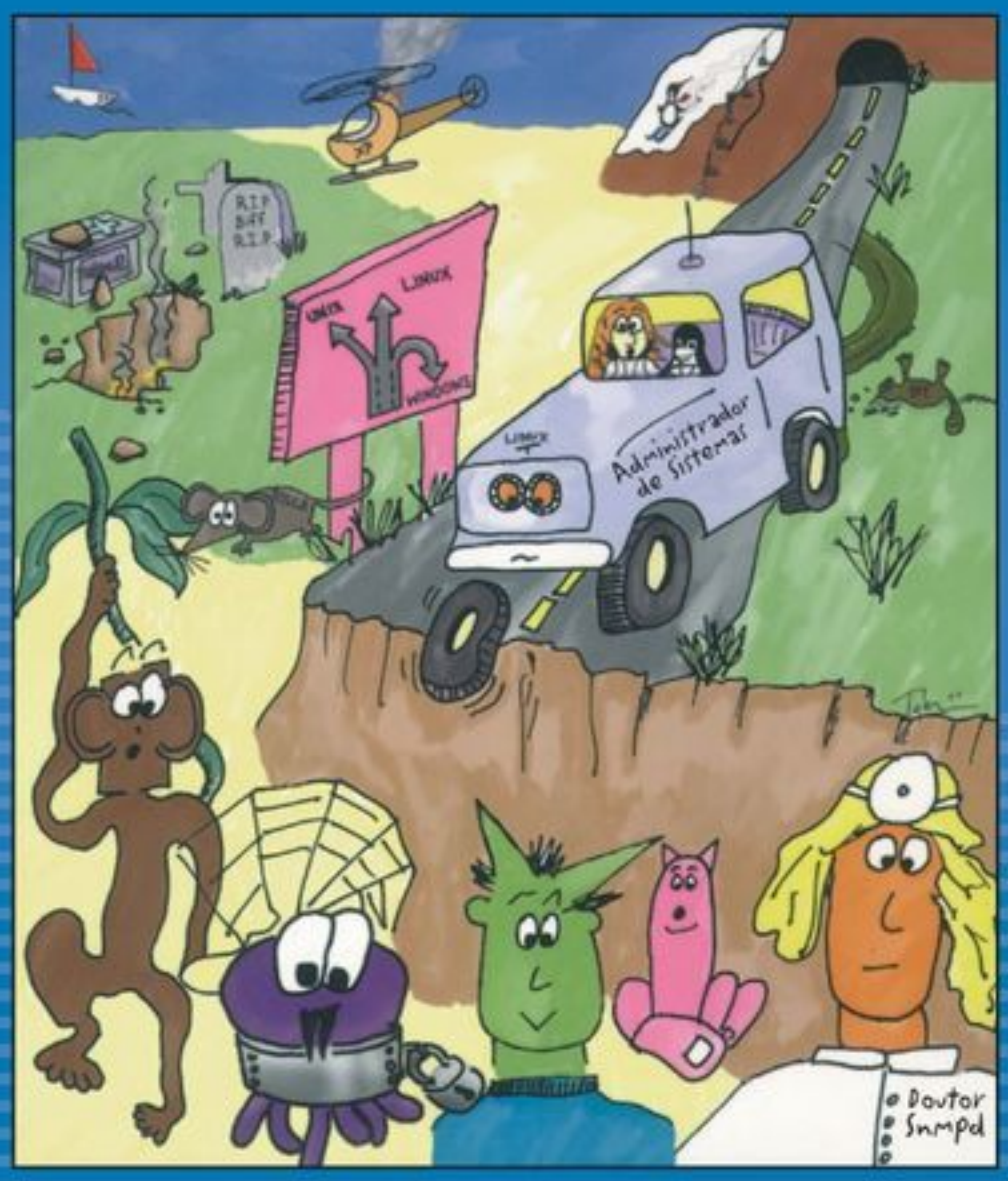
MANUAL COMPLETO DO LINUX

Guia do administrador

SEGUNDA EDIÇÃO



EVI NEMETH / GARTH SNYDER / TRENT R. HEIN
Com Lynda McGinley, Ben Whaley, Adam Boggs, Jeffrey S. Haemer, Tobi Oetiker, Fritz Zaucker, Scott Seidel, Bryan Buus, Ned McClain e David Schweikert

PEARSON
Prentice Hall

MANUAL COMPLETO DO LINUX

Guia do administrador

SEGUNDA EDIÇÃO

MANUAL COMPLETO DO LINUX

Guia do administrador

SEGUNDA EDIÇÃO

EVI NEMETH / GARTH SNYDER / TRENT R. HEIN

com Lynda McGinley, Ben Whaley, Adam Boggs,
Jeffrey S. Haemer, Tobi Oetiker, Fritz Zaucker, Scott Seidel,
Bryan Buus, Ned McClain e David Schweikert

Tradução

Carlos Schafranski e Edson Furmankiewicz
Docware Traduções Técnicas

Revisão Técnica

Nivaldo Foresti
Consultor técnico da Information Week Brasil

PEARSON
Prentice Hall

São Paulo
Brasil • Argentina • Colômbia • Costa Rica • Chile • Espanha
Guatemala • México • Peru • Porto Rico • Venezuela

© 2007 by Pearson Education do Brasil
© 2007 by Prentice Hall, uma empresa do grupo Pearson Education, Inc.
Tradução autorizada a partir da edição original em inglês, *Linux Administration Handbook 2nd Edition*, de Evi Nemeth, Garth Snyder e Trent Hein, publicada pela Pearson Education, Inc, sob o selo Prentice Hall PTR

Todos os direitos reservados. Nenhuma parte desta publicação pode ser reproduzida ou transmitida de qualquer modo ou por qualquer meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Pearson Education do Brasil.

Muitas das designações utilizadas por fabricantes e vendedores para distinguir seus produtos são protegidas como marcas comerciais. Onde essas aparecem no livro, e a Manning Publications estava ciente de uma proteção de marca comercial, as designações foram impressas com a primeira letra ou todas as letras maiúsculas.

Os nomes de empresas e produtos mencionados neste livro são marcas comerciais ou registradas de seus respectivos proprietários.

Gerente editorial: Roger Trimer
Editora sênior: Sabrina Cairo
Editor de desenvolvimento: Marco Pace
Editora de texto: Eugênia Pessotti
Revisão de texto: Lucrécia Freitas
Capa: Vitor Nadu (sobre original de Leslie Hamies)
Editoração Eletrônica: Docware Traduções Técnicas

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Nemeth, Evi

Manual Completo do Linux / Evi Nemeth; Garth Snyder; Trent R. Hein ; com Adam Boggs, Matt Crosby e Ned McClain ; tradução Edson Furmankiewicz , Carlos Schafranski; revisão técnica Mario Olimpio de Menezes. -- São Paulo : Pearson Prentice Hall, 2007

Título original: Linux Administration Handbook.
ISBN 978-85-7605-112-1

1. Linux (Sistema operacional de computador) 2. Redes de computadores I. Snyder, Garth. II . Hein, Trent R. III. Título..

04-1903 CDD-005.432

Índices para catálogo sistemático
1. Linux : sistema operacional específico: Computadores : Processamento de dados 005.432

2007
Direitos exclusivos para a língua portuguesa cedidos à
Pearson Education do Brasil,
uma empresa do grupo Pearson Education
Av. Ermano Marchetti,1435
CEP: 05038-001 – São Paulo – SP
Tel.: (11) 2178-8686 – Fax: (11) 3611-0444

Sumário

PREFÁCIO XIII
PREFÁCIO À PRIMEIRA EDIÇÃO XV
AGRADECIMENTOS XVI
SOBRE OS AUTORES XVII
SOBRE OS COLABORADORES XVIII

PARTE 1 ADMINISTRAÇÃO BÁSICA

Capítulo 1 POR ONDE COMEÇAR 3

1.1 Background sugerido 3
1.2 O relacionamento do Linux com o Unix 4
1.3 Linux em contexto histórico 4
1.4 Distribuições do Linux 5
1.5 Convenções tipográficas e de notação 7
1.6 Onde buscar informações 8
1.7 Como localizar e instalar software 10
1.8 Tarefas essenciais de um administrador de sistema 11
1.9 Administração de sistema sob pressão 13
1.10 Leitura recomendada 13
1.11 Exercícios 14

Capítulo 2 INICIALIZAÇÃO E DESLIGAMENTO 15

2.1 Inicialização 15
2.2 Inicializando PCs 18
2.3 Utilizando rotinas de inicialização: LILO e GRUB 19
2.4 Inicializando no modo monousuário 22
2.5 Trabalhando com scripts de inicialização 23
2.6 Reinicializando e desligando 28
2.7 Exercícios 31

Capítulo 3 PODERES DA CONTA ROOT 32

3.1 Posse de arquivos e processos 32
3.2 O superusuário 33
3.3 Escolhendo uma senha para root 34
3.4 Tornando-se o root 34
3.5 Outros pseudo-usuários 37
3.6 Exercícios 37

Capítulo 4 CONTROLANDO PROCESSOS 38

4.1 Componentes de um processo 38
4.2 O ciclo de vida de um processo 40
4.3 Sinais 40
4.4 Kill e killall: enviando sinais 42
4.5 Estados de processos 43
4.6 Nice e renice: influência sobre a prioridade de programação 43

4.7 ps: monitorando processos ... 44
4.8 top: monitorando ainda melhor os processos ... 46
4.9 O sistema de arquivos /proc ... 47
4.10 strace: rastreando sinais e chamadas de sistema ... 47
4.11 Processos descontrolados ... 48
4.12 Leitura recomendada ... 49
4.13 Exercícios ... 49

Capítulo 5 O SISTEMA DE ARQUIVOS ... 50

5.1 Nomes de caminho ... 51
5.2 Montagem e desmontagem de sistema de arquivos ... 52
5.3 A organização da árvore de arquivos ... 53
5.4 Tipos de arquivos ... 54
5.5 Atributos de arquivo ... 57
5.6 Listas de controle de acesso ... 62
5.7 Exercícios ... 65

Capítulo 6 ADICIONANDO NOVOS USUÁRIOS ... 66

6.1 O arquivo /etc/passwd ... 66
6.2 O arquivo /etc/shadow ... 70
6.3 O arquivo /etc/group ... 71
6.4 Adicionando usuários ... 72
6.5 Eliminando usuários ... 76
6.6 Desativando logins ... 76
6.7 Gerenciando contas ... 76
6.8 Exercícios ... 78

Capítulo 7 ADICIONANDO UM DISCO ... 79

7.1 Interfaces de disco ... 79
7.2 Geometria do disco ... 84
7.3 Sistemas de arquivos do Linux ... 85
7.4 Uma visão geral sobre o procedimento de instalação de discos ... 86
7.5 hdparm: configurando parâmetros da interface IDE ... 91
7.6 fsck: verificando e reparando sistemas de arquivos ... 92
7.7 Adicionando um disco: um guia passo a passo ... 94
7.8 Gerenciamento avançado de disco: RAID e LVM ... 97
7.9 Montando unidades USB ... 104
7.10 Exercícios ... 105

Capítulo 8 PROCESSOS PERIÓDICOS ... 106

8.1 cron: agendando comandos ... 106
8.2 O formato de arquivos crontab ... 107
8.3 Gerenciamento de crontab ... 108
8.4 Alguns usos comuns para cron ... 108
8.5 Outros agendadores: anacron e fcron ... 110
8.6 Exercícios ... 111

Capítulo 9 BACKUPS ... 112

9.1 Boas práticas ... 112
9.2 Dispositivos e mídia para backup ... 115
9.3 Configurando um regime de backup incremental via dump ... 119
9.4 Restaurando dumps com restore ... 122
9.5 Realizando dump e restauração para atualizações ... 124
9.6 Utilizando outros programas de arquivamento ... 124
9.7 Usando vários arquivos em uma única fita ... 126
9.8 Bacula ... 126

9.9 Produtos comerciais para backup ... 139
9.10 Leitura recomendada ... 140
9.11 Exercícios ... 141

Capítulo 10 SYSLOG E ARQUIVOS DE LOG ... 143

10.1 Diretivas de registro em log ... 143
10.2 Arquivos de log no Linux ... 145
10.3 logrotate: gerenciando arquivos de log ... 148
10.4 Syslog: o registrador de eventos do sistema ... 149
10.5 Condensando arquivos de log em informações úteis ... 157
10.6 Exercícios ... 158

Capítulo 11 GERENCIAMENTO DE SOFTWARE E CONFIGURAÇÃO ... 159

11.1 Instalação básica do Linux ... 159
11.2 Clientes sem disco ... 165
11.3 Gerenciamento de pacotes ... 166
11.4 Sistemas de gerenciamento de pacotes de alto nível ... 168
11.5 Controle de versão ... 174
11.6 Localização e configuração ... 180
11.7 Ferramentas de gerenciamento de configuração ... 183
11.8 Compartilhando software pelo NFS ... 185
11.9 Software recomendado ... 187
11.10 Leitura recomendada ... 188
11.11 Exercícios ... 188

PARTE 2 REDES

Capítulo 12 REDES TCP/IP ... 191

12.1 O TCP/IP e a Internet ... 191
12.2 Guia de redes ... 193
12.3 Pacotes e encapsulamento ... 194
12.4 Endereços IP: os mínimos detalhes ... 198
12.5 Roteamento ... 205
12.6 ARP: o protocolo de conversão de endereço ... 207
12.7 Adição de uma máquina a uma rede ... 208
12.8 Configuração de rede específica do fornecedor ... 215
12.9 DHCP: o Dynamic Host Configuration Protocol ... 217
12.10 Reconfiguração dinâmica e ajuste ... 219
12.11 Questões de segurança ... 221
12.12 Linux NAT ... 223
12.13 PPP: o Point-to-Point Protocol ... 223
12.14 Idiossincrasias de rede do Linux ... 230
12.15 Leitura recomendada ... 231
12.16 Exercícios ... 232

Capítulo 13 ROTEAMENTO ... 233

13.1 Encaminhamento de pacotes: uma visão mais detalhada ... 233
13.2 Daemons de roteamento e protocolos de roteamento ... 235
13.3 Visão geral dos protocolos ... 237
13.4 routed: criando suas próprias rotas ... 239
13.5 gated: o fim de um bom software livre ... 239
13.6 Critérios para seleção de uma estratégia de roteamento ... 240
13.7 Roteadores Cisco ... 241
13.8 Leitura recomendada ... 242
13.9 Exercícios ... 243

Capítulo 14 HARDWARE DE REDE 244

14.1 LAN, WAN ou MAN? 244
14.2 Ethernet: a LAN comum 244
14.3 Redes sem fio: a LAN do nômade 250
14.4 FDDI: uma LAN decepcionante, cara e ultrapassada 251
14.5 ATM: a LAN prometida (mas fracassada) 252
14.6 Frame relay: a WAN sacrificial 253
14.7 ISDN: a WAN nativa 253
14.8 DSL e modems a cabo: a WAN pública 254
14.9 Para onde as redes estão caminhando? 254
14.10 Teste e depuração de redes 254
14.11 Cabeamento 255
14.12 Questões de projeto de rede 256
14.13 Questões de gerenciamento 258
14.14 Fornecedores recomendados 258
14.15 Leitura recomendada 259
14.16 Exercícios 259

Capítulo 15 DNS: O DOMAIN NAME SYSTEM 260

15.1 DNS para o impaciente: acrescentando uma nova máquina 260
15.2 A história do DNS 261
15.3 Quem precisa do DNS? 263
15.4 O espaço de nomes DNS 263
15.5 Funcionamento do DNS 267
15.6 Novidades no DNS 269
15.7 O banco de dados DNS 270
15.8 O software BIND 285
15.9 Projetando seu ambiente de DNS 288
15.10 Questões relacionadas a clientes no BIND 291
15.11 Configuração de servidores BIND 292
15.12 Exemplos de configuração BIND 305
15.13 Iniciando named 311
15.14 Atualizando arquivos de zonas 311
15.15 Questões de segurança 314
15.16 Testando e depurando 324
15.17 Especificidades das distribuições 333
15.18 Leitura recomendada 335
15.19 Exercícios 336

Capítulo 16 O SISTEMA DE ARQUIVOS DE REDE 337

16.1 Informações gerais sobre o NFS 337
16.2 NFS no lado servidor 340
16.3 NFS no lado cliente 342
16.4 nfsstat: fazendo dump de estatísticas do NFS 345
16.5 Servidores de arquivos NFS dedicados 345
16.6 Montagem automática 346
16.7 Leitura recomendada 348
16.8 Exercícios 348

Capítulo 17 COMPARTILHANDO ARQUIVOS DE SISTEMA 349

17.1 O que compartilhar 349
17.2 nscd: armazenando em cache os resultados das pesquisas 350
17.3 Copiando arquivos 351
17.4 NIS: o Network Information Service 355
17.5 LDAP: o Lightweight Directory Access Protocol 361
17.6 Leitura recomendada 365
17.7 Exercícios 365

Capítulo 18 E-MAIL 366

18.1 Sistemas de e-mail 368
18.2 A anatomia de uma mensagem de e-mail 370
18.3 Filosofia do e-mail 374
18.4 Aliases de e-mail 377
18.5 Listas de discussão e software relacionado 382
18.6 sendmail: o diretor de espetáculos do circo do e-mail 386
18.7 Configuração de sendmail 391
18.8 Elementos básicos da configuração de sendmail 394
18.9 Elementos mais sofisticados de configuração de sendmail 397
18.10 Recursos relacionados com spam no sendmail 407
18.11 Estudo de caso de arquivo de configuração 414
18.12 Segurança e sendmail 417
18.13 O desempenho do sendmail 422
18.14 Estatísticas, testes e depuração de sendmail 425
18.15 O sistema de e-mail Exim 429
18.16 Postfix 431
18.17 Leitura recomendada 442
18.18 Exercícios 443

Capítulo 19 GERENCIAMENTO E DEPURAÇÃO DE REDES 445

19.1 Solucionando problemas de rede 445
19.2 ping: verificando se um host está funcionando 446
19.3 traceroute: rastreando pacotes IP 447
19.4 netstat: obtendo estatísticas de rede 449
19.5 sar: inspecionando as atividades de interface "ao vivo" 452
19.6 Analisadores de pacotes 453
19.7 Protocolos de gerenciamento de redes 455
19.8 SNMP: o Simple Network Management Protocol 455
19.9 O agente NET-SMNP 457
19.10 Aplicativos para gerenciamento de redes 458
19.11 Leitura recomendada 461
19.12 Exercícios 462

Capítulo 20 SEGURANÇA 463

20.1 O Linux é seguro? 463
20.2 Como a segurança é comprometida 464
20.3 Certificações e padrões 466
20.4 Dicas e filosofia de segurança 467
20.5 Problemas de segurança em /etc/passwd e /etc/shadow 469
20.6 Capacidades do POSIX 472
20.7 Programas setuid 472
20.8 Permissões de arquivos importantes 472
20.9 Problemas de segurança diversos 473
20.10 Ferramentas de segurança poderosas 475
20.11 Ferramentas de segurança criptográficas 480
20.12 Firewalls 484
20.13 Recursos de firewall do Linux: tabelas de IP 486
20.14 Redes privadas virtuais (VPNs) 489
20.15 Distribuições Linux aprimoradas 490
20.16 O que fazer quando sua instalação tiver sido atacada 490
20.17 Fontes de informação sobre segurança 491
20.18 Leitura recomendada 493
20.19 Exercícios 494

Capítulo 21 HOSPEDAGEM WEB E SERVIDORES INTERNET 496

21.1 Fundamentos sobre hospedagem Web 496
21.2 Instalação de servidores HTTP 499
21.3 Interfaces virtuais 501
21.4 A Secure Sockets Layer (SSL) 503
21.5 O servidor proxy e de cache Squid 505
21.6 Configuração de servidor FTP anônimo 506
21.7 Exercícios 507

PARTE 3 DIVERSOS

Capítulo 22 O X WINDOW SYSTEM 511

22.1 O gerenciador de vídeo X 512
22.2 Executando um aplicativo X 513
22.3 Configuração do servidor X 516
22.4 Solução de problemas e depuração 519
22.5 Uma breve nota sobre os ambientes de área de trabalho 521
22.6 Leitura recomendada 523
22.7 Exercícios 523

Capítulo 23 IMPRESSÃO 524

23.1 Impressoras são complicadas 524
23.2 Linguagens de impressora 525
23.3 Arquitetura CUPS 527
23.4 Administração do servidor CUPS 531
23.5 Dicas para solução de problemas 537
23.6 Recursos avançados de impressora 538
23.7 Outros conselhos sobre impressoras 539
23.8 Imprimindo no KDE 542
23.9 Leitura recomendada 543
23.10 Exercícios 543

Capítulo 24 MANUTENÇÃO E AMBIENTE 544

24.1 Princípios básicos da manutenção de hardware 544
24.2 Contratos de manutenção 544
24.3 Experiência com o tratamento de equipamentos eletrônicos 545
24.4 Monitores 546
24.5 Módulos de memória 546
24.6 Manutenção preventiva 546
24.7 Ambiente 547
24.8 Energia elétrica 549
24.9 Racks 549
24.10 Padrões dos centros de dados 550
24.11 Ferramentas 550
24.12 Leitura recomendada 551
24.13 Exercícios 551

Capítulo 25 ANÁLISE DE DESEMPENHO 552

25.1 O que você pode fazer para melhorar o desempenho 553
25.2 Fatores que afetam o desempenho 553
25.3 Verificação de desempenho do sistema 554
25.4 Socorro! Meu sistema está muito lento! 561
25.5 Leitura recomendada 562
25.6 Exercícios 562

Capítulo 26 COOPERANDO COM O WINDOWS 564

26.1 Conectando-se ao sistema Linux a partir do Windows 564
26.2 Acessando desktops remotos 565
26.3 Executando o Windows e aplicativos do tipo Windows 566
26.4 Utilizando ferramentas de linha de comando com o Windows 567
26.5 Conformidade do Windows com padrões de e-mail e da Web 568
26.6 Compartilhando arquivos com Samba e CIFS 568
26.7 Compartilhando impressoras com o Samba 574
26.8 Depurando o Samba 576
26.9 Leitura recomendada 578
26.10 Exercícios 578

Capítulo 27 DISPOSITIVOS SERIAIS 579

27.1 O padrão RS-232C 579
27.2 Conectores alternativos 581
27.3 Portadora via hardware e software 585
27.4 Controle de fluxo via hardware 585
27.5 Comprimento do cabo 586
27.6 Arquivos de dispositivos seriais 586
27.7 setserial: configurando parâmetros de porta serial 586
27.8 Configuração de software para dispositivos seriais 587
27.9 Configuração de terminais cabeados 587
27.10 Caracteres especiais e o driver de terminal 590
27.11 stty: opções de configuração de terminal 591
27.12 tset: configurando opções automaticamente 591
27.13 Destravando terminais 592
27.14 Modems 592
27.15 Depurando uma linha serial 594
27.16 Outras portas de E/S comuns 594
27.17 Exercícios 595

Capítulo 28 DRIVERS E O KERNEL 596

28.1 Adaptação do kernel 596
28.2 Drivers e arquivos de dispositivo 597
28.3 Por que e como configurar o kernel? 599
28.4 Ajustando parâmetros do kernel Linux 600
28.5 Construindo um kernel Linux 601
28.6 Adicionando um driver de dispositivo Linux 603
28.7 Módulos de kernel carregáveis 604
28.8 Conexão automática 605
28.9 Configurando as opções de inicialização 606
28.10 Leitura recomendada 606
28.11 Exercícios 607

Capítulo 29 DAEMONS 608

29.1 init: o processo primordial 608
29.2 cron e atd: Agendando comandos 609
29.3 xinetd e inetd: daemons de gerenciamento 609
29.4 Daemons de kernel 613
29.5 Daemons de impressão 614
29.6 Daemons de serviços de arquivos 614
29.7 Daemons de banco de dados administrativos 615
29.8 daemons de e-mails 616
29.9 Daemons de execução de login e comando remoto 616
29.10 Inicialização e configuração de daemons 617
29.11 Outros daemons de rede 618
29.12 ntpd: daemon de sincronização de tempo 619
29.13 Exercícios 620

Capítulo 30 GERENCIAMENTO, DIRETIVAS E POLÍTICAS 621

30.1 Faça todo mundo feliz 621
30.2 Componentes de uma empresa de TI funcional 622
30.3 O papel do gerenciamento 622
30.4 O papel da administração 628
30.5 O papel do desenvolvimento 630
30.6 O papel das operações 634
30.7 O trabalho do suporte 635
30.8 Documentação 638
30.9 Sistemas de rastreamento de solicitações e de geração de relatórios de problemas 640
30.10 Recuperação após desastre 643
30.11 Diretiva escrita 646
30.12 Questões jurídicas 650
30.13 Patentes de softwares 655
30.14 Padrões 656
30.15 A cultura Linux 658
30.16 A onda Linux 659
30.17 Organizações, conferências e outros recursos 660
30.18 Leitura recomendada 663
30.19 Exercícios 664

ÍNDICE 667

Prefácio

Quando escrevíamos a primeira edição deste livro (há aproximadamente cinco anos), o Linux estava apenas começando a se revelar para o mundo corporativo. Esperamos que o *Manual completo do Linux* ajude a espalhar a notícia de que o Linux tornou-se um sistema operacional de primeira grandeza capaz de corresponder aos produtos da Sun, HP e IBM.

Agora o Linux *é* IBM. Para qualquer pessoa esperando um sinal não-ambíguo de que as águas do Linux eram seguras para nadadores corporativos, o anúncio de suporte ao Linux em 2004 pela IBM por toda sua linha de servidores deve ser bem reconfortante. Ninguém nunca foi despedido por comprar IBM; hoje em dia, o Linux é, em geral, uma proposta igualmente segura.[1]

Decidimos escrever um livro que fosse o melhor amigo do administrador de sistema Linux profissional. Onde apropriado, adaptamos os comprovados conceitos e materiais de nosso popular livro, *UNIX System Administration Handbook*. Acrescentamos uma grande quantidade de material específico ao Linux e atualizamos o restante, mas grande parte dos assuntos abordados continua a mesma. Esperamos que você concorde que o resultado é um guia de alta qualidade para administração de sistema Linux que se beneficia de uma experiência anterior.

Nenhum dos outros livros sobre administração de sistema Linux fornece a amplitude e a profundidade do material necessário para utilizar o Linux eficientemente em ambientes comerciais reais. Eis as características que diferenciam nosso livro:

- Adotamos uma postura prática. Nosso objetivo não é o de reafirmar o conteúdo de seus manuais, mas sim o de sintetizar nossa experiência coletiva em administração de sistema. Este livro contém um grande número de histórias reais e uma rica quantidade de conselhos pragmáticos.
- Este livro não é sobre como rodar o Linux em casa, em sua garagem ou em seu PDA. Descrevemos o emprego do Linux em ambientes de produção como empresas, órgãos governamentais e universidades.
- Abordamos as redes Linux em detalhe. Esse é o aspecto mais difícil da administração de sistema e a área onde acreditamos poder ser de maior ajuda.
- Não simplificamos excessivamente a abordagem dos assuntos. Nossos exemplos refletem situações da vida real com todas as suas imperfeições e complicações imprevistas. Na maioria dos casos, os exemplos foram extraídos diretamente de sistemas de produção.
- Abordamos cinco distribuições principais do Linux.

Nossas distribuições de exemplo

Como tantos outros sistemas operacionais, o Linux cresceu e se ramificou em diversas direções. Embora o desenvolvimento do kernel tenha permanecido surpreendentemente centralizado, o empacotamento e a distribuição de sistemas operacionais Linux completos são supervisionados por uma série de grupos, cada qual com sua própria agenda.

Analisamos cinco distribuições em detalhe:

- Red Hat® Enterprise Linux® 4.3 ES
- Fedora™ Core 5
- SUSE® Linux Enterprise 10.2
- Debian® GNU/Linux 3.2 "Etch" (versão/lançamento de teste de setembro de 2006)
- Ubuntu® 6.06 "Dapper Drake"

Optamos por essas distribuições porque elas se encontram entre as mais populares e porque representam a comunidade Linux como um todo. Entretanto, grande parte do material apresentado neste livro também se aplica a outras distribuições.

Damos informações detalhadas sobre cada uma dessas distribuições para cada tópico que discutimos. Comentários específicos a um determinado sistema operacional são marcados com o logotipo da distribuição.

---

1. Pelo menos para servidores. O campo de batalha atual é o desktop, um domínio sobre o qual o Microsoft Windows ainda mantém um quase monopólio. O resultado dessa luta continua sendo difícil de prever. Na época em que escrevíamos este livro, o Windows ainda fornecia uma interface com o usuário mais polida. Mas é difícil argumentar com "liberdade".

A ORGANIZAÇÃO DESTE LIVRO

Este livro é dividido em três grandes partes: Administração básica, Redes e Diversos

A seção "Administração básica" apresenta uma visão geral do Linux sob a perspectiva de um administrador de sistema. Os capítulos dessa parte cobrem a maior parte dos dados e técnicas necessários para rodar um sistema Linux dedicado.

A seção "Redes" descreve os protocolos usados em sistemas Linux e as técnicas empregadas para configurar, ampliar e manter redes. Também é visto o software de rede de alto nível. Entre os tópicos apresentados estão o DNS (Domain Name System), o NFS (Network File System), roteamento, **sendmail** e gerenciamento de redes.

A seção "Diversos" inclui uma série de informações suplementares. Alguns capítulos discutem pacotes de software opcional como o sistema de impressão Linux. Outros oferecem conselhos sobre tópicos que vão desde a manutenção de hardware até as políticas de instalação de um sistema Linux.

Cada capítulo é seguido por um conjunto de exercícios práticos. Os itens são classificados por nossa estimativa de empenho necessário para completá-los, onde "empenho" é um indicador tanto da dificuldade da tarefa como do tempo necessário para completá-la.

Há quatro níveis:

sem estrelas	Fáceis, devem ser simples e diretos.
★	Mais difíceis ou mais longos, podem exigir trabalho laboratorial.
★★	Os mais difíceis ou os mais longos, exigem trabalho laboratorial e grande aprofundamento.
★★★★★	Projetos de duração semestral (somente em alguns poucos capítulos)

Alguns dos exercícios exigem acesso de root ou **sudo** ao sistema; outros exigem a permissão do grupo sysadmin local. Ambas as exigências são mencionadas no texto do exercício.

NOSSOS COLABORADORES

Ficamos encantados por Adam Boggs, Bryan Buus e Ned McClain poderem se juntar a nós mais uma vez como autores colaboradores. Com esta edição, também damos boas-vindas a Ben Whaley, Tobi Oetiker, Fritz Zaucker, Jeffrey S. Haemer, David Schweikert e Scott Seidel como colaboradores e amigos. O conhecimento profundo desses profissionais em uma série de áreas enriqueceu muito o conteúdo deste livro. Acima de tudo, agradecemos a Lynda McGinley, que, além de assumir uma quantidade substancial de texto, também trabalhou incansavelmente para organizar e facilitar o trabalho de nossos colaboradores.

INFORMAÇÕES PARA CONTATO

Para sugestões, comentários e informes de erros, envie um e-mail para linux@book.admin.com. Os autores costumam responder à maioria dos e-mails, porém você tem de ser paciente; às vezes levam alguns dias até que um deles possa responder. Devido ao volume de e-mail que esse endereço recebe, lamentamos não poder responder a perguntas técnicas. Para obter uma cópia atualizada da lista de bugs dos autores e outras informações de última hora, visite o site www.admin.com. Esperamos que você goste deste livro e desejamos muita sorte em suas aventuras no mundo da administração de sistema!

Evi Nemeth
Garth Snyder
Trent R. Hein
Outubro de 2006

Prefácio à primeira edição

Eu estava bastante ansioso por ver essa versão do *UNIX® System Administration Handbook* para Linux. A terceira edição deste último incluía uma cobertura sobre o Red Hat Linux, mas era apenas uma das quatro variantes distintas do UNIX. Esta versão do livro abrange as principais distribuições Linux e omite a maior parte do material que não é relevante ao Linux. Estava curioso para saber quanta diferença ela faria.

Muita, como você pode ver aqui. As distribuições do Linux partem de um conjunto comum de softwares de código-fonte aberto e, portanto, elas são muito mais semelhantes entre si do que acontece com outras versões do UNIX. Como resultado, o texto parece ter se tornado consideravelmente mais específico. Em vez de sugerir várias maneiras de como seu sistema *deveria* se comportar, os autores agora podem informá-lo exatamente sobre como ele *realmente* se comporta.

Ao mesmo tempo, fica claro que toda a riqueza e a variedade do software UNIX ainda estão representadas aqui. Quase todos os softwares mais populares do mundo rodam no Linux hoje em dia, e as instalações Linux estão se defrontando com cada vez menos comprometimentos. À medida que fornecedores de peso como IBM, Oracle e Silicon Graphics adotam o Linux, ele está rapidamente se tornando o padrão universal com o qual as demais versões do UNIX estão sendo comparadas (e, nem sempre, favoravelmente!).

Como este próprio livro demonstra, os sistemas Linux são tão funcionais, seguros e confiáveis quanto seus equivalentes proprietários. Graças ao esforço contínuo de seus milhares de desenvolvedores, o Linux está mais pronto do que nunca para emprego nas linhas de frente do mundo real. Os autores deste livro conhecem bem o terreno, e estou feliz por deixá-lo em mãos tão capazes. Aproveite!

Linus Torvalds
Abril de 2002

Agradecimentos

Muitas pessoas auxiliaram na realização deste livro, de uma maneira ou de outra, ajudando em tudo, desde revisões técnicas ou sugestões de exercícios até apoio moral. Estas pessoas merecem agradecimentos especiais por estarem sempre presentes:

Bo Connell	Jon Corbet	Jim Lane
Sam Leffler	Cricket Liu	Derek Martin
Laszlo Nemeth	Eric Robinson	Sam Stoller
Paul Vixie	Aaron Weber	Greg Woods

Nossos editores da Prentice Hall, Catherine Nolan e Mary Franz, merecem não só nossos agradecimentos como também um prêmio por lidar bem com autores insanos e uma equipe de suporte que às vezes parecia envolver milhares de colaboradores.

Mary Lou Nohr mais uma vez realizou um trabalho extraordinário como editora de texto. Ela é rolo compressor e uma flor, tudo em uma só pessoa. Gostaríamos de dizer que ficaríamos felizes de trabalhar novamente com ela no futuro, mas esse tempo verbal não é permitido.

O paciente e meticuloso trabalho de indexação de Mark G. Sobell foi recompensador. Estamos muito satisfeitos com o resultado e apreciamos muito a ajuda.

Por fim, Evi agradece e pede desculpas aos diversos bares e cafés das praias do Caribe cujas conexões sem fio ela seqüestrou ancorando seu barco no ponto de amplitude máxima de sinal. Como estava alheia ao paraíso a sua volta e trabalhava arduamente nos capítulos do livro, ela jurou que esta seria sua última edição. Mas a quem ela está enganando?

Sobre os autores

Evi Nemeth (Sailingevi@yahoo.com) aposentou-se pela faculdade de ciências da computação da Universidade de Colorado, mas ainda dedica-se à pesquisa de rede na CAIDA (Cooperative Association for Internet Data Analysis) no San Diego Supercomputer Center. Atualmente, ela está explorando o Caribe em seu novo brinquedo, um veleiro de 40 pés chamado *Wonderland*.

Garth Snyder (Garth@grsweb.us) trabalhou na NeXT e na Sun, é formado em engenharia elétrica pelo Swarthmore College e obteve um MD e um MBA pela Universidade de Rochester.

Trent R. Hein (Trent@atrust.com) foi co-fundador da Applied Trust Engineering, uma empresa que fornece segurança de infraestrutura de rede e serviços de consultoria sobre desempenho de sistemas de computação. Trent é bacharel em ciência da computação pela Universidade de Colorado.

Sobre os colaboradores

Lynda McGinley trabalha com administração de sistema e de redes há 20 anos em um ambiente educacional e foi colaboradora na terceira edição do *UNIX System Administration Handbook*. Atualmente, trabalha na University Corporation for Atmospheric Research como engenheira de segurança de sistema. Lynda acredita ser mais lembrada por ser sysadmin de Evi por vários anos na Universidade do Colorado. :-)

Ben Whaley é engenheiro sênior da Applied Trust Engineering. É formado em ciência da computação pela Universidade do Colorado e é engenheiro com certificação Red Hat. Além de sujar as mãos com a programação Perl e PHP, Ben é um dos mentores do programa Big Brother Big Sister.

Adam Boggs foi estudante de administração de rede dos laboratórios da Universidade do Colorado no final da década de 1990. Ele passou os últimos vários anos como um desenvolvedor de kernel trabalhando em sistemas de arquivos e armazenamento para Solaris e Linux. Atualmente, ele trabalha como administrador de rede, armazenamento e clusters para supercomputadores utilizados em modelagem atmosférica.

Jeffrey S. Haemer (jsh@usenix.org) faz trabalhos em UNIX e Linux comerciais desde 1983 e até hoje nunca utilizou o Microsoft Word. Onze desses anos foram para as várias encarnações do fabricante de impressoras QMS. Evi ensinou Jeff a guiar. Seu palíndromo favorito é "A man, a plan, a canal, Suez".

Tobi Oetiker (Tobi@oetiker.ch) é engenheiro elétrico por formação, e administrador de sistema por vocação. Ele está na área de administração de sistema desde 1994. Depois de equipar um ambiente de computação para os alunos e para a equipe da ETH Zurich por anos e escrever alguns aplicativos de código-fonte aberto populares como MRTG, RRDtool e SmokePing, agora ele trabalha na Oetiker+Partner AG, um provedor serviços de TI e consultoria. Ele é casado e vive na Suíça. Informações adicionais sobre Tobi podem ser encontradas em tobi.oetiker.ch.

Fritz Zaucker é físico por treinamento. Depois de alguns anos de pesquisa sobre mudanças climáticas, ele passou os últimos 10 anos dirigindo o IT Support Group (ISG.EE) no Department of Information Technology and Electrical Engineering da ETH Zurich. Junto com um grupo de colaboradores dedicados, ele transformou o ISG.EE em uma organização profissional voltada para administração de sistema eficiente, infra-estrutura de TI confiável e satisfação do cliente. Agora, juntou-se a alguns de seus amigos e colegas da Oetiker+Partner AG, uma empresa de consultoria de gestão de TI e provedor de serviços de TI. Visite Fritz em www.zaucker.ch. Fritz navegou com Evi e suas sobrinhas no Mediterrâneo e sinceramente recomenda que você se una a ela na *Wonderland*.

Scott Seidel é engenheiro sênior da Applied Trust Engineering, onde é especialista em segurança corporativa e gerenciamento de desempenho. Ele formou-se em administração de empresas com ênfase em sistemas de informações e finanças da Universidade do Colorado em Boulder. Os interesses de pesquisa do Scott são monitoramento de sistema e virtualização de servidor. Quando não está administrando sistemas Linux, Scott pode ser encontrado na cozinha aperfeiçoando um novo drink caseiro.

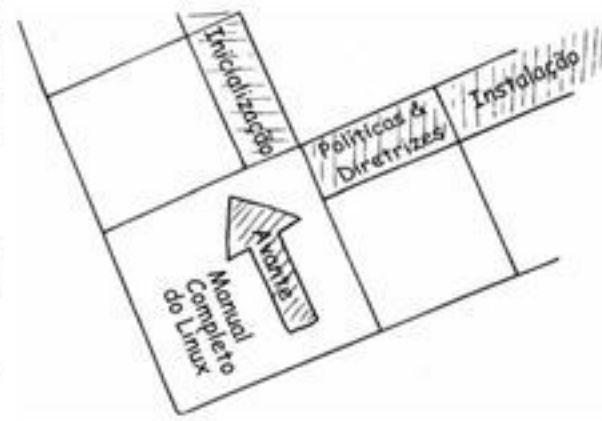
Bryan Buus é vice-presidente de engenharia da Adeptive Software, uma empresa de desenvolvimento de software personalizado. Antes do estouro da bolha do ponto-com, Bryan era vice-presidente de desenvolvimento da XOR, chefiando a divisão de serviços Web e Internet da empresa. Bryan tem bacharelado e mestrado em ciência da computação pela Universidade de Boston.

Ned McClain (ned@atrust.com) é co-fundador e CTO da Applied Trust Engineering. Ele dá conferências sobre vários tópicos de administração e segurança de sistema em conferências técnicas como USENIX nos Estados Unidos e APRICOT na Ásia. Ned formou-se em ciência da computação pelo College of Engineering da Cornell University e é CISSP #39389.

David Schweikert trabalha como administrador de sistema na ETH Zurich, Suíça, onde é, entre outros, responsável pelo sistema de e-mail do departamento de engenharia elétrica. É o desenvolvedor dos projetos de código-fonte aberto Mailgraph (uma ferramenta que plota estatísticas de e-mail) e Postgrey (uma implementação de "listas cinzas" — um método de bloquear spam — para Postfix). David também tem interesses em bancos de dados e administração de sistema de grande porte.

PARTE 1
ADMINISTRAÇÃO BÁSICA

1 Por onde começar



Decidimos escrever um livro que fosse um verdadeiro companheiro do administrador de sistema, fornecendo conselhos práticos e a teoria básica sobre administração de sistema que não podem ser obtidos da leitura de manuais. Como resultado, este livro foi projetado para complementar — não substituir — a documentação do seu sistema Linux.

Este livro o ajuda de cinco maneiras:

- Revisa os principais sistemas administrativos, identificando as diferentes partes de cada um e explicando como elas funcionam em conjunto.
- Apresenta técnicas administrativas gerais que, por experiência, consideramos eficientes e úteis.
- Ajuda a escolher soluções que continuam a funcionar bem mesmo quando sua instalação cresce em tamanho e complexidade.
- Ajuda a fazer uma separação entre boas e as más idéias e mostra como os distribuidores de sistema operacional são capazes de cometer as mais variadas atrocidades contra o bom gosto.
- Resume procedimentos comuns de modo que você não precise lidar com a grande quantidade de detalhes dos manuais para executar tarefas simples.

É impossível executar essas funções com perfeita objetividade, mas acreditamos que deixamos relativamente claras nossas tendências ao longo do texto. Uma das coisas interessantes em relação à administração de sistema é que pessoas razoáveis podem ter noções significativamente diferentes sobre o que constitui as políticas e procedimentos mais apropriados. Oferecemos a você nossas opiniões subjetivas como dados brutos. Você terá de decidir por conta própria quanto aceitar e em que extensão aplicar nossos comentários ao seu ambiente.

1.1 BACKGROUND SUGERIDO

Neste livro, partimos do princípio de que você já tem certa experiência em Linux ou UNIX. Em particular, você deve ter uma noção geral da aparência e funcionamento gerais do Linux sob a perspectiva do usuário antes de entrar na sua administração. Vários livros podem ajudá-lo a acelerar este processo; consulte a lista apresentada na página 13.

A maioria das tarefas administrativas é executada editando arquivos de configuração e escrevendo scripts, de modo que você deve estar familiarizado com um editor de texto. Para o desalento de muitos, usar o Microsoft Word como único editor de texto é um obstáculo significativo para uma administração de sistema eficiente.

Recomendamos veementemente que você aprenda o **vi** (que é visto mais comumente em sistemas Linux em sua forma reescrita, o **vim**). Ele é padrão em todos os sistemas UNIX e Linux, e, embora possa parecer um pouco fraco quando comparado com opções mais sofisticadas como **emacs**, ele é poderoso e completo. Também gostamos do **pico**, que é um editor simples e fácil de usar para administradores de sistemas iniciantes. Ele é incluído em várias distribuições. Seja cauteloso em relação a editores não-padrão; se você se viciar no uso de um deles, acabará se cansando de levá-lo consigo para instalá-lo em cada novo sistema.

Um dos pilares da administração (e um tema que é abordado ao longo do livro) é o emprego de scripts para automatizar tarefas administrativas. Para ser um administrador eficiente, você deve ser capaz de ler e modificar Perl e os scripts **sh** (que no mundo Linux são na realidade scripts **bash**). Os scripts que criamos a partir do zero podem ser escritos na linguagem de script ou de shell de sua preferência.

Consulte cpan.org para obter uma seleção completa dos softwares Perl úteis.

Para novos projetos de script, recomendamos Perl ou Python. Como linguagem de programação, Perl é um pouco estranho (OK, mais que um pouco). Entretanto, ele inclui muitos recursos indispensáveis para os administradores de sistemas. O livro editado pela O'Reilly, *Programming Perl*, de Larry Wall et al., é um texto-padrão; ele também é um modelo para escrever bem assuntos técnicos. Uma relação completa é dada na página 10.

Muitos administradores preferem o Python ao Perl e sabemos de instalações que estão fazendo um esforço concentrado para conversão de Perl em Python. O Python é uma linguagem mais elegante que o Perl, e os scripts em Python são geral-

mente mais legíveis e mais fáceis de manter. Um conjunto de links úteis comparando Python com outras linguagens de script (inclusive Perl) pode ser encontrado em

www.python.org/doc/Comparisons.html

Recomendamos também que você aprenda **expect**, que não é uma linguagem de programação, mas um front-end para programas interativos. É provável que você adote o **expect** rapidamente.

1.2 O RELACIONAMENTO DO LINUX COM O UNIX

Utilizar os nomes Linux e UNIX juntos em uma frase é como pisar em um campo político minado ou talvez como cair em um grande terreno de areia movediça. Eis nossa breve versão dos fatos, declarados da forma mais clara e objetiva possível.

O Linux é uma reimplementação e reelaboração do UNIX. Ele obedece ao padrão POSIX, executa em várias plataformas de hardware e é compatível com a maioria dos softwares UNIX existentes. Ele difere da maioria das outras variantes do UNIX pelo fato de ser gratuito, com código-fonte aberto e desenvolvido de maneira cooperativa, com colaborações provenientes de milhares de indivíduos e organizações. O Linux incorpora refinamentos técnicos que não existiam nas versões originais UNIX, então é mais do que um simples clone do UNIX. Ele também é uma entidade juridicamente distinta e não pode ser adequadamente referido como "UNIX".

Vale notar que o Linux não é o único sistema operacional do tipo UNIX no mundo. FreeBSD, NetBSD e OpenBSD, todos os quais são ramificações do Berkley Software Distribution da UC Berkley, possuem seus próprios e ardorosos seguidores. Esses sistemas operacionais são geralmente comparáveis ao Linux em termos de recursos e confiabilidade, embora eles tenham um pouco menos de suporte por parte de fornecedores de software independentes.

Software Linux é software UNIX. Graças principalmente ao Projeto GNU, a maioria dos principais softwares que dão aos sistemas UNIX seu valor foi desenvolvida sob alguma forma de modelo de código-fonte aberto. O mesmo código roda em sistemas Linux e não-Linux. Por exemplo, o servidor Web Apache não se importa realmente se está rodando Linux ou HP-UX. Sob o ponto de vista dos aplicativos, o Linux é simplesmente uma das variantes do UNIX mais bem suportadas.

Os sistemas UNIX e Linux vêm sendo usados em ambientes de produção há vários anos. Este livro, diferentemente da maioria dos demais sobre administração de sistema Linux, concentra-se no emprego efetivo do Linux em um ambiente de produção — não simplesmente como um desktop de um único usuário.[1]

1.3 LINUX EM CONTEXTO HISTÓRICO

O Linux se originou em 1991 como um projeto pessoal de Linus Torvalds, um universitário finlandês. Ele concebeu originalmente o projeto como uma modesta ramificação do Minix, um sistema operacional modelo escrito por Andrew S. Tannenbaum. Entretanto, o Linux gerou interesse substancial no mundo todo, e o kernel logo ganhou vida própria. Explorando o poder do desenvolvimento cooperativo, Linus foi capaz de empreender uma tarefa muito mais ambiciosa. A versão 1.0 do kernel foi lançado em 1994; na época em que a primeira edição deste livro estava sendo escrita (setembro de 2006), a versão estável mais recente do kernel do Linux era a 2.6.17.

Pelo fato de o Linux possuir grande parte das características de seus ancestrais UNIX, não é muito razoável localizar a origem da era Linux no ano de 1991. A história do UNIX remonta a várias décadas antes, mais precisamente a 1969 quando o UNIX nasceu como um projeto de pesquisa no AT&T Bell Labs. Em 1976, o UNIX tornou-se disponível gratuitamente nas universidades e, portanto, tornou-se a base de vários cursos de sistemas operacionais e de projetos de pesquisa acadêmicos.

O Berkley UNIX começou em 1977 quando o Computer Systems Research Group (CSRG) na Universidade da Califórnia, Berkeley, licenciou o código da AT&T. As versões do Berkeley (chamadas BSD, acrônimo para Berkeley Software Distribution) começaram com o 1BSD para o PDP-11 e culminaram em 1993 com o 4.4BSD.

À medida que o UNIX ganhava aceitação comercial, o preço das licenças dos códigos-fonte crescia rapidamente. Por fim, Berkeley estabeleceu o objetivo de longo prazo de eliminar do BSD o código pertencente à AT&T, um processo tedioso e demorado. Antes que o trabalho pudesse ser completado, Berkeley perdeu as verbas para a pesquisa de sistemas operacionais e o CSRG foi abandonado.

Antes de seu desmantelamento, o CSRG lançou sua versão final sem o código da AT&T, conhecida como 4.4 BSD-Lite. A maioria das versões atuais do BSD UNIX (incluindo o FreeBSD, o NetBSD, o Mac OS X,[2] e o OpenBSD) afirma ter o pacote 4.4 BSD-Lite como seu antepassado.

A maioria das outras versões do UNIX (incluindo HP-UX e Solaris) é descendente da linhagem AT&T original. O Linux não compartilha código com as versões AT&T ou BSD do UNIX, mas de uma perspectiva funcional ele inclui-se em algum lugar entre as duas.

---

1. Um ambiente de "produção" é aquele com o qual uma organização conta para realizar trabalhos reais (em oposição a teste, pesquisa ou desenvolvimento).
2. O kernel Mac OS X é, estritamente falando, uma variante do Mach, um sistema híbrido que inclui partes do BSD que não são UNIX.

1.4 DISTRIBUIÇÕES DO LINUX

Consulte a seção que inicia na página 659 para obter comentários adicionais sobre as distribuições.

O Linux difere de outras variantes UNIX pelo fato de que o projeto do kernel básico define apenas um kernel de OS. O kernel tem de ser empacotado junto com comandos, daemons e outros softwares para formar um sistema operacional completo e utilizável — em termos do Linux, uma "distribuição". Todas as distribuições do Linux compartilham da mesma linhagem do kernel, porém os materiais auxiliares que vêm junto com esse kernel podem variar significativamente entre estas distribuições.

Esses "materiais auxiliares" consistem em uma vasta coleção de software desenvolvido nos últimos 30 anos por milhares de pessoas. Argumentou-se, com certa razão, que o ato de referir-se ao sistema operacional concluído simplesmente como "Linux" falha em reconhecer as contribuições desses desenvolvedores e o contexto histórico em que eles trabalharam. Infelizmente, a alternativa mais comumente sugerida, "GNU/Linux" tem suas próprias questões políticas e só foi endossada oficialmente pela distribuição Debian. A entrada da Wikipedia "GNU/Linux naming controversy" descreve os argumentos de ambos os lados.

As distribuições variam em seu foco, suporte e popularidade. A Tabela 1.1 lista as distribuições de uso geral mais populares. As distribuições são listadas em ordem alfabética, não por preferência ou popularidade.

Tabela 1.1 Distribuições Linux de uso geral mais populares

Distribuição	Site Web	Comentários
CentOS	www.centos.org	Correspondente livre/gratuito do Red Hat Enterprise Linux.
Debian	www.debian.org	Uma distribuição não-comercial popular.
Fedora	fedora.redhat.com	Red Hat Linux descorporatizado.
Gentoo	www.gentoo.org	Distribuição baseada em código-fonte.
Mandriva[a]	www.mandriva.com	Uma das distribuições mais amigáveis ao usuário.
openSUSE	www. opensuse. org	Correspondente livre/gratuito do SUSE Linux Enterprise.
Red Hat Enterprise	www.redhat.com	Red Hat Linux supercorporatizado
Slackware	www.slackware.com	Distribuição esquelética, básica e estável.
SUSE Linux Enterprise	www.novell.com/linux	Forte na Europa, vários idiomas.
TurboLinux	www.turbolinux.com	Forte na Ásia, vários idiomas.
Ubuntu	www.ubuntu.com	A versão simplificada do Debian.

a. Ex-Mandrakelinux

Muitas distribuições menores não são listadas na Tabela 1.1 e muitas distribuições de uso especial não listadas são destinadas a grupos com necessidades especializadas (como desenvolvedores de sistemas embutidos).

Uma distribuição útil não encontrada na Tabela 1.1 é o Knoppix (www.knoppix.com), uma versão do Linux que reside em um CD-ROM inicializável. Seu principal valor reside em sua utilidade como um CD de recuperação para um sistema de Linux que não inicializa devido a um comprometimento da segurança ou um problema técnico. O conceito de CD inicializável revelou-se tão popular que a maioria das distribuições importantes está adotando essa idéia. Agora que o Ubuntu pode inicializar do CD de distribuição, o Knoppix está tornando-se menos importante. Uma lista atualizada de distribuições Linux inicializáveis pode ser encontrada em www.frozentech.com/content/livecd.php.

O Red Hat foi uma força dominante no mundo Linux durante a maior parte da última década e suas distribuições são predominantes na América do Norte. Em 2003, a distribuição Red Hat Linux original foi dividida em uma linha de produção centralizada chamada Red Hat Enterprise Linux (a qual às vezes chamamos de RHEL neste livro) e um projeto de desenvolvimento baseado em comunidades chamado Fedora. A divisão foi motivada por várias razões técnicas, econômicas, logísticas, jurídicas, mas até agora as distribuições continuam semelhantes. O RHEL oferece grande suporte e estabilidade, mas é efetivamente impossível utilizar sem pagar as taxas de licença para a Red Hat.

O CentOS Project (www.centos.org) reúne códigos-fonte que o Red Hat é obrigado a liberar sob vários acordos de licença (mais notavelmente, a licença pública GNU) e o monta em uma distribuição completa que é sinistramente semelhante ao Red Hat Enterprise Linux, mas é gratuita. A distribuição não tem a marca Red Hat e algumas ferramentas "proprietárias" (patenteadas) do Red Hat, mas é equivalente em outros aspectos. O CentOS aspira plena compatibilidade do código binário (incluindo os bugs!) com o RHEL.

O CentOS é uma excelente escolha para sites que querem implantar uma distribuição orientada a produção sem pagar nada para a Red Hat. Uma abordagem híbrida também é possível: servidores de produção podem executar Red Hat Enterprise e tirar proveito do excelente suporte do Red Hat, enquanto os desktops executam CentOS. Esse arranjo abrange as bases importantes em termos de risco e suporte e ao mesmo tempo também oferece um mínimo de custo e complexidade administrativo.

O SUSE, agora parte da Novell, recentemente pegou o caminho do Red Hat e o bifurcou em duas distribuições relacionadas: uma (openSUSE) que contém apenas software gratuito; e outra (SUSE Linux Enterprise) que tem um custo, inclui um caminho de suporte formal e oferece alguns recursos extras. No passado parecia haver um esforço para ocultar a existência da versão livre do SUSE, mas a Novell inverteu essa orientação dos antigos proprietários do SUSE com sua nova edição. Agora, você pode ir direto para www.opensuse.org a fim de obter as últimas informações. Nada neste livro é específico de uma ou outra distribuição SUSE, então simplesmente nos referimos a elas coletivamente como "SUSE".

As distribuições Debian e Ubuntu mantêm um compromisso ideológico com o desenvolvimento da comunidade e abre acesso, para que nunca existam questões sobre que partes da distribuição são gratuitas ou redistribuíveis. O Debian sobrevive do zelo e da boa vontade da comunidade GNU, enquanto o Ubuntu se beneficia atualmente do financiamento filantrópico do empresário sul-africano Mark Shuttleworth. O Ubuntu pode até enviar CDs pelo correio de graça, não cobrando nem a postagem.

Então, qual é a melhor distribuição?

Uma rápida pesquisa na Internet revelará que isto é uma das coisas mais freqüentemente perguntadas — e menos freqüentemente respondidas — sobre o Linux. A resposta correta para o seu caso depende de como você pretende usar o sistema, as variedades de UNIX com as quais você já está familiarizado, suas preferências políticas e suas necessidades de suporte.

A maioria das distribuições do Linux pode fazer praticamente tudo o que você imagina querer fazer com um sistema Linux. Algumas delas podem exigir a instalação de software adicional para serem totalmente funcionais e outras podem facilitar determinadas tarefas; entretanto, suas diferenças não são muito significativas. Na realidade, há certo mistério no motivo de existirem tantas distribuições diferentes, cada uma das quais alardeando sua "instalação fácil" e uma "gigantesca biblioteca de programas" como características distintivas. É difícil evitar a conclusão de que as pessoas simplesmente gostam de criar distribuições do Linux.

Por outro lado, já que o foco deste livro está no gerenciamento de instalações Linux de grande porte, preferimos distribuições como o Red Hat Enterprise Linux que levam em consideração o gerenciamento de redes de máquinas. Algumas distribuições são desenvolvidas tendo em mente ambientes de produção, e outras, não. As migalhas extras de assistência que os sistemas orientados a ambientes de produção oferecem podem representar uma diferença significativa na facilidade com que os sistemas são administrados.

Ao adotar uma distribuição, você está investindo na maneira de fazer as coisas de um determinado fornecedor. Em vez de olhar somente os recursos do software instalado é melhor considerar como a sua empresa e aquele fornecedor em questão irão trabalhar um com o outro nos anos vindouros. Algumas perguntas importantes a serem feitas são as seguintes:

- Esta distribuição ainda existirá daqui a cinco anos?
- Esta distribuição estará no topo dos patches de segurança mais recentes?
- Esta distribuição lançará software atualizado prontamente?
- Se tiver problemas, o fornecedor conversará com você?

Analisadas sob este ângulo, algumas das pequenas distribuições mais interessantes e não convencionais não parecem muito convidativas. Por outro lado, as distribuições mais viáveis não são necessariamente as mais indicadas para um ambiente corporativo. Por exemplo, acreditamos que o Debian Linux (OK, OK, o Debian GNU/Linux!) permanecerá viável por um bom tempo apesar do fato de a Debian não ser uma empresa; não vender nada e não oferecer nenhum tipo de suporte.

Uma lista abrangente de distribuições, incluindo muitas não-inglesas, pode ser encontrada em www.linux.org/dist, lwn.net/Distributions ou distrowatch.com.

Neste livro, utilizamos cinco distribuições populares como nossos exemplos: o Red Hat Enterprise Linux 4.3 ES, o Fedora Core 5, o SUSE Linux Enterprise 10.2, o Ubuntu 6.06 ("Dapper Drake") e o lançamento de teste (em setembro de 2006) do Debian GNU/Linux 3.2 ("Etch"). Esses sistemas representam uma fatia do mercado corporativo do Linux e contabilizam a maioria das instalações do Linux em uso hoje em dia em grandes sistemas.

Ferramentas administrativas específicas a cada distribuição

Muitas distribuições incluem ferramentas orientadas visualmente (como o Red Hat Network Administration Tool ou o YaST do SuSE) que o ajudam a configurar ou administrar aspectos selecionados do sistema. Essas ferramentas podem ser muito úteis, especialmente para os administradores novatos, mas tendem a obscurecer os detalhes daquilo que está realmente ocorrendo quando você faz modificações. Neste livro, abordamos os mecanismos subjacentes a que as ferramentas gráficas recorrem em vez de abordarmos as próprias ferramentas, por várias razões.

Em primeiro lugar, as ferramentas visuais tendem a ser proprietárias ou pelo menos específicas de determinada distribuição — elas introduzem variações nos processos que podem ser bastante significativas entre as distribuições em um nível mais baixo. Em segundo lugar, acreditamos que é importante que os administradores tenham uma compreensão exata de como os seus sistemas funcionam. Quando um sistema falha, em geral as ferramentas visuais não são úteis para rastrear e

resolver os problemas. Por fim, a configuração manual costuma ser melhor: é mais rápida, mais flexível e mais confiável e também mais fácil de ser programada em scripts.

1.5 CONVENÇÕES TIPOGRÁFICAS E DE NOTAÇÃO

Neste livro, os nomes de arquivos, comandos e argumentos literais para comandos estão em negrito. Marcadores de lugar (isto é, argumentos de comandos que não devem ser interpretados literalmente) estão em itálico. Por exemplo, no comando:

```
cp arquivo diretório
```

supõe-se que você substituirá *arquivo* e *diretório* pelo nome de um arquivo e um diretório realmente existentes.

Trechos de arquivos de configuração e sessões em terminais são mostrados em uma fonte de largura fixa[3]. Algumas vezes, fazemos comentários sobre sessões interativas com texto em itálico. Por exemplo:

```
$ grep Bob /pub/phonelist /* Verifica o telefone de Bob # */
Bob Knowles 555-2834
Bob Smith 555-2311
```

Exceto por esses casos específicos, tentamos manter em um mínimo possível as fontes especiais e as convenções de formatação até onde a compreensão não fosse comprometida. Por exemplo, muitas vezes falamos de entidades como o grupo Linux chamado daemon e o printer anchor-lw, sem nenhuma formatação especial.

Em geral, utilizamos as mesmas convenções das páginas do comando manual para indicar a sintaxe de comandos:

- Qualquer coisa entre colchetes ("[" e "]") é opcional.
- Qualquer coisa seguida de reticências ("...") pode ser repetida.
- Chaves ("{" e "}") significam que devemos selecionar um dos itens separados por barras verticais ("|").

Por exemplo, a especificação:

```
bork [ -x ] { on | off } nomeDoArquivo …
```

corresponderia a qualquer um dos seguintes comandos:

```
bork on /etc/passwd
bork -x off /etc/passwd /etc/termcap
bork off /usr/lib/tmac
```

Utilizamos caracteres no estilo curinga de shell para correspondência de padrão:

- Um asterisco (*) corresponde a zero ou mais caracteres.
- Um ponto de interrogação (?) corresponde a um caractere.
- Um til (~) representa o diretório inicial do usuário atual.
- *~usuário* representa o diretório inicial de *usuário*.

Por exemplo, poderíamos referenciar os diretórios dos scripts de inicialização do Debian **/etc/rc0.d**, **/etc/rc1.d** e assim por diante com o padrão abreviado **/etc/rc*.d**.

O texto dentro das aspas normalmente tem um significado técnico preciso. Nesses casos, ignoramos as regras normais do inglês e colocamos a pontuação fora das aspas para que não haja dúvida sobre o que está e o que não está incluído.

Informações específicas de sistema

As informações neste livro geralmente se aplicam a todos os exemplos de distribuições de sistemas a menos que seja dada uma atribuição específica. Detalhes particulares da distribuição de um sistema são marcados com o logotipo do fornecedor:

Red Hat® Enterprise Linux® 4.3 ES

Ubuntu® 6.06 "Dapper Drake"

Fedora™ Core 5

Debian® GNU/Linux 3.2 "Etch" (versão de teste de 9/06)

SUSE® Linux Enterprise 10.2

3. Na verdade, ela não é realmente uma fonte de largura fixa, mas parece. Gostamos mais dela do que das fontes de largura fixa verdadeiras que experimentamos. Essa é a razão pela qual as colunas de alguns exemplos podem não estar perfeitamente alinhadas.

Esses logotipos são usados com a gentil permissão de seus respectivos proprietários. Entretanto, os distribuidores não revisaram nem endossaram o conteúdo deste livro.

1.6 ONDE BUSCAR INFORMAÇÕES

A documentação do Linux está distribuída por uma série de fontes, algumas das quais você encontrará instaladas em seu sistema e outras que residem na Internet. As mais importantes são:

- Páginas man, lidas por meio do comando **man**
- Documentos Texinfo, lidos por meio do comando **info**
- HOWTOs, pequenas notas sobre vários assuntos (www.tdlp.org)
- Guias, tratados mais longos sobre vários assuntos (www.tdlp.org)
- Documentação específica de uma determinada distribuição
- Páginas Web associadas a projetos de software específicos

As páginas man e os documentos Texinfo constituem a documentação "on-line" tradicional (embora, obviamente, toda a documentação esteja on-line de uma forma ou outra). Esses documentos são tipicamente instalados com o sistema; páginas man específicas de um determinado programa normalmente vêm de carona sempre que você instala um sistema.

As páginas man são descrições concisas de comandos individuais, drivers, formatos de arquivos ou rotinas de biblioteca. Elas não cobrem tópicos mais gerais do tipo "Como instalo um novo dispositivo?" ou "Por que o meu sistema é tão lento?". Para essas perguntas, consulte os HOWTOs.

Os documentos Texinfo foram criados há muito tempo pelo pessoal do GNU em reação ao fato de o comando **nroff** usado para formatar páginas man ser proprietário da AT&T. Hoje em dia, temos o **groff** próprio do GNU para executar essa tarefa para nós, e a questão do **nroff** já não é mais importante. Infelizmente, muitos pacotes GNU persistem em documentar-se com arquivos Texinfo em vez de páginas man. Além de definir um segundo padrão desnecessário para documentação, o Texinfo é considerado um sistema de hipertexto muito confuso.

Para fugir do inferno do Texinfo, redirecione a saída de **info** com o comando **less** para escapar do sistema de navegação predefinido de **info**. Como um efeito colateral, esse procedimento também permite tirar proveito dos recursos de pesquisa integrados em **less**.

Felizmente, os pacotes documentados com Texinfo normalmente instalam comentários sobre as páginas man que o informam como usar o comando **info** para ler informações de determinados pacotes. Você pode seguramente usar apenas o comando **man** para fazer pesquisas nos manuais e aprofundar-se no uso de **info** apenas quando instruído a fazer isso. **info** inicia você nos mistérios obscuros do Texinfo.

HOWTOs e guias são mantidos pelo Linux Documentation Project, acessível on-line em www.linuxdoc.org. O LDP é um repositório central para todo tipo de informação útil sobre o Linux. Ele também centraliza os esforços para traduzir documentos relativos ao Linux em outros idiomas.

Alguns guias LDP on-line e gratuitos de particular relevância para os administradores de sistema são *The Linux System Administrators' Guide*, de Lars Wirzenius, Joanna Oja, Stephen Stafford e Alex Weeks; o *Advanced Bash-Scripting Guide*, de Mendel Cooper; *The Linux Network Administrator's Guide, Second Edition*, de Olaf Kirch e Terry Dawson; e *Linux System Administration Made Easy*, de Steve Frampton.

Infelizmente, muitos dos documentos LDP não são assiduamente mantidos. Já que o tempo Linux corre mais rápido que o tempo real (assim como os anos de vida de um cão em relação aos anos de vida de um homem), documentos não atualizados logo se tornam obsoletos. Sempre verifique a data de um HOWTO ou de um guia e pondere sua credibilidade de acordo com sua atualidade.

Muitas das partes mais importantes da base de software do Linux são mantidas por terceiros, como o Internet Systems Consortium e a Apache Software Foundation. Esses grupos geralmente fornecem documentação adequada para os pacotes que eles distribuem. As distribuições algumas vezes empacotam o software, mas economizam na documentação, de modo que normalmente é útil verificar na fonte original se não há materiais adicionais disponíveis.

Uma outra fonte útil de informações sobre o projeto de muitos pacotes de software Linux é a série de documentos "Request for Comments" que descreve os protocolos e procedimentos usados na Internet. Consulte a página 97 para informações adicionais.

Organização das páginas man

As páginas man do Linux são divididas tipicamente em nove seções, conforme ilustrado na Tabela 1.2.

Tabela 1.2 As seções das páginas man do UNIX

Seção	Conteúdo
1	Aplicativos e comandos no nível do usuário
2	Chamadas de sistema e códigos de erros do kernel
3	Chamadas de biblioteca
4	Drivers de dispositivos e protocolos de redes
5	Formatos de arquivos padrão
6	Jogos e demonstrativos
7	Arquivos e documentos variados
8	Comandos de administração de sistema
9	Interfaces e especificações obscuras do kernel

Algumas seções são subdivididas mais ainda. Por exemplo, a seção 3M contém páginas man para a biblioteca de matemática do sistema. As seções 6 e 9 geralmente são vazias. Muitos sistemas possuem uma seção dos manuais chamada "l" para páginas man locais. Uma outra convenção comum é a seção "n" para subcomandos específicos a software (como os comandos **bash** integrados).

