usuário inserir o código de PIN, ambos os dispositivos gerarão uma *chave de link*. Depois disso, a chave de link pode ser armazenada nos dispositivos ou em um armazenamento persistente. Na próxima vez, os dois dispositivos usarão a chave de link gerada anteriormente. Este procedimento é chamado de *emparelhamento*. Observe que, se a chave de link for perdida por um dos dispositivos, o emparelhamento deverá ser repetido.

O daemon hcsecd(8) é responsável por tratar os pedidos de autenticação Bluetooth. O arquivo de configuração padrão é o /etc/bluetooth/hcsecd.conf. Uma seção de exemplo para um telefone celular com o código PIN definido como `1234` é mostrada abaixo:

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

A única limitação nos códigos de PIN é o comprimento. Alguns dispositivos, como fones de ouvido Bluetooth, podem ter um código PIN integrado fixo. A opção `-d` força o hcsecd(8) a ficar em primeiro plano, então é fácil ver o que está acontecendo. Configure o dispositivo remoto para receber o emparelhamento e inicie a conexão Bluetooth ao dispositivo remoto. O dispositivo remoto deve indicar que o pareamento foi aceito e solicitar o código de PIN. Digite o mesmo código de PIN listado em hcsecd.conf. Agora o computador e o dispositivo remoto estão emparelhados. Alternativamente, o emparelhamento pode ser iniciado no dispositivo remoto.

A seguinte linha pode ser adicionada ao /etc/rc.conf para configurar o hcsecd(8) para iniciar automaticamente quando o sistema inicializar:

```
hcsecd_enable="YES"
```

A seguir, um exemplo da saída do daemon hcsecd(8):

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
```

```
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

31.5.4. Acesso à rede com perfis PPP

Um perfil de rede dial-up (DUN) pode ser usado para configurar um telefone celular como um modem sem fio para a conexão a um servidor de acesso à Internet dial-up. Também pode ser usado para configurar um computador para receber chamadas de dados de um telefone celular.

O acesso à rede com um perfil PPP pode ser usado para fornecer acesso LAN a um único dispositivo Bluetooth ou a vários dispositivos Bluetooth. Ele também pode fornecer uma conexão PC para PC usando uma rede PPP sobre uma emulação de cabo serial.

No FreeBSD, esses perfis são implementados com o ppp(8) e o wrapper rfcomm_pppd(8) que converte uma conexão Bluetooth em algo que o PPP pode usar. Antes que um perfil possa ser usado, um novo label PPP deve ser criado em /etc/ppp/ppp.conf. Consulte rfcomm_pppd(8) para exemplos.

Neste exemplo, o rfcomm_pppd(8) é usado para abrir uma conexão com um dispositivo remoto com um `BD_ADDR` de `00:80:37:29:19:a4` em um canal DUNRFCOMM:

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

O número real do canal será obtido a partir do dispositivo remoto usando o protocolo SDP. É possível especificar manualmente o canal RFCOMM e, nesse caso, o rfcomm_pppd(8) não executará a consulta SDP. Use o sdpcontrol(8) para descobrir o canal RFCOMM no dispositivo remoto.

Para fornecer acesso à rede com o serviço PPPLAN, o sdpd(8) precisa estar sendo executado e uma nova entrada para clientes LAN deve ser criada em /etc/ppp/ppp.conf. Consulte rfcomm_pppd(8) para exemplos. Por fim, inicie o servidor RFCOMMPPP em um número de canal RFCOMM válido. O servidor RFCOMMPPP registrará automaticamente o serviço Bluetooth LAN com o daemon local SDP. O exemplo abaixo mostra como iniciar o servidor RFCOMMPPP.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

31.5.5. Protocolos Bluetooth

Esta seção fornece uma visão geral dos vários protocolos Bluetooth, suas funções e utilitários associados.

31.5.5.1. Controle de Link Lógico e Protocolo de Adaptação (L2CAP)

O Protocolo de Adaptação e Controle de Link Lógico (L2CAP) fornece serviços de dados orientados a conexão e sem conexão para protocolos de camada superior. O L2CAP permite que protocolos e aplicativos de alto nível transmitam e recebam pacotes de dados L2CAP de até 64 kilobytes de comprimento.

O L2CAP é baseado no conceito de *canais*. Um canal é uma conexão lógica em cima de uma conexão de banda base, na qual cada canal é vinculado a um único protocolo de maneira many-to-one. Vários canais podem ser vinculados ao mesmo protocolo, mas um canal não pode ser vinculado a vários protocolos. Cada pacote L2CAP recebido em um canal é direcionado para o protocolo apropriado de nível superior. Vários canais podem compartilhar a mesma conexão de banda base.

No FreeBSD, um nó netgraph L2CAP é criado para cada dispositivo Bluetooth. Esse nó é normalmente conectado ao nó Bluetooth HCI downstream e aos nós de soquete Bluetooth upstream. O nome padrão para o nó L2CAP é "devicel2cap". Para mais detalhes, consulte ng_l2cap(4).

Um comando útil é o l2ping(8), que pode ser usado para executar ping em outros dispositivos. Algumas implementações Bluetooth podem não retornar todos os dados enviados para elas, portanto, a saída `0 bytes` no exemplo a seguir é normal.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

O utilitário l2control(8) é usado para executar várias operações em nós L2CAP. Este exemplo mostra como obter a lista de conexões lógicas (canais) e a lista de conexões de banda base para o dispositivo local:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

Outra ferramenta de diagnóstico é o btsockstat(1). Ele é semelhante ao netstat(1), mas para estruturas de dados relacionadas à rede Bluetooth. O exemplo abaixo mostra a mesma conexão lógica que l2control(8) acima.

```
% btsockstat
Active L2CAP sockets
```

```
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

31.5.5.2. Comunicação por radiofrequência (RFCOMM)

O protocolo RFCOMM fornece emulação de portas seriais sobre o protocolo L2CAP. O RFCOMM é um protocolo de transporte simples, com disposições adicionais para emular os 9 circuitos das portas seriais RS-232 (EIATIA-232-E). Suporta até 60 conexões simultâneas (canais RFCOMM) entre dois dispositivos Bluetooth.

Para fins do RFCOMM, um caminho de comunicação completo envolve dois aplicativos em execução nos terminais de comunicação com um segmento de comunicação entre eles. O RFCOMM destina-se a abranger aplicativos que fazem uso das portas seriais dos dispositivos em que residem. O segmento de comunicação é um link Bluetooth de conexão direta de um dispositivo para outro.

O RFCOMM está relacionado apenas com a conexão entre os dispositivos no caso de conexão direta ou entre o dispositivo e um modem no caso de rede. O RFCOMM pode suportar outras configurações, como módulos que se comunicam via tecnologia sem fio Bluetooth de um lado e fornecem uma interface com fio no outro lado.

No FreeBSD, o RFCOMM é implementado na camada de sockets do Bluetooth.

31.5.5.3. Protocolo de Descoberta de Serviços (SDP)

O Protocolo de Descoberta de Serviços (SDP) fornece os meios para os aplicativos clientes descobrirem a existência de serviços fornecidos por aplicativos de servidor, bem como os atributos desses serviços. Os atributos de um serviço incluem o tipo ou classe de serviço oferecido e as informações de mecanismo ou protocolo necessárias para utilizar o serviço.

O SDP envolve a comunicação entre um servidor SDP e um cliente SDP. O servidor mantém uma lista de registros de serviço que descrevem as características dos serviços associados ao servidor. Cada registro de serviço contém informações sobre um único serviço. Um cliente pode recuperar informações de um registro de serviço mantido pelo servidor SDP emitindo uma solicitação SDP. Se o cliente, ou um aplicativo associado ao cliente, decidir usar um serviço, ele deverá abrir uma conexão separada com o provedor de serviços para utilizar o serviço. O SDP fornece um mecanismo para descobrir serviços e seus atributos, mas não fornece um mecanismo para utilizar esses serviços.

Normalmente, um cliente SDP procura serviços baseados em algumas características desejadas dos serviços. No entanto, há momentos em que é desejável descobrir quais tipos de serviços são descritos pelos registros de serviço de um servidor SDP, sem qualquer informação prévia sobre os serviços. Este processo de procurar por qualquer serviço oferecido é chamado de *navegação*.

O servidor Bluetooth SDP, sdpd(8) e o cliente de linha de comandos, sdpcontrol(8), estão incluídos

na instalação padrão do FreeBSD. O exemplo a seguir mostra como executar uma consulta de navegação SDP.

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
        LAN Access Using PPP (0x1102) ver. 1.0
```

Observe que cada serviço tem uma lista de atributos, como o canal RFCOMM. Dependendo do serviço, o usuário pode precisar anotar alguns dos atributos. Algumas implementações Bluetooth não suportam a navegação de serviço e podem retornar uma lista vazia. Nesse caso, é possível procurar pelo serviço específico. O exemplo abaixo mostra como pesquisar o serviço OBEX Object Push (OPUSH) :

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

A oferta de serviços no FreeBSD para clientes Bluetooth é feita com o servidor sdpd(8). A seguinte linha pode ser adicionada ao /etc/rc.conf:

```
sdpd_enable="YES"
```

Então o daemon sdpd(8) pode ser iniciado com:

```
# service sdpd start
```

O aplicativo de servidor local que deseja fornecer um serviço Bluetooth a clientes remotos registrará o serviço com o daemon SDP local . Um exemplo de tal aplicativo é o rfcomm_pppd(8). Uma vez iniciado, ele registrará o serviço LAN Bluetooth com o daemon local SDP.

A lista de serviços registrados no servidor SDPlocal pode ser obtida através da emissão de uma consulta de navegação SDP através do canal de controle local:

```
# sdpcontrol -l browse
```

31.5.5.4. OBEX Object Push (OPUSH)

Object Exchange (OBEX) é um protocolo amplamente utilizado para transferências de arquivos simples entre dispositivos móveis. Seu principal uso é na comunicação por infravermelho, onde é usado para transferências de arquivos genéricos entre notebooks ou PDAs, e para enviar cartões de visita ou entradas de calendário entre telefones celulares e outros dispositivos com Personal Information Manager (PIM).

O servidor e o cliente OBEX são implementados pelo obexapp, que pode ser instalado usando o pacote ou port comms/obexapp.

O cliente OBEX é usado para empurrar e/ou puxar objetos do servidor OBEX. Um exemplo de objeto é um cartão de visita ou um compromisso. O cliente OBEX pode obter o número do canal RFCOMM do dispositivo remoto via SDP. Isso pode ser feito especificando o nome do serviço em vez do número do canal RFCOMM. Os nomes de serviços suportados são: `IrMC`, `FTRN` e `OPUSH`. Também é possível especificar o canal RFCOMM como um número. Abaixo está um exemplo de uma sessão OBEX em que o objeto de informações do dispositivo é extraído do telefone celular e um novo objeto, o cartão de visita, é inserido no diretório do telefone.

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Para fornecer o serviço OPUSH, o sdpd(8) deve estar em execução e uma pasta raiz, onde todos os objetos recebidos serão armazenados, deve ser criado. O caminho padrão para a pasta raiz é /var/spool/obex. Por fim, inicie o servidor OBEX em um número de canal RFCOMM válido. O servidor OBEX registrará automaticamente o serviço OPUSH com o daemon SDP local. O exemplo abaixo mostra como iniciar o servidor OBEX.

```
# obexapp -s -C 10
```

31.5.5.5. Perfil de porta serial (SPP)

O perfil de porta serial (SPP) permite que dispositivos Bluetooth executem emulação de cabo serial. Este perfil permite que aplicativos legados usem o Bluetooth como um substituto de cabos, através de uma abstração de porta serial virtual.

No FreeBSD, o rfcomm_sppd(1) implementa o SPP e uma pseudo tty é usada como uma abstração de porta serial virtual. O exemplo abaixo mostra como se conectar ao serviço de porta serial de um dispositivo remoto. Um canal RFCOMM não precisa ser especificado uma vez que o rfcomm_sppd(1) pode obtê-lo a partir do dispositivo remoto via SDP. Para sobrescrever isso, especifique um canal RFCOMM na linha de comando.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t
rfcomm_sppd[94692]: Starting on /dev/pts/6...
/dev/pts/6
```

Uma vez conectado, o pseudo-tty pode ser usado como porta serial:

```
# cu -l /dev/pts/6
```

A pseudo-tty é impressa no stdout e pode ser lida por scripts de wrapper:

```
PTS=`rfcomm_sppd -a 00:07:E0:00:0B:CA -t`
cu -l $PTS
```

31.5.6. Solução de problemas

Por padrão, quando o FreeBSD está aceitando uma nova conexão, ele tenta executar uma troca de função e se tornar o mestre. Alguns dispositivos Bluetooth mais antigos que não suportam a troca de função não poderão se conectar. Como a troca de função é executada quando uma nova conexão está sendo estabelecida, não é possível perguntar ao dispositivo remoto se ele suporta a troca de função. No entanto, há uma opção HCI para desativar a alternância de funções no lado local:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

Para exibir pacotes Bluetooth, use o pacote de terceiros hcidump, que pode ser instalado usando o pacote ou port comms/hcidump. Este utilitário é semelhante ao tcpdump(1) e pode ser usado para exibir o conteúdo dos pacotes Bluetooth no terminal e para descarregar os pacotes Bluetooth para um arquivo.

31.6. Bridging

Às vezes, é útil dividir uma rede, como um segmento Ethernet, em segmentos de rede sem precisar criar subnets IP e usar um roteador para conectar os segmentos. Um dispositivo que conecta duas redes dessa maneira é chamado de "bridge".

Uma bridge funciona aprendendo os endereços MAC dos dispositivos em cada uma das suas interfaces de rede. Ele encaminha o tráfego entre as redes somente quando os endereços de origem e destino MAC estão em redes diferentes. Em muitos aspectos, uma brifge é como um switch Ethernet com poucas portas. Um sistema FreeBSD com múltiplas interfaces de rede pode ser

configurado para atuar como uma bridge.

Construir uma bridge pode ser útil nas seguintes situações:

Conectar Redes

A operação básica de uma bridge é unir dois ou mais segmentos de rede. Existem muitas razões para usar uma bridge baseada em host em vez de equipamentos de rede, tais como restrições de cabeamento ou firewall. Uma bridge também pode conectar uma interface sem fio em execução no modo hostap a uma rede com fio e atuar como um ponto de acesso.

Firewall de Filtragem / Limitação de Trafego

Uma bridge pode ser usada quando a funcionalidade de firewall é necessária sem a realização de roteamento ou conversão de endereços de rede (NAT).

Um exemplo é uma pequena empresa conectada via DSL ou ISDN a um ISP. Existem treze endereços IP públicos do ISP e dez computadores na rede. Nessa situação, é difícil usar um firewall baseado em roteador devido a problemas de sub-rede. Um firewall baseado em bridge pode ser configurado sem qualquer problema de endereçamento IP.

Inspeção de Rede

Uma bridge pode unir dois segmentos de rede para inspecionar todos os pacotes Ethernet que passam entre elas usando bpf(4) e tcpdump(1) na interface de bridge ou enviando uma cópia de todos os frames para uma interface adicional conhecida como span port.

VPN de Camada 2

Duas redes Ethernet podem ser unidas através de um link IP ligando as redes a um túnel EtherIP ou a uma solução baseada no tap(4) tal como o OpenVPN.

Redundância de Camada 2

Uma rede pode ser conectada com vários links e usar o protocolo Spanning Tree (STP) para bloquear caminhos redundantes.

Esta seção descreve como configurar um sistema FreeBSD como uma bridge usando o if_bridge(4). Um driver de bridge netgraph também está disponível e é descrito em ng_bridge(4).



A filtragem de pacotes pode ser usada com qualquer pacote de firewall que se conecte ao framework pfil(9). A bridge pode ser usada como um modelador de tráfego com o altq(4) ou dummynet(4).

31.6.1. Habilitando a Bridge

No FreeBSD, o if_bridge(4) é um módulo do kernel que é carregado automaticamente pelo ifconfig(8) ao criar uma interface de bridge. Também é possível compilar o suporte de bridge em um kernel customizado adicionando `device if_bridge` ao arquivo de configuração do kernel personalizado.

A bridge é criada usando clonagem de interface. Para criar a interface da bridge:

```
# ifconfig bridge create
bridge0
# ifconfig bridge0
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

Quando uma interface de bridge é criada, ela recebe automaticamente um endereço Ethernet gerado aleatoriamente. Os parâmetros `maxaddr` e `timeout` controlam quantos endereços MAC a bridge manterá em sua tabela de encaminhamento e quantos segundos o sistema irá esperar antes de cada entrada ser removida após um endereço MAC ser visto pela última vez. Os outros parâmetros controlam como o STP opera.

Em seguida, especifique quais interfaces de rede adicionar como membros da bridge. Para a bridge encaminhar pacotes, todas as interfaces de membros e a bridge precisam estar ativas:

```
# ifconfig bridge0 addm fxp0 addm fxp1 up
# ifconfig fxp0 up
# ifconfig fxp1 up
```

A bridge agora pode encaminhar quadros Ethernet entre fxp0 e fxp1. Adicione as seguintes linhas ao /etc/rc.conf para que a bridge seja criada na inicialização:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

Se o host de ponte precisar de um endereço IP, defina-o na interface de bridge, não nas interfaces de membro. O endereço pode ser definido estaticamente ou via DHCP. Este exemplo define um endereço IP estático:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

Também é possível atribuir um endereço IPv6 a uma interface de bridge. Para tornar as mudanças permanentes, adicione as informações de endereçamento ao /etc/rc.conf.



Quando a filtragem de pacotes está habilitada, os pacotes passarão pela entrada do filtro na interface de origem na interface da bridge e na saída nas interfaces apropriadas. Qualquer estágio pode ser desativado. Quando a direção do fluxo de pacotes é importante, é melhor usar o firewall nas interfaces de membros, em vez da própria bridge.

> A bridge tem várias opções configuráveis para o trafego de pacotes IP e não-IP, e a filtragem de pacotes layer2 com o ipfw(8). Veja if_bridge(4) para maiores informações.

31.6.2. Ativando o Spanning Tree

Para que uma rede Ethernet funcione corretamente, somente um caminho ativo pode existir entre dois dispositivos. O protocolo STP detecta loops e coloca links redundantes em um estado bloqueado. Se um dos links ativos falhar, o STP calcula uma árvore diferente e habilita um dos caminhos bloqueados para restaurar a conectividade a todos os pontos da rede.

O protocolo Rapid Spanning Tree (RSTP ou 802.1w) fornece compatibilidade retroativa com o STP legado. O RSTP fornece uma convergência mais rápida e troca informações com os switches vizinhos para fazer a transição rápida para o modo de encaminhamento sem criar loops. O FreeBSD suporta o RSTP e o STP como modos de operação, com o RSTP sendo o modo padrão.

O STP pode ser ativado nas interfaces de membro usando o ifconfig(8). Para uma bridge com fxp0 e fxp1 como as interfaces atuais, ative o STP com:

```
# ifconfig bridge0 stp fxp0 stp fxp1
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether d6:cf:d5:a0:94:6d
        id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 3 priority 128 path cost 200000 proto rstp
                role designated state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

Essa ponte possui um spanning tree ID de `00:01:02:4b:d4:50` e uma prioridade de `32768`. Como o `root id` é o mesmo, indica que esta é a bridge raiz para a árvore.

Outra bridge na rede também tem o STP ativado:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role root state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 5 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

A linha `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` mostra que a bridge raiz é `00:01:02:4b:d4:50` e tem um custo de caminho de `400000` desta bridge. O caminho para a bridge raiz é via `port 4`, que é fxp0.

31.6.3. Parâmetros da Interface de Bridge

Vários parâmetros do `ifconfig` são exclusivos para interligar interfaces. Esta seção resume alguns usos comuns para esses parâmetros. A lista completa de parâmetros disponíveis é descrita em ifconfig(8).

privado

Uma interface privada não encaminha qualquer tráfego para qualquer outra porta que também seja designada como uma interface privada. O tráfego é bloqueado incondicionalmente para que nenhum quadro Ethernet seja encaminhado, incluindo pacotes ARP. Se o tráfego precisar ser bloqueado seletivamente, um firewall deve ser usado no lugar.

span

Uma porta span transmite uma cópia de cada quadro Ethernet recebido pela bridge. O número de portas de span configuradas em uma bridge é ilimitado, mas se uma interface for designada como uma porta de span, ela também não poderá ser usada como uma porta de bridge comum. Isso é mais útil para espionar passivamente uma rede em bridge a partir de outro host conectado a uma das portas da bridge. Por exemplo, para enviar uma cópia de todos os quadros para fora da interface denominada fxp4:

```
# ifconfig bridge0 span fxp4
```

sticky

Se uma interface de membro de uma bridge estiver marcada como fixa, as entradas de endereço aprendidas dinamicamente serão tratadas como entradas estáticas no cache de encaminhamento. Entradas fixas nunca são eliminadas do cache ou substituídas, mesmo que o endereço seja visto em uma interface diferente. Isso oferece o benefício de entradas de endereço estático sem a necessidade de preencher previamente a tabela de encaminhamento. Os clientes aprendidos em um segmento específico da bridge não podem se deslocar para outro segmento.

Um exemplo do uso de endereços fixos é combinar a bridge com VLANs para isolar redes de clientes sem desperdiçar espaço de endereço IP. Considere que `CustomerA` está em `vlan100`, `CustomerB` está em `vlan101`, e a bridge tem o endereço `192.168.0.1`:

```
# ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101 sticky vlan101
# ifconfig bridge0 inet 192.168.0.1/24
```

Neste exemplo, os dois clientes vêem `192.168.0.1` como seu gateway padrão. Como o cache da bridge é fixo, um host não pode falsificar o endereço MAC do outro cliente para interceptar o tráfego.

Qualquer comunicação entre as VLANs pode ser bloqueada usando um firewall ou, como visto

neste exemplo, interfaces privadas:

```
# ifconfig bridge0 private vlan100 private vlan101
```

Os clientes são completamente isolados uns dos outros e o intervalo completo de endereços `/24` pode ser alocado sem criação de sub-redes.

O número de endereços MAC de origem exclusivos por trás de uma interface pode ser limitado. Quando o limite é atingido, os pacotes com endereços de origem desconhecidos são descartados até que uma entrada de cache do host existente expire ou seja removida.

O exemplo a seguir define o número máximo de dispositivos Ethernet para `CustomerA` em `vlan100` para 10:

```
# ifconfig bridge0 ifmaxaddr vlan100 10
```

As interfaces de bridge também suportam o modo monitor, onde os pacotes são descartados após processamento do bpf(4) e não são processados ou encaminhados. Isso pode ser usado para multiplexar a entrada de duas ou mais interfaces em um único fluxo bpf(4). Isso é útil para reconstruir o tráfego de taps de rede que transmitem os sinais RX/TX através de duas interfaces separadas. Por exemplo, para ler a entrada de quatro interfaces de rede como um fluxo:

```
# ifconfig bridge0 addm fxp0 addm fxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

31.6.4. Monitoramento SNMP

A interface de bridge e os parâmetros de STP podem ser monitorados via o bsnmpd(1) o qual está incluído no sistema básico do FreeBSD. A MIB exportada da bridge está em conformidade com os padrões IETF, portanto, qualquer cliente ou pacote de monitoramento SNMP pode ser usado para recuperar os dados.