A entrada **nroff** para páginas man é tradicionalmente armazenada nos diretórios **/usr/share/man/man***X*, onde X é um dígito de **1** a **9** ou **l** ou **n**. As páginas são normalmente compactadas com o **gzip** para economizar espaço. (O comando **man** sabe como descompactá-las durante a execução.) Versões formatadas dos manuais são mantidas em **/var/cache/man/cat***X*. O comando **man** formatará as páginas man conforme necessário; se os diretórios **cat** forem graváveis, **man** também depositará páginas formatadas à medida que forem criadas, gerando um cache de páginas man comumente lidas.

O comando **man** na verdade pesquisa uma série de diretórios diferentes para localizar as páginas man que você solicita. Você pode determinar o caminho de pesquisa por meio do comando **manpath**. Esse caminho (no Fedora) é típico:

```
$ manpath
/usr/kerberos/man:/usr/local/share/man:/usr/share/man/en:/usr/share/man
```

Se necessário, podemos configurar nossa variável de ambiente MANPATH para sobrescrever o caminho padrão. Você também pode configurar o padrão no nível de sistema em **/etc/man.config** (RHEL e Fedora) ou em **/etc/manpath.config** (SUSE, Debian e Ubuntu).

man: lendo as páginas dos manuais

man *título* formata uma página man específica e a envia para o seu terminal com **less** (ou qualquer que seja o programa especificado em sua variável de ambiente PAGER). *título* normalmente é um comando, dispositivo ou nome de arquivo. As seções do manual são pesquisadas mais ou menos em ordem numérica, embora as seções que descrevam comandos (seções 1, 8 e 6) normalmente sejam pesquisadas em primeiro lugar.

A forma **man** *seção título* obtém uma página man a partir de uma seção específica. Portanto, **man tty** fornece a página man para o comando **tty**, e **man 4 tty** fornece a página man correspondente ao driver de terminal controlador.

man -k *palavra-chave* imprime uma lista de páginas de manuais que possuem a *palavra-chave* em suas sinopses de uma linha. Por exemplo:

```
$ man -k translate
objcopy (1)        - copy and translate object files
dcgettext (3)      - translate message
tr (1)             - translate or delete characters
snmptranslate (1)  - translate SNMP OID values into more useful information
tr (1p)            - translate characters
gettext (1)        - translate message
ngettext (1)       - translate message and choose plural form
...
```

Outras fontes de informações sobre o Linux

O mundo do Linux é bem grande e interessante. Possivelmente não seríamos capazes de mencionar cada conjunto útil de informações sobre Linux, ou até mesmo somente os principais, mas algumas fontes significativas são apresentadas na Tabela 1.3.

Tabela 1.3 Recursos Linux na Web

Site Web	Descrição
linux.slashdot.org	O braço Linux específico do gigante Slashdot de notícias técnicas
lwn.net	Linux e agregador de notícias sobre código-fonte aberto
www.freshmeat.net	Enorme catálogo de software para Linux e UNIX
www.kernel.org	Site oficial sobre kernel do Linux
www.linux.com	Depósito central de informações sobre o Linux (não oficial)
www.linux.org	Um outro depósito central de informações sobre o Linux (não oficial)
www.linuxhq.com	Compilação de correções e informações relacionadas ao kernel
www.linuxworld.com	Revista on-line da Computerworld
www.tldp.org	Linux Documentation Project (Projeto de Documentação sobre o Linux)
www.tucows.com	Biblioteca de software multiplataforma com conteúdo Linux

Não fique envergonhado de também acessar recursos UNIX gerais — grande parte dessas informações se aplica diretamente ao Linux. Uma grande quantidade de informações sobre administração de sistema se encontra disponível na Internet, sob várias formas. Por exemplo, você pode digitar perguntas de sysadmin em qualquer um dos sistemas de pesquisa populares, como o Google, o Yahoo! ou o Ask. Uma lista de outros recursos para iniciar pode ser encontrada no Capítulo 30, *Gerenciamento, diretivas e políticas*.

Muitos sites suprem diretamente as necessidades dos administradores de sistema. Eis alguns que apreciamos especialmente:

- www.ugu.com — UNIX Guru Service; uma grande quantidade de informações para administradores de sistema.
- www.stolely.com — um belo conjunto de links de recursos para administradores de sistemas.
- www.tucows.com — softwares para Windows e Macintosh, filtrados por qualidade.
- slashdot.org — "o lugar" das notícias para geeks.
- www.cpan.org — uma central de bibliotecas e scripts Perl.
- securityfocus.com — informações relacionadas com segurança de sistemas; banco de dados imenso e pesquisável sobre vulnerabilidades.

Um outro recurso útil e divertido é a página "Roseta Stone" de Bruce Hamilton no endereço:

bhami.com/rosetta.html

Ele contém informação e localização para os comandos e ferramentas usadas para várias tarefas de administração em vários sistemas operacionais.

1.7 COMO LOCALIZAR E INSTALAR SOFTWARE

A maioria das distribuições do Linux divide seu software em pacotes que podem ser instalados independentemente um do outro. Ao instalar o Linux em um computador novo, normalmente selecionamos um conjunto de pacotes "iniciais" para serem copiados para o novo sistema.

Essa arquitetura simplifica muitos aspectos da configuração de um sistema e é uma das principais vantagens do Linux em relação a versões tradicionais do UNIX. Infelizmente, esse design também complica a tarefa de escrever sobre essas distribuições, pois nunca fica realmente claro quais pacotes "fazem parte de" uma determinada distribuição. Um pacote será "incluído" se ele estiver nos CDs de instalação, mas e se não fizer parte da instalação padrão? Somente se ele estiver em cada computador que estiver executando esta distribuição? Se ele se encontrar nos CDs de "bônus" que vêm apenas na versão completa da distribuição?

Neste livro, geralmente descrevemos a instalação padrão de cada uma de nossas distribuições de exemplo. Quando dizemos que um pacote particular não está na instalação padrão, não significa necessariamente que o pacote não estará no *seu* sistema ou que não é suportado pela sua distribuição. Eis como descobrir se você o tem e, em caso negativo, como obtê-lo.

Primeiramente, utilize o comando **which** do shell para descobrir se um comando relevante já se encontra em seu caminho de pesquisa. Por exemplo, o comando a seguir revela que o servidor GNU C já foi instalado nesta máquina em **/usr/bin**:

```
$ which gcc
/usr/bin/gcc
```

Se **which** não puder localizar o comando que você procura, tente **whereis**, que pesquisa um intervalo mais amplo de diretórios de sistema e é independente do caminho de pesquisa de seu shell. Fique ciente também de que o comando **which** de alguns sistemas não mostra arquivos que você não tem permissão de executar. Por exemplo:

```
$ which ipppd
/usr/bin/which: no ipppd in (/bin:/usr/bin:/sbin:/usr/sbin)
$ whereis ipppd
ipppd: /usr/sbin/ipppd
$ ls -l /usr/sbin/ipppd
-rwx------    1 root     root       124924 Aug  3  2000 /usr/sbin/ipppd
```

Uma outra alternativa é o comando incrivelmente útil chamado **locate**, que consulta um índice pré-compilado do sistema de arquivos para localizar nomes de arquivos que atendem a um determinado padrão de busca. Ele não é específico a comandos ou pacotes, mas pode localizar qualquer tipo de arquivo. Por exemplo, se não estiver certo sobre onde encontrar o arquivo include **signal.h** (que é a principal fonte para as definições de signal do Linux), você poderia tentar:

```
$ locate signal.h
/usr/include/asm/signal.h
/usr/include/linux/signal.h
/usr/include/signal.h
/usr/include/sys/signal.h
```

O banco de dados de **locate** normalmente é gerado de novo toda noite por meio do comando **updatedb**, que é executado a partir do **cron**. Conseqüentemente, os resultados de um **locate** nem sempre refletem mudanças recentes do sistema de arquivos.

Caso saiba o nome de um pacote que está procurando, você também pode usar os utilitários de empacotamento de seu sistema para verificar diretamente a presença do pacote. Por exemplo, em um sistema Red Hat, Fedora ou SuSE, o comando a seguir verifica a presença da linguagem de script Python:

```
$ rpm -q python
python-1.5.2-27
```

Consulte o Capítulo 11, *Gerenciamento de software e configuração*, para informações adicionais sobre nossos exemplos de comandos de empacotamento das distribuições.

Se o pacote em que você está interessado parece não estar instalado, o primeiro lugar em que procurá-lo é o sistema de gerenciamento automático de pacotes da sua distribuição. Cada distribuição suporta alguma forma de sistema baseado na Internet para atualizar pacotes antigos e localizar novos. Os sistemas mais comuns são o **yum** e o APT, ambos os quais são descritos na seção *Sistemas de gerenciamento de pacotes de alto nível*, que inicia na página 168.

Por exemplo, em um sistema Debian, que utiliza o APT, o comando a seguir poderia ser utilizado para obter e instalar a versão mais recente do Python:

```
# apt-get install python
```

A maioria dos softwares Linux é desenvolvida por grupos independentes que lançam o software na forma de código-fonte. Os distribuidores Linux pegam então o código-fonte, o compilam de maneira apropriada para as convenções em uso em seus sistemas específicos e empacotam os binários resultantes. Em geral, é mais fácil instalar um pacote binário específico de uma determinada distribuição do que recuperar e compilar o código-fonte original. Entretanto, os distribuidores às vezes estão uma ou duas versões atrás da versão atual.

O fato de duas distribuições usarem o mesmo sistema de empacotamento não significa necessariamente que os pacotes para os dois sistemas sejam intercambiáveis. Por exemplo, tanto o Red Hat como o SUSE utilizam RPM, mas os layouts de seus sistemas de arquivos são ligeiramente diferentes. É sempre melhor utilizar pacotes desenvolvidos para a sua distribuição em particular caso eles estejam disponíveis.

Se tudo o mais falhar, tente procurar o pacote em um site de downloads como o freshmeat.net ou realizar uma pesquisa no Google com o nome do pacote desejado.

1.8 TAREFAS ESSENCIAIS DE UM ADMINISTRADOR DE SISTEMA

As seções a seguir resumem brevemente algumas tarefas que em geral se espera que os administradores de sistema realizem. Estas atribuições não precisam necessariamente ser realizadas por apenas uma pessoa, e em várias instalações o trabalho é dividido entre vários profissionais. Entretanto, pelo menos uma pessoa deve entender todas as tarefas e certificar-se de que alguém as está realizando.

Adicionando, removendo e gerenciando contas de usuário

Consulte o Capítulo 6 para informações adicionais sobre como adicionar novos usuários.

O administrador de sistema acrescenta contas para novos usuários e elimina contas de usuário que não estão mais ativas. O processo de adicionar e remover usuários pode ser automatizado, mas certas decisões administrativas (onde colocar o diretório inicial do usuário, em quais máquinas criar a conta etc.) têm de ser feitas antes de um novo usuário poder ser acrescentado.

Quando um usuário não precisar mais ter acesso ao sistema, a conta deste deve ser desativada. Todos os arquivos possuídos pela conta têm de ser gravados em fita e depois excluídos, para que o sistema não acumule arquivos indesejáveis ao longo do tempo.

Adicionando e removendo hardware

Consulte os capítulos 7, 28 e 23 para informações adicionais sobre esses tópicos.

Quando um novo hardware é comprado ou quando um hardware é transferido de uma máquina para outra, o sistema tem de ser configurado para reconhecer e usar esse hardware. As tarefas de suporte de hardware podem variar da simples adição de uma impressora até atribuições mais complexas como adicionar um array de discos.

Realizando backups

Consulte o Capítulo 9 para informações adicionais sobre backups.

Realizar backups é talvez a tarefa mais importante do administrador de sistema e também é normalmente a mais ignorada ou feita de maneira descuidada. Os backups são demorados e tediosos, mas são absolutamente necessários. Os backups podem ser automatizados e delegados a um subordinado, contudo ainda é tarefa do administrador de sistema certificar-se de que os backups sejam executados corretamente e dentro do cronograma (e de que a mídia resultante possa ser realmente utilizada para restaurar arquivos).

Instalando e atualizando software

Consulte o Capítulo 11 para informações adicionais sobre o gerenciamento de software.

Quando um novo software é adquirido, ele deve ser instalado e testado, normalmente em diversos tipos de hardware e sistemas operacionais. Assim que o software estiver funcionando corretamente, os usuários têm de ser informados de sua disponibilidade e localização. À medida que são lançados patches e atualizações de segurança, eles devem ser incorporados no ambiente local.

Software local deve ser instalado em um lugar que seja fácil de diferenciar do software de sistema. Essa organização simplifica a tarefa de atualizar o sistema operacional, já que o software local não será sobrescrito pela atualização.

Monitorando o sistema

Grandes instalações exigem supervisão permanente. Dentre essas atividades diárias temos de garantir que o serviço Web e de e-mail esteja funcionando corretamente, verificar arquivos de log para reconhecer antecipadamente sinais de problemas, garantir que as redes locais estejam todas conectadas de modo apropriado e ficar de olho na disponibilidade dos recursos de sistema, por exemplo, espaço em disco.

Solução de problemas

Os sistemas Linux e o hardware no qual são executados ocasionalmente falham. É tarefa do administrador fazer o papel de mecânico, diagnosticando problemas e utilizando os serviços de especialistas caso necessário. Em geral, é mais difícil diagnosticar o problema do que resolvê-lo.

Mantendo a documentação local

Consulte a página 638 para sugestões relacionadas com a documentação.

À medida que um sistema é modificado para atender às necessidades de uma empresa, ele começa a diferir do sistema básico descrito pela documentação. É obrigação do administrador de sistema documentar aspectos do sistema específicos do ambiente local. Essa tarefa inclui documentar qualquer software que é instalado, mas não vem com o sistema operacional, documentar por onde os cabos passam e como eles são construídos, manter os registros de manutenção para todo o hardware, registrar o estado dos backups e documentar diretivas e procedimentos locais.

Monitorando a segurança vigilantemente

Consulte o Capítulo 20 para informações adicionais sobre segurança.

O administrador de sistema deve implementar uma diretiva de segurança e fazer uma verificação periódica para certificar-se de que a segurança do sistema não foi violada. Em sistemas de baixa segurança, essa tarefa poderia envolver apenas algumas checagens superficiais em termos de acesso não-autorizado. Em um sistema de alta segurança, isso pode incluir uma elaborada rede de pontos de controle e programas de auditoria.

Ajudando os usuários

Embora ajudar os usuários nos seus diversos problemas raramente esteja incluído na descrição das tarefas do cargo de administrador de sistema, isso consome uma parte significativa do dia de trabalho da maioria dos administradores. Os administradores de sistemas são bombardeados com problemas que vão desde "Meu programa funcionava ontem e agora não! O que você modificou?" até "Eu derramei café no meu teclado! Posso lavá-lo com água?".

1.9 ADMINISTRAÇÃO DE SISTEMA SOB PRESSÃO

Os administradores de sistemas atuam em várias posições. No mundo real, eles normalmente são pessoas com outros serviços às quais foi pedido que dessem uma olhadinha em alguns computadores. Se estiver nessa situação, talvez queira pensar um pouquinho a respeito de aonde isso poderia eventualmente levá-lo.

Quanto mais você aprende sobre seu sistema, mais a comunidade de usuários dependerá de você. As redes invariavelmente crescem e talvez você seja pressionado a gastar uma parte maior do seu tempo na administração do sistema. Logo você perceberá que é a única pessoa dentro da empresa que sabe como executar uma série de tarefas importantes.

Assim que os companheiros de trabalho começarem a vê-lo como o administrador do sistema local, dificilmente você conseguirá se livrar dessa incumbência. Conhecemos várias pessoas que mudaram de trabalho para escapar dela. Já que muitas tarefas administrativas são intangíveis, você também descobrirá que os outros já o vêem como administrador em tempo integral, bem como um engenheiro de sistemas, escritor ou secretária em tempo integral.

Alguns administradores de má vontade tentam desencorajar as solicitações adotando uma atitude mal-humorada e fornecendo serviços de baixa qualidade. Não recomendamos essa postura; ela faz você ser malvisto e cria problemas adicionais.

Em vez disso, sugerimos que você documente o tempo que gastou na administração do sistema. Nosso objetivo deve ser o de manter o trabalho em um nível controlável e juntar evidências para poderem ser usadas no momento em que você solicitar uma diminuição na carga de tarefas administrativas. Na maioria das empresas, você precisará fazer lobby com a gerência por seis meses a um ano para ser substituído; portanto, planeje-se.

Por outro lado, talvez você descubra que gosta da área de administração de sistema e que anseia se tornar um administrador de sistema em tempo integral. Suas perspectivas para o cargo são boas. Infelizmente, seus problemas políticos provavelmente aumentarão. Consulte o Capítulo 30, *Gerenciamento, diretivas e políticas*, para obter uma noção dos aspectos políticos da administração de sistema.

A síndrome de personalidade do administrador de sistema

Uma condição clínica infeliz, porém comum, resultante do trabalho como administrador de sistema é a Síndrome de Personalidade dos Administradores de Sistemas. Essa condição normalmente se manifesta logo no início do terceiro ano da carreira de um administrador de sistema e a síndrome pode durar até ele chegar à sua aposentadoria. Os sintomas característicos incluem, mas não se limitam aos seguintes:

- Pagerfobia fantasmagórica aguda: o sentimento perturbador de que seu pager pifou (quando na realidade isso não aconteceu) e que sua noite tranqüila com a sua cara-metade está prestes a ser interrompida abruptamente, resultando em uma maratona de trabalho de 72 horas sem comida.
- Vodugrafia de usuários: a criação compulsiva de representações de bonecos-vodu do subconjunto da sua população de usuários que parece não entender que a persistente falta de planejamento por parte deles não constitui uma emergência para você.
- Fitaplexia anal idiopática: ímpeto repentino tarde da noite de montar fitas de backup para verificar se elas são realmente legíveis e foram etiquetadas de modo correto.
- Fobia de inépcia científica: o forte desejo de sacudir violentamente outros administradores de sistema que parecem jamais ter encontrado o método científico.

Podem ser usadas várias terapias curativas para tratar essa triste condição. As mais eficientes são desenvolver um bom senso de humor e a construção de uma pequena, mas bem selecionada, adega. Talvez você também precise considerar a abordagem mais meditativa de olhar um ponto qualquer no espaço e juntar os calcanhares sempre que as palavras "O servidor caiu de novo?" forem proferidas nas vizinhanças. Se tudo isso falhar, tire umas férias.

1.10 LEITURA RECOMENDADA

Os melhores recursos para administradores de sistema no reino impresso (além deste livro :-) são a série de livros da O'Reilly. A série começou com *UNIX in a Nutshell* há mais de 20 anos e agora inclui um volume separado sobre quase todos os importantes subsistemas e comandos Linux e UNIX. A série também inclui livros sobre Internet, Windows e outros tópi-

cos não-UNIX. Todos os livros são razoavelmente baratos, atualizados e focados. Tim O' Reilly interessou-se bastante pelo movimento de código-fonte aberto e organiza uma conferência, a OSCON, sobre esse tópico bem como conferências sobre outros tópicos relacionados com as tecnologias mais modernas. A OSCON ocorre duas vezes por ano, uma vez nos Estados Unidos, e outra na Europa. Visite www.oreilly.com para obter informações adicionais.

Embora uma variedade de livros introdutórios de Linux esteja no mercado, ainda não encontramos um que pudéssemos recomendar sem reservas. Em geral, é melhor você ficar com os "clássicos" do UNIX. Quase tudo o que você ler a respeito do UNIX se aplicará igualmente ao Linux.

SIEVER, ELLEN; WEBER, AARON E FIGGINS, STEPHEN. *Linux in a Nutshell. 5. ed.* Sebastopol, CA: O'Reilly Media, 2006.

LAMB, LINDA E ROBBINS, ARNOLD. *Learning the vi Editor. 6. ed.* Sebastopol, CA: O'Reilly & Associates, 1998.

POWERS, SHELLEY E PEEK, JERRY E O'REILLY, TIM e LOUKIDES, MIKE. *Unix Power Tools.* 3. ed. Sebastopol, CA: O'Reilly Media, 2003.

WALL, LARRY E CHRISTIANSEN, TOM E ORWANT, Jon. *Programming Perl.* 3. ed. Cambridge, MA: O'Reilly Media, 2000.

CHRISTIANSEN, TOM E TORKINGTON, NATHAN. *Perl Cookbook.* 2. ed. Sebastopol, CA: O'Reilly Media, 2003.

GANCARZ, MIKE. Linux and the Unix Philosophy. Boston: Digital Press, 2003.

SALUS, PETER H. The Daemon, the GNU & the Penguim. Groklaw, 2006.

Essa história fascinante do movimento de código-fonte aberto escrita pelo historiador mais conhecido do UNIX está sendo serializada em groklaw.com sob a licença Creative Commons. Atualmente, ela está quase 75% completa. O URL para o livro em si é bem longo; procure um link atual em groklaw.com ou experimente esse equivalente compactado: tinyurl.com/d6u7j.

1.11 EXERCÍCIOS

E1.1 Que comando você usaria para ler informações sobre a chamada de sistema **sync** (*não* o comando **sync**)? Como você leria uma página man referente a **sync** mantida em **/usr/local/share/man**?

E1.2 Um arquivo config de nível de sistema controla o comportamento do comando **man** no seu site? Que linhas você acrescentaria a esse arquivo caso quisesse armazenar o material local em **/doc/man**? Que estrutura de diretórios você teria de usar em **/doc/man** para torná-lo membro da hierarquia de páginas man?

E1.3 Quais são as principais diferenças entre **man** e **info**? Quais são algumas vantagens de cada um deles?

★★E1.4 Qual é o estágio atual de desenvolvimento do kernel do Linux? Quais são as questões mais importantes? Quais são alguns dos principais participantes desse mercado? Como o projeto é gerenciado?

★E1.5 Pesquise várias distribuições do Linux (use a página 4 como ponto de partida) e recomende uma distribuição para cada um dos seguintes aplicativos. Explique os motivos de sua escolha:

a) Um único usuário trabalhando em um escritório.

b) O laboratório de computação de uma universidade.

c) Um servidor Web corporativo.

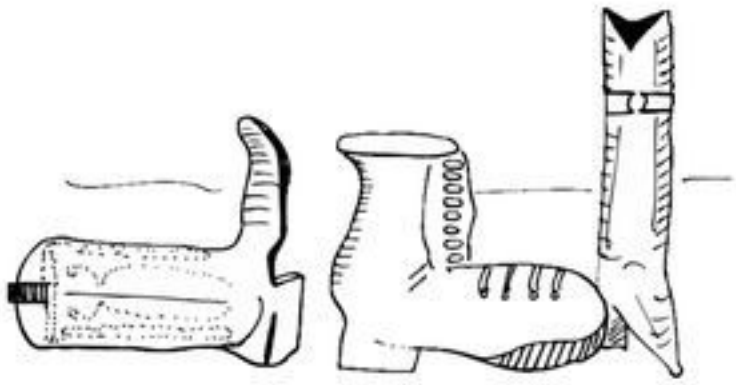
★E1.6 Suponha que você descubra que certo recurso do **httpd** Apache parece não funcionar conforme documentado no Fedora Core 5.

a) O que você deve fazer antes de relatar isso como um bug?

b) Caso decida que o bug seja real, quem você deve notificar e de que maneira?

c) Quais informações precisam ser incluídas para tornar útil um relatório de bug?

2 Inicialização e desligamento



O Linux é um sistema operacional complexo, e ligar e desligar sistemas Linux é mais complicado do que simplesmente pressionar o botão de energia. Ambas as operações têm de ser realizadas corretamente caso se queira que o sistema permaneça saudável.

Embora o processo de inicialização (*bootstrapping*) sempre tenha sido um pouco misterioso, tudo era mais simples nos dias em que os fabricantes controlavam cada aspecto do hardware e software do sistema. Agora que o Linux executa em hardware de PC, o procedimento de inicialização precisa seguir as regras para PC e lidar com diferentes possíveis configurações.

Este capítulo aparece logo no início do livro, mas se refere a assuntos que não são discutidos em detalhe até várias centenas de páginas adiante. Em particular, será útil a familiaridade com o material do Capítulo 5, *O sistema de arquivos*, do Capítulo 28, *Drivers e o kernel*, e do Capítulo 29, *Daemons*. Se seu sistema já inicializar sem qualquer problema, talvez você queira pular este capítulo e retornar a ele mais tarde.

2.1 Inicialização

Bootstrapping é o termo usado em inglês para "inicializar um computador".[1] Como os recursos normais do sistema operacional não estão disponíveis durante o processo de inicialização, o próprio computador precisa realizar os procedimentos de inicialização necessários. Durante a inicialização, o kernel é carregado na memória e começa a ser executado. Uma série de tarefas de inicialização é executada e o sistema é então tornado disponível aos usuários.

O momento da inicialização é um período de vulnerabilidade especial. Erros em arquivos de configuração, equipamentos faltantes ou não confiáveis, e sistemas de arquivos danificados podem impedir que um computador comece a funcionar. A configuração de inicialização normalmente é uma das primeiras tarefas que um administrador tem de realizar em um sistema novo. Infelizmente, também é uma das mais difíceis e requer alguma familiaridade com muitos outros aspectos do Linux.

Quando um computador é ligado, ele executa o código de boot que se encontra armazenado em ROM. Esse código, por sua vez, tenta descobrir como carregar e iniciar o kernel. O kernel investiga o hardware do sistema e depois gera o processo **init** do sistema, que sempre é PID 1.

Diversas coisas devem acontecer antes de um prompt de login poder aparecer na tela. Os sistemas de arquivos têm de ser verificados e montados e os daemons de sistema inicializados. Esses procedimentos são gerenciados por uma série de scripts shell que são executados em seqüência pelo **init**. Os scripts de inicialização são normalmente referidos como "arquivos rc" devido à maneira como são chamados; "rc" vem de "runcom" ou "run command", um remanescente histórico do sistema operacional CTSS, de cerca de 1965. O layout exato dos scripts de inicialização e a maneira como eles são executados varia entre os diferentes sistemas. Veremos os detalhes posteriormente, ainda neste capítulo.

Inicialização automática e manual

Os sistemas Linux podem ser inicializados tanto no modo automático como no modo manual. No modo automático, o sistema executa o procedimento de inicialização completo por sua conta, sem qualquer ajuda externa. No modo manual, o sistema segue o procedimento automático até um determinado ponto, mas então passa o controle a um operador antes de a maioria dos scripts de inicialização ter sido executada. Nesse ponto, o computador está no "modo monousuário". A maioria dos processos de sistema não está sendo executada e outros usuários não podem entrar no sistema.

---

1. N.T.: O termo "boot" deriva do inglês *booting* que, por sua vez, deriva do termo *bootstrapping* (de *boot straps* - cadarços de bota). Em computação, todas as três expressões têm basicamente o mesmo significado. O termo faz alusão às histórias sobre o Barão de Münchhausen, que, segundo a lenda, era capaz de se erguer do pântano (para não se molhar) puxando os cadarços das próprias botas (*pulling himself by his own boot straps*). Ou seja, o termo procura dar a imagem de um processo aparentemente impossível pelo qual o sistema se ergue (ou se coloca em funcionamento) por seus próprios esforços. (Fonte: http://pt.wikipedia.org/wiki/Bootstrap.)

Na operação do dia-a-dia, a inicialização automática é quase exclusivamente utilizada. Um procedimento de inicialização típico para uma máquina moderna é o de um usuário ligar o computador e aguardar até que o sistema entre no ar. Não obstante, é importante entender o procedimento de boot automático e saber como executar um boot manual. Normalmente, você terá de fazer um boot manual quando algum problema interferir no boot automático, por exemplo, no caso de um sistema de arquivos corrompido ou de uma interface de rede configurada de forma inadequada.

Etapas no processo de inicialização

Um processo de inicialização do Linux típico consiste em seis etapas distintas:

- Carregamento e inicialização do kernel
- Detecção e configuração de dispositivos
- Criação de threads do kernel para processos de sistema espontâneos
- Intervenção do operador (somente para inicialização manual)
- Execução dos scripts de inicialização do sistema
- Operação multiusuário

Os administradores têm pouco controle sobre a maioria dessas etapas. Efetuamos a maior parte da configuração de inicialização editando os scripts de inicialização do sistema.

Inicialização do kernel

Consulte o Capítulo 28 para informações adicionais sobre o kernel.

O kernel do Linux é por si só um programa, e a primeira tarefa de inicialização é carregar esse programa na memória de forma que ele possa ser executado. O nome de caminho do kernel depende do fornecedor, mas tradicionalmente é algo como **/unix** ou **/vmunix**. Em geral, o caminho do kernel é **/vmlinuz** ou **/boot/vmlinuz**.

O Linux implementa um processo de carregamento de duas etapas. Durante o primeiro estágio, a ROM do sistema carrega na memória um pequeno programa de inicialização a partir do disco. Esse programa prepara o ambiente para que o kernel seja carregado.

O kernel efetua testes de memória para descobrir quanta memória RAM está disponível. Algumas estruturas internas de dados do kernel são dimensionadas estaticamente, assim o kernel separa uma quantidade fixa de memória real para si ao iniciar. Essa memória é reservada para o kernel e não pode ser usada por processos em nível de usuário. O kernel imprime no console uma mensagem informando a quantidade total de memória física e a quantidade disponível para processos de usuário.

Configuração de hardware

Uma das primeiras tarefas do kernel é verificar o ambiente da máquina para ver que hardware está presente. Ao construir um kernel para seu sistema, você o informa sobre que dispositivos de hardware ele deve esperar encontrar; quando o kernel começa a executar, ele tenta localizar e inicializar cada dispositivo sobre os quais você o informou. O kernel imprime uma linha com informações obscuras sobre cada dispositivo que ele encontra. Hoje em dia, distribuições incluem kernels que funcionam na maioria das configurações de máquina, requerendo o mínimo de personalização (se alguma).

As informações de dispositivo fornecidas em tempo de configuração do kernel em geral não são especificadas completamente. Nesses casos, o kernel tenta determinar as demais informações de que ele precisa sondando o barramento em busca de dispositivos e buscando informações nos drivers apropriados. Os drivers para dispositivos que estão faltando ou que não respondem a uma investigação serão desativados. Se um dispositivo for conectado posteriormente ao sistema, também é possível carregar ou ativar um driver para ele imediatamente. Consulte o Capítulo 28, *Drivers e o kernel*, para obter mais detalhes. Mesmo se um dispositivo for conectado ao sistema mais tarde, ele não estará acessível aos processos Linux até a máquina ser reinicializada.

Threads do kernel

Assim que tiver terminado a inicialização básica, o kernel cria vários processos "espontâneos" no espaço de usuário. Eles são chamados de processos espontâneos porque não são criados por meio do mecanismo de sistema normal (**fork**); consulte a página 40 para obter mais detalhes.

Consulte a página 44 para informações adicionais sobre o ps.

O número e a natureza de processos espontâneos variam de sistema para sistema. Sob o Linux, não há nenhum PID 0 visível. **init** (sempre processo 1) é acompanhado por vários processos de rotinas de tratamento de memória e do kernel, incluindo aqueles mostrados na Tabela 2.1. Todos os processos têm PIDs de baixo número e podem ser identificados pelos nomes entre colchetes nas listagens **ps** (por exemplo, [kacpid]). Às vezes, os nomes dos processos têm uma barra e um dígito no final, como [kblockd/0]. O número indica o processador em que a thread está executando, o que talvez seja importante em um sistema com múltiplos processadores.

Tabela 2.1 Alguns processos comuns do kernel do Linux

Thread	Propósito
kjournald	Grava as atualizações do *journal* ext3 no disco[a]
kswapd	Troca os processos quando a memória física está baixa
kreclaimd	Reivindica páginas de memória que não foram utilizadas recentemente
ksoftirqd	Trata múltiplas camadas de interrupções de software
khubd	Configura dispositivos USB

a. Há um **kjournald** para cada sistema de arquivos ext3 montado.

Entre esses processos, somente **init** é realmente um processo de usuário completo com todos os recursos. Os demais são, na verdade, partes do kernel que foram arrumados para se parecerem com processos por razões de escalonamento ou de arquitetura.

Assim que os processos espontâneos forem criados, o papel do kernel na inicialização estará concluído. Entretanto, nenhum dos processos que manipulam operações básicas (como aceitação de logins) foi criado, nem a maioria dos daemons do Linux foi inicializada. Essas tarefas são realizadas (indiretamente, em alguns casos) pelo **init**.

Intervenção do operador (somente para inicialização manual)

Consulte o Capítulo 3 para informações adicionais sobre a conta root.

Se o sistema for executado no modo monousuário, um flag de linha de comando (a palavra "single") passado para o kernel notifica o **init** desse fato quando ele inicializa. **init** passa então o controle para **sulogin**, uma versão neutra, mas especial de **login** que solicita a senha da conta root.[2] Normalmente, na inicialização no modo monousuário você precisa inserir a senha de root. Se você inserir a senha correta, o sistema irá gerar um shell de root. Você pode dar um <Control-D> e digitar a senha para evitar o modo monousuário e continuar no modo multiusuário. Consulte a página 22 para mais detalhes.

Consulte o Capítulo 5 para informações adicionais sobre sistemas de arquivos e montagem.

A partir do shell monousuário, podemos executar comandos quase da mesma maneira como quando estamos conectados em um sistema completamente inicializado. Entretanto, nos sistemas SUSE, Debian e Ubuntu, normalmente apenas a partição root é montada; você precisa montar outros sistemas de arquivos manualmente para utilizar programas que não residem em **/bin**, **/sbin** ou **/etc**.

Em muitos ambientes de monousuário, inicialmente o diretório-raiz do sistema de arquivos é montado como de leitura. Se **/tmp** fizer parte do sistema de arquivos-raiz, vários comandos que usam arquivos temporários (como **vi**) não executarão. Para solucionar esse problema, você terá de iniciar sua sessão monousuário remontando / no modo leitura/gravação. O comando:

```
# mount -o rw,remount /
```

geralmente faz esse truque.

O modo monousuário do Red Hat e do Fedora é um pouco mais agressivo do que o normal. Quando você alcançar o prompt de shell, essas distribuições já tentaram montar todos os sistemas de arquivos locais. Embora isso pareça útil a princípio, pode ser problemático se você tiver um sistema de arquivos falho.

O comando **fsck** é normalmente executado durante uma inicialização automática para verificar e reparar sistemas de arquivos. Ao ativar o sistema no modo monousuário, talvez você precise executar **fsck** manualmente. Consulte a página 92 para informações adicionais sobre o **fsck**.

2. Consulte **inittab** e **sulogin** nas páginas man para informações adicionais. Infelizmente, mesmo as versões atuais do Red Hat e do Fedora não solicitam por padrão uma senha para entrar no modo monousuário.

Quando o shell monousuário é abandonado, o sistema tenta continuar inicializando no modo multiusuário.

Execução de scripts de inicialização

No momento em que o sistema estiver pronto para executar seus scripts de inicialização, já é possível reconhecer o Linux. Embora ele ainda não pareça um sistema totalmente inicializado, não resta mais nenhuma etapa "mágica" no processo de inicialização. Os scripts de inicialização são simplesmente scripts de shell comuns, e são selecionados e executados pelo **init** de acordo com um algoritmo que, embora algumas vezes tortuoso, é relativamente compreensível.

O cuidado, a alimentação e a taxonomia dos scripts de inicialização merecem uma seção própria. Isto é tratado em mais detalhes na página 23.

Operação multiusuário

Consulte a página 587 para informações adicionais sobre o processo de login.

Depois que os scripts de inicialização foram executados, o sistema estará totalmente operacional, exceto pelo fato de que ninguém pode fazer login. Para os logins serem aceitos em um determinado terminal (inclusive o console), um processo **getty** precisa estar ouvindo-o. O **init** gera diretamente esses processos **getty**, completando o processo de inicialização. O **init** também é responsável pela geração de sistemas de login gráficos como **xdm** ou **gdm**, se o sistema estiver configurado para utilizá-los.

Tenha em mente que o **init** continua a desempenhar um papel importante mesmo depois de a inicialização terminar. Nos sistemas baseados em BSD, **init** tem apenas dois estados: monousuário e multiusuário. Nos outros sistemas, **init** tem um modo monousuário e vários níveis de execução multiusuário, que determinam quais recursos do sistema estão ativados. Os níveis de execução serão descritos posteriormente, na página 23.

2.2 INICIALIZANDO PCS

Até agora vimos a descrição genérica do processo de inicialização. Vamos agora revisar os vários passos mais importantes (e complicados).

A inicialização de PCs é uma longa provação que exige uma boa fundamentação para ser entendida. Quando uma máquina é inicializada, ela começa executando código armazenado em ROMs. A localização e a natureza exatas desse código variam, dependendo do tipo de máquina que você tem. Em uma máquina projetada explicitamente para UNIX ou um outro sistema operacional proprietário, o código está tipicamente em firmware, ou seja, ele sabe como usar os dispositivos conectados à máquina, como conversar com a rede em um nível básico e como interpretar sistemas de arquivos baseados em disco. Um firmware tão inteligente é conveniente para administradores de sistema. Por exemplo, podemos simplesmente digitar o nome de arquivo de um novo kernel que o firmware saberá como localizar e ler esse arquivo.

Em PCs, o código de inicialização inicial é geralmente chamado de BIOS (Basic Input/Output System) e é extremamente simplista quando comparado ao firmware de uma máquina proprietária. Na verdade, um PC tem vários níveis de BIOS: um para a própria máquina, um para a placa de vídeo, um para a placa SCSI, se o sistema tiver um, e, às vezes, para outros periféricos como placas de rede.

A BIOS embutida sabe da existência de alguns dispositivos que residem na placa mãe, tipicamente a controladora IDE (e discos), a placa de rede, o teclado, as portas seriais e as portas paralelas. As placas SCSI em geral não estão cientes dos dispositivos conectados a elas. Felizmente, as interações complexas necessárias para que esses dispositivos funcionem conjuntamente já foram padronizadas há anos e pouca intervenção manual é exigida.

As BIOS modernas são um pouco mais espertas do que costumavam ser. Elas normalmente permitem entrar em um modo de configuração em tempo de inicialização mantendo uma ou duas teclas especiais pressionadas; a maioria das BIOS informa quais são essas teclas especiais em tempo de inicialização para que você não precise procurá-las no manual.

Em geral, a BIOS nos permite selecionar os dispositivos a partir dos quais queremos inicializar, o que soa mais promissor do que de fato acontece. Normalmente podemos especificar algo como: "Tente inicializar a partir da unidade de disquete, depois tente inicializar a partir do CD-ROM e, por fim, do disco rígido". Infelizmente, algumas BIOS se limitam a inicializar a partir da primeira unidade de CD-ROM IDE ou do primeiro disco rígido IDE. Se você foi muito, mas muito bonzinho mesmo no ano que passou, talvez Papai Noel lhe dê de presente uma BIOS que reconheça a presença de placas SCSI.

Assim que a máquina descobrir o dispositivo a partir do qual inicializar, ela tentará carregar os primeiros 512 bytes do disco. Este segmento de 512 bytes é conhecido como registro mestre de inicialização (*master boot record* – MBR). O MBR contém um programa que informa ao computador a partição de disco a partir da qual ele deve carregar um programa de inicialização secundário (a rotina de partida, ou *boot loader*). Para maiores informações sobre partições de disco do tipo PC e sobre o MBR, consulte o Capítulo 7, *Adicionando um disco*.

O MBR padrão contém um programa simples que instrui o computador a obter sua rotina de inicialização na primeira partição no disco. Tanto o Linux como o FreeBSD oferecem um MBR mais sofisticado que sabe como lidar com múltiplos sistemas operacionais e kernels.

Depois que o MBR tiver escolhido a partição a partir da qual inicializar, ele tenta carregar a rotina de inicialização específica daquela partição. A rotina de inicialização fica então responsável pelo carregamento do kernel.

2.3 UTILIZANDO ROTINAS DE INICIALIZAÇÃO: LILO E GRUB

O que seria da vida sem as opções? Duas rotinas de inicialização são utilizadas no mundo do Linux: LILO e GRUB. O LILO é a rotina de inicialização tradicional. Ele é muito estável e bem documentado, mas começa a ser ofuscado pelo GRUB, que se tornou a rotina de inicialização padrão nos sistemas Red Hat, SUSE e Fedora. De fato, as distribuições atuais do Red Hat e do Fedora simplesmente não incluem o LILO. Por outro lado, o Debian ainda utiliza o LILO como sua rotina de inicialização preferida.

GRUB: O GRand Unified Boot loader

O GRUB é particularmente popular entre os usuários que rodam uma variedade de sistemas operacionais (como Windows, OpenBSD, FreeBSD etc.) na mesma máquina ou que trabalham ativamente em desenvolvimento de kernels. O GRUB também é útil para o pessoal que muda a configuração do sistema com freqüência. Diferentemente do LILO, que deve ser reinstalado no registro de inicialização ou no MBR toda vez que é reconfigurado, o GRUB lê seu arquivo de configuração em tempo de inicialização, eliminando uma etapa administrativa fácil de esquecer.

O GRUB é instalado em uma unidade de inicialização executando **grub-install**. Esse comando utiliza como argumento o nome do dispositivo a partir do qual você irá inicializar. A maneira como o GRUB nomeia os dispositivos físicos de disco difere da convenção Linux padrão (embora o GRUB também possa utilizar nomes Linux padrão). O nome de um dispositivo GRUB é parecido com o seguinte:

```
(hd0,0)
```

O primeiro valor numérico indica o número da unidade física (começando do zero), e o segundo valor numérico representa o número da partição (mais uma vez, começando do zero). Neste exemplo, (hd0,0) é equivalente ao dispositivo Linux **/dev/hda1**. Logo, caso queira instalar o GRUB em sua unidade primária, você utilizaria o comando

```
# grub-install '(hd0,0)'
```

As aspas são necessárias para impedir que o shell tente interpretar os parênteses de seu modo particular.

Por padrão, o GRUB lê sua configuração padrão de inicialização a partir de **/boot/grub/grub.conf**. Eis um exemplo de arquivo **grub.conf**:

```
default=0
timeout=10
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.6.9-5)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.9-5 ro root=/dev/hda1
```

Esse exemplo configura somente um único sistema operacional que o GRUB inicializará automaticamente (`default=0`) caso ele não receba qualquer entrada do teclado dentro de 10 segundos (`timeout=10`). O sistema de arquivos-raiz para a configuração "Red Hat Linux" é o dispositivo GRUB (hd0,0). O GRUB carrega o kernel a partir de **/boot/vmlinuz-2.6.9-5** e exibe uma tela de abertura a partir do arquivo **/boot/grub/splash.xpm.gz** ao ser carregado.

O GRUB suporta uma interface de linha de comando poderosa e recursos para editar entradas de arquivo de configuração instantaneamente. Para entrar modo de linha de comando, digite **c** a partir da imagem de inicialização do GRUB. A partir da linha de comando você pode inicializar sistemas operacionais que não estão no **grub.conf**, exibir informações sobre o sistema e realizar testes rudimentares do sistema de arquivos. Talvez você também goste dos recursos do tipo shell da linha de comando, incluindo resultado de comandos e movimento de cursor. Qualquer coisa que possa ser feita por meio do arquivo **grub.conf** também pode ser feita por meio da linha de comando do GRUB.

Pressione a tecla <Tab> para obter uma lista rápida dos possíveis comandos. A Tabela 2.2, a seguir, lista alguns dos comandos mais úteis.

Tabela 2.2 Opções de linha de comando GRUB

Comando	Significado
reboot	Reinicialização do software do sistema
find	Localiza um arquivo em todas as partições montáveis
root	Especifica o dispositivo-raiz (uma partição)
kernel	Carrega um kernel a partir do dispositivo-raiz
help	Fornece ajuda interativa para um comando
boot	Inicializa o sistema a partir da imagem do kernel especificada

Para informações detalhadas sobre o GRUB e suas opções de linha de comando, consulte o manual oficial:
www.gnu.org/software/grub/manual/

LILO: a rotina de partida tradicional do Linux

O LILO é configurado e instalado com o comando **lilo**. O **lilo** baseia a configuração instalada no conteúdo do arquivo **/etc/lilo.conf**. Para mudar sua configuração de inicialização, simplesmente atualize o arquivo **/etc/lilo.conf** e execute novamente o comando **lilo**. Você deve reconfigurar o LILO toda vez que o processo de inicialização mudar — em particular sempre que quiser acrescentar uma nova partição de inicialização e sempre que tiver um novo kernel para ser inicializado.

Você pode instalar o LILO tanto no MBR do disco como no registro de inicialização da partição-raiz do Linux.

A seguir, um arquivo **lilo.conf** básico para um sistema Linux que tem um kernel de produção, bem como um kernel de backup:

```
boot=/dev/hda            # Coloca a rotina de partida no MBR
root=/dev/hda1           # Especifica a partição-raiz
install=/boot/boot.b
map=/boot/map
delay=20                 # 2 segundos para interrupção pelo usuário
image=/vmlinuz           # Kernel a inicializar
    label=linux          # Nome para referenicar esta entrada
    read-only
image=/vmlinuz-backup    # Entrada de backup
    label=backup
    read-only
```

Cada situação de inicialização possível tem um nome. Em tempo de inicialização, podemos informar ao LILO qual delas utilizar inserindo o nome apropriado. O primeiro nome a aparecer em **lilo.conf** torna-se o padrão.

O cenário padrão (chamado `linux`) inicializa o arquivo **/vmlinuz**. A tag `read-only` especifica que o kernel deve montar seu sistema de arquivos-raiz somente para leitura. Essa opção sempre deve estar presente; os scripts de inicialização cuidarão da remontagem da partição para leitura e gravação no momento adequado. Esse sistema também é configurado para inicializar um kernel de backup, **/vmlinuz-backup**. Sempre é uma boa idéia ter essa alternativa; uma configuração do kernel não-funcional pode levar a um sistema não-inicializável.

Executar o comando **lilo** sem argumentos irá gerar e instalar a rotina de inicialização e lhe informar quais entradas se encontram disponíveis. Ele coloca uma estrela ao lado da imagem padrão. Entretanto, se você tivesse cometido um erro no arquivo **lilo.conf**, o **lilo** normalmente não conseguiria descobrir o problema até chegar à metade do processo de instalação da rotina de inicialização. Quando isso acontece, a rotina de inicialização está em um estado confuso. *Não reinicialize* até o **lilo** executar com sucesso.

Para evitar cair nessa situação, você pode executar **lilo -t** para testar a configuração sem realmente instalá-la. Se tudo parecer OK, você poderá então executar realmente o comando **lilo**. É um tanto misteriosa a razão pela qual o **lilo** não executa esse pré-teste para o usuário por padrão.

A saída do **lilo** quando executado com o arquivo config acima é:

```
# lilo
Added linux*
Added backup
```

Quando o sistema é inicializado, o LILO imprime o seguinte prompt:

```
LILO:
```

Ele então aguarda 2 segundos (20 décimos de segundo, configurado por meio da tag `delay`), depois inicializa o kernel **/vmlinuz** e monta a primeira partição do primeiro disco IDE como partição-raiz. Você pode ver uma lista de cenários de inicialização definidos pressionando a tecla <Tab>:

```
LILO: <Tab>
linux   backup
LILO:
```

Para inicializar utilizando uma situação alternativa, simplesmente insira seu nome no prompt.

Opções do kernel

O LILO e o GRUB permitem que opções de linha de comando sejam passadas para o kernel. Em geral, essas opções modificam os valores de parâmetros do kernel, instruem o kernel a investigar dispositivos particulares, especifica o caminho para o **init** ou designa um dispositivo-raiz específico. A Tabela 2.3 mostra alguns exemplos.

Tabela 2.3 Exemplos de opções do kernel em tempo de inicialização

Opção	Significado
init=/sbin/init	Instrui o kernel a utilizar **/sbin/init** como seu programa **init**
init=/bin/bash	Inicia somente o shell **bash**; útil para recuperação de emergência
root=/dev/foo	Instrui o kernel a utilizar **/dev/foo** como o dispositivo-raiz
single	Inicializa no modo monousuário

Inicialização múltipla em PCs

Como vários sistemas operacionais rodam em PCs, é relativamente comum configurar-se uma máquina capaz de inicializar vários sistemas diferentes. Para que isso funcione, você precisa configurar uma rotina de partida para reconhecer todos os sistemas operacionais distintos presentes nos discos. Nas próximas seções, veremos alguns impedimentos comuns para sistemas de inicialização múltipla e depois veremos alguns exemplos de configuração.

Cada partição de disco tem sua própria rotina de inicialização de segundo estágio. Mas há somente um MBR. Ao definirmos uma configuração inicialização múltipla, você deve decidir qual rotina de partida será a "mestra". Para melhor ou para pior, sua escolha geralmente será determinada pelos caprichos do sistema operacional em questão. O LILO e o GRUB são as melhores opções para um sistema que tem uma partição Linux. O GRUB é melhor que o LILO em uma situação de inicialização múltipla.

Configuração de inicialização múltipla com o GRUB

Um sistema GRUB de inicialização múltipla é bem parecido com o correspondente de inicialização única. Instale todos os sistemas operacionais desejados antes de fazer alterações em **/boot/grub/grub.conf**.

Uma configuração **grub.conf** para inicializar o Windows parece diferente de uma para inicializar o UNIX ou o Linux:

```
title Windows XP
      rootnoverify (hd0,0)
      chainloader +1
```

A opção `chainloader` carrega a rotina de inicialização a partir de uma localização especificada (nesse caso, setor 1 na primeira partição da unidade IDE primária). A opção `rootnoverify` garante que GRUB não tentará montar a partição especificada. Essa opção evita que o GRUB modifique partições que ele não pode entender, como partições NTFS ou partições fora da área que GRUB pode ler.

O arquivo **grub.conf** a seguir pode inicializar o Windows XP a partir da partição 1, o Enterprise Linux a partir da partição 2 e o Fedora a partir da partição 3:

```
default=0
timeout=5
splashimage=(hd0,2)/boot/grub/splash.xpm.gz
hiddenmenu
title Windows XP
      rootnoverify (hd0,0)
      chainloader +1
title Red Hat
```

```
        root (hd0,1)
        kernel /boot/vmlinuz
title Fedora
        root (hd0,2)
        kernel /boot/vmlinuz
```

Configuração de inicialização múltipla com o LILO

Para configurar um sistema com inicialização múltipla que use o LILO no MBR (por exemplo, Linux com Windows XP), comece com a configuração padrão do LILO conforme descrito na página 20. Depois você poderá retornar e adicionar entradas para os demais sistemas operacionais ao arquivo **/etc/lilo.conf**.

Eis a entrada de **lilo.conf** necessária para inicializar o Windows na primeira partição de seu primeiro disco IDE:

```
other = /dev/hda1
label = windows
table = /dev/hda
```

Um arquivo **lilo.conf** que inicializa o Windows a partir da partição 1, o Linux a partir da partição 2 e o FreeBSD a partir da partição 3 seria parecido com o seguinte:

```
boot = /dev/hda       # instala no MBR da primeira unidade IDE
delay = 20            # Espera 2 segundos pela escolha de inicialização do usuário
default = linux       # Se o usuário não escolher nada, o linux inicializa a partir da 2ª partição
image = /boot/vmlinuz-2.6.9
   root = /dev/hda2
   label = linux
   read-only
other = /dev/hda1    # inicializa a partir da 1ª partição
   label = windows
   table = /dev/hda
other = /dev/hda3    # inicializa a partir da 3ª partição
   label = freebsd
   table = /dev/hda
```

Você precisará executar o **lilo** novamente depois de adicionar essas entradas ao **lilo.conf**. Lembre-se de executar primeiro **lilo -t** para testar o arquivo de configuração. Consulte a página 87 para informações adicionais.

Fornecedores (ou voluntários) freqüentemente distribuem "patches", isto é, correções, para distribuições do Linux, e o kernel não é uma exceção. Vulnerabilidades de segurança, bugs e recursos são adicionados regularmente. Ao contrário de outros pacotes de software, porém, os patches do kernel não são atualizados, mas sim instalados lado a lado com o kernel existente. Isso ajuda os administradores a facilmente reverter uma atualização se um patch do kernel fizer seus sistemas falharem. Com o passar do tempo, os menus de inicialização do LILO e do GRUB serão preenchidos com todas as diferentes versões do kernel. Normalmente é seguro utilizar a seleção padrão, mas esteja ciente dessa correção potencialmente simples se seu sistema não inicializar depois da correção.

2.4 INICIALIZANDO NO MODO MONOUSUÁRIO

O modo monousuário é uma excelente maneira de alterar a configuração do sistema ou realizar tarefas de manutenção sem se preocupar em afetar (ou sendo incomodado por) outros usuários. Também é um salva-vidas quando você está trabalhando em um sistema falho.

Consulte a página 23 para informações adicionais sobre os níveis de execução.

É mais comum entrar no modo monousuário passando argumentos para a rotina de inicialização. Mas, em geral, você pode entrar no modo monousuário a partir de um outro nível de execução especificando o comando **telinit 1**. Não é necessário reinicializar, a menos que você esteja depurando um problema relacionado com a inicialização.

Como uma medida preventiva contra um sistema possivelmente instável, o diretório-raiz do sistema de arquivos começa a ser montado como de leitura. Isso pode ser contraproducente para sua tarefa caso você esteja tentando solucionar um problema com um arquivo de configuração ou comando que reside no sistema de arquivos-raiz ou se precisar executar um comando que modifique arquivos. Para corrigir esse problema, remonte o sistema de arquivos-raiz no modo de leitura/gravação com:

```
# mount -o remount -w /
```

O procedimento exato para invocar o modo monousuário em tempo de inicialização difere entre o GRUB e o LILO.

Modo monousuário com o GRUB

Você não precisa utilizar a linha de comando para inicializar o modo monousuário sob o GRUB. Os autores do GRUB perceberam que as opções de inicialização são facilmente modificáveis e optaram por uma tecla 'a' como a ferramenta apropriada. Na tela de abertura do GRUB, destaque o kernel desejado e pressione 'a' para acrescentá-lo às opções de inicialização. Para inicializar no modo monousuário, adicione o flag **single** ao final das opções do kernel existentes. Um exemplo para uma configuração típica poderia ser:

```
grub append> ro root=LABEL=/ rhgb quiet single
```

Modo monousuário com o LILO

As distribuições fornecem diferentes maneiras de alcançar o prompt de comando do LILO. Se você instalou o LILO no lugar do GRUB no Red Hat, Fedora ou SUSE, escolha a opção de menu de linha de comando a partir da interface gráfica. Usuários do Debian e do Ubuntu devem pressionar e segurar a tecla Shift assim que a BIOS realizar as verificações de memória e outros autotestes de sistema.

No prompt LILO, digite o nome da configuração que você quer inicializar (conforme especificado em **lilo.conf**) seguido de **-s** ou **single**. Por exemplo, a configuração padrão que vem com o Red Hat é chamada "linux"; portanto, para inicializar essa configuração no modo monousuário, você usaria:

```
LILO: linux single
```

2.5 TRABALHANDO COM SCRIPTS DE INICIALIZAÇÃO

Após sair do modo monousuário (ou, na seqüência de inicialização automatizada, no ponto onde o shell monousuário teria sido executado), **init** executa os scripts de inicialização de sistema. Esses scripts são na verdade apenas scripts de shell corriqueiros que são interpretados por **sh** (bem, **bash**, na realidade). A localização exata, o conteúdo e a organização dos scripts variam consideravelmente de um sistema para outro.

Algumas tarefas que costumam ser realizadas nos scripts de inicialização são:

- Configurar o nome do computador
- Configurar o fuso horário
- Verificar os discos com **fsck** (somente no modo automático)
- Montar os discos do sistema
- Eliminar arquivos antigos do diretório **/tmp**
- Configurar interfaces de rede
- Inicializar daemons e serviços de rede

A maioria dos scripts de inicialização é relativamente prolixa e gera uma descrição de tudo o que eles estão fazendo. Essa loquacidade pode ser de uma ajuda tremenda caso o sistema trave no meio do caminho do processo de inicialização ou se você estiver tentando localizar um erro em um dos scripts.

Em sistemas antigos, os administradores costumavam modificar os scripts de inicialização para fazer com que eles realizassem a coisa certa para um determinado ambiente. Mas o empacotamento refinado dos softwares e as atualizações freqüentes da Internet forçaram a adoção de uma abordagem mais robusta. Hoje em dia, os sistemas acomodam diversos pequenos scripts de inicialização instalados por softwares individuais e os scripts lêem as informações locais sobre a configuração a partir de arquivos separados. Os arquivos de configuração locais normalmente assumem a forma de miniscripts **sh** que configuram os valores das variáveis de shell; essas variáveis são então consultadas pelos scripts.

init e níveis de execução

Tradicionalmente, **init** define sete níveis de execução, cada um dos quais representando um determinado complemento de serviços que o sistema deve estar executando:

- O nível 0 é o nível no qual o sistema está completamente desligado
- O nível 1 ou S representa o modo monousuário
- Os níveis 2 a 5 são níveis multiusuário
- O nível 6 é um nível de "reinicialização"

Os níveis 0 e 6 são especiais no sentido de que o sistema não pode realmente permanecer nesses níveis; o sistema é desligado ou reinicializado como efeito colateral de entrar nesses níveis. O nível de execução multiusuário normal é 2 ou 3. O nível de execução 5 é usado normalmente para os processos de login do X Windows como **xdm**. O nível de execução 4 raramente é usado, e os níveis de execução 1 e S são definidos diferentemente em cada sistema.

O modo monousuário era tradicionalmente **init** nível 1. Ele desativava todos os processos de login remotos e multiusuários e certificava-se de que o sistema estava executando apenas o complemento de software mínimo. Mas como o modo monousuário permite acesso de root ao sistema, os administradores queriam que o sistema solicitasse a senha de root sempre que fosse inicializado no modo monousuário. O nível S era criado para atender a essa necessidade: ele gera um processo que solicita a senha de root. No Linux, o nível S atende somente a essa função e não é um fim por si só.

Parece haver mais níveis de execução definidos do que são estritamente necessários ou úteis. A explicação usual para isso é que uma mesa de telefonia tinha sete níveis de execução, assim pensou-se que um sistema UNIX devesse ter pelo menos essa quantidade de níveis. O Linux na verdade suporta até dez níveis de execução, porém os níveis de 7 a 9 são indefinidos.

O arquivo **/etc/inittab** instrui **init** sobre o que fazer em cada um desses níveis. Seu formato varia de sistema para sistema, porém a idéia básica é de que **inittab** define comandos que devem ser executados (ou mantidos em execução) quando o sistema entra em cada nível.

Quando a máquina inicializa, **init** avança gradativamente do nível 0 para o nível de execução padrão, que é configurado em **/etc/inittab**. Para realizar a transição entre cada par de níveis de execução adjacentes, **init** executa as ações expressas para essa transição em **/etc/inittab**. A mesma progressão é feita na ordem inversa quando a máquina é desligada.

Infelizmente, a semântica do arquivo **inittab** é um pouco rudimentar. Para mapear os recursos do arquivo **inittab** para algo um pouco mais flexível, os sistemas Linux implementam uma camada de abstração adicional na forma de um script de "mudança de níveis de execução" (em geral, **/etc/init.d/rc**) que é chamado por meio de **inittab**. Esse script, por sua vez, executa outros scripts em um diretório dependente do nível de execução para trazer o sistema para o seu novo estado.

Atualmente, a maioria das distribuições Linux é inicializada para executar no nível 5 por padrão, o que talvez não seja adequado aos servidores que não precisam executar X. O nível de execução padrão é facilmente modificado. Esse fragmento de **inittab** de uma máquina SUSE assume o padrão de nível de execução 5:

```
id:5:initdefault:
```

Os administradores de sistema em geral não têm de lidar diretamente com **/etc/inittab** porque a interface baseada em script é adequada para a maioria dos aplicativos. No restante deste capítulo ignoraremos tacitamente o arquivo **inittab** e outras formas de "cola" que unem **init** à execução de scripts de inicialização. Apenas tenha em mente que quando dizemos que **init** executa um script tal e tal, talvez a conexão não seja tão direta.

As cópias mestras dos scripts de inicialização residem no diretório **/etc/init.d**. Cada script é responsável por um daemon ou por um determinado aspecto do sistema. Os scripts reconhecem os argumentos **start** e **stop** como significando que o serviço com o qual eles lidam deve ser inicializado ou interrompido. A maioria também reconhece **restart**, que é tipicamente o mesmo que um **stop** seguido de um **start**. Como um administrador de sistemas, você pode iniciar e interromper serviços individuais executando manualmente os seus respectivos scripts **init.d**.

Por exemplo, eis um script de inicialização simples que pode iniciar, interromper e reiniciar **sshd**:

```
#! /bin/sh
test -f /usr/bin/sshd || exit 0
case "$1" in
      start)
            echo -n "Starting sshd: sshd"
            /usr/sbin/sshd
            echo "."
            ;;
       stop)
            echo -n "Stopping sshd: sshd"
            kill `cat /var/run/sshd.pid`
            echo "."
            ;;
       restart)
           echo -n "Stopping sshd: sshd"
           kill `cat /var/run/sshd.pid`
           echo "."
           echo -n "Starting sshd: sshd"
           /usr/sbin/sshd
           echo "."
           ;;
```

```
        *)
            echo "Usage: /etc/init.d/sshd start|stop|restart"
            exit 1
            ;;
esac
```

Embora os scripts em **/etc/init.d** possam iniciar e interromper serviços individuais, o script de controle mestre executado por **init** precisa de informações adicionais sobre quais scripts executar (e com quais argumentos) para entrar em qualquer nível de execução dado. Em vez de pesquisar diretamente no diretório **init.d** quando ele conduz o sistema a um novo nível de execução, o script mestre pesquisa em um diretório chamado **rc***nível***.d**, onde *nível* é o nível de execução em que se entrará (por exemplo, **rc0.d**, **rc1.d** e assim por diante).

Os diretórios **rc***nível***.d** contêm tipicamente links simbólicos que apontam de volta para os scripts contidos no diretório **init.d**. Os nomes desses links simbólicos começam com **S** ou **K** seguidos por um número e o nome do serviço que o script controla (por exemplo, **S34named**). Quando **init** faz a transição de um nível mais baixo para um mais alto, ele executa todos os scripts que iniciam com **S** em ordem numérica ascendente com o argumento **start**. Quando **init** faz a transição de um nível de execução mais alto para um mais baixo, ele executa todos os scripts que iniciam com **K** (de "kill") em ordem numérica descendente com o argumento **stop**.

Esse esquema fornece aos administradores um controle refinado sobre a ordem em que os serviços são iniciados. Por exemplo, não faz sentido iniciar SSH antes de as interfaces de rede estarem ativas. Embora tanto network como **sshd** sejam configurados para iniciar no nível de execução 2 em um sistema Fedora, o script **network** obtém o número de seqüência 10 e o script **sshd** obtém número de seqüência 55; portanto, **network** certamente deve ser executado primeiro. Certifique-se de considerar esse tipo de dependência ao adicionar um novo serviço.

Para instruir o sistema sobre quando iniciar um daemon, temos de colocar links simbólicos no diretório apropriado. Por exemplo, para instruir o sistema a iniciar CUPS no nível de execução 2 e pará-lo elegantemente antes do desligamento, o par de links a seguir seria suficiente:

```
# ln -s /etc/init.d/cups /etc/rc2.d/S80cups
# ln -s /etc/init.d/cups /etc/rc0.d/K80cups
```

A primeira linha instrui o sistema a executar o script de inicialização **/etc/initi.d/sshd** como uma das últimas coisas a fazer ao entrar no nível de execução 2 e a executar o script com o argumento **start**. A segunda linha instrui o sistema a executar **/etc/init.d/sshd** relativamente cedo ao desligar o sistema e a executar o script com o argumento **stop**. Alguns sistemas tratam o desligamento e a reinicialização de maneira diferente; portanto, temos de colocar um link no diretório **/etc/rc6.d**, bem como nos certificar de que o daemon desliga apropriadamente quando o sistema é reinicializado.

Os scripts de inicialização do Red Hat e do Fedora

Os scripts de inicialização do Red Hat e do Fedora estão, historicamente, no lado confuso. Embutido no código, talvez você veja uma variedade de comentários como este:

```
# Yes, this is an ugly, but necessary hack (Sim, este um hack horrível, mas necessário)
```

Em cada nível de execução, **init** invoca o script **/etc/rc.d/rc** com o novo nível de execução como um argumento. **/etc/rc.d/rc** geralmente executa no modo "normal", no qual ele simplesmente faz seu trabalho, mas também pode executar no modo de "confirmação", no qual ele pergunta antes de executar cada script de inicialização individual.

O Red Hat e Fedora têm um comando **chkconfig** para ajudá-lo a gerenciar os serviços. Esse comando adiciona ou remove scripts de inicialização do sistema, gerencia os níveis de execução em que eles operam e lista os níveis de execução para os quais um script está atualmente configurado. Consulte **man chkconfig** para as informações sobre o uso dessa ferramenta simples e útil.

O Red Hat também possui um script **rc.local** muito parecido com aquele encontrado nos sistemas BSD. **rc.local** é o último script executado como parte do processo de inicialização. Historicamente, **rc.local** foi sobrescrito pelo pacote **initscripts**. Isso, porém, mudou e agora é seguro adicionar suas próprias personalizações de inicialização aqui.

Eis um exemplo de uma sessão de inicialização do Red Hat:

```
[kernel information]
INIT: version 2.85 booting
Setting default font (latarcyrhev-sun16):                    [  OK  ]
           Welcome to Red Hat Linux
    Press 'I' to enter interactive startup.
Starting udev:                                                [  OK  ]
Initializing hardware... storage network audio done
Configuring kernel parameters:                                [  OK  ]
```

```
Setting clock  (localtime): Tue Mar 29 20:50:41 MST 2005:      [  OK  ]
...
```

Assim que vir a mensagem "Welcome to Red Hat Linux", você pode pressionar a tecla 'i' para entrar no modo de confirmação. Infelizmente o Red Hat não dá nenhuma confirmação de que você pressionou a tecla correta. Ele prossegue alegremente para montar os sistemas de arquivos locais, ativar as partições de swap, carregar os mapas de teclas e localizar os seus módulos do kernel. Somente depois de passar para o nível de execução 3 é que ele realmente começará a pedir-lhe informações:

```
           Welcome to Red Hat Enterprise Linux WS
    Press 'I' to enter interactive startup.
Starting udev:                                              [  OK  ]
Initializing hardware... storage network audio done
Configuring kernel parameters:                              [  OK  ]
setting clock  (localtime): tue mar 29 20:50:41 mst 2005:   [  OK  ]
Setting hostname rhel4:                                     [  OK  ]
Checking root filesystem
/dev/hda1: clean, 73355/191616 files, 214536/383032 blocks  [  OK  ]
Remounting root filesystem in read-write mode:              [  OK  ]
Setting up Logical Volume Management:                       [  OK  ]
Checking filesystems
Mounting local filesystems:                                 [  OK  ]
Enabling local filesystem quotas:                           [  OK  ]
Enabling swap space:                                        [  OK  ]
INIT: Entering runlevel: 3
Entering interactive startup
Start service kudzu (Y)es/(N)o/(C)ontinue? [Y]
```

A inicialização interativa e o modo monousuário começam no mesmo ponto no processo de inicialização. Quando o processo de inicialização é tão dividido que você não pode atingir esse ponto de maneira segura, você pode utilizar um disquete ou um CD-ROM de segurança para fazer a inicialização.

Também podemos passar o argumento **init=/bin/sh** para o LILO a fim de enganá-lo e fazê-lo executar um shell monousuário antes mesmo de **init** iniciar.[3] Se adotar essa conduta, você terá de fazer toda a faxina de inicialização manualmente, inclusive dar o comando **fsck** manualmente e montar os sistemas de arquivos locais.

Grande parte da configuração do processo de inicialização do Red Hat deve ser realizada por meio da manipulação dos arquivos de configuração em **/etc/sysconfig**. A Tabela 2.4 resume a função de alguns itens populares no diretório **/etc/sysconfig**.

Tabela 2.4 Arquivos e subdiretórios do diretório /etc/sysconfig do Red Hat

Arquivo/Diretório	Função ou conteúdo
clock	Especifica o tipo de relógio que o sistema possui (quase sempre UTC)[a]
console	Um diretório misterioso que está sempre vazio
httpd	Determina qual modelo de processamento Apache utilizar
hwconf	Contém todas as informações de hardware do sistema. Usado pelo Kudzu
i18n	Contém as configurações locais do sistema (formatos de data, idiomas etc.)
init	Configura a maneira como as mensagens dos scripts de inicialização são exibidas
keyboard	Configura o tipo de teclado (utilize "us" para o teclado norte-americano padrão de 101 teclas)
mouse	Configura o tipo de mouse. Utilizado pelo X e **gpm**
network	Configura opções de rede globais (nome do host, gateway, encaminhamento etc.)
network-scripts	Contém scripts acessórios e arquivos de configuração de rede
sendmail	Configura as opções para **sendmail**

a. Se você tem múltiplos sistemas em seu PC, cuidado como o fuso horário do relógio deve ser configurado.

3. Certa vez tínhamos um arquivo de mapa de teclas corrompido e como esse tipo de arquivo é carregado até mesmo no modo monousuário, o monousuário era inútil. Configurar **init=/bin/sh** era a única maneira de inicializar o sistema para um estado monousuário utilizável para corrigir o problema. Isso também pode ser um truque útil em outras situações.

Vários dos itens contidos na Tabela 2.4 merecem comentários adicionais:

- O arquivo **hwconf** contém todas as informações de hardware. O irritante serviço Kudzu verifica se adicionamos ou eliminamos qualquer dispositivo de hardware e nos pergunta o que fazer em relação às modificações. Talvez você queira desativar esse serviço em um sistema de produção, pois ele retarda o processo de inicialização sempre que detecta uma mudança na configuração de hardware, resultando em 30 segundos adicionais de tempo de inatividade para cada mudança de hardware feita.
- O diretório **network-scripts** contém material adicional relacionado com a configuração de rede. As únicas coisas que devemos precisar modificar são os arquivos chamados **ifcfg**-*interface*. Por exemplo, **network-scripts/ifcfg-eth0** contém os parâmetros de configuração para a interface eth0. Ele configura o endereço IP da interface e as opções de rede. Consulte as páginas 199 e 200 para informações adicionais sobre como configurar interfaces de rede.
- O arquivo **sendmail** contém duas variáveis: `DAEMON` e `QUEUE`. Se a variável `DAEMON` estiver configurada como `yes`, o sistema inicia **sendmail** no modo daemon (**-bd**) quando o sistema é inicializado. `QUEUE` informa **sendmail** de quanto tempo ele deve esperar entre execuções de fila (**-q**); o padrão é uma hora.

Scripts de inicialização do SUSE

Embora o sistema de inicialização do SUSE seja semelhante àquele do RHEL e do Fedora, os scripts de inicialização do SUSE são uma das áreas em que ele realmente ofusca as outras variantes do Linux. Os scripts do SUSE são bem organizados, robustos e bem documentados. O pessoal que faz a manutenção dessa parte do sistema operacional merece uma medalha de ouro.

Como no Red Hat e no Fedora, **init** invoca o script **/etc/init.d/rc** em cada nível de execução, fornecendo o novo nível de execução como um argumento. Scripts específicos de pacotes residem no diretório **/etc/init.d** e seus arquivos de configuração residem em **/etc/sysconfig**. Uma excelente introdução ao processo de inicialização do SUSE pode ser encontrada em **/etc/init.d/README**.

Embora tanto SUSE como RHEL/Fedora concentrem seus arquivos de configuração de inicialização em **/etc/sysconfig**, os arquivos específicos dentro desse diretório são bem diferentes. (Mas os arquivos do SUSE geralmente são bem comentados.) As opções são invocadas configurando variáveis de shell, e essas variáveis são então referenciadas por scripts dentro de **/etc/init.d**. Alguns subsistemas requerem mais configuração que outros e aqueles que precisam de múltiplos arquivos de configuração têm subdiretórios privados, como o diretório **sysconfig/network**.

O arquivo **windowmanager** é um exemplo típico do diretório **sysconfig**:

```
## Path:          Desktop/Window manager
## Description:
## Type:          string(kde,fvwm,gnome,windowmaker)
## Default:       kde
## Config:        profiles,kde,susewm
#
# Here you can set the default window manager (kde, fvwm, ...)
# changes here require at least a re-login
DEFAULT_WM="kde"

## Type:          yesno
## Default:       yes
#
# install the SUSE extension for new users
# (theme and additional functions)
#
INSTALL_DESKTOP_EXTENSIONS="yes"
```

Cada variável é precedida por informações de configuração legíveis pelo YaST-readable[4] e por uma descrição prolixa do propósito da variável. Por exemplo, no arquivo **windowmanager**, a variável DEFAULT_WM configura o gerenciador de janelas da área de trabalho utilizado pelo X.

O SUSE fez um trabalho particularmente interessante com os arquivos de configuração de rede localizados no subdiretório **/etc/sysconfig/network**. Esse diretório contém tanto arquivos de configuração globais (que configuram opções pertinentes a todas as interfaces de rede) como arquivos específicos de uma rede. Por exemplo, o arquivo **network/routes** contém informações de roteamento globais. Em uma instalação SUSE típica, seu conteúdo talvez se pareça com isto:

4. O YaST é um utilitário gráfico de configuração específico do SUSE que mantém muitos aspectos de um sistema SUSE. Consulte a página 97 para informações adicionais.

```
# Destination     Dummy/Gateway         Netmask       Device
default           192.168.10.254        0.0.0.0       eth0
```

Rotas que só devem estar presentes quando uma interface particular está ativa e em execução podem ser especificadas em um arquivo chamado **ifroute-*nomeDaInterface***. Por exemplo, em uma interface chamada eth1, o arquivo pode se chamar **ifroute-eth1** e seu conteúdo talvez seja:

```
# Destination     Dummy/Gateway         Netmask       Device
10.10.0.0/24      10.10.0.254
```

A máscara de rede e o dispositivo podem ser especificados se você desejar, mas os scripts de inicialização vão inferir os valores corretos.

O SUSE também inclui um comando **chkconfig** para gerenciar os scripts de inicialização. Ele é inteiramente diferente da versão fornecida pelo Red Hat, mas é uma ferramenta eficiente e deve ser utilizada no gerenciamento manual de scripts.

Independentemente de optarmos por utilizar o YaST ou **chkconfig** ou por manter nossos scripts de inicialização manualmente, é uma boa idéia pesquisar dentro de **/etc/sysconfig** e considerar seu conteúdo:

Uma sessão de inicialização SUSE típica se parece com isto:

```
[kernel information]
INIT: version 2.85 booting
System Boot Control: Running /etc/init.d/boot
Mounting /proc filesystem                                    done
Mounting sysfs on /sys                                       done
Mounting /dev/pts done
Boot logging started on /dev/tty1(/dev/console) at Tue Mar 29 14:04:12 2005
Mounting shared memory FS on /dev/sh                         done
Activating swap-devices in /etc/fstab...
Adding 1052248k swap on /dev/hda2. Priority:42 extents:1     done
Checking root file system...
...
```

Scripts de inicialização do Debian e do Ubuntu

Se o SUSE é o exemplo máximo de um plano bem projetado e bem executado para gerenciar scripts de inicialização, o Debian é exatamente o oposto. Os scripts Debian são frágeis, não-documentados e vergonhosamente inconsistentes. Infelizmente, parece que a falta de uma maneira padronizada de configurar scripts resultou em um caos nesse caso. Debian malcriado!

Em cada nível de execução, **init** chama o script **/etc/init.d/rc** com o novo nível de execução na forma de um argumento. Cada script é responsável por localizar as suas próprias informações de configuração que podem estar na forma de outros arquivos em **/etc**, **/etc/default**, um outro subdiretório de **/etc** ou em algum outro ponto do script em si.

Se estiver procurando o nome de host do sistema, ele se encontra armazenado em **/etc/hostname**, que é lido pelo script **/etc/init.d/hostname.sh**. A interface de rede e os parâmetros de gateway padrão estão armazenados em **/etc/network/interfaces**, que é lido pelo comando **ifup** chamado em **/etc/init.d/networking**. Algumas opções de rede também podem ser configuradas em **/etc/network/options**.

O Debian e o Ubuntu têm um tipo de programa de gerenciamento de script de inicialização clandestino na forma de **update-rc.d**. Embora sua página man não recomende a utilização interativa, achamos que ele é um substituto útil, mesmo que menos amigável, ao **chkconfig**. Por exemplo, para iniciar **sshd** nos níveis de execução 2, 3, 4 e 5 e pará-lo nos níveis 0, 1 e 6, utilize:

```
$ sudo /usr/sbin/update-rc.d sshd start 0123 stop 456
```

2.6 REINICIALIZANDO E DESLIGANDO

O buffer de sistemas de arquivos do Linux muda na memória e atualiza essas informações no disco apenas esporadicamente. Esse esquema torna o I/O de disco mais rápido, porém também torna o sistema de arquivos mais suscetível a perda de dados quando o sistema é interrompido rudemente.

As máquinas UNIX e Linux tradicionais eram muito delicadas em relação a seu desligamento. Os sistemas modernos se tornaram menos sensíveis (especialmente quando você utiliza um sistema de arquivos robusto como o ext3fs), mas sempre é uma boa idéia desligar a máquina de maneira correta sempre que possível. O desligamento inadequado pode resultar em qualquer coisa, desde problemas sutis e insidiosos até uma grande catástrofe.

Em sistemas operacionais voltados ao mercado de consumo, reinicializar o sistema operacional é um primeiro tratamento apropriado para quase todos os problemas. Em um sistema Linux, é melhor pensar duas vezes antes de reiniciar o sistema. Os problemas no Linux tendem a ser mais sutis e mais complexos; portanto, reinicializar cegamente é eficiente apenas em uma pequena porcentagem de casos. Os sistemas Linux também levam um longo tempo para inicializar e vários usuários poderão ser incomodados.

Talvez seja necessário reinicializarmos o sistema ao adicionarmos um dispositivo de hardware novo ou quando algum dispositivo de hardware existente se torna tão problemático que ele não é capaz de se reinicializar. Se modificarmos um arquivo de configuração que é usado somente em tempo de inicialização, temos de reinicializar para que nossas alterações entrem em efeito. Se o sistema estiver tão travado que você não consegue fazer o login para realizarmos um diagnóstico apropriado do problema, obviamente você não tem nenhuma alternativa a não ser reinicializar o sistema.

Sempre que modifica um script de inicialização, você deve reinicializar simplesmente para certificar-se de que o sistema é inicializado com sucesso. Se descobrir um problema várias semanas mais tarde, provavelmente você não irá se lembrar dos detalhes das suas alterações mais recentes.

Diferentemente da inicialização, que pode ser feita essencialmente de uma única maneira, o desligamento ou reinicialização pode ser feito de várias maneiras:

- Desligando a energia
- Utilizando o comando **shutdown**
- Utilizando os comandos **halt** e **reboot**
- Utilizando **telinit** para alterar os níveis de execução de **init**
- Utilizando o comando **poweroff** para informar o sistema para desligar a energia

Desligando a energia

Até mesmo em sistemas desktop, desligar a energia não é uma boa maneira de efetuar o processo de desligamento do sistema. Você pode perder dados e corromper os sistemas de arquivos.

Muitas máquinas têm um recurso conhecido como "*soft power*", que significa que, ao pressionarmos o botão de energia, a máquina executa alguns comandos para executar uma seqüência de desligamento apropriada. Se não tiver certeza de que sua máquina fornece esse recurso, não pressione o botão de energia para descobrir! É melhor que você mesmo faça a seqüência de desligamento apropriada.

Dito isso, porém, desligar a energia não é o fim do mundo. Não há problemas em desligar a energia em uma emergência se você não tiver tempo para desligar as máquinas da maneira adequada. Salas de máquinas antigas normalmente tinham um botão de pânico que desligava tudo de uma só vez. Em uma ocasião, nossos administradores de sistema o dispararam acidentalmente com um arremesso malfeito de uma bola de futebol americano.

shutdown: a maneira gentil de desligar um sistema

shutdown é a maneira mais completa, segura e ajuizada de iniciar o processo de parada ou de reinicialização ou de retornar ao modo monousuário.

Você poderá pedir para o comando **shutdown** aguardar um pouco antes de desligar o sistema. Durante o período de espera, **shutdown** envia mensagens para usuários conectados em intervalos progressivamente menores, alertando-os do iminente tempo de inatividade. Por padrão, os alertas simplesmente dizem que o sistema está sendo desligado e informa o tempo que resta até o desligamento; você também pode fornecer uma pequena mensagem personalizada. Essa mensagem poderia informar a razão por que o sistema está sendo desligado e fornecer uma estimativa do tempo em que ele ficará fora do ar (por exemplo, "o sistema estará de volta às 11h00"). Os usuários não poderão entrar no sistema quando um **shutdown** for iminente, mas verão essa mensagem caso ela tenha sido especificada.

shutdown permite especificar se a máquina deve parar (**-h**) ou reinicializar (**-r**) depois que o processo de desligamento é concluído. Também podemos especificar se queremos forçar a execução do comando **fsck** nos discos após uma reinicialização (**-F**) ou não (**-f**). Por padrão, o Linux pula as verificações **fsck** sempre que os sistemas de arquivos tiverem sido desmontados apropriadamente.

Por exemplo, um comando **shutdown** que lembre os usuários de uma manutenção programada e pare o sistema às 09h30 seria parecido com o seguinte:

```
# shutdown -h 09:30 "O sistema será colocado fora do ar para manutenção agendada. O período de
    inatividade estimada é de 1 hora."
```

Também é possível especificar um tempo de inatividade relativo. Por exemplo, o comando a seguir desligaria a máquina depois de 15 minutos:

```
# shutdown -h +15 " O sistema será colocado fora do ar para reparo de disco de emergência."
```

halt: uma maneira mais simples de desligar um sistema

O comando **halt** realiza as tarefas essenciais necessárias para desligar o sistema. Ele é chamado por **shutdown -h**, mas também pode ser usado por si próprio. **halt** registra o desligamento em log, elimina processos não-essenciais, executa a chamada de sistema **sync** (chamado por e equivalente ao comando **sync**), aguarda as gravações do sistema de arquivos terminarem e depois pára o kernel.

halt -n impede a chamada de **sync**. Ele é utilizado por **fsck** depois dele reparar a partição-raiz. Se **fsck** não utilizar **-n**, o kernel poderia sobrescrever os reparos feitos pelo **fsck** com versões antigas do superbloco que foram armazenadas na memória cache.

reboot: reinicialização rápida e suja

reboot é quase idêntico a **halt**, mas faz a máquina reinicializar em vez de parar. **reboot** é chamado por **shutdown -r**. Como **halt**, ele suporta os flags **-n**.

telinit: modificando o nível de execução de init

Em sistemas com o poderoso **init** de múltiplos níveis, você pode utilizar **telinit** a fim de direcionar **init** para a um nível de execução específico. Por exemplo,

```
# telinit S1
```

leva o sistema ao modo monousuário no Solaris e no HP-UX. Sob o Red Hat, você precisa alterar para o nível de execução 1; do contrário, você só inicia um shell de raiz sem realmente alterar o nível de execução:

```
# telinit 1
```

leva o sistema para o modo monousuário.

Ao usarmos **telinit**, não recebemos as mensagens de alerta amistosas ou o período "extra" que temos no caso de **shutdown**; portanto, na maioria das vezes você vai querer evitá-lo. **telinit** é mais útil para testar mudanças feitas no arquivo **inittab**. O comando

```
# shutdown -i1
```

leva o sistema ao mesmo estado com mais elegância. **telinit** é mais útil para testar alterações no arquivo **inittab**. **telinit -q** faz com que **init** releia o arquivo **inittab**.

poweroff: pedindo para o Linux desligar o sistema

O comando **poweroff** é idêntico ao comando **halt**, exceto pelo fato de que depois que o Linux foi desativado, **poweroff** envia uma solicitação ao sistema de gerenciamento de energia (em sistemas que possuem um) para desligar a alimentação de energia principal do sistema. Esse recurso facilita o desligamento de máquinas remotas (por exemplo, durante uma tempestade com raios).

Infelizmente, não há nenhum comando **poweron** correspondente. A razão para essa aparente omissão é deixada como exercício para o leitor.

2.7 EXERCÍCIOS

E2.1 Por que é importante executar o comando **lilo -t** antes de instalar a rotina de inicialização do LILO? Como se inicializa um kernel cujo nome seja diferente de **vmlinuz**?

E2.2 Por que um sistema Linux não deve ser desligado com o botão de energia do computador? Quais são alguns métodos alternativos?

E2.3 Utilize a linha de comando do GRUB para inicializar um kernel que não está em **grub.conf**.

★★ E2.4 Explique o conceito de níveis de execução. Liste os níveis de execução definidos no Linux e descreva brevemente cada um. Qual a relação entre o nível de execução 1 e o nível de execução S?

★ E2.5 Crie um script de inicialização para ativar o daemon "foo" (**/usr/local/sbin/foo**), um serviço de rede. Mostre como agregá-lo ao sistema para iniciar automaticamente em tempo de inicialização.

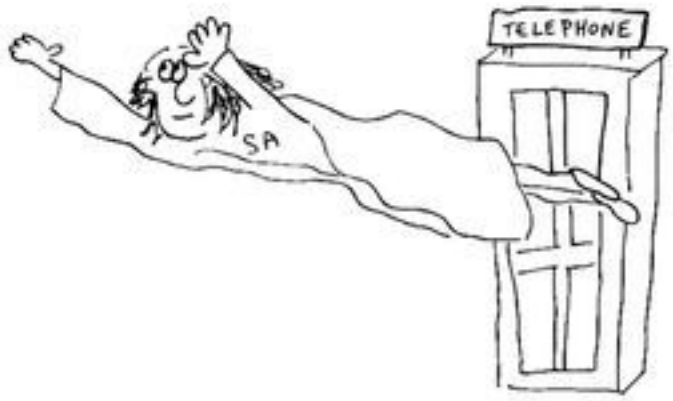
★ E2.6 Obtenha e instale o programa **mactime** de Dan Farmer e Wietse Venema (ele é parte do conjunto de ferramentas TCT). Execute **mactime** para criar um banco de dados inicial dos registros de data/hora associados aos seus arquivos de sistema. Reinicialize a máquina. Execute **mactime** de novo e determine quais arquivos foram modificados devido à reinicialização da máquina. Que arquivos foram acessados, mas não modificados? (Exige acesso root.)

★★ E2.7 Se um sistema se encontra no nível de execução 4 e você executar o comando **telinit 1**, que etapas serão seguidas por **init**? Qual será o resultado final do comando?

★★ E2.8 Desenhe um gráfico de dependência que ilustre quais daemons têm de ser inicializados antes de outros daemons em seu sistema Linux.

★★ E2.9 Liste na ordem os passos utilizados para criar um sistema com múltiplos sistemas operacionais que inclua o Linux e o Windows. Utilize o GRUB e a rotina de inicialização do Windows.

3 Poderes da conta root



Todos os processos e arquivos em um sistema Linux pertencem a uma conta de usuário em particular. Os demais usuários não podem acessar esses objetos sem a permissão do proprietário, de modo que essa convenção ajuda a proteger os usuários contra as ações indevidas cometidas por outros, sejam elas intencionais ou acidentais.

Os arquivos e processos de sistema geralmente pertencem a um usuário fictício chamado "root", também conhecido como superusuário. Como acontece com qualquer conta, a propriedade de root é protegida contra a interferência dos outros usuários. Para fazer alterações administrativas, você precisará usar um dos métodos de acesso à conta root descritos neste capítulo.

A conta root possui várias propriedades "mágicas". Ela pode atuar como o proprietário de qualquer arquivo ou processo. Root também pode executar várias operações especiais que estão fora do alcance dos demais usuários. Essa conta é poderosa e, em mãos negligentes ou maliciosas, potencialmente perigosa.

Este capítulo apresenta aos administradores de sistemas os fundamentos do acesso de superusuário. O Capítulo 20, *Segurança*, descreve como impedir acesso de superusuário indesejado e embaraçoso por parte de terceiros. O Capítulo 30, *Gerenciamento, diretivas e políticas*, cobre os aspectos políticos e administrativos relevantes.

3.1 Posse de arquivos e processos

Todo arquivo tem tanto um usuário proprietário como um "grupo proprietário". O proprietário do arquivo desfruta um privilégio especial que não é compartilhado com todo mundo no sistema: a capacidade de modificar as permissões do arquivo. Em particular, o proprietário pode configurar as permissões em um arquivo de maneira tão restritiva que ninguém mais poderá acessá-lo.[1] Discutimos o assunto permissões sobre arquivos no Capítulo 5, *O sistema de arquivos*.

Consulte a página 69 para informações adicionais sobre grupos.

Embora o proprietário de um arquivo seja sempre uma pessoa individual, muitas pessoas podem ser proprietários de um arquivo, desde que todas façam parte de um único grupo Linux. Grupos são tradicionalmente definidos no arquivo **/etc/group**, mas hoje em dia informações sobre grupos são mais comumente armazenadas em um servidor NIS ou LDAP na rede; veja o Capítulo 17, *Compartilhando arquivos de sistema*, para detalhes.

O proprietário de um arquivo deve especificar o que os grupos proprietários podem fazer ou não com ele. Esse esquema permite que arquivos possam ser compartilhados por membros do mesmo projeto. Por exemplo, usamos um grupo para controlar o acesso a arquivos-fonte para o site Web www.admin.com.

Ambas as propriedades de um arquivo podem ser visualizadas por meio de **ls -l** *nomeDoArquivo*. Por exemplo:

```
$ ls -l /staff/scott/todo
-rw------- 1      scott  staff  1258   Jun 4 18:15   /staff/scott/todo
```

Esse arquivo é de propriedade do usuário "scott" e do grupo "staff".

O Linux, na verdade, controla os proprietários e grupos como números, e não como nomes na forma de texto. No caso mais básico, os números de identificação (*identification numbers* – UIDs) são mapeados para nomes de usuário no arquivo **/etc/passwd** e os números de identificação dos grupos (*group identification numbers* – GIDs) são mapeados para nomes de grupos em **/etc/group**. Os nomes baseados em texto que correspondem a UIDs e GIDs só são definidos para a conveniência dos usuários humanos do sistema. Quando comandos com **ls** querem exibir informações de posse em um formato legível, eles têm de pesquisar cada nome no arquivo ou banco de dados apropriado.

O proprietário de um processo pode enviar os sinais de processo (consulte a página 40) e você também pode reduzir (degradar) a prioridade de agendamento do processo. Na verdade, os processos têm até pelo menos sete identidades associadas com eles: um UID real, efetivo e salvo; um GID real, efetivo e salvo; e, sob o Linux, um "UID de sistema de arquivos" que só é utilizado para determinar as permissões de acesso a arquivos. Falando de maneira genérica, os números reais são usados para fins de contabilidade e os

1. De fato, as permissões podem ser configuradas de maneira tão restritiva que mesmo o proprietário de um arquivo não consegue acessá-lo.

números efetivos são utilizados para determinação das permissões de acesso. Os números reais e efetivos são normalmente os mesmos.

IDs salvos não têm nenhum efeito direto. Eles permitem que programas "guardem" um ID inativo para uso posterior, facilitando assim a utilização parcimoniosa dos privilégios aprimorados. O UID de sistema de arquivos geralmente é explicado como um detalhe de implementação do NFS e é, em geral, o mesmo do UID efetivo.

Consulte a página 58 para informações adicionais sobre os bits de permissão.

Embora normalmente não seja possível para um processo alterar suas credenciais de posse, há uma situação especial em que os IDs de grupos e usuários efetivos podem ser modificados. Quando o kernel executa um arquivo executável que tem seus bits de permissão "setuid" ou "setgid" configurados, ele altera o UID ou o GID efetivo do processo resultante para o UID ou GID do arquivo que contém a imagem do programa em vez do UID e GID do usuário que executou o comando. Os privilégios do usuário são portanto "promovidos" somente para a execução desse comando específico.

O recurso setuid do Linux permite que programas sejam executados por usuários comuns fazendo uso da conta root de uma forma limitada e fortemente controlada. Por exemplo, o comando **passwd** que os usuários executam para modificar suas senhas de login é um programa setuid. Ele modifica o arquivo **/etc/shadow** (ou **/etc/passwd**) de uma maneira bem definida e depois finaliza. Obviamente, até mesmo essa tarefa limitada corre o risco de mau uso; desse modo **passwd** requer que os usuários provem que conhecem a senha da conta atual antes de ele concordar em fazer a modificação solicitada.

3.2 O SUPERUSUÁRIO

A característica que define uma conta root é o seu UID igual a 0. O Linux não o impede de modificar o nome de usuário nesta conta ou de criar contas adicionais cujos UIDs sejam 0, mas as duas ações não são recomendadas. Essas mudanças apresentam uma tendência de criar inadvertidamente brechas na segurança do sistema. Elas também provocam confusão quando outras pessoas têm de lidar com a maneira estranha como você configurou o sistema.

Tradicionalmente o UNIX permite ao superusuário (isto é, qualquer processo cujo UID efetivo seja 0) executar qualquer operação válida em qualquer arquivo ou processo[2]. Além disso, algumas chamadas de sistema (solicitações ao kernel) podem ser executadas somente pelo superusuário. Alguns exemplos dessas operações restritas são:

- Modificar o diretório-raiz de um processo com **chroot**
- Criar arquivos de dispositivos
- Configurar o relógio do sistema
- Aumentar os limites de uso de recursos e as prioridades de processos[3]
- Definir o nome de host (hostname) do sistema
- Configurar interfaces de rede
- Abrir portas de rede privilegiadas (aquelas com números abaixo de 1.024)
- Desligar o sistema

Um exemplo dos poderes do superusuário é a capacidade de um processo pertencente a root modificar o seu próprio UID e GID. O programa **login** e seus equivalentes de sistema de janelas são um caso pertinente; o processo que nos solicita a senha ao fazermos a entrada em um sistema é executado inicialmente como root. Se a senha e o nome de usuário introduzidos forem legítimos, o programa de login altera seu UID e GID para o nosso UID e GID e inicia nosso ambiente de usuário. Depois que um processo de root modificou seus direitos de posse para se tornar um processo de usuário comum, ele não pode recuperar seu estado privilegiado anterior.

Teoricamente, os sistemas Linux são capazes de subdividir os privilégios da conta root de acordo com o padrão POSIX para "capacidades". Por várias razões, incluindo problemas com a implementação atual, essa facilidade não é tão útil nem relevante aos administradores de sistema quanto inicialmente poderia parecer. Para mais comentários sobre capacidades, veja a discussão na página 472.

---

2. "Válida" é um eufemismo significativo nesse caso. Certas operações (como executar um arquivo no qual o bit de permissão de execução não está configurado) são proibidas até mesmo para o superusuário.

3. Na versão 2.6.12 do kernel, um novo limite de recursos permite que os usuários, além do superusuário, aumentem as prioridades de um processo se o administrador de sistema permite isso.

3.3 ESCOLHENDO UMA SENHA PARA ROOT

Consulte a página 477 para informações adicionais sobre quebra de senhas.

A senha de root deve ter pelo menos oito caracteres; senhas de sete caracteres são substancialmente mais fáceis de serem descobertas. Em sistemas que utilizam senhas DES, não adianta usar uma senha maior do que oito caracteres, pois somente os primeiros oito caracteres são significativos. Consulte a seção *Senha criptografada* começando na página 68 para informações adicionais sobre como ativar senhas MD5, que podem ter mais que oito caracteres.

É importante que a senha de root seja selecionada de maneira que não seja adivinhada com facilidade ou descoberta por tentativa e erro. Teoricamente, o tipo de senha mais seguro é formado por uma seqüência aleatória de letras, pontuação e dígitos. Mas como esse tipo de senha é difícil de ser memorizado e geralmente difícil de ser digitado, talvez não seja otimamente seguro se os administradores a anotarem em papel ou a digitarem muito lentamente.

Até recentemente, uma senha formada de duas palavras selecionadas aleatoriamente separadas por algum sinal de pontuação oferecia um excelente relação entre segurança e facilidade de memorização. Infelizmente, essas senhas agora podem ser quebradas de modo relativamente rápido; hoje em dia não recomendamos esse esquema.

Atualmente, sugerimos que você forme uma senha de root criando uma frase baseada em um "absurdo chocante", definida por Grady Ward em uma versão anterior do FAQ "PGP Pass-phrase":

> *"Absurdo chocante" significa criar uma frase ou sentença curta que seja ao mesmo tempo absurda e chocante na cultura do usuário. Isto é, ela contém uma justaposição de idéias vulgarmente obscenas, racistas e impossíveis ou de outro modo extremistas. Essa técnica é permissível, pois a frase senha, por sua natureza, jamais é revelada a alguém que pudesse ser ofendido por ela.*
>
> *O "absurdo chocante" tem poucas chances de ser duplicado, pois não descreve uma questão concreta que poderia ser acidentalmente redescoberta por uma outra pessoa. A evocação emocional faz com que se torne difícil que o criador se esqueça dela. Um exemplo "suave" poderia ser o seguinte: "Moluscos gostam de beijar minha genitália". O leitor, sem dúvida, será capaz de criar vários exemplos chocantes ou cômicos para uso próprio.*

Você pode reduzir uma frase desse tipo a uma senha de oito caracteres registrando apenas a primeira letra de cada palavra ou por meio de algum processo de transformação semelhante. A segurança de senhas será aumentada enormemente se você incluir números, sinais de pontuação e letras maiúsculas. (Alguns sistemas agora requerem isso.)

Você deve trocar a senha de root:

- Pelo menos a cada três meses
- Cada vez que alguém que conheça a senha saia da empresa
- Sempre que você julgar que a segurança pode ter sido comprometida
- Um dia em que você não esteja planejando divertir-se na noite tanto que terá esquecido a senha na manhã seguinte

3.4 TORNANDO-SE O ROOT

Como root é simplesmente mais um usuário, você pode efetuar login diretamente com a conta root. Mas isso acaba se revelando ser uma má idéia. Para começar, esse procedimento não deixa nenhum registro de quais operações foram efetuadas como root. Isso já é ruim o suficiente quando você se lembra de que mexeu em algo que não devia às três da madrugada, mas não consegue lembrar o que foi que você alterou; e é ainda pior quando um acesso não foi autorizado e você está tentando descobrir o que um intruso fez ao seu sistema. Uma outra desvantagem é que a situação de efetuar login como root não deixa nenhum vestígio de quem estava realmente fazendo a operação. Se várias pessoas têm acesso à conta root, você não será capaz de informar quem a usou e quando a usou.

Por essas razões, a maioria dos sistemas permite que logins de root sejam desativados nos terminais e pela rede — em todos os locais, exceto no console do sistema.[4] Sugerimos que você utilize esse esquema. Consulte a seção *Terminais seguros* na página 473 para descobrir que arquivo você precisa editar em seu sistema em particular.

su: identidade de usuário substituto

Uma maneira ligeiramente melhor de acessar a conta root é usar o comando **su**. Se chamado sem qualquer argumento, **su** solicitará a senha de root e depois iniciará um shell de root. Os privilégios desse shell permanecem em efeito até que o

4. O Linux Ubuntu vai ainda mais longe. Por padrão, o sistema não tem nenhuma senha de root válida e exige o uso do **sudo**, detalhado mais adiante nesta seção.

shell seja encerrado (via <Control-D> ou por meio do comando **exit**). **su** não registra nenhum comando executado como root, mas cria uma entrada de log atestando quem se tornou root e quando.

O comando **su** também pode substituir outras identidades que não sejam a de root. Às vezes, a única maneira de reproduzir ou depurar o problema de um usuário é utilizar **su** como a conta dele para reproduzir a situação em que ocorreu o problema.

Se conhecer a senha de alguém, você pode acessar a conta da pessoa diretamente executando o comando **su** *nomeDoUsuario*. Assim como um **su** para root, você será solicitado, a informar a senha para o *nomeDoUsuario*. Em alguns sistemas, a senha de root permitirá um **su** ou **login** para qualquer conta. Você também pode primeiro utilizar **su** explicitamente para root antes de utilizar **su** para outra conta; root pode ser **su** para qualquer conta sem fornecer uma senha.

É uma boa idéia adquirir o hábito de digitar o nome de caminho completo para o comando **su** (por exemplo, **/bin/su**) em vez de depender do shell para encontrar o comando para você. Isso lhe dará alguma proteção contra programas chamados **su** que podem ter sido sorrateiramente inseridos em seu caminho de busca com a intenção de capturar senhas.[5]

sudo: um su limitado

Como os privilégios da conta superusuário não podem ser subdivididos (pelo menos, não arbitrariamente), é difícil dar a alguém a capacidade de realizar uma outra tarefa (backups, por exemplo) sem dar a essa pessoa livre acesso ao sistema. E se a conta root for usada por vários administradores, você realmente só terá uma vaga idéia de quem a está usando ou o que eles fizeram.

A solução mais amplamente utilizada para esses problemas é um programa chamado **sudo** que atualmente é mantido por Todd Miller. Esse programa é incluído por padrão em todas as nossas distribuições de exemplo, mas também está disponível na forma de código-fonte em www.courtesan.com.

sudo utiliza como argumento uma linha de comando a ser executada como root (ou como algum outro usuário restrito). **sudo** consulta o arquivo **/etc/sudoers**, que lista as pessoas que estão autorizadas a utilizar **sudo** e os comandos que elas têm permissão de executar em cada host. Se o comando proposto for permitido, **sudo** solicita a senha do *próprio usuário* e executa o comando.

Comandos **sudo** adicionais podem ser executados sem que o usuário de **sudo** precise digitar uma senha até um período de cinco minutos (configurável) ter passado sem nenhuma atividade de **sudo**. Esse tempo-limite serve como uma proteção modesta contra usuários com privilégios de **sudo** que deixam terminais abandonados.

sudo mantém um log das linhas de comando que foram executadas, os hosts em que elas foram executadas, as pessoas que as solicitaram, o diretório a partir do qual elas foram executadas e as data/horas em que elas foram invocadas. Essas informações podem ser registradas por syslog ou colocadas em arquivo de sua escolha. Recomendamos usar syslog para encaminhar as entradas de log para um host central seguro.

Uma entrada de log para o usuário randy executando **sudo /bin/cat /etc/sudoers** ficaria parecida com o seguinte:

```
Dec 7 10:57:19 tigger sudo: randy: TTY=ttyp0 ; PWD=/tigger/users/randy; USER=root ;
    COMMAND=/bin/cat /etc/sudoers
```

O arquivo **sudoers** é concebido para que uma única versão possa ser usada em vários hosts distintos ao mesmo tempo. Eis um exemplo típico:

```
# Define aliases para máquinas nos departamentos CS & Physics
Host_Alias CS = tigger, anchor, piper, moet, sigi
Host_Alias PHYSICS = eprince, pprince, icarus

# Define coleções de comandos
Cmnd_Alias DUMP = /usr/sbin/dump, /usr/sbin/restore
Cmnd_Alias PRINTING = /usr/sbin/lpc, /usr/bin/lprm
Cmnd_Alias SHELLS = /bin/sh, /bin/tcsh, /bin/csh, /bin/bash, /bin/ash, /bin/bsh

# Permissões
mark, ed    PHYSICS = ALL
herb        CS = /usr/sbinlocal/bin/tcpdump : PHYSICS = (operator) DUMP
lynda       ALL = (ALL) ALL, !SHELLS
%wheel      ALL, !PHYSICS = NOPASSWD: PRINTING
```

5. Pela mesma razão, recomendamos enfaticamente que você *não* inclua "." (o diretório atual) no caminho de busca do seu shell. Embora conveniente, essa configuração facilita a execução inadvertida de versões "especiais" dos comandos de sistema que um usuário ou invasor plantou como armadilha. Naturalmente, esse cuidado deve ser dobrado para o root.

As cinco primeiras linhas não comentadas definem os grupos de hosts e comandos que são referidos posteriormente nas especificações de permissão do arquivo. As listas poderiam ser incluídas literalmente nas especificações, mas o uso de nomes alternativos faz com que arquivo **sudoers** seja mais fácil de ser lido e compreendido; isso também torna mais fácil suas atualizações no futuro. Também é possível definir aliases (nomes alternativos) para conjuntos de usuários e para conjuntos de usuários com que os comandos possam ser executados.

Cada linha de especificação de permissão inclui informações sobre:

- Os usuários a que a linha se aplica
- Os hosts em que a linha deve ser necessária
- Os comandos que os usuários especificados podem executar
- Os usuários com que os comandos podem ser executados

A primeira linha de permissão se aplica aos usuários mark e ed nas máquinas do grupo `PHYSICS` (eprince, pprince e icarus). O alias do comando integrado `ALL` permite que eles executem qualquer comando. Como nenhuma lista de usuário é especificada nos parênteses, **sudo** executará apenas comandos como root.

A segunda linha de permissão permite que o usuário herb execute **tcpdump** em máquinas `CS` e comandos de dump relacionados nas máquinas `PHYSICS`. Entretanto, os comandos de dump somente podem ser executados como operador, não como root. A verdadeira linha de comando que herb digitaria seria algo parecido com o seguinte:

```
$ sudo -u operator /usr/sbin/dump 0u /dev/hda2
```

A usuária lynda pode executar comandos como qualquer usuário em qualquer máquina, exceto que ela não é capaz de executar vários shells comuns. Isso significa que lynda de fato não pode efetivamente obter um shell de root? Claro que não:

```
$ cp -p /bin/bash /tmp/bash
$ sudo /tmp/bash
```

Em termos genéricos, qualquer tentativa de permitir "todos os comandos exceto ..." é fadada ao insucesso, pelo menos em termos técnicos. Entretanto, ainda pode valer a pena configurar o arquivo **sudoers** dessa maneira como um lembrete de que os shells de root não são vistos com bons olhos. Isso pode desencorajar o uso casual.

A linha final permite que os usuários do grupo wheel executem os comandos **lpc** e **lprm** como root em todas as máquinas exceto eprince, pprince e icarus. Além disso, nenhuma senha é necessária para executar os comandos.

Observe que os comandos em **/etc/sudoers** são especificados com nomes de caminhos completos para evitar que terceiros executem seus próprios programas e scripts como root. Embora não seja apresentado nenhum exemplo acima, também é possível especificar os argumentos que são permitidos para cada comando. De fato, essa configuração simples é apenas uma amostra da beleza e esplendor que o arquivo **sudoers** oferece.

Para modificar **/etc/sudoers**, usamos o comando **visudo**, que faz uma verificação para garantir que ninguém mais esteja editando o arquivo, chama um editor e depois verifica a sintaxe do arquivo editado antes de instalá-lo. Essa última etapa é particularmente importante porque um arquivo **sudoers** inválido poderia impedi-lo de usar **sudo** novamente para corrigi-lo.

O uso de **sudo** apresenta as seguintes vantagens:

- O controle de contas é aprimorado em muito devido ao registro em log dos comandos
- Os operadores podem realizar tarefas sem a necessidade de privilégios de root ilimitados
- A verdadeira senha pode ser conhecida somente por uma ou duas pessoas
- É mais rápido utilizar **sudo** para executar um único comando do que executar **su** ou efetuar login como root
- Privilégios podem ser revogados sem a necessidade de modificar a senha de root
- É mantida uma lista canônica de todos os usuários com privilégios de root
- Há menos chances de um shell de root ficar abandonado
- Um único arquivo pode ser utilizado para controlar o acesso a uma rede inteira

Consulte a página 477 para informações adicionais sobre John the Ripper.

Mas também há uma série de desvantagens. A pior delas é que qualquer brecha na segurança de uma conta pessoal de um usuário de sudo pode ser equivalente a uma brecha na própria conta root. Não há muito que fazer para conter essa ameaça a não ser alertar os usuários de **sudo** para protegerem suas próprias contas como se fossem contas root. Você também pode executar John the Ripper regularmente contra as senhas dos usuários de **sudo** para se certificar de que eles estão escolhendo boas senhas.

O registro em log do comando **sudo** pode ser subvertido por truques como escapes de shell a partir de dentro de um programa permitido ou por **sudo sh** e **sudo su** se você os permitir.

3.5 OUTROS PSEUDO-USUÁRIOS

Root é o único usuário que tem status especial aos olhos do kernel, porém vários outros pseudo-usuários são definidos pelo sistema. É comum substituir o campo de senha criptografado desses usuários especiais em **/etc/shadow** por um asterisco para que as senhas não sejam visíveis.

bin: proprietário herdado de comandos de sistema

Em alguns sistemas UNIX mais antigos, o usuário bin possuía os diretórios que continham os comandos de sistema e a maioria dos demais comandos também. Essa conta é normalmente considerada supérflua nos dias atuais (ou talvez até mesmo ligeiramente insegura), de modo que sistemas modernos (inclusive o Linux) geralmente usam apenas a conta root. De qualquer maneira, agora que a conta bin é "padrão", é impossível livrar-se dela.

daemon: proprietário de software de sistema não-privilegiado

Arquivos e processos que são parte do sistema operacional, mas que não precisam ser possuídos por root são freqüentemente atribuídos a daemon. Essa convenção ajuda a evitar os riscos de segurança associados com a posse por root. Há também um grupo chamado "daemon" que existe por razões semelhantes. Como a conta bin, a conta daemon não é muito usada pela maioria das distribuições Linux.

nobody: o usuário NFS genérico

Consulte página 339 para informações adicionais sobre a conta nobody.

O Network File System (NFS) utiliza a conta nobody para representar usuários root em outros sistemas para fins de compartilhamento de arquivos. Para que roots remotos sejam despojados de seus poderes root, o UID 0 remoto tem de ser mapeado para alguma coisa que não seja o UID 0 local. A conta nobody atua como o alterego genérico desses roots remotos.

Como a conta nobody supostamente deve representar um usuário genérico e relativamente menos poderoso, ela não deve ter propriedade sobre quaisquer arquivos. Se nobody efetivamente possuir arquivos, roots remotos serão capazes de assumir o controle deles. Nobody não deve possuir nenhum arquivo!

Um UID de -1 ou -2 era comum para nobody, e o kernel do Linux ainda aplica o padrão de UID 65534 (a versão de 16 bits do complemento de dois de -2). Algumas distribuições atribuem um UID de número baixo a nobody (por exemplo, Red Hat e Fedora utilizam 99), o que faz mais sentido que 65534 agora que temos UIDs de 32 bits. O único problema é que **exportfs** aparentemente não presta atenção ao arquivo **passwd**, portanto você deve instruí-lo explicitamente, com a opção **anonuid**, a utilizar um UID diferente para nobody.

3.6 EXERCÍCIOS

★ E3.1 Utilize o comando **find** com a opção **-perm** para localizar cinco arquivos setuid em seu sistema. Para cada arquivo, explique por que o mecanismo setuid é necessário para o comando funcionar apropriadamente.

★ E3.2 Crie três frases-senha "absurdas e chocantes" mas guarde-as em segredo. Execute suas três frases-senha com **md5sum** e informe os resultados. Por que é mais seguro compartilhar os resultados de MD5?

★ E3.3 Enumere uma seqüência de comandos que modifiquem a entrada de senha de alguém e mostre como você poderia encobrir os seus rastros. Suponha que você tenha somente o poder **sudo** (todos os comandos permitidos, mas nenhum shell ou **su**).

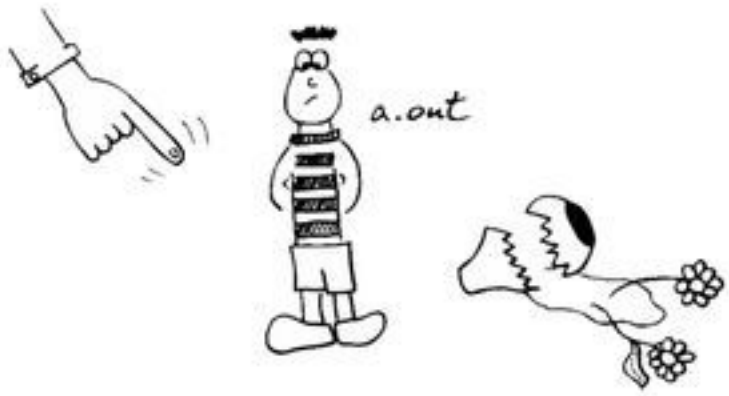
★ E3.4 Crie duas entradas para o arquivo de configuração **sudoers**:

a) Uma entrada que permite aos usuários matt, adam e drew operar o serviço da impressora, descongestioná-la e reiniciar os daemons de impressora na máquina printserver.

b) Uma entrada que permite que drew, smithgr e jimlane eliminem tarefas e reinicializem as máquinas no laboratório dos alunos.

★ E3.5 Instale o **sudo** configurado para enviar a você fofocas feitas por e-mail sobre mau uso. Use-o para testar as entradas **sudo** da questão anterior com nomes de usuários e nomes de máquinas; verifique se **sudo** está registrando apropriadamente no syslog. Observe as entradas de syslog produzidas pelo seu teste. (Requer acesso de root; muito provavelmente você terá também de ajustar **/etc/syslog.conf**.)

4 Controlando processos



Um processo é uma abstração utilizada pelo Linux para representar um programa em execução. É o objeto por meio do qual o uso de memória, tempo de processador e recursos de E/S de um programa podem ser gerenciados e monitorados.

É parte integrante da filosofia do Linux e do UNIX que a maior quantidade possível de trabalho possa ser feita dentro do contexto dos processos, em vez de ser tratada especialmente pelo kernel. Todos os processos de sistema e de usuário seguem as mesmas regras, de modo que você pode utilizar um único conjunto de ferramentas para controlar ambos.

4.1 Componentes de um processo

Um processo é constituído de um espaço de endereços e um conjunto de estruturas de dados dentro do kernel. O espaço de endereços é um conjunto de páginas[1] da memória que o kernel marcou para ser empregado pelo processo. Ele contém o código e as bibliotecas que o processo está executando, as variáveis do processo, suas pilhas e várias informações adicionais de que o kernel precisa enquanto um processo está em execução. Como o Linux é um sistema de memória virtual, não há uma correlação entre a localização de uma página dentro de um espaço de endereços e sua localização dentro da memória física da máquina ou do espaço de troca.

As estruturas de dados internas do kernel registram vários tipos de informação sobre cada processo. Algumas das mais importantes são descritas a seguir:

- O mapa do espaço de endereços do processo
- O status atual do processo (espera, parado, em execução etc.)
- A prioridade de execução do processo
- Informações sobre os recursos que o processo utilizou
- Informações sobre os arquivos e as portas de rede que o processo abriu
- A máscara de sinalização do processo (um registro de quais sinais são bloqueados)
- O proprietário do processo

Alguns desses atributos podem ser compartilhados entre vários processos para criar um "grupo de threads", que é a analogia do Linux de um processo multithread no UNIX tradicional. Embora eles possam compartilhar um espaço de endereçamento, os membros de um grupo de threads têm suas próprias prioridades de execução e estados de execução. Na prática, poucos processos de interesse para os administradores de sistema utilizam múltiplos threads de execução, e, mesmo aqueles que utilizam (como o **named** do BIND 9), geralmente não exigem atenção administrativa nesse nível de granularidade.

Muitos dos parâmetros associados a um processo afetam diretamente sua execução: a quantidade de tempo de processador que ele consome, os arquivos que ele pode acessar e assim por diante. Nas seções a seguir, discutimos o significado e a importância dos parâmetros mais interessantes sob a perspectiva de um administrador de sistemas. Esses atributos são comuns a todas as versões do UNIX e de Linux.

PID: número de identificação de processo

O kernel atribui um número de identificação exclusivo a cada processo. A maioria das chamadas de sistema e comandos que manipulam processos exige que especifiquemos um PID para identificar o objeto da operação. Os PIDs são atribuídos na ordem em que os processos são criados.

---

1. Páginas são as unidades em que a memória é gerenciada, geralmente com um tamanho entre 4 K nos PCs.

PPID: PID pai

O Linux não fornece uma chamada de sistema que crie um processo executando um determinado programa. Em vez disso, um processo existente tem de se clonar para criar um processo novo. O clone pode então trocar o programa que ele está executando por um outro diferente.

Quando um processo é clonado, o original é chamado de pai e, a cópia, de filho. O atributo PPID de um processo é o PID do pai a partir do qual ele foi clonado.[2]

O PID pai é uma parte útil das informações quando você é confrontado com um processo não reconhecido (e possivelmente se comportando mal). Rastrear o processo até sua origem (seja um shell ou um outro programa) pode fornecer uma melhor idéia do propósito e significado desse processo.

UID e EUID: ID de usuário real e efetivo

Consulte a página 68 para informações adicionais sobre UIDs.

O UID de um processo é o número de identificação de usuário da pessoa que o criou ou, mais precisamente, uma cópia do valor EUID do processo pai. Normalmente, apenas o criador (também conhecido como "proprietário") e o superusuário têm permissão para manipular um processo.

O EUID é o ID de usuário "efetivo", é um UID extra utilizado para determinar quais recursos e arquivos um processo tem permissão para acessar em um dado momento. Para a maioria dos processos, o UID e o EUID são os mesmos, sendo a exceção usual programas aos quais são aplicados setuid.

Por que ter ambos, UID e EUID? Simplesmente porque é útil manter uma distinção entre identidade e permissão e porque pode ser que um programa setuid não queira operar com permissões expandidas durante todo o tempo. Na maioria dos sistemas, o UID efetivo pode ser configurado e redefinido para permitir ou restringir as permissões adicionais que ele concede.

O Linux também controla um "UID salvo", que é uma cópia do EUID do processo no ponto em que o processo começou a execução pela primeira vez. A menos que o processo tome medidas para eliminar esse UID salvo, ele permanecerá disponível para uso da mesma forma que o UID real ou efetivo. Um programa setuid escrito de maneira conservadora pode, portanto, renunciar seus privilégios especiais para a maior parte de sua execução, acessando-os somente em pontos específicos em que privilégios extras são necessários.

O Linux também define um parâmetro de processo FSUID que controla a determinação das permissões sobre o sistema de arquivos. Ele é raramente utilizado fora do kernel.

As implicações desse sistema multi-UID podem ser bem sutis. Se precisar aprofundar-se nos detalhes, o livro on-line gratuito *Secure Programming for Linux and Unix HOWTO*, de David A. Wheeler, é um excelente recurso. Ele está disponível em: www.dwheeler.com.

GID e EGID: ID de grupo real e efetivo

Consulte a página 69 para informações adicionais sobre grupos.

O GID é o número de identificação de grupo de um processo. O EGID está relacionado ao GID da mesma maneira que o EUID está relacionado ao UID pelo fato de ele poder ganhar um "upgrade" pela execução de um programa setgid. O kernel mantém um GID salvo semelhante em intenção ao UID salvo.

O atributo GID de um processo é amplamente vestigial. Para os propósitos da determinação de acesso, um processo pode ser membro de muitos grupos de uma vez. A lista de grupos é armazenada separadamente do GID e o EGID distinguidos. As determinações de acesso normalmente levam em conta o EGID e a lista de grupo suplementar, mas não o GID.

A única vez em que o GID desempenha um papel é quando um processo cria novos arquivos. Dependendo de como as permissões do sistema de arquivos foram configuradas, novos arquivos podem adotar o GID do processo criador. Consulte a página 58 para informações adicionais.

Niceness

A prioridade de agendamento de um processo determina quanto tempo de CPU ele recebe. O kernel utiliza um algoritmo dinâmico para calcular prioridades, levando em conta a quantidade de tempo de CPU que um processo consumiu recentemente e o tempo que ele ficou aguardando para ser executado. O kernel também presta atenção a um valor configurado administrativamente que em geral é chamado de "niceness" ("gentileza"), assim chamado porque ele informa o quão gentil você planeja ser com os demais usuários do sistema. Tratamos desse assunto com mais detalhes na página 43.

Em um esforço para fornecer melhor suporte a aplicativos de baixa latência, o Linux adicionou "classes de agendamento" ao tradicional modelo de agendamento do UNIX. Atualmente há três classes de agendamento e cada processo é atribuído a uma classe. Infelizmente, as classes em tempo real não são amplamente usadas nem bem suportadas a partir da linha de

---

2. Pelo menos inicialmente. Se o pai original expirar, **init** (processo 1) se torna o novo pai. Consulte a página 40.

comando. Todos os processos de sistema utilizam o tradicional agendador baseado em valores de *niceness*. Neste livro, discutimos somente o agendador padrão. Visite www.realtimelinuxfoundation.org para mais discussão sobre as questões relacionadas ao agendamento em tempo real.

Terminal de controle

A maioria dos processos não-daemon tem um terminal de controle associado. O terminal de controle determina links padrão para os canais de entrada padrão, saída padrão e erro padrão. Ao iniciar um comando no shell, seu terminal normalmente se torna o terminal de controle do processo. O conceito de um terminal de controle também afeta a distribuição de sinais, que são discutidos na página 40.

4.2 O CICLO DE VIDA DE UM PROCESSO

Para criar um novo processo, um processo geralmente faz uma cópia de si mesmo por meio da chamada de sistema **fork**. **fork** cria uma cópia do processo original que é em grande parte idêntica ao pai. O novo processo possui um PID distinto e tem suas próprias informações contábeis.

fork tem a propriedade exclusiva de retornar dois valores diferentes. Sob o ponto de vista do filho, ele retorna zero. O pai recebe o PID do filho recém-criado. Como os dois processos são idênticos sob outros aspectos, ambos precisam examinar o valor retornado para descobrir que papel eles supostamente devem assumir.

Após um **fork**, o processo filho normalmente usará uma das chamadas de sistema da família **exec** para começar a execução de um novo programa.[3] Essas chamadas mudam o texto do programa que o processo está executando e reinicializam os segmentos de dados e de pilha para um estado inicial predefinido. As várias formas de **exec** diferem somente na maneira como especificam o ambiente e os argumentos de linha de comando a serem dados ao novo programa.

O Linux define uma alternativa a **fork** chamada **clone**. Essa chamada cria conjuntos dos processos que compartilham memória, espaços de E/S ou ambos. O recurso é análogo ao recurso de multithreading encontrado na maioria das versões do UNIX, mas cada thread de execução é representada como um processo completo com todos os recursos em vez de um objeto "thread" especializado.

Consulte o Capítulo 2 para informações adicionais sobre a inicialização e o daemon ***init***.

Quando o sistema é inicializado, o kernel cria e instala vários processos de maneira autônoma. O mais notável destes é **init**, que sempre é o número de processo 1. **init** é responsável por executar os scripts de inicialização de sistema. Todos os processos, exceto aqueles que o kernel cria, são descendentes de **init**.

init também desempenha um importante papel no gerenciamento de processos. Quando um processo se completa, ele chama uma rotina de nome **_exit** para notificar o kernel de que ele está pronto para expirar. Ele fornece um código de saída (um inteiro) que informa a razão de ele estar se terminando. Por convenção, utiliza-se 0 para indicar um término normal ou "bem-sucedido".

Antes de um processo ter a permissão para desaparecer completamente, o Linux requer que sua expiração seja reconhecida pelo processo pai, o que o pai faz por meio de uma chamada a **wait**. O pai recebe uma cópia do código de saída do filho (ou uma indicação da razão de o filho ser eliminado se o filho não tiver se terminado voluntariamente) e também pode obter um resumo do uso de recursos do filho, caso ele queira.

Esse esquema funciona bem se os pais sobreviverem aos seus filhos e se estiverem cientes sobre a necessidade de chamarem **wait** para que os processos expirados possam ser descartados. Mas se o pai for extinto primeiro, o kernel reconhecerá que nenhum **wait** deve ser esperado e ajustará o processo para que o órfão se torne um filho de **init**. **init** aceita os processos órfãos e executa o **wait** necessário para se livrar deles quando eles expirarem.

4.3 SINAIS

Sinais são solicitações de interrupção em nível de processo. Cerca de 30 tipos diferentes estão definidos e eles são utilizados de várias maneiras:

- Podem ser enviados entre processos como um meio de comunicação
- Podem ser enviados pelo driver de terminal para extinguir, interromper ou suspender processos quando teclas especiais como <Control-C> e <Control-Z> forem digitadas[4]

3. Na verdade todas, exceto uma, são rotinas de biblioteca, e não chamadas de sistema.
4. As funções de <Control-Z> e <Control-C>podem ser reatribuídas a outras teclas por meio do comando **stty**, mas isso é raro na prática. Neste capítulo, elas serão referidas pelas suas associações convencionais.

- Podem ser enviados pelo administrador (com **kill**) para obter vários resultados
- Podem ser enviados pelo kernel quando um processo comete uma infração como, por exemplo, uma divisão por zero
- Eles podem ser enviados pelo kernel para notificar um processo sobre uma condição "interessante" como a morte de um processo filho ou sobre a disponibilidade dos dados em um canal de E/S

Um dump de núcleo é a imagem de um processo na memória. Ele pode ser utilizado para depuração.

Quando um sinal é recebido, pode ocorrer uma de duas coisas. Se o processo receptor tiver designado uma rotina de manipulação para esse sinal em particular, o manipulador é chamado com informações sobre o contexto em que o sinal foi entregue. Caso contrário, o kernel toma alguma atitude padrão em nome do processo. A ação padrão varia de sinal para sinal. Muitos sinais terminam o processo; alguns também geram uma imagem da memória (*core dump*).

Especificar uma rotina de manipulação para um sinal dentro de um programa é conhecido como "capturar" o sinal. Quando o manipulador termina a sua ação, a execução é retomada a partir do ponto em que o sinal foi recebido.

Para impedir a chegada de sinais, os programas podem solicitar que eles sejam ignorados ou então bloqueados. Um sinal que é ignorado é simplesmente descartado e não possui nenhum efeito sobre o processo. Um sinal bloqueado é jogado em fila para remessa, porém o kernel não requer que o processo atue em seu nome até que o sinal tenha sido desbloqueado explicitamente. O manipulador para um sinal desbloqueado recentemente é chamado apenas uma vez, mesmo que o sinal tenha sido recebido várias vezes enquanto a recepção estava bloqueada.

A Tabela 4.1 listas alguns sinais que todos os administradores devem conhecer. A convenção em letras maiúsculas para nomes de sinais deriva da tradição da linguagem C. Talvez você também veja eventualmente nomes de sinais escritos com um prefixo SIG (por exemplo, SIGHUP) por razões semelhantes.

Tabela 4.1 Sinais que todos os administradores devem conhecer

#	Nome	Descrição	Padrão	Pode capturar?	Pode bloquear?	Imagem da memória?
1	HUP	Suspender	Terminar	Sim	Sim	Não
2	INT	Interromper	Terminar	Sim	Sim	Não
3	QUIT	Abandonar	Terminar	Sim	Sim	Sim
9	KILL	Matar	Terminar	Não	Não	Não
[a]	BUS	Erro de barramento	Terminar	Sim	Sim	Sim
11	SEGV	Falha de segmentação	Terminar	Sim	Sim	Sim
15	TERM	Término do software	Terminar	Sim	Sim	Não
[a]	STOP	Parar	Parar	Não	Não	Não
[a]	TSTP	Parada de teclado	Parar	Sim	Sim	Não
[a]	CONT	Continuar após uma parada	Ignorar	Sim	Não	Não
[a]	WHINCH	Janela modificada	Ignorar	Sim	Sim	Não
[a]	USR1	Definido pelo usuário	Terminar	Sim	Sim	Não
[a]	USR2	Definido pelo usuário	Terminar	Sim	Sim	Não

a. Varia conforme a arquitetura de hardware; veja **man 7 signal**.

Há outros sinais que não são apresentados na Tabela 4.1, a maioria dos quais são utilizados para informar erros obscuros como "instrução ilegal". O tratamento padrão para sinais como esses é terminar com uma imagem da memória. A captura e o bloqueio são normalmente permitidos, pois alguns programas podem ser inteligentes o bastante para tentar limpar seja lá qual tenha sido o problema que provocou o erro antes de prosseguir.

Os sinais BUS e SEGV também são sinais de erro. Nós os incluímos na tabela, pois são muito comuns: 99% das vezes que um programa trava é, em última instância, um desses dois sinais que o derruba. Por si só, os sinais não têm nenhum valor específico em termos de diagnóstico. Ambos indicam uma tentativa de utilizar ou acessar a memória de forma inadequada.[5]

5 Mais especificamente, erros de barramento resultam das violações dos requisitos de alinhamento ou do uso de endereços sem sentido. As violações de segmentação representam violações de proteção como tentativas de gravar nas partes somente de leitura do espaço de endereçamento.

Os sinais chamados KILL e STOP não podem ser capturados, bloqueados ou ignorados. O sinal KILL destrói o processo receptor e STOP suspende sua execução até que um sinal CONT seja recebido. CONT pode ser capturado ou ignorado, mas não bloqueado.

TSTP é uma versão "soft" de STOP que poderia ser mais bem descrita como uma solicitação de parada. É o sinal gerado pelo driver de terminal quando <Control-Z> é digitado no teclado. Programas que capturam esse sinal normalmente limpam seus estados, depois enviam a eles próprios um sinal STOP para completar a operação de parada. Alternativamente, programas podem ignorar o TSTP a fim de evitar que eles próprios sejam parados a partir do teclado.

Emuladores de terminal enviam um sinal WINCH quando seus parâmetros de configuração (como o número de linhas no terminal virtual) mudam. Essa convenção permite que programas que entendem emuladores como editores de texto se reconfigurem em resposta a mudanças. Se você não obtiver um redimensionamento de janelas apropriado, certifique-se de que WINCH esteja sendo gerado e propagado corretamente.[6]

Os sinais KILL, INT, TERM, HUP e QUIT podem parecer todos iguais já que significam aproximadamente a mesma coisa, mas na verdade suas aplicações são bem diferentes. É lamentável que terminologia tão vaga tenha sido escolhida para eles. Eis um guia de decodificação:

- KILL não pode ser bloqueado e termina um processo em nível de sistema operacional. Um processo nunca "recebe" esse sinal.
- INT é o sinal enviado pelo driver de terminal quando digitamos <Control-C>. É uma solicitação para terminar a operação atual. Programas simples poderiam sair (se eles capturarem o sinal) ou simplesmente permitir que eles próprios sejam eliminados, que é o padrão se o sinal não for capturado. Programas que têm uma linha de comando devem interromper o que estavam fazendo, fazer uma limpeza e aguardar novamente por uma entrada de usuário.
- TERM é uma solicitação para terminar completamente a execução. Espera-se que o processo receptor limpe o seu estado e encerre.
- HUP possui duas interpretações comuns. A primeira delas é como uma solicitação de reinicialização por vários daemons. Se um daemon for capaz de reler seu arquivo de configuração e ajustar-se às alterações sem reinicializar, um HUP geralmente pode ser utilizado para disparar esse tipo de comportamento.

  Em segundo lugar, os sinais HUP são algumas vezes gerados pelo driver de terminal em uma tentativa de "limpar" (isto é, extinguir) os processos associados a um determinado terminal. Esse comportamento é um remanescente da época dos terminais e conexões de modem com fios, daí o nome "hangup".

  Os shells na família de shells C (**tcsh** et al.) normalmente tornam os processos em segundo plano imunes a sinais HUP para poderem continuar a executar depois de o usuário sair do sistema. Os usuários de shells tipo Bourne (**ksh**, **bash** etc.) podem emular esse comportamento por meio do comando **nohup**.
- QUIT é similar a TERM exceto pelo fato de seu padrão gerar uma descarga de memória, caso não seja capturado. Alguns poucos programas canibalizam esse sinal e o interpretam como tendo outro significado.

Os sinais USR1 e USR2 não possuem um significado fixo. Eles estão disponíveis para os programas os utilizarem como quiserem. Por exemplo, o servidor Web Apache interpreta o sinal USR1 como uma solicitação para reiniciar elegantemente.

4.4 KILL E KILLALL: ENVIANDO SINAIS

Como o próprio nome indica, o comando **kill** é utilizado mais freqüentemente para terminar um processo. **kill** pode enviar qualquer sinal, mas, por padrão, envia um TERM. **kill** pode ser utilizado por usuários comuns em seus próprios processos ou pelo superusuário em qualquer processo. A sintaxe é:

```
kill [-sinal] pid
```

onde *sinal* é o número ou nome simbólico do sinal a ser enviado (conforme indicado na Tabela 4.1) e *pid* é o número de identificação do processo alvo. Um *pid* -1 transmite o sinal a todos os processos, exceto **init**.

Um **kill** sem um número de sinal não garante que o processo irá expirar, pois o sinal TERM pode ser capturado, bloqueado ou ignorado. O comando:

```
kill -KILL pid
```

6 O que talvez seja mais fácil dizer do que fazer. O emulador de terminal (por exemplo, **xterm**), o driver de terminal e os comandos no nível do usuário podem desempenhar um papel na propagação de SIGWINCH. Problemas comuns incluem o envio do sinal apenas a um processo em primeiro plano do terminal (em vez de a todos os processos associados a esse terminal) e a falha em propagar uma notificação da alteração de um tamanho pela rede para um computador remoto. Protocolos como Telnet e SSH reconhecem explicitamente alterações no tamanho dos terminais locais e comunicam essas informações para o host remoto. Protocolos mais simples (por exemplo, linhas seriais diretas) não podem fazer isso.

"garantirá" que o processo irá expirar, pois o sinal 9, KILL, não pode ser capturado. Colocamos a palavra "garantirá" entre aspas, pois os processos ocasionalmente podem se tornar tão travados que nem mesmo KILL os afetará (normalmente devido a alguma degeneração de E/S como, por exemplo, a espera por um disco que parou de girar). A reinicialização normalmente é a única maneira de se livrar desses processos.

A maioria dos shells tem sua própria implementação de **kill** integrada que obedece à sintaxe descrita. De acordo com a página do man para o comando **kill** independente, o número ou nome de sinal deve ser precedido, na verdade, por um flag **-s** (por exemplo, **kill -s HUP** *pid*). Mas como alguns shells não entendem essa versão da sintaxe, sugerimos que você fique com a forma **-HUP**, que o comando **kill** independente também compreende. Dessa maneira, você não precisa se preocupar com que versão de **kill** você está realmente utilizando.

Se não conhecer o PID do processo que deseja sinalizar, em geral você o pesquisa por meio do comando **ps**, que é descrito a partir da página 44. Uma outra opção é utilizar o comando **killall**, que executa essa pesquisa para você. Por exemplo, para fazer o daemon **xinetd** atualizar sua configuração, você pode executar:

```
$ sudo killall -USR1 xinetd
```

Observe que se múltiplos processos corresponderem com a string que você fornece, **killall** enviará sinais a todos eles.

O comando **kill** convencional na verdade tem um recurso similar, mas não parece ser tão inteligente quanto **killall** na pesquisa de nomes de comandos. Fique com **killall**.

4.5 ESTADOS DE PROCESSOS

Um processo não é automaticamente elegível para receber tempo de CPU simplesmente porque ele existe. Você precisa estar ciente dos quatro estados de execução listados na Tabela 4.2.

Tabela 4.2 Estados de processos

Estados	Significado
Executável	O processo pode ser executado
Dormente	O processo está aguardando algum recurso
Zumbi	O processo está tentando se destruir
Parado	O processo é suspenso (não recebe permissão para ser executado)

Um processo executável é aquele que está pronto para ser executado toda vez que houver tempo de CPU disponível. Ele obteve todos os recursos necessários e está apenas esperando tempo de CPU para processar seus dados. Assim que o processo fizer uma chamada de sistema que não pode ser completada imediatamente (como uma requisição de leitura de parte de um arquivo), o Linux o colocará para dormir.

Os processos dormentes estão aguardando a ocorrência de um evento específico. Daemons de sistema e shells interativos passam grande parte de seu tempo dormindo, aguardando entrada por terminais ou conexões de rede. Como um processo em estado de dormência é efetivamente bloqueado até que sua requisição seja atendida, ele não obterá nenhum tempo de CPU a menos que receba um sinal.

Algumas operações fazem com que os processos entrem em um estado de dormência ininterrupto. Esse estado normalmente é transitório e não observado na saída de **ps** (indicada por um `D` na coluna `STAT`; consulte a página 44). Entretanto, algumas situações degenerativas podem fazer com que ele persista. A causa mais comum envolve problemas de servidor em um sistema de arquivos NFS montado com a opção "hard". Como os processos no estado de repouso ininterrupto não podem ser acordados mesmo para servir um sinal, eles não podem ser eliminados. Para se livrar deles, você deve corrigir o problema subjacente ou reinicializar.

Zumbis são processos que terminaram a execução, mas ainda não tiveram seus status coletados. Se vir zumbis por aí, verifique os PPIDs destes com **ps** a fim de descobrir de onde eles estão vindo.

Processos parados são administrativamente proibidos de executar. Os processos são interrompidos ao receber um sinal STOP ou TSTP e são reiniciados com CONT. Ser parado é similar a adormecer, mas não há como sair do estado parado a não ser ter algum outro processo que nos acordar (ou elimine você).