Para ativar o monitoramento na bridge, descomente esta linha em /etc/snmpd.config removendo o símbolo inicial `#`:

```
begemotSnmpdModulePath."bridge" = "/usr/lib/snmp_bridge.so"
```

Outras configurações, como nomes de comunidades e listas de acesso, podem precisar ser modificadas nesse arquivo. Consulte bsnmpd(1) e snmp_bridge(3) para maiores informações. Depois que essas edições forem salvas, adicione esta linha ao /etc/rc.conf:

```
bsnmpd_enable="YES"
```

Em seguida, inicie o bsnmpd(1):

```
# service bsnmpd start
```

Os exemplos a seguir usam o software Net-SNMP (net-mgmt/net-snmp) para consultar uma bridge a partir de um sistema cliente. O port net-mgmt/bsnmptools também pode ser usado. Do cliente SNMP que está executando o Net-SNMP, adicione as seguintes linhas ao $HOME/.snmp/snmp.conf para importar as definições da bridge MIB:

```
mibdirs +/usr/shared/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

Para monitorar uma única bridge usando o IETF BRIDGE-MIB (RFC4188):

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centi-
seconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
...
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding(5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled(1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp(2)
```

O valor `dot1dStpTopChanges.0` é dois, indicando que a topologia da bridge STP foi alterada duas vezes. Uma alteração de topologia significa que um ou mais links na rede foram alterados ou falharam e uma nova árvore foi calculada. O valor de `dot1dStpTimeSinceTopologyChange.0` será exibido quando isso acontecer.

Para monitorar várias interfaces de bridge, o BEGEMOT-BRIDGE-MIB privado pode ser usado:

```
% snmpwalk -v 2c -c public bridge1.example.com
enterprises.fokus.begemot.begemotBridge
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge0" = STRING: bridge0
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge2" = STRING: bridge2
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge0" = STRING: e:ce:3b:5a:9e:13
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge2" = STRING: 12:5e:4d:74:d:fc
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge0" = INTEGER: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge2" = INTEGER: 1
...
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge0" = Timeticks:
(116927) 0:19:29.27 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge2" = Timeticks:
(82773) 0:13:47.73 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge0" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge2" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge0" = Hex-STRING: 80 00 00 40
95 30 5E 31
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge2" = Hex-STRING: 80 00 00 50
8B B8 C6 A9
```

Para alterar a interface da bridge que está sendo monitorada através da subárvore `mib-2.dot1dBridge`:

```
% snmpset -v 2c -c private bridge1.example.com
BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 s bridge2
```

31.7. Agregação de links e failover

O FreeBSD fornece a interface lagg(4) que pode ser usada para agregar várias interfaces de rede em uma interface virtual para fornecer failover e agregação de links. O failover permite que o tráfego continue a fluir, desde que pelo menos uma interface de rede agregada tenha um link estabelecido. A agregação de links funciona melhor em switches compatíveis com LACP, pois esse protocolo distribui o tráfego bidirecionalmente ao responder à falha de links individuais.

Os protocolos de agregação suportados pela interface lagg determinam quais portas são usadas para o tráfego de saída e se uma porta específica aceita tráfego de entrada. Os seguintes protocolos são suportados pelo lagg(4):

failover

Este modo envia e recebe tráfego somente através da porta principal. Se a porta principal ficar indisponível, a próxima porta ativa será usada. A primeira interface adicionada à interface virtual é a porta principal e todas as interfaces adicionadas posteriormente são usadas como dispositivos de failover. Se ocorrer um failover em uma porta não mestre, a porta original se tornará a principal quando estiver disponível novamente.

fec / loadbalance

Cisco™ Fast EtherChannel™ (FEC) é encontrado em versões anteriores de switches Cisco ™. Ele fornece uma configuração estática e não negocia a agregação com o par ou troca quadros para monitorar o link. Se o switch suportar LACP, isso deve ser usado em seu lugar.

lacp

O protocolo de controle de agregação de links IEEE™ 802.3ad (LACP) negocia um conjunto de links agregáveis com o peer em um ou mais grupos agregados de links (LAGs). Cada LAG é composto de portas da mesma velocidade, configuradas para operação full-duplex e o tráfego é balanceado entre as portas no LAG com a maior velocidade total. Normalmente, há apenas um LAG que contém todas as portas. No caso de alterações na conectividade física, o LACP

convergirá rapidamente para uma nova configuração.

O LACP equilibra o tráfego de saída nas portas ativas com base nas informações de hash do cabeçalho do protocolo e aceita tráfego de entrada de qualquer porta ativa. O hash inclui o endereço Ethernet de origem e destino e, se disponível, a tag VLAN e o endereço de origem e destino IPv4 ou IPv6.

roundrobin

Esse modo distribui o tráfego de saída usando um agendador round-robin por meio de todas as portas ativas e aceita tráfego de entrada de qualquer porta ativa. Como esse modo viola a ordenação de quadros Ethernet, ele deve ser usado com cautela.

31.7.1. Exemplos de configuração

Esta seção demonstra como configurar um switch Cisco™ e um sistema FreeBSD para balanceamento de carga LACP. Em seguida, ele mostra como configurar duas interfaces Ethernet no modo de failover, além de como configurar o modo de failover entre uma Ethernet e uma interface sem fio.

Exemplo 48. Agregação LACP com um switch Cisco™

Este exemplo conecta duas interfaces Ethernet fxp(4) em uma máquina FreeBSD às duas primeiras portas Ethernet em um switch Cisco™ como um link de carga única balanceada e tolerante a falhas. Mais interfaces podem ser adicionadas para aumentar o rendimento e a tolerância a falhas. Substitua os nomes das portas Cisco™, dos dispositivos Ethernet, do número do grupo de canais e do endereço IP mostrado no exemplo para corresponder à configuração local.

A ordenação de quadros é obrigatória em links Ethernet e qualquer tráfego entre duas estações sempre flui pelo mesmo link físico, limitando a velocidade máxima àquela de uma interface. O algoritmo de transmissão tenta usar o máximo de informações possível para distinguir diferentes fluxos de tráfego e equilibrar os fluxos entre as interfaces disponíveis.

No switch Cisco™, adicione as interfaces *FastEthernet0/1* e *FastEthernet0/2* ao grupo de canais *1*:

```
 interface FastEthernet0/1
 channel-group 1 mode active
 channel-protocol lacp
!
interface FastEthernet0/2
 channel-group 1 mode active
 channel-protocol lacp
```

No sistema FreeBSD, crie a interface lagg(4) usando as interfaces físicas *fxp0* e *fxp1* e suba as interfaces com o endereço IP de *10.0.0.3/24*:

```
# ifconfig fxp0 up
```

```
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24
```

Em seguida, verifique o status da interface virtual:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        inet 10.0.0.3 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet autoselect
        status: active
        laggproto lacp
        laggport: fxp1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: fxp0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

Portas marcadas como `ACTIVE` fazem parte do LAG que foi negociado com o switch remoto. O tráfego será transmitido e recebido através dessas portas ativas. Adicione `-v` ao comando acima para ver os identificadores LAG.

Para ver o status da porta no switch Cisco™:

```
switch# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

                LACP port                       Oper    Port     Port
Port      Flags   Priority  Dev ID         Age     Key     Number   State
Fa0/1     SA      32768     0005.5d71.8db8  29s    0x146   0x3      0x3D
Fa0/2     SA      32768     0005.5d71.8db8  29s    0x146   0x4      0x3D
```

Para mais detalhes, digite `show lacp neighbor detail`.

Para manter esta configuração através de reinicializações, adicione as seguintes entradas ao /etc/rc.conf no sistema FreeBSD:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24"
```

Exemplo 49. Modo de Failover

O modo de failover pode ser usado para alternar para uma interface secundária se o link for perdido na interface principal. Para configurar o failover, certifique-se de que as interfaces físicas subjacentes estejam ativadas e crie a interface lagg(4). Neste exemplo, *fxp0* é a interface principal, *fxp1* é a interface secundária e a interface virtual recebeu um endereço IP de *10.0.0.15/24*:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24
```

A interface virtual deve ser algo como isto:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        inet 10.0.0.15 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet autoselect
        status: active
        laggproto failover
        laggport: fxp1 flags=0<>
        laggport: fxp0 flags=5<MASTER,ACTIVE>
```

O tráfego será transmitido e recebido em *fxp0*. Se o link for perdido em *fxp0*, *fxp1* se tornará o link ativo. Se o link for restaurado na interface principal, ele se tornará novamente o link ativo.

Para manter essa configuração através de reinicializações, adicione as seguintes entradas ao /etc/rc.conf:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24"
```

Exemplo 50. Modo de failover entre interfaces Ethernet e sem fio

Para usuários de laptop, geralmente é desejável configurar o dispositivo sem fio como secundário, que é usado somente quando a conexão Ethernet não está disponível. Com lagg(4), é possível configurar um failover que preferia a conexão Ethernet por motivos de desempenho e de segurança, mantendo a capacidade de transferência dados através da conexão sem fio.

Isso é obtido substituindo o endereço MAC da interface Ethernet com o da interface wireless.



Em teoria, o endereço MAC da Ethernet ou da wireless pode ser alterado para corresponder ao outro. No entanto, algumas interfaces wireless populares não têm suporte para substituir o endereço MAC. Portanto, recomendamos substituir o endereço MAC da Ethernet para esse fim.



Se o driver para a interface wireless não estiver carregado no kernel `GENERIC` ou customizado, e o computador estiver rodando o FreeBSD 12.1, carregue o .ko correspondente no arquivo /boot/loader.conf adicionando `_driver__load="YES"` e reiniciando a maquina. Outra forma melhor, é carregar o driver no arquivo /etc/rc.conf adicionando a variável `kld_list` (veja rc.conf(5) para maiores detalhes) nesse arquivo e reiniciar. Isso é necessário porque de outra forma o driver não estará carregado no tempo em que a interface lagg(4) for configurada.

Neste exemplo, a interface Ethernet, *re0*, é a interface principal e a interface sem fio, *wlan0*, é o failover. A interface *wlan0* foi criada a partir da interface wireless *ath0*, e a interface Ethernet será configurada com o endereço MAC da interface wireless. Primeiro, determine o endereço MAC da interface wireless:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether b8:ee:65:5b:32:59
    groups: wlan
    ssid Bbox-A3BD2403 channel 6 (2437 MHz 11g ht/20) bssid 00:37:b7:56:4b:60
    regdomain ETSI country FR indoor ecm authmode WPA2/802.11i privacy ON
    deftxkey UNDEF AES-CCM 2:128-bit txpower 30 bmiss 7 scanvalid 60
    protmode CTS ampdulimit 64k ampdudensity 8 shortgi -stbctx stbcrx
    -ldpc wme burst roaming MANUAL
    media: IEEE 802.11 Wireless Ethernet MCS mode 11ng
    status: associated
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

Substitua *wlan0* para corresponder ao nome da interface wireless do sistema. A linha `ether` conterá o endereço MAC da interface especificada. Agora, altere o endereço MAC da interface Ethernet subjacente:

```
# ifconfig re0 ether b8:ee:65:5b:32:59
```

Suba a interface sem fio (substituindo *FR* pelo seu próprio código de país com duas letras), mas não defina um endereço IP:

```
# ifconfig wlan0 create wlandev ath0 country FR ssid my_router up
```

Certifique-se de que a interface *re0* esteja ativa, então crie a interface lagg(4) com a *re0* como master com failover para a_wlan0_:

```
# ifconfig re0 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport re0 laggport wlan0
```

A interface virtual deve ser algo como isto:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether b8:ee:65:5b:32:59
        laggproto failover lagghash l2,l3,l4
        laggport: re0 flags=5<MASTER,ACTIVE>
        laggport: wlan0 flags=0<>
        groups: lagg
        media: Ethernet autoselect
        status: active
```

Em seguida, inicie o cliente DHCP para obter um endereço IP:

```
# dhclient lagg0
```

Para manter essa configuração através de reinicializações, adicione as seguintes entradas ao /etc/rc.conf:

```
ifconfig_re0="ether b8:ee:65:5b:32:59"
wlans_ath0="wlan0"
ifconfig_wlan0="WPA"
create_args_wlan0="country FR"
cloned_interfaces="lagg0"
ifconfig_lagg0="up laggproto failover laggport re0 laggport wlan0 DHCP"
```

31.8. Operação Diskless com PXE

O Ambiente de execução de pré-inicialização da Intel™ (PXE) permite que um sistema operacional inicialize pela rede. Por exemplo, um sistema FreeBSD pode inicializar através da rede e operar sem um disco local, usando sistemas de arquivos montados a partir de um servidor NFS. O suporte para PXE geralmente está disponível no BIOS. Para usar o PXE quando a máquina iniciar, selecione a opção `Inicialização da rede` na configuração do BIOS ou digite uma tecla de função durante a inicialização do sistema.

Para fornecer os arquivos necessários para um sistema operacional inicializar pela rede, uma configuração do PXE também requer o DHCP, TFTP configurado corretamente e Servidores NFS, onde:

- Parâmetros iniciais, como endereço de IP, nome e localização do arquivo de inicialização executável, nome do servidor e caminho do root são obtidos do servidor DHCP.
- O arquivo do carregador do sistema operacional é inicializado usando TFTP.
- Os sistemas de arquivos são carregados usando o NFS.

Quando um computador PXE inicializa, ele recebe informações por meio do DHCP sobre onde obter o arquivo inicial do carregador de boot. Depois que o computador host recebe essa informação, ele faz o download do carregador de boot via TFTP e, em seguida, executa o carregador de boot. No FreeBSD, o arquivo do gerenciador de boot é o /boot/pxeboot. Depois que o /boot/pxeboot é executado, o kernel do FreeBSD é carregado e o resto da seqüência de inicialização do FreeBSD continua, como descrito em O processo de inicialização do FreeBSD.

Esta seção descreve como configurar estes serviços em um sistema FreeBSD para que outros sistemas possam inicializar o PXE a partir do FreeBSD. Consulte diskless(8) para obter maiores informações.



Conforme descrito, o sistema que fornece esses serviços é inseguro. Ele deve ficar em uma área protegida de uma rede e não deve ser considerado confiável por outros hosts.

31.8.1. Configurando o ambiente PXE

As etapas mostradas nesta seção configuram os servidores internos de NFS e TFTP. A próxima seção demonstra como instalar e configurar o servidor DHCP. Neste exemplo, o diretório que conterá os arquivos usados pelos usuários do PXE é o /b/tftpboot/FreeBSD/install. É importante que este diretório exista e que o mesmo nome de diretório seja configurado no /etc/inetd.conf e no /usr/local/etc/dhcpd.conf.

1. Crie o diretório raiz que irá conter uma instalação do FreeBSD para ser montado por NFS:

```
# export NFSROOTDIR=/b/tftpboot/FreeBSD/install
# mkdir -p ${NFSROOTDIR}
```

2. Ative o servidor NFS adicionando esta linha ao /etc/rc.conf:

```
nfs_server_enable="YES"
```

3. Exporte o diretório raiz sem disco via NFS adicionando o seguinte ao /etc/exports:

```
/b -ro -alldirs -maproot=root
```

4. Inicie o servidor NFS:

```
# service nfsd start
```

5. Ative o inetd(8) adicionando a seguinte linha ao /etc/rc.conf:

```
inetd_enable="YES"
```

6. Descomente a seguinte linha no /etc/inetd.conf certificando-se de que ela não comece com um símbolo #:

```
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /b/tftpboot
```



Algumas versões do PXE exigem a versão TCP do TFTP. Neste caso, remova o comentário da segunda linha `tftp` que contém `stream tcp`.

7. Inicie o inetd(8):

```
# service inetd start
```

8. Instale o sistema básico em ${NFSROOTDIR}, seja descompactando os arquivos oficiais ou recompilando o kernel do FreeBSD e o userland (consulte Atualizando o FreeBSD a partir do código fonte para instruções mais detalhadas, mas não esqueça de adicionar `DESTDIR=${NFSROOTDIR}` ao executar os comandos `make installkernel` e `make installworld`.
9. Teste que o servidor TFTP funciona e que pode baixar o gerenciador de boot que será obtido via PXE:

```
# tftp localhost
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

10. Edite o ${NFSROOTDIR}/etc/fstab e crie uma entrada para montar o sistema de arquivos raiz por meio do NFS:

```
# Device                                    Mountpoint    FSType   Options
Dump Pass
myhost.example.com:/b/tftpboot/FreeBSD/install      /         nfs      ro
0    0
```

Substitua *myhost.example.com* pelo nome do host ou pelo endereço IP do servidor NFS. Neste exemplo, o sistema de arquivos raiz é montado como somente leitura para evitar que os clientes do NFS excluam potencialmente o conteúdo do sistema de arquivos raiz.

11. Defina a senha de root no ambiente PXE para as máquinas clientes que serão inicializadas

por PXE:

```
# chroot ${NFSROOTDIR}
# passwd
```

12. Se necessário, ative o login do root via ssh(1) para as máquinas clientes que estão inicializando por PXE editando o ${NFSROOTDIR}/etc/ssh/sshd_config e habilitando o `PermitRootLogin`. Esta opção está documentada em sshd_config(5).
13. Execute qualquer outra customização necessária do ambiente PXE no ${NFSROOTDIR}. Estas customizações podem incluir coisas como instalar pacotes ou editar o arquivo de senha com o vipw(8).

Ao inicializar de um volume raiz NFS, o /etc/rc detecta a inicialização do NFS e executa o /etc/rc.initdiskless. Neste caso, o /etc e /var precisam ser sistemas de arquivos montados em memória para que estes diretórios sejam graváveis mas o diretório raiz NFS seja apenas de leitura:

```
# chroot ${NFSROOTDIR}
# mkdir -p conf/base
# tar -c -v -f conf/base/etc.cpio.gz --format cpio --gzip etc
# tar -c -v -f conf/base/var.cpio.gz --format cpio --gzip var
```

Quando o sistema inicializar, os sistemas de arquivos em memória para o /etc e o /var serão criados e montados e o conteúdo dos arquivos cpio.gz será copiado para eles. Por padrão, esses sistemas de arquivos têm uma capacidade máxima de 5 megabytes. Se seus arquivos não couberem, o que geralmente é o caso do /var quando pacotes binários foram instalados, solicite um tamanho maior colocando o número de setores de 512 bytes necessários (por exemplo, 5 megabytes é 10240 setores) nos arquivos ${NFSROOTDIR}/conf/base/etc/md_size e ${NFSROOTDIR}/conf/base/var/md_size para os sistemas de arquivos /etc e o /var respectivamente.

31.8.2. Configurando o servidor DHCP

O servidor DHCP não precisa ser a mesma máquina que o servidor TFTP e NFS, mas ele precisa estar acessível na rede.

O DHCP não faz parte do sistema básico do FreeBSD, mas pode ser instalado usando o port ou pacote net/isc-dhcp44-server.