4.6 NICE E RENICE: INFLUÊNCIA SOBRE A PRIORIDADE DE PROGRAMAÇÃO

O valor "*niceness*" de um processo é uma dica numérica para o kernel sobre como o processo deve ser tratado em relação a outros processos disputando recursos de CPU. O nome estranho é derivado do fato de que ele determina quão gentil seremos com os demais usuários do sistema. Um valor nice (gentil) alto significa baixa prioridade para o seu processo: você

será gentil. Um valor nice baixo ou negativo significa alta prioridade: você não será gentil. O intervalo de valores nice permitidos vai de -20 a +19.

A menos que o usuário tome alguma ação em especial, um processo recém-criado herda o valor nice de seu processo pai. O proprietário do processo pode aumentar seu valor nice, mas não diminuí-lo, mesmo para retornar o processo de *niceness* padrão. Essa restrição impede que processos com baixa prioridade sejam gerados a partir de filhos de alta prioridade. O superusuário pode configurar valores nice arbitrariamente.

Hoje em dia é raro ter a oportunidade de configurar prioridades manualmente. Nos sistemas da década de 1970 e 1980, o desempenho era significativamente afetado pelo processo que estava na CPU. Hoje em dia, com poder de CPU mais do que adequado na maioria dos desktops, o agendador normalmente faz um bom trabalho no atendimento aos processos. A adição das classes de agendamento fornece aos desenvolvedores um controle adicional nos casos em que a baixa latência de resposta é essencial.

Entretanto, o desempenho de E/S não acompanhou a velocidade cada vez maior das CPUs, e o maior gargalo na maioria dos sistemas passou a ser as unidades de disco. Infelizmente, o valor nice de um processo não tem nenhum efeito sobre o gerenciamento do kernel de sua memória ou E/S; processos com valores nice altos ainda podem monopolizar uma fatia desproporcional desses recursos.

O valor nice de um processo pode ser configurado no momento da criação pelo comando **nice** e pode ser ajustado mais tarde com o comando **renice**. **nice** utiliza uma linha de comando como argumento e **renice** utiliza um PID ou (às vezes) um nome de usuário. Confusamente, **renice** exige uma prioridade absoluta, mas **nice** quer um *incremento* de prioridade que ele então acrescenta ou subtrai da prioridade atual do shell.

Alguns exemplos:

```
$ nice -n 5 ~/bin/longtask        // Reduz a prioridade (eleva nice) por um fator de 5
$ sudo renice -5 8829             // Configura o valor nice como -5
$ sudo renice 5 -u boggs          // Configura o valor nice dos processos de bogg como 5
```

Para complicar as coisas ainda mais, uma versão do **nice** é integrada no shell C e alguns outros shells comuns (mas não **bash**). Se não digitarmos o caminho completo para o comando **nice**, obteremos a versão do shell em vez da do sistema operacional. Essa duplicação pode ser confusa, pois o **-nice** do shell e o comando **-nice** independente utilizam sintaxes diferentes: o shell quer que seu incremento de prioridade seja expresso como *+incr* ou *-incr*, mas comandos independentes querem um flag **-n** seguido pelo incremento de prioridade.[7]

O processo ao qual **nice** é mais comumente aplicado no mundo moderno é **xntpd**, o daemon de sincronização do relógio. Como a prontidão da CPU é crítica para esse caso, ele normalmente é executado com um valor nice cerca de 12 unidades abaixo do padrão (isto é, em uma prioridade mais alta do que a normal).

Se um processo ficar frenético e levar a média de carga do sistema para 65, talvez seja necessário utilizar **nice** para iniciar um shell de alta prioridade antes de executar comandos para investigar o problema. Caso contrário, pode ser difícil executar até mesmo comandos simples.

4.7 PS: MONITORANDO PROCESSOS

ps é a principal ferramenta do administrador de sistemas para monitoramento de processos. Você pode utilizá-lo para exibir o PID, o UID, a prioridade e o terminal de controle dos processos. Ele também dá informações sobre quanta memória foi consumida e seu estado atual (em execução, parado, dormente etc.). Zumbis aparecem em uma listagem **ps** como <defunct>.

O comportamento de **ps** tende a variar enormemente entre as variantes do UNIX, e muitas implementações se tornaram bem complexas ao longo dos últimos anos. Em um esforço de acomodar as pessoas que estão acostumadas com comandos **ps** de outros sistemas, o Linux fornece uma versão trissexual e hermafrodita que entende outros conjuntos de opções de muitas implementações e utiliza uma variável de ambiente para instruí-lo sobre qual personalidade assumir.

Não se assuste com toda essa complexidade: ela está lá principalmente para os desenvolvedores de kernels, não para administradores de sistema. Embora vá utilizar **ps** com freqüência, você precisa saber apenas alguns truques específicos.

Você pode obter uma visão geral dos processos em execução no sistema via **ps aux**. Eis um exemplo (eliminamos a coluna START para que o exemplo coubesse na página e selecionamos somente uma amostra das linhas de saída):

```
$ ps aux
USER      PID   %CPU  %MEM  VSZ   RSS   TTY   STAT  TIME  COMMAND
```

7 Na verdade, é até pior do que isso: o **nice** independente interpretará **nice -5** como significando um incremento *positivo* de 5, enquanto o **nice** integrado interpretará essa mesma forma como um incremento *negativo* de 5.

```
root        1   0.1   0.2   3356   560   ?   S    0:00   init [5]
root        2   0     0        0   0     ?   SN   0:00   [ksoftirqd/0]
root        3   0     0        0   0     ?   S<   0:00   [events/0]
root        4   0     0        0   0     ?   S<   0:00   [khelper]
root        5   0     0        0   0     ?   S<   0:00   [kacpid]
root       18   0     0        0   0     ?   S<   0:00   [kblockd/0]
root       28   0     0        0   0     ?   S    0:00   [pdflush]
...
root      196   0     0        0   0     ?   S    0:00   [kjournald]
root     1050   0     0.1   2652   448   ?   S<s  0:00   udevd
root     1472   0     0.3   3048   1008  ?   S<s  0:00   /sbin/dhclient -1
root     1646   0     0.3   3012   1012  ?   S<s  0:00   /sbin/dhclient -1
root     1733   0     0        0   0     ?   S    0:00   [kjournald]
root     2124   0     0.3   3004   1008  ?   Ss   0:00   /sbin/dhclient -1
root     2182   0     0.2   2264   596   ?   Ss   0:00   syslogd -m 0
root     2186   0     0.1   2952   484   ?   Ss   0:00   klogd -x
rpc      2207   0     0.2   2824   580   ?   Ss   0:00   portmap
rpcuser  2227   0     0.2   2100   760   ?   Ss   0:00   rpc.statd
root     2260   0     0.4   5668   1084  ?   Ss   0:00   rpc.idmapd
root     2336   0     0.2   3268   556   ?   Ss   0:00   /usr/sbin/acpid
root     2348   0     0.8   9100   2108  ?   Ss   0:00   cupsd
root     2384   0     0.6   4080   1660  ?   Ss   0:00   /usr/sbin/sshd
root     2399   0     0.3   2780   828   ?   Ss   0:00   xinetd -stayalive
root     2419   0     1.1   7776   3004  ?   Ss   0:00   sendmail: accepi
...
```

Os nomes dos comandos entre colchetes não são realmente comandos, mas antes threads do kernel agendados como processos. O significado de cada campo é explicado na Tabela 4.3, a seguir.

Tabela 4.3 Explicação da saída gerada por **ps aux**.

Campo	Sumário
USER	Nome de usuário do proprietário do processo
PID	ID do processo
%CPU	Porcentagem dos recursos de CPU que esse processo está utilizando
%MEM	Porcentagem da memória real que esse processo está utilizando
VSZ	Tamanho virtual do processo
RSS	Tamanho configurado residente (número de páginas na memória)
TTY	ID do terminal controlador
STAT	Status do processo atual: R = Executável D = Em repouso não interrompível S = Em repouso (< 20 sec) T = Rastreado ou parado Z = Zumbi Flags adicionais: W = Processo sofre swap < = O processo tem prioridade mais alta que a normal N = O processo tem prioridade mais baixa que a normal L = Algumas páginas são bloqueadas no núcleo s = O processo é um líder de sessão
START	A data/hora em que o processo foi iniciado
TIME	O tempo de CPU que o processo consumiu
COMMAND	Nome e argumentos do comando[a]

a. Os programas podem modificar essa informação; portanto, ela não é necessariamente uma representação precisa da verdadeira linha de comando.

Um outro conjunto útil de argumentos é **lax**, que fornece informações mais técnicas. Ele também é um pouco mais rápido de executar porque não tem de converter cada UID em um nome de usuário — a eficiência pode ser importante se o sistema já estiver travado por algum outro processo.

Mostramos a seguir um breve exemplo de como **ps lax** inclui campos como o ID do processo pai (PPID), o valor nice (NI) e o recurso que o processo está esperando (WCHAN).

```
$ ps lax
  F  UID   PID   PPID   PRI    NI     VSZ    RSS    WCHAN  STAT   TIME   COMMAND
  4    0     1      0    16     0    3356    560    select S      0:00   init [5]
  1    0     2      1    34    19       0      0    ksofti SN     0:00   [ksoftirqd/0
  1    0     3      1     5   -10       0      0    worker S<     0:00   [events/0]
  1    0     4      3     5   -10       0      0    worker S<     0:00   [khelper]
  5    0  2186      1    16     0    2952    484    syslog Ss     0:00   klogd -x
  5   32  2207      1    15     0    2824    580    -      Ss     0:00   portmap
  5   29  2227      1    18     0    2100    760    select Ss     0:00   rpc.statd
  1    0  2260      1    16     0    5668   1084    -      Ss     0:00   rpc.idmapd
  1    0  2336      1    21     0    3268    556    select Ss     0:00   acpid
  5    0  2384      1    17     0    4080   1660    select Ss     0:00   sshd
  1    0  2399      1    15     0    2780    828    select Ss     0:00   xinetd -sta
  5    0  2419      1    16     0    7776   3004    select Ss     0:00   sendmail: a
...
```

4.8 TOP: MONITORANDO AINDA MELHOR OS PROCESSOS

Uma vez que comandos como **ps** oferecem apenas um instantâneo de seu sistema em um dado momento, normalmente é difícil ter visão geral do que realmente está acontecendo. O comando **top** fornece um resumo regularmente atualizado dos processos ativos e o uso de recursos.

Por exemplo:

```
top - 16:37:08 up  1:42,  2 users,  load average: 0.01, 0.02, 0.06
Tasks:  76 total,   1 running,  74 sleeping,   1 stopped,   0 zombie
Cpu(s):  1.1% us,  6.3% sy,  0.6% ni, 88.6% id,  2.1% wa,  0.1% hi,  1.3% si
Mem:     256044k  total, 254980k used,  1064k  free,  15944k buffers
Swap:    524280k  total, 0k      used,  524280k free,  153192k cached

 PID       USER    PR     NI     VIRT    RES    SHR    S      %CPU   %MEM   TIME+    COMMAND
 3175      root    15      0    35436    12m    4896   S      4.0    5.2    01:41.9  X
 3421      root    25     10    29916    15m    9808   S      2.0    6.2    01:10.5  rhn-applet-gui
    1      root    16      0     3356    560    480    S      0.0    0.2    00:00.9  init
    2      root    34     19        0    0      0      S      0.0    0      00:00.0  ksoftirqd/0
    3      root     5    -10        0    0      0      S      0.0    0      00:00.7  events/0
    4      root     5    -10        0    0      0      S      0.0    0      00:00.0  khelper
    5      root    15    -10        0    0      0      S      0.0    0      00:00.0  kacpid
   18      root     5    -10        0    0      0      S      0.0    0      00:00.0  kblockd/0
   28      root    15      0        0    0      0      S      0.0    0      00:00.0  pdflush
   29      root    15      0        0    0      0      S      0.0    0      00:00.3  pdflush
   31      root    13    -10        0    0      0      S      0.0    0      00:00.0  aio/0
   19      root    15      0        0    0      0      S      0.0    0      00:00.0  khubd
   30      root    15      0        0    0      0      S      0.0    0      00:00.2  kswapd0
  187      root     6    -10        0    0      0      S      0      0      00:00.0  kmirrord/0
  196      root    15      0        0    0      0      S      0      0      00:01.3  kjournald
...
```

Por padrão, a tela é atualizada a cada 10 segundos. Os processos mais ativos aparecem no alto. **top** também aceita entrada pelo teclado e permite que enviemos sinais e alteremos os valores nice dos processos, para que possamos observar como nossas ações afetam a condição geral da máquina.

Root pode executar **top** com a opção **q** para elevá-lo ao mais alto nível de prioridade possível. Isso pode ser muito útil quando tentamos rastrear um processo que já derrubou o sistema.

4.9 O SISTEMA DE ARQUIVOS /PROC

As versões Linux de **ps** e **top** lêem suas informações sobre o status de um processo a partir do diretório **/proc**, um pseudo-sistema de arquivos em que o kernel expõe uma variedade de informações interessantes sobre o estado do sistema. Apesar do nome **/proc** (e do nome do tipo de sistema de arquivos subjacente, "proc"), as informações não estão limitadas às informações sobre processos — todas as informações sobre o status e as estatísticas geradas pelo kernel são representadas aqui. Você pode até modificar alguns parâmetros gravando no arquivo **/proc** apropriado — veja alguns exemplos na página 600.

Embora algumas informações sejam fáceis de acessar por meio de comandos front-end como **vmstat** e **ps**, algumas informações menos populares devem ser lidas diretamente de **/proc**. Vale a pena pesquisar esse diretório a fim de se familiarizar com tudo que está aí. **man proc** também lista algumas dicas e truques úteis.

Como o kernel cria o conteúdo dos arquivos **/proc** instantaneamente (à medida que são lidos), a maioria parece estar vazia quando listada com **ls -l**. Você terá de executar **cat** ou **more** no conteúdo para ver o que ele realmente contém. Mas seja cauteloso — alguns arquivos contêm ou estão vinculados a dados binários que podem confundir seu emulador de terminal se visualizados diretamente.

Informações específicas aos processos são divididas em subdiretórios identificados pelo PID. Por exemplo, **/proc/1** sempre é o diretório que contém as informações sobre **init**. A Tabela 4.4 lista os arquivos mais úteis por processo individual.

Tabela 4.4 Arquivos de informações sobre processos em /proc (subdiretórios numerados)

Arquivo	Conteúdo
cmd	Comando ou programa que o processo está executando
Cmdline[a]	Linha de comando completa do processo (separada por null)
cwd	Link simbólico para o diretório atual do processo
environ	As variáveis de ambiente do processo (separadas por null)
exe	Link simbólico para o arquivo sendo executado
fd	Subdiretório contendo os links para cada descritor de arquivo aberto
maps	Informações sobre o mapeamento de memória (segmentos compartilhados, bibliotecas etc.)
root	Link simbólico para o diretório-raiz do processo (configurado com **chroot**)
stat	Informações gerais sobre o status do processo (mais bem decodificadas com **ps**)
statm	Informações sobre o uso de memória

a. Pode estar indisponível se o processo sofrer swap de memória.

Os componentes individuais contidos dentro dos arquivos **cmdline** e **environ** são separados por caracteres nulos em vez de por nova linha. Você pode filtrar o conteúdo desses componentes com **tr "\000" "\n"** a fim de torná-los mais legíveis.

O subdiretório **fd** representa os arquivos abertos na forma de links simbólicos. Descritores de arquivo conectados a pipes ou a soquetes de rede não têm um nome de arquivo associado. Em vez disso, o kernel fornece uma descrição genérica como o alvo de link.

O arquivo **maps** pode ser útil para determinar a quais bibliotecas um programa está vinculado ou de quais ele depende.

4.10 STRACE: RASTREANDO SINAIS E CHAMADAS DE SISTEMA

Em um sistema UNIX tradicional, pode ser difícil descobrir o que um processo está realmente fazendo. Talvez você tenha de fazer suposições educadas com base nos dados indiretos do sistema de arquivos e de ferramentas como **ps**. Em comparação, o Linux permite observar um processo diretamente com o comando **strace**, que mostra cada chamada de sistema que o processo faz e cada sinal que ele recebe. Você pode até anexar **strace** a um processo em execução, dar uma espiada rápida e então desanexá-lo do processo sem incomodá-lo.[8]

Embora chamadas de sistema ocorram em um nível de abstração relativamente baixo, normalmente podemos obter várias informações sobre a atividade de um processo a partir da saída de **strace**. Por exemplo, o log a seguir foi produzido pela execução de **strace** contra uma cópia ativa de **top**:

8 Bem, normalmente. Em alguns casos, **strace** pode interromper chamadas de sistema. O processo monitorado deve então estar preparado para reiniciá-las. Essa é uma regra padrão da higiene de software UNIX, mas uma nem sempre observada.

```
$ sudo strace -p 5810
gettimeofday( {1116193814, 213881}, {300, 0} )              = 0
open("/proc", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY)   = 7
fstat64(7, {st_mode=S_IFDIR|0555, st_size=0, ...} )         = 0
fcntl64(7, F_SETFD, FD_CLOEXEC)                             = 0
getdents64(7, /* 36 entries */, 1024)                       = 1016
getdents64(7, /* 39 entries */, 1024)                       = 1016
stat64("/proc/1", {st_mode=S_IFDIR|0555, st_size=0, ...} )  = 0
open("/proc/1/stat", O_RDONLY)                              = 8
read(8, "1 (init) S 0 0 0 0 -1 4194560 73"..., 1023)        = 191
close(8)   = 0
...
```

Não apenas **strace** mostra o nome de cada chamada de sistema feita pelo processo, como também decodifica os argumentos e mostra o código resultante retornado pelo kernel.

Nesse exemplo, **top** começa verificando a data/hora atual. Ele então abre, obtém informações (*stats*) sobre o diretório **/proc** e lê o conteúdo desse diretório, obtendo assim uma lista dos processos que estão atualmente em execução. **top** prossegue e obtém informações (*stats*) sobre o diretório que representa o processo **init** e então abre **/proc/1/stat** para ler as informações de status de **init**.

4.11 PROCESSOS DESCONTROLADOS

Consulte a página 561 para informações adicionais sobre processos descontrolados.

Os processos descontrolados dividem-se em duas categorias: processos de usuário que consomem quantidades excessivas de um recurso de sistema como tempo de CPU ou espaço em disco, e processos de sistema que de repente se enfurecem e demonstram um comportamento indomável. O primeiro tipo de descontrole não é necessariamente um defeito; pode ser simplesmente que o processo seja um comilão de recursos. Os processos de sistema supostamente sempre devem se comportar de maneira razoável.

Podemos identificar processos que utilizam tempo de CPU em excesso observando a saída gerada por **ps** ou **top**. Se estiver evidente que um processo de usuário está consumindo mais CPU do que seria razoável esperar, investigue esse processo. O passo um em um servidor ou sistema compartilhado é entrar em contato com o proprietário do processo e perguntar o que está acontecendo. Se o proprietário não puder ser encontrado, você terá de verificar por conta própria. Embora normalmente você deva evitar a investigação dos diretórios dos usuários, é aceitável quando estamos tentando descobrir a origem de um processo descontrolado para descobrir o que está acontecendo.

Há duas razões para descobrir o que um processo está tentando fazer antes de mexer com ele. Em primeiro lugar, o processo pode realmente ser legítimo e importante para o usuário. Não é razoável eliminar processos a esmo apenas porque casualmente eles utilizam muito a CPU. Em segundo lugar, o processo pode ser malicioso ou destrutivo. Nesse caso, você tem de descobrir o que o processo está fazendo (por exemplo, quebrando senhas) para que você possa corrigir o estrago.

Se a razão para a existência de um processo descontrolado não puder ser determinada, suspenda-o com um sinal STOP e envie e-mail para o proprietário dele explicando o que aconteceu. O processo poderá ser reiniciado posteriormente emitindo-se um sinal CONT. Esteja ciente de que alguns processos podem ser arruinados após um repouso muito longo, de modo que esse procedimento nem sempre é inteiramente benigno. Por exemplo, um processo poderia despertar e descobrir que algumas de suas conexões de rede foram desfeitas.

Se um processo utiliza uma quantidade excessiva de CPU, mas parece realizar algo razoável e funcionar corretamente, você deve utilizar **renice** para elevar seu valor nice (ou seja, prioridade menor) e solicitar ao proprietário para utilizar **nice** no futuro.

Processos que utilizam excessivamente a memória relativa à RAM física do sistema podem causar sérios problemas de desempenho. Você pode verificar o tamanho da memória dos processos utilizando **top**. A coluna VIRT mostra a quantidade total de memória virtual alocada por cada processo e a coluna RES mostra a parte dessa memória que atualmente é mapeada para páginas de memória específicas (o "conjunto residente").

Esses dois números podem incluir recursos compartilhados como bibliotecas; o que os torna potencialmente enganosos. Uma medida mais direta de consumo de memória específico aos processos está localizada na coluna DATA, que não é mostrada por padrão. Para adicionar essa coluna à exibição de **top**, pressione a tecla **f** quando **top** está em execução e selecione DATA na lista. O valor DATA indica a quantidade de memória nos segmentos de dados e de pilha de cada processo; portanto, ele é relativamente específico a processos individuais (segmentos de memória compartilhada de módulo). Procure por crescimento ao longo do tempo bem como do tamanho absoluto.

Processos descontrolados que geram saídas podem esgotar um sistema de arquivos inteiro, provocando diversos problemas. Quando um sistema de arquivos se enche, um grande número de mensagens é registrado no log do console e tentativas de se gravar no sistema de arquivos produzem mensagens de erro.

A primeira coisa a fazer nessa situação é interromper o processo que está preenchendo o disco. Se você vem mantendo uma reserva de espaço em disco razoável, pode estar quase certo de que algo está errado quando ele repentinamente se preenche. Não há comando análogo a **ps** que irá informá-lo sobre quem está consumindo espaço em disco à mais rápida taxa, mas várias ferramentas podem identificar arquivos que estão abertos atualmente e os processos que os estão utilizando. Veja as informações sobre **fuser** e **lsof** na página 52, para informações adicionais.

Talvez você queira suspender todos os processos que parecem suspeitos até localizar aquele que está provocando o problema, mas lembre-se de reiniciar os inocentes quando tiver terminado o serviço. Ao localizar o processo problemático, elimine os arquivos que ele estava criando.

Um velho e bem conhecido truque é iniciar um loop infinito no shell que faz o seguinte:

```
while 1
    mkdir adir
    cd adir
    touch afile
end
```

Esse programa ocasionalmente aparece executando a partir de um sistema publicamente acessível que foi de modo inadvertido deixado conectado. Ele não consome muito espaço de disco real, mas preenche a tabela de inodes do sistema de arquivos e impede que outros usuários criem novos arquivos. Não há muito que possamos fazer, exceto limpar os seus efeitos e alertar os usuários para protegerem suas contas. Pelo fato de a árvore de diretórios que é deixada para trás por essa pequena "jóia" ser normalmente muito grande para ser tratada por **rm -r**, talvez você tenha de criar um script que vá até as partes mais internas da árvore e então elimine os diretórios à medida que ele for voltando pela árvore.

Se o problema ocorre em **/tmp** e você configurou **/tmp** como um sistema de arquivos distinto, é possível reinicializar **/tmp** com **mkfs** em vez de tentar eliminar arquivos individuais. Consulte o Capítulo 7, *Adicionando um disco*, para informações adicionais sobre gerenciamento de sistemas de arquivos.

4.12 LEITURA RECOMENDADA

BOVET, DANIEL E CESATI, MARCO. *Understanding the Linux Kernel. 3. ed.* Sebastopol, CA: O'Reilly Media, 2006.

4.13 EXERCÍCIOS

E4.1 Explique a relação existente entre o UID de um arquivo e os UIDs real e efetivo de um processo em execução. Além de controle de acesso a arquivos, qual é o propósito do UID efetivo de um processo?

E4.2 Suponha que um usuário em sua instalação tenha iniciado um processo de longa duração que está consumindo uma fração significativa dos recursos de uma máquina.

a) Como você reconheceria um processo que está exaurindo recursos?

b) Suponha que o processo mal-comportado possa ser legítimo e não deva ser eliminado. Enumere os comandos que você poderia utilizar para colocá-los "na geladeira" (interrompê-lo temporariamente enquanto investiga).

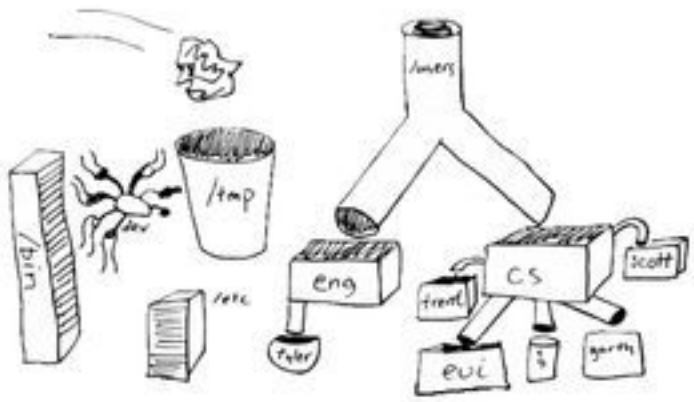
c) Mais tarde, você descobre que o processo pertence a seu chefe e tem de continuar executando. Enumere os comandos que você utilizaria para retomar a tarefa.

d) Alternativamente, suponha que o processo tenha de ser eliminado. Que sinal você deveria emitir e por quê? E se você precisasse garantir que o processo fosse efetivamente extinto?

E4.3 Localize um processo com vazamento de memória (escreva seu próprio programa se não tiver um à mão). Utilize **ps** ou **top** para monitorar o uso de memória do programa à medida que ele é executado.

★ E4.4 Escreva um script Perl simples que processe a saída de **ps** para determinar o VSZ e RSS total dos processos em execução no sistema. Como esses números se relacionam com a quantidade real de memória física e com o espaço de troca do sistema?

5 O sistema de arquivos



Responda rapidamente: quais dos seguintes itens você esperaria encontrar em um "sistema de arquivos"?

- Processos
- Portas seriais
- Estruturas de dados do kernel e parâmetros de configuração
- Canais de comunicação entre processos

Se o sistema for Linux, a resposta seria "todas as anteriores". E, logicamente, você também poderá encontrar alguns arquivos lá.[1]

Embora o objetivo básico de um sistema de arquivos seja o de representar e organizar os recursos de armazenamento de um sistema, os programadores procuram evitar reinventar a roda quando se trata de gerenciar outros tipos de objetos. Freqüentemente se comprova que é natural e conveniente associar tais objetos com o espaço de nomes do sistema de arquivos. Essa unificação tem algumas vantagens (interface de programação consistente, acesso fácil a partir do shell) e algumas desvantagens (implementações de sistema de arquivos quase como o monstro do Frankenstein), mas goste ou não, essa é a maneira UNIX (e, portanto, do Linux).

O sistema de arquivos pode ser imaginado como compreendendo quatro componentes principais:

- Um espaço de nomes — uma maneira de dar nomes às coisas e organizá-las de forma hierárquica
- Uma API[2] — um conjunto de chamadas de sistema para navegar e manipular objetos
- Um modelo de segurança — um esquema para proteger, ocultar e compartilhar coisas
- Uma implementação — software que amarra o modelo lógico ao hardware em si

NFS, o Network File System, é descrito no Capítulo 16.

O Linux define uma interface abstrata no nível do kernel que acomoda vários sistemas de arquivos auxiliares distintos. Algumas partes da árvore de arquivos são manipuladas por implementações tradicionais baseadas em disco; outras são tratadas por drivers distintos dentro do kernel. Por exemplo, sistemas de arquivos NFS são manipulados por um driver que encaminha as operações requisitadas a um servidor ou a um outro computador.

Infelizmente, os limites arquitetônicos não são definidos claramente e há um número razoável de casos especiais. Por exemplo, arquivos de dispositivo fornecem uma maneira para os programas se comunicarem com drivers dentro do kernel. Eles não são realmente arquivos de dados, mas são manipulados pelo driver de sistema de arquivos básico e suas características são armazenadas no disco. Talvez os detalhes sejam um tanto diferentes caso o sistema de arquivos fosse reimplementado à luz da experiência das últimas décadas.

Consulte www.namesys.com para informações adicionais sobre o ReiserFS.

Outro fator complicador (mas em última instância benéfico) é o fato de o Linux suportar mais do que um sistema de arquivos baseado em disco. Na categoria melhores e mais modernos está o sistema de arquivos ext3fs que serve como padrão para a maioria das distribuições, junto com o ReiserFS, JFS da IBM e XFS da SGI. O ext2fs mais antigo, precursor do ext3fs, ainda é suportado por todas as distribuições e permanecerá assim por muito tempo.

Também há muitas implementações de sistemas de arquivos estrangeiros, como os sistemas de arquivos FAT e NTFS, utilizados pelos sistemas de arquivos do Microsoft Windows, e o ISO 9660, utilizados em CD-ROMs. O Linux suporta mais tipos de sistemas de arquivos do que qualquer outra variante do UNIX. Sua ampla gama de opções nos dá grande flexibilidade e torna mais fácil compartilhar arquivos com outros sistemas.

---

1. Talvez seja mais preciso dizer que essas entidades são *representadas* dentro do sistema de arquivos. Na maioria dos casos, o sistema de arquivos é utilizado como um ponto de encontro para conectar clientes com os drivers e servidores que eles estão buscando.
2. Interface de programas aplicativos (*application programming interface* – API), um termo genérico para o conjunto de rotinas que uma biblioteca, sistema operacional ou pacote de software fornece aos programadores para chamada.

O sistema de arquivos é um tópico rico que abordamos sob diversos ângulos. Este capítulo informa onde encontrar coisas sobre seu sistema e descreve as características dos arquivos, os significados dos bits de permissão e o emprego de alguns comandos básicos que visualizam e configuram atributos. O Capítulo 7, *Adicionando um disco*, é onde você encontrará os tópicos mais técnicos sobre sistemas de arquivos como partição de discos. O Capítulo 16, *O sistema de arquivos de rede*, descreve os sistemas de compartilhamento de arquivos que são comumente utilizados com o Linux. Talvez você também queira consultar o Capítulo 26, *Cooperando com o Windows*, que discute a utilização de sistemas de Linux de softwares para compartilhar sistemas de arquivos com computadores que são executados no Microsoft Windows.

Com tantas implementações de sistema de arquivos diferentes disponíveis para o Linux, pode parecer estranho que este capítulo apresente os fatos como se houvesse apenas um único sistema de arquivos Linux. Podemos ser vagos quanto às implementações porque os sistemas de arquivos mais modernos tentam fornecer a funcionalidade tradicional de sistema de arquivos de uma maneira mais confiável e mais rápida ou adicionam recursos extras como uma camada por cima da semântica de sistema de arquivos padrão. (Alguns sistemas de arquivos fazem ambos.) Para nossa alegria ou nossa tristeza, a idéia de que existe muito software depende do modelo descrito neste capítulo para que esse modelo seja descartado.

5.1 NOMES DE CAMINHO

O sistema de arquivos é apresentado como uma única hierarquia unificada[3] que inicia no diretório / e continua descendo por um número arbitrário de subdiretórios. / também é chamado de diretório-raiz.

A lista de diretórios que deve ser percorrida para localizar um determinado arquivo, juntamente com o seu nome de arquivo, forma um nome de caminho. Os nomes de caminho podem ser tanto absolutos (**/tmp/foo**) como relativos (**book4/filesystem**). Os nomes de caminho são interpretados no diretório atual. Você pode estar acostumado a pensar no diretório atual como um recurso do shell, mas cada processo possui um.

Os termos *arquivo*, *nome de arquivo*, *nome de caminho* e *caminho* são mais ou menos intercambiáveis (ou, pelo menos, nós os utilizamos de forma intercambiável neste livro). *Nome de arquivo* e *caminho* podem ser utilizados tanto para caminhos absolutos como para caminhos relativos; *nome de caminho* geralmente sugere um caminho absoluto.

O sistema de arquivos pode ser arbitrariamente profundo. Entretanto, cada componente de um nome de caminho tem de ter um nome que não supere 255 caracteres, e um caminho único não pode conter mais do que 4.095 caracteres. Para acessar um arquivo com um nome de caminho superior a isso, você tem de mudar de diretório com o comando **cd** para um diretório intermediário e utilizar um nome de caminho relativo.[4]

A nomeação de arquivos e diretórios é essencialmente irrestrita, exceto que os nomes têm comprimento limitado e não devem conter o caractere de barra ou nulos. Em particular, os espaços são permitidos. Infelizmente, o UNIX tem uma longa tradição de separar argumentos de linha de comando com espaços em branco, de modo que softwares mais antigos tendem a parar quando encontram espaços dentro de nomes de arquivos.

Os espaços em nomes de arquivo eram principalmente encontrados em sistemas de arquivos compartilhados com Macs e PCs, mas agora foram assimilados pela cultura Linux e também são encontrados em alguns pacotes de software padrão. Não há duas maneiras de fazer isso: scripts administrativos devem ser preparados para lidar com espaços.

No shell e em scripts, os nomes de arquivo longos podem ser colocados entre aspas para manter suas partes unidas. Por exemplo, o comando:

```
$ less "My excellent file.txt"
```

conserva **My excellent file.txt** como um único argumento para **less**. Você também pode "escapar" (isto é, tratar literalmente) espaços individuais com uma barra invertida. O recurso de completamento de nome de arquivo dos shells comuns (normalmente associados com a tecla <Tab>) faz isso para você.

Ao escrever scripts, uma arma útil é conhecer é a opção **-print0** do **find**. Em combinação com **xargs -0**, essa opção faz a combinação **find/xargs** funcionar de modo correto, independentemente do espaço em branco contido no nome de arquivo. Por exemplo, o comando:

```
$ find /home -size +1M -print0 | xargs -0 ls -l
```

imprime uma longa listagem **ls** de cada arquivo da partição **/home** com um tamanho de mais de um megabyte.

---

3. O sistema de hierarquia única difere daquele utilizado pelo Windows que conserva o conceito de espaços de nomes específicos a discos.
4. No caso de não ter ficado claro: a maioria dos formatos de sistemas de arquivos de disco não impõe por si próprio um limite em relação ao comprimento total de nomes de arquivos. Entretanto, as chamadas de sistema que acessam o sistema de arquivos não permitem que seus argumentos string sejam mais longos do que 4.095 caracteres.

5.2 MONTAGEM E DESMONTAGEM DE SISTEMA DE ARQUIVOS

O sistema de arquivos é composto por blocos menores — também chamados de sistemas de arquivos —, cada um dos quais formado por um diretório e seus subdiretórios e arquivos. É normalmente evidente a partir do contexto qual tipo de "sistema de arquivos" está sendo discutido, mas, por clareza, utilizaremos o termo "árvore de arquivos" para nos referirmos ao layout geral do sistema de arquivos e reservaremos o termo "sistema de arquivos" para os blocos anexados à árvore.

A maioria dos sistemas de arquivos são partições de disco, porém, conforme mencionado anteriormente, eles podem ser qualquer coisa desde que obedeçam à API apropriada: servidores de arquivos de rede, componentes do kernel, emuladores de disco baseado em memória etc. O Linux tem até um sistema de arquivos "loopback" engenhoso que permite montar arquivos individuais como se fossem dispositivos distintos.

Os sistemas de arquivos são anexados à árvore com o comando **mount**. **mount** associa um diretório dentro da árvore de arquivos existente, chamado de ponto de montagem, à raiz do sistema de arquivos recém-anexado. O conteúdo anterior do ponto de montagem se torna inacessível enquanto um outro sistema de arquivos está montado lá. Entretanto, os pontos de montagem são normalmente diretórios vazios.

Por exemplo,

```
# mount /dev/hda4 /users
```

instala o sistema de arquivos armazenado na partição de disco representada por **/dev/hda4** sob o caminho **/users**. Poderíamos então utilizar **ls /users** para ver o conteúdo do sistema de arquivos.

Uma lista dos sistemas de arquivos que são habitualmente montados em um sistema particular é mantida no arquivo **/etc/fstab**. A informação contida nesse arquivo permite que os sistemas de arquivos sejam checados (**fsck -A**) e montados (**mount -a**) automaticamente no momento da inicialização. Ela também serve como documentação para o layout da árvore de arquivos no disco e permite utilizar comandos abreviados, como **mount /usr** (a localização do sistema de arquivos a ser montado é procurada no arquivo **fstab**). Consulte a página 89 para uma discussão completa do arquivo **fstab** e seus irmãos.

Os sistemas de arquivos são desanexados por meio do comando **umount**. Não podemos desmontar um sistema de arquivos que está "ocupado"; não deve haver nenhum arquivo ou processo aberto cujos diretórios atuais estejam localizados lá, e se o sistema de arquivos contiver programas executáveis, eles não podem estar em execução.

Os kernels do Linux 2.4.11 e versões superiores definem uma opção unmount "sob demanda" (invocada com **umount -l**) que remove o sistema de arquivos da hierarquia de nomeação, mas que não o desmonta realmente até que todas as referências de arquivo existentes tenham sido fechadas. É uma polêmica se essa é ou não uma opção útil. Para começar, não há nenhuma garantia de que as referências existentes irão alguma vez se fechar por conta própria. Além disso, o estado "semidesmontado" pode apresentar inconsistência na semântica de sistema de arquivos para os programas que o estão utilizando; eles podem ler e gravar por meio de handles de arquivos existentes, mas não podem abrir novos arquivos nem realizar outras operações de sistema de arquivos.

Se o kernel reclamar que um sistema de arquivos que você está tentando desmontar está ocupado, você pode executar **fuser** para descobrir a razão. Quando invocado com flags **-mv** e um ponto de montagem, **fuser** exibe cada processo que está utilizando um arquivo ou diretório nesse sistema de arquivos:

```
$ fuser -mv /usr
          USER     PID    ACCESS  COMMAND
/usr      root     444    ....m   atd
          root     499    ....m   sshd
          root     520    ....m   lpd
          ...
```

Os códigos de letras na coluna ACCESS mostram o que cada processo está fazendo para interferir na sua tentativa de desmontagem. A Tabela 5.1 descreve o significado de cada código.

Tabela 5.1 Códigos de atividade exibidos por fuser

Código	Significado
f	O processo possui um arquivo aberto para leitura ou gravação
c	O diretório atual do processo se encontra no sistema de arquivos
e	O processo está atualmente executando um arquivo
r	O diretório-raiz do processo (configurado por meio de **chroot**) se encontra no sistema de arquivos
m	O processo mapeou um arquivo ou biblioteca compartilhada (normalmente um executável inativo)

Para determinar exatamente quais são os processos problemáticos, basta executar **ps** com a lista de PIDs retornados por **fuser**. Por exemplo,

```
$ ps -fp "444 499 520"
    UID   PID   PPID   C     STIME  TTY   TIME        CMD
 daemon   444   1      0     Apr11  ?     00:00:00    /usr/sbin/atd
   root   499   1      0     Apr11  ?     00:00:23    /usr/sbin/sshd
     lp   520   1      0     Apr11  ?     00:00:00    [lpd]
```

As aspas forçam o shell a passar a lista de PIDs para **ps** como um argumento único.

fuser também pode relatar o uso de arquivos específicos (ao contrário de sistemas de arquivos inteiros); a sintaxe é **fuser -v** *nomeDoArquivo*. **fuser** também aceita a opção **-k** para excluir (ou enviar um sinal para) cada um dos processos problemáticos. Perigoso — e você tem de ser o root (ou utilizar **sudo**, veja a página 35).

Uma alternativa a **fuser** é o utilitário **lsof** de Vic Abell da Universidade de Purdue. Ele é executado em muitas variantes UNIX e Linux diferentes, o que o torna útil para chamar a partir de scripts que devem ser executados em uma variedade de sistemas. **lsof** é um programa mais complexo e sofisticado do que o **fuser** e sua saída é correspondentemente prolixa.

Os scripts em pesquisa de partes específicas de informações também têm a opção de ler diretamente os arquivos em **/proc**. Entretanto, **lsof -F**, que formata a saída do **lsof** para análise sintática fácil, é uma solução mais fácil e mais portável. Utilize flags de linha de comando adicionais para solicitar apenas as informações de que você precisa.

5.3 A ORGANIZAÇÃO DA ÁRVORE DE ARQUIVOS

Os sistemas de arquivos na família UNIX nunca foram muito organizados. Várias convenções de nomes incompatíveis são utilizadas ao mesmo tempo e diferentes tipos de arquivos estão espalhados aleatoriamente pelo espaço de nomes. Em muitos casos, os arquivos são divididos pela função e não pela probabilidade com que eles mudam, dificultando a atualização do sistema operacional. O diretório **/etc**, por exemplo, contém alguns arquivos que jamais são personalizados e alguns deles são inteiramente locais. Como saber quais arquivos preservar durante a atualização? Bem, você simplesmente tem de saber...

Inovações como **/var** ajudaram a solucionar alguns problemas, mas a maioria dos sistemas ainda é uma perfeita bagunça. Não obstante, há um local culturalmente correto para tudo. É particularmente importante não bagunçar a estrutura padrão da árvore de arquivos sob o Linux, pois os pacotes de software e suas ferramentas de instalação normalmente fazem suposições abrangentes em relação à localização dos arquivos (assim como fazem outros administradores de sistemas!).

Consulte o Capítulo 28 para informações adicionais sobre a configuração do kernel.

O sistema de arquivos-raiz inclui o diretório-raiz e um conjunto mínimo de arquivos e subdiretórios. O arquivo que contém o kernel reside dentro do sistema de arquivos-raiz no diretório **/boot**; seu nome normalmente inicia com **vmlinuz**.[5] Também fazem parte do sistema de arquivos-raiz os **/dev** para arquivos de dispositivo (exceto **/dev/pts**, que é montado separadamente), **/etc** para arquivos de sistema críticos, **/sbin** e **/bin** para utilitários importantes e, às vezes, **/tmp** para arquivos temporários.

Consulte página 87 para algumas das razões por que o particionamento pode ser desejável e algumas regras gerais para orientá-lo.

Os diretórios **/usr** e **/var** são também de grande importância. **/usr** é onde a maioria dos programas padrão é mantida, junto com vários outros materiais sobre inicialização, como manuais on-line e a maioria das bibliotecas. Não é estritamente necessário que **/usr** seja um sistema de arquivos separado, mas, para conveniência administrativa, geralmente é. Tanto **/usr** quanto **/var** têm de estar disponíveis para ativar o sistema a trilhar todo o caminho até o modo multiusuário.

/var hospeda diretórios de spool, arquivos de log, informações de contabilidade e vários outros itens que crescem ou mudam rapidamente e variam em cada host. Como **/var** contém arquivos de log, que estão aptos a crescer em momentos de problemas, é uma boa idéia colocar **/var** em seu próprio sistema de arquivos se isso for prático.

Os diretórios home dos usuários normalmente são mantidos em um sistema de arquivos separado, usualmente montado no diretório-raiz. Sistemas de arquivos separados também podem ser utilizados para armazenar itens volumosos como bancos de dados e bibliotecas de código-fonte.

Alguns dos diretórios padrão mais importantes são listados na Tabela 5.2 a seguir (a alternação de sombreamento nas linhas serve para melhorar a legibilidade).

5. Era comum **/boot** localizar-se em um sistema de arquivos separado, principalmente porque o kernel devia estar perto do começo do disco de inicialização para ser acessível ao BIOS. Os modernos PCs não têm mais esse problema e **/boot** com mais freqüência faz parte do sistema de arquivos-raiz.

O Filesystem Hierarchy Standard (www.pathname.com/fhs) em desenvolvimento tenta codificar, racionalizar e explicar os diretórios padrão. É um excelente recurso a consultar quando você está tentando descobrir onde colocar algo. Discutimos algumas regras e sugestões adicionais para o design de hierarquias locais na página 181.

Tabela 5.2 Diretórios padrão e seu conteúdo.

Nome do caminho	Conteúdo
/bin	Comandos necessários para a mínima operação do sistema
/boot	Kernel e arquivos necessários para carregar o kernel
/dev	Entradas de dispositivo para discos, impressoras, pseudoterminais etc.
/etc	Arquivos de configuração e inicialização críticos
/home	Diretórios iniciais (home directories) para usuários
/lib	Bibliotecas e partes do compilador C
/media	Pontos de montagem para sistemas de arquivos em mídia removível
/opt	Pacotes de software aplicativo opcional (ainda não amplamente utilizados)
/proc	Informações sobre todos os processos em execução (alguns sistemas mais novos)
/root	Diretório de usuário do superusuário (normalmente apenas **/**)
/sbin	Comandos para inicializar, reparar e recuperar o sistema
/tmp	Arquivos temporários que podem desaparecer entre as reinicializações
/usr	Hierarquia de arquivos e comandos secundários
/usr/bin	A maioria dos arquivos executáveis e comandos
/usr/include	Arquivos de cabeçalho para compilar programas em C
/usr/lib	Bibliotecas; também arquivos de suporte para programas padrão
/usr/local	Software local (software que pode ser gravado ou instalado)
/usr/local/bin	Executáveis locais
/usr/local/etc	Arquivos de configuração e comandos locais
/usr/local/lib	Arquivos de suporte locais
/usr/local/sbin	Comandos locais de manutenção de sistema linkados estaticamente.
/usr/local/src	Código-fonte para **/usr/local/***
/usr/man	Páginas man on-line
/usr/sbin	Comandos menos essenciais para administração e reparo de sistema
/usr/share	Itens que podem ser comuns a vários sistemas (somente para leitura)
/usr/share/man	Páginas man on-line (se presentes)
/usr/src	Código-fonte de pacotes de software não-locais (não amplamente utilizados)
/var	Arquivos de configuração e dados específicos ao sistema
/var/adm	Vários: logs, registros de configuração de sistema e outros itens necessários para administração do sistema
/var/log	Vários arquivos de log de sistema (em alguns sistemas)
/var/spool	Diretórios de spool para impressão, e-mails etc.
/var/tmp	Mais espaço temporário (preservado entre reinicializações)

5.4 TIPOS DE ARQUIVOS

O Linux define sete tipos de arquivos. Mesmo quando os desenvolvedores adicionam alguma coisa nova e maravilhosa à árvore de arquivos (como as informações de processo listadas em **/proc**), ela ainda tem de ser feita de maneira a se parecer com um dos sete tipos a seguir:

- Arquivos regulares
- Diretórios
- Arquivos de dispositivo de caracteres
- Arquivos de dispositivo de blocos

- Sockets de domínio local
- Pipes identificados (FIFOs)
- Links simbólicos

Você pode determinar o tipo de um arquivo existente com **ls -ld**. O primeiro caractere do relatório de saída do **ls** codifica o tipo. O exemplo a seguir demonstra que **/usr/include** é um diretório:

```
$ ls -ld /usr/include
drwxr-xr-x   27 root     root        4096  Jul 15 20:57  /usr/include
```

ls utiliza os códigos mostrados na Tabela 5.3 para representar os vários tipos de arquivos.

Tabela 5.3 Codificação de tipos de arquivos utilizados por **ls**

Tipo de arquivo	Símbolo	Criado por	Removido por
Arquivo regular	-	editores, **cp** etc.	**rm**
Diretório	d	**mkdir**	**rmdir**, **rm -r**
Arquivo de dispositivo de caracteres	c	**mknod**	**rm**
Arquivo de dispositivo de blocos	b	**mknod**	**rm**
Socket de domínio local	s	**socket**(2)	**rm**
Pipe identificado	p	**mknod**	**rm**
Link simbólico	l	**ln -s**	**rm**

Como você pode ver a partir da Tabela 5.3, **rm** é a ferramenta universal para excluir arquivos que você não quer mais. Mas como você excluiria um determinado arquivo, digamos, **-f**? É um nome de arquivo perfeitamente legítimo na maioria dos sistemas de arquivos, mas **rm -f** não funciona, pois **-f** é interpretado como um flag **rm**. A resposta é referir-se ao arquivo por um nome de caminho completo (como **./-f**) ou utilizar o argumento -- de **rm** para informá-lo de que tudo o que vier a seguir é um nome de arquivo, e não uma opção (isto é, **rm -- -f**).

Os nomes de arquivo que contêm caracteres de controle apresentam um problema semelhante já que reproduzir esses nomes a partir do teclado pode ser difícil ou impossível. Nessa situação, você pode utilizar o *shell globbing* (correspondência de padrão) para identificar quais arquivos excluir. Ao utilizar correspondência de padrões, é uma boa idéia adquirir o hábito de utilizar a opção **-i** de **rm** para fazer **rm** confirmar a exclusão de cada arquivo. Esse recurso protege você contra exclusão de qualquer arquivo "bom" que seu padrão identifique inadvertidamente. Por exemplo, para excluir um arquivo chamado **foo**<Control-D>**bar**, você utilizaria:

```
$ ls
foo?bar    foose  kde-root

$ rm -i foo*
rm: remove `foo\004bar'? y
rm: remove `foose'? n
```

Observe que **ls** mostra o caractere de controle como um ponto de interrogação, o que pode ser um pouco enganoso.[6] Se não se lembrar de que **?** é um caractere de correspondência padrão de shell e tentar utilizar **rm foo?bar**, você poderia remover mais de um arquivo (embora não nesse exemplo). **-i** é seu amigo!

Para excluir os arquivos de nomes mais complicados, talvez seja necessário executar **rm -i ***.

Outra opção para remover arquivos com nomes suspeitos e estranhos é utilizar uma interface alternativa ao sistema de arquivos como o modo dired do **emacs** ou uma ferramenta visual como Nautilus.

Arquivos regulares

Um arquivo regular é apenas uma coletânea de bytes; o Linux não impõe nenhuma estrutura sobre seu conteúdo. Os arquivos de texto, arquivos de dados, programas executáveis e bibliotecas compartilhadas são armazenados como arquivos regulares. Tanto o acesso seqüencial como o aleatório são permitidos.

---

6. **ls -b** mostra os caracteres especiais como números octais, o que pode ser útil se você precisar identificá-los especificamente. <Control-A> é 1 (\001 em octal), <Control-B> é 2 e assim por diante.

Diretórios

Um diretório contém referências com nomes para outros arquivos. Você pode criar diretórios com **mkdir** e excluí-los com **rmdir** se estiverem vazios. Você pode excluir diretórios não vazios com **rm -r**.

As entradas especiais "." e ".." referem-se ao próprio diretório e ao seu diretório pai; elas não podem ser excluídas. Como o diretório-raiz não tem nenhum diretório pai, o caminho "/.." é equivalente ao caminho "/." (e ambos são equivalentes a /).

O nome de um arquivo é armazenado dentro de seu diretório pai, não com o próprio arquivo. De fato, mais de um diretório (ou mais de uma entrada em um único diretório) pode referenciar um arquivo por vez, e as referências podem ter nomes diferentes. Esse arranjo cria a ilusão de que um arquivo existe em mais de um lugar ao mesmo tempo.

Essas referências adicionais ("links") são indistinguíveis do arquivo original; no que diz respeito ao Linux, elas são equivalentes. O Linux mantém uma contagem do número de links que apontam para cada arquivo e não libera os blocos de dados do arquivo até que seu último link tenha sido excluído. Os links não podem cruzar os limites do sistema de arquivos.

Referências desse tipo normalmente são chamadas "links físicos" (*hard links*) para distingui-los dos links simbólicos, que são descritos abaixo. Você cria links físicos com **ln** e os remove com **rm**.

É fácil lembrar a sintaxe de **ln** se você tiver em mente que ela espelha a do comando **cp**. O comando **cp oldfile newfile** cria uma cópia de **oldfile** chamada **newfile** e **ln oldfile newfile** torna o nome **newfile** uma referência adicional a **oldfile**.

É importante compreender que os links físicos não são um tipo distinto de arquivo. Em vez de definir uma "coisa" distinta chamada link físico, o sistema de arquivos simplesmente permite que mais de uma entrada de diretório aponte para um arquivo. Além do conteúdo do arquivo, os atributos subjacentes do arquivo (como as propriedades e permissões) também são compartilhados entre links.

Arquivos de dispositivo de blocos e de dispositivo de caracteres

Consulte o Capítulo 28 para informações adicionais sobre dispositivos e drivers.

Os arquivos de dispositivo permitem que os programas se comuniquem com o hardware e periféricos do sistema. Quando o kernel é configurado, os módulos que sabem como se comunicar com cada um dos dispositivos do sistema são associados.[7] O módulo para um determinado dispositivo, chamado de driver de dispositivo, cuida dos confusos detalhes do gerenciamento de dispositivos.

Os drivers de dispositivo apresentam uma interface de comunicação padrão que se parece com um arquivo regular. Quando o kernel recebe uma requisição que se refere a um arquivo de dispositivo de blocos ou caracteres, ele simplesmente passa a requisição ao driver de dispositivo apropriado. É importante, porém, distinguir *arquivos de driver* de *arquivos de dispositivo*. Os arquivos são simplesmente pontos de encontro utilizados para se comunicarem com os drivers. Eles não são os drivers em si.

Os arquivos de dispositivo de caracteres permitem que seus drivers associados façam seus próprios armazenamentos em buffer de entrada e saída. Os arquivos de dispositivo de blocos são utilizados por drivers que manipulam E/S em grandes blocos e querem que o kernel execute o armazenamento em buffer para eles. No passado, alguns tipos de hardware eram representados tanto por arquivos de dispositivo de bloco como de caracteres, mas essa configuração é rara hoje.

Os arquivos de dispositivo são caracterizados por dois números, chamados de números de dispositivos principal e secundário. O número de dispositivo principal informa ao kernel o driver a que o arquivo se refere; e o número de dispositivo secundário geralmente informa ao driver que unidade física tratar. Por exemplo, o número de dispositivo principal 6 em um sistema Linux indica o driver de porta paralela. A primeira porta paralela (**/dev/lp0**) teria o número de dispositivo principal 6 e o número de dispositivo secundário seria 0.

Os drivers podem interpretar como bem quiserem os números de dispositivo secundários que lhes são passados. Por exemplo, os drivers de fita utilizam o número de dispositivo secundário para selecionar uma densidade em que gravar fitas e determinar se a fita deve ser retrocedida quando o arquivo de dispositivo for fechado.

Você pode criar arquivos de dispositivo com **mknod** e removê-los com **rm**. Entretanto, raramente é necessário criar arquivos de dispositivo manualmente. A maioria das distribuições utiliza **udev** automaticamente para criar e remover arquivos de dispositivo à medida que o hardware é detectado pelo kernel. **udev** mantém **/dev** organizado limitando o número de arquivos de dispositivo espúrios e garantindo que os números de dispositivo atribuídos a arquivos sejam consistentes com os esperados pelo kernel. Consulte o Capítulo 28, *Drivers e o kernel*, para informações adicionais.

Um script mais antigo chamado **MAKEDEV** faz um bom backup para **udev** no caso de alguma vez você precisar criar arquivos de dispositivo manualmente. O script codifica os nomes convencionais e números de dispositivo para várias classes de dispositivos a fim de que você não precise pesquisar esses valores. Por exemplo, **MAKEDEV pty** cria os arquivos de dispositivo para pseudoterminais.

---

7. Esses módulos também podem ser carregados dinamicamente pelo kernel.

Sempre que precisar determinar os números de dispositivo primários e secundários que são utilizados pelo driver, você pode localizar essas informações na página man do driver na Seção 4 dos manuais (por exemplo, **man 4 tty**).

Sockets de domínio local

Sockets são conexões entre processos que permitem que elas se comuniquem higienicamente. O Linux fornece vários tipos de sockets, a maioria dos quais envolve o uso de uma rede. Sockets de domínio local são acessíveis somente a partir do host local e referenciados por meio de um objeto do sistema de arquivos em vez de uma porta de rede. Eles são chamados às vezes de "sockets de domínio UNIX".

Consulte o Capítulo 10 para informações adicionais sobre syslog.

Embora os arquivos de socket sejam visíveis para outros processos na forma de entradas de diretório, eles não podem ser lidos ou gravados por processos não envolvidos na conexão. Alguns recursos padrão que utilizam sockets de domínio local são o sistema de impressão, o sistema X Window e o syslog.

Sockets de domínio local são criados com a chamada de sistema **socket** e podem ser excluídos com o comando **rm** ou com a chamada de sistema **unlink** assim que eles não tiverem mais usuários.

Pipes identificados

Assim como os sockets de domínio local, os pipes identificados permitem a comunicação entre dois processos sendo executados no mesmo host. Eles também são conhecidos como "arquivos FIFO" (FIFO é a abreviatura para "*first in, first out*", isto é, "o primeiro a entrar é o primeiro a sair"). Você pode criar pipes identificados via **mknod** e removê-los com **rm**.

Links simbólicos

Um link simbólico ou *soft link* aponta para um arquivo pelo nome. Quando o kernel se depara com um link simbólico no curso de uma pesquisa de um nome de caminho, ele redireciona sua atenção para o nome de caminho armazenado como conteúdo do link. A diferença entre links físicos e links simbólicos é que o link físico é uma referência direta, enquanto um link simbólico é uma referência por nome; links simbólicos são distintos dos arquivos para os quais eles apontam.

Você cria links simbólicos com **ln -s** e os remove com **rm**. Como os links simbólicos podem conter caminhos arbitrários, eles podem referenciar arquivos em outros sistemas de arquivos ou arquivos inexistentes. Links simbólicos múltiplos também podem formar um loop.

Um link simbólico pode conter tanto um caminho absoluto como um caminho relativo. Por exemplo,

```
# ln -s archived/secure /var/log/secure
```

vincula **/var/log/secure** a **/var/log/archived/secure** a um caminho relativo. Ele cria o link simbólico **/var/log/secure** com um alvo de "**archived/secure**", como demonstrado por essa saída de **ls**:

```
$ ls -l /var/log/secure
lrwxrwxrwx 1 root root 18 2005-07-05 12:54 /var/log/secure -> archived/secure[8]
```

O diretório **/var/log** inteiro poderia ser movido para algum outro ponto sem fazer o link simbólico parar de funcionar (não que mover o diretório seja recomendável).

É um erro comum pensar que o primeiro argumento para **ln -s** tem algo a ver com seu diretório de trabalho atual. Ele *não* é interpretado como um nome de arquivo por **ln**; ele é simplesmente utilizado literalmente como destino do link simbólico.

5.5 ATRIBUTOS DE ARQUIVO

Sob o modelo de sistema de arquivos UNIX e Linux tradicional, todo arquivo tem um conjunto de nove bits de permissão que controlam quem pode ler, gravar e executar o conteúdo do arquivo. Juntos com três outros bits que afetam primariamente a operação dos programas executáveis, esses bits constituem o "modo" do arquivo.

Os 12 bits de modo são armazenados juntos com quatro bits de informações sobre o tipo de arquivo. Os quatro bits de tipo de arquivo são configurados quando o arquivo é criado pela primeira vez e não podem ser modificados, mas o proprietário do arquivo e o superusuário podem modificar os 12 bits de modo usando o comando **chmod** (change mode). Utilize **ls -l** (ou **ls -ld** para um diretório) para inspecionar os valores desses bits. Um exemplo é apresentado na página 58.

8. As permissões de arquivo que **ls** mostra para um link simbólico, lrwxrwxrwx, são valores fictícios. A permissão para criar, eliminar ou seguir o link é controlada pelo diretório contêiner, ao passo que as permissões de leitura, gravação e execução no link alvo são concedidas pelas próprias permissões do alvo. Portanto, os links simbólicos não precisam (e não têm) de informações sobre permissões próprias.

Os bits de permissão

Nove bits de permissão determinam as operações que podem ser realizadas em um arquivo e por quem. O UNIX tradicional não autoriza que permissões sejam configuradas por usuário (embora o Linux agora suporte listas de controle de acesso em todos os sistemas de arquivos importantes; consulte a página 62). Em vez disso, três conjuntos de permissões definem acesso para o usuário proprietário do arquivo, os grupos proprietários do arquivo e todos os outros usuários. Cada conjunto possui três bits: um bit de leitura, um bit de gravação e um bit de execução.

É conveniente discutir as permissões de arquivo em termos de números octais (base 8) porque cada dígito de um número octal representa três bits e cada grupo de bits de permissão consiste em três bits. Os três bits mais significativos (com valores 400, 200 e 100) controlam o acesso para o proprietário. Os próximos três (40, 20 e 10) controlam o acesso para o grupo. Os três últimos (4, 2 e 1) controlam o acesso para todos os demais ("o mundo"). Em cada trio, o bit alto é o bit de leitura, o bit intermediário é o bit de gravação e o bit baixo é o bit de execução.

Cada usuário somente pode ter um desses três conjuntos de permissão. As permissões utilizadas são aquelas mais específicas. Por exemplo, o proprietário de um arquivo tem o acesso determinado pelos bits de permissão de usuário e jamais pelos bits de permissão de grupo. É possível para as categorias "outros" e "grupo" obter mais acesso que o proprietário, embora essa configuração raramente seja utilizada.

Em um arquivo regular, o bit de leitura permite que o arquivo seja aberto e lido. O bit de gravação permite que o conteúdo do arquivo seja modificado ou truncado; entretanto, a capacidade de excluir ou renomear (ou excluir e depois recriar!) o arquivo é controlada pelas permissões em seu diretório pai, porque é aí que o mapeamento de nome para espaço de dados é realmente armazenado.

O bit de execução permite que o arquivo seja executado. Há dois tipos de arquivos executáveis: binários, que a CPU executa diretamente, e scripts, que devem ser interpretados por um shell ou algum outro programa. Por convenção, os scripts iniciam com uma linha semelhante a:

```
#!/usr/bin/perl
```

que especifica um interpretador apropriado. O shell assume que os arquivos executáveis não-binários que não especificam um interpretador sejam scripts **bash** ou **sh**.[9]

Para um diretório, o bit de execução (normalmente chamado neste contexto de bit de "pesquisa" ou de "varredura") permite que o diretório seja introduzido ou passado enquanto um nome de caminho está sendo avaliado, mas não permite que o seu conteúdo seja listado. A combinação de bits de leitura e execução permite que o conteúdo do diretório seja listado. A combinação de bits de gravação e execução permite que os arquivos sejam criados, excluídos e renomeados dentro do diretório.

Os bits setuid e setgid

Os bits com valores octais 4000 e 2000 são os bits setuid e setgid. Quando configurados em arquivos executáveis, esses bits permitem que os programas acessem arquivos e processos que, de outro modo, estariam fora dos limites para o usuário que os executa. O mecanismo setuid/setgid para executáveis é descrito na página 33.

Quando configurado em um diretório, o bit setgid faz com que arquivos recém-criados dentro do diretório pertençam ao grupo proprietário do diretório em vez de ao grupo padrão do usuário que criou o arquivo. Essa convenção torna mais fácil compartilhar um diretório de arquivos entre vários usuários, desde que todos eles pertençam a um grupo comum. Essa interpretação do bit setgid não está relacionada ao seu significado quando configurado em um arquivo executável, mas nunca há qualquer ambigüidade no que diz respeito a qual significado é apropriado.

Você também pode configurar o bit setgid em arquivos simples não-executáveis a fim de solicitar o comportamento de bloqueio especial quando o arquivo é aberto. Entretanto, nunca vimos esse recurso sendo utilizado.

O sticky bit

O bit com valor octal 1000 é chamado de *sticky bit*. Ele foi de importância histórica como um modificador para arquivos executáveis nos primeiros sistemas UNIX. Mas, esse significado do sticky bit é agora obsoleto, e os sistemas modernos o ignoram.

Se o sticky bit estiver configurado em um diretório, o sistema de arquivos não irá permitir que você exclua ou renomeie um arquivo a menos que seja o proprietário do diretório, o proprietário do arquivo ou o superusuário. Ter permissão de gravação no diretório não é o suficiente. Essa convenção ajuda a tornar diretórios como **/tmp** um pouco mais privados e seguros.

Visualizando atributos de arquivo

O sistema de arquivos mantém cerca de 40 blocos de informação distintos para cada arquivo, mas a maioria deles é útil apenas para o sistema de arquivos em si. Como administrador de sistemas, você deverá se preocupar principalmente com a

---

9. O kernel interpreta a sintaxe #! ("shebang") e atua diretamente sobre ela. Entretanto, se o interpretador não for especificado completa e corretamente, o kernel se recusará a executar o arquivo. O shell faz então uma segunda tentativa de executar o script chamando **sh**.

contagem de links, proprietário, grupo, modo, tamanho, data/hora do último acesso, data/hora da última modificação e tipo. Você pode inspecionar todos eles por meio de **ls -l** (ou **ls -ld** para um diretório).

Um atributo *change time* (data/hora da última modificação) também é mantido para cada arquivo. O nome convencional para essa data/hora (o "ctime", abreviatura de "change time") leva algumas pessoas a acreditar que ele seja a data/hora de criação do arquivo. Infelizmente, não é o caso; ele apenas registra o momento em que os atributos do arquivo (proprietário, modo etc.) foram modificados pela última vez (e não a data/hora em que o conteúdo do arquivo foi modificado).

Considere o exemplo a seguir:

```
$ ls -l /bin/gzip
-rwxr-xr-x    3 root     root        57136 Jun 15  2004 /bin/gzip
```

O primeiro campo especifica o tipo e modo do arquivo. O primeiro caractere é um traço e, portanto, trata-se de um arquivo regular. (Consulte a Tabela 5.3 na página 55 para outros códigos.)

Os próximos nove caracteres desse campo são os três conjuntos de bits de permissão. A ordem é proprietário-grupo-outros, e a ordem dos bits dentro de cada conjunto é leitura-gravação-execução. Embora esses bits tenham apenas valores binários, **ls** os exibe simbolicamente por meio das letras r, w e x de *read*, *write* e *execute*. Nesse caso, o proprietário tem todas as permissões sobre o arquivo e todos os demais somente têm permissão de leitura e execução.

Se o bit setuid tivesse sido configurado, o x representando a permissão de execução do proprietário teria sido substituído por um `s` e, se o bit setgid tivesse sido ativado, o `x` para o grupo também teria sido substituído por um `s`. O último caractere das permissões (permissão de execução para "outros") é mostrado como `t` se o sticky bit do arquivo estiver ativado. Se o bit setuid/setgid ou o sticky bit estiver ativado, mas o bit de execução correspondente não, esses bits aparecerão como `S` ou `T`.

O próximo campo na listagem é a contagem de links para o arquivo. Nesse caso ele é 3, indicando que **/bin/gzip** é apenas um dos três nomes para esse arquivo (os outros são **/bin/gunzip** e **/bin/zcat**). Sempre que é criado um link físico para um arquivo, a contagem é incrementada por 1.

Todos os diretórios têm ao menos dois links físicos: o link proveniente do diretório pai e o link proveniente do arquivo especial "." dentro do próprio diretório. Links simbólicos não afetam a contagem de links.

Os próximos dois campos na saída gerada por **ls** são o usuário proprietário e o grupo proprietário do arquivo. Nesse exemplo, o usuário proprietário do arquivo é root e o arquivo também pertence ao grupo chamado root. O sistema de arquivos na realidade armazena esses dois campos como o número de ID do usuário e o número de ID do grupo em vez de como nomes. Se as versões na forma de texto (nomes) não puderem ser determinadas, esses campos contêm números. Isso poderia acontecer se o usuário ou grupo que possui o arquivo tiver sido excluído do arquivo **/etc/passwd** ou **/etc/group**. Isso também poderia indicar um problema com o seu banco de dados NIS ou LDAP (se você utiliza um); consulte o Capítulo 17.

O próximo campo é o tamanho do arquivo em bytes. Esse arquivo tem 57.136 bytes ou aproximadamente 56 K.[10] Em seguida, vem a data da última modificação: Jun 15 2004. O último campo na listagem é o nome do arquivo, **/bin/gzip**.

A saída de **ls** é ligeiramente diferente para um arquivo de dispositivo. Por exemplo:

```
$ ls -l /dev/tty0
crw-rw----  1 root root     4, 0 Jun 11 20:41 /dev/tty0
```

A maioria dos campos é a mesma, mas, em vez de um tamanho em bytes, **ls** mostra os números de dispositivo principais e secundários. **/dev/tty0** é o primeiro console virtual, controlado pelo driver de dispositivo 4 (o driver de terminal).

Uma opção de **ls** que é útil para examinar links físicos é **-i**, que exibe o "número de inode" de cada arquivo. Sem entrar em muitos detalhes sobre a implementação do sistema de arquivos, diremos apenas que o número de inode é um índice para uma tabela que enumera todos os arquivos no sistema de arquivos. Inodes são as "coisas" que são apontadas pelas entradas de diretórios; entradas que são links físicos para o mesmo arquivo têm o mesmo inode. Para ter uma visão geral de uma teia complexa de links, você precisará utilizar **ls -li** para exibir as contagens de links e números de inode juntamente com **find** para procurar correspondências.[11]

O sistema controla automaticamente as informações referentes a registros de data/hora de modificação, contagens de links e tamanho de arquivo. Inversamente, bits de permissão, usuário proprietário e grupos proprietários somente mudam quando são especificamente alterados.