Uma vez instalado, edite o arquivo de configuração, /usr/local/etc/dhcpd.conf. Configure as diretivas `next-server`, `filename` e `root-path` conforme mostrado neste exemplo:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
   range 192.168.0.2 192.168.0.3 ;
   option subnet-mask 255.255.255.0 ;
   option routers 192.168.0.1 ;
   option broadcast-address 192.168.0.255 ;
   option domain-name-servers 192.168.35.35, 192.168.35.36 ;
```

```
    option domain-name "example.com";

    # IP address of TFTP server
    next-server 192.168.0.1 ;

    # path of boot loader obtained via tftp
    filename "FreeBSD/install/boot/pxeboot" ;

    # pxeboot boot loader will try to NFS mount this directory for root FS
    option root-path "192.168.0.1:/b/tftpboot/FreeBSD/install/" ;

}
```

A diretiva `next-server` é usada para especificar o endereço IP do servidor TFTP.

A diretiva `filename` define o caminho para o /boot/pxeboot. Um nome de arquivo relativo é usado, significando que /b/tftpboot não está incluído no caminho.

A diretiva `root-path` define o caminho para o sistema de arquivos raiz a ser montado por NFS.

Depois que as edições forem salvas, ative o DHCP no momento da inicialização adicionando a seguinte linha ao /etc/rc.conf:

```
dhcpd_enable="YES"
```

Então inicie o serviço DHCP:

```
# service isc-dhcpd start
```

31.8.3. Depurando problemas de PXE

Uma vez que todos os serviços estejam configurados e iniciados, os clientes de PXE devem poder carregar automaticamente o FreeBSD pela rede. Se um determinado cliente não conseguir se conectar, quando a máquina cliente inicializar, entre no menu de configuração da BIOS e confirme se ela está configurada para inicializar a partir da rede.

Esta seção descreve algumas dicas de solução de problemas para isolar a origem do problema de configuração, caso nenhum cliente seja capaz de inicializar o PXE.

1. Use o pacote ou port net/wireshark para depurar o tráfego de rede envolvido durante o processo de inicialização do PXE, que está ilustrado no diagrama abaixo.

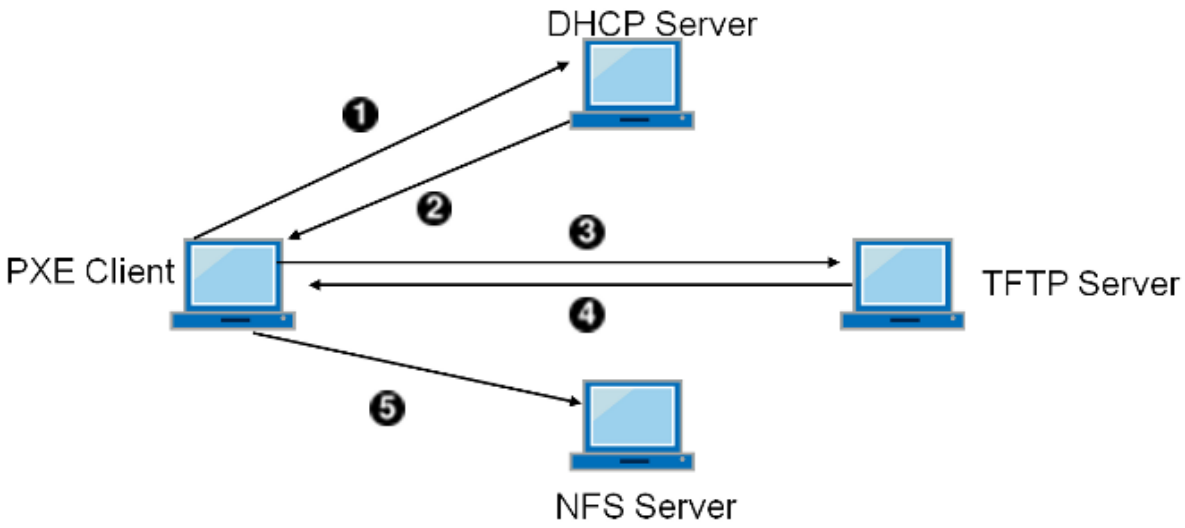


Figura 61. Processo de inicialização PXE com o sistema de arquivos raiz montado por NFS

2. No servidor TFTP, leia o /var/log/xferlog para garantir que o pxeboot esteja sendo recuperado do local correto. Para testar esta configuração de exemplo:

```
# tftp 192.168.0.1
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

   As seções de BUGS do tftpd(8) e tftp(1) documenta algumas limitações com o TFTP.

3. Certifique-se de que o sistema de arquivos raiz possa ser montado via NFS. Para testar esta configuração de exemplo:

```
# mount -t nfs 192.168.0.1:/b/tftpboot/FreeBSD/install /mnt
```

31.9. IPv6

O IPv6 é a nova versão do conhecido protocolo IP, também conhecido como IPv4. O IPv6 oferece várias vantagens sobre o IPv4, além de muitos recursos novos:

- Seu espaço de endereços de 128 bits permite 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços. Isso corrige a falta de endereços do IPv4 e o eventual esgotamento do endereço de IPv4.
- Os roteadores armazenam apenas endereços de agregação de rede em suas tabelas de roteamento, reduzindo assim o espaço médio de uma tabela de roteamento para 8192 entradas. Isso resolve os problemas de escalabilidade associados ao IPv4, que exigia que cada bloco alocado de endereços IPv4 fossem trocados entre roteadores da Internet, fazendo com que suas tabelas de roteamento ficassem muito grandes para permitir um roteamento eficiente .
- Autoconfiguração de endereço (RFC2462).
- Endereços multicast obrigatórios.

- IPsec Embutido (Segurança IP).
- Estrutura simplificada do cabeçalho.
- Suporte para mobile IP.
- Mecanismos de transição IPv6-to-IPv4.

O FreeBSD inclui a implementação de referência do http://www.kame.net/IPv6 e vem com tudo necessário usar o IPv6. Esta seção se concentra em configurar e executar o IPv6.

31.9.1. Informações sobre endereços de IPv6

Existem três tipos diferentes de endereços de IPv6:

Unicast

Um pacote enviado para um endereço unicast chega à interface pertencente ao endereço.

Anycast

Esses endereços são sintaticamente indistinguíveis dos endereços unicast, mas eles tratam de um grupo de interfaces. O pacote destinado a um endereço anycast chegará à interface do roteador mais próxima. Endereços anycast são usados apenas por roteadores.

Multicast

Esses endereços identificam um grupo de interfaces. Um pacote destinado a um endereço multicast chegará a todas as interfaces pertencentes ao grupo multicast. O endereço de broadcast IPv4 , geralmente `xxx.xxx.xxx.255`, é expresso por endereços multicast em IPv6.

Ao ler um endereço IPv6, a forma canônica é representada como `x:x:x:x:x:x:x:x`, onde cada `x` representa um valor hexadecimal de 16 bits. Um exemplo é `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`.

Muitas vezes, um endereço terá substrings longas apenas com zeros. Um `::` (dois-pontos duplos) pode ser usado para substituir uma subcadeia por endereço. Além disso, até três valores `0`s iniciais por valor hexadecimal podem ser omitidos. Por exemplo, `fe80::1` corresponde à forma canônica `fe80:0000:0000:0000:0000:0000:0000:0001`.

Uma terceira forma é escrever os últimos 32 bits usando a conhecida notação IPv4. Por exemplo, `2002::10.0.0.1` corresponde à representação canônica hexadecimal `2002:0000:0000:0000:0000:0000:0a00:0001`, que por sua vez é equivalente a `2002::a00:1`.

Para visualizar o endereço IPv6 do sistema FreeBSD, use ifconfig(8):

```
# ifconfig
```

```
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
        inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
        ether 00:00:21:03:08:e1
        media: Ethernet autoselect (100baseTX )
```

```
        status: active
```

Neste exemplo, a interface rl0 está usando `fe80::200:21ff:fe03:8e1%rl0`, um endereço local de link auto-configurado que foi gerado automaticamente a partir do endereço MAC.

Alguns endereços do IPv6 são reservados. Um resumo destes endereços reservados é visto em Endereços IPv6 reservados:

Tabela 30. Endereços IPv6 reservados

endereço IPv6	Prefixlength (Bits)	Descrição	Notas
`::`	128 bits	não especificado	Equivalente a `0.0.0.0` em IPv4.
`::1`	128 bits	endereço de loopback	Equivalente ao `127.0.0.1` no IPv4.
`::00:xx:xx:xx:xx`	96 bits	IPv4 Embarcado	Os 32 bits inferiores são o endereço IPv4 compatível.
`::ff:xx:xx:xx:xx`	96 bits	O endereço IPv4 mapeado do endereço IPv6	Os 32 bits mais baixos são o endereço IPv4 para hosts que não suportam o IPv6.
`fe80::/10`	10 bits	link-local	Equivalente a 169.254.0.0/16 em IPv4.
`fc00::/7`	7 bits	unique-local	Endereços locais exclusivos são destinados à comunicação local e só podem ser roteados dentro de um conjunto de sites cooperantes.
`ff00::`	8 bits	multicast	
`2000::-3fff::`	3 bits	unicast global	Todos os endereços unicast globais são atribuídos a partir desse pool. Os primeiros 3 bits são `001`.

Para maiores informações sobre a estrutura dos endereços do IPv6, consulte a RFC3513.

31.9.2. Configurando o IPv6

Para configurar um sistema FreeBSD como um cliente IPv6, adicione estas duas linhas ao rc.conf:

```
ifconfig_rl0_ipv6="inet6 accept_rtadv"
```

```
rtsold_enable="YES"
```

A primeira linha permite que a interface especificada receba mensagens de solicitação do roteador. A segunda linha ativa o daemon de solicitação do roteador, rtsol(8).

Se a interface precisar de um endereço IPv6 atribuído estaticamente, adicione uma entrada para especificar o endereço estático e o comprimento do prefixo associado:

```
ifconfig_rl0_ipv6="inet6 2001:db8:4672:6565:2026:5043:2d42:5344 prefixlen 64"
```

Para atribuir um roteador padrão, especifique seu endereço:

```
ipv6_defaultrouter="2001:db8:4672:6565::1"
```

31.9.3. Conectando-se a um provedor

Para se conectar a outras redes IPv6, é necessário ter um provedor ou um túnel que suporte IPv6:

- Entre em contato com um provedor de serviços de Internet para saber se eles oferecem IPv6.
- O Hurricane Electric oferece túneis com endpoints em todo o mundo.



Instale o pacote ou port net/freenet6 para uma conexão dial-up.

Esta seção demonstra como obter as direções de um provedor de túneis e convertê-las em configurações do /etc/rc.conf que persistirão durante as reinicializações.

A primeira entrada /etc/rc.conf cria a interface de encapsulamento genérica gif0:

```
cloned_interfaces="gif0"
```

Em seguida, configure essa interface com os endereços IPv4 dos pontos de extremidade locais e remotos. Substitua *MY_IPv4_ADDR* e *REMOTE_IPv4_ADDR* pelos endereços atuais de IPv4:

```
create_args_gif0="tunnel MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

Para aplicar o endereço IPv6 que foi atribuído para uso como o ponto final do túnel IPv6, adicione esta linha, substituindo *MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR* pelo endereço atribuído:

```
ifconfig_gif0_ipv6="inet6 MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Em seguida, defina a rota padrão para o outro lado do túnel IPv6. Substitua *MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR* pelo endereço do gateway padrão atribuído pelo provedor:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

Se o sistema FreeBSD irá rotear pacotes IPv6 entre o resto da rede e o mundo, habilite o gateway usando esta linha:

```
ipv6_gateway_enable="YES"
```

31.9.4. Anúncio do roteador e configuração automática do host

Esta seção demonstra como configurar o rtadvd(8) para anunciar a rota padrão de IPv6.

Para ativar rtadvd(8), inclua o seguinte no /etc/rc.conf:

```
rtadvd_enable="YES"
```

É importante especificar a interface na qual fazer a solicitação do roteador IPv6. Por exemplo, para informar o rtadvd(8) para usar rl0:

```
rtadvd_interfaces="rl0"
```

Em seguida, crie o arquivo de configuração, /etc/rtadvd.conf como visto neste exemplo:

```
rl0:\
    :addrs#1:addr="2001:db8:1f11:246::":prefixlen#64:tc=ether:
```

Substitua rl0 com a interface a ser usada e `2001:db8:1f11:246::` com o prefixo da alocação.

Para uma sub-rede `/64` dedicada, nada mais precisa ser alterado. Caso contrário, altere o `prefixlen#` para o valor correto.

31.9.5. IPv6 e o mapeamento de endereços IPv6

Quando o IPv6 está habilitado em um servidor, pode ser necessário ativar a comunicação de endereços IPv4 mapeados para IPv6. Esta opção de compatibilidade permite que endereços IPv4 sejam representados como endereços de IPv6. Permitir que aplicativos IPv6 se comuniquem com IPv4 e vice-versa pode ser um problema de segurança.

Essa opção pode não ser necessária na maioria dos casos e está disponível apenas para compatibilidade. Esta opção permitirá que os aplicativos que suportam apenas o IPv6 funcionem com IPv4 em um ambiente de pilha dupla. Isso é mais útil para aplicativos de terceiros que podem não suportar um ambiente somente de IPv6. Para habilitar esse recurso, adicione o seguinte ao /etc/rc.conf:

```
ipv6_ipv4mapping="YES"
```

Revisar as informações da RFC 3493, seção 3.6 e 3.7, bem como da RFC 4038 seção 4.2, pode ser útil para alguns administradores.

31.10. Protocolo Comum de Redundância de Endereços (CARP)

O Protocolo Comum de Redundância de Endereços (CARP) permite que vários hosts compartilhem o mesmo endereço IP e ID de Host Virtual (VHID) para fornecer *alta disponibilidade* para um ou mais serviços. Isso significa que um ou mais hosts podem falhar e os outros hosts assumem o controle de modo transparente, de modo que os usuários não percebam uma falha de serviço.

Além do endereço IP compartilhado, cada host tem seu próprio endereço IP para gerenciamento e configuração. Todas as máquinas que compartilham um endereço IP têm o mesmo VHID. O VHID para cada endereço virtual de IP deve ser exclusivo no domínio de broadcast da interface de rede.

A alta disponibilidade usando o CARP é nativa no FreeBSD, embora os passos para configurá-lo variem um pouco dependendo da versão do FreeBSD. Esta seção fornece a mesma configuração de exemplo para versões anteriores, iguais ou posteriores ao FreeBSD 10.

Este exemplo configura o suporte a failover com três hosts, todos com endereços exclusivos de IP, mas que fornecem o mesmo conteúdo da web. Ele tem dois mestres diferentes chamados `hosta.example.org` e `hostb.example.org`, com um backup compartilhado chamado `hostc.example.org`.

O balanceamento de carga destas máquinas é feito por meio de uma configuração de DNS Round Robin. As máquinas principais e de backup são configuradas de forma idêntica, exceto por seus nomes de host e endereços de gerenciamento IP. Esses servidores devem ter a mesma configuração e executar os mesmos serviços. Quando o failover ocorre, as solicitações para o serviço no endereço IP compartilhado só podem ser respondidas corretamente se o servidor de backup tiver acesso ao mesmo conteúdo. A máquina de backup tem duas interfaces CARP adicionais, uma para cada endereço IP do servidor de conteúdo mestre. Quando ocorre uma falha, o servidor de backup selecionará o endereço IP da máquina mestre com falha.

31.10.1. Usando CARP no FreeBSD 10 e Posteriores

Ative o suporte para CARP na inicialização do sistema, adicionando uma entrada para o módulo do kernel carp.ko em /boot/loader.conf:

```
carp_load="YES"
```

Para carregar o módulo agora sem reiniciar:

```
# kldload carp
```

Para usuários que preferem usar um kernel personalizado, inclua a seguinte linha no arquivo de configuração do kernel personalizado e compile o kernel como descrito em Configurando o kernel do FreeBSD:

```
device  carp
```

O nome do host, o endereço IP de gerenciamento e a máscara de sub-rede, o endereço IP compartilhado e o VHID são definidos adicionando entradas ao /etc/rc.conf. Este exemplo é para o `hosta.example.org`:

```
hostname="hosta.example.org"
ifconfig_em0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 1 pass testpass alias 192.168.1.50/32"
```

O próximo conjunto de entradas é para o `hostb.example.org`. Como ele representa um segundo mestre, ele usa um endereço IP compartilhado diferente e VHID. No entanto, as senhas especificadas com `pass` devem ser idênticas, pois o CARP somente ouvirá e aceitará anúncios de máquinas com a senha correta.

```
hostname="hostb.example.org"
ifconfig_em0="inet 192.168.1.4 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 2 pass testpass alias 192.168.1.51/32"
```

A terceira máquina, `hostc.example.org`, é configurada para lidar com o failover de um dos mestres. Esta máquina é configurada com dois CARPVHIDs, um para manipular o endereço IP virtual para cada um dos hosts principais. O desvio de publicidade CARP, `advskew`, é definida para garantir que o host de backup seja anunciado depois do mestre, pois `advskew` controla a ordem de precedência quando existem vários servidores de backup.

```
hostname="hostc.example.org"
ifconfig_em0="inet 192.168.1.5 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 1 advskew 100 pass testpass alias 192.168.1.50/32"
ifconfig_em0_alias1="inet vhid 2 advskew 100 pass testpass alias 192.168.1.51/32"
```

Ter dois CARPVHIDs configurados significa que o `hostc.example.org` notará se um dos servidores principais ficar indisponível. Se um mestre falhar em anunciar antes do servidor de backup, o servidor de backup selecionará o endereço IP compartilhado até que o mestre se torne disponível novamente.



Se o servidor mestre original se tornar disponível novamente, o `hostc.example.org` não liberará o endereço virtual IP de volta a ele automaticamente. Para que isso aconteça, a preempção deve ser ativada. O recurso está desabilitado por padrão, ele é controlado por meio da variável sysctl(8)`net.inet.carp.preempt`. O administrador pode forçar o servidor de backup a retornar o endereço IP para o mestre:

```
# ifconfig em0 vhid 1 state backup
```

Quando a configuração estiver concluída, reinicie a rede ou reinicie cada um dos sistemas. A alta disponibilidade está agora ativada.

A funcionalidade CARP pode ser controlada através de diversas variáveis sysctl(8) documentadas nas páginas de manual do carp(4). Outras ações podem ser acionadas a partir de eventos CARP usando devd(8).

31.10.2. Usando CARP no FreeBSD 9 e Anteriores

A configuração para estas versões do FreeBSD é similar àquela descrita na seção anterior, exceto que o dispositivo CARP deve ser criado primeiro e referenciado na configuração.

Ative o suporte de tempo de inicialização para o CARP carregando o módulo do kernel if_carp.ko no /boot/loader.conf:

```
if_carp_load="YES"
```

Para carregar o módulo agora sem reiniciar:

```
# kldload carp
```

Para usuários que preferem usar um kernel personalizado, inclua a seguinte linha no arquivo de configuração do kernel personalizado e compile o kernel como descrito em Configurando o kernel do FreeBSD:

```
device  carp
```

Em seguida, em cada host, crie um dispositivo CARP:

```
# ifconfig carp0 create
```

Defina o nome do host, o endereço IP de gerenciamento, o endereço IP compartilhado e o VHID adicionando as linhas necessárias ao /etc/rc.conf. Como um dispositivo virtual CARP é usado em vez de um alias, uma máscara de subrede real `/24` é usada em vez de uma `/32`. Aqui estão as entradas para o `hosta.example.org`:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpass 192.168.1.50/24"
```

Em `hostb.example.org`:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```

A terceira máquina, `hostc.example.org`, está configurada para lidar com o failover de qualquer um dos hosts principais:

```
hostname="hostc.example.org"
ifconfig_fxp0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass testpass 192.168.1.51/24"
```



A preempção está desabilitada no kernel GENERIC do FreeBSD. Se a preempção tiver sido ativada com um kernel personalizado, o `hostc.example.org` poderá não liberar o endereço IP de volta ao servidor de conteúdo original. O administrador pode forçar o servidor de backup a retornar o endereço IP para o mestre com o comando:

```
# ifconfig carp0 down && ifconfig carp0 up
```

Isso deve ser feito na interface carp, que corresponde ao host correto.

Quando a configuração estiver concluída, reinicie a rede ou reinicie cada um dos sistemas. A alta disponibilidade está agora ativada.

31.11. VLANs

As VLANs são uma forma de dividir virtualmente uma rede em várias sub-redes diferentes, também conhecida como segmentação. Cada segmento terá seu próprio domínio de broadcast e será isolado de outras VLANs.

No FreeBSD, as VLANs devem ser suportadas pelo driver da placa de rede. Para ver quais drivers suportam vlans, consulte a página de manual vlan(4).

Ao configurar uma VLAN, algumas informações devem ser conhecidas. Primeiro, qual a interface de rede? Segundo, qual é a tag da VLAN?

Para configurar uma VLANs em tempo de execução, com uma NIC `em0` e uma tag VLAN de `5` o comando ficaria assim:

```
# ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.20.20/24
```



Viu como o nome da interface inclui o nome do driver da NIC e a tag VLAN, separados por um ponto final? Essa é uma prática recomendada para facilitar a manutenção da configuração de VLAN quando muitas VLANs estiverem presentes em uma máquina.

Para configurar uma VLANs no momento da inicialização, o /etc/rc.conf deve ser atualizado. Para duplicar a configuração acima, será necessário adicionar o seguinte:

```
vlans_em0="5"
ifconfig_em0_5="inet 192.168.20.20/24"
```

VLANs adicionais podem ser inseridas, simplesmente adicionando a tag ao campo `vlans_em0` e incrementando uma linha de configuração da rede nessa interface da tag VLAN.

É útil atribuir um nome simbólico a uma interface para que, quando o hardware associado for alterado, apenas algumas variáveis de configuração precisem ser atualizadas. Por exemplo, câmeras de segurança precisam ser executadas pela VLAN 1 em `em0`. Posteriormente, se a placa `em0` for substituída por uma placa que use o driver ixgb(4), todas as referências a `em0.1` não precisarão ser alterado para `ixgb0.1`.

Para configurar a VLAN `5`, na NIC `em0`, atribua o nome de interface `cameras`, e atribua à interface um endereço IP de `192.168.20.20` com um prefixo `24`-bit, use este comando:

```
# ifconfig em0.5 create vlan 5 vlandev em0 name cameras inet 192.168.20.20/24
```

Para uma interface denominada `video`, use o seguinte:

```
# ifconfig video.5 create vlan 5 vlandev video name cameras inet 192.168.20.20/24
```

Para aplicar as mudanças no momento da inicialização, adicione as seguintes linhas ao /etc/rc.conf:

```
vlans_video="cameras"
create_args_cameras="vlan 5"
ifconfig_cameras="inet 192.168.20.20/24"
```

Parte V: Apêndices

Apêndice A: Obtendo o FreeBSD

A.1. CD and DVD Sets

Os conjuntos de CD and DVD do FreeBSD estão disponíveis em vários varejistas on-line:

- FreeBSD Mall, Inc.
  2420 Sand Creek Rd C-1 #347
  Brentwood, CA
  94513
  USA
  Phone: +1 925 240-6652
  Fax: +1 925 674-0821
  Email: <info@freebsdmall.com>
  WWW: https://www.freebsdmall.com
- Getlinux
  78 Rue de la Croix Rochopt
  Épinay-sous-Sénart
  91860
  France
  Email: <contact@getlinux.fr>
  WWW: http://www.getlinux.fr/
- Dr. Hinner EDV
  Kochelseestr. 11
  D-81371 München
  Germany
  Phone: (0177) 428 419 0
  Email: <infow@hinner.de>
  WWW: http://www.hinner.de/linux/freebsd.html
- Linux Center
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Russia
  Phone: +7-812-309-06-86
  Email: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/shop/freebsd

A.2. Sites de FTP

As fontes oficiais do FreeBSD estão disponíveis no FTP anônimo de um conjunto mundial de sites espelho. O site ftp://ftp.FreeBSD.org/pub/FreeBSD/ está disponível via HTTP e FTP. Ele é composto de muitas máquinas operadas pelos administradores de cluster do projeto e fica atrás de uma estrutura de GeoDNS que direciona os usuários para o espelho disponível mais próximo.

Adicionalmente, o FreeBSD está disponível via FTP anônimo a partir dos seguintes sites espelho. Ao obter o FreeBSD via FTP anônimo, por favor tente usar um site próximo. Os sites espelhos listados como "Sites Espelhos Primários" geralmente possuem o arquivo completo do FreeBSD (todas as versões atualmente disponíveis para cada uma das arquiteturas), mas velocidades de download mais rápidas provavelmente estão disponíveis em um site que esteja em seu país ou região. Os sites regionais carregam as versões mais recentes para a(s) arquitetura(s) mais populare(s), mas podem não carregar o arquivo completo do FreeBSD. Todos os sites fornecem acesso via FTP anônimo, mas alguns sites também fornecem acesso por meio de outros métodos. Os métodos de acesso disponíveis para cada site são fornecidos entre parênteses após o nome do host.

Central Servers, Primary Mirror Sites, Armenia, Australia, Austria, Brazil, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Ireland, Japan, Korea, Latvia, Lithuania, Netherlands, New Zealand, Norway, Poland, Russia, Saudi Arabia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Ukraine, United Kingdom, United States of America.

(as of UTC)

Central Servers

ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.FreeBSD.org/pub/FreeBSD/ / http://ftp.FreeBSD.org/pub/FreeBSD/)

Primary Mirror Sites

In case of problems, please contact the hostmaster <mirror-admin@FreeBSD.org> for this domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.FreeBSD.org/pub/FreeBSD/ / http://ftp4.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp10.FreeBSD.org/pub/FreeBSD/ / http://ftp10.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.FreeBSD.org/pub/FreeBSD/)

Armenia

In case of problems, please contact the hostmaster <hostmaster@am.FreeBSD.org> for this domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.am.FreeBSD.org/pub/FreeBSD/ / rsync)

Australia

In case of problems, please contact the hostmaster <hostmaster@au.FreeBSD.org> for this domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

Austria

In case of problems, please contact the hostmaster <hostmaster@at.FreeBSD.org> for this domain.

- ftp://ftp.at.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.at.FreeBSD.org/pub/FreeBSD/ / http://ftp.at.FreeBSD.org/pub/FreeBSD/)

Brazil

In case of problems, please contact the hostmaster <hostmaster@br.FreeBSD.org> for this domain.

- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http://ftp2.br.FreeBSD.org/)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)

Czech Republic

In case of problems, please contact the hostmaster <hostmaster@cz.FreeBSD.org> for this domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.cz.FreeBSD.org/pub/FreeBSD/)

Denmark

In case of problems, please contact the hostmaster <staff@dotsrc.org> for this domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.dk.FreeBSD.org/pub/FreeBSD/ / http://ftp.dk.FreeBSD.org/pub/FreeBSD/)

Estonia

In case of problems, please contact the hostmaster <hostmaster@ee.FreeBSD.org> for this domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

Finland

In case of problems, please contact the hostmaster <hostmaster@fi.FreeBSD.org> for this domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

France

In case of problems, please contact the hostmaster <hostmaster@fr.FreeBSD.org> for this domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.fr.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

Germany

In case of problems, please contact the hostmaster <de-bsd-hubs@de.FreeBSD.org> for this domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http://www1.de.FreeBSD.org/freebsd/ / rsync://rsync3.de.FreeBSD.org/freebsd/)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.de.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http://ftp4.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp7.de.FreeBSD.org/pub/FreeBSD/)

Greece

In case of problems, please contact the hostmaster <hostmaster@gr.FreeBSD.org> for this domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

Hong Kong

ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

Ireland

In case of problems, please contact the hostmaster <hostmaster@ie.FreeBSD.org> for this domain.

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Japan

In case of problems, please contact the hostmaster <hostmaster@jp.FreeBSD.org> for this domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

Korea

In case of problems, please contact the hostmaster <hostmaster@kr.FreeBSD.org> for this domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.kr.FreeBSD.org/pub/FreeBSD/)

Latvia

In case of problems, please contact the hostmaster <hostmaster@lv.FreeBSD.org> for this domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lv.FreeBSD.org/pub/FreeBSD/)

Lithuania

In case of problems, please contact the hostmaster <hostmaster@lt.FreeBSD.org> for this domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lt.FreeBSD.org/pub/FreeBSD/)

Netherlands

In case of problems, please contact the hostmaster <hostmaster@nl.FreeBSD.org> for this domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nl.FreeBSD.org/os/FreeBSD/ / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

New Zealand

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nz.FreeBSD.org/pub/FreeBSD/)

Norway

In case of problems, please contact the hostmaster <hostmaster@no.FreeBSD.org> for this domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Poland

In case of problems, please contact the hostmaster <hostmaster@pl.FreeBSD.org> for this domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)

Russia

In case of problems, please contact the hostmaster <hostmaster@ru.FreeBSD.org> for this domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ru.FreeBSD.org/FreeBSD/ / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp5.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

Saudi Arabia

In case of problems, please contact the hostmaster <ftpadmin@isu.net.sa> for this domain.

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org (ftp)

Slovenia

In case of problems, please contact the hostmaster <hostmaster@si.FreeBSD.org> for this domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

South Africa

In case of problems, please contact the hostmaster <hostmaster@za.FreeBSD.org> for this domain.

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

Spain

In case of problems, please contact the hostmaster <hostmaster@es.FreeBSD.org> for this domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.es.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

Sweden

In case of problems, please contact the hostmaster <hostmaster@se.FreeBSD.org> for this domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.se.FreeBSD.org/)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.se.FreeBSD.org/pub/FreeBSD/)

Switzerland

In case of problems, please contact the hostmaster <hostmaster@ch.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ch.FreeBSD.org/pub/FreeBSD/)

Taiwan

In case of problems, please contact the hostmaster <hostmaster@tw.FreeBSD.org> for this domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.tw.FreeBSD.org/ / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp11.tw.FreeBSD.org/FreeBSD/)
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

Ukraine

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ua.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.ua.FreeBSD.org/pub/FreeBSD / rsync://ftp6.ua.FreeBSD.org/FreeBSD/)
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

United Kingdom

In case of problems, please contact the hostmaster <hostmaster@uk.FreeBSD.org> for this domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.uk.FreeBSD.org/ftp.freebsd.org/pub/FreeBSD/)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

United States of America

In case of problems, please contact the hostmaster <hostmaster@us.FreeBSD.org> for this domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.us.FreeBSD.org/pub/FreeBSD/ / http://ftp4.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp13.us.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3. Usando o Subversion

A.3.1. Introdução

Desde de julho de 2012, o FreeBSD usa o Subversion como o único sistema de controle de versão para armazenar todo o código-fonte do FreeBSD, a documentação e a coleção de ports.



O Subversion é geralmente uma ferramenta de desenvolvimento. Os usuários podem preferir usar o `freebsd-update` (Atualização do FreeBSD) para atualizar o sistema básico do FreeBSD, e o `portsnap` (Usando a Coleção de Ports) para atualizar a coleção de ports do FreeBSD.

Esta seção demonstra como instalar o Subversion em um sistema FreeBSD e usá-lo para criar uma cópia local de um repositório do FreeBSD. Informações adicionais sobre o uso de Subversion estão incluídas.

A.3.2. Certificados Raiz SSL

A instalação do security/ca_root_nss permite que o Subversion verifique a identidade dos servidores de repositório HTTPS. Os certificados raiz SSL podem ser instalados a partir de um port:

```
# cd /usr/ports/security/ca_root_nss
# make install clean
```

ou como um pacote:

```
# pkg install ca_root_nss
```

A.3.3. Svnlite

Uma versão leve do Subversion já está instalada no FreeBSD como `svnlite`. A versão do port ou pacote do Subversion é necessária apenas se a API do Python ou do Perl for necessária, ou se uma versão posterior do Subversion for desejada.

A única diferença do uso normal do Subversion é que o nome do comando é `svnlite`.

A.3.4. Instalação

Se o `svnlite` não estiver disponível ou a versão completa do Subversion for necessária, ele deverá ser instalado.

O Subversion pode ser instalado a partir da coleção de ports:

```
# cd /usr/ports/devel/subversion
# make install clean
```

O Subversion também pode ser instalado como um pacote:

```
# pkg install subversion
```

A.3.5. Executando o Subversion

Para obter uma cópia limpa do código-fonte em um diretório local, use `svn`. Os arquivos neste diretório são chamados de *cópia de trabalho local*.



Mova ou exclua o diretório de destino existente antes de usar o `checkout` pela primeira vez.

O checkout em cima de um diretório não-`svn` existente pode causar conflitos entre os arquivos existentes e aqueles trazidos do repositório.

O Subversion usa URLs para designar um repositório, sob a forma de *protocol://hostname/path*. O primeiro componente do caminho é o repositório do FreeBSD para acessar. Existem três repositórios diferentes, `base` para o código-fonte do sistema básico do FreeBSD, `ports` para a coleção de ports, e `doc` para a documentação. Por exemplo, o URL `https://svn.FreeBSD.org/ports/head/` especifica a ramificação principal do repositório de ports, usando o protocolo `https`.

Um checkout de um determinado repositório é executado com um comando como este:

```
# svn checkout https://svn.FreeBSD.org/repository/branch lwcdir
```

Onde:

- O *repository* é um dos repositórios do Projecto: `base`, `ports`, ou `doc`.
- A *branch* depende do repositório usado. O `ports` e o `doc` são normalmente atualizados na ramificação `head`, enquanto `base` mantém a última versão de -CURRENT em `head` e as respectivas versões mais recentes das ramificações -STABLE em `stable/9` (9.*x*) e `stable/10` (10.*x*).
- O *lwcdir* é o diretório de destino onde o conteúdo do ramo especificado deve ser colocado. Isso geralmente é /usr/ports para o `ports`, /usr/src para a `base`, e /usr/doc para o `doc`.

Este exemplo obtém a coleção de ports do repositório do FreeBSD usando o protocolo HTTPS, colocando a cópia de trabalho local em /usr/ports. Se o /usr/ports já estiver presente, mas não tiver sido criado pelo `svn`, lembre-se de renomeá-lo ou excluí-lo antes do checkout.

```
# svn checkout https://svn.FreeBSD.org/ports/head /usr/ports
```

Como o checkout inicial deve fazer o download da ramificação completa do repositório remoto, isso pode demorar um pouco. Por favor, seja paciente.

Após o checkout inicial, a cópia de trabalho local pode ser atualizada executando:

```
# svn update lwcdir
```

Para atualizar o /usr/ports criado no exemplo acima, use:

```
# svn update /usr/ports
```

O update é muito mais rápido do que um checkout, transferindo apenas os arquivos que foram alterados.

Uma maneira alternativa de atualizar a cópia de trabalho local após o checkout é fornecida pelo Makefile existente em /usr/ports, /usr/src, e /usr/doc. Configure o `SVN_UPDATE` e use o destino `atualizar`. Por exemplo, para atualizar /usr/src:

```
# cd /usr/src
# make update SVN_UPDATE=yes
```

A.3.6. Sites Espelho do Subversion

O repositório Subversion do FreeBSD é:

```
svn.FreeBSD.org
```

Essa é uma rede de espelhos acessível publicamente a qual usa o GeoDNS para selecionar um

servidor de backend apropriado. Para visualizar os repositórios Subversion do FreeBSD através de um navegador, use https://svnweb.FreeBSD.org/.

O HTTPS é o protocolo preferido, mas o pacote security/ca_root_nss precisará ser instalado para validar os certificados automaticamente.

A.3.7. Para Maiores Informações

Para outras informações sobre o uso do Subversion, por favor veja o "Subversion Book", intitulado Version Controle com Subversion, ou o Documentação do Subversion.

A.4. Usando o rsync

Estes sites disponibilizam o FreeBSD através do protocolo rsync. O utilitário rsync transfere apenas as diferenças entre dois conjuntos de arquivos. Isto é útil para sites espelho do servidor de FTP do FreeBSD . O pacote rsync está disponível para muitos sistemas operacionais, no FreeBSD, veja o port net/rsync ou use o pacote.

República Checa

rsync://ftp.cz.FreeBSD.org/

Coleções disponíveis:

- ftp: Um espelho parcial do servidor de FTP do FreeBSD.
- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.

Países Baixos

rsync://ftp.nl.FreeBSD.org/

Coleções disponíveis:

- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.

Rússia

rsync://ftp.mtu.ru/

Coleções disponíveis:

- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.
- FreeBSD-Archive: Um espelho do servidor de FTP do FreeBSD Archive.

Suécia

rsync://ftp4.se.freebsd.org/

Coleções disponíveis:

- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.

Taiwan

rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

Coleções disponíveis:

- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.

Reino Unido

rsync://rsync.mirrorservice.org/

Coleções disponíveis:

- ftp.freebsd.org: Um espelho completo do servidor de FTP do FreeBSD.

Estados Unidos da America

rsync://ftp-master.FreeBSD.org/

Este servidor só pode ser usado por sites espelhos primários do FreeBSD.

Coleções disponíveis:

- FreeBSD: O arquivo master do servidor de FTP do FreeBSD.
- acl: A lista do ACL mestre do FreeBSD.

  rsync://ftp13.FreeBSD.org/

  Coleções disponíveis:

- FreeBSD: Um espelho completo do servidor de FTP do FreeBSD.

Apêndice B: Bibliografia

Enquanto páginas manuais fornecem uma referência definitiva para partes individuais do sistema operacional FreeBSD, elas raramente ilustram como juntar as peças para fazer todo o sistema operacional rodar sem problemas. Para isso, não há substituto para um bom livro ou manual do usuário na administração do sistema UNIX™.

B.1. Livros específicos para o FreeBSD

Livros internacionais:

- Using FreeBSD (in Traditional Chinese), published by Drmaster, 1997. ISBN 9-578-39435-7.
- FreeBSD Unleashed (Simplified Chinese translation), published by China Machine Press. ISBN 7-111-10201-0.
- FreeBSD From Scratch Second Edition (in Simplified Chinese), published by China Machine Press. ISBN 7-111-10286-X.
- FreeBSD Handbook Second Edition (Simplified Chinese translation), published by Posts & Telecom Press. ISBN 7-115-10541-3.
- FreeBSD & Windows (in Simplified Chinese), published by China Railway Publishing House. ISBN 7-113-03845-X
- FreeBSD Internet Services HOWTO (in Simplified Chinese), published by China Railway Publishing House. ISBN 7-113-03423-3
- FreeBSD (in Japanese), published by CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (in Japanese), published by Shoeisha Co., Ltd. ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (in Japanese), published by ASCII. ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (Japanese translation), published by ASCII. ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (in German), published by Computer und Literatur Verlag/Vertrieb Hanser, 1998. ISBN 3-932311-31-0.
- FreeBSD de Luxe (in German), published by Verlag Modere Industrie, 2003. ISBN 3-8266-1343-0.
- FreeBSD Install and Utilization Manual (in Japanese), published by Mainichi Communications Inc., 1998. ISBN 4-8399-0112-0.
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Building Internet Server with FreeBSD* (in Indonesia Language), published by Elex Media Komputindo.
- Absolute BSD: The Ultimate Guide to FreeBSD (Traditional Chinese translation), published by GrandTech Press, 2003. ISBN 986-7944-92-5.
- The FreeBSD 6.0 Book (in Traditional Chinese), published by Drmaster, 2006. ISBN 9-575-27878-X.

Livros de língua inglesa:

- Absolute FreeBSD, 2nd Edition: The Complete Guide to FreeBSD, published by No Starch Press, 2007. ISBN: 978-1-59327-151-0
- The Complete FreeBSD, published by O'Reilly, 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide, published by Addison-Wesley, 2000. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer, published by The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, published by Sams, 2002. ISBN: 0672324245
- FreeBSD 6 Unleashed, published by Sams, 2006. ISBN: 0672328755
- FreeBSD: The Complete Reference, published by McGrawHill, 2003. ISBN: 0072224096

B.2. Guias de usuários

- Ohio State University has written a UNIX Introductory Course which is available online in HTML and PostScript format.

  An Italian translation of this document is available as part of the FreeBSD Italian Documentation Project.

- Jpman Project, Japan FreeBSD Users Group. FreeBSD User's Reference Manual (Japanese translation). Mainichi Communications Inc., 1998. ISBN4-8399-0088-4 P3800E.
- Edinburgh University has written an Online Guide for newcomers to the UNIX environment.

B.3. Guias de Administradores

- Jpman Project, Japan FreeBSD Users Group. FreeBSD System Administrator's Manual (Japanese translation). Mainichi Communications Inc., 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD 2nd Ed. (in French), Eyrolles, 2004. ISBN 2-212-11463-X

B.4. Guias de programadores

- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th Ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7
- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9

- Spinellis, Diomidis. Code Reading: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5
- Spinellis, Diomidis. Code Quality: The Open Source Perspective. Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard and Stephen A. Rago. *Advanced Programming in the UNIX Environment*. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X

B.5. Internals do sistema operacional

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobb's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4

  (Chapter 2 of this book is available online as part of the FreeBSD Documentation Project.)
- Marshall Kirk McKusick, George V. Neville-Neil *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Marshall Kirk McKusick, George V. Neville-Neil, Robert N. M. Watson *The Design and Implementation of the FreeBSD Operating System, 2nd Ed.*. Westford, Mass. : Pearson Education, Inc., 2014. ISBN 0-321-96897-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals — The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

B.6. Referências de segurança

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily*

Hacker. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4

- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7. Referências de Hardware

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Intel Corporation publishes documentation on their CPUs, chipsets and standards on their developer web site, usually as PDF files.
- Shanley, Tom. *80486 System Architecture*. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th Ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass : Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.8. História do UNIX™

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Also known as the Jargon File
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Out of print, but available online.
- Don Libes, Sandy Ressler *Life with UNIX* — special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. https://svnweb.freebsd.org/base/head/shared/misc/bsd-family-tree?view=co or /usr/shared/misc/bsd-family-tree on a FreeBSD machine.
- *Networked Computer Science Technical Reports Library*.
- *Old BSD releases from the Computer Systems Research group (CSRG)*. http://www.mckusick.com/csrg/: The 4CD set covers all BSD versions from 1BSD to 4.4BSD and 4.4BSD-Lite2 (but not 2.11BSD, unfortunately). The last disk also holds the final sources plus the SCCS files.

- Kernighan, Brian *Unix: A History and a Memoir*. Kindle Direct Publishing, 2020. ISBN 978-169597855-3

B.9. Periódicos, Jornais e Revistas

- Admin Magazin (in German), published by Medialinx AG. ISSN: 2190-1066
- BSD Magazine, published by Software Press Sp. z o.o. SK. ISSN: 1898-9144
- BSD Now — Video Podcast, published by Jupiter Broadcasting LLC
- BSD Talk Podcast, by Will Backman
- FreeBSD Journal, published by S&W Publishing, sponsored by The FreeBSD Foundation. ISBN: 978-0-615-88479-0

Apêndice C: Recursos na Internet

O ritmo acelerado do progresso do FreeBSD torna a mídia impressa impraticável como um meio de acompanhar os desenvolvimentos mais recentes. Os recursos eletrônicos são a melhor maneira, se não a única, de se manter informado sobre os últimos avanços. Como o FreeBSD é um esforço voluntário, a própria comunidade de usuários geralmente serve como um "departamento de suporte técnico", com o correio eletrônico, fóruns na web e notícias da USENET sendo a maneira mais eficaz de alcançar essa comunidade.

Os pontos mais importantes de contato com a comunidade de usuários do FreeBSD são descritos abaixo. Por favor, envie outros recursos não mencionados aqui para a lista de discussão do projeto de documentação do FreeBSD para que eles também possam ser incluídos.

C.1. Websites

- The FreeBSD Forums fornecem um fórum de discussão baseado na web para questões sobre o FreeBSD e para discussão técnica.
- O Canal do YouTube BSDConferences oferece uma coleção de vídeos de alta qualidade de conferências sobre o BSD em todo o mundo. Esta é uma ótima maneira de assistir desenvolvedores-chave fazerem apresentações sobre novos trabalhos no FreeBSD.

C.2. Listas de Discussão

As listas de discussão são a maneira mais direta de abordar questões ou abrir uma discussão técnica para um público concentrado do FreeBSD. Há uma grande variedade de listas em vários tópicos diferentes do FreeBSD. Enviar perguntas para a lista de discussão mais adequada invariavelmente garantirá uma resposta mais rápida e precisa.

Os charters das várias listas são dadas na parte inferior deste documento. *Por favor, leia o regulamento antes de se cadastrar ou enviar e-mails para qualquer lista.* A maioria dos assinantes de listas recebe muitas centenas de mensagens relacionadas ao FreeBSD todos os dias, e os charters e regras de uso visam manter a relação sinal-ruído das listas altas. Para fazer menos, as listas de discussão acabarão por falhar como um meio de comunicação eficaz para o Projeto.



Para testar a capacidade de enviar email para as listas do FreeBSD, envie uma mensagem de teste para freebsd-test. Por favor, não envie mensagens de teste para qualquer outra lista.

Em caso de dúvida sobre a lista para colocar uma pergunta, consulte Como obter os melhores resultados da lista de discussão FreeBSD-questions.

Antes de postar em qualquer lista, aprenda sobre a melhor forma de usar as listas de discussão, por exemplo, como ajudar a evitar discussões repetidas com frequência, lendo o documento de Perguntas Frequentes das Mailing Lists (FAQ).

Os arquivos são mantidos para todas as listas de discussão e podem ser pesquisados usando o servidor da World Wide Web do FreeBSD. A busca por palavras-chaves no arquivo oferece uma

excelente maneira de encontrar respostas para perguntas freqüentes e deve ser consultada antes de postar uma pergunta. Note que isso também significa que as mensagens enviadas para as listas de discussão do FreeBSD são arquivadas perpetuamente. Se a proteção da sua privacidade é uma preocupação, considere usar um endereço de e-mail secundário descartável e postar apenas informações públicas.

C.2.1. Sumário

Listas gerais: A seguir, listas gerais das quais qualquer pessoa é livre (e incentivada) a participar:

Lista	Propósito
freebsd-advocacy	Evangelismo do FreeBSD
freebsd-announce	Eventos importantes e marcos do projeto (moderado)
freebsd-arch	Discussões de arquitetura e design
freebsd-bugbusters	Discussões relativas à manutenção do banco de dados de relatórios de problemas do FreeBSD e ferramentas relacionadas
freebsd-bugs	Relatório de erros
freebsd-chat	Itens não técnicos relacionados à comunidade FreeBSD
freebsd-chromium	Problemas do Chromium específicos do FreeBSD
freebsd-current	Discussão sobre o uso do FreeBSD-CURRENT
freebsd-isp	Problemas para provedores de serviços de Internet usando o FreeBSD
freebsd-jobs	Vagas de empregos para trabalhar com FreeBSD e oportunidades de consultoria
freebsd-quarterly-calls	Chamadas para relatórios de status trimestrais (moderado)
freebsd-questions	Perguntas de usuários e suporte técnico
freebsd-security-notifications	Notificações de segurança (moderadas)
freebsd-stable	Discussão sobre o uso do FreeBSD-STABLE
freebsd-test	Lista para qual enviar mensagens de teste em vez de para uma das listas reais
freebsd-women	Defesa do FreeBSD para mulheres

Listas técnicas: As listas a seguir são para discussão técnica. Leia atentamente o regulamento de cada lista antes de aderir ou enviar e-mails para uma, pois há diretrizes firmes para seu uso e conteúdo.

Lista	Propósito
freebsd-acpi	ACPI e desenvolvimento de gerenciamento de energia
freebsd-amd64	Portando FreeBSD para sistemas AMD64 (moderado)
freebsd-apache	Discussão sobre ports relacionados ao Apache
freebsd-arm	Portando o FreeBS para processadores ARM ™
freebsd-atm	Usando a rede ATM com o FreeBSD
freebsd-bluetooth	Usando a tecnologia Bluetooth ™ no FreeBSD
freebsd-cloud	FreeBSD em plataformas em nuvem (EC2, GCE, Azure, etc.)
freebsd-cluster	Usando o FreeBSD em um ambiente clusterizado
freebsd-database	Discutindo o uso e desenvolvimento do banco de dados no FreeBSD
freebsd-desktop	Usando e melhorando o FreeBSD na área de trabalho
dev-ci	Construa e teste relatórios dos servidores de integração contínua
dev-reviews	Notificações do sistema de revisão do FreeBSD
freebsd-doc	Criando documentos relacionados ao FreeBSD
freebsd-drivers	Escrevendo drivers de dispositivos para o FreeBSD
freebsd-dtrace	Usando e trabalhando no DTrace no FreeBSD
freebsd-eclipse	Usuários FreeBSD do Eclipse IDE, ferramentas, aplicativos rich client e ports.
freebsd-elastic	Discussões específicas sobre ElasticSearch no FreeBSD
freebsd-embedded	Usando o FreeBSD em aplicativos embarcados
freebsd-eol	Peer suporte a softwares relacionados ao FreeBSD que não são mais suportado pelo Projeto FreeBSD.
freebsd-emulation	Emulação de outros sistemas como o Linux/MS-DOS™/Windows™
freebsd-enlightenment	Portando o Enlightenment e aplicativos Enlightenment
freebsd-erlang	Discussões específicas sobre Erlang no FreeBSD
freebsd-firewire	Discussão técnica do FreeBSD FireWire™ (iLink, IEEE 1394)

Lista	Propósito
freebsd-fortran	Fortran no FreeBSD
freebsd-fs	Sistemas de arquivos
freebsd-games	Suporte para jogos no FreeBSD
freebsd-gecko	Problemas do Gecko Rendering Engine
freebsd-geom	Discussões e implementações específicas do GEOM
freebsd-git	Discussão sobre o uso do git no projeto FreeBSD
freebsd-gnome	Portando aplicativos GNOME e o GNOME
freebsd-hackers	Discussão técnica geral
freebsd-haskell	Questões e discussões sobre o Haskell específicas do FreeBSD
freebsd-hardware	Discussão geral de hardware para executar o FreeBSD
freebsd-i18n	Internacionalização do FreeBSD
freebsd-infiniband	Infiniband no FreeBSD
freebsd-ipfw	Discussão técnica sobre o redesenho do código de firewall de IP
freebsd-isdn	Desenvolvedores ISDN
freebsd-jail	Discussões sobre jail(8)
freebsd-java	Desenvolvedores Java™ e pessoas trabalhando no port dos JDK™s para o FreeBSD
freebsd-kde	Portando aplicativos KDE e o KDE
freebsd-lfs	Portando o LFS para o FreeBSD
freebsd-mips	Portando o FreeBS para MIPS ™
freebsd-mono	Aplicativos Mono e C # no FreeBSD
freebsd-multimedia	Aplicações multimídia
freebsd-new-bus	Discussões técnicas sobre arquitetura de barramento
freebsd-net	Discussão de rede e código-fonte TCP/IP
freebsd-numerics	Discussões sobre a implementação de alta qualidade de funções libm
freebsd-ocaml	Discussões específicas sobre OCaml no FreeBSD
freebsd-office	Aplicativos do Office no FreeBSD
freebsd-performance	Perguntas de ajuste de desempenho para instalações de alto desempenho/carga

Lista	Propósito
freebsd-perl	Manutenção de vária\os ports relacionados ao Perl
freebsd-pf	Discussão e perguntas sobre o sistema de firewall de filtro de pacotes
freebsd-pkg	Gerenciamento de pacotes binários e discussão de ferramentas de pacote
freebsd-pkg-fallout	Registros de fallout da construção de pacotes
freebsd-pkgbase	Empacotando o sistema básico do FreeBSD
freebsd-platforms	No que diz respeito ao port para plataformas de arquitetura não Intel™
freebsd-ports	Discussão da Coleção de Ports
freebsd-ports-announce	Notícias e instruções importantes sobre a coleção de ports (moderada)
freebsd-ports-bugs	Discussão dos bugs/PRs dos ports
freebsd-ppc	Portando o FreeBSD para o PowerPC ™
freebsd-proliant	Discussão técnica do FreeBSD em plataformas de servidores HP ProLiant
freebsd-python	Problemas específicos do Python para FreeBSD
freebsd-rc	Discussão relacionada ao sistema rc.d e seu desenvolvimento
freebsd-realtime	Desenvolvimento de extensões em tempo real para o FreeBSD
freebsd-riscv	Portando o FreeBSD para sistemas RISC-V™
freebsd-ruby	Discussões específicas sobre o Ruby no FreeBSD
freebsd-scsi	O subsistema SCSI
freebsd-security	Problemas de segurança que afetam o FreeBSD
freebsd-snapshots	Anúncios de Snapshots dos ramos de Desenvolvimento do FreeBSD
freebsd-sparc64	Portando o FreeBSD para sistemas baseados em SPARC ™
freebsd-standards	Conformidade do FreeBSD com os padrões C99 e POSIX ™
freebsd-sysinstall	Desenvolvimento do sysinstall(8)
freebsd-tcltk	Discussões Tcl / Tk específicas do FreeBSD
freebsd-testing	Testando no FreeBSD
freebsd-tex	Portando o TeX e seus aplicativos para o FreeBSD

Lista	Propósito
freebsd-threads	Threads no FreeBSD
freebsd-tilera	Portando o FreeBSD para a família Tilera de CPUs
freebsd-tokenring	Suporte Token Ring no FreeBSD
freebsd-toolchain	Manutenção do toolchain integrado do FreeBSD
freebsd-translators	Traduzindo documentos e programas do FreeBSD
freebsd-transport	Discussões de protocolos de rede em nível de transporte no FreeBSD
freebsd-usb	Discutindo o suporte do FreeBSD para USB
freebsd-virtualization	Discussão de várias técnicas de virtualização suportadas pelo FreeBSD
freebsd-vuxml	Discussão sobre a infraestrutura VuXML
freebsd-x11	Manutenção e suporte do X11 no FreeBSD
freebsd-xen	Discussão do port do FreeBSD para o Xen™ - implementação e uso
freebsd-xfce	XFCE para o FreeBSD - portando e mantendo
freebsd-zope	Zope para o FreeBSD - portando e mantendo

Listas limitadas: As listas a seguir são para públicos mais especializados (e exigentes) e provavelmente não são de interesse para o público em geral. Também é uma boa ideia estabelecer uma presença nas listas técnicas antes de entrar em uma dessas listas limitadas para entender a etiqueta de comunicação envolvida.

Lista	Propósito
freebsd-hubs	Pessoas executando sites espelho (suporte infraestrutural)
freebsd-user-groups	Coordenação de grupo de usuários
freebsd-wip-status	FreeBSD Work-In-Progress Status
freebsd-wireless	Discussões da pilha 802.11, ferramentas, desenvolvimento de drivers de dispositivos

Listas de resumo: Todas as listas acima estão disponíveis em formato resumido. Uma vez inscrito em uma lista, as opções de resumo podem ser alteradas na seção de opções da conta.

Listas de SVN: As listas a seguir são para pessoas interessadas em ver as mensagens de log para alterações em várias áreas da árvore de código-fonte. Elas são listas de *somente leitura* e não devem ter correio enviado para elas.

Lista	Área do Fonte	Descrição da área (fonte para)
svn-doc-all	/usr/doc	Todas as alterações no repositório Subversion do doc (exceto para user, projects e translations)
svn-doc-head	/usr/doc	Todas as alterações na ramificação "head" do repositório do Subversion do doc
svn-doc-projects	/usr/doc/projects	Todas as alterações na área projects do repositório Subversion do doc
svn-doc-svnadmin	/usr/doc	Todas as mudanças nos scripts administrativos, hooks e outros dados de configuração do repositório do Subversion do doc
svn-ports-all	/usr/ports	Todas as mudanças no repositório do Subversion do ports
svn-ports-head	/usr/ports	Todas as mudanças na ramificação " head " do repositório do Subversion do ports
svn-ports-svnadmin	/usr/ports	Todas as mudanças nos scripts administrativos, hooks e outros dados de configuração do repositório Subversion das portas
svn-src-all	/usr/src	Todas as mudanças no repositório src Subversion (exceto para user e projects)
svn-src-head	/usr/src	Todas as mudanças na ramificação " head " do repositório src Subversion (a ramificação FreeBSD-CURRENT)
svn-src-projects	/usr/projects	Todas as mudanças na área projects do repositório src do Subversion
svn-src-release	/usr/src	Todas as mudanças na área releases do repositório src do Subversion

Lista	Área do Fonte	Descrição da área (fonte para)
svn-src-releng	/usr/src	Todas as mudanças nas ramificações releng do repositório src Subversion (as ramificações de engenharia de segurança/release)
svn-src-stable	/usr/src	Todas as mudanças para todos os ramos estáveis do repositório src Subversion
svn-src-stable-6	/usr/src	Todas as alterações na ramificação stable/6 do repositório src Subversion
svn-src-stable-7	/usr/src	Todas as alterações na ramificação stable/7 do repositório src Subversion
svn-src-stable-8	/usr/src	Todas as mudanças na ramificação stable/8 do repositório src Subversion
svn-src-stable-9	/usr/src	Todas as alterações na ramificação stable/9 do repositório src Subversion
svn-src-stable-10	/usr/src	Todas as mudanças na ramificação stable/10 do repositório src do Subversion
svn-src-stable-11	/usr/src	Todas as alterações na ramificação stable/11 do repositório src Subversion
svn-src-stable-12	/usr/src	Todas as mudanças na ramificação stable/12 do repositório src do Subversion
svn-src-stable-other	/usr/src	Todas as mudanças para os ramos mais antigos stable do repositório src Subversion
svn-src-svnadmin	/usr/src	Todas as mudanças nos scripts administrativos, hooks e outros dados de configuração do repositório src do Subversion
svn-src-user	/usr/src	Todas as mudanças na área experimental user do repositório src do Subversion

Lista	Área do Fonte	Descrição da área (fonte para)
svn-src-vendor	/usr/src	Todas as mudanças na área de trabalho do fornecedor do repositório src Subversion

C.2.2. Como se inscrever

Para se inscrever em uma lista, clique no nome da lista em http://lists.FreeBSD.org/mailman/listinfo . A página exibida deve conter todas as instruções de inscrição necessárias para essa lista.

Para realmente postar em uma determinada lista, envie um email para listname@FreeBSD.org. Ele será então redistribuído para membros da lista de discussão em todo o mundo.

Para cancelar a inscrição em uma lista, clique no URL encontrado na parte inferior de todos os e-mails recebidos da lista. Também é possível enviar um email para listname-unsubscribe@FreeBSD.org para cancelar a inscrição.

É importante manter a discussão nas listas de discussão técnicas em uma trilha técnica. Para receber apenas os anúncios importantes, junte-se à lista de discussão de anúncios do FreeBSD, que é destinada a tráfego pouco frequente .

C.2.3. Estatutos das Listas

Todas as listas de discussão do FreeBSD possuem certas regras básicas que devem ser seguidas por qualquer pessoa que as utilize. O não cumprimento destas diretrizes resultará em dois (2) avisos escritos do Postmaster do FreeBSD postmaster@FreeBSD.org, após o que, em uma terceira ofensa, o usuário será removido de todas as listas de discussão do FreeBSD e filtrados de postagem posterior para elas. Lamentamos que tais regras e medidas sejam absolutamente necessárias, mas a Internet de hoje é um ambiente bastante hostil, ao que parece, e muitos não conseguem perceber o quão frágeis são alguns de seus mecanismos.

Regras Básicas:

- O tópico de qualquer postagem deve estar de acordo com o regulamento básico da lista para a qual ele é postado. Se a lista for sobre questões técnicas, a mensagem deve conter discussão técnica. Conversa irrelevante em curso ou provocações apenas prejudicam o valor da lista de discussão para todos e não será tolerado. Para discussões de forma livre sobre um tópico em particular, a lista de discussão do chat do FreeBSD está disponível gratuitamente e deve ser usada para isso.
- Nenhuma postagem deve ser feita para mais de 2 listas de discussão, e apenas para 2 quando houver necessidade clara e óbvia de postar nas duas listas. Para a maioria das listas, já existe uma grande quantidade de sobreposições de assinantes e, exceto pelas mixagens mais esotéricas (digamos " -stable & -scsi "), não há motivo para postar em mais de uma lista ao mesmo tempo. Se uma mensagem for recebida com várias listas de discussão na linha `Cc`, ajuste a linha `Cc` antes de responder. *A pessoa que responde ainda é responsável por postagens cruzadas, independentemente de quem tenha sido o remetente.*
- Ataques pessoais e palavrões (no contexto de um argumento) não são permitidos, e isso inclui

usuários e desenvolvedores. Violações brutais da netiqueta, como a extração ou repostagem de correspondência privada quando a permissão para fazer isso não existe, são desaprovadas, mas não especificamente forçadas. *No entanto*, também existem muito poucos casos em que tal conteúdo se encaixaria no estatuto de uma lista e, portanto, provavelmente ele geraria uma advertência (ou proibição).

- A publicidade de produtos ou serviços não relacionados ao FreeBSD é estritamente proibida e resultará em uma proibição imediata se for claro que o ofensor está anunciando por spam.

Estatutos individuais das listas:

freebsd-acpi

ACPI e desenvolvimento de gestão de energia

freebsd-announce

Eventos / marcos importantes

Esta é a lista de discussão para pessoas interessadas apenas em anúncios ocasionais de eventos significativos do FreeBSD. Isso inclui anúncios sobre snapshots e outros releases. Ela contém anúncios de novos recursos do FreeBSD. Pode conter chamadas para voluntários, etc. Esta é uma lista de discussão de baixo volume, estritamente moderada.

freebsd-arch

Discussão sobre arquitetura e design

Esta lista é para discussão da arquitetura do FreeBSD. As mensagens serão principalmente mantidas estritamente de natureza técnica. Exemplos de tópicos adequados são:

- Como fazer um re-vamp do sistema de compilação para ter várias compilações personalizadas em execução ao mesmo tempo.
- O que precisa ser corrigido com o VFS para fazer com que as camadas Heidemann funcionem.
- Como podemos mudar a interface do driver de dispositivo para poder usar os mesmos drivers de forma limpa em muitos barramentos e arquiteturas.
- Como escrever um driver de rede.

freebsd-bluetooth

Bluetooth ™ no FreeBSD

Este é o fórum onde os usuários de Bluetooth™ no FreeBSD se reúnem. Problemas de design, detalhes de implementação, patches, relatórios de bugs, relatórios de status, solicitações de recursos e todos os assuntos relacionados a Bluetooth ™ são bem vindos.

freebsd-bugbusters

Coordenação sobre o esforço de manuseio dos Relatórios de Problemas

O objetivo desta lista é servir como um fórum de coordenação e discussão para o Bugmeister, seus Bugbusters e quaisquer outras partes que tenham interesse genuíno no banco de dados de RP. Esta lista não é para discussões sobre bugs específicos, patches ou PRs.

freebsd-bugs

Relatórios de bugs

Esta é a lista de discussão para reportar bugs no FreeBSD. Sempre que possível, os bugs devem ser submetidos usando a interface web.

freebsd-chat

Itens não técnicos relacionados à comunidade FreeBSD

Esta lista contém o overflow de outras listas sobre informações sociais não técnicas. Ela inclui discussões sobre se Jordan se parece com um furão ou não, se deve ou não digitar em maiúsculas, quem está tomando muito café, onde a melhor cerveja é preparada, quem está fazendo cerveja no porão, e assim por diante. Anúncios ocasionais de eventos importantes (como festas, casamentos, nascimentos, novos empregos, etc) podem ser feitos para as listas técnicas, mas os acompanhamentos devem ser direcionados para esta lista de bate-papo.

freebsd-chromium

Questões específicas sobre o Chromium no FreeBSD

Esta é uma lista para a discussão do suporte ao Chromium no FreeBSD. Esta é uma lista técnica para discutir o desenvolvimento e a instalação do Chromium.

freebsd-cloud

Executando o FreeBSD em várias plataformas de nuvem

Esta lista discute a execução do FreeBSD no Amazon EC2, no Google Compute Engine, no Microsoft Azure e em outras plataformas de computação em nuvem.

freebsd-core

FreeBSD core team

Esta é uma lista de discussão interna para uso pelos membros do core team. Mensagens podem ser enviadas quando um assunto sério relacionado ao FreeBSD requer arbitragem ou escrutínio de alto nível.

freebsd-current

Discussões sobre o uso do FreeBSD-CURRENT

Esta é a lista de discussão para usuários do FreeBSD-CURRENT. Ela inclui avisos sobre novos recursos que estão sendo lançados no -CURRENT que afetarão os usuários e instruções sobre as etapas que devem ser seguidas para permanecer no -CURRENT. Qualquer um que esteja executando o "CURRENT" deve se inscrever nesta lista. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-desktop

Usando e melhorando o FreeBSD no desktop

Este é um fórum para discussão do FreeBSD no desktop. É principalmente um lugar para portadores de desktop e usuários discutirem problemas e melhorarem o suporte do FreeBSD

para desktops.

dev-ci

Coordenação do Relatório de Problemas sobre o esforço de manuseio

Todos os relatórios de integração contínua, resultados de compilação e testes

dev-reviews

Notificações do trabalho em andamento na ferramenta de revisão do FreeBSD

Notificações automatizadas de trabalhos em andamento para revisão nas ferramentas de revisão do FreeBSD, incluindo patches.

freebsd-doc

Projeto de Documentação

Esta lista de discussão é para a discussão de questões e projetos relacionados à criação de documentação para o FreeBSD. Os membros desta lista são coletivamente referidos como "The FreeBSD Documentation Project". É uma lista aberta; sinta-se à vontade para participar e contribuir!

freebsd-drivers

Escrevendo drivers de dispositivos para o FreeBSD

Este é um fórum para discussões técnicas relacionadas a drivers de dispositivos no FreeBSD. É principalmente um lugar para os criadores de drivers de dispositivo fazerem perguntas sobre como escreverem drivers de dispositivo usando as APIs no kernel do FreeBSD.

freebsd-dtrace

Usando e trabalhando no DTrace no FreeBSD

O DTrace é um componente integrado do FreeBSD que fornece uma estrutura para entender o kernel, bem como programas de espaço do usuário em tempo de execução. A lista de discussão é uma discussão arquivada para desenvolvedores do código, bem como aqueles que a usam.

freebsd-eclipse

Usuários FreeBSD do IDE Eclipse, ferramentas, aplicativos e ports rich clients.

A intenção desta lista é fornecer suporte mútuo para tudo relacionado com a escolha, instalação, uso, desenvolvimento e manutenção do IDE Eclipse, ferramentas, aplicativos rich client na plataforma FreeBSD e para assistência na portabilidade do IDE Eclipse e seus plugins para o ambiente FreeBSD.

A intenção é também facilitar a troca de informações entre a comunidade Eclipse e a comunidade FreeBSD para benefício mútuo de ambas.

Embora essa lista esteja focada principalmente nas necessidades dos usuários do Eclipse, ela também fornecerá um fórum para aqueles que gostariam de desenvolver aplicativos específicos para o FreeBSD usando o Framework do Eclipse.

freebsd-embedded

Usando o FreeBSD em aplicações embarcadas

Esta lista discute tópicos relacionados ao uso do FreeBSD em sistemas embarcados. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico. Para o propósito desta lista, os sistemas embarcados são aqueles dispositivos de computação que não são desktops e que geralmente servem a um único propósito, ao invés de serem ambientes de computação geral. Os exemplos incluem, mas não estão limitados a, todos os tipos de aparelhos telefônicos, equipamentos de rede, como roteadores, switches e PBXs, equipamentos de medição remota, PDAs, sistemas Point Of Sale e assim por diante.

freebsd-emulation

Emulação de outros sistemas como o Linux/MS-DOS™/Windows™

Este é um fórum para discussões técnicas relacionadas à execução no FreeBSD de programas escritos para outros sistemas operacionais.

freebsd-enlightenment

Enlightenment

Discussões sobre o Ambiente de Desktop Enlightenment para sistemas FreeBSD. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-eol

Suporte de pares para softwares relacionados ao FreeBSD que não são mais suportados pelo Projeto FreeBSD.

Esta lista é para aqueles interessados em fornecer ou fazer uso de suporte de software relacionado ao FreeBSD para o qual o Projeto FreeBSD não fornece mais suporte oficial na forma de avisos e patches de segurança.

freebsd-firewire

FireWire™ (iLink, IEEE 1394)

Esta é uma lista para discussão do design e implementação de um subsistema FireWire™ (também conhecido como IEEE 1394 aka iLink) para o FreeBSD. Tópicos relevantes incluem especificamente os padrões, dispositivos de barramento e seus protocolos, placas adaptadoras / placas / chips sets e a arquitetura e implementação de código para seu suporte adequado.

freebsd-fortran

Fortran no FreeBSD

Esta é a lista para discussão de ports relacionados ao Fortran no FreeBSD: compiladores, bibliotecas, aplicativos científicos e de engenharia, de laptops a clusters de HPC.

freebsd-fs

Sistemas de arquivos

Discussões sobre os sistemas de arquivos do FreeBSD. Esta é uma lista de discussão técnica para

a qual é esperado conteúdo estritamente técnico.

freebsd-games

Jogos no FreeBSD

Esta é uma lista técnica para discussões relacionadas a trazer jogos para o FreeBSD. É para indivíduos trabalhando ativamente em portar jogos para o FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

freebsd-gecko

Motor Gecko de Renderização

Este é um fórum sobre aplicativos Gecko usando o FreeBSD.

Discussão em torno dos Ports dos aplicativos Gecko, sua instalação, seu desenvolvimento e seu suporte dentro do FreeBSD.

freebsd-geom

GEOM

Discussões específicas sobre o GEOM e implementações relacionadas. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-git

Uso do git no projeto FreeBSD

Discussões sobre como usar o git na infra-estrutura do FreeBSD, incluindo o espelho do github e outros usos do git para colaboração no projeto. Área de discussão para pessoas usando o git no espelho do FreeBSD no github. Pessoas que querem começar com o espelho ou o git em geral no FreeBSD podem fazer perguntas aqui.

freebsd-gnome

GNOME

Discussões relativas ao ambiente de trabalho GNOME para sistemas FreeBSD. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-infiniband

Infiniband no FreeBSD

Lista técnica para discutir Infiniband, OFED e OpenSM no FreeBSD.

freebsd-ipfw

Firewall IP

Este é o fórum para discussões técnicas sobre o redesenho do código de firewall IP no FreeBSD. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-isdn

Comunicações ISDN

Esta é a lista de discussão para pessoas discutindo o desenvolvimento do suporte a ISDN para o FreeBSD.

freebsd-java

Desenvolvimento Java™

Esta é a lista de discussão para as pessoas que discutem o desenvolvimento de aplicações Java™ para o FreeBSD e a portabilidade e manutenção de JDK™s.

freebsd-jobs

Ofertas e Procura de Emprego

Este é um fórum para postar avisos de emprego especificamente relacionados ao FreeBSD e currículos daqueles que buscam emprego relacionado ao FreeBSD. Esta *não* é uma lista de discussão para questões gerais de emprego, já que fóruns adequados para isso já existem em outros lugares.

Note que esta lista, como as demais listas de discussão do `FreeBSD.org`, é distribuída em todo o mundo. Seja claro sobre a localização geográfica e até que ponto o trabalho remoto ou a assistência à realocação estão disponíveis.

O email deve usar somente formatos abertos - preferencialmente texto puro, mas o formato básico de documento portátil (PDF), HTML e alguns outros são aceitáveis para muitos leitores. Formatos fechados como Microsoft™ Word (.doc) serão rejeitados pelo servidor da lista de discussão.

freebsd-kde

KDE

Discussões sobre o KDE em sistemas FreeBSD. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-hackers

Discussões técnicas

Este é um fórum para discussões técnicas relacionadas ao FreeBSD. Esta é a principal lista de discussão técnica. É para indivíduos trabalhando ativamente no FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-hardware

Discussão geral sobre hardware no FreeBSD

Discussão geral sobre os tipos de hardware em que o FreeBSD executa, vários problemas e sugestões sobre o que comprar ou evitar.

freebsd-hubs

Sites Espelhos

Anúncios e discussões para pessoas que executam sites espelho do FreeBSD.

freebsd-isp

Problemas para provedores de serviços de Internet

Esta lista de discussão é para discutir tópicos relevantes para provedores de serviços de Internet (ISPs) usando o FreeBSD. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-mono

Aplicações Mono e C# no FreeBSD

Esta é uma lista de discussões relacionadas ao framework de desenvolvimento Mono no FreeBSD. Esta é uma lista de discussão técnica. É para indivíduos trabalhando ativamente na portabilidade de aplicativos Mono ou C# para o FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

freebsd-ocaml

Discussões específicas sobre OCaml no FreeBSD

Esta é uma lista para discussões relacionadas ao suporte OCaml no FreeBSD. Esta é uma lista de discussão técnica. É para pessoas que trabalham com ports OCaml, bibliotecas de terceiros e frameworks. Indivíduos interessados na discussão técnica também são bem vindos.

freebsd-office

Aplicativos de Escritório no FreeBSD

Discussão em torno de aplicativos de escritório, sua instalação, seu desenvolvimento e seu suporte dentro do FreeBSD.

freebsd-ops-announce

Anúncios de infra-estrutura do projeto

Esta é a lista de discussão para pessoas interessadas em mudanças e questões relacionadas à infra-estrutura do Projeto FreeBSD.org.

Esta lista moderada é estritamente para anúncios: sem respostas, pedidos, discussões ou opiniões.

freebsd-performance

Discussões sobre o tunning ou aceleração do FreeBSD

Esta lista de discussão existe para fornecer um local para hackers, administradores e/ou partes interessadas discutirem tópicos relacionados ao desempenho do FreeBSD. Temas aceitáveis incluem falar sobre instalações do FreeBSD que estão sob alta carga, que estão tendo problemas de desempenho ou que estão forçando os limites do FreeBSD. Partes interessadas que estejam

dispostas a trabalhar para melhorar o desempenho do FreeBSD são altamente encorajadas a assinar esta lista. Esta é uma lista altamente técnica destinada para usuários experientes do FreeBSD, hackers ou administradores interessados em manter o FreeBSD rápido, robusto e escalável. Essa lista não é uma lista de perguntas e respostas que substitui a leitura da documentação, mas é um local para fazer contribuições ou perguntar sobre tópicos não respondidos relacionados ao desempenho.

freebsd-pf

Discussão sobre o sistema de firewall de filtro de pacotes

Discussão sobre o sistema de firewall de filtro de pacotes (pf) no FreeBSD. A discussão técnica e as perguntas dos usuários são bem-vindas. Esta lista também é um lugar para discutir o framework ALTQ QoS.

freebsd-pkg

Discussão sobre o gerenciamento de pacotes binários e as ferramentas de pacotes

Discussão de todos os aspectos de gerenciamento de sistemas FreeBSD usando pacotes binários para instalar software, incluindo toolkits e formatos de pacotes binários, seu desenvolvimento e suporte dentro do FreeBSD, gerenciamento de repositórios de pacotes e pacotes de terceiros.

Observe que a discussão de ports que não conseguem gerar pacotes corretamente geralmente deve ser considerada como um problema do port e, portanto, é inadequada para essa lista.

freebsd-pkg-fallout

Registros de fallout da construção de pacotes

Todos os logs de falha na compilação de pacotes nos clusters de compilação de pacotes

freebsd-pkgbase

Empacotando do sistema básico do FreeBSD.

Discussões sobre a implementação e questões relacionadas ao empacotamento do sistema base do FreeBSD.

freebsd-platforms

Portando para plataformas não Intel™

Problemas de plataforma cruzada do FreeBSD, discussão geral e propostas para ports do FreeBSD para plataformas não Intel™. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-ports

Discussão dos "ports"

Discussões relativas à coleção de "ports" (/usr/ports) do FreeBSD, infra-estrutura de ports e esforços gerais de coordenação de ports. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-ports-announce

Notícias e instruções importantes sobre a "Coleção de Ports" do FreeBSD

Notícias importantes para desenvolvedores, porters e usuários da "Coleção de Ports" (/usr/ports), incluindo alterações de arquitetura / infraestrutura, novos recursos, instruções críticas de upgrade e informações sobre a engenharia de releases. Esta é uma lista de discussão de baixo volume, destinada a anúncios.

freebsd-ports-bugs

Discussão de bugs dos "ports"

Discussões sobre relatórios de problemas para a "coleção de ports" do FreeBSD (/usr/ports), ports propostos ou modificações nos ports. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-proliant

Discussão técnica do FreeBSD nas plataformas de servidores HP ProLiant

Esta lista de discussão deve ser usada para a discussão técnica do uso do FreeBSD em servidores HP ProLiant, incluindo a discussão de drivers específicos do ProLiant, software de gerenciamento, ferramentas de configuração e atualizações do BIOS. Como tal, este é o principal local para discutir os módulos hpasmd, hpasmcli e hpacucli.

freebsd-python

Python no FreeBSD

Esta é uma lista de discussões relacionadas à melhoria do suporte ao Python no FreeBSD. Esta é uma lista de discussão técnica. É para indivíduos que estão trabalhando na portabilidade do Python, seus módulos de terceiros e coisas do Zope para o FreeBSD. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

freebsd-questions

Questões do usuário

Esta é a lista de discussão para questões sobre o FreeBSD. Não envie perguntas do tipo "how to" para as listas técnicas, a menos que a questão seja bastante técnica.

freebsd-ruby

Discussões sobre Ruby específicas para o FreeBSD

Esta é uma lista para discussões relacionadas ao suporte Ruby no FreeBSD. Esta é uma lista de discussão técnica. É para pessoas que trabalham com ports Ruby, bibliotecas de terceiros e frameworks.

Indivíduos interessados na discussão técnica também são bem vindos.

freebsd-scsi

Subsistema SCSI

Esta é a lista de discussão para pessoas que trabalham no subsistema SCSI do FreeBSD. Esta é

uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-security

Questões de segurança

Problemas de segurança do FreeBSD (DES, Kerberos, falhas de segurança conhecidas e correções, etc). Esta é uma lista de discussão técnica para a qual se espera uma discussão estritamente técnica. Note que esta não é uma lista de perguntas e respostas, mas as contribuições (ambas as perguntas e respostas) para o FAQ são bem-vindas.

freebsd-security-notifications

Notificações de segurança

Notificações de problemas de segurança e correções do FreeBSD. Esta não é uma lista de discussão. A lista de discussão é a FreeBSD-security.

freebsd-snapshots

Anúncios de Snapshots de Desenvolvimento do FreeBSD

Esta lista fornece notificações sobre a disponibilidade de novos snapshots de desenvolvimento do FreeBSD para head/ e stable/ branches.

freebsd-stable

Discussões sobre o uso do FreeBSD-STABLE

Esta é a lista de discussão para usuários do FreeBSD-STABLE. O "STABLE" é o ramo onde o desenvolvimento continua depois de um RELEASE, incluindo correções de bugs e novos recursos. O ABI é mantido estável para compatibilidade binária. Ela inclui avisos sobre novos recursos que estarão sendo incorporados no -STABLE e que afetarão os usuários e instruções sobre as etapas que devem ser seguidas para permanecer -STABLE. Qualquer um que esteja executando o "STABLE" deve assinar esta lista. Esta é uma lista de discussão técnica para a qual é esperado conteúdo estritamente técnico.

freebsd-standards

Conformidade C99 & POSIX

Este é um fórum para discussões técnicas relacionadas à Conformidade do FreeBSD com os padrões C99 e POSIX.

freebsd-teaching

Ensinando com o FreeBSD

Lista de discussão não técnica para discutir o ensino com o FreeBSD.

freebsd-testing

Testando no FreeBSD

Lista de discussão técnica discutindo testes no FreeBSD, incluindo ATF/Kyua, infraestrutura de testes, testes de port para o FreeBSD de outros sistemas operacionais (NetBSD, ...), etc.

freebsd-tex

Portando o TeX e seus aplicativos para o FreeBSD

Esta é uma lista de discussão técnica para discussões relacionadas ao TeX e suas aplicações no FreeBSD. É destinada aos indivíduos que estão trabalhando ativamente na portabilidade do TeX para o FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

freebsd-toolchain

Manutenção do toolchain integrado do FreeBSD

Esta é a lista para discussões relacionadas à manutenção do conjunto de ferramentas fornecido com o FreeBSD. Isso pode incluir o estado do Clang e do GCC, mas também partes de software, como assemblers, vinculadores e depuradores.

freebsd-transport

Discussões de protocolos de rede em nível de transporte no FreeBSD

A lista de discussão de transporte existe para a discussão de problemas e projetos em torno dos protocolos de nível de transporte na pilha de rede do FreeBSD, incluindo TCP, SCTP e UDP. Outros tópicos de rede, incluindo questões específicas de driver e protocolos de rede, devem ser discutidos na lista de discussão FreeBSD networking .

freebsd-translators

Traduzindo documentos e programas do FreeBSD

Uma lista de discussão em que tradutores de documentos do FreeBSD do inglês para outros idiomas podem falar sobre métodos e ferramentas de tradução. Novos membros são convidados a se apresentar e mencionar os idiomas em que estão interessados em traduzir.

freebsd-usb

Discutindo o suporte do FreeBSD para USB

Esta é uma lista de discussão para discussões técnicas relacionadas ao suporte do FreeBSD para USB.

freebsd-user-groups

Lista de Coordenação do Grupo de Usuários

Esta é a lista de discussão dos coordenadores de cada um dos Grupos de Usuários da área local para discutir assuntos entre si e um indivíduo designado do Core Team. Essa lista de e-mail deve se limitar a atender a sinopse e a coordenação de projetos que abranjam Grupos de usuários.

freebsd-virtualization

Discussão de várias técnicas de virtualização suportadas pelo FreeBSD

Uma lista para discutir as várias técnicas de virtualização suportadas pelo FreeBSD. Por um lado, o foco estará na implementação da funcionalidade básica, bem como na adição de novos recursos. Por outro lado, os usuários terão um fórum para pedir ajuda em caso de problemas ou

para discutir seus casos de uso.

freebsd-wip-status

Status do andamento do trabalho no FreeBSD

Esta lista de discussão pode ser usada pelos desenvolvedores para anunciar a criação e o progresso do trabalho relacionado ao FreeBSD. As mensagens serão moderadas. Sugere-se enviar a mensagem "Para:" uma lista mais atual do FreeBSD e apenas "BCC:" esta lista. Dessa forma, o WIP também pode ser discutido na lista de tópicos, já que nenhuma discussão é permitida nesta lista.

Olhe dentro dos arquivos para exemplos de mensagens adequadas.

Um resumo editorial das mensagens para esta lista pode ser postado no site do FreeBSD todos os meses como parte dos Relatórios de Status . Relatórios anteriores são arquivados.

freebsd-wireless

Discussões da pilha 802.11, desenvolvimento de driver de dispositivo de ferramentas

A lista FreeBSD-wireless se concentra na pilha 802.11 (sys/net80211), no driver do dispositivo e no desenvolvimento de ferramentas. Isso inclui bugs, novos recursos e manutenção.

freebsd-xen

Discussão do port do FreeBSD para Xen ™ - implementação e uso

Uma lista focada no port do Xen™ para o FreeBSD. O nível de tráfego previsto é pequeno o suficiente para servir como um fórum para discussões técnicas sobre os detalhes de implementação e design, bem como problemas administrativos de implantação.

freebsd-xfce

XFCE

Este é um fórum para discussões relacionadas a trazer o ambiente XFCE para o FreeBSD. Esta é uma lista de discussão técnica. É para indivíduos que trabalham ativamente portando o XFCE para o FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

freebsd-zope

Zope

Este é um fórum para discussões relacionadas a trazer o ambiente Zope para o FreeBSD. Esta é uma lista de discussão técnica. É para indivíduos que trabalham ativamente portando o Zope para o FreeBSD, para trazer problemas ou discutir soluções alternativas. Indivíduos interessados em acompanhar a discussão técnica também são bem vindos.

C.2.4. Filtros nas Listas de Discussão

As listas de discussão do FreeBSD são filtradas de várias maneiras para evitar a distribuição de spam, vírus e outros e-mails indesejados. As ações de filtragem descritas nesta seção não incluem todas aquelas usadas para proteger as listas de discussão.

Apenas determinados tipos de anexos são permitidos nas listas de discussão. Todos os anexos com um tipo de conteúdo MIME não encontrado na lista abaixo serão removidos antes que um email seja distribuído nas listas de discussão.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch



Algumas das listas de discussão podem permitir anexos de outros tipos de conteúdo MIME, mas a lista acima deve ser aplicável para a maioria das listas de discussão.

Se um email contiver uma versão em HTML e uma em texto simples, a versão em HTML será removida. Se um email contiver somente uma versão em HTML, ele será convertido em texto simples.

C.3. Grupos de Notícias Usenet

Além de dois grupos de noticias específicos sobre FreeBSD, existem muitos outros em que o FreeBSD é discutido ou que são relevantes para usuários do FreeBSD.

C.3.1. Grupos de notícias específicos do BSD

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (German)
- fr.comp.os.bsd (French)

C.3.2. Outros Newsgroups de interesse sobre UNIX™

- comp.unix
- comp.unix.questions
- comp.unix.admin

- comp.unix.programmer
- comp.unix.shell
- comp.unix.misc
- comp.unix.bsd

C.3.3. X Window System

- comp.windows.x

C.4. Espelhos Oficiais

Central Servers, Armenia, Australia, Austria, Czech Republic, Denmark, Finland, France, Germany, Hong Kong, Ireland, Japan, Latvia, Lithuania, Netherlands, Norway, Russia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States of America.

(as of UTC)

Central Servers

- https://www.FreeBSD.org/

Armenia

- http://www.at.FreeBSD.org/ (IPv6)

Australia

- http://www.au.FreeBSD.org/
- http://www2.au.FreeBSD.org/

Austria

- http://www.at.FreeBSD.org/ (IPv6)

Czech Republic

- http://www.cz.FreeBSD.org/ (IPv6)

Denmark

- http://www.dk.FreeBSD.org/ (IPv6)

Finland

- http://www.fi.FreeBSD.org/

France

- http://www1.fr.FreeBSD.org/

Germany

- http://www.de.FreeBSD.org/

Hong Kong

- http://www.hk.FreeBSD.org/

Ireland

- http://www.ie.FreeBSD.org/

Japan

- http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

Latvia

- http://www.lv.FreeBSD.org/

Lithuania

- http://www.lt.FreeBSD.org/

Netherlands

- http://www.nl.FreeBSD.org/

Norway

- http://www.no.FreeBSD.org/

Russia

- http://www.ru.FreeBSD.org/ (IPv6)

Slovenia

- http://www.si.FreeBSD.org/

South Africa

- http://www.za.FreeBSD.org/

Spain

- http://www.es.FreeBSD.org/
- http://www2.es.FreeBSD.org/

Sweden

- http://www.se.FreeBSD.org/

Switzerland

- http://www.ch.FreeBSD.org/ (IPv6)
- http://www2.ch.FreeBSD.org/ (IPv6)

Taiwan

- http://www.tw.FreeBSD.org/
- http://www2.tw.FreeBSD.org/
- http://www4.tw.FreeBSD.org/
- http://www5.tw.FreeBSD.org/ (IPv6)

United Kingdom

- http://www1.uk.FreeBSD.org
- http://www3.uk.FreeBSD.org/

United States of America

- http://www5.us.FreeBSD.org/ (IPv6)

Apêndice D: Chaves OpenPGP

As chaves OpenPGP dos Administradores do FreeBSD.org são mostradas aqui. Essas chaves podem ser usadas para verificar uma assinatura ou para enviar um email criptografado para um dos administradores. A lista completa das chaves OpenPGP do FreeBSD está disponível no artigo PGP Keys. O keyring completo pode ser baixado em https://www.FreeBSD.org/doc/pgpkeyring.txt .

D.1. Administradores

D.1.1. Equipe de Oficiais de Segurança <security-officer@FreeBSD.org>