Algumas outras opções **ls** que são importantes conhecer são: **-a** para exibir todas as entradas em um diretório (até arquivos cujos nomes iniciam com um ponto), **-t** para classificar arquivos pela data/hora de modificação (ou **-tr** para classi-

---

10. K significa kilo, um prefixo de medida que significa 1.000; entretanto, os profissionais da informática degeneraram seu significado, passando a 2^{10} ou 1.024. De forma similar, um megabyte em sistemas computacionais não é realmente um milhão de bytes, mas sim 2^{20}, ou seja, 1.048.576 bytes. A International Electrotechnical Commission está promovendo um novo conjunto de prefixos numéricos (como, por exemplo, kibi- e mebi-) que se baseiam explicitamente em potências de 2. Nesta altura, parece pouco provável que o uso comum mudará. Para deixar a coisa ainda mais confusa, até as unidades de potência de 2 não são utilizadas de maneira consistente. RAM é denominada em potências de 2, mas a largura de banda é sempre uma potência de 10. O espaço de armazenamento é representado em unidades de potência de 10 pelos fabricantes e unidades de potência de 2 pelo resto do mundo.

11. Tente **find** *pontoDeMontagem* **-xdev -inum** *inode* **-print**.

ficar em ordem cronológica inversa), **-F** para mostrar os nomes de arquivos de certa maneira que distinga diretórios e arquivos executáveis, **-R** para listar recursivamente e **-h** para mostrar tamanhos de arquivo de forma legível para humanos (por exemplo, `8K` ou `53M`).

chmod: modificando permissões

O comando **chmod** modifica as permissões em um arquivo. Somente o proprietário do arquivo e o superusuário podem alterar suas permissões. Para empregarmos o comando em sistemas UNIX antigos, tínhamos que aprender um pouco sobre notação octal, mas as versões atuais aceitam tanto notação octal como uma sintaxe mnemônica. A sintaxe octal geralmente é mais conveniente para administradores, contudo ela somente pode ser utilizada para especificar um valor absoluto para os bits de permissão. A sintaxe mnemônica pode modificar alguns bits enquanto deixa outros sem modificação.

O primeiro argumento para **chmod** é uma especificação das permissões a serem atribuídas e o segundo argumento e os subseqüentes são o nome dos arquivos nos quais as permissões devem ser modificadas. No caso octal, o primeiro dígito octal da especificação é para o proprietário, o segundo é para o grupo e o terceiro para o restante. Se quiser ativar o bit setuid, o bit setgid ou sticky bit, utilize quatro dígitos octais em vez de três, com os três bits especiais formando o primeiro dígito.

A Tabela 5.4 ilustra as oito possíveis combinações para cada conjunto de três bits, onde `r`, `w` e `x` significam *read*, *write* e *execute*.

Tabela 5.4 Codificação de permissão para o comando **chmod**

Octal	Binário	Permissões	Octal	Binário	Permissões
0	000	`- - -`	4	100	`r- -`
1	001	`- - x`	5	101	`r-x`
2	010	`-w-`	6	110	`rw-`
3	011	`-wx`	7	111	`rwx`

Por exemplo, **chmod 711 myprog** dá todas as permissões ao proprietário e permissão somente para execução aos demais.[12]

Os detalhes completos da sintaxe mnemônica de **chmod** podem ser encontrados na página man do **chmod**. Alguns exemplos de especificações mnemônicas são mostrados na Tabela 5.5.

Tabela 5.5 Exemplos de sintaxe mnemônica de **chmod**

Especificação	Significado
u+w	Acrescenta permissão de gravação ao proprietário do arquivo
ug=rw,o=r	Dá permissão de leitura/gravação ao proprietário e ao grupo e permissão de leitura aos demais
a-x	Remove a permissão de execução para todas as categorias (proprietário/grupo/outros)
ug=srx,o=	Torna o arquivo setuid e setgid e dá permissão de leitura/gravação somente ao proprietário e ao grupo
g=u	Faz com que as permissões do grupo sejam as mesmas que as do proprietário

A parte realmente difícil do uso da sintaxe mnemônica é lembrar se **o** significa "owner" (proprietário) ou "other" (outros) ("other" é o correto). Lembre-se apenas de **u** e **g** por analogia a UID e GID; somente uma possibilidade restará.

Você também pode especificar os modos a serem atribuídos por analogia com um arquivo existente. Por exemplo, **chmod --reference=filea fileb** faz com que o modo de **fileb** seja igual ao de **filea**.

chmod é capaz de atualizar recursivamente as permissões dentro de um diretório com a opção **-R**. Entretanto, isto é mais capcioso do que parece, já que os arquivos e diretórios inclusos talvez não compartilhem os mesmos atributos (por exemplo, alguns poderiam ser arquivos executáveis enquanto outros poderiam ser arquivos de texto). A sintaxe mnemônica é particularmente útil com **-R**, pois quaisquer bits cujos valores não ativemos explicitamente são deixados de lado. Por exemplo,

```
$ chmod -R g+w mydir
```

acrescenta permissão de gravação a **mydir** e todo o seu conteúdo sem mexer nos bits de execução de diretórios e programas.

12. Se **myprog** fosse um script de shell, ele precisaria das permissões de leitura e execução ativadas. Para que o script seja executado por um interpretador, ele tem de ser aberto e lido como um arquivo de texto. Arquivos binários são executados diretamente pelo kernel e, portanto, não precisam ter permissão de leitura ativada.

chown: modificando posse e grupo

O comando **chown** modifica o usuário proprietário e o grupo proprietário de um arquivo. Sua sintaxe espelha a do **chmod**, exceto que o primeiro argumento especifica o novo proprietário e grupo na forma *usuário:grupo*. Você pode omitir o *usuário* ou *grupo*. Se não houver grupo, você também não precisa dos dois-pontos, embora possa incluí-lo para fazer **chown** configurar o grupo como o grupo padrão do *usuário*. A forma *usuário.grupo* também é aceita, por razões históricas, embora esteja um pouco velha já que nomes de usuário podem incluir pontos.

Para modificar o grupo de um arquivo, você tem de ser o proprietário do arquivo e pertencer ao grupo que estamos modificando ou então ser o superusuário. Você deve ser o superusuário para modificar a propriedade de um arquivo.

Como o **chmod**, **chown** oferece o flag recursivo **-R** para alterar as configurações de um diretório e todos os arquivos abaixo dele. Por exemplo, a seqüência:

```
# chmod 755 ~matt
# chown -R matt:staff ~matt
```

poderia ser utilizada para o diretório de usuário de um novo usuário depois de termos copiado os arquivos de inicialização padrão. Certifique-se de não tentar utilizar **chown** sobre os arquivos ponto do novo usuário com o de um comando como:

```
# chown -R matt matt:staff ~matt/.*
```

O padrão identificará com **~matt/..** e, portanto, acabará modificando as propriedades do diretório pai e provavelmente os diretórios iniciais dos demais usuários.

O UNIX tradicional utiliza um comando distinto, **chgrp**, para modificar o grupo proprietário de um arquivo. O Linux também tem **chgrp**. Ele funciona da mesma maneira que **chown**; fique à vontade para usá-lo caso ache mais fácil de memorizar.

umask: atribuindo permissões padrão

Você pode utilizar o comando de shell chamado **umask** para manipular as permissões padrão dos arquivos que você cria. **umask** é especificado com um valor octal de três dígitos que representa as permissões a serem revogadas. Quando um arquivo é criado, suas permissões são configuradas para que atendam a tudo que o programa criador exigir menos aquilo que **umask** proibir. Portanto, os dígitos individuais do **umask** consentem as permissões mostradas na Tabela 5.6.

Por exemplo, **umask 027** consente todas as permissões para o proprietário, mas proíbe permissão de gravação para o grupo e não dá nenhum tipo de permissão para os demais. O valor padrão de **umask** normalmente é 022, que nega permissão de gravação para o grupo e os demais, mas consente permissão de leitura.

Tabela 5.6 Codificação de permissões do comando **umask**

Octal	Binário	Permissões	Octal	Binário	Permissões
0	000	rwx	4	100	-wx
1	001	rw-	5	101	-w-
2	010	r -x	6	110	- -x
3	011	r - -	7	111	- - -

Consulte o Capítulo 6 para mais informações sobre arquivos de inicialização.

Você não pode forçar para que os usuários tenham um determinado valor **umask** porque eles podem sempre redefini-lo para o que quiserem. Mas você pode atribuir um padrão adequado nos arquivos **.cshr** e **.profile** de exemplo que você dá aos novos usuários.

Flags adicionais

Os sistemas de arquivos ext2fs e ext3fs do Linux definem alguns atributos suplementares que você pode ativar para solicitar semântica de sistema de arquivos especial (sendo "solicitar" a palavra operativa, visto que muitos dos flags não foram realmente implementados). Por exemplo, um flag torna um arquivo somente para anexação e um outro o torna imutável e indelével.

Como esses flags não se aplicam a sistemas de arquivos diferentes da série ext*, o Linux utiliza comandos especiais, **lsattr** e **chattr**, para visualizá-los e alterá-los. A Tabela 5.7 lista os flags que realmente funcionam (atualmente apenas cerca de 50% dos mencionados na página man).

Tabela 5.7 Flags adicionais do ext2fs e do ext3fs

Flag	Significado
A	Nunca atualiza a data/hora de acesso (`st_atime`; para fins de desempenho)
a	Permite gravação somente no modo acréscimo (append) (apenas root é capaz de configurá-lo)
D	Força a gravação síncrona das atualizações de diretório
d	Sem backup — faz **dump** ignorar esse arquivo
i	Torna o arquivo imutável e indelével (apenas root pode configurar)
j	Mantém um registro seqüencial (*journal*) para alterações de dados bem como de metadados
S	Força a gravação síncrona das alterações (sem armazenamento em buffer)

Com a possível exceção do flag "Sem backup", não é claro se algum desses recursos seja de muita valia no dia-a-dia. Os flags imutáveis e somente acrescentáveis foram, em grande parte, concebidos como formas de tornar o sistema mais resistente a ataques por parte de hackers ou código hostil. Infelizmente, eles podem confundir o software e somente protegem contra hackers que não sabem muito sobre o uso de **chattr –ia**.[13] A experiência prática tem demonstrado que esses flags são mais freqüentemente utilizados *pelos* hackers do que *contra* eles.

As opções **S** e **D** para gravações síncronas também merecem uma atenção especial. Como elas forçam a gravação de todas as páginas do sistema de arquivos associadas a um arquivo ou diretório imediatamente após modificações, elas poderiam parecer oferecer proteção adicional contra perda de dados no caso de uma queda do sistema. Entretanto, a ordem das operações para atualizações síncronas é incomum e tem sido conhecida por confundir o **fsck**; a recuperação de um sistema de arquivos danificado poderia ser, portanto, mais difícil em vez de mais confiável. O journaling de sistema de arquivos, como suportado por ext3fs, é, em geral, uma opção melhor. A opção **j** pode forçar um journaling de dados para arquivos específicos, embora com algum custo de desempenho.

5.6 LISTAS DE CONTROLE DE ACESSO

O sistema de controle de acesso proprietário-grupo-outros de 9 bits revelou-se poderoso o bastante para acomodar a maioria das necessidades administrativas. Embora o sistema tenha limitações claras, ele está em consonância com as tradições UNIX (alguns diriam, as "antigas tradições") de simplicidade e previsibilidade.

Praticamente todos os sistemas operacionais não-UNIX usa uma maneira substancialmente mais complicada de regular o acesso a arquivos: listas de controle de acesso, ou ACLs. As ACLs não têm um comprimento configurado, e podem incluir especificações de permissão para múltiplos usuários ou grupos. Os sistemas mais sofisticados permitem que os administradores especifiquem conjuntos parciais de permissões ou permissões negativas; alguns também têm recursos de herança que permitem que o acesso dependa de mais de uma ACL por vez. Esses sistemas são claramente mais poderosos do que o modelo UNIX tradicional, mas também são uma ordem de magnitude mais complexa, tanto para administradores como para desenvolvedores de software.

*Consulte a página 89 para informações sobre o comando **mount** e a montagem de sistema de arquivos.*

Como resultado dos esforços para incluir ACLs na especificação POSIX, muitas variantes UNIX começaram a suportar um mecanismo de ACL relativamente padrão que opera paralelamente com o modelo de 9 bits tradicional do UNIX. Sob o Linux, as ACLs são suportadas por ext2, ext3, ReiserFS, XFS e JFS. Em geral, elas estão desativadas por padrão; use a opção **-o acl** para o comando **mount** ativá-las.

Para que essa discussão seja completa, descrevemos o modelo de ACL do Linux aqui. Mas não seja seduzido pelas belas cores — as ACLs não são necessariamente "melhores" que as tradicionais permissões de arquivo; e administradores informados devem utilizá-las com certa cautela. Seu uso não só é complicado e tedioso, como também pode causar problemas com o NFS, sistemas de backup e programas como editores de textos. As ACLs tendem para a entropia e, portanto, podem tornar a manutenção impossível ao longo do tempo.

Consulte a página 568 para informações adicionais sobre o Samba.

Talvez a razão mais plausível para usar ACLs seja aumentar a compatibilidade com outros sistemas operacionais. Em particular, o Samba utilizado para compartilhamento de arquivos com sistemas Windows é compatível com ACL e faz um bom esforço para traduzir ACLs entre o Linux e o Windows.

13. O mecanismo da capacidade descrita na página 472 pode dificultar o desligamento desses bits, mas o recurso não é amplamente utilizado.

Visão geral das ACLs

As ACLs do Linux são uma extensão simples e direta em sua maior parte do modelo de 9 bits padrão. As permissões de leitura, gravação e execução são as únicas capacidades com as quais o sistema lida. Adornos como o bit setuid e sticky bit são tratados exclusivamente pelos bits de modo tradicional.

As ACLs permitem que os bits rwx sejam configurados independentemente para qualquer combinação de usuários e grupos. A Tabela 5.8 mostra como são as entradas individuais em uma ACL.

Os usuários e grupos podem ser identificados por nome ou por UID/GID. O número exato de entradas que uma ACL pode conter varia com a implementação do sistema de arquivos e vão desde um valor baixo de 25 no XFS a um número praticamente ilimitado no ReiserFS e no JFS. Os sistemas de arquivos ext2 e ext3 permitem 32 entradas, o que é provavelmente um limite razoável para gerenciabilidade em qualquer caso.

O comando **getfacl** exibe a ACL atual de um arquivo e o **setfacl** a modifica ou configura. Utilize **setfacl –b** *arquivo* para limpar a ACL, **setfacl -m** *especificaçãoDaACL arquivo* para modificá-la ou estendê-la e **setfacl -x** *especificaçãoDaACL arquivo* para excluir entradas específicas na lista. (Omita a parte da especificação de permissão da *especificaçãoDaACL* quando usar **-x**.) A *especificaçãoDaACL* pode conter mais de uma entrada de lista contanto que as entradas sejam separadas por uma vírgula.

Tabela 5.8 Entradas que podem aparecer em uma lista de controle de acesso

Formato	Exemplo	Configura permissões para
user : : *perms*	`user::rw-`	O proprietário do arquivo
user : *nomedousuário* : *perms*	`user:trent:rw-`	Um usuário específico
group : : *perms*	`group::r-x`	O grupo que possui o arquivo
group : *nomedogrupo*: *perms*	`group:staff:rw-`	Um grupo específico
other : : *perms*	`other::---`	Todos os outros
mask : : *perms*	`mask::rwx`	Tudo, exceto o proprietário e os outros[a]

a. As máscaras um pouco difíceis e são explicadas mais adiante nesta seção.

Os arquivos com ACLs retêm seus bits de modo original, mas a consistência é automaticamente imposta e os dois conjuntos de permissão nunca podem entrar em conflito. O exemplo a seguir demonstra que as entradas de ACL são automaticamente atualizadas em resposta a alterações feitas com **chmod**:

```
$ touch /tmp/example
$ ls -l /tmp/example
-rw-rw-r--  1 garth  garth     0 Jun 14 15:57 /tmp/example
$ getfacl /tmp/example
getfacl: Removing leading '/' from absolute path names
# file: tmp/example
# owner: garth
# group: garth
user::rw-
group::rw-
other::r--
$ chmod 640 /tmp/example
$ getfacl --omit-header /tmp/example
user::rw-
group::r--
other::---
```

A imposição dessa consistência permite que softwares mais antigos sem compatibilidade com ACLs executem razoavelmente bem no mundo ACL. Mas há uma "pegadinha". Embora a entrada ACL group:: no exemplo acima pareça estar monitorando o conjunto intermediário dos bits de modo tradicional, esse nem sempre será o caso.

Para entender o motivo, suponha que um programa legado limpe os bits de gravação dentro dos três conjuntos de permissão do modo tradicional (por exemplo, **chmod ugo-w** *arquivo*). A intenção é claramente tornar o arquivo não-gravável por qualquer pessoa. Mas e se a ACL resultante fosse assim?

```
user::r--
group::r--
group:staff:rw-
other::r--
```

Da perspectiva de programas legados, o arquivo parece ser não-modificável; contudo, ele é de fato gravável por qualquer pessoa do grupo. Isso é não bom. Para reduzir a chance de ambigüidade e mal-entendidos, o Linux adotou as seguintes regras:

- As entradas ACL user:: e other:: são, por definição, idênticas aos bits de permissão "owner" e "other" do modo de arquivo tradicional. Alterar o modo altera as entradas ACL correspondentes e vice-versa.
- Em todo caso, a permissão de acesso efetiva dada ao proprietário do arquivo e usuários não mencionados de outra maneira é aquela especificada na entrada ACL user:: e other::, respectivamente.
- Se um arquivo não tiver definido a ACL explicitamente ou tiver uma ACL que consiste somente em uma entrada user::, uma group:: e uma other::, essas entradas ACL serão idênticas aos três conjuntos de bits de permissão tradicionais. Esse é o caso ilustrado no exemplo de **getfacl** acima. (Essa ACL é chamada de "mínima" e não precisa ser realmente implementada como uma ACL logicamente separada.)
- Em ACLs mais complexas, os bits de permissão de grupo tradicionais correspondem a uma entrada ACL especial chamada de "máscara" em vez da entrada ACL group::. A máscara limita o acesso que a ACL pode conferir a *todos* os usuários identificados, *todos* os grupos identificados *e* o grupo padrão.

Em outras palavras, a máscara especifica um limite superior sobre o acesso que a ACL pode atribuir a grupos e usuários individuais. Ela é conceitualmente semelhante a **umask**, exceto que a máscara ACL está sempre em vigor e especifica as permissões concedidas em vez das permissões negadas. As entradas ACL para usuários identificados, grupos identificados e o grupo padrão podem incluir bits de permissões que não estão presentes na máscara, mas o kernel simplesmente os ignora.

Como resultado, os bits de modo tradicional nunca podem subestimar o acesso permitido pela ACL como um todo. Além disso, limpar um bit da parte grupo do modo tradicional limpa o bit correspondente na máscara ACL e assim proíbe essa permissão a todos, exceto ao proprietário do arquivo e àqueles que entram na categoria "other".

Quando a ACL mostrada no exemplo anterior é expandida para incluir entradas para um usuário e grupo específico, **setfacl** fornece automaticamente uma máscara apropriada:

```
$ setfacl -m user::r,user:trent:rw,group:admin:rw /tmp/example
$ ls -l /tmp/example
-r--rw----+ 1 garth  staff    0 Jun 14 15:57 /tmp/example
$ getfacl --omit-header /tmp/example
user::r--
user:trent:rw-
group::r--
group:admin:rw-
mask::rw-
other::---
```

Como visto aqui, **setfacl** gera uma máscara que permite que todas as permissões concedidas na ACL entrem em vigor. Se quiser configurar a máscara manualmente, inclua a máscara na lista de entrada ACL dada a **setfacl** ou utilize a opção **-n** para evitar **setfacl** de regenerá-la.

Quando o acesso é tentado, o UID efetivo é comparado com o UID que possui o arquivo. Se eles forem os mesmos, o acesso é determinado pelas permissões user:: na ACL. Caso contrário, se houver uma entrada ACL específica ao usuário correspondente, as permissões serão determinadas por essa entrada em combinação com a máscara ACL. Se nenhuma entrada específica ao usuário estiver disponível, o sistema de arquivos tentará localizar uma entrada válida relacionada com o grupo que forneça o acesso solicitado; essas entradas também são processadas em conjunção com a máscara ACL. Se nenhuma entrada correspondente puder ser localizada, a entrada other:: prevalecerá.

Se você utilizar o comando **chmod** tradicional para manipular as permissões de grupo sobre um arquivo com ACL configurado, esteja ciente de que suas alterações afetam apenas a máscara. Para continuar o exemplo anterior:

```
$ chmod 770 /tmp/example
$ ls -l /tmp/example
-rwxrwx---+ 1 garth  staff    0 Jun 14 15:57 /tmp/example
$ getfacl --omit-header /tmp/example
user::rwx
user:trent:rw-
```

```
group::r--
group:admin:rw-
mask::rwx
other::---
```

A saída de **ls** nesse caso é enganosa. Apesar de as permissões de grupo serem aparentemente generosas, ninguém realmente tem permissão de executar o arquivo em razão da sua posse pelo grupo. Para conceder essa permissão, você deve editar a própria ACL.

Entradas padrão

Além dos tipos de entrada ACL listados na Tabela 5.8, as ACLs de diretórios podem incluir entradas "padrão" que são propagadas para as ACLs de arquivos e subdiretórios recém-criados dentro deles. Os subdiretórios recebem essas entradas tanto na forma de entradas ACL ativas como na forma de padrões. Portanto, os padrões originais podem por fim se propagar para baixo por várias camadas da hierarquia de diretórios.

A conexão entre a ACL pai e as ACLs filhas não continuam, uma vez que as entradas padrão foram copiadas. Se as entradas padrão do pai mudarem, as alterações não serão refletidas nas ACLs de subdiretórios existentes.

5.7 EXERCÍCIOS

E5.1 O que é uma **umask**? Crie uma **umask** que não dê nenhuma permissão ao grupo ou aos demais usuários.

E5.2 Qual a diferença entre links físicos (*hard links*) e links simbólicos (*soft links*)? Quando é aconselhável utilizar um tipo ou o outro?

★ E5.3 Leia a página de manual referente ao arquivo **/etc/fstab**. Escreva uma entrada que monte automaticamente uma partição NTFS Windows, **/dev/hda1**, na inicialização. Utilize o ponto de montagem **/mnt/win_c**.

★ E5.4 Ao instalar um sistema Linux, é importante particionar a unidade de disco de tal maneira que cada sistema de arquivos (**/var**, **/usr** etc.) tenha espaço adequado tanto para as necessidades atuais como para as futuras. A distribuição "Foobar Linux" utiliza os seguintes padrões:

/	100MB
/var	50MB
/boot	10MB
<swap>	128MB
/usr	espaço restante

Quais são alguns problemas em potencial com esse arranjo em um servidor ocupado?

★ E5.5 Por que é uma boa idéia colocar algumas partições (como **/var**, **/home** e **swap**) em um drive separado dos demais arquivos de dados e programas? E quanto a **/tmp**? Dê razões específicas para cada um dos sistemas de arquivos listados.

★ E5.6 Escreva um script que localize todos os links físicos em um sistema de arquivos.

★ E5.7 Forneça comandos para realizar as seguintes tarefas:

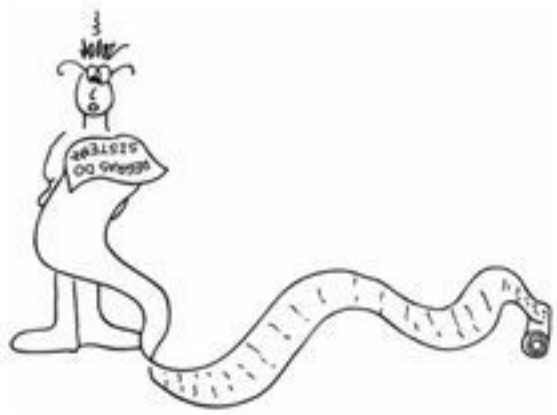
a) Configurar as permissões no arquivo **README** como de leitura/gravação para o proprietário e de leitura para os demais.

b) Ativar o bit setuid de um arquivo sem modificar (ou conhecer) as permissões atuais.

c) Listar o conteúdo do diretório atual, ordenando-o conforme a data/hora de modificação e listando por último os modificados mais recentemente.

d) Modifique o grupo de um arquivo chamado **shared** de "user" para "friends".

6 Adicionando novos usuários



Adicionar e remover usuários é uma tarefa de rotina na maioria dos sistemas. Essas tarefas são simples, porém, entediantes; a maioria dos administradores constrói ferramentas para automatizar esse processo e então delega o trabalho real a um assistente ou operador.

Hoje em dia estamos vendo um ressurgimento dos servidores centralizados com contas de login para centenas de pessoas e também servidores distribuídos com um número de usuários tão baixo quanto dois. Os administradores precisam ter um entendimento completo do sistema de contas para poder gerenciar serviços de rede e configurar contas de maneira apropriada para o ambiente computacional local.

A higiene de contas também é um determinante fundamental da segurança do sistema. Contas usadas com pouca freqüência são os principais alvos para hackers, assim como as contas com senhas facilmente adivinháveis. Mesmo que você use ferramentas automatizadas do sistema para acrescentar e eliminar usuários, é importante compreender as mudanças que essas ferramentas estão realizando.

Neste capítulo, examinaremos primeiro o modelo subjacente que as ferramentas automatizadas implementam, depois descreveremos as ferramentas em si (**useradd**, **userdel** etc.). A ferramenta padrão **useradd** é realmente muito boa e deve ser suficiente para atender às necessidades da maioria das instalações. Infelizmente, **userdel** não é tão completa quanto gostaríamos que ela fosse.

6.1 O ARQUIVO /ETC/PASSWD

O arquivo **/etc/passwd** representa uma lista de usuários reconhecidos pelo sistema. O sistema consulta o arquivo em tempo de login para determinar o UID e o diretório inicial (*home directory*) de um usuário, entre outras coisas. Cada linha no arquivo indica um usuário e contém sete campos separados por sinais de dois-pontos (:):

- Nome de login
- Senha criptografada ou marcador de lugar de senha (ver página 68)
- Número de UID (ID de usuário)
- Número de GID padrão (ID de grupo)
- Informações "GECOS": nome completo, empresa, ramal, telefone residencial
- Diretório inicial
- Shell de login

Por exemplo, as linhas a seguir são todas as entradas **/etc/passwd** sintaticamente válidas:

```
root:lga5FjuGpZ2so:0:0:The System,,x6096,:/:/bin/sh
jl:x:100:0:Jim Lane,ECT8-3,,:/staff/jl:/bin/sh
dotty:$1$Ce8QpAQI$L.DvJEWiHlWetKTMLXFZO/s93ZiY:101:20:  :/home/dotty:/bin/tcsh
```

Atualmente, não é aceitável deixar senhas encriptadas à vista. Com hardware rápido, elas podem ser "quebradas" em minutos. Todas as versões do UNIX e do Linux permitem ocultar as senhas encriptadas colocando-as em um arquivo separado que não é legível. Isso é conhecido como mecanismo de senha sombra e é (apropriadamente) o padrão na maioria dos sistemas.

O sistema de senhas sombras faz mais sentido quando explicado como uma extensão do tradicional **/etc/passwd** (como ele era historicamente), de modo que iremos protelar a discussão sobre esse recurso até a página 70. Uma discussão mais genérica sobre implicações de segurança das senhas sombras pode ser encontrada na página 469.

O conteúdo de **/etc/passwd** é normalmente compartilhado entre sistemas com um banco de dados como NIS ou LDAP. Veja o Capítulo 17, *Compartilhando arquivos de sistema*, para obter maiores informações.

As seções a seguir discutem os campos de **/etc/passwd** de maneira mais detalhada.

Nome de login

Consulte a página 355 para obter maiores informações sobre o NIS.

Os nomes de login (também conhecidos como nomes de usuário) devem ser únicos e, dependendo do sistema operacional, normalmente não podem ter mais de 8 ou 32 caracteres. Eles podem conter quaisquer caracteres, exceto dois-pontos e caractere de nova linha. Se você utiliza NIS, os nomes de login são limitados a 8 caracteres independentemente do sistema operacional.

Algumas versões mais antigas do UNIX limitam os caracteres permissíveis a caracteres alfanuméricos e impõem um limite de comprimento de 8 caracteres. Em sites heterogêneos, uma boa idéia é considerar os limites mais restritivos. Tal política evitará possíveis conflitos com softwares mais antigos e garantirá que os usuários possam ter o mesmo nome de login em cada uma das máquinas. Lembre-se, só por que você tem um ambiente homogêneo hoje não significa que isso será verdade no futuro.

Os nomes de login diferenciam letras maiúsculas de minúsculas; entretanto, a RFC822 requer que a diferenciação entre maiúsculas e minúsculas seja ignorada nos endereços de e-mail. Não sabemos de nenhum caso de problemas causados por mistura de caixas em nomes de login, porém os nomes minúsculos são tradicionais e mais fáceis de serem digitados.

Nomes de login devem ser fáceis de ser memorizados, de modo que seqüências aleatórias de letras não criam bons nomes de login. Sugerimos evitar apelidos, mesmo se sua empresa for relativamente informal. Eles não são tão divertidos assim e tendem a provocar gozações; nomes como DarkLord e QTPie aparecem na frente de @hotmail.com. Mesmo que os seus usuários não tenham auto-estima, pense pelo menos na credibilidade geral de sua instalação.

Como os nomes de login são normalmente utilizados como endereços de e-mail, é útil estabelecer uma maneira padrão de formá-los. Deveria ser possível para os usuários fazer suposições baseadas em fatos. Nomes, sobrenomes, iniciais ou alguma combinação destes são esquemas de nomes razoáveis.

Consulte a página 377 para informações adicionais sobre aliases de correio.

Qualquer esquema fixo para seleção de nomes de logins acaba resultando em nomes duplicados ou nomes que são muito longos, de modo que às vezes você terá de abrir exceções. No caso de um nome longo, você pode utilizar os recursos de aliases do sistema de correio para comparar duas versões do nome, pelo menos no que diz respeito ao correio.

Suponha, por exemplo, que você use a primeira letra do nome e o sobrenome de um funcionário como paradigma. Assim, Brent Browning seria bbrowning, que tem 9 caracteres e, portanto, potencialmente incompatível com alguns sistemas. Em vez disso, você poderia atribuir ao usuário o login brentb, deixando bbrowning como uma entrada do arquivo **aliases**:

```
bbrowning: brentb
```

Se sua instalação tiver um arquivo de aliases de e-mail global, cada novo nome de login deve ser distinto de qualquer alias neste arquivo. Caso contrário, a mensagem será enviada ao nome alternativo, e não ao novo usuário.

É comum grandes instalações implementarem um esquema de endereços de e-mail com nomes completos (por exemplo, John.Q.Public@mysite.com) que oculta nomes de login do mundo externo. Essa é uma boa idéia, mas realmente não elimina nenhum dos conselhos sobre atribuição de nomes dados anteriormente. Mesmo que por nenhuma outra razão, a não ser a sanidade dos administradores, é melhor que os nomes de login tenham uma correspondência clara e previsível com os verdadeiros nomes dos usuários.

Nomes de login devem ser únicos em dois sentidos. No primeiro deles, um usuário deve ter o mesmo nome de login em todas as máquinas. Essa regra é seguida mais por conveniência, tanto para a sua como para o do usuário.

Consulte a página 473 para uma discussão de questões de equivalência de login.

Em segundo lugar, um determinado nome de login deve sempre se referir à mesma pessoa. Alguns comandos (como **ssh**) podem ser configurados para validar usuários remotos tomando com base seus nomes de login. Mesmo que `scott@boulder` e `scott@refuge` fossem duas pessoas diferentes, uma poderia ser capaz de entrar na conta da outra sem fornecer uma senha, caso os sistemas não fossem configurados apropriadamente.

A experiência também mostra que nomes duplicados podem induzir a confusão de e-mails. O sistema de e-mail poderia perfeitamente esclarecer de qual `scott` estamos tratando, porém os usuários normalmente enviariam mensagens para o endereço errado.

Senha criptografada

Um rápido lembrete antes de entrarmos de cabeça nos detalhes das senhas: a maioria dos sistemas agora mantém senhas criptografadas em **/etc/shadow**, e não em **/etc/passwd**. Entretanto, os comentários desta seção se aplicam independentemente de onde as senhas sejam efetivamente mantidas.

As senhas são armazenadas de uma forma encriptada. A menos que você consiga executar algoritmos de criptografia em sua cabeça (queremos conhecê-lo!), você terá de configurar senhas utilizando o comando **passwd** (**yppasswd** caso utilize NIS) ou copiando uma string de senha criptografada de uma outra conta.

Se editar **/etc/passwd** manualmente para criar uma nova conta, coloque um asterisco ou um x no campo da senha criptografada. O asterisco impede o uso desautorizado da conta até que você tenha configurado uma senha real. Jamais deixe esse campo vazio — isso gera uma brecha enorme em termos de segurança, pois não é necessária nenhuma senha para acessar a conta. Mesmo se você estiver utilizando senhas sombras, é inteligente ser um pouco meticuloso quanto à higiene de senhas no arquivo **/etc/passwd**. Você nunca sabe quando algum programa ou script obsoleto irá espiá-lo para tomar algum tipo de decisão de segurança.[1]

As principais distribuições do Linux reconhecem múltiplos métodos de criptografia de senha e elas podem determinar o método de criptografia utilizado para cada senha examinando os dados criptografados. Não é necessário que todas as senhas do sistema utilizem a mesma forma de criptografia.

A maioria dos sistemas que utilizam as distribuições Linux assume o padrão do uso da criptografia DES padrão MD5. O MD5 é criptograficamente um pouco melhor que o padrão DES anterior; e o esquema MD5 permite senhas de comprimento arbitrário. Senhas mais longas são mais seguras — se você realmente as utilizar. Como o uso do MD5 não prejudicará, e talvez ajude, nós o recomendamos para todos os sistemas que o suportam.

Senhas criptografadas têm comprimento constante (34 caracteres para MD5, 13 para DES) independentemente do comprimento da senha descriptografada. As senhas são criptografadas em combinação com um "tempero" aleatório para que uma dada senha possa corresponder a várias formas criptografadas distintas. Se por acaso dois usuários selecionarem a mesma senha, esse fato normalmente não poderá ser descoberto por inspeção das senhas criptografadas. As senhas MD5 são fáceis de ser localizadas, pois sempre iniciam com um 1.

O SUSE assume o padrão da criptografia Blowfish para senhas novas. Como ocorre com o MD5, esse é um algoritmo forte e um padrão muito razoável. Entretanto, você não pode copiar as senhas Blowfish do SUSE para sistemas não SUSE uma vez que apenas o SUSE entende essas senhas. Você pode identificar senhas Blowfish pelo seu prefixo de $2a$.

Número de UID (ID de usuário)

UIDs são inteiros de 32 bits sem sinal. Entretanto, devido a questões de interoperabilidade com sistemas mais antigos, sugerimos limitar o maior UID no seu site a 32.767 (o maior inteiro de 16 bits com sinal), se possível.

Por definição, root tem o UID 0. A maioria dos sistemas também define um bin de pseudo-usuários, daemon e vários outros. É comum colocar esses logins falsos no começo do arquivo **/etc/passwd** e atribuir UIDs baixos; nunca atribua a esses logins um shell real. Para ter bastante espaço para quaisquer usuários não humanos que você queira adicionar no futuro, recomendamos atribuir UIDs aos usuários reais iniciando em 500 (ou maior).

Consulte a página 35 para obter maiores informações sobre ***sudo****.*

Nunca será uma boa idéia ter várias contas com UID 0. Embora possa parecer conveniente ter vários logins root com diferentes shells e/ou senhas, essa configuração apenas cria mais brechas de segurança em potencial e faz com que você tenha de proteger vários logins. Se as pessoas precisam ter maneiras alternativas para se conectar como root, é melhor que elas utilizem um programa como o **sudo**.

Evite reaproveitar UIDs pelo maior tempo possível, mesmo que sejam UIDs de pessoas que tenham saído da organização e tenham suas contas eliminadas permanentemente. Essa precaução impede confusão caso arquivos sejam restaurados posteriormente de backups, em que os usuários podem ser identificados pelo UID em vez de por um nome de login.

Consulte o Capítulo 16 para informações adicionais sobre backups.

Os UIDs devem ser mantidos exclusivos por toda a empresa. Ou seja, um determinado UID deve referir-se ao mesmo nome de login e à mesma pessoa em cada máquina. A falha em manter UIDs distintos pode resultar em problemas de segurança com sistemas como o NFS e também pode criar confusão quando um usuário se transferir de um grupo a outro.[2]

Pode ser difícil manter UIDs exclusivos quando grupos de máquinas são administrados por pessoas diferentes ou empresas. Os problemas podem ser tanto técnicos como políticos. A melhor solução é ter um banco de dados centralizado contendo um registro para cada usuário e reforçar a exclusividade. (Utilizamos um banco de dados doméstico para resolver esse problema.) Um esquema mais simples é atribuir a cada grupo dentro de uma empresa um intervalo de UIDs e deixar que cada grupo gerencie o seu próprio conjunto. Essa solução mantém os espaços de UID separados (uma exigência caso vá utilizar NFS para compartilhar sistemas de arquivos), porém não atende à questão paralela de nomes de login exclusivos. O LDAP também está se tornando uma ferramenta popular de gerenciamento de UIDs.

---

1. Jon Corbet, um de nossos revisores técnicos, comentou: "Se você não sabe quando decisões de segurança estão sendo tomadas, você já tem um problema. Os administradores não devem ficar surpresos com esse tipo de coisa".
2. Uma outra questão relacionada ao NFS é o UID "nobody", tradicionalmente utilizado para impedir acesso por usuários root remotos. Veja a página 339 para obter detalhes.

Número GID Padrão

Consulte a página 58 para informações adicionais sobre diretórios setgid.

Como ocorre com um UID, um número de ID de grupo é um inteiro de 32 bits. O GID 0 é reservado para o grupo chamado "root". O GID 1 é o grupo "bin" e o GID 2 é o grupo "daemon".

Os grupos são definidos em **/etc/group**, com o campo GID em **/etc/passwd** fornecendo o GID padrão (ou "efetivo") no momento do login. O GID padrão não é tratado especialmente quando o acesso é determinado;[3] é relevante somente para a criação de arquivos e diretórios novos. Arquivos novos normalmente são de propriedade do grupo efetivo do usuário. Entretanto, em diretórios nos quais o bit setgid (02000) tiver sido ativado e em sistemas de arquivos montados com a opção **grpid**, arquivos novos terão como padrão o grupo de seus diretórios pai.

Campo GECOS[4]

O campo GECOS é comumente utilizado para registrar informações pessoais sobre cada usuário. Ele não possui uma sintaxe bem definida. O campo GECOS originalmente armazenava as informações de login necessárias para transferir tarefas de sistemas UNIX na Bell Labs para um mainframe executando o sistema operacional GECOS (General Electric Comprehensive Operating System); hoje, somente o nome permanece. Alguns programas expandirão um '&' no campo GECOS para o nome de login do usuário, o que evita um pouco de digitação. Tanto **finger** como **sendmail** realizam essa expansão, mas muitos programas não. É melhor não contar com esse recurso.

Embora possamos utilizar as convenções de formatação que quisermos, **finger** interpreta entradas GECOS separadas por vírgulas na seguinte ordem:

- Nome completo (normalmente o único campo utilizado)
- Prédio e número do escritório
- Ramal do telefone comercial
- Telefone residencial

Consulte a página 361 para obter maiores informações sobre LDAP.

O comando **chfn** permite aos usuários modificar suas próprias informações GECOS. **chfn** é útil para manter atualizados itens como números de telefones, porém ele pode ser utilizado de maneira abusiva: um usuário pode alterar as informações para que se tornem obscenas ou incorretas. A maioria dos campi universitários desativa o **chfn**. Informações GECOS são o candidato perfeito para o uso do LDAP.

Diretório inicial

Shells dos usuários são posicionados de acordo com seus diretórios iniciais quando eles efetuam o login. Se o diretório inicial do usuário estiver ausente em tempo de login, o sistema imprimirá uma mensagem como "no home directory" (nenhum diretório inicial)".[5] Se `DEFAULT_HOME` for configurado como `no` em **/etc/login.defs**, o login não terá permissão de continuar; do contrário, o usuário será colocado no diretório-raiz.

Esteja ciente de que se os diretórios são montados por meio de um sistema de arquivos de rede, eles podem estar indisponíveis no caso de problemas de rede ou servidor.

Shell de login

O shell de login normalmente é um interpretador de comandos como o shell Bourne ou o shell C (**/bin/sh** ou **/bin/csh**), porém pode ser qualquer programa. **bash** é o padrão e é utilizado se **/etc/passwd** não especificar um shell de login. Em sistemas Linux, **sh** e **csh** são realmente apenas links para **bash** (o shell GNU "Bourne novamente") e **tcsh** (um superconjunto do shell C), respectivamente. A maioria das distribuições também fornece uma versão de domínio público do shell Korn, **ksh**.

Os usuários podem alterar seus shells com o comando **chsh**. O arquivo **/etc/shells** contém uma lista de shells "válidos" que **chsh** permitirá que os usuários selecionem; SUSE impõe essa lista, mas o Red Hat apenas o alerta caso o shell seleciona-

---

3. O Linux considera todas as posses de grupo quando faz cálculos de acesso. Kernels anteriores à versão 2.6.4 permitem um máximo de 32 posses de grupo, mas os kernels mais recentes não impõem restrições.
4. Quando a Honeywell assumiu a divisão de computação da GE, o GECOS foi alterado para GCOS; as duas ortografias sobrevivem hoje.
5. Essa mensagem aparece quando você faz login no console ou em um terminal, mas não aparece ao fazer login por meio de um gerenciador de exibição como **xdm**, **gdm** ou **kdm**. Você não só não verá a mensagem, como também em geral será colocado fora do sistema imediatamente devido à incapacidade do gerenciador de exibição gravar no diretório apropriado (por exemplo, **~/.gnome**).

do não se encontre na lista. Se você acrescentar entradas ao arquivo **shells**, certifique-se de utilizar caminhos absolutos já que o **chsh** e outros programas têm a expectativa de usá-lo.

6.2 O ARQUIVO /ETC/SHADOW

O arquivo **/etc/shadow** só é legível pelo superusuário e serve para manter senhas criptografadas protegidas contra o acesso não-autorizado. Ele também fornece informações sobre contas que não estão disponíveis em **/etc/passwd**. O emprego de senhas sombras é padrão em algumas distribuições e configurado como pacote opcional em outros. Mesmo quando as senhas sombras são opcionais, é uma boa idéia tratá-las como se elas fossem padrão.

Quando as senhas sombras estão em uso, os campos de senha de estilo antigo contidos em **/etc/passwd** devem sempre conter um x.

O arquivo **shadow** não é um superconjunto do arquivo **passwd**, e o arquivo **passwd** não é gerado dele; você tem de manter ambos os arquivos (ou utilizar ferramentas como **useradd** que mantêm ambos em seu nome). Como **/etc/passwd**, **/etc/shadow** contém uma linha para cada usuário. Cada linha contém nove campos, separados por dois-pontos (:):

- Nome de login
- Senha criptografada
- Data da última mudança de senha
- Número mínimo de dias entre mudanças de senha
- Número máximo de dias entre mudanças de senha
- Número de dias antecipados para alertar os usuários sobre a expiração de suas senhas
- Número de dias após a expiração da senha que a conta será desabilitada
- Data de expiração da conta

Consulte a página 77 para informações adicionais sobre usermod.

Os únicos campos em que se exige que não estejam vazios são os referentes a nome de usuário e senha. Campos de datas absolutos em **/etc/shadow** são especificados em termos de dias (*não* em segundos) desde 1° de janeiro de 1970, que não é uma maneira-padrão de reconhecimento de horários em sistemas UNIX. Felizmente, podemos utilizar o programa **usermod** para configurar o campo de expiração.

Uma entrada **shadow** típica seria parecida com o seguinte:

```
millert:$1$buJ6v3Ch$BwLIoF5eaCh9Nv.OEzD3T0:13348:0:180:14::14974:
```

Eis uma descrição mais completa de cada campo:

- O nome de login é o mesmo que em **/etc/passwd**. Esse campo conecta as entradas **passwd** e **shadow** de um usuário.
- A senha criptografada é idêntica em conceito e execução àquela previamente armazenada em **/etc/passwd**.
- O campo da última mudança registra o momento no qual a senha do usuário foi modificada pela última vez. Esse campo geralmente é preenchido por **passwd**.
- O quarto campo configura o número de dias que devem decorrer entre mudanças de senha. A idéia é forçar alterações autênticas impedindo que os usuários revertam imediatamente para uma senha familiar depois de uma alteração requerida. Entretanto, pensamos que esse recurso pode ser um pouco perigoso quando ocorre uma invasão de segurança. Recomendamos configurar esse campo com 0.
- O quinto campo configura o número máximo de dias permitido entre mudanças de senha. Esse recurso permite ao administrador forçar a expiração de senhas; consulte a página 470 para obter informações adicionais. O verdadeiro número máximo de dias imposto para expiração é a soma desse e do sétimo campo (período de tolerância).
- O sexto campo configura o número de dias antes da expiração da senha que o programa **login** deve começar a alertar o usuário da expiração iminente.
- O sétimo campo especifica o número máximo de dias que se pode esperar após o limite de existência de uma senha ter sido atingido antes de tratar o login como expirado. O propósito exato desse recurso não está claro.
- Sob o Linux, o sétimo campo tem um significado completamente diferente: ele especifica o número máximo de dias que se pode esperar após o limite de existência de uma senha ter sido atingido antes de tratar o login como expirado.
- O oitavo campo especifica o dia (a partir de 1° de janeiro de 1970) no qual uma conta de usuário irá expirar. Talvez o usuário não consiga conectar-se após essa data até que o campo tenha sido reinicializado por um administrador. Se o campo for deixado em branco, a conta jamais expirará.
- O nono campo é reservado para uso futuro.

Agora que sabemos o significado de cada um desses campos, vejamos nosso exemplo novamente:

```
millert:$1$buJ6v3Ch$BwLIoF5eaCh9Nv.OEzD3T0:13348:0:180:14::14974:
```

Nesse exemplo, o usuário millert alterou sua senha pela última vez em 18 de julho de 2006. A senha tem de ser modificada novamente dentro de 180 dias, e millert receberá alertas de que a senha precisa ser modificada nas duas últimas semanas desse período. A conta expira em 31 de dezembro de 2010.

Podemos utilizar o utilitário **pwconv** para reconciliar o conteúdo do arquivo **shadow** com aquele do arquivo **passwd**, selecionando quaisquer novas adições e eliminando usuários que não se encontrem mais listados em **passwd**. **pwconv** preenche a maioria dos parâmetros shadow por meio dos padrões definidos em **/etc/login.defs**.

O exemplo a seguir ilustra o formato do arquivo **login.defs**. Os comentários realizam um bom trabalho ao explicar os vários parâmetros. Esse exemplo particular é de um sistema Fedora; o conteúdo padrão varia substancialmente entre distribuições e também os parâmetros que podem ser especificados.

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail

# QMAIL_DIR        Maildir
MAIL_DIR           /var/spool/mail
# MAIL_FILE       .mail

# Password aging controls:
# PASS_MAX_DAYS   Maximum # of days a password may be used.
# PASS_MIN_DAYS   Minimum # of days allowed between password changes.
# PASS_MIN_LEN    Minimum acceptable password length.
# PASS_WARN_AGE   Number of days warning given before a password expires.

PASS_MAX_DAYS     99999
PASS_MIN_DAYS     0
PASS_MIN_LEN      5
PASS_WARN_AGE     7

# Min/max values for automatic uid selection in useradd

UID_MIN           500
UID_MAX           60000

# Min/max values for automatic gid selection in groupadd

GID_MIN           500
GID_MAX           60000

# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).

# USERDEL_CMD     /usr/sbin/userdel_local

# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.

CREATE_HOME       yes
```

6.3 O ARQUIVO /ETC/GROUP

O arquivo **/etc/group** contém os nomes dos grupos UNIX e uma lista dos membros de cada grupo. Por exemplo:

```
wheel:x:10:trent,ned,evi,garth,lynda,boggs,millert
csstaff:*:100:lloyd,evi
student:*:200:dotty
```

Cada linha representa um grupo e contém quatro campos:

- Nome do grupo
- Senha criptografada ou contém um x, indicando um arquivo **gshadow**
- Número GID
- Lista de membros, separados por vírgulas (atenção para não adicionar espaços)

Assim como acontece em **/etc/passwd**, os campos são separados por dois-pontos. Os nomes de grupos devem ser limitados a oito caracteres para fins de compatibilidade, apesar de o Linux não exigir isso. Embora seja possível inserir uma senha de grupo (para permitir que usuários não pertencentes a um grupo mudem para ele utilizando o comando **newgrp**), isso raramente é feito.[6] A maioria dos sites coloca asteriscos no campo de senha, mas é seguro deixar o campo de senha em branco se desejar. O comando **newgrp** não mudará para um grupo a menos que o usuário já se encontre listado como membro desse grupo. Todas as nossas distribuições de exemplo vêm configuradas com os arquivos **/etc/gshadow**, que são análogos em conceito a **/etc/shadow**, mas bem menos importantes (senhas de grupo são raramente utilizadas).

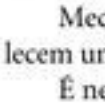
Como acontece com nomes de usuários e UIDs, os nomes de grupos e GIDs devem ser mantidos consistentes entre as máquinas que compartilham arquivos por um sistema de arquivos. Pode ser difícil manter a consistência em um ambiente heterogêneo, já que vários sistemas operacionais utilizam GIDs diferentes para os mesmos nomes de grupos. Descobrimos que a melhor maneira de lidar com essa questão é evitar utilizar um grupo de sistema como grupo de login padrão para um usuário.

Se um usuário tem como padrão um determinado grupo em **/etc/passwd**, mas não parece estar nesse grupo de acordo com **/etc/group**, **/etc/passwd** fica com o argumento. As posses de grupos concedidas no momento do login são realmente a união daquelas encontradas nos arquivos **passwd** e **group**. Entretanto, é uma boa idéia manter os dois arquivos consistentes.

Para minimizar o potencial para conflitos com GIDs fornecidos por terceiros, sugerimos iniciar grupos locais em GID 500.

A tradição do UNIX sempre foi a de acrescentar novos usuários a um grupo que representa sua categoria genérica como "estudantes" ou "financeiro". Entretanto, vale a pena notar que tal convenção aumenta a probabilidade de os usuários serem capazes de ler os arquivos uns dos outros devido à configuração descuidada de permissões, mesmo que essa não seja realmente a intenção do proprietário. Para evitar esse problema, preferimos criar um grupo único para cada usuário. Você pode utilizar o mesmo nome tanto para o usuário como para o grupo. Você também pode tornar o GID o mesmo do UID.

O grupo pessoal de um usuário deve conter apenas esse usuário. Se você quiser permitir aos usuários o compartilhamento de arquivos pelo mecanismo de grupos, crie grupos separados para essa finalidade. A idéia implícita nos grupos pessoais não é a de desencorajar o uso de grupos em si — é a de simplesmente estabelecer um grupo *padrão* mais restritivo para cada usuário para que esses arquivos não sejam compartilhados inadvertidamente.


Os utilitários **useradd** em todas as nossas distribuições de exemplo, exceto SUSE, assumem o padrão de colocar os usuários em grupos pessoais próprios.

6.4 ADICIONANDO USUÁRIOS

Antes de criar uma conta para um novo usuário em uma instalação corporativa, governamental ou educacional, é *muito* importante que o usuário assine e date uma cópia do acordo de usuário local e políticas a serem observadas. (O quê?! Você não tem esses documentos? Consulte a página 648 para informações adicionais sobre o motivo pelo qual você precisa deles e como preenchê-los.)

Os usuários não têm nenhuma razão em particular para querer assinar um acordo de políticas gerais; portanto, é para seu benefício garantir que eles assinem esses documentos enquanto você ainda tem alguma influência. Acreditamos que dá mais trabalho garantir a assinatura de um acordo após uma conta ter sido liberada. Se seu processo permitir, faça com que o trabalho burocrático preceda a criação da conta.

Mecanicamente, o processo de acrescentar um novo usuário consiste em três etapas exigidas pelo sistema, duas que estabelecem um ambiente útil para o novo usuário e várias outras etapas adicionais para conveniência do administrador.

É necessário:

- Editar os arquivos **passwd** e **shadow** para definir a conta do usuário
- Adicionar o usuário ao arquivo **/etc/group**
- Configurar uma senha inicial
- Criar e aplicar os comandos **chown** e **chmod** do diretório inicial do usuário

---

6. A única razão que temos notícia para alguém querer utilizar o comando **newgrp** no Linux seria para configurar o grupo padrão de arquivos recém-criados.

Para o usuário:

- Copiar arquivos de inicialização padrão para o diretório inicial do usuário
- Configurar o diretório inicial de e-mail do usuário e estabelecer nomes alternativos (aliases) para e-mail

Para você:

- Verificar se a conta está configurada corretamente
- Adicionar as informações de contato e o status da conta do usuário ao seu banco de dados

Iniciando na página 593, discutimos o comando **useradd** e seus irmãos, que automatizam alguns desses passos. Mas, em algumas seções a seguir, repassamos esses passos da maneira como você os executaria manualmente. Isto é feito mais para você poder entender o que as ferramentas estão fazendo. Na vida real, em geral é preferível (mais rápido e menos sujeito a erros) executar **useradd** ou algum script caseiro semelhante.

Você tem de executar cada um dos passos como o **sudo**, que lhe permite executar comandos como root. Consulte a página 35 para informações adicionais sobre o **sudo**.

Editando os arquivos passwd e shadow

Para editar de maneira segura o arquivo **passwd**, execute **vipw** para invocar um editor de texto em uma cópia dele. O editor padrão é o **vi**, porém podemos especificar um editor diferente configurando o valor de sua variável de ambiente EDITOR. A existência do arquivo de edição temporário serve como um bloqueio; **vipw** permite que somente uma pessoa por vez edite o arquivo **passwd** e impede que os usuários modifiquem suas senhas enquanto o arquivo **passwd** estiver sendo utilizado. Quando o editor finalizar o processo, **vipw** substituirá o arquivo **passwd** original pela sua cópia editada.

Nos sistemas Fedora e RHEL, **vipw** pergunta automaticamente se você gostaria de editar o arquivo **shadow** depois de editar o arquivo **passwd**. Sistemas SUSE, Debian e Ubuntu utilizam **vipw -s** para essa função.

Por exemplo, adicionar a linha a seguir a **/etc/passwd** definiria uma conta chamada "tyler":

```
tyler:x:2422:2422:Tyler Stevens, ECEE 3-27, x7919,:/home/tyler:/bin/sh
```

Também adicionaríamos uma entrada correspondente a **/etc/shadow**:

```
tyler:*::::::14974:
```

Essa linha **shadow** para tyler não possui senha criptografada nem data de expiração de senha e configura a expiração da conta para 31 de dezembro de 2010.

Editando o arquivo /etc/group

Depois disso, devemos criar uma entrada no arquivo **/etc/group** para o grupo pessoal do tyler, que também chamaremos "tyler". Esse grupo deve ter o GID 2422 para corresponder com UID do tyler de 2422. Esse é o GID padrão que atribuímos a ele no arquivo **passwd**.[7]

```
tyler::2422:tyler
```

Na realidade, tyler estará no grupo 2422 independentemente de ele estar listado ou não em **/etc/group**, pois sua entrada **passwd** já atribuiu a ele essa associação. O kernel não se preocupa com o conteúdo do **/etc/passwd** e **/etc/group**; ele só se preocupa com os números UID e GID. O principal propósito do registro de grupos pessoais no arquivo **group** é certificar-se de que comandos como **ls** exibem os nomes desses grupos de modo correto. Naturalmente, sempre é interessante ter uma lista autorizada dos grupos que você criou e dos usuários neles incluídos.

Se quiséssemos atribuir tyler a grupos adicionais, simplesmente adicionaríamos seu nome de login aos grupos adicionais dentro do arquivo **/etc/group**.

Configurando uma senha inicial

Root pode modificar a senha de qualquer usuário pelo comando **passwd**:

```
# passwd usuário
```

ou

```
$ sudo passwd usuário
```

---

7. Essa nomeação e numeração são puramente convencionais; consulte a página 72.

Regras para selecionar boas senhas são fornecidas na página 469.

passwd solicita que você insira uma nova senha e que a digite novamente para fins de confirmação. Se você optar por uma senha curta, toda em letras minúsculas ou de alguma outra forma obviamente inadequada, **passwd** reclamará e solicitará o uso de algo mais complexo. A maioria dos sistemas Linux também verifica potenciais senhas em um dicionário por maior segurança.

O utilitário **mkpasswd** que vem junto com o pacote **expect** de Don Libes torna mais fácil a geração de senhas aleatórias para novos usuários. Para melhor ou pior, a atribuição de uma senha aleatória "força" novos usuários a alterar suas senhas imediatamente, uma vez que aquelas aleatórias são difíceis de lembrar.[8] Não confunda **mkpasswd** do **expect** com o comando **mkpasswd** padrão, que simplesmente codifica uma dada string como uma senha.

Nunca deixe uma conta nova — ou qualquer conta que tenha acesso a um shell — sem uma senha.

Criando o diretório inicial do usuário

Qualquer diretório que criamos como root pertence inicialmente ao root, de modo que precisamos modificar o seu proprietário e grupo por meio dos comandos **chown** e **chgrp**. A seqüência de comandos a seguir cria um diretório inicial adequado para nosso usuário de exemplo:

```
# mkdir /home/tyler
# chown tyler:staff /home/tyler
# chmod 700 /home/tyler
```

Copiando arquivos de inicialização padrão

Você pode personalizar alguns comandos e utilitários colocando arquivos de configuração no diretório inicial de um usuário. Tradicionalmente, os arquivos de inicialização começam com um ponto e terminam com as letras **rc**, abreviação de "run command", uma relíquia do sistema operacional CTSS. O ponto inicial faz com que **ls** suprima esses arquivos das listagens do diretório a menos que seja utilizada a opção **-a**; os arquivos são considerados "desinteressantes". A Tabela 6.1 lista alguns arquivos de inicialização comuns.

Tabela 6.1 Arquivos de inicialização comuns e suas aplicações.

Comando	Nome de arquivo	Empregos típicos
csh/tcsh	**.login**	Configura o tipo de terminal (se necessário). Configura os switches **biff** e **mesg**
	.cshrc	Configura as variáveis de ambiente. Configura os aliases de comando. Configura o caminho de busca. Configura o valor de **umask** para controlar permissões Configura cdpath para pesquisas do nome de arquivo Configura variáveis prompt, history e savehist.
bash[a]	**.bashrc**	Semelhante a **.cshrc** para **bash**
	.bash_profile	Semelhante a **.login** para **bash**
vim	**.vimrc**	Configura as opções do editor **vim**
emacs	**.emacs**	Configura as opções do editor **emacs** Configura vinculações de chave do **emacs**
mail/mailx	**.mailrc**	Define aliases de e-mails. Configura as opções do leitor de mensagens.
xrdb[b]	**.Xdefaults**	Especifica a configuração X11: fontes, cores etc.
startx[b]	**.xinitrc**	Especifica o ambiente X11 inicial
	.Xclients	Especifica o ambiente X11 inicial (RHEL, Fedora)
xdm[b]	**. xsession**	Especifica o ambiente X11 inicial

a. **bash** também lerá **.profile** or **/etc/profile** na emulação de **sh**.

b. Os detalhes exatos do X Windows variam com a implementação e o gerenciador de janelas em uso; consulte o Capítulo 22 para mais detalhes.

8. As senhas não são verdadeiramente aleatórias, porém, mais exatamente, pseudo-aleatórias. Se uma ou mais senhas em uma seqüência gerada de uma maneira pseudo-aleatória forem quebradas, poderia ser possível utilizar reengenharia reversa na seqüência e descobrir senhas adicionais. Possivelmente, mas improvável na prática. Estamos relativamente confortáveis com esse risco.

Caso ainda não tenha um bom conjunto de arquivos de inicialização padrão, **/usr/local/lib/skel** é um bom lugar para colocá-los. Copie alguns arquivos para usar como ponto de partida e modifique-os com um editor de texto. Talvez você deseje iniciar com arquivos disponibilizados pelos fornecedores do diretório **/etc/skel**, se seu sistema fornecer esses arquivos. Certifique-se de configurar um valor padrão razoável para **umask**; sugerimos 077, 027 ou 022, dependendo da afabilidade e tamanho do seu site.

Dependendo do shell do usuário, **/etc** poderá conter arquivos de inicialização no nível do sistema que são processados antes dos arquivos de inicialização do usuário. Por exemplo, **bash** lê **/etc/profile** antes de processar **~/.bash_profile**. Para outros shells, consulte a página man correspondente ao shell em questão para obter maiores detalhes.

Tornou-se comum os arquivos de inicialização no nível de sistema para shells procurarem no diretório **/etc/profile.d** fragmentos de configuração adicionais a executar. Essa convenção fornece uma maneira limpa para que pacotes de software especifiquem padrões no nível do shell. Por exemplo, os arquivos **/etc/profile.d/colorls.*** no Fedora e no RHEL são responsáveis pela saída **ls** tecnicolor nesses sistemas. (Sim, eles podem ser excluídos de maneira segura.)

A seqüência de comandos para instalar arquivos de inicialização para o novo usuário tyler ficaria parecida com esta:

```
# cp /usr/local/lib/skel/.[a-zA-Z]* ~/tyler
# chown tyler:staff ~tyler/.[a-zA-Z]*
# chmod 600 ~tyler/.[a-zA-Z]*
```

Note que não podemos utilizar

```
# chown tyler:staff ~tyler/.*
```

pois tyler ficaria proprietário não somente de seus próprios arquivos como também do diretório pai ".." ((**/home**). Esse é um erro muito comum e perigoso em termos de administração de sistemas.

Configurando o diretório inicial de e-mail do usuário

É conveniente para cada usuário receber e-mails somente em uma máquina. Esse esquema é normalmente implementado por meio de uma entrada no arquivo global de nomes alternativos **/etc/mail/aliases** ou **sendmail** `userDB` no servidor de e-mail central. Veja o Capítulo 18 para obter informações gerais sobre e-mail; as diversas maneiras de implementar diretórios iniciais de e-mails são discutidas na página 375.

Verificando o novo login

Para verificar se uma nova conta foi configurada apropriadamente, primeiramente saia do sistema, depois se conecte como o novo usuário e execute os seguintes comandos:

```
$ pwd      /* Para verificar o diretório inicial */
$ ls -la   /* Verifica proprietário/grupo dos arquivos de inicialização */
```

Você precisará notificar os novos usuários a respeito de seus nomes de logins e senhas iniciais. Muitas instalações enviam essas informações via e-mail, mas, por questões de segurança, isso nem sempre é uma boa idéia. A conta de um novo usuário pode ser comprometida e invadida (backdoors) antes mesmo de ele conseguir fazer o login. Esse também é um bom momento de direcionar os usuários para documentação adicional sobre costumes locais, se houver alguma.

Consulte a página648 para informações adicionais sobre contratos escritos de usuário.

Se seu site exigir que os usuários assinem um acordo de diretiva escrito ou uma diretiva de uso apropriada, certifique-se de que esse passo foi completado antes de liberar a conta. A verificação impedirá omissões e reforçará a base legal de quaisquer sanções que você tenha de impor no futuro.

Não deixe de lembrar aos novos usuários de alterar suas senhas imediatamente. Se desejar, você pode impor isso configurando a senha para expirar dentro de um curto período de tempo. Uma outra opção é fazer com que um script verifique os novos usuários e certifique de que suas senhas criptografadas no arquivo **shadow** foram alteradas.[9]

Registrando o status e informações de contato do usuário

Em um ambiente em que você conhece todos os usuários pessoalmente é de certo modo fácil controlar quem está utilizando o sistema e o motivo para tal. Mas se você gerencia uma grande e mutável base de usuários, precisará de uma maneira mais formal de controlar essas contas. Manter um banco de dados contendo informações para contato e status das contas o

9. Como a mesma senha pode ter muitas representações criptografadas, esse método só verifica se o usuário reconfigurou a senha, não se ela foi realmente alterada em uma senha diferente. Não há nenhuma maneira prática de forçar os usuários a realmente alterarem suas senhas, exceto mantendo um banco de dados de todos os valores anteriores.

ajudará a descobrir quem é quem e por que tem a conta, uma vez que o processo de adição não está mais na memória. É uma boa idéia manter informações de contato completas à mão para que possa localizar os usuários na eventualidade de problemas ou mau comportamento.

6.5 ELIMINANDO USUÁRIOS

Quando um usuário deixa uma empresa, a conta de login deste usuário e seus respectivos arquivos devem ser eliminados do sistema. Esse procedimento envolve a eliminação de todas as referências ao nome de login que foram adicionadas por você ou por seu programa **useradd**. Se você eliminar um usuário manualmente, talvez queira utilizar a seguinte lista de verificação:

- Eliminar o usuário de todos os bancos de dados e listas telefônicas de usuários
- Eliminar o usuário do arquivo **aliases** ou adicionar um endereço de encaminhamento
- Eliminar o arquivo crontab do usuário e quaisquer tarefas **at** pendentes
- Eliminar qualquer processo do usuário que ainda esteja em execução
- Eliminar o usuário dos arquivos **passwd**, **shadow**, **group** e **gshadow**
- Eliminar o diretório inicial do usuário
- Eliminar o spool de e-mail do usuário

Antes de eliminar o diretório inicial de um usuário, certifique-se de realocar quaisquer arquivos que sejam necessários a outros usuários. Como normalmente você não pode ter certeza de quais arquivos possam ser esses, é sempre uma boa idéia fazer um backup extra do diretório inicial do usuário e fazer o spool de e-mail antes de eliminá-los.

Assim que tiver eliminado um usuário, talvez seja interessante verificar se o antigo UID do usuário não tem mais posse sobre outros arquivos do sistema. Para localizar os caminhos de arquivos órfãos, você pode utilizar o comando **find** com o argumento **-nouser**. Como **find** tem uma maneira de "escapar" em servidores de redes caso você não seja cauteloso, é melhor verificar os sistemas de arquivos individualmente com **-xdev**:

```
# find sistemaDeArquivos -xdev -nouser
```

Se sua empresa atribui estações de trabalho individuais aos usuários, geralmente é mais fácil e mais eficiente reinstalar todo o sistema por meio de uma template mestre antes de passar o sistema a um novo usuário. Porém, antes de reinstalar é uma boa idéia fazer backup de todos os arquivos locais no disco rígido do sistema no caso de eles virem a ser necessários no futuro.

6.6 DESATIVANDO LOGINS

De vez em quando, o login de um usuário precisa ser temporariamente desativado. Uma maneira simples e direta de fazer isso é colocar um asterisco ou algum outro caractere na frente da senha encriptada do usuário no arquivo **/etc/shadow**. Essa medida impede a maioria dos tipos de acesso regulados por senhas porque a senha não é mais decriptada para algo sensato. Entretanto, comandos como **ssh** que não verificam necessariamente a senha de sistema podem continuar a funcionar.

Em todas as nossas distribuições de exemplo, exceto SUSE, os comandos **usermod -L** *usuário* e **usermod -U** *usuário* fornecem uma maneira fácil de bloquear e de desbloquear senhas.

Uma maneira alternativa (e talvez uma mais segura) de alcançar um fim semelhante é substituir o shell do usuário por um programa que imprime uma mensagem que explica por que o login foi desativado e fornece instruções para retificar essa situação. Esse pseudo-shell não deve ser listado em **/etc/shells**; muitos daemons que fornecem acesso não login ao sistema (por exemplo, **ftpd**) fazem uma verificação para ver se o shell de login de um usuário se encontra em **/etc/shells** e negará acesso caso ele não esteja presente (que é o comportamento que desejamos). Infelizmente, essa mensagem pode não ser vista se o usuário tentar efetuar o login por meio de um sistema de janelas.

Entretanto, há um problema com esse método de desativação de logins. Por padrão, **sendmail** não encaminhará mensagens a um usuário cujo shell não apareça em **/etc/shells**. Geralmente é uma má idéia interferir no fluxo de e-mails, mesmo se o receptor não estiver habilitado para lê-lo imediatamente. Podemos anular o comportamento padrão do **sendmail** adicionando um shell falso chamado **/SENDMAIL/ANY/SHELL/** ao arquivo **/etc/shells** (embora possam ocorrer efeitos colaterais indesejados ao fazermos isso).

6.7 GERENCIANDO CONTAS

O comando **useradd** adiciona usuários ao arquivo **passwd** (e ao arquivo **shadow**, se aplicável). Ele fornece uma interface orientada a linhas de comando que é fácil de ser executada manualmente ou de chamar de um script **adduser** caseiro. O

comando **usermod** altera as entradas **passwd** de usuários existentes. O comando **userdel** elimina um usuário do sistema, excluindo opcionalmente o diretório inicial do usuário. Os comandos **groupadd**, **groupmod** e **groupdel** operam sobre o arquivo **/etc/group**.

Por exemplo, para criar um usuário chamado "hilbert" com o comando **useradd** (utilizando os padrões do sistema), você poderia simplesmente fazer o seguinte:

```
# useradd hilbert
```

Esse comando criaria a seguinte entrada em **/etc/passwd**. Note que **useradd** coloca um asterisco no campo de senha, desabilitando efetivamente a conta até você atribuir uma senha real.

```
hilbert:*:105:20:  :/home/hilbert:/bin/bash
```

Por alguma razão, o SUSE utiliza um conjunto semelhante, mas desenvolvido independentemente dos comandos de manipulação de usuários e de grupos. Esses comandos têm os mesmos nomes, mas há diferenças sutis nos significados de algumas opções e nos comportamentos padrão. Por exemplo, a maioria das distribuições cria um grupo pessoal dedicado para os novos usuários caso você não especifique o contrário na linha de comando. O **useradd** do SUSE coloca os novos usuários no grupo 100. (Na configuração padrão, ele também os adiciona aos grupos "video" e "dialout". Hmm.)

useradd geralmente é mais útil quando argumentos adicionais são utilizados. No próximo exemplo, especificamos o grupo primário de hilbert como "faculty" e que ele também deve ser adicionado ao grupo "famous". Também sobrescrevemos a localização de diretório inicial padrão e solicitamos a **useradd** que crie o diretório inicial caso ele ainda não exista:

```
# useradd -c "David Hilbert" -d /home/math/hilbert -g faculty -G famous -m -s /bin/sh hilbert
```

Esse comando cria a seguinte entrada **passwd**:

```
hilbert:x:1005:30:David Hilbert:/home/math/hilbert:/bin/sh
```

(o UID atribuído é uma unidade maior do que o UID mais elevado do sistema) e a entrada **shadow** correspondente:

```
hilbert:!:11508:0:99999:7:0::
```

Ele também acrescenta hilbert aos grupos "faculty" e "famous" em **/etc/groups**, cria o diretório **/home/math/hilbert** e o preenche de acordo com o conteúdo do diretório **/etc/skel**.

Em todas as nossas distribuições de exemplo, exceto SUSE, você pode determinar as configurações padrão para **useradd** executando **useradd -D**. Você também pode utilizar o flag **-D** em combinação com outros argumentos a fim de configurar esses padrões.

Mesmo no SUSE, os padrões são armazenados em **/etc/default/useradd** e podem ser editados diretamente, se você preferir.

usermod modifica uma conta que já existe e utiliza vários dos mesmos flags de **useradd**. Poderíamos, por exemplo, utilizar o comando a seguir para configurar uma data de expiração igual a 4 de julho de 2007 na conta de hilbert:

```
# usermod -e 2007-07-04 hilbert
```

O comando **userdel** elimina contas de usuário, desfazendo efetivamente todas as alterações feitas pelo **useradd**. Para eliminar hilbert, utilizaríamos o seguinte comando:

```
# userdel hilbert
```

Esse comando elimina referências a hilbert nos arquivos **passwd**, **shadow** e **group**. Por padrão, ele não eliminaria o diretório inicial de hilbert.[10] A opção **-r** faz com que **userdel** também elimine o diretório inicial do usuário, mas, mesmo em seu modo mais agressivo, **userdel** ainda executa apenas as três últimas tarefas da lista de "obrigações de eliminação de usuário".

Embora os comandos **useradd** e **userdel** sejam convenientes, eles normalmente não são suficientes para implementar as políticas locais de uma instalação. Não hesite em criar seus próprios scripts **adduser** e **rmuser**; a maioria das grandes instalações faz isso. (Perl geralmente é a ferramenta indicada para essa tarefa.) Seus scripts caseiros podem chamar os utilitários padrão para realizarem parte de suas tarefas.

---

10. Em nossa instalação geralmente preservamos por algumas semanas os diretórios iniciais de usuários eliminados. Essa política minimiza a necessidade de restauração de dados por meio de fitas de backup caso um usuário eliminado retorne ou se outros usuários precisarem ter acesso aos arquivos de trabalho do usuário eliminado.

6.8 EXERCÍCIOS

E6.1 Como o grupo padrão de um usuário é determinado? Como poderíamos modificá-lo?

E6.2 Explique as diferenças entre os seguintes valores **umask**: 077, 027, 022 e 755. Como você implementaria um desses valores como um padrão válido para todo o sistema no caso de novos usuários? É possível impor um padrão **umask** a seus usuários?

E6.3 Qual é o objetivo do arquivo de senhas sombras?

★ E6.4 Liste as etapas necessárias para adicionar um usuário a um sistema sem utilizar o comando **useradd**. Que etapas adicionais seriam necessárias para o seu ambiente local?

★ E6.5 Determine a convenção de atribuição de nomes para novos usuários no seu site. Quais são as regras? Como a exclusividade é preservada? Você seria capaz de imaginar algum inconveniente? Como os usuários são eliminados?

★ ★ E6.6 Localize uma lista de nomes (eventualmente de uma lista telefônica on-line) e use-a como entrada para um script que forma nomes de login de acordo com a convenção de atribuição de nomes de sua instalação. Quantos usuários podem ser acomodados antes de ser provocado um conflito? Quantos conflitos existem no todo? Use esses dados para avaliar a convenção de atribuição de nomes de sua instalação e sugira melhorias.

★ ★ E6.7 Crie um script para ajudar a monitorar a saúde de seu arquivo **/etc/passwd**. (Os itens *b* e *e* exigem acesso root a menos que você seja esperto.)

a) Localize quaisquer entradas que possuam UID 0.

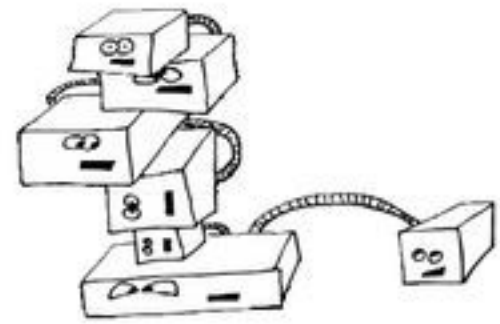
b) Localize quaisquer entradas que não possuam nenhuma senha (necessário **/etc/shadow**).

c) Localize quaisquer entradas que possuam UIDs duplicados.

d) Localize quaisquer entradas que possuam nomes de login duplicados.

e) Localize quaisquer entradas que não possuam data de expiração (necessário **/etc/shadow**).

7 Adicionando um disco



É difícil acreditar que mesmo com os avanços das últimas décadas em termos de chip, rede e tecnologia de software, ainda usemos basicamente a mesma tecnologia de armazenamento de dados a longo prazo que já era popular há 40 anos. A densidade aumentou significativamente (e os preços caíram), mas a idéia básica permanece inalterada.

Infelizmente, novos usos para o espaço em disco continuam a aparecer, especialmente com a ampla aceitação da Internet. Em muitos casos, o espaço em disco ainda é fundamental. MP3s, vídeo em stream e conteúdo multimídia impõem novos desafios aos administradores de sistemas. Nunca o gerenciamento adequado de disco foi tão importante.

A maioria dos sistemas de servidor se conecta por meio de um barramento periférico padrão chamado SCSI (Small Computer Systems Interface). Uma interface alternativa chamada IDE (Integrated Drive Electronics) é padrão nos PCs desktop e laptop. Começamos este capítulo com uma discussão geral sobre os padrões SCSI e IDE e a estrutura dos discos rígidos modernos. Discutimos então os mecanismos gerais para formatar e particionar discos e o procedimento para inicializar sistemas de arquivos. Por fim, discutiremos os recursos avançados como RAID e gerenciadores de volume.

7.1 INTERFACES DE DISCO

Hoje, somente alguns poucos padrões de interface são utilizados, embora surjam várias tecnologias novas. É importante selecionar unidades de disco que combinem com as interfaces do sistema nas quais elas serão instaladas. Se um sistema suporta várias interfaces diferentes, você deve utilizar aquela que melhor atenda às suas necessidades em termos de velocidade, redundância, mobilidade e preço:

- PATA (também conhecida como IDE) foi desenvolvido como uma interface simples e de baixo custo para PCs. Originalmente ele foi chamado de Integrated Drive Electronics, pois colocava o controlador de hardware no mesmo dispositivo que os discos da unidade de disco e usava um protocolo de alto nível para comunicação entre o computador e os discos. Essa agora é a arquitetura padrão para todos os discos modernos, porém o mesmo nome persiste. Os discos IDE apresentam uma velocidade de média para rápida, alta capacidade de armazenamento e são incrivelmente baratos. Veja a próxima seção para informações adicionais sobre IDE.
- Serial ATA, SATA, é o sucessor do IDE convencional. Além de suportar taxas de transferência muito mais altas, SATA simplifica a conectividade com um cabeamento mais aprimorado e um comprimento de cabo máximo mais longo. SATA tem suporte nativo ao swapping (troca) automático e enfileiramento de comandos (opcional), dois recursos que por fim tornaram o IDE uma alternativa viável à SCSI nos ambientes de servidor.
- Embora não tão comum quanto era, a SCSI é uma das interfaces de disco mais amplamente suportadas. Há vários modelos diferentes, mas todos suportam discos múltiplos em um barramento e várias velocidades e estilos de comunicação. SCSI é descrito em mais detalhes na página 81.
- Fibre Channel é uma interface serial que está ganhando popularidade no ambiente corporativo graças à sua alta largura de banda e ao grande número de dispositivos que podem ser conectados a ela de uma só vez. Os dispositivos Fibre Channel se conectam por meio de uma fibra óptica ou cabo de cobre axial duplo. As velocidades atuais são de 100 MB/s ou mais. Topologias comuns incluem loops, chamados FC-AL (Fibre Channel Arbitrated Loop), e estruturas que são construídas com switches Fibre Channel. Fibre Channel é capaz de "falar" vários protocolos diferentes, inclusive SCSI e até mesmo IP. Dispositivos Fibre Channel são identificados por um número de ID gravado internamente, chamado World Wide Name, ue é similar a um endereço MAC Ethernet.
- O Universal Serial Bus (USB) tornou-se popular para conexão de dispositivos como teclados e mouses, mas as versões atuais têm capacidade suficiente para suportar unidades de disco e de CD-ROM. O USB é comum em PCs e nos permite facilmente mover um disco entre sistemas.

SCSI e IDE são de longe as concorrentes dominantes no mercado de unidades de disco. Essas são assim as únicas interfaces que discutiremos detalhadamente.

A interface PATA

PATA (Parallel Advanced Technology Attachment), também chamado de IDE, foi projetado para ser simples e barato. Normalmente é mais encontrado em PCs ou estações de trabalho de baixo custo. O IDE se tornou popular no final dos anos 80. Logo depois, foi desenvolvido o ATA-2 para satisfazer a demandas crescentes por parte de consumidores e fabricantes de unidades de disco.

O ATA-2 adiciona E/S programada mais rápida (*programmed I/O* – PIO) e modos de acesso de memória direta (*direct memory access* – DMA) e estende os recursos Plug and Play do barramento. Ele também fornece um recurso chamado LBA (Logical Block Addressing, endereçamento de blocos lógicos) que (em combinação com uma BIOS de PC mais sofisticada) suplanta uma limitação que impedia as BIOS de acessar além dos primeiros 1.024 cilindros de um disco. Essa restrição limitava anteriormente os tamanhos dos discos a 504 MB. Quem poderia imaginar que um disco rígido fosse ficar tão grande?

Como a BIOS gerencia parte do processo de inicialização, anteriormente era necessário criar uma pequena partição inicializável dentro dos primeiros 1.024 cilindros para assegurar que o kernel fosse carregado por uma BIOS antiga. Assim que o kernel estivesse ativo e em funcionamento, a BIOS não era mais necessária e poderíamos acessar o restante do disco. Essa manobra boba é desnecessária em hardware moderno, pois o LBA se livra do endereçamento CHS (Cylinder-Header-Sector) em favor de um esquema de endereçamento linear.

O ATA-3 acrescenta maior confiabilidade, gerenciamento de energia mais sofisticado e recursos de automonitoramento. O Ultra-ATA tenta cobrir o intervalo deixado entre o ATA-3 e o ATA-4, acrescentando modos de alto desempenho chamados Ultra DMA/33 e Ultra DMA/66 que ampliam a largura de banda do barramento de 16 MB/s para, respectivamente, 33 MB/s ou 66 MB/s. O ATA-4 também é uma tentativa muito necessária de mesclar o ATA-3 com o protocolo ATA Packet Interface (ATAPI), que permite que unidades de CD-ROM e unidades de fita funcionem em um barramento IDE.

As adições mais recentes a essa família, ATA-5 e ATA-6, incluem gerenciamento aprimorado de desempenho e tratamento de erros, ambos os quais aprimoram o desempenho em um ambiente multiusuário como o Linux. Espera-se que o ATA-7 seja a atualização final para o ATA. Ele suporta taxas de transferência de dados tão altas quanto 133 MB/s.

Os discos IDE são quase sempre utilizados internamente (a menos que você considere um disco pendurado fora do computador como "externo" para fins de testes). O comprimento máximo do cabo para um barramento ATA-2 são meros 4,5 m, o que pode dificultar o acesso até mesmo ao compartimento de unidades superior de seu sistema. Além do curto comprimento do cabo, um barramento IDE é capaz de acomodar somente dois dispositivos. Para compensar essas deficiências, a maioria dos fabricantes adiciona mais de um barramento IDE às suas placas mãe (em geral dois, conhecidos como primário e secundário).

Os dispositivos IDE são acessados de uma maneira conectada, o que significa que somente um dispositivo por vez pode estar ativo. O desempenho nessas unidades IDE legadas é melhor se esses dispositivos forem distribuídos por múltiplos barramentos. O padrão Extended IDE supera essa limitação simulando duas interfaces IDE em uma única interface, permitindo que duas unidades estejam ativas de cada vez. Naturalmente, os dispositivos não podem enviar simultaneamente dados pelo cabo, mas isso também se aplica à SCSI.

O conector IDE é um conector de 40 pinos que conecta a unidade à placa de interface por meio de um flat cable. Padrões IDE mais novos como o Ultra DMA/66 utilizam um cabo diferente que dispõe de mais pinos terra e, conseqüentemente, reduz o ruído elétrico. Se um cabo ou unidade não estiverem encaixados, certifique-se de que o pino 1 na unidade vá para o pino 1 da placa de interface. O pino 1 normalmente está marcado com um pequeno "1" no lado do conector. Caso ele não esteja marcado, uma regra prática é que o pino 1 é normalmente o mais próximo do conector de alimentação. O pino 1 em um flat cable é usualmente identificado pela cor vermelha. Caso não haja essa faixa vermelha em um cabo, basta se certificar de que o cabo esteja orientado para que o pino 1 esteja conectado ao pino 1 e identifique-o claramente para a próxima vez.

Caso tenha mais de um dispositivo em um barramento IDE, você tem de designar um como o mestre e os demais como escravos. Uma configuração com jumpers "para seleção de estado" ("cable select") nas unidades modernas (que é normalmente a padrão) permite que os dispositivos resolvam a questão do mestre e do escravo por conta própria. Não há nenhum ganho de desempenho pelo fato de uma delas ser o mestre. Algumas unidades IDE mais antigas não gostam de ser os escravos; assim, se você tiver problemas para colocar uma configuração em funcionamento, tente inverter os papéis dos discos. Se as coisas ainda não funcionarem, tente tornar cada dispositivo o mestre do seu próprio barramento IDE.

Ao considerar o emprego de hardware IDE, tenha em mente os seguintes pontos:

- Unidades IDE novas funcionam em placas mais antigas e unidades IDE antigas funcionam em placas mais novas. Naturalmente, apenas os recursos comuns a ambos os dispositivos serão suportados.
- Em 4,5 m, o comprimento do cabo é excessivamente curto, o que pode tornar complicado o ato de adicionar um dispositivo extra ao barramento. Caso passe por problemas estranhos, verifique o comprimento do cabo.
- Novas técnicas de design de cabo utilizam cabeamento arredondado em vez do cabo de fita mais comum para dispositivos IDE. Os novos cabos efetivamente organizam o chassi e aprimoram o fluxo de ar.

A interface SATA

À medida que as taxas de transferência de dados para unidades PATA aumentaram (especialmente com o advento do ATA-7), as desvantagens desse padrão tornaram-se óbvias. Interferência eletromagnética e outros problemas elétricos começaram a causar preocupações quanto à confiabilidade em velocidades altas. Serial ATA, SATA, foi inventado para resolver esses problemas. Embora o SATA esteja, conceitualmente, disponível desde 2000, só recentemente ele começou a aparecer nas estações de trabalho dos consumidores.

O SATA minimiza muitos dos problemas do PATA. Embora as unidades SATA iniciais só sejam um pouco mais rápidas que o ATA-7 a 150 MB/s, o SATA-2, em última instância, suportará taxas de transferência a 600 MB/s até 2008.[1] Outras alterações notáveis incluem uma superior verificação de erros, a capacidade para unidades de swapping automático, enfileiramento de comandos nativo e diversos aprimoramentos de desempenho. Por fim, o SATA elimina a necessidade das designações mestre e escravo porque somente um único dispositivo pode ser conectado a cada canal.

O SATA supera a limitação dos cabos de 18 polegadas do PATA e introduz novos padrões de cabos e conectores com condutores de 7 e 15, respectivamente.[2] Esses cabos são infinitamente mais flexíveis e mais fáceis de trabalhar do que seus predecessores, os cabos de fita — não há mais necessidade de curvar e torcer os fios para que as unidades se ajustem no mesmo cabo!

A interface SCSI

Vários chipsets implementam o padrão SCSI, de modo que os fornecedores algumas vezes fornecem suporte SCSI diretamente na placa mãe. A SCSI define um canal de dados genérico que pode ser usado por todos os tipos de periféricos. No passado ele era utilizado para discos, unidades de fita, scanners e impressoras, mas hoje em dia a maioria dos periféricos trocou a SCSI pelo USB. O padrão SCSI não especifica como um disco é construído ou disposto, somente a maneira como ele se comunica com outros dispositivos.

O padrão SCSI passou por diversas revisões, com SCSI-3 sendo a versão atual. A SCSI-1 foi desenvolvida em 1986 como um padrão ANSI baseado no SASI (Shugart Associates System Interface) que era um sistema de barramento disponível comercialmente. A SCSI-2 foi desenvolvida em 1990. Ela é compatível com a versão anterior SCSI-1, mas acrescenta diversos recursos em termos de desempenho. Entre esses recursos, temos o enfileiramento de comandos, que permite aos dispositivos reordenar solicitações de E/S para otimizar a vazão de E/S "scatter-gather", que permite acesso direto à memória (DMA) em regiões de memória descontínuas.

Talvez você possa ver os termos "fast" e "wide" aplicados ao dispositivo SCSI-2; esses significam que a velocidade de barramento é dobrada ou que o número de bits transferidos simultaneamente é maior, tipicamente 16 bits ou 32 bits em vez dos usuais 8 bits.[3] Cadeias wide SCSI também podem suportar até 16 dispositivos; narrow SCSI permitem somente 8. Velocidade e largura são recursos distintos comumente utilizados juntos para aumentos sinérgicos.

SCSI-3 é, de fato, uma família de padrões. Ele inclui especificações para várias mídias físicas, inclusive os tradicionais barramentos paralelos e mídia serial de alta velocidade, como Fibre Channel e IEEE 1394 ("FireWire"). Ele também define os conjuntos de comandos SCSI e introduz melhorias para suporte à autoconfiguração de dispositivos, aplicações multimídia e novos tipos de dispositivos.

Embora a especificação SCSI-3 esteja em desenvolvimento desde 1993, ela ainda não foi finalizada. Mas muitos dos seus recursos já chegaram ao mercado, freqüentemente sob o nome "Ultra SCSI". O SCSI-3 engloba o SCSI-2, de forma que certo grau de compatibilidade se encontra disponível. Tenha em mente, entretanto, que colocar um dispositivo mais antigo em um barramento mais novo pode diminuir a velocidade do barramento inteiro e também afetar o comprimento máximo do cabo.

A Tabela 7.1, a seguir, resume as diferentes versões SCSI e as larguras de banda do barramento e comprimentos dos cabos associados.

O comprimento máximo de cabo para Ultra SCSI e Ultra Wide SCSI de terminação simples depende do número de dispositivos em uso. Para 8 dispositivos, 1,5 metro é o máximo; se forem utilizados somente 4 dispositivos, o barramento pode ser estendido a 3 metros. O Wide Ultra SCSI suporta todos os 16 dispositivos apenas no modo diferencial.

São utilizados diversos tipos de conectores para dispositivos SCSI. Eles variam dependendo da versão do padrão SCSI em uso e do tipo de conexão: internos ou externos. Os dispositivos narrow SCSI possuem 50 pinos e os dispositivos wide SCSI possuem 68 pinos. Os dispositivos internos aceitam tipicamente um conector de 50 pinos ou um conector minimicro de 68 pinos macho conectado a um flat cable. As unidades externas normalmente se conectam ao computador por meio de um conector minimicro de alta densidade de 50 pinos ou 68 pinos.

---

1. Haverá unidades de disco que realmente podem ler e gravar dados a 600 MB/s? Não, mas, mesmo agora, o barramento não é mais um fator limitante.
2. Isso mesmo: por alguma razão, o cabo de força é mais complicado do que o cabo de dados.
3. Barramentos SCSI de 32 bits não são muito comuns. Alguns podem requerer vários cabos, conhecidos como o cabo A e o cabo B.

Tabela 7.1 A evolução da SCSI

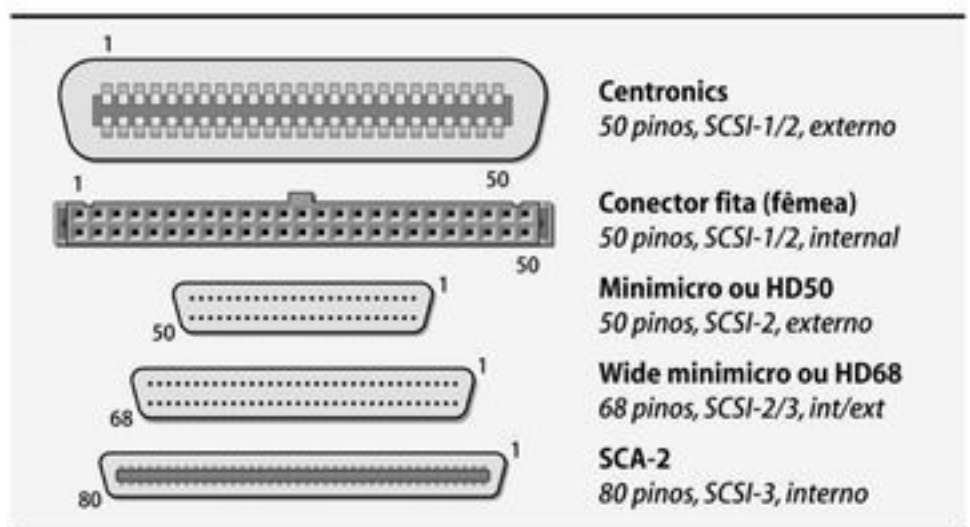
Versão	Freq.	Largura	Velocidade	Comprimento	Comprimento diferencial
SCSI-1	5 MHz	8 bits	5 MB/s	6 m	25 m
SCSI-2	5 MHz	8 bits	5 MB/s	6 m	25 m
Fast SCSI-2	10 MHz	8 bits	10 MB/s	3 m	25 m
Fast/wide SCSI-2	10 MHz	16 bits	20 MB/s	3 m	25 m
Ultra SCSI	20 MHz	8 bits	20 MB/s	1,5 m[a]	25 m
Wide Ultra SCSI[b]	20 MHz	16 bits	40 MB/s	1,5 m[a]	25 m
Wide Ultra2 SCSI[b]	40 MHz	16 bits	80 MB/s	–[c]	25 m (HVD)[d] 12 m (LVD)
Wide Ultra3 SCSI[e]	80 MHz	16 bits	160 MB/s	–[c]	12 m (LVD)

a. Varia, veja os comentários no texto.
b. Wide Ultra SCSI e wide Ultra2 SCSI são às vezes chamados Fast-20 wide SCSI e Fast-40 wide SCSI, respectivamente.
c. Essas versões da SCSI só usam sinalização diferencial.
d. HVD significa diferencial de alta voltagem e LVD significa diferencial de baixa voltagem. HVD é utilizada para as primeiras versões SCSI e não é definida acima de Ultra2 SCSI.
e. Wide Ultra3 SCSI é às vezes chamada de Ultra-160. Padrões semelhantes que duplicam e triplicam as taxas de transferência de dados são Ultra-320 e Ultra-640, respectivamente. A maioria das novas unidades é Ultra-320.

Uma variante interessante que é especialmente útil para os arrays de unidades que podem ser trocadas com o sistema em funcionamento é o plugue SCA (Single Connector Attachment). Trata-se de um conector de 80 pinos que inclui conexões de barramento, alimentação e configuração SCSI, permitindo a um único conector suprir todas as necessidades da unidade.

A Ilustração A mostra figuras dos conectores mais comuns. Cada conector é mostrado de frente, como se estivesse para ser conectado na sua parte frontal.

Ilustração A Conectores SCSI comuns (vista frontal, macho, exceto onde indicado em contrário)



Os barramentos SCSI utilizam uma configuração em cadeia de modo que a maioria dos dispositivos externos possui duas portas SCSI. As portas são idênticas e intercambiáveis, então cada uma delas pode ser a entrada. Por alguma razão, os fabricantes de scanner parecem considerar-se isentos das regras normais da física e algumas vezes fornecem somente uma porta SCSI. Se ela não for terminada internamente, esses dispositivos exigirão um tipo especial de terminador.

Os dispositivos SCSI internos são usualmente conectados a um flat cable; somente uma porta é necessária no dispositivo SCSI em si, pois os conectores podem ser conectados no meio do flat cable. Ao utilizar um flat cable, certifique-se de que o pino 1 do barramento SCSI esteja conectado ao pino 1 da unidade de disco rígido. (Pino 1 normalmente é marcado com uma listra vermelha.)

Cada final do barramento SCSI deve ter um resistor de terminação ("terminador"). Esses resistores absorvem sinais à medida que eles atingem o final do barramento e evitam que ruído se reflita de volta para o barramento. Os terminadores assumem várias formas, desde pequenos plugues externos que ligamos em uma porta regular a conjuntos de minúsculos pacotes de resistor que se instalam nas placas de circuito de um dispositivo. A maioria dos dispositivos modernos é autofinalizadora.

Uma extremidade do barramento normalmente termina dentro do computador host, em um controlador SCSI ou em uma unidade SCSI interna. A outra ponta normalmente termina em um dispositivo externo ou no controlador SCSI caso não haja dispositivos externos. Se você passar por problemas de hardware aparentemente aleatórios no seu barramento SCSI, verifique primeiramente se ambas as extremidades do barramento estão apropriadamente terminadas. Terminação imprópria é um dos enganos de configuração mais comuns em interfaces SCSI, e os erros por elas produzidos podem ser obscuros e intermitentes.

Cada dispositivo possui um endereço SCSI ou "número de destino" que o distingue dos demais dispositivos presentes no barramento. Os números de destino começam em 0 e vão até 7 ou 15, dependendo de o barramento ser narrow ou wide. O controlador SCSI em si é contado como um dispositivo e normalmente é o destino 7 (mesmo em um barramento wide, para fins de compatibilidade com versões anteriores). Todos os demais dispositivos devem ter seus números de destino configurados a valores únicos. É um erro comum se esquecer de que o controlador SCSI possui um número de destino e configurar um dispositivo para o mesmo número de destino do controlador.

Um endereço SCSI é basicamente arbitrário. Tecnicamente, ele determina a prioridade do dispositivo no barramento, mas na prática as prioridades exatas não fazem muita diferença. Alguns sistemas selecionam o disco com o número de destino mais baixo, como o disco de inicialização, e outros exigem que o disco de inicialização seja o de número 0.

Se você tiver sorte, um dispositivo terá um botão giratório por meio do qual o número de destino pode ser configurado. Outras maneiras comuns de configurar o número de destino são jumpers e DIP switches. Se não for óbvio como configurar o número de destino em um dispositivo, consulte o manual de hardware. Atualmente, a maioria das especificações de hardware pode ser encontrada nos sites do fabricante; tentar configurar um disco costumava envolver um processo de tentativa e erro.

O padrão SCSI suporta uma forma de subendereçamento chamada "número de unidade lógica". Cada destino pode ter várias unidades lógicas internas. Um exemplo plausível poderia ser um array de unidades com vários discos, mas somente um controlador SCSI. Entretanto, as unidades lógicas são raramente utilizadas na vida real. Ao escutar o "número da unidade lógica SCSI", você pode pensar que ele é realmente um número de destino que está em discussão até provem o contrário. Se um dispositivo SCSI contiver apenas uma unidade lógica, o LUN normalmente terá como padrão 0.

Os barramentos SCSI são em geral relativamente fáceis de configurar, mas uma série de coisas pode dar errado:

- Muitas estações de trabalho possuem dispositivos SCSI internos. Verifique a listagem dos dispositivos atuais antes de reinicializar para adicionar um novo dispositivo. Lembre-se de que a maioria dos sistemas de fita e algumas unidades de disquete (mais notavelmente, HP) é SCSI.
- Depois de ter acrescentado um dispositivo SCSI novo, verifique a listagem dos dispositivos descobertos pelo kernel do Linux quando ele reinicializa para garantir que tudo o que você espera esteja realmente lá. A maioria dos drivers SCSI não detecta múltiplos dispositivos que têm o mesmo endereço SCSI — uma configuração ilegal. Conflitos de endereço SCSI podem levar a um comportamento muito estranho.
- Alguns dispositivos de expansão (que contêm uma fonte de alimentação e um ou mais dispositivos SCSI) terminam o barramento dentro do dispositivo. Se os dispositivos forem conectados após o dispositivo de expansão ter sido anexado, você poderá ter problemas de confiabilidade com qualquer um dos dispositivos na cadeia SCSI. Sempre verifique e confira de novo se você tem exatamente dois terminadores e se ambos estão nos finais do barramento.
- O botão giratório utilizado para configurar o endereço SCSI de um dispositivo é algumas vezes conectado ao contrário. Quando isso acontece, o botão giratório mudará o endereço SCSI, mas não o valor exibido.
- Ao calcular o comprimento de um barramento SCSI-2, certifique-se de contar os cabos internos e os dispositivos de expansão. Eles podem ser bem longos. Lembre-se também de que o comprimento admissível é menor se você estiver adicionando dispositivos SCSI mais antigos a um barramento SCSI mais novo.
- Jamais se esqueça de que um controlador SCSI usa um dos endereços SCSI!

Qual é melhor, SCSI ou IDE?

Nas edições anteriores deste livro, SCSI era a melhor opção para aplicativos servidores. Entretanto, com o advento das unidades SATA, SCSI simplesmente não mais traz as vantagens que habitualmente trazia. Unidades SATA estão, em quase todas as categorias, no mesmo patamar (e em alguns casos superam) que os discos SCSI. Ao mesmo tempo, os dispositivos SATA são muito mais baratos e mais amplamente disponíveis.

Uma das vantagens das unidades SCSI é a de um processador integrado à controladora, o que libera a CPU do sistema para que ela se concentre em outras tarefas. Nos sistemas ocupados, isso pode significar bons ganhos de desempenho. Naturalmente, cabe a você decidir se esse ganho compensa o oneroso custo financeiro.

Entretanto, em algumas situações, a SCSI é recomendável ou até mesmo obrigatória:

- Se você precisa absolutamente do melhor desempenho possível, parta para SCSI. Fabricantes de unidades de disco utilizam a separação IDE/SCSI para ajudá-los a estratificar o mercado dessas unidades de disco. Algumas unidades IDE podem ser superiores em termos de desempenho máximo, contudo a SCSI quase sempre fornece um desempenho sustentado melhor.
- Os servidores e sistemas multiusuário requerem SCSI. Não há paralelo ao protocolo SCSI quanto à sua capacidade de gerenciar múltiplas solicitações simultâneas eficientemente. Em um sistema congestionado, você verá uma melhoria concreta e mensurável em termos de desempenho.
- Caso queira conectar vários dispositivos, a SCSI ganha novamente. Os dispositivos SCSI se dão bem com os demais; os dispositivos IDE monopolizam e lutam pelo barramento.

7.2 GEOMETRIA DO DISCO

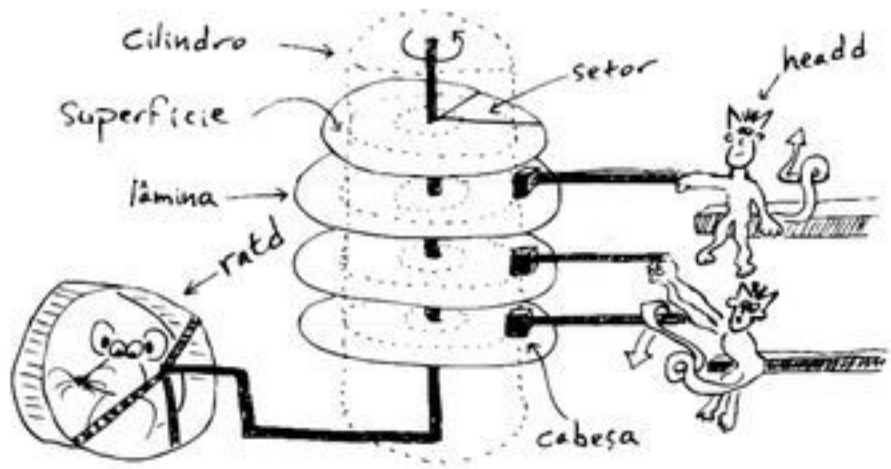
A geometria de um disco rígido e a terminologia utilizada para se referir às suas várias partes são mostradas na Ilustração B. Essas informações são fornecidas basicamente para melhorar seu conhecimento geral. As unidades de disco modernas ainda se apóiam no mesmo projeto básico, porém o software não sabe mais (ou precisa saber) muito a respeito da construção física da unidade.

Uma típica unidade de disco é formada por discos em rotação cobertos com um filme magnético. Os dados são lidos e gravados por uma pequena cabeça que muda a orientação das partículas magnéticas na superfície dos discos. os discos de dados são completamente selados para que nenhuma sujeira ou pó seja capaz de entrar. Esse recurso torna os discos rígidos fixos muito mais confiáveis do que as mídias removíveis.

Nos primórdios do hardware de computadores, as unidades de disco normalmente tinham um disco. A capacidade de armazenamento aumentou com o aumento no diâmetro do disco. Na parede de uma das nossas áreas de usuário era comum exibir um disco antigo com mais de quatro pés de diâmetro que continha aproximadamente 280 KB de dados.

Hoje, as unidades de disco rígido possuem vários discos pequenos empilhados uns sobre os outros em vez de um único e grande disco. Ambos os lados armazenam dados, embora um dos lados normalmente contenha informações sobre posicionamento e não pode ser utilizado para armazenamento. A densidade de um disco individual é nos dias atuais de aproximadamente 130 GB.

Ilustração B Lição sobre geometria de discos



Os discos giram numa velocidade constante. Eles são lidos e gravados por pequenas cabeças deslizantes que se movem para frente e para trás como a agulha de um toca-discos. As cabeças flutuam muito próximas da superfície dos discos, mas não as tocam realmente. A distância entre a cabeça e o disco em rotação pode ser comparada a um jato de combate F-16 voando a plena velocidade a 3 m do solo. Se uma cabeça tocar um disco, esse evento é chamado de "head crash"; ele pode ser muito destrutivo.

As velocidades rotacionais aumentaram drasticamente ao longo do tempo. Os discos mais antigos rodavam a 3.600 RPM ou 5.400 RPM. Hoje, 7.200 RPM é o padrão de todo o mercado e unidades a 10.000 RPM e 15.000 RPM são populares no mercado de ponta. Velocidades maiores diminuem a latência e aumentam a largura de banda da transferência de dados, mas podem potencialmente introduzir problemas térmicos que se originam do aumento de produção de calor. Certifique-se de ter uma circulação de ar adequada caso planeje adquirir uma unidade de disco de ponta.

É necessária pelo menos uma cabeça para cada superfície. As cabeças em unidades antigas tinham de percorrer grandes distâncias, porém a geometria moderna de discos pequenos empilhados é mais eficiente. O diâmetro dos discos continua a diminuir, de um padrão de 14 polegadas há 20 anos, a 5 ¼ polegadas há 10 anos e, hoje, 3 ½ polegadas para menos.

Movimentar a cabeça para a posição correta para ler um determinado trecho de dados é chamado de busca. Cada posição que uma cabeça pode ocupar à medida que o disco gira é chamada de trilha. As trilhas são divididas ainda mais em setores, que têm normalmente um comprimento de 512 bytes.

Um conjunto de trilhas em discos diferentes que estão à mesma distância a partir do eixo é chamado de cilindro. Se todas as cabeças se movimentarem juntas, como é típico na maioria das unidades do mercado de armazenamento de massa, os dados armazenados em um único cilindro pode ser lido sem qualquer movimento adicional. Embora as cabeças se movam com incrível velocidade, elas ainda se movimentam de maneira muito mais lenta do que a rotação dos discos. Portanto, qualquer acesso de disco que não requeira que as cabeças busquem uma nova posição será mais rápido.

7.3 SISTEMAS DE ARQUIVOS DO LINUX

O suporte do sistema de arquivos Linux evoluiu rapidamente à medida que ele absorveu os recursos provenientes de diferentes sistemas operacionais. A camada Virtual File System (VFS) do kernel é particularmente útil nos PCs, porque fornece a estrutura necessária para montar sistemas de arquivos "nativos" como os infames sistemas de arquivos FAT do Windows.

Ext2fs e ext3fs

O segundo sistema de arquivos estendido conhecido comumente como ext2fs foi, por muito tempo, o principal sistema de arquivos do Linux. Ele foi concebido e implementado basicamente por Rémy Card, Theodore Ts'o e Stephen Tweedie. Embora o código para ext2 tenha sido escrito especificamente para Linux, ele adota muitos conceitos e tipos de arquivos do sistema de arquivos FFS do BSD concebido e implementado por Kirk McKusick e equipe em 1984.

O terceiro sistema de arquivos estendido, ext3fs, é uma extensão notável ao ext2fs, originalmente desenvolvida por Stephen Tweedie. Ele agora é o sistema de arquivos padrão utilizado pela maioria das distribuições. Ext3fs adiciona a capacidade de *journaling* ao código ext2fs existente, uma modificação conceitualmente simples que aumenta de modo significativo a confiabilidade. Mais interessante ainda, as extensões ext3fs foram implementadas sem alterar a estrutura fundamental do ext2fs. De fato, ainda podemos montar um sistema de arquivos ext3fs como um sistema de arquivos ext2fs — ele simplesmente não terá os recursos de *journaling* ativados.

É possível converter um sistema de arquivos ext2fs existente em um sistema de arquivos ext3fs com **tune2fs**. Se você tivesse, por exemplo, um sistema de arquivos ext2fs em **/dev/hda4**, poderia convertê-lo com

```
#  tune2fs -j /dev/hda4
```

Você precisaria então modificar a entrada correspondente em **/etc/fstab** para ler `ext3` em vez de `ext2` (veja a página 89 para informações adicionais sobre o arquivo **fstab**).

Ext3fs separa uma área do disco para o arquivo *journal*. Quando uma operação de sistema de arquivos ocorre, as modificações necessárias são primeiramente gravadas no arquivo journaling. Quando a atualização do journal é completada, um registro de complemento ("*commit record*") é gravado para marcar o final da entrada. Então apenas o sistema de arquivos normal é modificado. Se ocorrer um travamento durante a atualização, utilize o log journal para reconstruir um sistema de arquivos perfeitamente consistente.

O journaling reduz o tempo necessário para execução de verificações de consistência no sistema de arquivos (consulte a seção **fsck** na página 92) a aproximadamente um segundo por sistema de arquivos. Excetuando-se algum tipo de falha de hardware, o estado de um sistema de arquivos ext3fs pode ser determinado e restaurado quase instantaneamente.

Ext3fs é o único sistema de arquivos que descrevemos em detalhe neste capítulo.

ReiserFS

O ReiserFS, escrito por Hans Reiser, é um outro sistema de arquivos com boas perspectivas de êxito para o Linux e é o sistema de arquivos favorito para a distribuição SUSE. Como o ext3fs, ele também possui um sistema de arquivos com journaling e, portanto, é capaz de manter a consistência do sistema de arquivos apesar de incidentes como quedas de sistema e reinicializações não planejadas (que podem ser comuns em um ambiente laptop ou desktop). A versão 4 do ReiserFS foi distribuída em agosto de 2004 e incluída na instalação padrão do Ubuntu. Por enquanto, o Fedora e o SUSE ainda utilizam uma distribuição anterior.

Além dos recursos de journaling, o Reiser4 fornecerá uma interface de sistema de arquivos modular pela qual desenvolvedores de aplicativos e administradores de sistemas podem especificar como os arquivos devem ser manipulados (e protegidos) de uma forma muito detalhada. Esse recurso aumenta a segurança dos arquivos em ambientes especializados.

O ReiserFS foi patrocinado pelo DARPA[4] e é o único sistema de arquivos de código-fonte aberto que foi supostamente arquitetado para segurança militar.

Novos algoritmos no Reiser4 são mais eficientes no uso do espaço do que seus predecessores. Outros sistemas de arquivos (bem como as primeiras versões do ReiserFS) utilizam algoritmos de árvore balanceados para alocar blocos do espaço em disco. Embora essa abordagem testada e comprovada seja considerada sólida, em geral há uma compensação entre a velocidade e o uso de disco. O algoritmo "dancing tree" (árvore dançante) do Reiser4 não força os administradores a escolher um lado.

XFS e JFS

Dois outros competidores na categoria "principal sistema de arquivos Linux moderno" são XFS da SGI e JFS da IBM. Ambos são suportados nativamente pelos kernels e distribuições atuais e os dois são sistemas de arquivos com *journaling* de alto desempenho. Esses dois sistemas de arquivos têm proponentes, mas nenhum é aparentemente utilizado de modo tão amplo quanto o ext3fs ou o ReiserFS.

É difícil explicar com alguma precisão por que isso ocorre, mas o problema fundamental parece residir no fato de que esses sistemas de arquivos quase não diferem do ext3fs e do ReiserFS. Na realidade, os detalhes de implementação variam e a superioridade de cada sistema de arquivos é modesta (por exemplo, XFS tem melhor gerenciamento de fragmentação e de extensão). Contudo, o desempenho entre os concorrentes mais importantes é semelhante e são raros os aplicativos em que um sistema de arquivos individual realmente se destaca.

Se quiser avaliar esses sistemas de arquivos em detalhes, você poderá encontrar informações adicionais na Web. Para o JFS, veja jfs.sourceforge.net. XFS está em oss.sgi.com/projects/xfs.

7.4 UMA VISÃO GERAL SOBRE O PROCEDIMENTO DE INSTALAÇÃO DE DISCOS

O procedimento para adicionar um novo disco envolve as seguintes etapas:

- Conectar o disco ao computador
- Formatar o disco se necessário (normalmente não)
- Identificar e particionar o disco
- Estabelecer volumes lógicos
- Criar sistemas de arquivos dentro de partições de disco
- Montar os sistemas de arquivos
- Configurar montagem automática
- Configurar o "swapping" em partições de swap

As seções a seguir elaboram essas etapas básicas. A partir da página 94, mostramos um exemplo contendo o processo completo do início ao fim de uma unidade de disco.

Conectando o disco

A maneira como o disco está conectado ao sistema depende basicamente da interface que é utilizada. O restante é tudo questão de montagem e cabeamento. Se o disco for IDE, tente configurar o sistema somente com um disco IDE por barramento. Verifique bem a orientação dos cabos e as configurações mestre/escravo de cada disco. Se o disco for SCSI, verifique bem se você efetuou as terminações apropriadas em ambas as extremidades do barramento SCSI, se o comprimento de cabo é menor do que o máximo apropriado para a variante SCSI que está usando, e se o novo destino SCSI não entra em conflito com o controlador ou um outro dispositivo no barramento. Para maiores detalhes, veja as seções específicas do sistema mais para o final deste capítulo.

Antes de poder acessar um disco novo, você precisa de arquivos de dispositivos em **/dev** que apontem para ele. O Linux cria automaticamente arquivos para os dispositivos de disco mais comuns. Consulte a página 56 para informações adicionais sobre arquivos de dispositivos.

É possível destruir um sistema de arquivos em segundos gravando aleatoriamente no disco, de modo que devemos configurar permissões bem restritivas para acesso a arquivos de dispositivos de disco. Considere fornecer acesso de leitura ao proprietário ("operador" ou "disco") do grupo; essa configuração permite que operadores sem privilégios de superusuário executem o comando **dump** e evita que meros mortais leiam a partir do dispositivo bruto.

---

4. O Defense Advanced Research Projects Agency, a mesma entidade que financiou a criação da Internet.

Formatando o disco

Os fornecedores citam as capacidades de disco em termos do número de bytes não formatados. Cerca de 10% dessa capacidade é utilizada tipicamente para marcar as superfícies do disco para que o hardware e o software possam localizar os dados que se encontram gravados ali. Ao comprar discos, sempre pense em termos de tamanho formatado e compare os preços de acordo.

O tamanho dos discos é especificado em megabytes que são milhões de bytes, em oposição à memória, que é especificada em megabytes de 2^{20} (1.048.576) bytes. A diferença é aproximadamente 5%.[5] Não deixe de verificar suas unidades ao estimar e comparar as capacidades.

O processo de formatação grava informações de endereços e marcas de temporização em cada um dos discos para delinear cada setor. Ele também identifica "blocos ruins", imperfeições na mídia que resultam em áreas que não podem ser lidas ou gravadas de forma confiável. Os discos modernos possuem gerenciamento de blocos ruins incorporado, de modo que nem você nem o driver precisam se preocupar com isso.[6]

Todos os discos vêm pré-formatados, e a formatação de fábrica normalmente é mais precisa do que qualquer formatação que você possa vir a fazer em campo. É melhor evitar a formatação de baixo nível caso ela não seja realmente necessária. Caso encontre erros de leitura ou de gravação num disco, verifique primeiramente cabos, terminações e problemas de endereçamento, os quais podem provocar sintomas similares àqueles de um bloco ruim. Se após esse procedimento você ainda estiver convencido de que o disco está ruim, talvez seja melhor substituí-lo por um novo do que ficar aguardando longas horas para uma formatação terminar.

Os discos IDE normalmente não são projetados para serem formatados fora de fábrica. Entretanto, talvez você seja capaz de obter software de formatação especial do fabricante, normalmente para Windows. Certifique-se de que o software seja adequado para a unidade que pretende formatar e siga cuidadosamente as instruções do fabricante.

Os discos SCSI se formatam em resposta a um comando que enviamos do computador host. O procedimento para envio desse comando varia de sistema para sistema. Em PCs, normalmente podemos dar o comando pela BIOS do controlador SCSI.

Vários utilitários nos permitem verificar a integridade de um disco gravando padrões aleatórios nele e depois os lendo de volta. Testes completos levam muito tempo (horas) e tendem a ter pouco valor no prognóstico. A menos que você suspeite de que um disco tenha defeitos e simplesmente não consiga substituí-lo (ou cobrar por hora), você deve pular esses testes. Excetuando isso, faça com que os testes sejam efetuados de madrugada. Não se preocupe com o "desgaste" de um disco ao fazer testes agressivos ou usá-lo excessivamente. Os discos são projetados para suportar atividades constantes.

Identificando e particionando o disco

Depois de um disco ter sido formatado e seus setores ruins remapeados, ele precisa ser dividido em blocos chamados partições ou *fatias*. O particionamento permite ao disco ser tratado como um grupo de áreas de dados independentes em vez de uma vasta expansão de blocos. O particionamento também permite que itens de "bonificação" como os blocos de inicialização e a tabela de partição em si sejam ocultos de software de alto nível (por exemplo, o sistema de arquivos). O layout do disco é simplificado pelo kernel; outros softwares funcionam com a abstração da limpeza de partições.

As partições facilitam os backups, impedem que usuários invadam ilicitamente o espaço de disco de terceiros, melhoram o desempenho e limitam o potencial de programas fora de controle. A tabela de partição é mantida no disco em um registro chamado label. O label normalmente ocupa os primeiros blocos do disco. Seu conteúdo exato varia, mas ele geralmente contém informações suficientes para fazer com que o sistema seja inicializado.

As partições são, conceitualmente, distintas e separadas umas das outras. Entretanto, o arquivo **/dev/sda** é o arquivo de dispositivo de bloco do disco, essencialmente uma imagem do disco inteiro. Comandos de nível do usuário podem acessar o disco "diretamente" por meio desse arquivo de dispositivo. Por exemplo, um processo em nível de usuário pode gravar o label do disco ou duplicar seu conteúdo para um disco de backup usando o comando **dd**. Obviamente, esse arquivo especial tem de ser utilizado cuidadosamente, já que ele permite que todas as partições no disco sejam danificadas de uma só vez.

Alguns sistemas vão mais além dessa traiçoeira trajetória e nos permitem definir vários conjuntos de partições sobrepostos. Espera-se que você utilize um conjunto de partições autoconsistentes e simplesmente ignore os demais. Na vida real, tais partições sobrepostas são um convite a erros de operadores e uma causa potencial de corrupção aleatória de dados.

Os sistemas modernos tendem a utilizar menos partições do que seus predecessores, porém, na maioria dos sistemas, teremos pelo menos duas:

- **A partição-raiz**: Tudo o que é necessário para acionar o sistema no modo de monousuário é mantido aqui. Uma réplica dessa partição armazenada em um disco diferente pode ser útil em emergências.

---

5. Obviamente, o prefixo "mega" realmente significa "milhão", de modo que a prática não é inteiramente indefensável.
6. Entretanto, quaisquer blocos ruins que apareçam depois de um disco ter sido formatado não serão "manipulados"; eles podem se manifestar na forma de erros de leitura e de gravação e de dados perdidos.

- **A partição de swap**: Uma área de swap armazena páginas de memória virtual quando não há memória física suficiente para armazená-las. Todo sistema deve ter ao menos uma partição de swap.

As opiniões diferem sobre a melhor maneira de dividir discos em partições. Eis algumas dicas:

- Se tiver vários discos, faça uma cópia do sistema de arquivos-raiz em um deles e verifique se consegue inicializar por meio dele.
- À medida que for acrescentando memória a uma máquina, você também deve adicionar espaço de swap. Aloque espaço de swap conforme necessário a fim de acomodar a carga de trabalho no seu sistema. Consulte a página 556 para obter maiores informações sobre memória virtual.
- Dividir o espaço de swap entre vários discos aumenta o desempenho. Essa técnica também funciona para sistemas de arquivos; coloque os de maior atividade em discos diferentes. Consulte a página 559 para obter notas sobre esse assunto.
- Se pretende fazer backup de alguma partição, não a torne maior do que a capacidade de seu dispositivo de backup. Consulte a página 114.
- Tente agrupar informações que mudam com rapidez em algumas poucas partições cujo backup é feito freqüentemente.
- Colocar **/tmp** em um sistema de arquivos separado limita os arquivos a um tamanho finito e evita que você precise fazer o backup deles.
- Já que os arquivos de log são mantidos em **/var**, é uma boa idéia que **/var** seja uma partição de disco separada. Deixar **/var** como parte de uma pequena partição-raiz fica mais fácil quer a raiz encha e pare a máquina.
- Pode ser útil definir **/home** como uma partição separada. Mesmo se a partição-raiz for corrompida ou destruída, há uma boa probabilidade de os dados do usuário permanecerem intactos. Inversamente, o sistema pode continuar a operar mesmo depois de um script de shell desorientado do usuário preencher **/home**.

Criando sistemas de arquivos dentro de partições de disco

Mesmo depois de um disco rígido ter sido dividido conceitualmente em partições, ele ainda não está pronto para armazenar arquivos. O sistema de arquivos precisa adicionar um pouco do seu custo próprio antes de o disco estar pronto para uso.

Para instalar um sistema de arquivos ext3fs dentro de uma partição de disco, utilize **mke2fs -j**. (Não há nenhum comando **mke3fs**; a opção **-j** instrui **mke2fs** a criar um diário (*journal*) ext3fs.) A menos que esteja fazendo algo estranho, você deve ser capaz de construir o sistema de arquivos especificando nada mais do que o comando **mke2fs -j** e o nome da partição. Tenha cuidado com **mke2fs**; um equívoco pode destruir seu sistema, tornando a recuperação de dados extremamente difícil, se não impossível.

Um sistema de arquivos ext3fs consiste em cinco componentes estruturais:

- Um conjunto de células de armazenamento inode
- Um conjunto de "superblocos" distribuídos
- Um mapa dos blocos de disco no sistema de arquivos
- Um resumo do emprego de blocos
- Um conjunto de blocos de dados

O diário é alocado como se fosse um arquivo regular na raiz do novo sistema de arquivos, portanto ele não é realmente um componente estrutural distinto.

Cada partição de sistema de arquivos é dividida em grupos de blocos. Estruturas como tabelas de inode são alocadas entre os grupos de blocos para que os blocos que são acessados juntos possam ser armazenados próximos uns dos outros no disco. Esse agrupamento reduz a necessidade de ficar procurando por todo o disco ao acessar blocos do mesmo arquivo.

Inodes são entradas de tabelas de comprimento fixo, cada uma das quais armazenando informações sobre um arquivo. Como o espaço para inodes é separado quando o sistema de arquivos é inicialmente estruturado, você deve decidir antecipadamente quantos deles criar. É impossível prever exatamente quantos arquivos (inodes) serão necessários algum dia; o Linux usa uma fórmula empírica para conjeturar um número apropriado, baseado no tamanho da partição e em um tamanho médio de arquivo.

Podemos ajustar o número de inodes tanto para cima como para baixo ao criarmos o sistema de arquivos: mais inodes para sistemas de arquivos com grande quantidade de arquivos pequenos (como repositórios de código-fonte) e menos inodes para sistema de arquivos com um número menor de arquivos grandes (por exemplo, um sistema de arquivos contendo um banco de dados). O valor padrão fornece inodes suficientes para preencher o disco com arquivos de 2 KB, o que normalmente é muito.

Um superbloco é um registro que descreve as características do sistema de arquivos. Ele contém informações sobre o comprimento de um bloco de disco, o tamanho e localização das tabelas de inodes, o mapa de blocos de disco e informações sobre utilização, o tamanho dos grupos de blocos e alguns outros poucos parâmetros importantes do sistema de arquivos. Pelo fato de danos ao superbloco poderem apagar algumas informações extremamente cruciais, várias cópias dele são mantidas em localizações espalhadas (no início de cada grupo de bloco).

Para cada sistema de arquivos montado, o kernel mantém tanto uma cópia em memória do superbloco como várias cópias em disco. A chamada de sistema **sync** despeja os superblocos que estavam armazenados em cache para suas localizações permanentes em disco, tornando o sistema de arquivos consistente num piscar de olhos. Esse salvamento periódico minimiza o nível de dano que ocorreria caso a máquina caísse quando o sistema de arquivos ainda não tivesse atualizado os superblocos. **sync** também descarrega os inodes modificados e blocos de dados em cache. As sincronizações são tradicionalmente feitas em intervalos de 30 segundos, mas as sincronizações ext3fs, a cada 5 segundos. A maioria dos sistemas realiza um **sync** a cada 30 segundos para minimizar a quantidade de dados perdida no caso de um travamento.

O mapa de blocos do disco de um sistema de arquivos é uma tabela dos blocos livres que ele contém. Quando são gravados arquivos novos, esse mapa é examinado para planejar um esquema de layout eficiente. O resumo de emprego dos blocos registra informações básicas sobre os blocos que já se encontram em uso.

Montando os sistemas de arquivos

Um sistema de arquivos deve ser montado antes de torna-se visível aos processos. O ponto de montagem para um sistema de arquivos pode ser qualquer diretório, mas os arquivos e subdiretórios abaixo dele não serão acessíveis enquanto um sistema de arquivos não for montado ali. Consulte a página 52 para informações adicionais sobre a montagem de sistemas de arquivos.

Após instalar um disco novo, você deve montar manualmente os novos sistemas de arquivos para testar se tudo está funcionando corretamente. Por exemplo, o comando:

```
# mount /dev/sda1 /mnt
```

montaria o sistema de arquivos na partição representada pelo arquivo de dispositivo **/dev/sd1** no diretório **/mnt**, que é um caminho tradicional utilizado para montagens temporárias. Se o sistema de arquivos for novo em folha, seu conteúdo deve ser parecido com o seguinte:

```
# ls /mnt
lost+found
```

O diretório **lost+found** é criado automaticamente quando você cria um sistema de arquivos. Ele é utilizado por **fsck** em emergências; não o remova. O diretório **lost+found** possui algum espaço extra pré-alocado de modo que **fsck** poderá armazenar aí arquivos "desassociados" sem ter de alocar entradas de diretório adicionais num sistema de arquivos instável. Podemos verificar o tamanho de um sistema de arquivos por meio do comando **df**. Eis um exemplo:

```
$ df /home
Filesystem        1K-blocks     Used   Available     Use%   Mounted on
/dev/hda5         4128448       697968 3220768       18%    /home
```

As unidades padrão informadas por **df** são blocos de 1 K, mas você pode utilizar a opção **-h** para solicitar valores legíveis por humanos (por exemplo, 2,4 GB).

Configurando a montagem automática

Em geral, é recomendável configurar o sistema para que ele monte os sistemas de arquivos locais de automaticamente em tempo de inicialização. O arquivo **/etc/fstab** tem uma lista dos dispositivos que correspondem aos sistemas de arquivos.

Um arquivo **fstab** que inclua o sistema de arquivos acima poderia ser parecido com o seguinte:

```
LABEL=/     /                ext3        defaults            1     1
none        /dev/pts         devpts      gid=5,mode=620      0     0
/dev/hda5   /home            ext3        defaults            1     2
none        /proc            proc        defaults            0     0
none        /dev/shm         tmpfs       defaults            0     0
/dev/cdrom  /media/cdrom     iso9660     ro,noauto,owner     0     0
/dev/hda3   /usr             ext3        defaults            1     2
/dev/hda6   /var             ext3        defaults            1     2
/dev/hda2   swap             swap        defaults            0     0
```

Há seis campos por linha, separados por espaços em branco. Cada linha descreve um único sistema de arquivos. Os campos são tradicionalmente alinhados para fins de legibilidade, mas o alinhamento não é obrigatório.

Consulte o Capítulo 16 para informações adicionais sobre NFS.

O primeiro campo dá o nome do dispositivo ou o rótulo que foi associado a ele por **e2label** (o `LABEL=`form). O arquivo **fstab** pode incluir montagens de sistemas remotos, caso em que o primeiro campo contém um caminho NFS, como na última linha. A notação *server: /export* indica o diretório */export* na máquina chamada *server*. Sistemas de arquivos sem armazenamento de apoio real (como **/proc** e **/dev/shm** anteriores) têm o marcador de lugar `none` no campo de dispositivo.

O segundo campo especifica o ponto de montagem e o terceiro campo nomeia o tipo de sistema de arquivos. O nome do tipo exato utilizado para identificar sistemas de arquivos locais varia dependendo da configuração do sistema. Os nomes `ext2` e `ext3` são utilizados para ext*fs e `reiserfs` denota ReiserFS.

O quarto campo lista as opções de montagem. A palavra-chave `defaults` indica uma combinação das opções `rw`, `suid`, `dev`, `exec`, `auto`, `nouser` e `async` (consulte **mount** na página man para obter detalhes sobre cada uma dessas opções). Talvez você queira incluir `acl` para ativar as listas de controle de acesso também; veja a página 62.

O modo de gerenciamento de diário do ext3 é configurado com a opção `data=`*mode*, onde *mode* é `ordered`, `writeback` ou `journal`. O modo é uma escolha operacional, não uma propriedade do próprio sistema de arquivos; essa é a razão pela qual ele aparece no **fstab** e não como um argumento para **mke2fs**.

As opções definem três diferentes relações de troca ou compensação entre desempenho e confiabilidade:

- O modo padrão, `ordered` (ordenado), garante que o sistema de arquivos sempre seja consistente e, como resultado, os arquivos jamais serão corrompidos por uma queda de sistema. Essa é a melhor opção para a maioria dos ambientes.
- O modo `writeback` pode resultar em arquivos corrompidos após uma queda de sistema, mas em alguns casos é mais rápido. Essa é realmente a maneira como funciona a maioria dos sistemas de arquivos journaling e não é substancialmente mais arriscada do que o comportamento de ext2 padrão.
- O modo `journal` (registro de ações) usa um arquivo journal maior, que pode vir a retardar a recuperação na reinicialização, mas pode ser mais rápido quando utilizado com um aplicativo de banco de dados.

O quinto campo no arquivo **fstab** especifica um valor de "freqüência de dump" que, teoricamente, pode ser utilizado por produtos de backup, mas normalmente não é utilizado.

fsck é descrito na página 92.

O sexto campo especifica o passo no qual **fsck** deve verificar o sistema de arquivos. Sistemas de arquivos com o mesmo valor nesse campo são verificados concorrentemente, se possível. Esse campo era muito importante antes do advento dos sistemas de arquivos com *journaling* quando **fsck** era um processo demorado, mas isso não é tão demorado agora. Não configure dois sistemas de arquivos não *journaling* no mesmo disco com o mesmo valor; caso contrário você fará com que a cabeça de leitura do disco permaneça avançando e retrocedendo de tal modo que o desempenho será degradado significativamente. Apenas sistemas de arquivos em discos separados devem ser verificados em paralelo.

mount, **umount**, **swapon** e **fsck** todos lêem o arquivo **fstab**, de maneira que é importante que os dados apresentados aqui sejam corretos e completos. **mount** e **umount** utilizam o arquivo **fstab** para descobrir o que você quer que seja feito caso especifique somente um nome de partição ou ponto de montagem na linha de comandos. Por exemplo, usando o arquivo **fstab** mostrado, o comando

```
# mount /media/cdrom
```

seria o mesmo que digitar

```
# mount -t iso9660 -o ro,noauto,owner /dev/cdrom /media/cdrom
```

O comando **mount -a** monta todos os sistemas de arquivos regulares listados no arquivo **fstab**; ele é usualmente executado por meio dos scripts de inicialização no momento da inicialização. O flag **-t** restringe a operação a certo tipo de sistemas de arquivos. Por exemplo,

```
# mount -at ext3
```

montaria todos os sistemas de arquivos de disco ext3. O comando **mount** lê **fstab** seqüencialmente. Portanto, sistemas de arquivos que são montados embaixo de outros sistemas de arquivos devem seguir suas partições pai no arquivo **fstab**. Por exemplo, a linha para **/var/log** deve seguir a linha para **/var** se **/var** for um sistema de arquivos separado.

O comando **umount** para desmontar sistemas de arquivos aceita uma sintaxe similar. Você não pode desmontar um sistema de arquivos que um processo está utilizando como seu diretório atual ou no qual arquivos estão abertos. Há maneiras de evitar essa restrição; veja a página 52.

Ativando swapping

Uma das primeiras vantagens do UNIX era sua implementação de memória virtual. Com esse recurso, o sistema operacional pode fingir que a máquina tem mais memória do que realmente possui. Se processos tentarem utilizar essa memória "extra", os discos do sistema serão utilizados como um tipo de RAM ultralenta. Manipular o conteúdo da memória para e a partir do disco é conhecido como paginação. Idealmente, havendo memória física suficiente para o sistema, essa atividade extremamente lenta nunca ocorre.

Partições brutas, em vez de sistemas de arquivos estruturados, são normalmente utilizadas para o espaço de swap, tornando a troca mais eficiente. Em vez de utilizar uma estrutura de sistema de arquivos para controlar o conteúdo da área de swap, o kernel mantém seu próprio mapeamento simplificado de blocos de memória para blocos de disco. Também é possível fazer swap para um arquivo em uma partição com sistema de arquivos, mas essa configuração é mais lenta do que utilizar uma partição dedicada.

Consulte a página 559 para informações adicionais sobre como dividir áreas de swap.

Quanto mais espaço de swap você tiver, mais memória virtual os processos poderão alocar. O melhor desempenho de memória virtual é atingido quando a área de swap é dividida entre várias unidades (ou melhor ainda, entre vários barramentos SCSI).

Você pode ativar a troca manualmente para um determinado dispositivo, mas, em geral, você vai querer que essa função seja executada automaticamente no momento da inicialização. Áreas de swap podem ser listadas no arquivo **fstab**, o mesmo arquivo que é utilizado para enumerar sistema de arquivos montáveis. Uma entrada de swap é semelhante a:

```
/dev/hda2          swap          swap          defaults              0       0
```

Durante a inicialização, um comando **swapon** (freqüentemente, **swapon** ou **swap**) é executado para ativar a troca em todas as partições de swap listadas no arquivo **fstab**.

7.5 HDPARM: CONFIGURANDO PARÂMETROS DA INTERFACE IDE

O programa **hdparm** interage com o driver IDE do Linux para obter e alterar parâmetros de disco. Entre outras coisas, **hdparm** pode configurar as opções de energia da unidade, ativar ou desativar o DMA, configurar o flag de leitura e imprimir informações detalhadas sobre a unidade. Você pode perceber ganhos significativos de desempenho ajustando alguns desses parâmetros de disco. Esse utilitário é específico ao IDE e não funciona nas unidades SCSI ou USB.

A sintaxe é:

hdparm [*opções*] [*dispositivo*]

Executar **hdparm** sem nenhuma opção imprime o estado atual de algumas configurações mais interessantes.