```
pub   rsa4096/D9AD2A18057474CB 2022-12-11 [C] [expires: 2026-01-24]
      Key fingerprint = 0BE3 3275 D74C 953C 79F8  1107 D9AD 2A18 0574 74CB
uid                           FreeBSD Security Officer <security-officer@freebsd.org>
sub   rsa4096/6E58DE901F001AEF 2022-12-11 [S] [expires: 2026-01-15]
sub   rsa4096/46DB26D62F6039B7 2022-12-11 [E] [expires: 2026-01-15]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGOVdeUBEADHF5VGg1iPbACB+7lomX6aDytUf0k2k2Yc/Kp6lfYv7JKU+1nr
TcNF7Gt1YkajPSeWRKNZw/X94g4w5TEOHbJ6QQWx9g+N7RjEq75actQ/r2N5zY4S
ujfFTepbvgR55mLTxlxGKFBmNrfNbpHRyh4GwFRgPlxf5Jy9SB+0m54yFS4QlSd0
pIzO0CLkjHUFy/8S93oSK2zUkgok5gLWruBXom+8VC3OtBElkWswPkE1pKZvMQCv
VyM+7BS+MCFXSdZczDZZoEzpQJGhUYFsdg0KqlLv6z1rP+HsgUYKTkRpcrumDQV0
MMuCE4ECU6nFDDTnbR8Wn3LF5oTt0GtwS0nWf+nZ1SFTDURcSPR4Lp/PKjuDAkOS
P8BaruCNx1ItHSwcnXw0gS4+h8FjtWNZpsawtzjjgApcl+m9KP6dkBcbN+i1DHm6
NG6YQVtVWyN8aOKmoC/FEm1CWh1bv+ri9XOkF2EqT/ktbjbT1hFoFGBkS9/35y1G
3KKyWtwKcyF4OXcArl6sQwGgiYnZEG3sUMaGrwQovRtMf7le3cAYsMkXyiAnEufa
deuabYLD8qp9L/eNo+9aZmhJqQg4EQb+ePH7bGPNDZ+M5oGUwReX857FoWaPhs4L
dAKQ1YwASxdKKh8wnaamjIeZSGP5TCjurH7pADAIaB3/D+ZNl2a7od+C1wARAQAB
tDdGcmVlQlNEIFNlY3VyaXR5IE9mZmljZXIgPHNlY3VyaXR5LW9mZmljZXJAZnJl
ZWJzZC5vcmc+iQJSBBMBCgA8AhsBBAsJCAcEFQoJCAUWAgMBAAIeBQIXgBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJjlXeQBQkF3u+rAAoJENmtKhgFdHTLOVoQALS3
cj7rqYkHiV4zDYrgPEp9O1kAyGI8VdfGAMkDVTqr+wP4v/o7LIUrgwZl5qxesVFB
VknFr0Wp5g9h0iAjasoI5sDd6tH2SmumhBHXFVdftzDQhrugxH6fWRhHs0SaFYCk
Qt5nFbcpUfWgtQ35XTbsL8iENdYpjKXsSFQrJneGSwxIjWYTFn6ps/AI3gwR8+Bn
OffEFdYugJ049O6Vu6YBFJHrnMO7NbF4v95dVYuLtpMIaXWM+V9KITmhaBzFz5fM
Q7UOzcLlbxOYKNIWcp8QQk429mayKW5VUeUExUD1ZzBHn+P6ZG7QTMDu/RmBqiHo
ewCMVz4n9uXT5BiOngE4CvS0WQwHzK+k9MLpG2u/Bo9+LT0Ceh9Ou1rfU5+0tRwl
GyOFFjf3INS7I7gkcAwxQ7dzDItN/UQPZpg8y9mABU2x4enz0AvTnb61d/1dnTEr
tdNgU433he0ZnD1HurZCjBEWC656wv6iMdWcD8gjhMbmEpPmjvXcYlTO6zhEygSM
DiwdQCWK2W4++YJerA6ULBi3niNWBpofOFH8XylV56ruhjtHCo7+/3carcMoPOJv
lVZ1zCKxLro3TRBT15JTFBGqblRyTopFK3PuxW//GTnZOtpQEOV6yL4RAXcWeC1d
1hb5k/YxUmRF6XsDNEH4b08T8ZO8dV3dAV43Wh1oiQEzBBABCAAdFiEEuyjUCzYO
7pNq7RVv5fe8y6O93fgFAmObXVYACgkQ5fe8y6O93fiBlwf/W8y1XXJIx1ZA3n6u
f7aS70rbP9KFPr4U0dixwKE/gbtIQ9ckeNXrDDWz0v0NCz4qS+33IPiJg1WcY3vR
W90e7QgAueCo5TdZPImPbCs42vadpa5byMXS4Pw+xyT+d/yp2oLKYbj3En4bg1GM
```