```
$ sudo /sbin/hdparm -d /dev/hdb

/dev/hdb:
    multcount     =  0 (off)
    IO_support    =  0 (default 16-bit)
    unmaskirq     =  0 (off)
    using_dma     =  0 (off)
    keepsettings  =  0 (off)
    readahead     = 256 (on)
    geometry      = 36481/255/63, sectors = 300069052416, start = 0
```

Da perspectiva do desempenho de disco, essas configurações deixam muito a desejar.

- O Direct Memory Access (DMA) está desativado. Todos os discos modernos podem enviar dados diretamente à memória, driblando a CPU. Quase sempre uma boa idéia é ativar o DMA, mas ocasionalmente isso resulta em falha em um hardware mal projetado. Sempre teste primeiro em um sistema que não está em produção.
- Suporte à E/S de 32 bits está desativado. Sem a E/S de 32 bits, os dados percorrem o barramento PCI em fragmentos de 16 bits, diminuindo potencialmente o desempenho pela metade.
- A interrupção não-mascarada está desativada. Ativar esse recurso (um pouco obscuro) permite que o driver de disco exiba as interrupções pendentes enquanto serve uma interrupção atual do disco.

hdparm inclui um teste de desempenho de unidade predefinido que ajuda a avaliar o impacto dessas configurações. Com esses valores padrão em uma unidade IDE mais antiga, nosso sistema produziu estes resultados:

```
$ sudo /sbin/hdparm -Tt /dev/hdb
```

```
/dev/hdb:
     Timing cached reads:       228 MB in  2.41 seconds =  94.70 MB/sec
     Timing buffered disk reads: 6 MB in  4.62 seconds   =   1.30 MB/sec
```

Leituras armazenadas em cache indicam a velocidade de transferência de dados no barramento IDE (independente do throughput da mídia de disco física), enquanto leituras armazenadas em buffer incluem o tempo da leitura dos discos físicos. Como você poderia esperar, o disco físico é um muito mais lento que o barramento. Mesmo assim, 1,3 MB/s é bem desolador.

Como o DMA é um recurso bem importante dos hardwares modernos, seu ajuste merece atenção. Unidades mais recentes suportam múltiplos modos DMA. A Tabela 7.2 resume as velocidades dos vários modos DMA para cada uma das tecnologias DMA comuns.

Tabela 7.2 Taxas de transferência máximas (em MB/s) para os vários modos DMA

Modo	0	1	2	3	4	5	6
PIO	3,3	5,2	8,3	11,1	16,7		
SDMA	2,4	4,2	8,3				
MDMA	4,2	13,3	16,7				
UDMA	16,7	25,0	33,3	44,4	66,7	100,0	133,0

Utilizando essa tabela, podemos facilmente identificar o argumento apropriado a passar para **hdparm**. Ativamos o DMA com o uso do UDMA2 inserindo o seguinte comando **hdparm**:

```
$ sudo /sbin/hdparm -d1 -Xudma2 /dev/hdb

/dev/hdb:
     setting using_dma to 1 (on)
     setting xfermode to 66 (UltraDMA mode2)
     using_dma = 1 (on)
```

Ativamos o suporte à E/S de 32 bits:

```
$ sudo /sbin/hdparm -c1 /dev/hdb

/dev/hdb:
     setting 32-bit IO_support flag to 1
     IO_support = 1 (32-bit)
```

Por fim, o valor da interrupção não mascarada é binário. Nós o ativamos com

```
$ sudo /sbin/hdparm -u1 /dev/hdb

/dev/hdb:
     setting unmaskirq to 1 (on)
     unmaskirq = 1 (on)
```

Podemos determinar o efeito dessas alterações executando o benchmark de **hdparm** novamente:

```
$ sudo /sbin/hdparm -Tt /dev/hdb

/dev/hdb:
     Timing cached reads:        256 MB in  2.00 seconds = 127.83 MB/sec
     Timing buffered disk reads: 40 MB in  3.01 seconds  = 13.30 MB/sec
```

Alguns ajustes de configuração do **hdparm** resultaram em aumentos significativos nos dois valores.

Tente diferentes valores na sua unidade a fim de encontrar as melhores configurações. É inteligente executar os benchmarks várias vezes em cada configuração para obter dados mais precisos.

7.6 FSCK: VERIFICANDO E REPARANDO SISTEMAS DE ARQUIVOS

Os sistemas de arquivos modernos são surpreendentemente confiáveis e mesmo implementações sem journaling parecem realizar um trabalho notável em lidar com um travamento inesperado de sistema ou anormal de hardware. Entretanto, os sistemas de arquivos podem se tornar danificados ou inconsistentes de diversas maneiras.

Qualquer hora que o kernel falha ou quando falta energia, pequenas inconsistências podem ser introduzidas nos sistemas de arquivos que estavam ativos imediatamente antes da pane. Já que o kernel armazena em buffers tanto blocos de dados como informações de resumo, a imagem mais recente do sistema de arquivos é dividida entre o disco e a memória. Durante uma pane, a parte da imagem em memória é perdida. Os blocos colocados em buffer são efetivamente "sobrescritos" com as versões que foram salvas em disco mais recentemente.

Há uma série de abordagens para corrigir esse problema. Danos menores podem normalmente ser resolvidos com o comando **fsck** ("filesystem consistency check", ou seja, verificação da consistência do sistema de arquivos). Essa não é uma forma muito elegante arquitetonicamente de abordar o problema, mas funciona muito bem para todas as inconsistências comuns.

Sistemas journaling (conhecidos como sistemas de arquivo de log) como o ReiserFS e ext3fs gravam metadados em um arquivo de log seqüencial que é despejado para o disco antes do retorno de cada comando. Os metadados acabam migrando do log para sua residência permanente dentro do sistema de arquivos. Se o sistema travar, o log pode ser retornado até o ponto de consistência mais recente; não é necessária uma checagem cruzada completa do sistema de arquivos. **fsck** ainda é executado no momento do boot para garantir que o sistema de arquivos se encontra num estado consistente, porém ele roda muito mais rápido quando comparado com uma verificação de sistema de arquivos ext2. Esse recurso pode nos poupar horas de tempo de inicialização em um sistema com grandes sistemas de arquivos.

Se alguma forma de journaling não estiver disponível, temos de aguardar o **fsck** fazer os seus procedimentos. Os cinco tipos mais comuns de danos são os seguintes:

- Inodes não referenciados
- Contagens de links inexplicavelmente grandes
- Blocos de dados não utilizados não registrados nos mapas de blocos
- Blocos de dados listados como livres e que também são utilizados em um arquivo
- Informações de resumo incorretas no superbloco

O **fsck** pode corrigir esses cinco problemas de maneira segura e automática. Se o **fsck** fizer correções em um sistema de arquivos, você deve executá-lo novamente até que o sistema de arquivos se torne completamente ordenado e o fsck não mostre nenhuma mensagem.

Os discos são normalmente verificados no momento do boot por meio de **fsck -p**, que examina todos os sistemas de arquivos locais listados em **/etc/fstab** e corrige os cinco erros listados acima. Se alguma forma de journaling estiver ativada, o **fsck** simplesmente informa que o sistema de arquivos está registrado em journal e retorna o log ao último estado consistente.

Você também pode executar **fsck -p** em um sistema de arquivos particular. Por exemplo:

```
# fsck -p /dev/sda5
```

Quando **fsck -p** lê o arquivo **fstab** para descobrir quais sistemas de arquivos devem ser verificados, ele obedece à seqüência indicada pelo último campo de cada entrada. Os sistemas de arquivos são verificados em ordem numérica crescente. Se dois sistemas de arquivos estiverem em discos diferentes, eles podem receber o mesmo número de seqüência; essa configuração faz com que **fsck** verifique-os simultaneamente, minimizando o tempo gasto na espera de E/S de disco. Verifique sempre a partição-raiz em primeiro lugar.

*Consulte a página man do **tune2fs** para algumas opções (menos úteis).*

O Linux pode ser configurado para forçar uma verificação do sistema de arquivos depois de ele montar um sistema de arquivos algumas vezes mesmo se todas as desmontagens foram "limpas". Essa precaução é uma boa higiene e, na maioria dos casos, o valor padrão (normalmente 20 montagens) é aceitável. Entretanto, nos sistemas que montam sistemas de arquivos com freqüência, como estações de trabalho desktop, utilizar **fsck** a cada 20 montagens pode ser cansativo. Para aumentar o intervalo para 50 montagens, utilize o comando **tune2fs**:

```
$ sudo /sbin/tune2fs -c 50 /dev/sda3
tune2fs 1.35 (28-Feb-2004)
Setting maximal mount count to 50
```

Erros que não caem nas cinco categorias anteriores são potencialmente graves. Eles fazem com que **fsck -p** peça ajuda e depois encerre. Nesse caso, execute **fsck** sem a opção **-p**. Quando executado em modo manual, **fsck** pede para você confirmar cada um dos reparos que ele pretende fazer. A lista a seguir mostra alguns dos erros que **fsck** considera perigosos o bastante para garantir a intervenção do operador.

- Blocos reivindicados por mais de um arquivo
- Blocos reivindicados fora do intervalo do sistema de arquivos
- Contagens de links muito pequenos
- Blocos não contabilizados

- Diretórios que se referem a inodes não alocados
- Vários erros de formato

Infelizmente, é difícil corrigir um disco manualmente sem ter conhecimento profundo da implementação do sistema de arquivos. Nunca tente gravar diretamente no sistema de arquivos por meio dos arquivos de dispositivos.

Na prática, esse estado de coisas significa que temos poucas opções exceto a de aceitar as correções sugeridas por **fsck**. Podemos minimizar os problemas registrando cuidadosamente as mensagens produzidas por **fsck**, já que algumas vezes elas fornecem uma dica sobre o arquivo ou arquivos que estão provocando problemas. Se **fsck** solicitar permissão para eliminar um arquivo, você deve tentar copiá-lo para um sistema de arquivos diferente antes de permitir que **fsck** prossiga. Esteja atento ao fato de que toda vez que você tentar acessar um sistema de arquivos danificado, corre o risco de provocar uma pane no sistema.

Consulte o Capítulo 9 para as informações sobre backups.

Se um sistema de arquivos danificado (um que o **fsck** não é capaz de reparar automaticamente) contém dados valiosos, não fique experimentando antes de fazer um backup consistente. Você pode tentar fazer um **dump** para o disco, mas já que **dump** tem como expectativa ler um sistema de arquivos não danificado, a imagem resultante pode ter falta de dados (ou o comando pode travar). A melhor política de segurança é aplicar **dd** ao disco inteiro para um arquivo de backup ou disco de backup.

Se **fsck** localiza um arquivo cujo diretório pai não pode ser determinado, ele coloca o arquivo no diretório **lost+found** no nível mais alto do sistema de arquivos. Como o nome atribuído a um arquivo só é registrado no diretório pai desse arquivo, nomes para arquivos órfãos não estarão disponíveis e os arquivos colocados em **lost+found** serão nomeados com seus números de inode. A tabela de inodes registra o UID do proprietário do arquivo, portanto retornar um arquivo ao seu proprietário original é relativamente fácil.

7.7 ADICIONANDO UM DISCO: UM GUIA PASSO A PASSO

Nesta seção analisamos a configuração de uma nova unidade de disco. Configuramos a unidade com várias partições, incluindo uma para o espaço de swap. Nas partições restantes, criamos sistemas de arquivos ext3fs.

Depois de seu novo disco ser instalado, é uma boa idéia certificar-se de que o sistema consiga detectar o novo dispositivo antes de fazer o boot no kernel. Se ele for um disco IDE, faça uma verificação para garantir que o disco seja reconhecido na tela de configuração da BIOS, que normalmente podemos acessar digitando uma seqüência de teclas mágicas antes de o sistema inicializar. Consulte os manuais que vêm com o seu computador ou placa mãe para obter informações específicas sobre configuração de BIOS para dispositivos IDE. Na maioria dos casos, não é necessária nenhuma configuração especial.

Muitas placas SCSI também possuem uma tela de configuração de BIOS que podemos invocar antes de o sistema ser inicializado. Se essa opção estiver disponível, você poderá varrer o barramento SCSI para ter certeza de que o novo dispositivo apareça. Se esse procedimento parar ou produzir uma mensagem de alerta, é possível que você tenha escolhido um ID de SCSI que já estivesse em uso ou que você não tenha instalado terminadores nos lugares corretos.

Você também pode utilizar a BIOS do SCSI para aplicar uma formatação de baixo nível em um disco. Essa operação leva um longo tempo em alguns discos e não pode ser interrompida; portanto, planeje com antecedência.

Consulte a página 602 para informações adicionais sobre a instalação de drivers de dispositivo.

Se uma placa SCSI não fornecer uma interface com o usuário própria, sempre podemos tentar inicializar o sistema e verificar as mensagens exibidas pelo kernel. Se não vir mensagem de um driver SCSI, é possível que você precise instalar o driver antes de o disco poder ser reconhecido pelo kernel.

Em nosso caso, observamos as seguintes mensagens de nosso adaptador host SCSI BusLogic:

```
scsi0 : BusLogic BT-948
scsi : 1 host.
  Vendor: SEAGATE   Model: ST446452W         Rev: 0001
  Type:   Direct-Access                      ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 3, lun 0
scsi0: Target 3: Queue Depth 28, Asynchronous
SCSI device sda: hdwr sector=512 bytes. Sectors=91923356 [44884 MB] [44.9 GB]
sda: unknown partition table
```

Ignore alertas acerca da tabela de partições, já que essa é a primeira vez que o disco foi utilizado. Assim que o sistema tiver terminado a inicialização, você poderá prosseguir para o processo de particionamento do disco.

Antes do particionamento, devemos primeiro verificar se os arquivos de dispositivos do disco já existem (eles devem existir). No Linux, os nomes para os arquivos de dispositivos de disco SCSI assumem a forma **/dev/sd**XN, onde X é uma

letra minúscula que identifica a unidade (**a** é o disco SCSI de menor numeração, **b** é o segundo menor e assim por diante[7]) e N é um número de partição iniciando em 1. Ao referir-se ao disco todo, simplesmente omita o número da partição. Não há nenhum dispositivo de disco do tipo caractere (bruto) no Linux.

Neste exemplo, nosso disco é o primeiro na cadeia SCSI. A primeira partição é portanto **/dev/sda1** e o disco como um todo é referido como **/dev/sda**. Se esses arquivos de dispositivo não existissem, poderíamos criá-los com **MAKEDEV** ou **mknod**.

O disco agora está pronto para ser particionado. Como acontece na maioria dos sistemas operacionais para PCs, a ferramenta utilizada para partição no Linux é chamada de **fdisk**. Embora todas as versões de **fdisk** façam praticamente a mesma coisa (elas implementam o sistema de partição padrão da Microsoft), há muitas variações entre elas. É aconselhável ler o manual referente ao seu sistema em particular para garantir que ele atenda àquilo que mostramos aqui.

```
# fdisk /dev/sda
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

The number of cylinders for this disk is set to 5721.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
```

Já que estamos utilizando apenas esse disco no nosso sistema Linux, podemos ignorar o alerta. No passado, às vezes era importante diminuir a primeira partição para assegurar seu funcionamento com uma BIOS antiga e com outros sistemas operacionais que podem estar instalados no sistema.

O programa **fdisk** é interativo; pressionando-se **m** é exibida uma lista de todos os seus comandos. Os utilizados aqui são os seguintes:

- **n**, ou **new**, para criar uma partição nova
- **t**, ou **type**, para alterar o tipo de uma partição
- **p**, ou **print**, para imprimir a tabela de partições
- **w**, ou **write**, para gravar a tabela de partições no disco

Como nosso disco ainda não tem partições, começaremos com a criação de uma. Se seu disco tiver partições definidas a partir de outros discos, talvez seja necessário removê-las com o comando **delete** do **fdisk** antes de criar novas. O programa **fdisk** não modificará nada até o informarmos para gravar a tabela de partições.

A tabela de partição tem espaço para quatro partições "primárias" que podem armazenar dados. Alternativamente, você pode criar uma partição "estendida", que é uma partição primária que aponta para uma outra tabela de partições, informando outras quatro partições "lógicas". Embora o emprego de partições estendidas possa suplantar a restrição normal de quatro partições, é mais simples adotar as partições primárias se apenas algumas forem necessárias e é isso que faremos neste caso:

```
Command (m for help): new
Command action
 e  extended
 p  primary partition (1-4): p
Partition number (1-4): 1
First cylinder (1-5721, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-5721, default 5721): +2G

Command (m for help): print
Disk /dev/sda: 255 heads, 63 sectors, 5721 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start   End    Blocks   Id    System
/dev/sda1            1   255    2048256  83    Linux
```

7. Note que essa letra refere-se à *ordem* dos números de destino dos dispositivos SCSI, não aos números em si. Se você adicionar ou remover um disco, todas as letras mudarão!

Criamos uma partição de swap de forma similar, exceto pelo fato de mudarmos o tipo de "Linux" para "Linux swap". Embora o kernel não se preocupe com o tipo de partição, alguns programas e scripts utilizam o tipo para descobrir o que é cada partição. Especificamos um tamanho de 2 GB, que é provavelmente um exagero para a maioria dos aplicativos, mas já que temos espaço em disco para gastar, podemos muito bem ser generosos. **mkswap** nos alertará se ele não puder utilizar todo o espaço que alocamos.

```
Command (m for help): new
   e   extended
   p   primary partition (1-4): p
Partition number (1-4): 2
First cylinder (256-5721, default 256): 256
Last cylinder or +size or +sizeM or +sizeK (256-1275, default 1275): 511
Command (m for help): type
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
```

A terceira partição, que contém o restante do disco, é definida de uma maneira semelhante. Vejamos a tabela de partições mais uma vez antes de gravá-la.

```
Command (m for help): print
Disk /dev/sda: 255 heads, 63 sectors, 5721 cylinders
Units = cylinders of 16065 * 512 bytes
    Device  Boot  Start    End     Blocks        Id     System
 /dev/sda1            1    255    2048256        83     Linux
 /dev/sda2          256    511    2056320        82     Linux swap
 /dev/sda3          512   5721   41849325        83     Linux
```

Um asterisco aparece próximo ao número de blocos se a partição não terminar em um limite de cilindro. Podemos eliminar a partição e recriá-la introduzindo um número de cilindros (conforme ilustrado acima) ou então conviver com o fato de que uma pequena quantidade de espaço em disco não possa ser utilizada. Estamos satisfeitos com a nova tabela de partições, de modo que podemos gravá-la no disco.

```
Command (m for help): write
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Agora estamos prontos para criar sistemas de arquivos. Para criar um sistema de arquivos ext3fs, simplesmente executamos **mke2fs -j** e especificamos o nome do dispositivo como um argumento.

```
# mke2fs -j /dev/sda1
mke2fs 1.36 (05-Feb-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
245280 inodes, 489974 blocks
...
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 34 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
done
```

Ao criar seu sistema de arquivos, você também poderia utilizar a opção **-J** para especificar explicitamente o tamanho do arquivo de diário (journal) que residirá no novo sistema de arquivos (**-J size=***x*) ou a identidade de um dispositivo externo que conterá o arquivo de diário (**-J device=***y*). Instalações típicas localizam o arquivo de *journal* (que deve ter entre 1.024 e 102.400 blocos do sistema de arquivos) dentro do sistema de arquivos em si.

O processo para criar o sistema de arquivos maior é o mesmo, mas isso provavelmente levará mais tempo. Se souber que não precisará de todos os inodes que **mke2fs** aloca por padrão, você poderá reduzir o número de inodes por grupo, acelerando **mke2fs** e reservando mais espaço para dados efetivos. Da mesma forma, você pode querer aumentar o número de inodes dos sistemas de arquivos que abrigarão um grande número de arquivos muito pequenos. É bem melhor ter inodes em excesso do que muito poucos, uma vez que não ter um número suficiente de inodes impedirá a criação de mais arquivos. Não é possível adicionar mais inodes depois de o sistema de arquivos ter sido criado. Caso se depare com essa situação, você precisará fazer um **dump** dos dados do sistema de arquivos para fita ou para um arquivo em uma outra partição, executar novamente **mke2fs** com um número maior de inodes (**-i**) e depois restaurar (**restore**) os dados para a partição. Tempo de diversão!

Executamos **fsck** em nossos sistemas de arquivos para garantir que eles foram criados apropriadamente. O flag **-f** força **fsck** para verificar novos sistemas de arquivos em vez de supor que eles estejam organizados.

```
# fsck -f /dev/sda1
fsck 1.36 (05-Feb-2005)
e2fsck 1.36 (05-Feb-2005)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda1: 11/245280 files (9.1% non-contiguous), 16629/489974 blocks
```

Podemos montar novos sistemas de arquivos assim que seus pontos de montagem forem criados:

```
# mkdir /bkroot
# mount /dev/sda1 /bkroot
# df /bkroot
Filesystem        1k-blocks      Used    Available      Use%    Mounted on
/dev/sda1         1981000        13      1878575        0%      /bkroot
```

Para garantirmos que o sistema de arquivos monte os novos sistemas de arquivos em tempo de inicialização, adicionamos uma linha a cada um deles ao arquivo **/etc/fstab/**. Cada linha lista, conforme solicitado, o nome do dispositivo, o ponto de montagem, o tipo de sistema de arquivos, as opções de montagem, a freqüência de backup e o número de passagem para **fsck**:

```
/dev/sda1  /bkroot        ext3   defaults       0       1
/dev/sda3  /new           ext3   defaults       0       2
```

Um carregador de inicialização tem de ser gravado no dispositivo de disco para torná-lo inicializável. Dependendo da instalação, o comando **lilo** ou o **grub** faz a instalação real. Consulte a página 19 para informações adicionais sobre como configurar e instalar um carregador de inicialização.

O passo final é criar o espaço de swap e adicioná-lo ao sistema. Inicializamos as partições de swap com **mkswap**, que recebe o nome do dispositivo da partição como um argumento. Não é mais necessário especificar o tamanho da partição de swap. Com a área de swap criada, nós a ativamos com o comando **swapon**. **swapon** também verifica se a área de swap foi adicionada adequadamente.

```
# mkswap /dev/sda2
Setting up swapspace version 1, size = 2105667584 bytes
# swapon /dev/sda2
# swapon -s
Filename   Type          Size          Used   Priority
/dev/hda5  partition     133020        688    -1
/dev/sda2  partition     2056316       0      -2
```

Como com sistemas de arquivos regulares, precisamos adicionar a nova partição de swap ao arquivo **/etc/fstab** para que o sistema se lembre dela na próxima vez em que reinicializarmos. A entrada a seguir é apropriada para nosso disco de exemplo:

```
/dev/sda2  swap   swap   defaults       0       0
```

Por fim, reinicializamos para testar as alterações feitas no arquivo **/etc/fstab** e nos certificarmos de que os novos sistemas de arquivos e o espaço de swap funcionam corretamente.

7.8 GERENCIAMENTO AVANÇADO DE DISCO: RAID E LVM

Os procedimentos que discutimos até agora para adicionar novos discos, sistemas de arquivos e áreas de swap são semelhantes aos usados na maioria dos sistemas UNIX. Entretanto, o Linux esconde na manga alguns truques adicionais com os

quais vários outros sistemas operacionais continuam a sonhar. Duas tecnologias distintas, RAID de software e gerenciamento de volume lógico, fornecem uma camada adicional de flexibilidade e confiabilidade ao gerenciamento de disco Linux.

Os discos rígidos falham com freqüência, e, mesmo com os backups atuais, as conseqüências de uma falha de disco em um servidor podem ser desastrosas. RAID, Redundant Array of Independent Disks, é um sistema que utiliza múltiplas unidades de disco rígido para distribuir ou replicar dados por vários discos. RAID não apenas ajuda a evitar a perda de dados, mas também minimiza o tempo de inatividade associado a uma falha de hardware (freqüentemente zero) e também aumenta potencialmente o desempenho. Sistemas RAID podem ser implementados no hardware, mas o que o sistema Linux implementa faz toda conexão necessária no software.

Uma segunda ferramenta, e igualmente útil, chamada LVM (*logical volume management*) ajuda os administradores a alocar de modo eficiente o espaço em disco disponível entre as partições. Imagine um mundo em que você não sabe exatamente qual o tamanho que uma partição deve ter. Seis meses depois de criar uma partição, você descobre que ela é excessivamente grande e que uma partição vizinha não tem espaço suficiente... Parece familiar? O LVM permite que o espaço seja realocado dinamicamente da partição gananciosa à partição que necessita de espaço.

Embora essas ferramentas sejam poderosas quando utilizadas individualmente, elas são especialmente potentes se combinadas. As seções a seguir dão uma visão geral conceitual dos dois sistemas e um exemplo que ilustra a configuração detalhada.

RAID de software Linux

Recentemente tivemos uma falha na controladora de disco em um importante servidor de produção. Embora os dados tenham sido replicados para várias unidades físicas, uma controladora RAID no hardware defeituoso destruiu os dados em todos os discos. Após isso, ocorreu um processo demorado de restauração a partir da fita, e isso após mais de dois meses antes de o servidor ter sido completamente recuperado. O servidor reconstruído agora conta com o software do kernel para gerenciar seu ambiente RAID, eliminando a possibilidade de uma outra falha da controladora RAID.

O RAID pode fazer duas coisas básicas. Primeiro, ele pode aprimorar o desempenho "dividindo" (*striping*) os dados em múltiplas unidades, permitindo assim que várias unidades funcionem simultaneamente para fornecer ou absorver um único fluxo de dados. Segundo, ele pode duplicar ou "espelhar" os dados em múltiplas unidades, diminuindo o risco associado com um disco falho individual. O RAID do Linux tem algumas diferenças sutis em relação ao RAID tradicional, mas ainda é dividido logicamente em vários níveis:

- O modo linear não fornece nenhuma redundância de dados ou ganhos de desempenho. Ele simplesmente concatena os endereços de bloco das múltiplas unidades para criar uma unidade virtual única (e maior).
- O nível RAID 0 é utilizado estritamente para aprimorar o desempenho. Ele utiliza duas ou mais unidades de tamanho igual para reduzir os tempos de gravação e acesso.
- O nível RAID 1 é o primeiro nível a oferecer redundância. Os dados são duplicados em duas ou mais unidades simultaneamente. Esse modo espelha os dados, mas prejudica o desempenho porque as informações devem ser gravadas mais de uma vez.
- O nível RAID 4 compete com (e consistentemente perde do) o nível RAID 5. Ele divide os dados, mas dedica um disco às informações de paridade, incorrendo assim em tempos de espera ao gravar no disco de paridade. A menos que você tenha uma razão muito boa para utilizar o RAID 4, ignore-o e dê preferência ao RAID 5.
- O RAID nível 5 é o local idílico do RAID. Dividindo tanto os dados como as informações de paridade, ele cria uma arquitetura redundante e, ao mesmo tempo, aprimora simultaneamente os tempos de gravação e de leitura. O RAID 5 requer pelo menos três unidades de disco.

O RAID de software foi incorporado ao kernel do Linux desde a versão 2.0, mas as versões anteriores eram falhas e incompletas. Recomendamos evitar implementações mais antigas do que aquelas do kernel 2.4.

Gerenciamento de volume lógico

LVM é um subsistema opcional que define um tipo de versão turbinada do particionamento de disco. Ele permite o agrupamento de discos individuais em "grupos de volumes". A capacidade agregada de um grupo de volume pode então ser alocada a volumes lógicos que são acessados como dispositivos de blocos comuns. O gerenciamento de volume lógico permite:

- Usar e alocar armazenamento em disco de maneira mais eficiente
- Mover volumes lógicos entre diferentes dispositivos físicos
- Aumentar e diminuir o tamanho de volumes lógicos com o sistema no ar
- Capturar "instantâneos" de sistemas de arquivos inteiros
- Substituir unidades on-line sem interrupção de serviços

Os componentes de um volume lógico podem ser agrupados de diversas maneiras. A concatenação mantém juntos os blocos físicos de cada dispositivo e alinha os dispositivos um após o outro. A divisão (por *striping*) intercala os componentes para que blocos virtuais adjacentes sejam realmente distribuídos por meio de vários discos físicos. Reduzindo os gargalos de um único disco, a divisão pode normalmente oferecer uma largura de banda maior e uma latência menor.

Uma configuração de exemplo com LVM e RAID

Nosso exemplo anterior ilustrou a configuração de um disco básico. Nesta seção, analisamos um procedimento de configuração que inclui tanto o RAID como o LVM. Esse tipo de configuração é especialmente útil para servidores de produção.

Nosso objetivo é criar um array RAID 5 a partir de três discos vazios. Na parte superior desse array RAID, definimos duas partições LVM, web1 e web2. Essa estrutura fornece várias vantagens em relação a um sistema tradicional:

- O RAID 5 verifica redundância; se um dos discos falhar, nossos dados permanecem intactos. Diferentemente do RAID 4, não importa qual disco falha!
- Graças ao LVM, as partições são redimensionáveis. Quando um webmaster entusiástico preenche a partição web2, podemos facilmente roubar algum espaço extra da web1.
- Conseqüentemente, mais espaço em disco seria necessário nas duas partições. Esse design permite que mais discos sejam adicionados ao array RAID 5. Depois que isso foi feito, os grupos LVM existentes podem ser estendidos para incluir o espaço adicional, tudo sem recriar nenhuma partição.

Depois de mostrar a configuração inicial, descreveremos como tratar um disco falho e mostraremos como redimensionar uma partição LVM.

No nosso sistema de exemplo, há quatro discos SCSI igualmente dimensionados:

```
# fdisk -l

Disk /dev/sda: 18.2 GB, 18210036736 bytes
255 heads, 63 sectors/track, 2213 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start         End     Blocks        Id     System
/dev/sda1     *       1          13     104391        83     Linux
/dev/sda2    14     144     1052257+        82        Linux  swap
/dev/sda3   145    2213    16619242+        8e        Linux  LVM1G

Disk /dev/sdb: 18.2 GB, 18210036736 bytes
255 heads, 63 sectors/track, 2213 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start   End     Blocks  Id     System

Disk /dev/sdc: 18.2 GB, 18210036736 bytes
255 heads, 63 sectors/track, 2213 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start   End     Blocks  Id     System

Disk /dev/sdd: 18.2 GB, 18210036736 bytes
255 heads, 63 sectors/track, 2213 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start   End     Blocks  Id     System
```

O primeiro disco SCSI, **/dev/sda**, contém as partições do nosso sistema. Os outros três (**sdb**, **sdc** e **sdd**) não têm nenhuma tabela de partição.

Inicialmente, criamos as partições em cada um dos nossos discos SCSI. Como os discos são idênticos, executamos o mesmo conjunto de comandos para cada um.

```
# fdisk /dev/sdb
...
Command (m for help): new
Command action
```

```
   e   extended
   p   primary partition (1-4): p
Partition number (1-4): 1
First cylinder (1-2213, default 1): <Enter>
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-2213, default 2213): <Enter>
Using default value 2213

Command (m for help): type
Selected partition 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): write
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Depois de definir os nomes das partições para os outros dois discos, é o momento de arregaçar as mangas e construir o array RAID. A maioria das distribuições modernas utiliza o comando **mdadm** individual para gerenciar o RAID. Versões anteriores do RHEL utilizavam o conjunto **raidtools**, mas como **mdadm** é mais poderoso e mais fácil de utilizar do que **raidtools**, é isso o que demonstraremos aqui.

O comando a seguir constrói um array RAID 5 a partir das nossas três partições SCSI:

```
# mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: array /dev/md0 started.
```

Enquanto o array está sendo construído, o arquivo **/proc/mdstat** mostra as informações do progresso:

```
# cat /proc/mdstat
Personalities : [raid5]
md0 : active raid5 sdb1[3] sdc1[1] sdd1[2]
35566336 blocks level 5, 64k chunk, algorithm 2 [3/2] [_UU]
      [====>.............]  recovery = 22.4% (3999616/17783168) finish=5.1min speed=44800K/sec
unused devices: <none>
```

Esse arquivo sempre reflete o estado atual do sistema RAID no kernel. Ele é útil para monitorar esse arquivo depois da adição de um novo disco ou da substituição de uma unidade defeituosa. (**watch cat /proc/mdstat** é um idioma útil.)

Quando a montagem do array está completa, vemos uma mensagem de notificação no arquivo **/var/log/messages**:

```
RAID5 conf printout:
--- rd:3 wd:3 fd:0
disk 0, o:1, dev:sdb1
disk 1, o:1, dev:sdc1
disk 2, o:1, dev:sdd1
```

O comando inicial de criação também serve para "ativar" o array (torná-lo disponível para uso), mas nas reinicializações subseqüentes precisamos ativar o array como um passo separado, normalmente fora de um script de inicialização. RHEL, Fedora e SUSE todos incluem scripts de inicialização de exemplo para o RAID.

mdadm tecnicamente não exige um arquivo de configuração, embora utilize um se ele for fornecido (em geral, **/etc/mdadm.conf**). Recomendamos veementemente o uso de um arquivo de configuração. Ele documenta a configuração RAID de maneira padrão, fornecendo aos administradores um lugar óbvio para procurar informações se ocorrer algum problema. A alternativa ao uso de um arquivo de configuração é especificar a configuração na linha de comando toda vez que o array é ativado.

mdadm --detail --scan faz um dump da configuração RAID atual em um arquivo de configuração. Infelizmente, a configuração que ele imprime não é muito completa. Os comandos a seguir criam um arquivo de configuração completo para nossa configuração de exemplo:

```
# echo DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1 > /etc/mdadm.conf
# mdadm --detail --scan >> /etc/mdadm.conf
# cat /etc/mdadm.conf
DEVICE /dev/sdb1 /dev/sdc1 /dev/sdd1
ARRAY /dev/md0 level=raid5 num-devices=3 UUID=21158de1:faaa0dfb:841d3b41:76e93a16
```

```
devices=/dev/sdb1,/dev/sdc1,/dev/sdd1
```

mdadm pode agora ler esse arquivo na inicialização ou no fechamento para gerenciar facilmente o array. Para ativar o array na inicialização utilizando o recém-criado **/etc/mdadm.conf**, executaríamos

```
# mdadm -As /dev/md0
```

Para interromper o array manualmente, utilizaríamos o comando

```
# mdadm -S /dev/md0
```

Os nossos três discos rígidos estão agora montados em um único disco RAID lógico. Agora é hora de definir os grupos de volume lógico em que é possível criar sistemas de arquivos expansíveis (e reduzíveis). A configuração LVM ocorre em algumas fases distintas:

- Criar (definir, na verdade) e inicializar os volumes físicos
- Adicionar os volumes físicos a um grupo de volumes
- Criar volumes lógicos no grupo de volumes

O conjunto LVM2 de ferramentas aborda todas essas tarefas e facilita o gerenciamento posterior dos volumes. **man lvm** é uma boa introdução a esse sistema e suas ferramentas.

Na terminologia LVM, os volumes físicos são as "coisas" agregadas para formar pools de armazenamento ("grupos de volumes"). "Volume físico" é, porém, um termo um pouco enganoso, porque os volumes físicos não precisam ter uma correspondência direta com os dispositivos físicos. Eles *podem* ser discos, mas também podem ser partições de disco ou (como neste exemplo) objetos RAID de alto nível com uma estrutura própria subjacente.

Comandos LVM iniciam com letras que deixam claro em qual nível de abstração eles operam: comandos **pv** manipulam volumes físicos, comandos **vg** manipulam grupos de volumes e comandos **lv** manipulam volumes lógicos.

Versões mais antigas do LVM exigiam que você executasse o comando **vgscan** como um passo inicial, mas isso não é mais necessário. Em vez disso, você começa inicializando diretamente cada dispositivo físico com **pvcreate**. Para esse exemplo, utilizamos o dispositivo RAID 5 **/dev/md0** que acabamos de criar.

```
# pvcreate /dev/md0
Physical volume "/dev/md0" successfully created
```

Essa operação destrói todos os dados do dispositivo ou partição, por isso fomos extramente cuidadosos! Embora só um único dispositivo físico tenha sido utilizado neste exemplo, o LVM permitiu adicionar múltiplos dispositivos de tipos diferentes a um único grupo de volumes.

Agora nosso dispositivo físico está pronto para ser adicionado a um grupo de volume:

```
# vgcreate LVM1 /dev/md0
Volume group "LVM1" successfully created
```

Para voltar e examinar nosso trabalho, utilizamos o comando **vgdisplay**:

```
# vgdisplay LVM1
--- Volume group ---
VG Name    LVM1
System ID
Format                    lvm2
Metadata Areas            1
Metadata Sequence No      1
VG Access                 read/write
VG Status                 resizable
MAX LV                    0
Cur LV                    0
Open LV                   0
Max PV                    0
Cur PV                    1
Act PV                    1
VG Size                   33.92 GB
PE Size                   4.00 MB
Total PE                  8683
Alloc PE / Size           0 / 0
Free                      PE / Size 8683 / 33.92 GB
VG UUID                   nhkzzN-KHmY-BfV5-6F6Y-3LFB-dpd5-JM5lMp
```

Os últimos passos são para criar volumes lógicos dentro do grupo de volumes LVM1 e criar as partições nos volumes. Criamos os dois volumes lógicos com 10 GB:

```
# lvcreate -L 10G -n web1 LVM1
Logical volume "web1" created

# lvcreate -L 10G -n web2 LVM1
Logical volume "web2" created
```

Agora que criamos dois volumes lógicos, web1 e web2, no nosso grupo de volume LVM1, podemos criar e montar nossos sistemas de arquivos.

```
# mke2fs -j /dev/LVM1/web1
...
# mke2fs -j /dev/LVM1/web2
...
# mkdir /web1 /web2
# mount /dev/LVM1/web1 /web1
# mount /dev/LVM1/web2 /web2
```

Os sistemas de arquivos estão finalmente prontos para uso. Adicionamos os novos sistemas de arquivos a **/etc/fstab** e reinicializamos o sistema a fim de assegurar que tudo funciona adequadamente.

Lidando com um disco falho

Nosso sistema muito bem arquitetado parece interessante agora, mas, por causa das múltiplas camadas em que está operando, as coisas podem ficar feias se feitas às pressas. Quando uma unidade de disco falha ou uma partição é corrompida (ou simplesmente fica cheia), é essencial saber como repará-la rápida e facilmente. Utilize as mesmas ferramentas da configuração inicial anterior para manter o sistema e recuperá-lo de falhas.

Considere o caso de um disco rígido falho. Como o RAID 5 fornece alguma redundância de dados, o array RAID 5 que construímos nas seções anteriores funcionará normalmente no caso de uma pane de disco; os usuários não serão necessariamente informados de quaisquer problemas. Você precisará prestar bastante atenção aos logs de sistema para perceber o problema do início (ou utilizar um programa que faça isso para você; consulte a página 157).

mdadm oferece uma opção útil que simula um disco falho:

```
# mdadm /dev/md0 -f /dev/sdc1
mdadm: set /dev/sdc1 faulty in /dev/md0

# tail /var/log/messages
May 30 16:14:55 harp kernel: raid5: Disk failure on sdc, disabling device. Operation continuing on
    2 devices
kernel: RAID5 conf printout:
kernel:  --- rd:3 wd:2 fd:1
kernel:  disk 0, o:1, dev:sdb1
kernel:  disk 1, o:0, dev:sdc1
kernel:  disk 2, o:1, dev:sdd1
kernel: RAID5 conf printout:
kernel:  --- rd:3 wd:2 fd:1
kernel:  disk 0, o:1, dev:sdb1
kernel:  disk 2, o:1, dev:sdd1
```

Como mostrado aqui, o log de sistema **/var/log/messages** contém informações sobre a falha (simulada) assim que ela ocorre. Informações semelhantes estão disponíveis no arquivo de status **/proc/mdstat** do RAID. Nesse ponto, o administrador deve adotar essas medidas:

- Remover o disco do array RAID
- Agendar o tempo de inatividade e desligar o computador (se necessário)
- Substituir a unidade física
- Adicionar a nova unidade ao array

Para remover a unidade da configuração RAID, utilize **mdadm**:

```
# mdadm /dev/md0 -r /dev/sdc1
mdadm: hot removed /dev/sdc1
```

Depois que o disco foi logicamente removido, você pode substituir a unidade. Um hardware com unidade "*hot-swappable*" permite fazer a alteração sem desligar ou reinicializar o sistema.

Se seus componentes RAID forem discos somente, você só deve substituí-los por uma unidade idêntica. Componentes baseados em partição podem ser substituídos por qualquer partição de tamanho semelhante, embora seja melhor para a correspondência de largura de banda se o hardware da unidade for semelhante. (Se sua configuração RAID for construída na parte superior das partições, você deverá executar **fdisk** para definir as partições apropriadamente antes de adicionar o disco de substituição ao array.)

No nosso exemplo, a falha é apenas uma simulação, portanto, podemos adicionar a unidade de volta ao array sem substituir nenhum hardware:

```
# mdadm /dev/md0 -a /dev/sdc1
mdadm: hot added /dev/sdc1
```

O Linux reconstrói o array e reflete o progresso, como sempre, em **/proc/mdstat**.

Realocando espaço de armazenamento

Ainda mais comum que os travamentos de disco são os casos em que os usuários ou arquivos de log enchem as partições. Experimentamos tudo: desde servidores utilizados para armazenamento pessoal de MP3 até um departamento cheio de lixo de e-mails.

Suponha que no nosso exemplo, **/web1** aumentou mais do que prevíamos e precisa de mais espaço. Redimensionar as partições LVM só envolve alguns passos curtos. Os comandos exatos dependem do sistema de arquivos em uso. Os passos no exemplo a seguir são para um sistema de arquivos ext3. Esses passos são:

- Examinar a configuração LVM atual
- Redimensionar as partições com **lvextend** e **ext2online**
- Verificar as alterações

Felizmente, deixamos algum espaço extra no nosso grupo de volume para aumentar o tamanho de **/web1**, portanto não temos de procurar espaço em outro volume. Utilizamos **vgdisplay** para ver o espaço disponível no grupo de volumes e **df** para determinar como realocá-lo:

```
# vgdisplay LVM1
--- Volume group ---
VG Name                 LVM1
System ID
Format                  lvm2
Metadata Areas          1
Metadata Sequence No    3
VG Access               read/write
VG Status               resizable
MAX LV                  0
Cur LV                  2
Open LV                 2
Max PV                  0
Cur PV                  1
Act PV                  1
VG Size                 33.92 GB
PE Size                 4.00 MB
Total PE                8683
Alloc PE / Size         5120 / 20.00 GB
Free                    PE / Size 3563 / 13.92 GB
VG UUID                 nhkzzN-KHmY-BfV5-6F6Y-3LF8-dpd5-JM51Mp

# df -h /web1
Filesystem              Size   Used   Avail  Use%   Mounted on
/dev/mapper/LVM1-web1   9.9G   7.1G   2.3G   76%    /web1
```

Esses comandos mostram 13,92 GB livres no grupo de volumes e 76% de uso em **/web1**. Adicionaremos 10 GB a **/web1**.

Primeiro utilizamos **lvextend** para adicionar espaço ao volume lógico, então **ext2online** para redimensionar as estruturas do sistema de arquivos e incluir o espaço adicional.

```
# lvextend -L+10G /dev/LVM1/web1
```

```
  Extending logical volume web1 to 20.00 GB
  Logical volume web1 successfully resized
# ext2online -d /dev/LVM1/web1
...
ext2_close
```

É isso aí. Examinar a saída de **df** novamente mostra as alterações:

```
# df -h /web1
Filesystem Size          Used   Avail  Use%   Mounted on
/dev/mapper/LVM1-web1    20G    7.1G   12G    38%    /web1
```

Os usuários do ReiserFS devem desmontar a partição antes de executar **lvextend**. Adicionalmente, uma ferramenta criativamente conhecida como **resize_reiserfs** é utilizada no lugar de **ext2online** para redimensionar o sistema de arquivos.

7.9 MONTANDO UNIDADES USB

Feliz e finalmente os disquetes tiveram o mesmo fim do pássaro dodó. Para substituí-los vieram as amigáveis, rápidas e divertidas unidades USB. Há muitas versões desses dispositivos: unidades pessoais do tamanho de um polegar, câmeras digitais, iPods e grandes discos externos, para citar alguns. Distribuições recentes do Linux incluem suporte nativo no kernel a toda essa útil quinquilharia eletrônica.

Ao conectar uma unidade USB, primeiro certifique-se de que o Linux reconhece o dispositivo. O comando a seguir lista os dispositivos USB que o kernel descobriu:

```
$ sudo /sbin/lsusb
Bus 001 Device 003: ID 0781:7108 SanDisk Corp.
Bus 001 Device 001: ID 0000:0000
```

Nesse caso, uma unidade SanDisk foi anexada. Se nenhum dispositivo for listado e o dispositivo estiver conectado, talvez o kernel não tenha suporte a USB e precisará ser recompilado.

Em seguida, descubra como o kernel identificou a unidade e qual arquivo de dispositivo ele utiliza para representá-la. As mensagens do kernel são registradas por meio do syslog.

```
$ sudo tail -n 20 /var/log/messages | grep kernel '
Jul 27 20:52:13 harp kernel: USB Mass Storage support registered.
Jul 27 21:02:57 harp kernel: usb 1-2: USB disconnect, address 2
Jul 27 21:14:09 harp kernel: ohci_hcd 0000:00:0f.2: wakeup
Jul 27 21:14:09 harp kernel: usb 1-2: new full speed USB device using addr 3
Jul 27 21:14:09 harp kernel: scsi3 : SCSI emulation for USB Storage devices
Jul 27 21:14:09 harp kernel: Vendor: SanDisk Model: Cruzer Titanium Rev: 2000
Jul 27 21:14:09 harp kernel: Type: Direct-Access ANSI SCSI revision: 02
Jul 27 21:14:09 harp kernel: SCSI device sde: 512-byte hdwr sectors (520 MB)
Jul 27 21:14:09 harp kernel: sde: Write Protect is off
Jul 27 21:14:09 harp kernel: sde: assuming drive cache: write through
Jul 27 21:14:10 harp kernel: sde: sde1
Jul 27 21:14:10 harp kernel: Attached scsi removable disk sde at scsi3, channel 0, id 0, lun 0
```

As mensagens do kernel indicam que isso é um SanDisk Cruzer Titanium de 520 MB. (Se você estiver no mercado, essa é uma excelente chave USB!) O kernel associou o dispositivo **/dev/sde** ao disco e o disco contém apenas uma única partição, **sde1**.

A unidade precisa ser montada pelo Linux antes de poder ser utilizada. Crie um ponto de montagem e monte a unidade:

```
$ sudo mkdir /mnt/usb
$ sudo mount /dev/sde1 /mnt/usb
```

A unidade foi agora montada em **/mnt/usb** e está pronta para uso. Para facilitar o processo da próxima vez que você utilizar a unidade, adicione esta linha a **/etc/fstab**:

```
/dev/sde1     /mnt/usb        auto    users,noauto,uid=ben,gid=users  0       0
```

As opções listadas detectam automaticamente o tipo de sistema de arquivos e permitem que o usuário ben monte e grave nessa unidade.

---

8. Veja em **/var/log/kern.log** sob os sistemas Debian e Ubuntu.

7.10 EXERCÍCIOS

E7.1 Quais conectores SCSI acompanham as variantes da especificação? Ignorando-se as diferenças nos conectores, quais são os problemas de compatibilidade entre as diversas versões SCSI?

E7.2 Qual a diferença entre formatar e particionar um disco? Qual a diferença entre particionar e criar um sistema de arquivos?

E7.3 Liste todos os comandos e argumentos que você utilizaria para criar um sistema de arquivos em um disco em cada uma das seguintes circunstâncias:

a) O disco será utilizado como armazenamento para diretórios de usuários.

b) O disco será utilizado como espaço de swap.

c) O disco armazenará a fila de e-mails numa grande fonte de spams.

d) O disco irá conter um banco de dados MySQL InnoDB.

E7.4 O conjunto de ferramenta LVM é poderoso, mas pode ser confuso se não for bem entendido. Pratique a adição, remoção e redimensionamento de discos em um grupo de volumes. Mostre como você removeria um dispositivo com sucesso de um grupo de volumes e o adicione a outro.

★ E7.5 Usando recursos impressos ou da Internet, identifique as unidades IDE e SCSI de melhor desempenho. Os benchmarks utilizados para avaliar essas unidades refletem a maneira como um servidor Linux ocupado utilizaria o seu disco de inicialização? Quanto em termos de custo adicional você pagaria pela SCSI e quanto de melhoria em termos de desempenho (se efetivamente alguma) você obteria pelo investimento adicional?

★ E7.6 Adicione um disco ao seu sistema. Crie uma partição no disco novo como uma partição-raiz de backup; instale um kernel e faça a inicialização a partir dele. Mantenha um arquivo journal de todos os passos necessários para completar essa tarefa. Talvez você ache o comando **script** útil. (Exige acesso root.)

★ E7.7 O que é um superbloco e para que ele é utizado? Pesquise a definição da estrutura de superblocos ext3fs nos arquivos de cabeçalho do kernel e discuta o que representa cada um destes campos na estrutura.

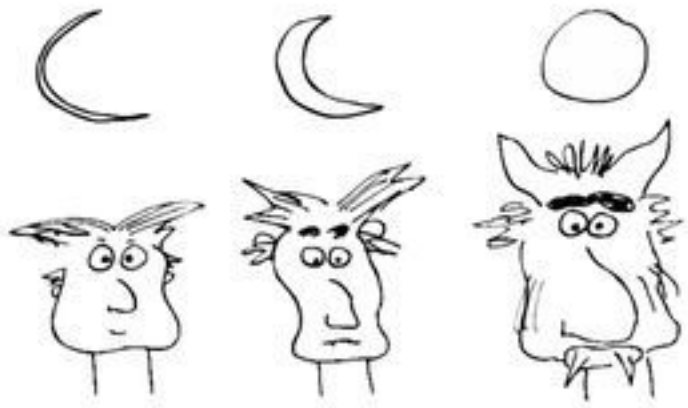
★ E7.8 Utilize **mdadm** e sua opção **-f** para simular um disco falho em um array RAID. Remova o disco do array e o adicione novamente. Como **/proc/mdstat** é em cada passo?

★ ★ E7.9 Que campos são armazenados em um inode de um sistema de arquivos ext3fs? Liste o conteúdo do inode que representa o arquivo **/etc/motd**. Onde o nome de arquivos desse arquivo é armazenado? (Ferramentas como **hexdump** e **ls -i** talvez ajudem.)

★ ★ E7.10 Examine o conteúdo de um arquivo de diretórios com um programa como **hexdump** ou **od**. Cada registro de comprimento variável representa um arquivo nesse diretório. Examine a estrutura de entradas de diretórios e explique a função de cada campo usando um exemplo de um arquivo de diretório real. A seguir, examine o diretório **lost+found** em qualquer sistema de arquivos. Por que há tantos nomes lá quando o diretório **lost+found** parece estar vazio?

★ ★ ★ ★ ★ E7.11 Escreva um programa que percorra o sistema de arquivos e imprima o conteúdo dos arquivos **/etc/motd** e **/etc/termcap**. Mas não abra os arquivos diretamente; abra o arquivo de dispositivo bruto para a partição-raiz e utilize as chamadas de sistema **seek** e **read** para decodificar o sistema de arquivos e localizar os blocos de dados apropriados. **/etc/motd** é normalmente curto e provavelmente irá conter apenas blocos diretos. **/etc/termcap** provavelmente exigirá a decodificação dos blocos indiretos. Dica: ao ler os arquivos de cabeçalho de sistema, certifique-se de ter localizado a estrutura de inodes em disco do sistema de arquivos, e não a estrutura de inodes em memória. (Exige acesso root.)

8 Processos periódicos



A chave para manter-se no controle de um sistema é automatizar o maior número possível de tarefas. Por exemplo, um programa **adduser** é capaz de adicionar usuários de forma mais rápida que nós e com uma chance menor de cometer erros. Quase todas as tarefas podem ser codificadas em um script em Perl, Python, shell ou **expect**.

Em geral, é mais útil ter um script ou comando executado sem qualquer intervenção humana. Poderíamos, por exemplo, querer que um script verificasse (digamos, de meia em meia hora) se os roteadores e as pontes de uma rede estão funcionando corretamente e fazer com que esse script envie um e-mail para nós quando problemas forem descobertos.[1]

8.1 CRON: AGENDANDO COMANDOS

Sob o Linux, a execução periódica é normalmente tratada pelo daemon **cron**. **cron** é iniciado quando o sistema inicializa e permanece em execução enquanto o sistema estiver de pé. **cron** lê um ou mais arquivos de configuração contendo listas de linhas de comandos e datas/hora em que esses comandos são invocados. As linhas de comandos são executadas por **sh**; portanto, qualquer coisa que podemos fazer manualmente também pode ser feita com o **cron**.[2]

Originalmente, o **cron** apareceu na árvore da família UNIX na década de 1970. As distribuições do Linux incluem uma versão conhecida como ISC **cron** ou "Vixie-**cron**", cujo nome advém do seu autor, Paul Vixie. Ela é uma moderna reformulação que fornece funcionalidades adicionais e menos confusão.

Um arquivo de configuração **cron** é chamado de "crontab", abreviatura de "cron table". **cron** procura arquivos crontab em três locais: **/var/spool/cron** (**/var/spool/cron/tabs** no SUSE e **/var/spool/cron/crontabs** no Debian), **/etc/cron.d** e **/etc/crontab**.

Os arquivos crontab para usuários individuais são armazenados sob **/var/spool/cron**. Tipicamente, há (pelo menos) um arquivo crontab por usuário: um para root, um para jsmith e assim por diante. Os arquivos crontab são nomeados com os nomes de login dos usuários a quem pertencem, e **cron** utiliza esses nomes de arquivos para descobrir qual UID utilizar ao executar os comandos que cada arquivo contém. O comando **crontab** transfere arquivos crontab de e para esse diretório.

Arquivos crontab que agendam tarefas de manutenção de sistema e outras tarefas definidas pelo administrador de sistemas são armazenados no arquivo **/etc/crontab** e em outros arquivos localizados no diretório **/etc/cron.d**. Esses arquivos possuem um formato ligeiramente diferente dos arquivos crontab de usuário, pois permitem que comandos sejam executados como um usuário arbitrário. **cron** trata as entradas **/etc/crontab** e **/etc/cron.d** exatamente da mesma maneira. Em geral, **/etc/crontab** é proposto como um arquivo para o administrador de sistemas manter manualmente, enquanto **/etc/cron.d** é fornecido como um local onde pacotes de software podem instalar todas as entradas crontab de que eles possam vir a precisar.

Quando **cron** é iniciado, ele lê todos os seus arquivos de configuração, armazena-os na memória e então entra em estado de repouso. A cada minuto, **cron** desperta, verifica os horários de modificação nos arquivos crontab, recarrega quaisquer arquivos que foram modificados e então executa quaisquer tarefas programadas para aquele instante antes de retornar ao estado de repouso.

Por razões que não são claras, **cron** foi renomeado para **crond** nos sistemas Red Hat.

cron registra em log suas atividades por meio do syslog, utilizando o recurso "cron" com a maioria das mensagens submetidas no nível "info". Configurações syslog padrão geralmente enviam dados do log **cron** ao seu próprio arquivo.

---

1. Muitas instalações vão ainda mais longe e enviam uma mensagem de texto ao pager de um administrador assim que um problema é detectado.
2. **/bin/sh** sob o Linux é realmente um link para o shell **bash**, uma versão aprimorada (e reimplementada) do shell Bourne tradicional encontrado em sistemas UNIX. Você também pode configurar **cron** para utilizar outros shells.

8.2 O FORMATO DE ARQUIVOS CRONTAB

Todos os arquivos crontab em um sistema compartilham um formato similar. Comentários são introduzidos com o sinal # na primeira coluna de uma linha. Cada linha não comentada contém seis ou sete campos e representa um comando:

```
minuto hora dia mês diaDaSemana [nomeDoUsuário] comando
```

Os primeiros seis campos são separados por um espaço em branco, mas, dentro do campo *comando*, o espaço em branco é considerado literalmente. *nomeDoUsuário* é encontrado somente em **/etc/crontab** e nos arquivos do diretório **/etc/cron.d**; ele especifica o usuário em nome do qual o *comando* deve ser executado. Esse campo não está presente ou é necessário nos arquivos crontab específicos (aqueles armazenados em **/var/spool/cron**), pois o UID é implícito pelo nome de arquivo.

Os campos *minuto*, *dia*, *mês* e *diaDaSemana* informam quando executar o comando. As interpretações desse comando são mostradas na Tabela 8.1 a seguir.

Tabela 8.1 Especificações de data/hora para **crontab**

Campo	Descrição	Intervalo
minuto	Minutos da hora	0 a 59
hora	Hora do dia	0 a 23
dia	Dia do mês	1 a 31
mês	Mês do ano	1 a 12
diaDaSemana	Dia da semana	0 a 6 (0 = Domingo)

Cada um dos campos relacionados a data/hora pode conter:

- Um asterisco, que coincide com tudo
- Um inteiro único, que coincide exatamente
- Dois inteiros separados por um traço, coincidindo com um intervalo de valores
- Uma série de inteiros ou intervalos separados por vírgulas, coincidindo com qualquer valor listado

Intervalos de valor podem incluir um valor incremental. Por exemplo, a série 0,3,6,9,12,15,18 pode ser escrita mais concisamente como 0-18/3. Você também pode utilizar mnemônica de texto para os nomes dos meses e dias, mas não em combinação com os intervalos.

A especificação de data/hora

```
45  10  *  *  1-5
```

significa "10h45 da manhã, de 2ª à 6ª". Uma dica: nunca coloque um asterisco no primeiro campo a menos que queira que o comando seja executado a cada minuto.

Há uma ambigüidade potencial a observar com os campos *diaDaSemana* e *dia*. Cada dia é tanto um dia da semana como um dia do mês. Se *diaDaSemana* e *dia* são especificados, um dia precisa satisfazer somente uma de duas condições para ser selecionado. Por exemplo,

```
0,30  *  13  *  5
```

significa "a cada meia hora na 6ª feira e a cada meia hora no 13º dia do mês" e não "a cada meia hora da sexta-feira 13".

comando é a linha de comando **sh** a ser executada. Ela pode ser qualquer comando shell válido e não deve ser colocada entre aspas. *comando* é considerado para continuar até o final da linha e pode conter espaços ou tabulações.

Um sinal de porcentagem (%) é utilizado para indicar novas linhas dentro do campo *comando*. Somente o texto até o primeiro sinal de porcentagem é incluso no comando em si; as linhas remanescentes são dadas ao comando como entrada padrão.

Eis alguns exemplos de comandos crontab válidos:

```
echo The time is now `date` > /dev/console
mail -s Reminder evi@anchor % Don't forget to write your chapters.
cd /etc; /bin/mail -s "Password file" evi < passwd
```

E aqui alguns exemplos completos de entradas crontab:

```
30  2  *  *  1    (cd /users/joe/project; make)
```

Essa entrada será ativada às 2h30 da madrugada de cada segunda-feira. Ela executará **make** no diretório **/users/joe/project**. Uma entrada como essa poderia ser utilizada para iniciar uma longa compilação em um horário em que os demais usuários não estariam usando o sistema. Usualmente, toda saída produzida por um comando **cron** é enviada para o "proprietário" do crontab.[3]

```
20  1  *  *  *    find /tmp -atime +3 -exec rm -f {  } ';'
```

Esse comando será executado todo dia à 1h20. Ele elimina todos os arquivos do diretório **/tmp** que não foram acessados nos últimos 3 dias.

```
55  23  *  *  0-3,6       /staff/trent/bin/checkservers
```

Essa linha executa **checkservers** todos os dias às 23h55 exceto às quintas e sextas-feiras.

Também é possível especificar variáveis de ambientes e seus valores em um arquivo crontab. Consulte a página man **crontab**(5) para detalhes adicionais.

8.3 GERENCIAMENTO DE CRONTAB

crontab *nomeDoArquivo* instala *nomeDoArquivo* como seu crontab, substituindo qualquer versão anterior. **crontab -e** verifica uma cópia de seu crontab, ativa o seu editor sobre ele (conforme especificado pela variável de ambiente EDITOR) e então o submete novamente ao diretório crontab. **crontab -l** lista o conteúdo de seu crontab para a saída padrão e **crontab -r** o elimina, deixando-o sem nenhum arquivo crontab.

Root pode fornecer argumento *nomeDoArquivo* para editar ou visualizar crontabs de outros usuários. Por exemplo, **crontab -u jsmith -r** apaga o crontab pertencente ao usuário jsmith.

Sem argumentos de linha de comando, a maioria das versões do **crontab** tentará ler um crontab a partir da entrada padrão. Caso entre acidentalmente nesse modo, não tente sair com <Control-D>; fazer isso apagará o crontab inteiro. Em vez disso, utilize <Control-C>. Algumas versões foram modificadas para exigir que você forneça um traço como o argumento *nomeDoArquivo* a fim de fazer com que o **crontab** preste atenção à sua entrada padrão.

Dois arquivos de configuração, **/etc/cron.deny** e **/etc/cron.allow**, especificam quais usuários podem submeter arquivos crontab. Se o arquivo allow existir, então ele contém uma lista de todos os usuários que podem submeter crontabs, um por linha. Nenhuma pessoa não listada pode chamar o comando **crontab**. Se o arquivo allow não existir, então o arquivo deny é verificado. Ele, também, é apenas uma lista de usuários, mas o significado é inverso: todo mundo, exceto os usuários listados, tem permissão de acesso. Se nem o arquivo allow nem o arquivo deny existir, a maioria dos sistemas só permitirá que root submeta crontabs. (O Debian e o Ubuntu assumem o padrão de permitir submissões por todos os usuários.)

É importante notar que o controle de acesso é implementado por **crontab**, não por **cron**. Se um usuário colocar um arquivo crontab no diretório apropriado por outros meios, **cron** executará cegamente os comandos que ele contém

8.4 ALGUNS USOS COMUNS PARA CRON

Um número de tarefas padrão é essencialmente feito para elas serem chamadas pelo cron, e isso é usualmente o que existe no crontab do root.

Nesta seção, analisamos uma série de tarefas desse tipo e as linhas crontab utilizadas para implementá-las.

Os sistemas Linux freqüentemente vêm com algumas entradas crontab pré-instaladas, principalmente em **/etc/cron.d**. Se quiser desativar essas entradas padrão, remova o caractere de comentário inserindo um sinal de cerquilha (#) no começo de cada linha. Não as elimine; talvez você vá precisar delas no futuro.

Além do mecanismo **/etc/cron.d**, as distribuições Linux também pré-instalam entradas crontab que executam os scripts em um conjunto de diretórios bem conhecidos, fornecendo portanto uma outra forma para os pacotes de software instalarem tarefas periódicas sem a necessidade de editar nenhum arquivo crontab. Por exemplo, scripts em **/etc/cron.daily** são executados uma vez por dia e scripts em **/etc/cron.weekly** são executados uma vez por semana.

Muitos sites passaram por pequenos problemas de rede, sutis, mas recorrentes, que ocorrem porque os administradores configuram **cron** para executar o mesmo comando em centenas de máquinas exatamente ao mesmo tempo. A sincronização do relógio com o NTP piora o problema. Esse problema é fácil de corrigir com um script aleatório de retardo ou com um ajuste no arquivo de configuração, mas pode ser difícil diagnosticá-lo porque os sintomas são resolvidos bem rapidamente e de uma maneira completa.

---

3. Isto é, o nome do usuário com o qual o arquivo crontab é nomeado. Na maioria dos sistemas (mas não em todos), o proprietário real dos arquivos crontab é root.

Limpando o sistema de arquivos

Alguns dos arquivos em qualquer sistema Linux são puro lixo (não, não os arquivos de sistema). Por exemplo, quando um programa trava, o kernel pode gravar um arquivo chamado **core** que contém uma imagem do espaço de memória do programa.[4] Os arquivos core são úteis para os desenvolvedores de software, mas, para os administradores, eles normalmente são um desperdício de espaço. Os usuários normalmente nem sabem a respeito dos arquivos core, de modo que eles tendem a não os eliminar por conta própria.

O NFS, Network File System, é descrito no Capítulo 16.

O NFS é outra fonte de arquivos extras. Como os servidores NFS são sem estado, eles precisam utilizar uma convenção especial para preservar arquivos que foram eliminados localmente, mas que continuam em uso por uma máquina remota. A maioria das implementações renomeia esses arquivos para **.nfs***xxx*, onde *xxx* é um número. Várias situações podem resultar no esquecimento e no abandono desses arquivos quando supostamente eles deveriam ter sido excluídos.

Muitos programas criam arquivos temporários em **/tmp** ou **/var/tmp** que não são apagados por um motivo ou por outro. Alguns programas, especialmente os editores, adoram criar cópias de backup de cada arquivo com o qual eles trabalham.

Uma solução parcial para o problema de arquivos lixo é instituir algum tipo de reivindicação noturna de espaço em disco por parte do **cron**. Os sistemas modernos geralmente já vêm com algo assim configurado, mas é uma boa idéia revisar o comportamento padrão de seu sistema para certificar-se de que é apropriado para o seu caso.

Abaixo estão vários exemplos de "limpeza" implementados com o comando **find**.

```
find / -xdev -type f '(' -name core -o name 'core.[0-9]*' ')' -atime +7 -exec rm -f {   } ':'
```

Esse comando elimina imagens de memória que não foram acessadas durante a última semana. O argumento **-xdev** assegura que **find** não cruzará outros sistemas de arquivos, exceto o sistema de arquivo-raiz; essa restrição é importante nas redes em que muitos sistemas de arquivos podem ser montados de uma maneira cruzada. Se quiser limpar mais de um sistema de arquivos, utilize um comando separado para cada um (observe que **/var** freqüentemente é um sistema de arquivos separado).

A convenção para atribuição de nomes a arquivos core pode ser configurada pelo administrador de sistemas por meio do arquivo **/proc/sys/kernel/core_pattern**. O comando **find** anterior só trata os nomes padrão (**core** ou **core.***PID*). O argumento **-type f** é importante porque a origem do kernel do Linux contém um diretório chamado **core** e você não iria querer excluir isso.

```
find / -xdev -atime +3 '(' -name '#*' -o -name '.#*' -o -name '*.CKP' -o -name '*~' -o -name
    '.nfs*' ')' -exec rm -f {   } ':'
```

Esse comando exclui arquivos que não foram acessados por três dias e que iniciam com **#** ou **.#** ou **.nfs** ou terminam com ~ ou **.CKP**. Esses padrões são típicos de várias espécies de arquivos temporários e de backup de editores.

Por razões de desempenho, alguns administradores usam a opção de montagem `noatime` para evitar que o sistema de arquivos mantenha registros de data/hora de acesso. Essa configuração confundirá os dois comandos **find** mostrados anteriormente porque os arquivos parecerão não terem sido referenciados ainda que estivessem ativos há pouco tempo. Infelizmente, o modo de falha é excluir esses arquivos; certifique-se de que você mantém data/horas de acesso antes de utilizar esses comandos como mostrado.