```
w71DezIjvV+e01UR++u1t9yZ8LOWM5Kumz1zyQLZDZ8qIKt1bBfpa+E0cEqtNQWu
iGhQE3AHI8eWV+jBkg5y2zHRIevbWb1UPsj43lgkFtAGHk9rrM8Rmgr4AXr531iD
srBwauKZ/MElcF3MINuLH+gkPPaFHw/YIpLRLaZXZVsw3Xi1RNXI2n2ea29dvs/C
Lcf1vYkCMwQQAQgAHRYhBPwOh4rlr+eIAo1jVdOXkvSep+XCBQJjm14FAAoJENOX
kvSep+XC0DcP/1ZB7k9p1T+9QbbZZE1PJiHby3815ccH3XKexbNmmakHIn3L6Cet
F891Kqt9ssbhFRMNtyZ/k/8y8Hv5bKxVep5/HMyK+8aqfDFN0WMrqZh0/CiR6DJh
gnAmPNw/hAVHMHaYGII9kCrFfPFJ02FKoc81g9F08odb7TV+UlvRjkErhRxF+dGS
wQoO0RCbf0Z1cs7nd0Vb2z4IJh4XMxBjWc/uQ2Q9dH/0uRzwpAnR4YX+MG5YrX7Z
zBvDyR0r76iQwRSDKgioNgkr6R3rq1NZGdaj+8b0LzdOqtzKJ/eupDe3+H67e/EN
qymtreGjrubpiU9bKvYArisUqhE5KtguryvR6Qz9bj87nPg33DT3WWGVrwFRxBox
dbWzjQFv0wug8m4GAwVF7fPR5/eW7IHw8zvgn0vSPcZz7MZ4e6Y5jN4kA5/xWJYZ
Sps54qQWB+FA30unIXN68KqdIzONIbtaY3W4/JjJUCm4T+wEjKaH+wJX8w1DMjlg
mkTmGh/UrTyC1vXbPgk9Sy3cRTICR1T9z7W8UlmTtnKrUklrjlFR7SXzrEXzLGOX
Fm+NEHpHNXqzcm6c3QfzY/yQ9HSAQ/t7SUQ9caRePbDz3/msyPxtGFor9roQv6VN
wRXCyRgkH4Y5tPhJAQ8G/FxX+VXFb93QL0lfelb23/BBu6cUwW63SRn5iHUEExYI
AB0WIQQeB2Johg/5/ikUnJwDU9SVF1S13AUCZISO3wAKCRADU9SVF1S13NnqAP95
LA1Om9XSAKl76VtV+L3JPDdAwIdbNa0OsRT4Wm7U3wD/YoFrdHXVHHQFKwYeUUhj
XZcxnZLe9Ixo0/JP+RVFVw65Ag0EY5V2yQEQAOqjzPpMUCGu8eElXnAd2PruC6hi
+lc/yC90KqizxIuW6qLQBaAkTCWq7suYpDqoygn7YM3rL50S285WAECAXrcst/cV
Aqr0UH/e6p4iJCUIiXcfjd/wq20RnN/+VuvLhjpCFLY5czfVS31D7Uh9MbC+zUTz
8nVTiNCsAao0qSdfJDIzB4nSO+9xIsme/dLsI5QlU5PdxOBV6HdEhCUXOoratJCb
KAOlLxtPwyMKxmv4oZ7Mqlt10peKjhpBb97qcIzJhHxujQZD0OmzIA6xoQ2eSCGd
xCEDsZ09kr3Esw1AwKnQ51xmWpFWNFk6627M1bo8+hzOz81CrTZhYrgE+1JXv6V1
L2A9lMsimdE1BHNycDS+dBOpIB9qxXCwAab4ykfvNxox/ZPDUrTy7v7mDI5uDNTN
CYYsKCj1UidycOKzSziB9Oa2uvmMJ5XstgNBf7Z8Cky1dtVd4ol6bU9L5nos9tbY
eSXFi4bmcWB7AJiVCMq6N+LBbUKWGLglB4TU1qhttpqv31X9V6ges5gARY/RuRTK
sVyhwsn7SDcqmNKRy0im2AYakwEp7hT07ulahOSLxjP+5hCf+nSJlwbxJ8ozwjjb
zeN2yLlJSI00klkIFBNUDt3wzFRW/n6qlf+/lepgzekfNrYMtfPB8AT07Z2A3U4x
lgiV346dZymbY/EjABEBAAGJBHIEGAEKACYWIQQL4zJ110yVPHn4EQfZrSoYBXR0
ywUCY5V2yQIbAgUJAgIpAAJACRDZrSoYBXR0y8F0IAQZAQoAHRYhBLYVJ36BCH33
XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5Ziz
IS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+Ks
Mu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6
wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXDrbEdAifbNp9KYxDigxdlOb0S14hj
CBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03
pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0
T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZ
WyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPxcwg7sbnKNApAKdi491dvH75s/U/O
wRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+Hs
V4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXW
HVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5v
fLrsTkaCLWF/SV9OdMykvYKU7ZAP+gKEwhp+HPFuOHZbOBhqFUdkfeCkdzX/QGdz
Tuz349roRhgz2vRfN7MtbtuzA6NWWhEWt5DcUgX/Y5I3Q4Z2bt3JiXQ6WJMgMMOX
Ar+XxtxyRRykc1HV3DQ/cq8OWYubNnIbgebPNIFr2OIWKsR9yDaucZzpmLfzaMZU
Au5hWmU9fIw5SIKGnQABBnNMhilfD+CkETp6baTvjTK4rpaobjJdeCTrsWgfXRNC
8x3hDvcrjPD70MylOGVQdx8GYChWJnCKXsLTGX7KwdfxkjclTyzWvdcCemp0eLha
mLGb9y1dtWdNIDcVCvZJy0lipHVUdFYYxb4iLZJANL631tlPM6AA8sO1/L4mqEGn
AIHVrUQd+2QkSiOl9mKlpgaR/fJz683BR5Qen9ywX0JPtBupqPW3t9VbO/uNxUql
HCeAhPi9NLOpujpYLfgW5QAfS3u0nkp5nrbkCoQUua2q0Oj7J0mFmtWtcE1c9+TH
mFJVb8j2G9yQw3ADe3Qp9ALazP5nVDVri8NZBhHK1/KuBmRYZtcyfqXUnKoiiWAl
m5rHaRiztW7e3wqm2oJu/RkEAagybutEuBWh2Ej2+gDxjEKKtIKGu54lif4kqTww
```

```
jKTcN1ekGihwwgCMUkBSBeNXk1ClkzLFHwESJCcFwdEgpVYQTKFsu0emYISyco3I
pUajGzfUiQRyBBgBCgAmAhsCFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmWFy28F
CQPyfaYCQMF0IAQZAQoAHRYhBLYVJ36BCH33XIGDO25Y3pAfABrvBQJjlXbJAAoJ
EG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5ZizIS02YHhSVMVYKS2T9jPIKi1qxnEi
Ew9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+KsMu8jwm1fUmIrefAx6fCVfCWRECT1
MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6wPCc0FkpiY7yDB9ihu1+gbE5Hg6d
vftttRXDrbEdAifbNp9KYxDigxdlOb0S14hjCBysLWH5Su/khcIlkeuqZcI8TmDl
dnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03pLsOdwhi31W6kjk4KvW5FKrOpoEw
UMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0T5031qk0CCRRa/+/iij/p2RG7c1m
x7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZWyCaoAecdbukwzDwUCUqHJFIWeFt
but7SOPxcwg7sbnKNApAKdi491dvH75s/U/OwRYO/2P+ymHlqtyix2jq0ReSVYcQ
PXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+HsV4Jnnp1zAJZ0KlKW/oJxbNFdI1TI
mkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXWHVAMR8Z+GvBY/A6OdexpibkTvC/z
Dr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5vfLrsTkaCLWF/SV9OdMykvYKUCRDZ
rSoYBXR0yyOqEACitDvbkbfjaton6izr4T8QU2yvhJHkf4B6KeVDbKY1J4784OxX
p2bJgPeF53SYBe8gm3YHjp8ULh4A/19U4hswyE8ymcm5nIs8OLyBdxkuBZJGEnzx
H3woiyYqWH7991kzhEjUkuMgKLuTI1HiO0oLMuPQNhUHOnWafSVPC0XO/tIL12Om
oUuc7ligY9Z9AceFjTZOuHamixHAAc6hpxdIW+yhC/qTpc2VK0niWeuQfq3453iR
Tf9MnR5Beztl3ZYRWcx7UiFuKGwZwBibNnNmUs6GyQcJ5UTa1oeJcLqHi0Lf/r0j
Xo3wgJq7EZjjVyU+GI2ZVoDOaS6c4/OvLm62XoeSlnn/dQxUcjUki+x8lb69IxSF
1xAgsC/oNtFZYd5rHdlnqIBUYK0lLtSCXBkzVeivSiQa0hL5on8LDu1nw2bXyW61
yt/YxVb4FanMxAqdYVBhOfU0RaPNifH01rbb4TwC9bTZN1LQ1KI/Swb/SruUE0Ry
T28fhYRtsReS2PnUODghJSFDJbwFbBZf6RKI16q1xqKRRvxIWPm+lMOi1NLOKR9P
+OKy9HmChMw0UJUcVl1cJ2xtRl3wi5t6AA6HoNv/TrLeYVgMR9wYmKlpvjTQ5jTd
rbHD1XP5jGsp8QsJMGja1m/7cryReCpcVxvImeReaOdgz+zDmQqq3O5zuIkEcgQY
AQoAJgIbAhYhBAvjMnXXTJU8efgRB9mtKhgFdHTLBQJnhEEJBQkF0/JAAkDBdCAE
GQEKAB0WIQS2FSd+gQh991yBgztuWN6QHwAa7wUCY5V2yQAKCRBuWN6QHwAa77gb
EADpUBT14cesITuMsOWYsyEtNmB4UlTFWCktk/YzyCotasZxIhMPXih9G1tDo9Ex
IWT8jNjSSA+w0Viua/PirDLvI8JtX1JiK3nwMenwlXwlkRAk9TJWy944YegHF/5y
tntwZ/L4BMYc3MztyZbw+sDwnNBZKYmO8gwfYobtfoGxOR4Onb37bbUVw62xHQIn
2zafSmMQ4oMXZTm9EteIYwgcrC1h+Urv5IXCJZHrqmXCPE5g5XZ1G9jqkwlaRYWj
cLD0qxwc5m9LNrF6OBS9N6S7DncIYt9VupI5OCr1uRSqzqaBMFDClTTH+dAx3b6J
1KFB0UiHP3FeTalFh8L3NE+dN9apNAgkUWv/v4oo/6dkRu3NZse2RAo/o2X5r40q
k/lhydQRZTSTFsiuH3VUWVsgmqAHnHW7pMMw8FAlKhyRSFnhbW7re0jj8XMIO7G5
yjQKQCnYuPdXbx++bP1PzsEWDv9j/sph5arcosdo6tEXklWHED17MEPIton1+NRf
sU0peEVggQXlwdTcZN/h7FeCZ56dcwCWdCpSlv6CcWzRXSNUyJpKa9qfIqBX/mon
jy7w5IHmhvLwAYI6IoT11h1QDEfGfhrwWPwOjnXsaYm5E7wv8w69PxMbOJbMpWSg
8L7xW3LXKR1VwXggUC1+b3y67E5Ggi1hf0lfTnTMpL2ClAkQ2a0qGAV0dMsNxRAA
suW1aLh+hgydW+iH6DmdQRMESsB1kE02kO1462TAQaziIAvNoxw5h48xvyEnrDA8
d+9IDMyxdrLmAbndUlSveMa9+EPiGHwr6VTyFL8nA5F7DcFi4mjEyGKe18JcaAlY
UtvHgWH6EjiX2iSXpsrJFEhtfFNolZ5sp9LFI6hOBihSJxZK4sbMR7Q6IkDuAVpT
FLiejBRlsXpFvTGL6040CtxbL5cqkVMYP38rFMTuc3pGGJA4wb5EC1dGjUi6XjbY
H7kuCAFyXqV9eQQP61x7K9W8qnXW+weCIMKfSX7AcCtH1jXBAM6lqpPrh6amc+/r
bg2eNA7DmgJnEY4apIcDB/b4khRMga2ozeGWWyIvOaVvR2R7ALQ+Rgut85cM+4+V
l2PHmOzW/yYdHVb5REQItFR5COb/mGUqYhkCtiV3nXo/K0uOQKu5SBbNzLuNvuwd
n+Eimxjl18VnrGG7sjtUa0MLmtr62GiEVrhrDqa/biHp8LdWkAQjLZ4aTRh2XZig
gaVFZHmkw3ILPyKKM21UXdM0YRk3TGVK8ODQy58ebPS4v9yYT9gUA9UDkDYeGcF2
qjoDPVNvcG6H8jCSsPRl1KZwtqqITCOSAIAPI4Nu97kO6nbOyQpYlwjd1MhvVXnP
66mHSvmqaxbNGX1mF9B/yERkBkooNZrKuJSvBTC2J1q5Ag0EY5V3BwEQAMpFVczZ
o9ZPNsgW791UW5o6wnrnd1nIO+S4rc37q2TEz8KGHCuxo5NwffZ2t6Ln04BI54pb
apg17b7a0hPka37HFkL28n4VyMdx0CsAm3QEfUsdK6xwKV2SucYeVcrV1upcN4Pd
```

```
XD7su1I7/A4CWXFJG047zJ0Z89lJZiQEiAq7ghvEoinC0sm+0a6ao/ocqCgWCKM1
yCPOyzJXleRrv29SRnYziMR+q2U0x9xg9Xl6GMwUmFwbJc9nORVvLH7fbU6/du8E
goAYrglFOFZG/TSolSGWRSMiavz0JSD/i+rEN4aIT4WfBe+L9Wy1AmrNxiAO+zKm
zHQu3JSxDncr+y+hcd+W0gqw10FoI9jWLcL7kR+6a0iOjuJSXSopq2l3DafiPxtC
Fmr4CGQhzBHM6e4/v/NNd3F0XpVbJ6RQph7lkfvfz8q2lvUlHhezJ0p1xXmhff9C
HjdVMhmAmz5+imBAXk2mottNfKb0pFEen1xY3K/UPA4g+oPsSj495MsvIg9eIMCc
C3/z0SEUMWH/styyJzPqfpyfGwZeTcIj9vg2o+RnGvmcLVYA/EGToPk905kv/cK7
3oy8bZyOB0zMg7T9PaWgLUO0sqjqo0Mw3knFySg3oRXlcilPQvfPdX0JvwLpc9DW
lr1+1GkCXJ08lWugJc96CJQupKRb1IbC0oUXABEBAAGJAjwEGAEKACYWIQQL4zJ1
10yVPHn4EQfZrSoYBXR0ywUCY5V3BwIbDAUJAgIpAAAKCRDZrSoYBXR0ywwtD/wI
DmEcHdFlyFRTomUBjbeK2uzcZIhkkgL58lc63UPle5iJ2FBvmYS+0rQS53sVEscc
n5KfkOwTryKllvWbl0IzuiqfawxALcfWpfZJHzTMSnDHfgXvOOyFMQruqRDAHAr7
PNC0CnbT0sEF2ZFzad8M9fLqtkXUx4mgECNGJ4CVqg75KY8uUzv/BmRwEf587FT5
/iAIed5MjFB2VFDX9GABcvTTbHxCZIxnxl3cs15SxT0lAofZ2ueU6kWYWZSXFeaE
M/4ymPJws2mmV0AkbJghLXCn9Mx3nX6NTZZ9Harbru+RzW3/Hg3DZd0J9vko8Paf
P0l1NWtgyX74CqvTgjzTxXTnqrRXzcczK7fhcC2u4i0prPtXXcyyi7SwpoLikaZC
LFFhUmOx+mS5TjtgFyFZBNxnO7iAwkzfcTcC9sPoWaFmiQf6q5EIYzG+WQpncj80
mxl3HWOP6oFj/hZJRYseKeMkvJzLTo87rFdM6CsMrLwETR6e+aWM0btPFil1rXVA
CNOjsy0bxTV8OJEfyxnYmyjvnBvB0kdiaVEDdVhxgSqzLAX4mgXa49/V6M/uzMr+
n3/A1Jdk4V6fVm8S5cFIXxoUat3cB4xGaT9OWD3o1NPr6eS9Vo0EsJlRl81SG68f
S+Qtk2fX27T68YG4Aa3zMfZxUsVuFLtTuQbRC+fJpIkCPAQYAQoAJgIbDBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJlhcuqBQkD8n1oAAoJENmtKhgFdHTLoO0QAJsT
E9fkleb7YzPEuP9GJ3jx8PGdWm7n+8UNdr24kS6gOXVUfPZrWa5So21hcIwZb4PZ
DqHSVSQnRciKhSnG7gplYPNGZ4+FWbLr/mBRYarjkVFLUuCPexSIjxV1KSGJnWs9
YTVAKZAz75GpCML6jD6biCOQCQ86wqOdWvZIZR8YvurrxR64ABB0rjbsaG8cNOUX
1cwAfdLwthf64dS+2m3lqNGDHkP5eNL0RIxC5gXYEp0lvmlMH3ZuO5WrfH73PTDg
89bxXeuhrFmSEwf4xWm603oi8/2qQvR9/7jb0o+t71NQuWrWIFONZWWgZBUGso+u
yT3XgY4YqKGR3z2QzKHYnJ6M7SvSYpqS7RtcxcCXF0HGNfES8cAgtKVpFtbtSwXX
p8O8oLyjmVIO/NjUpbLOGdFIsarsezLFV9f2fqZ63J34hyUSg8LrYVV1fA5DJUpe
bbX4hLpdk0MMtgG43BwKIGlJTpL5RkQ/uQU3YW2kairy7o+1imDD0TRzQxtdjVOI
5vnlTNcfJZIIfLx4drABA12OvpX3dfPV62R+8BAlJFT430CG6AISJIBqJRFvuikm
nZGUvEHmOUs/FLbbaXTPkKc7tR2WIwljRvMV+Qk84cWcX6YchMslMuiDM1mtlQZi
g34WHGSE+zCWnXAslIHlSwox7qfdO0Kz2XncSbIAiQI8BBgBCgAmAhsMFiEEC+My
dddMlTx5+BEH2a0qGAV0dMsFAmeEQS4FCQXT8gIACgkQ2a0qGAV0dMsm1Q//V09x
OutSbWU44KRurdnGKsk56DFlqXtjGYJqDPrODpX3M8IDf2MuTIN2yfPMv984bAbO
A9RL7EaGVlQUW9QWPURMsZKEFQljhfxRJO9JoGDYI7uRDnSEi6WjVvgUk5OiIh0K
EI4jEcaLCzveIEcswrVDSAn+7nGvewP8Rrx7qMUNvLAltxiMyfGXneavRs3sfusz
db9LTTY8lCU0xrslaXrrvfCkaRbskFi3S31I+1ZB/ewuAhHqfc13eRBjPwQOanJF
epAzP4GF41fVQN9GtssATCD+dV6OFhYjfWJbOIcPv277wCvGIFucM9XRjbkCIYFQ
E5W+1O/act3Obj1sB90C+cVOgCng37YqfObYLF19RE4+a7NUAh/GxHj+8TxUyvvr
aWWyfNqTMDjHMShNDjG0qSzFX7vfyUpmfAfz+6ad78aks9LMf+86iGkvBhXFs7cz
Vv4PWWYV1+WShNU2Y9yMaH3zpWUaREdB07HKbLva4Y1icqWVx+z6xs3PvsqbTei/
moXiZk7ohpbBm0htJlki22ARYrGXSK6w5RQtCZoBW0DEj5JNBjkK6XbAW3VFuAA1
J5wLS2z0eIR5adP+/SxQUbTq+ZFiOGdBP1g/783e7zEdyA0YfA2KU9OdLzupUix/
x+JYcxmrZXndSMObd0IiWOhwXlgarbJJMReOORg=
=cYaK
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.2. Secretário do Core Team <core-secretary@FreeBSD.org>

```
pub   rsa4096/4D632518C3546B05 2024-02-17 [SC] [expires: 2028-02-08]
      Key fingerprint = 1A23 6A92 528D 00DD 7965  76FE 4D63 2518 C354 6B05
uid                           FreeBSD Core Team Secretary <core-
secretary@FreeBSD.org>
sub   rsa4096/CABFDE12CA516ED2 2024-02-17 [E] [expires: 2028-02-08]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXQ1o8BEAC+Rcg8cmVxuP17Vu+q5KgCx/XiulQuqKXAqqBlYCH2jqk6DINP
yFrREGBhzd/qNmlAYEahQ4Zgl0bUZNTrZVDyzicOvPP0jH+KSTQwRs7NOawEdlVO
cyHrwDCPEqf5ZzD4NhfTriEOw+j0pEH/onitUGvoQRtx15xWyaJQxDEBMTYMLewE
86D1bltwnTNczE3UZb7oQLJXkAX5hcLtou70XJGgZITvJkK+kp/xot2eFjnqRz/u
WeXnKhYAmC07EKwZ1uw047eHKwMMRBYqzApLwoQtfE430Kxf2q8de64x8zDbi6YM
1J4r8OAxOtHVyfJ0j7Q23DEZz0VVb4b1Tx5OG2Re/KSNvqI0awJO4TcRmOR88OyY
dzyXgnX6Sa7GVQY1FXvn7vtFuDAt7egZOzeomSHL9bdX07LTQ4UtM88EV9wm3q4q
smoatV9jsvPQ1zxCU3aQD/5eWTJH2/kz1LIuBL/Qi5XQpJn91lBtUWJrCgkHWPGu
f//rnnXmsG7DAChW+yZ7cFO8lfNa8sFhPqSxCYphWmJTrvadyQtDngB8JakWdnmK
pfGS6y5lel+181vw38ZZKt04AKM+nDY8051lBM7Q9Q6kTLI33UZeImndx5xYukVD
kV6aQ31HYfEark15c7iEz+OAcwFnM2ntXMt7kKGd40CqzusiPcQkPqPbAQARAQAB
tDhGcmVlQlNEIENvcmUgVGVhbSBTZWNyZXRhcnkgPGNvcmUtc2VjcmV0YXJ5QEZy
ZWVCU0Qub3JnPokCVwQTAQoAQQIbAwgLCQ0IDAcLAwUVCgkICwUWAwIBAAIeBQIX
gBYhBBojapJSjQDdeWV2/k1jJRjDVGsFBQJnp8PiBQkHeofTAAoJEE1jJRjDVGsF
GMIQALhj+mNpH8OmTFeihQ6t9P8un3llz6Wmqe/Q+ULWeqJvV/uC1J5T9fnoGhwF
MgECuguXYJtoYQ16KXnsSOs1tcqMOK9GtEtFJTGe2DtfLiBednwUvu9j3HmTlwLN
M+7rqkiC0HCg2qSjcMjxbVbA5BSgNkgfyTSO2YdjfaZ+ceiHwo/qa5xWE6i2dMR1
PLGMMHTeELdtdSH6VR9/3h0qt3qzwdMBRCVAQHbim7CqwjUH9jg+IOySXl81jNoB
xuVZR3pKshyY4Oxo1dK+W86Uiff/+c4jCAxokGWIR7C4MkZZWUQqV7920gkZFC/5
qzpT1A1/sFUg8HfFTr0vCoPSVFWn2+Tto3vr78DICVaAf02aYAFlyKK18BMCoHkS
hDDO+/JQZmvHOIgEYK+T2WN1c9gm9IDJzGZuH0X2C/vkREnJKkccJfB4pXuN10wt
fqyP9fn8h6+/t+sv3qNLm4d8fkmLXofulG3WB4i/F+Hip2rjPvvBCF7zl4xy6cJ2
xVY5HUOBTqmIwVhYwUpXaqNoWa/qJBLTuot3z7ciKmKX6Lq+Dze5dzhrPNl/CalC
HBf3miHRK3TZbYLooG3bcEWgxU2BnBi3vl5NpCoUOKkPYRlALXi2TyHmPxO7oT4e
mIzS6BgnXOqO+AXvbKKfayuSePdBqkNMK/SMC4Dylkf6Xj25iQIzBBABCgAdFiEE
EBpxaxYrAOVb7eoFrbv4YQo3ibcFAmbu7x4ACgkQrbv4YQo3ibcr5A//TIcbD/EW
YJz0zrUQdc9xG3UNfU6uHmQzAuUy5ginevyqv0TSso5qvSkOHvdxbi41rfMiB2RJ
V3q0n0PSvHFld89fOTZUMZXaPvozCiBYWScrt+KA/2pq3K8mUumHS+IFtpHLL2Tu
+gI8XCHUFxO9HUThM/rFlIyEdzoctgmqQ7IG4uZG+o2J3w09llhDAUe0vraJK1On
p9yFACnRrhqvl41JeUWcv9MH9JcwHUqtUo9WLTCIb+hkByTOyRfHBYpYw//bdXdf
6rkCwKVWpyMDbk61zq2VsS1kqbP9IH/A8CsBnA6mg+zPq2i7HIFw8Swj4OGJcIvG
a9ubUYJRjDhX/vBpNrtncANZ88FQmA+Maq0vu0LS5IIGyIKkvd1fKIsyvBDDa9kE
nfCW0XMkJA0Gf+kxdB+elXQHBwKO2sr5BiKnJT5lJq3Yu9fxxGBnf93yiN4E9bmF
gG7cZxpyb1Bp76TJhLcANyybOTjCtiNRrgqeaSx9/6hSPfPigGXIne0H2lmJO6oq
jUrsYmFiWvU9sc7AcPVw/eHG8FgW35TuwKX71z8w994iaahUPNcSVyXOUD+QNROv
HhGUXrc81t613rivh22N0NZpNubVatq43KV7+/bnPyWBIi1Awh3vIFsNNSQSrYxF
lUuQaAHQXTeZMZ/7npE4t86seMt0T7BGn765Ag0EZdDWjwEQAL3VwFifpnRCYzQ4
VZ1dAjp8w542iRprqeA+C3tvNbk2OvKpN3DIc49l2bgNZ/VI7/T58lEKrFsgLK4w
AWtv18OV7xuh0AuOblmq5MvcPrUelkPj5HA7M6Ng/rAubuHfwdP/IjIrzzY6+XDh
```

```
fP9N5KIv9VRJYT23BbvLPeit4J4tQEB/NpxL3L6zI75qq4R3T/EkHP6Y9Buup9Lr
isJlcxkSK+CyORCgTpEz6fWsXiTDgS7cTaQ969XCygBpj4wRQzwkBdokxo1wsiFt
4zLt07/PrPYjeHltTDkrF9XDNhLNJ5GliHnd01oHg1j5/n+9Osvh1maoFwuBxXTN
nTZ2P+7RKLBaVQVSkUa5KJbXoM3v7bMbXM5aLs2XjglrAzrZOV+Y7zOu5UYQdpXd
7x8XAEtEozRJzt7dosQIhKx9h4LlltFFulDUpf5VCvcUEoAzMbiX0aju9n5Rwsql
DHatU9Mm3W8OdFXj1foIXZD+VX2Jp9DgxPLoAJ0CWhPXJ8f+WFSJZCjWoPJEJKWY
0EOxiXyAuvAyniAZAC/eKZkfGckXmu7edRgYbRTTwPmZ/axa/k9aHsHLwmbxuZo/
xxQXzqU70EleHZ31A3mOqsC1epjN6dn4AFKgvVesP/K/fbvSsfsfiABS2A/68ne4
zJG73Tnk4L6J79vrXc2iMJbmdg3ZABEBAAGJAjwEGAEKACYCGwwWIQQaI2qSUo0A
3Xlldv5NYyUYw1RrBQUCZ6fEFQUJB3qIBgAKCRBNYyUYw1RrBQd9EACMVckxzy4w
aUGlERguJ+kslS8MkMjNqnfDPLRDVxbUxNvbbhw7/u9blM65DCGCeNLc2n4oiu5C
E3I095AKmvq/0dOa81mEEdZkC1CVc64bXWbEyz5AtSHUpgdxRso6C+YopndSiz1T
WcIagQRXfWaw3FBWPooA87gmibmSmCegCtqx+uyc5QxX2eFI8mRK7vlfnGpYKHs0
D1/yUSGQOwoNRJ5FYm+ynfDE3FzHEQL7lv1vpv1k3xNKfBziMMg4IMEBKNHV4VKN
qJpa8UCodeTGSWQdNnCeCWPsxz5oQjCcVH9Z3e8sMpWLHhRcBZzSwXUOws2GbRMH
xHwrfrpHJcrBhe64pgfGOvZLUJ9BDs+8egTHsqRFacipbtTR+hhVhuJEHDaQQWuM
8IRHj9HIuTAczET8JTDHTMIoo8DZdOtiW/YgqCDwYghkI77d12oNQqYoeJ2HiqbK
cGzwCpsR0A+p/iOAxJG13tsxqZV8TQ8iTokWG6ACtZ7sfeWEhxqMbUKUMgogZn0I
3n1kV+tUZC6BQrIYI7TiKg95wLZsIydeoIsQoNZwvyKAXfVmQ62YjIX8njZwN+07
8/ipUPJxCYa8zL8BZyDmoFJqa3y9z+11+vtiZ9t+aTwGvjpHDwyeCJco7go9cU3m
GRFZYciqIoG4n3tl8Pob15vFlVqk47rRqg==
=8TzT
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.3. Secretário da equipe de gerenciamento do Ports <portmgr-secretary@FreeBSD.org>

```
pub   ed25519/E3C401F60D709D59 2023-03-06 [SC] [expires: 2027-03-05]
      Key fingerprint = BED4 A1D3 6555 B681 2E9F  ABDA E3C4 01F6 0D70 9D59
uid                           FreeBSD Ports Management Team Secretary <portmgr-
secretary@FreeBSD.org>
sub   cv25519/2C92B55E27A641C3 2023-03-06 [E] [expires: 2027-03-05]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEZAXJvxYJKwYBBAHaRw8BAQdASFAC20WL3R1T6uNyGMZbfJCxDkcP4C5vi3Op
tcZ2fbq0R0ZyZWVCU0QgUG9ydHMgTWFuYWdlbWVudCBUZWFtIFNlY3JldGFyeSA8
cG9ydG1nci1zZWNyZXRhcnlARnJlZUJTRC5vcmc+iJYEExYKAD4WIQS+1KHTZVW2
gS6fq9rjxAH2DXCdWQUCZAXJvwIbAwUJB4TOAAULCQgHAwUVCgkICwUWAwIBAAIe
BQIXgAAKCRDjxAH2DXCdWYN1AP43TjyfZtZ3DLYT++g0+SuPsoO/3yWVybA+UmFL
zb8MngEA+LLNUfvEwCuXS/soh+ww5bpfmi3UUmeGiQEAXug3iA+JATMEEAEKAB0W
IQT7N0XIbxXo7ayBMvzYKU7Du8TX1QUCZAXLkwAKCRDYKU7Du8TX1XHMB/9R1MX4
6zMgpKqPPt76GOI+eGEdBK6bY8aJZjQGdqTh9f6VtXVoTGIG7cvhc9X8tDBoB0PT
2KZWheF51AV1+NHU4HwLAQ1BMebrFvWSfkw4xg4fBGwDhz9/GN85No+Js772V5ey
8lRiL6meRVWxMlLyWcxGd8JjcC5yX/iAUQ3SBGCLqW7unWjjg7CTd+AMBwcqPGrv
ax8q6eFVguJcHJAjMnKf6HAy4cpK3s+uMoUBCGnszSN12B3ysKfyC4pNO/pix5tA
Q5v8aRqTeFPh5zmNhWo0KGPzplTPqRQSHDl7GDQC8Ru3MhzFkeWzHsexjZVwS6W2
DPcYpuuAsA0XOZIZiQIzBBABCgAdFiEEEBpxaxYrAOVb7eoFrbv4YQo3ibcFAmQF
```

```
0u0ACgkQrbv4YQo3ibccwg/9F2Xuic3nhKxRbB3mJeDo6SYQETa/Gh1qQ34+8zlt
8UMazOx67gnYQfy+pXjro6eQ2up0a4eUYezcNOudqAQD21nRz3HA6EQVNcE/TzEA
xl5CJntTaLOt7S+EDXFW5BuQIvhhoMGgm8+WNVgA0EJ7tfL0OcYBSvr19fqwChEn
9c14cSk6mgHSsleP5NvskYN053pxHwy0LTSb8YBBv52th37t/CRFC1363rS5q+D7
JixFopd1O5pKpA5ipvE4gGgRjPtwjx0SjjepwK/3fuhEJQQyKzTIKlMfu2Dj/iR2
Li1Sfccau5LQXOj9fUITU3u1YG7yrm8VGzT7ao4d+KRwgMLjd2pLqiGIbbJwGBiP
FRmtilWQoeIlmSlFX4obAA517DOK0pW1mH8+eEn4EJd3SekT3yzFyKTASv0J48Z8
3F928xg+eZvHxVC0t1J+J5IG0gt3EEncuWKIPQGR7PiQbti6R3FQVTz6WfMWOebP
Qi0E9F/Aqakr6Vj2sKGrDq+ebpaF5G8Yw1YrUl2IDiPzkCegp3ZbI0wh11Xvzhi8
LXPQGK4jBQas4G8cegfitzmtdGRHYrbMv0R9I4mvaL+WlOuD2AvyVG28lguqVhnN
AZP+ohdquYyX2CNCVvbKWAtXo6Ur0vWG8BL8m6defAtEkIwVBALaOHQOSI3aNUz4
lwy4OARkBcm/EgorBgEEAZdVAQUBAQdAsefmSfxEOdOr02+K/6noYCuJ1FeAWVz6
jFYQ+9w6jggDAQgHiH4EGBYKACYWIQS+1KHTZVW2gS6fq9rjxAH2DXCdWQUCZAXJ
vwIbDAUJB4TOAAAKCRDjxAH2DXCdWRl4AP9h5ot212BK29S6ZcMBhHvmtF5PG1oD
c7LnZycSRmbFiwEAndCMpAGOhDW8iVgDd0wLQq/ZMPe+xccfG1b3zFH2EgE=
=iiAT
-----END PGP PUBLIC KEY BLOCK-----
```

D.1.4. <doceng-secretary@FreeBSD.org>

```
pub   rsa2048/E1C03580AEB45E58 2019-10-31 [SC] [expires: 2022-10-30]
      Key fingerprint = F24D 7B32 B864 625E 5541  A0E4 E1C0 3580 AEB4 5E58
uid                           FreeBSD Doceng Team Secretary <doceng-
secretary@freebsd.org>
sub   rsa2048/9EA8D713509472FC 2019-10-31 [E] [expires: 2022-10-30]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF27FFcBCADeoSsIgyQUY8vREwkTikwFFlNg31MVy5s/Nq1cNK1PRfRMnprS
yfB62KqbYuz16bmQKaA9zHN4FGfiTvR6tl66LVHm1s/5HPiLv8sP14GsruLro9zN
v72dO7a9i68bMw+jarPOnu9dGiDFEI0dACOkdCGEYKEUapQeNpmWRrQ46BeXyFwF
JcNx76bJJUkwk6fWC0W63D762e6lCEX6ndoaPjjLBnFvtx13heNGUc8RukBwe2mA
U5pSGHj47J05bdWiRSwZaXa8PcW+20zTWaP755w7zWe4h60GANY7OsT9nuOqsioJ
QonxTrJuZweKRV8fNQ1EfDws3HZr7/7iXvO3ABEBAAG0PEZyZWVCU0QgRG9jZW5n
IFRlYW0gU2VjcmV0YXJ5IDxkb2Nlbmctc2VjcmV0YXJ5QGZyZWVic2Qub3JnPokB
VAQTAQoAPhYhBPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsDBQkFo5qABQsJ
CAcDBRUKCQgLBRYDAgEAAh4BAheAAAoJEOHANYCutF5YB2IIALw+EPYmOz9qlqIn
oTFmk/5MrcdzC5iLEfxubbF6TopDWsWPiOh5mAuvfEmROSGf6ctvdYe9UtQV3VNY
KeeyskeFrIBOFo2KG/dFqKPAWef6IfhbW3HWDWo5uOBg01jHzQ/pB1n6SMKiXfsM
idL9wN+UQKxF3Y7S/bVrZTV0isRUolO9+8kQeSYT/NMojVM0H2fWrTP/TaNEW4fY
JBDAl5hsktzdl8sdbNqdC0GiX3xb4GvgVzGGQELagsxjfuXk6PfOyn6Wx2d+yRcI
FrKojmhihBp5VGFQkntBIXQkaW0xhW+WBGxwXdaAl0drQlZ3W+edgdOl705x73kf
Uw3Fh2a5AQ0EXbsUVwEIANEPAsltM4vFj2pi5xEuHEcZIrIX/ZJhoaBtZkqvkB+H
4pu3/eQHK5hg0Dw12ugffPMz8mi57iGNI9TXd8ZYMJxAdvEZSDHCKZTX9G+FcxWa
/AzKNiG25uSISzz7rMB/lV1gofCdGtpHFRFTiNxFcoacugTdlYDiscgJZMJSg/hC
GXBdEKXR5WRAgAGandcL8llCToOt1lZEOkd5vJM861w6evgDhAZ2HGhRuG8/NDxG
r4UtlnYGUCFof/Q4oPNbDJzmZXF+8OQyTNcEpVD3leEOWG1Uv5XWS2XKVHcHZZ++
ISo/B5Q6Oi3SJFCVV9f+g09YF+PgfP/mVMBgif2fT20AEQEAAYkBPAQYAQoAJhYh
```

```
BPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsMBQkFo5qAAAoJEOHANYCutF5Y
kecIAMTh2VHQqjXHTszQMsy3NjiTVVITI3z+pzY0u2EYmLytXQ2pZMzLHMcklmub
5po0X4EvL6bZiJcLMI2mSrOs0Gp8P3hyMI40IkqoLMp7VA2LFlPgIJ7K5W4oVwf8
khY6lw7qg2l69APm/MM3xAyiL4p6MU8tpvWg5AncZ6lxyy27rxVflzEtCrKQuG/a
oVaOlMjH3uxvOK6IIxlhvWD0nKs/e2h2HIAZ+ILE6ytS5ZEg2GXuigoQZdEnv71L
xyvE9JANwGZLkDxnS5pgN2ikfkQYlFpJEkrNTQleCOHIIIp8vgJngEaP51xOIbQM
CiG/y3cmKQ/ZfH7BBvlZVtZKQsI=
=MQKT
-----END PGP PUBLIC KEY BLOCK-----
```