```
cd /tmp; find . ! -name . ! -name lost+found -type d -mtime +3 -exec /bin/rm -rf {   } ':'
```

Esse comando elimina recursivamente todos os subdiretórios de **/tmp** não modificados nas últimas 72 horas. Arquivos de texto puro em **/tmp** são eliminados no momento do boot do sistema pelos scripts de inicialização de sistema, mas alguns sistemas não eliminam diretórios. Se um diretório chamado **lost+found** existir, ele será tratado especialmente e não será eliminado. Isto é importante caso **/tmp** seja um sistema de arquivos separado. Consulte a página 94 para informações adicionais sobre **lost+found**.

Se você usar qualquer um desses comandos, deve ter certeza de que os usuários estejam alerta sobre as políticas de limpeza.

Distribuição de arquivos de configuração via rede

Consulte o Capítulo 17 para informações adicionais sobre o compartilhamento de arquivos de configuração.

Se estiver mantendo uma rede de máquinas, normalmente é conveniente manter para toda a rede uma única versão dos arquivos de configuração como o banco de dados de aliases de e-mail (normalmente **/etc/mail/aliases**). Versões mestre desses arquivos podem ser distribuídas toda noite com um script **rsync**, **rdist** ou **expect**.

4. A palavra "core" significa "memória". Esse termo é originário dos primeiros sistemas computacionais que usavam como elementos de memória pequenos anéis de ferrita montados numa malha trançada.

Algumas vezes, é necessário pós-processamento. Por exemplo, talvez você precise executar o comando **newaliases** para converter um arquivo de texto de aliases de e-mail para o formato *hashed* utilizado por **sendmail** se a opção `AuotRebuildAliases` não estiver configurada em seu arquivo **sendmail.cf**. Talvez você também precise carregar arquivos em um banco de dados administrativo como NIS.

Rodízio de arquivos de log

O Linux realiza um excelente trabalho de gerenciamento de seus arquivos de log, mas alguns deles crescem sem limite até serem reinicializados manualmente. Há várias maneiras de impedir que logs excedam seus limites; a mais simples é truncá-los em intervalos periódicos.

Uma estratégia mais conservadora é realizar um "rodízio" dos arquivos de log, mantendo várias versões mais antigas de cada um deles. Esse esquema evita que os arquivos de log fiquem fora de controle, mas jamais o deixa sem informações de log recentes. Como o rodízio de logs é um evento recorrente e programado regularmente, trata-se de uma tarefa ideal para **cron**. Consulte a seção *Rodízio de arquivos de log* no Capítulo 10 para mais detalhes.

8.5 OUTROS AGENDADORES: ANACRON E FCRON

Em geral, **cron** não executa comandos que foram perdidos enquanto o sistema esteve fora do ar ou quando ocorrem grandes descontinuidades no fluxo de tempo. Entretanto, o Vixie-**cron** faz um esforço de boa-fé a fim de fazer a coisa certa para os trabalhos agendados com base em uma freqüência menor que uma hora quando o fuso horário é alterado em pelo menos três horas. Ele faz um bom trabalho para tratar pequenos ajustes como aqueles relacionados ao horário de verão nos Estados Unidos.

Os laptops e outras máquinas que são ligados esporadicamente têm um relacionamento mais íntimo com o **cron** e podem se beneficiar de algum backup. Um complemento interessante ao **cron** pode ser localizado no **anacron**, que é um scheduler baseado em intervalos. Se **anacron** não estiver incluído na instalação padrão, você poderá instalá-lo por meio do gerenciamento de sistemas de pacotes padrão do seu sistema.

Em vez de especificar que um comando particular seja executado às segundas-feiras às 2h da manhã, no mundo **anacron** você simplesmente solicita que ele seja executado todas as semanas. Diferentemente do **cron**, o **anacron** mantém um registro da data/hora de execução mais recente de cada comando; assim, ele pode simplesmente comparar essa data/hora, o intervalo especificado, com a data/hora atual para determinar se o comando precisa ser executado novamente.

anacron fornece vários recursos que ajudam evitar que uma máquina recém-inicializada fique submersa sob uma onda de tarefas de manutenção simultâneas. Cada tarefa pode ter seu próprio retardo de inicialização, e o **anacron** pode serializar os comandos que ele executa de modo que somente um comando esteja ativo de cada vez.

No geral, o agendamento de intervalos talvez seja mais útil do que o agendamento de datas/horas fixas para tarefas administrativas. Infelizmente, o **anacron** requer que seus intervalos sejam expressos em dias, e esse fato limita os recursos do **anacron** como uma ferramenta de agendamento de primeira linha.

O próprio **anacron** deve expirar o **cron**. Embora a granularidade de agendamento do **anacron** esteja baseada em dias, faz sentido executar o **anacron** mais freqüentemente do que uma vez por dia — se soubesse que o **cron** consegue executar o **anacron** consistentemente no mesmo horário todos os dias, você provavelmente não utilizaria o **anacron**. De maneira semelhante, faz sentido também executar o **anacron** na inicialização do sistema.

Uma remodelagem mais agressiva do **cron** que inclui recursos do tipo **anacron** pode ser localizada no **fcron**; visite fcron.free.fr. Diferentemente do **anacron**, o principal objetivo do **fcron** é substituir o Vixie-**cron**. Ele permite algumas especificações sofisticadas de sincronização que não são expressas nas linguagens de configuração do **cron** e do **anacron**. Contudo, precisamos questionar se o valor prático desses recursos justifica a complexidade extra do **fcron** e a dificuldade de utilizar um agendador não-padrão.

8.6 EXERCÍCIOS

E8.1 Um usuário local em seu sistema tem abusado de seus privilégios sobre crontab executando tarefas caras em intervalos regulares. Depois de ter solicitado a ele várias vezes para parar com esse tipo de procedimento, você se vê forçado a revogar os privilégios desse usuário. Liste as etapas necessárias para eliminar o crontab desse usuário e certificar-se de que ele não adicionará um novo.

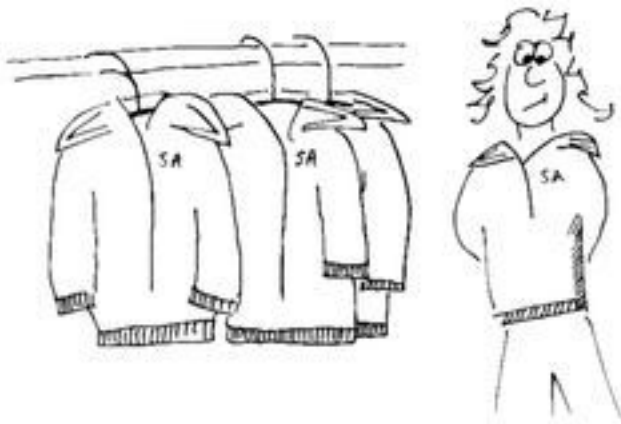
E8.2 Imagine três tarefas (além daquelas já mencionadas ao longo deste capítulo) que poderiam ser executadas periodicamente. Crie entradas crontab para cada tarefa e especifique em quais arquivos crontab as entradas devem entrar.

E8.3 Escolha três entradas a partir dos arquivos crontab do seu sistema. Decodifique cada uma delas e descreva quando é executada, o que faz e por que você imagina que a entrada é necessária. (Exige acesso de root.)

★ E8.4 Crie um script que mantenha seus arquivos de inicialização (**~/.[a-z]***) sincronizados entre todas as máquinas em que você tem uma conta. Programe esse script para ser executado regularmente a partir do **cron**. (É seguro copiar cegamente cada arquivo cujo nome inicia com um ponto? Você deve fazer o backup dos arquivos sendo substituídos nas máquinas de destino antes de eles serem sobrescritos?)

★ E8.5 Usando as páginas de manual para os comandos **du**, **sort** e **head** como referências, crie um script que determine os dez maiores diretórios de usuários no sistema. Programe o script para ser executado toda segunda à meia-noite e faça com que ele envie o relatório dele para você. Dica: será útil utilizar uma ordem numérica invertida. (Exige acesso de root.)

9 Backups



Na maioria das instalações as informações armazenadas nos computadores valem mais do que os próprios computadores. Elas também são muito mais difíceis de serem substituídas. Proteger essas informações é uma das tarefas mais importantes do administrador de sistema (e, infelizmente, a mais tediosa).

Há centenas de maneiras criativas e não tão criativas de perder dados. Bugs de software rotineiramente corrompem arquivos de dados. Muitos usuários apagam acidentalmente o trabalho de suas vidas. Hackers e funcionários descontentes apagam discos. Problemas de hardware e desastres naturais derrubam salas de máquinas inteiras.

Se executados corretamente, os backups (também conhecidos como "dumps") permitem a um administrador restaurar um sistema de arquivos (ou qualquer parte de um sistema de arquivos) à condição em que ele se encontrava no momento do último backup. Os backups têm de ser realizados cuidadosamente e em um cronograma rígido. O sistema de backup e a mídia de backup também precisam ser testados regularmente para verificar se estão funcionando corretamente.

A integridade de fitas de backup afeta diretamente o objetivo de uma empresa. A gerência sênior precisa entender o que os backups devem realmente fazer, e não o que eles *querem* que os backups façam. É aceitável perder um dia de trabalho no Departamento de Computação de uma universidade, porém, provavelmente, não é aceitável em uma empresa de comercialização de commodities.

Começamos este capítulo com alguma filosofia geral sobre backups, seguida de uma discussão dos dispositivos de backup e mídia mais comumente utilizados (seus pontos fortes, fraquezas e custos). Em seguida, discutimos os comandos de backup e arquivamento do Linux e sugerimos quais comandos são melhores para cada situação. A seguir falamos sobre como desenvolver um esquema de backups e revisamos a mecânica dos populares utilitários **dump** e **restore**. Finalmente, examinamos o Amanda, um pacote de backup de rede gratuito e depois apresentamos alguns comentários sobre alternativas comerciais.

9.1 Boas práticas

Antes de partirmos para os fundamentos dos backups, queremos passar algumas dicas genéricas que foram aprendidas ao longo do tempo (normalmente, da maneira mais difícil). Nenhuma destas sugestões é uma regra absoluta, mas você perceberá que quanto mais delas utilizar, mais estável será o seu processo de dump.

Execute todos os dumps a partir de uma máquina

Muitos utilitários de backup permitem realizar dumps pela rede. Embora haja alguma perda em termos de desempenho ao fazer dumps dessa maneira, o aumento na facilidade da administração vale a pena. Descobrimos que o melhor método é executar um script em uma localidade central que execute **rdump** (por meio de **rsh** ou **ssh**) em cada máquina em que o dump precisa ser feito; você pode utilizar um pacote de software (pago, gratuito ou shareware) para automatizar esse processo. Todos os dumps devem ir para o mesmo dispositivo de backup (sem rebobinar, obviamente).

Se seu sistema for muito grande para ter o backup feito por um único dispositivo, você ainda deve tentar manter o seu sistema de backup o mais centralizado possível. A centralização facilita a administração e permite verificar a integridade do backup. Dependendo da mídia de backup que estiver utilizando, você poderá colocar mais de um dispositivo em um servidor sem afetar o desempenho.

Rotule sua mídia

Rotule cada parte da mídia de backup clara e completamente — uma fita não rotulada é uma fita de rascunho. Rotule diretamente cada parte da mídia unicamente para identificar seu conteúdo. Nos casos, grave informações detalhadas como listas de sistemas de arquivos e datas de dump.

Você tem de ser capaz de restaurar os sistemas de arquivos-raiz e **/usr** cruciais sem examinar os scripts de backup. Rotule a mídia de modo correspondente, listando os sistemas de arquivos nela contidos, o formato dos backups, a sintaxe

exata dos comandos utilizados para criá-los e quaisquer outras informações necessárias para restaurar o sistema sem se referir à documentação on-line.

Há uma abundância de programas de rotulação gratuitos e comerciais. Poupe-se de uma grande dor de cabeça, investindo em um. Se você compra etiquetas para sua impressora laser, o fornecedor de rótulos normalmente pode lhe fornecer software (Windows) que gere rótulos. Melhor ainda, compre uma impressora para Windows dedicada à impressão de etiquetas. Elas são baratas e funcionam bem.

Selecione um intervalo de backup razoável

Quanto maior a freqüência dos backups, menor a quantidade de dados perdida em uma falha. Entretanto, os backups utilizam recursos de sistema e tempo de operador. Você deve fornecer integridade de dados adequada a um custo razoável de tempo e de material.

Em sistemas muito utilizados, geralmente é apropriado fazer backups de sistemas de arquivos com diretórios de usuários todo dia de trabalho. Em sistemas que são utilizados menos intensamente ou nos quais os dados são menos voláteis, talvez você decida que fazer backups várias vezes por semana é suficiente. Em um sistema pequeno com apenas um usuário, fazer backups uma vez por semana talvez seja adequado. Qual a quantidade de dados que os seus usuários estão propensos a perder?

Selecione sistemas de arquivos cuidadosamente

Os sistemas de arquivos que são raramente modificados não precisam ter o backup feito com tanta freqüência como os diretórios iniciais dos usuários. Se somente alguns poucos arquivos mudarem em um sistema de arquivos de outra maneira estático (como **/etc/passwd** no sistema de arquivos-raiz), você pode copiar esses arquivos diariamente para uma outra partição cujo backup é feito regularmente.

Se **/tmp** for um sistema de arquivos separado, seu backup não deve ser feito. O diretório **/tmp** não deve conter nada essencial, de forma que não há nenhuma razão para preservá-lo. No caso de isso parecer óbvio, conhecemos uma grande instalação que realiza backups diários de **/tmp**.

Faça dumps diários se ajustarem em uma parte da mídia

Consulte o Capítulo 8 para informações adicionais sobre ***cron****.*

Em um mundo perfeito poderíamos fazer dumps diários de todos os sistemas de arquivos de usuário em uma única fita. Mídias de alta densidade como DLT, AIT e LTO tornam esse objetivo prático para algumas instalações. Você pode automatizar o processo montando sua mídia de backup diariamente antes de deixar o trabalho e de permitir que **cron** execute o backup para você. Dessa maneira, os dumps ocorrem em um horário em que os arquivos provavelmente não ficarão mudando e os backups têm impacto mínimo sobre os usuários.

À medida que nossos hábitos de trabalho mudam e a telecomutação torna-se mais popular, o intervalo das "boas" horas para fazer backups está diminuindo. A julgar por nossos logs, vemos que entre 3h00 e 6h00 da manhã é a melhor hora. Entretanto, fazer backup de 80 TB de dados em três horas não vai acontecer.

Outro problema importante é a rápida expansão de espaço em disco que resultou das reduções no preço de discos rígidos. Você não pode mais comprar uma máquina desktop no mercado com menos de 100 GB de espaço em disco. Por que limpar discos e impor cotas quando você pode simplesmente investir um pouco mais no problema e adicionar mais espaço em disco? Infelizmente, é muito fácil a quantidade de arquivos disponibilizados on-line superar sua capacidade de backup.

Se não for possível colocar todos os seus backups diários em uma fita, você tem várias opções:

- Comprar um dispositivo de backup de maior capacidade
- Comprar um empilhador ou biblioteca e alimentar várias mídias num único dispositivo
- Mudar a seqüência de dump
- Criar um script mais inteligente
- Usar vários dispositivos de backup

Um sistema de dump automatizado deve sempre registrar o nome de cada sistema de arquivos do qual foi feito o dump. O registro cuidadoso sempre permite pular diretamente para o sistema de arquivos correto quando queremos restaurar um arquivo. Também é uma boa idéia registrar a ordem dos sistemas de arquivos no lado de fora da fita. (Dissemos isso antes e repetimos: não deixe de utilizar o dispositivo não rebobinável para gravar na mídia com vários dumps.)

Torne os sistemas de arquivos menores do que o dispositivo de dump

dump e outros utilitários prontos para serem utilizados são perfeitamente capazes de transferir sistemas de arquivos para várias partes da mídia. Entretanto, se um dump se distribui por várias fitas, o operador ou robô de biblioteca de fita deve estar presente para alterar a mídia e esta deve ser cuidadosamente rotulada para permitir que restaurações sejam realizadas facilmente. A menos que você tenha uma boa razão para criar um sistema de arquivos realmente grande, não faça isso.

Mantenha a mídia fora do local de trabalho

A maioria das empresas mantém backups fora do local de trabalho onde o sistema está instalado para que na ocorrência de algum incidente como um incêndio, por exemplo, não sejam destruídos tanto os arquivos originais quanto os backups. "Fora do local de trabalho" (*off-site*) pode ser qualquer coisa, desde um cofre em um banco até a casa do presidente da empresa. As empresas que se especializam em armazenamento seguro de mídias de backup garantem um ambiente seguro e climatizado para os seus arquivos. Certifique-se sempre de que esse tipo de fornecedor de serviços tenha boa reputação, tenha hipoteca e seguro. Há, atualmente, negócios on-line (mas fora do local de trabalho) que são especializados em salvaguardar dados.

A velocidade com que as mídias de backup são transferidas para fora do local de trabalho depende da freqüência com que você precisa restaurar arquivos e da quantidade de latência que pode aceitar. Alguns sites evitam tomar essa decisão realizando dois dumps para diferentes dispositivos de backup, um que permanece no local e outro que é imediatamente movido.[1]

Proteja seus backups

Dan Geer, um consultor em segurança, disse: "O que um backup faz? Ele viola de maneira confiável permissões de arquivos à distância". Hmmm.

Proteja suas mídias de backup. Elas contêm todos os dados de sua empresa e podem ser lidas por qualquer um que tenha acesso físico a elas. Você não só deve manter suas fitas fora do local de trabalho como também mantê-las a sete chaves. Se você utiliza um recurso de armazenamento comercial para essa finalidade, a empresa com a qual você negocia deve garantir a confidencialidade das fitas sob seus cuidados.

A criptografia de mídia de backup é uma opção a ser levada em consideração. Muitos utilitários comerciais de backup tornam a criptografia relativamente simples. Por outro lado, você deve certificar-se de que as chaves de criptografia não podem ser perdidas ou destruídas e que estão disponíveis para utilização em uma emergência.

Algumas empresas sentem tanto a importância dos backups a ponto de fazerem duplicatas, o que realmente não deixa de ser absolutamente uma ótima idéia.

Limite a atividade durante os dumps

A atividade do sistema de arquivos deve ser limitada durante dumps, pois mudanças podem fazer com que o utilitário de backup cometa erros. Podemos limitar a atividade fazendo os dumps quando poucos usuários estiverem usando o sistema (no meio da noite ou em finais de semana) ou tornando todos os sistemas de arquivos acessíveis somente para o utilitário de backup. (Essa precaução parece boa na teoria, porém raramente é usada na prática. Os usuários querem acesso 24 horas por dia, 7 dias por semana a todos os sistemas de arquivos. Hoje em dia é impossível fazer um backup sem nenhuma atividade no disco.)

Consulte a página 345 para informações adicionais sobre servidores de arquivos.

Servidores de arquivos dedicados como aqueles fabricados pela Network Appliance fornecem backups on-line com instantâneos do sistema de arquivos em intervalos regulares e ajustáveis. Esse recurso permite backups seguros de um sistema de arquivos ativo e é uma das vantagens importantes de utilizar um servidor de arquivos dedicado.

Verifique suas mídias

Já ouvimos muitas histórias de terror sobre administradores de sistemas que não viam problemas em seus regimes de dump a não ser depois de uma falha grave no sistema. É essencial monitorar continuamente o seu procedimento de backup e verificar se ele está funcionando corretamente. Erros de operadores arruinam mais dumps do que qualquer outra coisa.

A primeira verificação é fazer com que o seu software de backup tente reler fitas imediatamente depois de ele ter finalizado o dump.[2] Varrer uma fita para verificar se ela contém o número de arquivos esperado é uma boa verificação. Seria melhor se todas as fitas fossem varridas, porém isso não parece prático para uma grande empresa que usa centenas de fitas por dia. Uma amostra aleatória seria mais prudente neste ambiente.

Consulte a página 122 para informações adicionais sobre o **restore**.

Normalmente é útil gerar uma tabela de conteúdo de cada sistema de arquivos (usuários do **dump** podem utilizar **restore -t**) e armazenar os resultados em disco. Esses catálogos devem ser nomeados de uma maneira que os relacione à fita apropriada; por exemplo, **okra:usr.Jan.13**. Uma coletânea de

---

1. Uma grande instituição financeira localizada no World Trade Center mantinha seus backups "fora do local de trabalho" um ou dois andares abaixo de seus escritórios. Quando o edifício foi atacado (da primeira vez), as fitas de backup (bem como os computadores) foram destruídas. Certifique-se de que isso seja realmente "fora do local de trabalho".
2. Você pode utilizar **restore -C** para verificar uma fita de dump em uma árvore de diretórios.

uma semana desses registros torna mais fácil descobrir em que parte da mídia um arquivo perdido se encontra. Você simplesmente utiliza **grep** para o nome do arquivo e seleciona a instância mais nova.

Além de fornecer um catálogo de fitas, ser bem-sucedido na leitura da tabela de conteúdo da fita é uma boa indicação de que o backup está OK e de que você será capaz de ler a mídia quando precisar fazê-lo. Uma tentativa rápida de restaurar um arquivo aleatório dá a você ainda mais confiança na sua capacidade de restaurar a partir dessa parte da mídia.[3]

Você deve tentar periodicamente restaurar a partir da mídia aleatória para certificar-se de que a restauração ainda é possível. De vez em quando, tente restaurar de uma parte de uma mídia de dump antiga (meses ou anos). As unidades são conhecidas por saírem de alinhamento ao longo do tempo e de se tornarem incapazes de ler suas fitas antigas. A mídia pode ser recuperada por uma empresa especializada nesse serviço, mas isso é caro.

Uma verificação relacionada é examinar se você pode ler as mídias em um hardware diferente do seu. Se a sala de máquinas pegar fogo, não adianta muito saber se o backup poderia ter sido lido em uma unidade de fita que agora está destruída. As fitas DAT foram particularmente suscetíveis a esse problema, mas as versões mais recentes da tecnologia se aprimoraram.

Desenvolva um ciclo de vida de mídia

Toda mídia tem uma vida finita. É fantástico reciclar suas mídias, mas não deixe de seguir às recomendações do fabricante referentes à vida útil das mídias. A maioria dos fabricantes quantifica essa vida em termos do número de passagens que uma fita pode resistir: um backup, uma restauração e um **mt fsf**, (file skip forward, pula para o primeiro bloco do próximo arquivo), cada uma dessas operações representa uma passagem. As tecnologias que não usam fita têm uma vida muito mais longa do que aquela que às vezes é expressa no tempo médio para falha (Mean-Time-To-Failure — MTTF).

Antes de jogar suas velhas fitas no lixo, lembre-se de apagá-las ou torná-las ilegíveis. Um apagador de fita de volume (um grande eletroímã) pode ajudar com isso, mas não deixe de mantê-lo distante, longe de computadores e das mídias ativas. Cortar ou extrair parte de uma fita de backup realmente não é muito para proteger seus dados, porque a fita é fácil de emendar ou rebobinar. Empresas de destruição de documentos destroem as fitas mediante o pagamento de uma taxa.

Projete seus dados para backups

Com os discos tão baratos e novas arquiteturas de armazenamento tão confiáveis, é tentador não fazer backup de todos os nossos dados. Uma arquitetura de armazenamento sensata — projetada em vez de criada desordenadamente à medida que surge a necessidade de aumento de espaço em disco — pode colaborar muito para tornar os backups mais tratáveis. Comece fazendo um inventário de suas necessidades de armazenamento:

- Os diversos tipos de dados com os quais sua instalação lida
- A volatilidade esperada de cada tipo de dado
- A freqüência necessária para atender a um nível de perda de dados esperado
- Os limites políticos sobre quais dados são distribuídos

Use essas informações para projetar a arquitetura de armazenamento de sua instalação com backups e tendo crescimento potencial em mente. Preservar diretórios de projetos e diretórios iniciais de usuários em um servidor de arquivos dedicado pode facilitar o gerenciamento de dados e garantir sua segurança.

Prepare-se para o pior

Depois de ter estabelecido um procedimento de backup, explore a situação do pior caso: sua instalação foi completamente destruída. Determine o nível de dados que serão perdidos e quanto tempo levaria para ter o sistema no ar novamente (incluindo o tempo que levaria para adquirir hardware novo). A seguir, determine se você pode conviver com suas respostas.

9.2 DISPOSITIVOS E MÍDIA PARA BACKUP

Muitas falhas podem danificar várias partes do hardware de uma vez, então os backups devem ser gravados em algum tipo de mídia removível. Fazer backup de um disco rígido para outro na mesma máquina fornece pouca proteção contra uma falha de controladora, embora seja certamente melhor do que nenhum backup. Empresas que fazem backup de seus

3. Por exemplo, **restore -t** lê somente o índice analítico para o dump, que é armazenado no começo da fita. Quando você realmente restaura um arquivo, você está testando uma região mais extensa da mídia.

dados pela Internet estão se tornando mais populares entre organizações que encriptam seus dados, mas a maioria dos backups ainda é armazenada localmente.

Muitos tipos de mídia armazenam dados utilizando partículas magnéticas. Essas mídias estão sujeitas a danos por campos elétricos e magnéticos. Você deve ter cuidado com as seguintes fontes de campos magnéticos: alto-falantes, transformadores e fontes de energia, mecanismos elétricos não blindados, ventiladores de disco, monitores CRT e até a prolongada exposição à radiação de fundo da Terra. Todas as fitas magnéticas se tornarão ilegíveis em um período de anos. A maior parte das mídias é capaz de manter os dados por três anos, porém, se você pretende armazenar dados por mais tempo, deveria utilizar mídia óptica ou regravar os dados.

As seções a seguir descrevem algumas das mídias que podem ser utilizadas para backups. As mídias são apresentadas mais ou menos na ordem crescente de capacidade.

Os fabricantes gostam de especificar as capacidades do hardware em termos de dados compactados: eles normalmente supõem de forma otimista uma razão de compactação de 2:1 ou mais. Nas seções a seguir, ignoramos a compactação em favor do número real de bytes que pode ser fisicamente armazenado em cada mídia.

A relação de compactação também afeta a taxa de throughput de uma unidade. Se uma unidade é capaz de gravar 1 MB/s em fita, mas o fabricante supõe uma compactação de 2:1, o throughput sobe, num passe de mágica, para 2 MB/s. Como acontece com os números de capacidade, também ignoramos o aumento de throughput.

Embora custo e capacidade de mídias sejam considerações importantes, também é importante considerar o throughput. Mídias rápidas são mais agradáveis de lidar e oferecem uma maior flexibilidade no agendamento de backups.

Na edição anterior deste livro mencionamos os disquetes, tanto tradicionais como os com supercapacidade (por exemplo, discos Zip), como potencial mídia de backup. Essas mídias tiveram seu auge, mas como você dificilmente compra um laptop com uma unidade de disquete, agora excluímos esses itens de nossa lista. Embora disquetes sejam úteis para trocar dados, seu alto custo de mídia os torna uma escolha pobre para backups.

Mídia óptica: CD-R/RW, DVD±R/RW e DVD-RAM

Em um custo de aproximadamente US$ 0,40 cada, CDs e DVDs são uma atraente opção para backups de sistemas pequenos, isolados. Os CDs armazenam aproximadamente 650 MB e os DVDs, 4,7 GB. Os DVDs de camada dual, que estão se tornando os DVDs mais representativos da atualidade, começam a aproximadamente 8,5 GB.

As unidades que gravam essas mídias estão disponíveis para todos os barramentos comuns (SCSI, IDE, USB etc.) e em muitos casos são tão baratas quanto como essencialmente gratuitas. A mídia de CD também costumava ser gratuita, mas a fartura parece ter acabado. Agora que os preços de CDs e DVDs se equilibraram, não há razão para utilizar CDs em vez de DVDs, exceto que os leitores de CD ainda são, em geral, os mais comuns.

A mídia óptica é escrita por um processo fotoquímico que envolve a uso de um laser. Embora dados concretos sobre longevidade tenham sido evasivos, acredita-se amplamente que as mídias ópticas têm uma vida útil de armazenamento mais longa do que as mídias magnéticas. Entretanto, mesmo as versões de uma gravação (CD-R, DVD-R e DVD+R) não são tão duráveis quanto os CDs e DVDs manufaturados (selados).

O CD-R não é particularmente uma boa opção para backups normais, mas é bom para arquivamento de dados que talvez você possa querer recuperar daqui a um bom tempo. O CD-RW funciona bem para arquivar regularmente pequenas quantidades de dados pessoais.

Os rápidos escritores de DVD atuais oferecem velocidades tão altas quanto às das unidades de fita, se não mais altas do que elas. As versões de uma gravação são DVD-R (desenvolvidas pela Pioneer em 1997) e DVD+R (desenvolvidas por uma coalizão em 2002). DVD-RW, DVD+RW e DVD-RAM são regraváveis. O sistema DVD-RAM tem gerenciamento de defeito predefinido e, conseqüentemente, é mais confiável do que outras mídias ópticas. Por outro lado, é muito mais cara.

Os fabricantes estimam uma faixa de vida potencial de centenas de anos para essas mídias se elas forem armazenadas adequadamente. Suas recomendações para armazenamento adequado incluem casos individuais, armazenamento a uma temperatura constante na faixa de 41°–68° F com umidade relativa de 30%–50%, sem exposição direta à luz solar e marcação somente com marcadores solúveis à água. Sob condições médias, uma vida confiável de prateleira de 1–5 anos talvez seja mais realista.

Como suportada por numerosas avaliações independentes, a confiabilidade da mídia óptica revelou-se excepcionalmente dependente de fabricante. Esse é um caso em que se paga por investir dinheiro em mídia de qualidade premiada. Infelizmente, a qualidade varia de produto para produto mesmo dentro da linha de um fabricante, portanto, não há nenhuma aposta segura de fabricante. Nos últimos anos Taiyo Yuden dominou as estáticas de confiabilidade, mas apenas com mídia manufaturada no Japão; a mídia Taiyo Yuden fabricada em outros países tem um registro menos consistente.

Uma recente entrada no mercado de armazenamento óptico de dados é o disco Blu-Ray, cujas várias versões armazenam a partir de 25–100 GB de dados. A alta capacidade é um resultado do comprimento de ondas curtas (405 nm) do laser utilizado para ler e gravar os discos (daí o "blue" de Blu-Ray). Os primeiros players foram disponibilizados nos Estados Unidos em junho de 2006. Essa é uma tecnologia digna de nota e promete tornar-se uma boa solução para backups.

Discos rígidos removíveis (USB e FireWire)

Os dispositivos de armazenamento externo que se conectam por um USB ou por portas FireWire (IEEE1394) tornaram-se comuns. A tecnologia de armazenamento subjacente é em geral alguma forma de disco rígido, mas os dispositivos de memória flash são comuns em equipamentos menos sofisticados (*low end*) (as onipresentes "unidades de jump"). As capacidades variam: menos de 1 MB a 600 GB e acima. O limite em dispositivos de memória flash é de aproximadamente 4 GB hoje em dia.

A principal limitação dessas unidades é a velocidade do barramento, mas com o lançamento de USB 2.0 e FireWire 800, ambas as versões alcançaram um respeitável throughput na faixa de 50–90 MB/s.

O tempo de vida de dispositivos de memória flash é principalmente uma função do número de ciclos de gravação. As unidades de médio porte permitem supostamente vários milhões de ciclos.

Unidades de fita pequenas: 8 mm e DDS/DAT

Várias versões de unidades de fita de 8 mm e DDS/DAT compõem a linha *low-end* do mercado de armazenamento de fita. As unidades de fita de 8 mm Exabyte foram as primeiras favoritas, mas as unidades apresentaram uma tendência ao mau alinhamento a cada 6–12 meses, exigindo uma ida à assistência técnica. Não era incomum o fato de as fitas serem esticadas no mecanismo de transporte, tornando-se não confiáveis. A capacidade de 2–7 GB dessas fitas as torna ineficientes para fazer backup de sistemas de desktop atuais, muito menos de servidores.

As unidades DDS/DAT (Digital Audio Tape) são dispositivos de varreduras helicoidais que utilizam cartuchos de 4 mm. Embora essas unidades sejam normalmente conhecidas como unidades DAT, elas são na verdade unidades DDS (Digital Data Storage); a distinção exata não é importante. O formato original armazenava cerca de 2 GB, mas gerações sucessivas aumentaram significativamente a capacidade das DDSs. A geração atual (DAT 72) armazena até 32 GB de dados a uma taxa de transferência de 3,5 MB/s. As fitas devem durar por 100 backups e, segundo informações, têm uma vida de prateleira de 10 anos.

DLT/S-DLT

A Digital Linear Tape/Super Digital Linear Tape é uma das mídias de backup mais representativas da atualidade. Essas unidades são confiáveis, de preço razoável e alta capacidade. Elas foram desenvolvidas a partir das unidades de fita de cartucho TK-50 e TK-70 da DEC. A DEC vendeu a tecnologia para a Quantum, que popularizou as unidades aumentando sua velocidade e capacidade e abaixando seu preço. Em 2002, a Quantum adquiriu a DLT Super, uma tecnologia da Benchmark Storage Innovations que inclina o cabeçote de registro de um lado para outro para reduzir a conversação cruzada entre trilhas adjacentes.

A Quantum oferece agora duas linhas de hardware: uma linha de desempenho e outra de valor. O barato sai caro. As capacidades de fita variam de DLT-4 em 800 GB a DLT-4 na linha de valor em 160 GB, com taxas de transferência de 60 MB/s e 10 MB/s, respectivamente. Os fabricantes se vangloriam de que as fitas durarão de 20 a 30 anos — isto é, se o hardware para lê-las ainda existir. Quantas unidades de fita de 9 trilhas ainda estão funcionando e on-line atualmente?

A desvantagem de S-DLT é o preço de mídia, que custa por volta de US$ 30 a US$ 45 por fita. Um pouco caras para uma universidade; talvez não para uma empresa de investimento de Wall Street.

AIT e SAIT

Advanced Intelligent Tape é um produto de 8 mm proprietário da Sony. Em 1996, a Sony rompeu o seu relacionamento com a Exabyte e introduziu o AIT-1, um dispositivo de varredura helicoidal de 8 mm com o dobro da capacidade das unidades de 8 mm da Exabyte. Hoje, a Sony tem uma versão de AIT-1 de maior capacidade e sua tecnologia de investimento subseqüente AIT-4, que tem capacidade de 200 GB e uma taxa de transferência máxima de 24 MB/s.

O SAIT é a oferta de meia altura da Sony que utiliza mídia maior e tem maior capacidade do que o AIT. As fitas SAIT armazenam até 500 GB de dados e ostentam uma taxa de transferência de 30 MB/s. Esse produto é mais popular na forma de ofertas de biblioteca de fitas.

As fitas AME (Advanced Metal Evaporated) utilizadas em unidades AIT e SAIT possuem uma vida útil longa. Elas também contêm uma EEPROM embutida que confere certa inteligência à própria fita. Entretanto é necessário suporte de software para fazer uso efetivo da EEPROM. Os preços das unidades e das fitas equivalem aproximadamente aos da DLT.

VXA/VXA-X

Ofertas atuais da Exabyte incluem as tecnologias VXA e VXA-X. As unidades VXA utilizam o que a Exabyte descreve como uma tecnologia de pacote para transferência de dados. Os produtos VXA-X ainda contam com a Sony para as mídias AME; a série V é atualizável à medida que a mídia de maior capacidade torna-se disponível. As séries VXA e X reivindicam capacidades na faixa de 33–160 GB, com uma taxa de transferência de 24 MB/s.

LTO

A Linear Tape-Open foi desenvolvida pela IBM, HP e Quantum como uma alternativa ao formato proprietário (patenteado) de DLT. O LTO-3, a última versão, tem uma capacidade de 400 GB a uma velocidade de 80 MB/s. A mídia LTO tem uma vida estimada de armazenamento de 30 anos, mas é suscetível à exposição magnética. O custo da mídia é de aproximadamente US$ 80 para as fitas de 400 GB.

Jukeboxes, empilhadores e bibliotecas de fitas

Com o baixo custo dos discos atualmente, a maioria das instalações tem tanto espaço em disco que um backup completo exigiria muitas fitas, até mesmo fitas de 100 GB de capacidade. Uma possível solução para essas instalações é um empilhador, uma jukebox ou uma biblioteca de fitas.

Um empilhador é um trocador de fitas simples utilizado com uma unidade de fita padrão. Ele possui um depósito alimentador no qual carregamos fitas; ele descarrega fitas completas à medida que são ejetadas da unidade e as substitui por fitas em branco alojadas no depósito alimentador. A maioria dos empilhadores tem capacidade para até dez fitas.

Uma jukebox é um dispositivo de hardware capaz de trocar automaticamente mídia removível em um número limitado de unidades, de maneira muito parecida com aqueles trocadores de disco de vinil antigos. As jukeboxes estão disponíveis por todas as mídias discutidas aqui. As jukeboxes normalmente vêm com software para backup especial que sabe como lidar com o trocador. A Storage Technology (agora a Sun Microsystems) e a Sony são dois grandes fabricantes desses produtos.

As bibliotecas de fitas são uma solução de backup para grandes conjuntos de dados — terabytes, normalmente. Essas são mecanismos do tamanho de um armário com várias unidades de fita (ou CDs) e um braço mecânico que corrige e coloca mídia nas diversas prateleiras de sua biblioteca. Como é possível imaginar, elas são bem caras em termos de aquisição e de manutenção e exigem alimentação, espaço e condições de climatização especiais. A maioria dos compradores de bibliotecas de fitas também adquire um contrato de operações do fabricante para otimizar e comandar o equipamento. Obviamente, as bibliotecas possuem um componente de software que é o que realmente comanda o dispositivo. A Storage Technology (Sun Microsystems) é um importante fabricante de bibliotecas de fita.

Discos rígidos

A queda no preço das unidades de discos rígidos torna os backups de disco para disco uma opção atraente a ser considerada. Embora tenhamos sugerido que você não duplique um disco em outro dentro da mesma máquina física, os discos rígidos podem ser uma boa solução, de custo baixo para backups em uma rede.

Um problema óbvio é que o espaço de armazenamento em disco rígido é finito e, no final, deve ser reutilizado. Entretanto, os backups de disco para disco são uma excelente maneira de se proteger contra a exclusão acidental de arquivos. Se você mantiver uma imagem de disco defasada de um dia em um local bem conhecido que é compartilhado por meio de NFS ou CIFS, os usuários poderão se recuperar de seus próprios erros sem ter de envolver o administrador do sistema. Outras opções populares são os atuais discos FireWire e USB de alto volume para fazer backup de desktops individuais. O Cintre ZDisc 1-Button Instant Backup é uma unidade de disco USB 2.0 externa de 3.5", uma opção atraente que custa em torno de US$ 100.

Resumo dos tipos de mídia

Uau! Há realmente muitas possibilidades. A Tabela 9.1 sintetiza as características discutidas nas seções anteriores.

W. Curtis Preston compilou uma excelente lista de referência de dispositivos de backup classificados por fabricante. Ela se encontra disponível em www.backupcentral.com/hardware-drives.html.

O que comprar

Ao comprar um sistema de backup, você provavelmente obterá exatamente o que vimos na Tabela 9.1. Todas as mídias funcionam razoavelmente bem e, entre as tecnologias próximas em preço, geralmente não há uma forte razão para optar por uma ou por outra. Compre um sistema que atenda às suas especificações e caiba no seu orçamento.

As unidades DDS, AIT e LTO são excelentes soluções para pequenos grupos de trabalhos e máquinas individuais com grande capacidade de armazenamento. Os custos iniciais são relativamente modestos, as mídias são encontradas em abundância e vários

fabricantes estão utilizando os padrões de cada um deles. Todos esses sistemas são suficientemente rápidos para fazer o backup de muitos dados em uma quantidade finita de tempo.

DLT, AIT e LTO-3 são grosseiramente comparáveis. Não há um ganhador claro entre os três e, mesmo se houvesse, a situação sem dúvida nenhuma mudaria em poucos meses à medida que novas versões dos formatos fossem implantadas. Todos esses formatos são bem estabelecidos e teriam uma fácil integração no seu ambiente, seja uma universidade ou empresa.

Nas seções seguintes, utilizamos o termo genérico "fita" para nos referirmos à mídia escolhida para backups. Exemplos dos comandos de backup são feitos em termos de dispositivos de fitas.

Tabela 9.1 Comparação entre as mídias de backup

Mídia	Capacidade[a]	Velocidade[a]	Unidade	Mídia	Custo/GB	Reuso?	Aleatória?[b]
CD-R	650 MB	4 MB/s	15	0,20	0,32	Não	Sim
CD-RW	650 MB	4 MB/s	20	0,30	0,48	Sim	Sim
DVD±R	4,7 GB	10 MB/s	50	0,40	0,09	Não	Sim
DVD+R DL[c]	8,5 GB	10 MB/s	50	2,00	0,24	Não	Sim
DVD±RW	4,7 GB	10 MB/s	50	0,80	0,17	Sim	Sim
DDS-4 (4 mm)	20 GB	10 MB/s	300	5,00	0,25	Sim	Não
DLT/S-DLT	160 GB	10 MB/s	1.000	20,00	0,13	Sim	Não
AIT-4 (8 mm)	200 GB	6 MB/s	2.500	40,00	0,20	Sim	Não
SAIT-1	500 GB	30 MB/s	6.500	126,00	0,25	Sim	Não
VXA-172	86 GB	12 MB/s	600	30,00	0,35	Sim	Não
VXA-320	160 GB	12 MB/s	800	60,00	0,35	Sim	Não
LTO-3	400 GB	30 MB/s	3.000	65,00	0,16	Sim	Não

a. Capacidade e velocidade de descompactação.
b. Permite acesso aleatório a qualquer parte da mídia?
c. Camada dual.

9.3 CONFIGURANDO UM REGIME DE BACKUP INCREMENTAL VIA DUMP

Os comandos **dump** e **restore** são a maneira mais comum de criarmos e restaurarmos informações de backup. Esses programas existem há um longo tempo e seus comportamentos são bem conhecidos. Na maioria das instalações, **dump** e **restore** são comandos subjacentes utilizados por software de backup automatizado.

Talvez seja necessário instalar explicitamente **dump** e **restore** em seus sistemas Linux, dependendo das opções selecionadas durante a instalação original. Disponibilizamos um pacote para instalação fácil em todos os nossos exemplos de sistemas. As versões Red Hat e Fedora Core atuais oferecem um pacote de administração de sistema em tempo de instalação que inclui **dump**.

Sob o Linux, nada está estaticamente linkado, então você precisa das bibliotecas compartilhadas em **/lib** para que tudo entre em funcionamento. A linkagem estática facilita a recuperação após um desastre, pois **restore** é então completamente autocontido.

Fazendo o dump de sistemas de arquivos

dump cria uma lista de arquivos que foram modificados desde um dump anterior, depois empacota esses arquivos em um único grande arquivo para arquivamento em um dispositivo externo. **dump** apresenta várias vantagens em relação à maioria dos demais utilitários descritos neste capítulo:

- Backups podem abranger várias fitas
- Arquivos de qualquer tipo (até mesmo dispositivos) podem ser colocados em backup e restaurados
- Permissões, propriedades e horários de modificação são preservados
- Arquivos com buracos são tratados corretamentamente[4]
- Backups podem ser realizados incrementalmente (apenas com os arquivos recentemente modificados gravados na fita)

4. Brechas são blocos que nunca contiveram dados. Se você abrir um arquivo, gravar um byte, pesquisar 1 MB no arquivo, depois gravar outro byte, o arquivo "esparso" resultante ocupará apenas dois blocos de disco, muito embora seu tamanho lógico seja bem maior. Arquivos criados pelos utilitários Berkeley DB ou **ndbm** contêm muitos brechas.

A versão GNU do **tar** utilizada no Linux também fornece todos esses recursos. Entretanto, a manipulação por parte de **dump** de backups incrementais (discutidos posteriormente) é um pouco mais sofisticada do que a do **tar**. Talvez você ache esse poder adicional útil caso suas necessidades sejam complexas.

Na realidade, a razão mais forte para optarmos por **dump** em relação a **tar** em um ambiente Linux não tem nada a ver com o Linux em si. Infelizmente, a versão de **tar** que vem com as distribuições UNIX mais importantes não tem muitos dos recursos do GNU **tar**. Se você tiver de suportar backups tanto para Linux como para UNIX, **dump** é a melhor opção. É o único comando que trata dessas questões de forma (relativamente) consistente entre as plataformas, permitindo que você seja um expert em um comando em vez de familiarizado com os dois. Se tiver sorte o bastante para estar em um ambiente Linux completamente homogêneo, escolha o seu favorito. **dump** tem menos recheio, mas **tar** tem um excelente sabor!

O comando **dump** compreende o layout de sistemas de arquivos brutos e lê a tabela de inodes de um sistema de arquivos diretamente para decidir quais arquivos devem ser colocados em backup. Esse conhecimento do sistema de arquivos torna **dump** muito eficiente, mas também impõe algumas limitações.[5]

Consulte o Capítulo 16 para informações adicionais sobre o NFS.

A primeira limitação é que todo sistema de arquivos deve ter o seu dump feito individualmente. Se tivermos um disco particionado, temos de fazer o dump de cada partição separadamente. A outra limitação é que somente sistemas de arquivos na máquina local podem ter o dump feito; você não pode fazer o dump de um sistema de arquivos NFS montado de uma máquina remota. Entretanto, você pode fazer o dump de um sistema de arquivos local para uma unidade de fita remota com o irmão gêmeo de **dump**, o **rdump**.

O recurso mais importante de **dump** é o seu suporte ao conceito de backup "incremental". Embora fosse possível fazer backup do sistema inteiro diariamente, em geral, isso não é prático. Com os dumps incrementais, você só pode fazer backup dos arquivos que foram alterados desde o último backup.

Ao fazer um dump, você atribui a ele um nível de backup, que é um inteiro de 0 a 9. Um dump nível N faz o backup de todos os arquivos que foram modificados desde o último dump de nível menor do que N. Um backup nível 0 coloca todo o sistema de arquivos na fita. Com um sistema de backup incremental, você pode ter de restaurar arquivos de vários conjuntos de fitas de backup para recolocar um sistema de arquivos no estado em que se encontrava durante o último backup.[6]

Outro recurso interessante de **dump** é que ele não se importa com o comprimento de nomes de arquivos. As hierarquias podem ser arbitrariamente profundas, e nomes longos são manipulados corretamente.

O primeiro argumento para **dump** tem de ser o nível de dump incremental. **dump** usa o arquivo **/etc/dumpdates** para determinar até que nível se deve voltar em um dump incremental. O flag **-u** faz com que **dump** atualize automaticamente **/etc/dumpdates** quando o dump é completado. São gravados a data, o nível de dump e o nome de sistema de arquivos. Se você nunca usa o flag **-u**, todos os dumps se tornam nível 0, pois não há nenhum registro da criação de dump do sistema de arquivos. Se modificar o nome de um sistema de arquivos, você poderá editar manualmente o arquivo **/etc/dumpdates**.

Consulte a página 597 para obter informações sobre os números de dispositivo.

dump envia sua saída para algum dispositivo padrão, normalmente a unidade de fita primária. Para especificar um dispositivo diferente, utilize o flag **-f**. Se estiver colocando vários dumps em uma única fita, certifique-se de que especifica um dispositivo de fita não rebobinável (um arquivo de dispositivo que faz com que a fita não seja rebobinada quando for fechada — a maioria das unidades de fita tem tanto uma entrada de dispositivo padrão como uma não rebobinável[7]). Para a primeira unidade de fita SCSI do sistema, o Linux utiliza **/dev/st0** para o dispositivo rebobinável e **/dev/nst0** para o dispositivo não rebobinável.

Se, acidentalmente, optar pelo dispositivo rebobinável, você acabará salvando apenas o último sistema de arquivos cujo dump foi feito. Já que **dump** não possui a mínima idéia de onde a fita se encontra posicionada, esse engano não provoca erros. A situação só se torna aparente quando você tenta restaurar arquivos.

Ao utilizar **rdump** para fazer o dump para um sistema remoto, você especifica a identidade da unidade de fita remota como *hostname:device*; por exemplo,

```
# rdump -0u -f anchor:/dev/nst0 /spare
```

A permissão para acessar unidades de fita remotas deve ser controlada pelo SSH. Consulte a página 481 para obter informações adicionais.

No passado tínhamos de informar ao **dump** o comprimento exato das fitas para que pudesse parar a gravação antes de atingir o final da fita. Unidades de fitas modernas são capazes de informar quando elas atingem o final de uma fita e podem relatar esse fato de volta para **dump**, que, a seguir, rebobina e ejeta a fita atual e solicita uma nova. Como a variabilidade de

---

5 **dump** exige acesso a partições de disco bruto. Qualquer pessoa com permissão de fazer dumps pode ler todos os arquivos no sistema com um esforço mínimo.

6 Na verdade, a maioria das versões de **dump** não monitora arquivos que foram excluídos. Se você restaurar a partir de backups incrementais, os arquivos excluídos serão recriados.

7 Todas as entradas de uma unidade de fita usam o mesmo número de dispositivo principal. O número de dispositivo secundário informa o driver de dispositivo de fita sobre comportamentos especiais (rebobinagem, swap de bytes etc.).

compactação de hardware torna o "comprimento virtual" de cada fita um pouco indeterminado, é sempre melhor contar com a indicação do final de fita caso o nosso hardware o suporte. Caso contrário, você pode especificar o comprimento da fita em kilobytes com a opção **-B**.

```
# dump -5u -B 2000000 -f /dev/nst0 /work
DUMP: Date of this level 5 dump: Wed May  8 16:59:45 2006
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/hda2 (/work) to /dev/rst0nst0
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 18750003 tape blocks on .23 tape(s)
....
```

Os flags **-5u** são seguidos pelos flags **-B** (comprimento da fita: 20 GB) e **-f** (dispositivo de fita: **/dev/nst0**). O último argumento, que é obrigatório, é o nome do sistema de arquivos a ser colocado em dump (**/work**).

Seqüências de dump

Como os níveis de dump só têm significado em relação a outros níveis, os dumps podem ser realizados de acordo com várias agendas.

A agenda adequada para o seu caso depende da:

- Atividade de seus sistemas de arquivos
- Capacidade de seu dispositivo de dump
- Quantidade de redundância desejada
- Número de fitas que deseja adquirir

Na época em que se usava muitas fitas para se fazer o backup de um sistema de arquivos, seqüências de dump complicadas eram úteis para minimizar o número de fitas consumidas pelo backup diário. À medida que a capacidade das fitas foi aumentando, tornou-se menos útil fazer distinções entre níveis de dump.

Pelo fato de a maioria dos arquivos nunca mudar, até mesmo a programação incremental mais simples elimina muitos arquivos dos dumps diários. À medida que adiciona níveis à programação de dump, você divide os relativamente poucos arquivos em segmentos cada vez menores.

Uma agenda de dumps complexa gera os seguintes benefícios:

- Você pode fazer backup de dados com maior freqüência, limitando o potencial de perdas
- Você pode utilizar menos fitas diárias (ou fazer caber tudo em uma única fita)
- Você pode manter várias cópias de cada fita, para proteção contra erros nas fitas

Em geral, a maneira de selecionar a seqüência serve para determinar suas necessidades em cada uma dessas áreas. Dadas essas restrições, você pode elaborar um cronograma no nível apropriado de sofisticação. Descrevemos uma série de seqüências possíveis e a motivação por trás delas. Uma delas deve ser adequada ao seu caso — caso contrário, suas necessidades talvez ditem uma agenda completamente diferente.

Uma agenda simples

Se a quantidade total de espaço em disco for menor do que a capacidade de seu dispositivo de fita, você poderá utilizar uma agenda de dump completamente trivial. Faça dumps nível 0 de todos os sistemas de arquivos todo dia. Reutilize um grupo de fitas, mas, a cada N dias (onde N é determinado pelas necessidades de sua instalação), mantenha a fita para sempre. Essa agenda custa:

(365/N) * (preço da fita)

por ano. Não reutilize exatamente a mesma fita para o dump de cada noite. É melhor fazer o rodízio entre um conjunto de fitas para que, mesmo que um dump possa vir a estar defeituoso, você ainda possa retornar ao estado da noite anterior.

Essa agenda garante redundância maciça e facilita a recuperação de dados. É uma boa solução para uma instalação com muito dinheiro, mas tempo (ou habilidade) de operador escasso. Sob uma perspectiva de conveniência e de segurança, essa agenda é a ideal. Não se afaste dela sem ter uma razão específica (por exemplo, conservar fitas ou trabalho).

Uma agenda moderada

Uma agenda mais razoável para a maioria das instalações é atribuir uma fita a cada dia da semana, cada semana do mês (você precisará de 5) e cada mês do ano. Todo dia, faça um dump nível 9 para a fita diária. Toda semana, faça um dump nível 5 para a fita semanal. E todo mês, faça um dump nível 3 para a fita mensal. Faça um dump nível 0 toda vez que os

incrementos se tornarem muito grandes para caberem em uma fita, o que é mais provável de acontecer em uma fita mensal. Faça um dump nível 0 pelo menos uma vez por ano.

A escolha dos níveis 3, 5 e 9 é arbitrária. Você poderia utilizar os níveis 1, 2 e 3 com o mesmo efeito. Entretanto, os intervalos entre os níveis de dump dão alguma folga caso você decida posteriormente adicionar outro nível de dump.

Essa agenda exige mais 24 fitas, mas muitas fitas são necessárias para os dumps nível 0. Embora não exija um excesso de fitas, ela também não tem recursos para muita redundância.

9.4 RESTAURANDO DUMPS COM RESTORE

O programa que extrai dados de fitas gravadas com **dump** é chamado **restore**. Discutiremos primeiramente a restauração de arquivos individuais (ou um pequeno conjunto de arquivos), depois explicaremos como restaurar sistemas de arquivos inteiros.

Restaurando arquivos individuais

O primeiro passo a dar quando você é notificado da perda de um arquivo é determinar que fitas contêm versões do arquivo. Os usuários normalmente querem a versão mais recente de um arquivo, mas isso não acontece sempre. Por exemplo, um usuário que perde um arquivo copiando inadvertidamente um outro arquivo por cima dele gostaria de ter a versão que existia antes da ocorrência deste incidente. Ajuda se você puder intimidar os usuários a informar não só quais arquivos estão faltando, como também quando eles foram perdidos e modificados pela última vez. Achamos muito útil estruturar as respostas dos usuários com um formulário de solicitação.

Se não mantém catálogos on-line, você terá de montar fitas e tentar repetidamente restaurar os arquivos ausentes até encontrar a fita correta. Se o usuário se lembrar de quando os arquivos foram modificados pela última vez, talvez você possa fazer uma estimativa mais provável de em quais fitas eles poderiam estar.

Após determinar de que fitas fazer a extração, crie um diretório temporário e mude pare ele (**cd**), por exemplo, **/var/restore** onde uma hierarquia de diretórios maior pode ser criada; a maioria das versões de **restore** deve criar todos os diretórios que levam a um determinado arquivo antes de esse arquivo poder ser restaurado. Não use **/tmp** — seu trabalho poderia ser eliminado caso a máquina trave e reinicialize antes de os dados restaurados terem sido deslocados para suas posições originais.

O comando **restore** possui várias opções. As mais úteis são **-i** para restaurações interativas de arquivos e diretórios individuais e **-r** para uma restauração completa de um sistema de arquivos inteiro. Talvez você também precise de **-x**, que solicita uma restauração não interativa de arquivos especificados — tome cuidado para não sobrescrever arquivos existentes.

restore -i lê a tabela de conteúdo da fita e depois permite navegar por meio dela como se fosse uma árvore de diretórios comum, utilizando comandos chamados **ls**, **cd** e **pwd**. Marcamos os arquivos que queremos restaurar por meio do comando **add**. Quando tiver terminado de selecionar os arquivos, digite **extract** para extrair os arquivos da fita.

Consulte a página 126 para obter uma descrição de mt.

Se tiver colocado vários arquivos em uma única fita, você deve utilizar o comando **mt** para posicionar a fita no arquivo de dump correto antes de executar **restore**. Lembre-se de utilizar o dispositivo não rebobinável!

Por exemplo, para restaurar o arquivo **/users/janet/iamlost** em uma unidade de fita remota, talvez você tenha de emitir os seguintes comandos. Suponhamos que você tenha localizado a fita correta, a tenha montado em **tapehost:/dev/nst0** e tenha determinado que o sistema de arquivos contendo o diretório inicial de janet seja o quarto na fita.

```
# mkdir /var/restore
# cd /var/restore
# ssh tapehost mt -f /dev/nst0 fsf 3
# rrestore -i -f tapehost:/dev/nst0
restore> ls
.  :
janet/  garth/  lost+found/ lynda/
restore> cd janet
restore> ls
afile bfile cfile iamlost
restore> add iamlost
restore> ls[8]
afile bfile cfile iamlost*
restore> extract
You have not read any volumes yet.
```

8 O asterisco ao lado de **iamlost** indica que ele foi marcado para extração.

```
Unless you know which volume your files are on you should
start with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] n
```

Os volumes (fitas) são numerados a partir de 1, não 0, de modo que para um dump caber em uma única fita, você especifica 1. Quando **restore** pergunta se você quer configurar o proprietário e o modo para ".", ele está perguntando se ele deve ou não configurar o diretório atual para coincidir com a raiz da fita. A menos que você esteja restaurando um sistema de arquivos inteiro, provavelmente não vai querer fazer isso.

Assim que **restore** tiver terminado, você precisará dar o arquivo a Janet:

```
# cd /var/restore
# ls janet
iamlost
# ls ~janet
afile bfile cfile
# cp -p janet/iamlost ~janet/iamlost.restored
# chown janet ~janet/iamlost.restored
# rm -rf /var/restore
# mail janet
Your file iamlost has been restored as requested and has
been placed in /users/janet/iamlost.restored.

Your name, Humble System Administrator
```

Alguns administradores preferem restaurar arquivos em um diretório especial, permitindo aos usuários copiar seus arquivos manualmente. Nesse esquema, o administrador deve proteger a privacidade dos arquivos restaurados verificando quem são os seus proprietários e respectivas permissões. Se optar por utilizar um sistema como esse, lembre-se de limpar o diretório de vez em quando.

Se criar um backup com **rdump** e não conseguir restaurar arquivos dele com o comando **restore**, tente executar **rrestore**. Para minimizar a possibilidade de problemas, utilize o mesmo computador em que a fita foi gravada para lê-la.

restore -i normalmente é a maneira mais fácil de restaurar alguns arquivos ou diretórios de um dump. Entretanto, ele não funcionará caso o dispositivo de fita não possa ser movido para trás um registro por vez (um problema que ocorre com alguns dispositivos de 8 mm). Se **restore -i** falhar, tente **restore -x** antes de pular pela janela. **restore -x** requer a especificação do caminho completo do arquivo que você quer restaurar (relativo à raiz do dump) na linha de comandos. A seqüência de comandos a seguir repete o exemplo anterior, mas com **-x**:

```
# mkdir /var/restore
# cd /var/restore
# ssh tapehost mt -f /dev/nst0 fsf 3
# rrestore -x -f tapehost:/dev/nst0 ./janet/iamlost
```

Restaurando sistemas de arquivos inteiros

Com sorte, você nunca terá de restaurar um sistema de arquivos inteiro após uma falha no sistema. Entretanto, essa situação ocorre eventualmente. Antes de tentar restaurar o sistema de arquivos, certifique-se de qual problema que causou a destruição do sistema de arquivos foi resolvido. É inútil gastar horas e horas rebobinando fitas somente para perder o sistema de arquivos novamente.

Antes de iniciar uma restauração completa, você tem de criar e montar o sistema de arquivos de destino. Consulte o Capítulo 7, *Adicionando um disco*, para informações adicionais sobre como preparar o sistema de arquivos. Para iniciar a restauração, dê o comando **cd** para o ponto de montagem do novo sistema de arquivos, coloque a primeira fita do dump nível 0 mais recente na unidade de fita e digite **restore -r**.

restore irá solicitar cada fita no dump. Depois de o dump nível 0 ter sido restaurado, monte e restaure os dumps incrementais. Restaure dumps incrementais na ordem em que eles foram criados. Por causa da redundância entre dumps, talvez não seja necessário restaurar todos os incrementais. Eis o algoritmo para determinar que dumps restaurar:

Passo 1: Restaure o dump nível 0 mais recente.

Passo 2: Restaure o dump de nível mais baixo após o dump que você acabou de restaurar. Se forem feitos vários dumps neste nível, restaure o mais recente.

Passo 3: Se esse foi o último dump que foi feito, está terminado o processo.

Passo 4: Caso contrário, retorne para o passo 2.

Eis alguns exemplos de seqüências de dump. Você precisaria restaurar somente os níveis indicados em negrito.

00000**0**
05555
032545
0995993995**9**9
0359359

Consulte o Capítulo 7 para informações adicionais sobre mke2fs e mount.

Vamos dar uma olhada na seqüência de comandos completa. Se o dump mais recente fosse o primeiro mensal após o nível 0 anual na agenda "moderada" da página 121, os comandos para restaurar **/home**, residentes no dispositivo físico **/dev/sda1**, ficariam da seguinte maneira (os nomes do dispositivo são dependentes do hardware):

```
# /etc/mke2fs -j /dev/sda1 QUANTUM_PD1050S
# /etc/mount /dev/sda1 /home
# cd /home
/* Monta a primeira fita de dump nível 0 de /home. */
# restore -r
/* Monta as fitas solicitadas por restore. */
/* Monta a primeira fita de dump mensal nível 3. */
# restore -r
```

Se tiver vários sistemas de arquivos em uma única fita de dump, utilize o comando **mt** para pular para o sistema de arquivos correto antes de executar cada **restore**. Consulte a página 126 para obter uma descrição de **mt**.

Essa seqüência restauraria o sistema de arquivos ao estado em que ele se encontrava quando o dump nível 3 foi feito, exceto que todos os arquivos eliminados seriam ressuscitados de forma macabra. Esse problema pode ser especialmente ruim quando você estiver restaurando um sistema de arquivos ativo ou restaurando para um disco que está praticamente cheio. É bem possível um **restore** falhar por causa de o sistema de arquivos ter sido preenchido com arquivos fantasmas.

9.5 REALIZANDO DUMP E RESTAURAÇÃO PARA ATUALIZAÇÕES

Recomendamos que, ao realizar uma atualização de sistema operacional (OS) importante, você faça backup de todos os sistemas de arquivos com um dump nível 0 e, possivelmente, sua restauração. A restauração só é necessária se o novo sistema operacional utilizar um formato de sistema de arquivos distinto ou se você modificar o particionamento dos discos. Entretanto, você *deve* fazer backups como uma garantia contra quaisquer problemas que possam vir a ocorrer durante a atualização. Um conjunto completo de backups também dá a opção de reinstalar o sistema operacional antigo caso a nova versão não se mostre satisfatória. Felizmente, com os sistemas de atualização progressiva utilizados pela maioria das distribuições de hoje em dia, é improvável que você precise dessas fitas.

Certifique-se de fazer backups e restaurar quaisquer arquivos específicos que se encontram no sistema de arquivos root ou em **/usr**, como **/etc/passwd**, **/etc/shadow** ou **/usr/local**. A organização de diretórios do Linux mistura arquivos locais com arquivos distribuídos pelos fornecedores de sistemas, tornando difícil escolher suas personalizações locais.

Você deve fazer também um conjunto completo de dumps nível 0 imediatamente após uma atualização. Os procedimentos de atualização da maioria dos fornecedores configuram as datas de modificação de arquivos de sistema com o horário em que eles foram masterizados, e não com o horário atual. Portanto, os dumps incrementais feitos em relação ao nível 0 anterior à atualização não são suficientes para restaurar o sistema ao estado pós-atualização no caso de uma pane.

9.6 UTILIZANDO OUTROS PROGRAMAS DE ARQUIVAMENTO

dump não é o único programa que você pode utilizar para arquivar arquivos em fitas; entretanto, ele é, em geral, a maneira mais eficiente de fazer backup de um sistema inteiro. **tar**, **cpio** e **dd** também podem mover arquivos de uma mídia para outra.

tar: empacotando arquivos

tar lê vários arquivos ou diretórios e os empacota em um único arquivo, normalmente um arquivo em fita. **tar** é uma maneira útil de fazer backup de quaisquer arquivos cuja recuperação a curto prazo é prevista. Por exemplo, se um usuário se ausentar por seis meses e o sistema estiver com pouco espaço em disco, você pode utilizar **tar** para colocar os arquivos do usuário em uma fita e depois eliminá-los do disco.

tar também é útil para mover árvores de diretórios de um lugar para outro, especialmente se você estiver copiando arquivos como root. **tar** preserva informações de posse e data/hora, mas somente se solicitadas. Por exemplo,

```
tar -cf - diretórioDeOrigem | ( cd diretórioDeDestino ; tar --atime-preserve -xpf - )
```

cria uma cópia da árvore de diretórios *diretórioDeOrigem* para *diretórioDeDestino*. Evite utilizar ".." no argumento *diretórioDeDestino*, já que links simbólicos e automontadores podem fazer com que ele signifique algo diferente do esperado. Já caímos nessa várias vezes.

Como padrão **tar** não segue links simbólicos, porém podemos informá-lo para proceder desta forma. **tar** também pode ser informado para incluir somente arquivos que foram modificados desde uma determinada data, o que é útil para criar seu próprio esquema de backup incremental. Consulte a página de manual do **tar** referente a esse e outros recursos interessantes.

Um problema com algumas versões não Linux de **tar** é que os nomes de caminhos são limitados por padrão a 100 caracteres. Essa restrição impede que o **tar** arquive hierarquias mais profundas. Se estiver criando arquivos **tar** em seus sistemas Linux e trocando-os por outros, lembre-se de que pessoas com o **tar** padrão podem não ser capazes de ler as fitas ou arquivos que você criou.

A opção **-b** de **tar** permite especificar um "fator de bloco" a ser adotado ao gravar uma fita. O fator de bloco é especificado em blocos de 512 bytes; ele determina a quantidade de dados que o comando **tar** armazena internamente em buffer antes de executar uma operação de gravação. Alguns dispositivos DAT não funcionam corretamente a menos que o fator de bloco seja configurado com um valor especial, mas outras unidades não exigem esse tipo de configuração.

Em alguns sistemas, determinados fatores de bloco podem levar a um desempenho melhor do que outros. O fator de bloco ótimo varia muito, dependendo do hardware do computador e da unidade de fita. Em muitos casos, não perceberemos qualquer diferença em termos de velocidade. Na dúvida, tente um fator de bloco 20.

O **tar** do Linux expande brechas em arquivos a menos que você utilize a opção **-S** ao criar o arquivo original. O **tar** do Linux é relativamente intolerante a erros de fita.

cpio: utilitário de arquivamento arcaico

cpio é similar a **tar** em funcionalidade. Data desde o início dos tempos e não é mais muito utilizado.

Como o **tar**, o **cpio** pode mover árvores de diretórios. O comando:

```
find diretórioDeOrigem -depth -print | cpio -pdm diretórioDeDestino
```

cria uma cópia da árvore de diretórios *diretórioDeOrigem* para *diretórioDeDestino*. A versão GNU de **cpio** utilizada no Linux possibilita vários volumes de fitas, mas a maioria delas não permite. Apenas o superusuário pode copiar arquivos especiais. Mesmo se você estiver familiarizado com **cpio** a partir de um outro sistema, recomendamos revisar a página man atentamente porque as opções variam muito entre sistemas.

dd: lidando com bits

dd é um programa para cópia e conversão de arquivos. A menos que lhe seja dito para fazer algum tipo de conversão, **dd** simplesmente copia do seu arquivo de entrada para o seu arquivo de saída. Se um usuário lhe traz uma fita que foi escrita em algum sistema não Linux, **dd** pode ser o único modo de lê-lo.

Um emprego histórico para **dd** foi o de criar uma cópia de um sistema de arquivos inteiro. Porém, uma opção melhor nos dias de hoje é aplicar **mke2fs** ao sistema de arquivos de destino e depois executar **dump** acoplado a **restore**. **dd** pode algumas vezes danificar informações de particionamento se utilizado incorretamente. Ele só pode copiar sistemas de arquivos entre partições cujos tamanhos sejam exatamente iguais.

dd também pode ser utilizado para fazer uma cópia de uma fita magnética. Com duas unidades de fita, digamos, **/dev/st0** e **/dev/st1**, você deve utilizar o comando

```
$ dd if=/dev/st0 of=/dev/st1 cbs=16b
```

Com uma unidade (**/dev/st0**), você deveria utilizar a seguinte seqüência:

```
$ dd if=/dev/st0 of=tfile cbs=16b
/* Troca de fita. */
$ dd if=tfile of=/dev/st0 cbs=16b
$ rm tfile
```

Obviamente, caso você tenha apenas uma unidade de fita, precisará ter espaço em disco suficiente para armazenar uma imagem de toda a fita.

9.7 USANDO VÁRIOS ARQUIVOS EM UMA ÚNICA FITA

Na realidade, uma fita magnética contém uma longa string de dados. Entretanto, normalmente é útil armazenar mais do que uma "coisa" em uma fita, de modo que as unidades de fita e seus drivers Linux conspirem para fornecer-lhe um pouco mais de estrutura. Quando **dump** ou algum outro comando grava um fluxo de bytes em um dispositivo de fita e, então, fecha o arquivo de dispositivo, um marcador de final de arquivo (*end-of-file* – EOF) é automaticamente colocado na fita. Essa marca separa um fluxo de dados de outros fluxos de dados que são gravados subseqüentemente. Quando o fluxo é lido mais uma vez, a leitura pára automaticamente em EOF.

Você pode utilizar o comando **mt** para posicionar uma fita em determinado fluxo ou "conjunto de arquivos", à medida que **mt** os chama. **mt** é especialmente útil se você colocar vários arquivos (por exemplo, vários dumps) em uma única fita. Ele também possui algumas das mensagens de erro mais interessantes de qualquer utilitário Linux. O formato básico do comando é

```
mt [-f nomeDaFita] comando [contagem]
```

Há várias opções para *comando*. Elas variam entre plataformas, por isso discutiremos apenas as que são essenciais para fazer backups e restaurações:

rew	rebobina a fita para o começo.
offl	coloca a fita no estado off-line. Na maioria das unidades de fita, esse comando faz com que a fita se rebobine e saia da unidade. A maioria dos scripts utiliza esse comando para ejetar a fita ao terminarem, indicando claramente que tudo foi concluído de modo correto.
status	imprime informações sobre o estado atual da unidade de fita (se a fita está carregada etc.).
fsf [*contagem*]	avança rapidamente a fita. Se não for dada nenhuma *contagem*, **fsf** avança um arquivo. Com um argumento numérico, ele pula o número de arquivos especificados. Use esse comando para avançar para o sistema de arquivos correto em uma fita com vários dumps.
bsf [*contagem*]	deve retornar o número de arquivos indicado na *contagem*. O comportamento exato desta diretiva depende do hardware da unidade de fita e seu driver associado. Em algumas situações, o arquivo atual é contado. Em outras, não é assim. Em alguns equipamentos, **bsf** não faz nada silenciosamente. Se você avançar muito em uma fita, sua melhor opção é executar **mt rew** e começar desde o início.

Consulte a página man **mt** para obter uma lista de todos os comandos suportados.

Se tiver sorte o bastante para ter uma biblioteca de fitas robotizada, você pode ser capaz de controlar seu trocador de fitas instalando o pacote **mtx**, uma versão melhorada do comando **mt**. Nós o utilizamos, por exemplo, para troca de fitas sem a intervenção de operadores em nosso maravilhoso sistema de cartuchos de fita HP 6x24 DAT. É só para olhar, não para tocar!

9.8 BACULA

O Bacula é uma solução de backup cliente/servidor corporativa que gerencia backup, recuperação e verificação de arquivos em uma rede. O Bacula é executado em uma variedade de sistemas UNIX e Linux, inclusive em todos os nossos exemplos de distribuições. Ele também faz backup de dados a partir de vários sistemas operacionais, incluindo Microsoft Windows.

Na edição anterior deste livro, o Amanda era nossa ferramenta de backup não comercial favorita. Se você precisar de informações sobre o Amanda, consulte a primeira edição deste livro ou www.amanda.org. A lista de recursos a seguir explica por que o Bacula é nosso novo favorito:

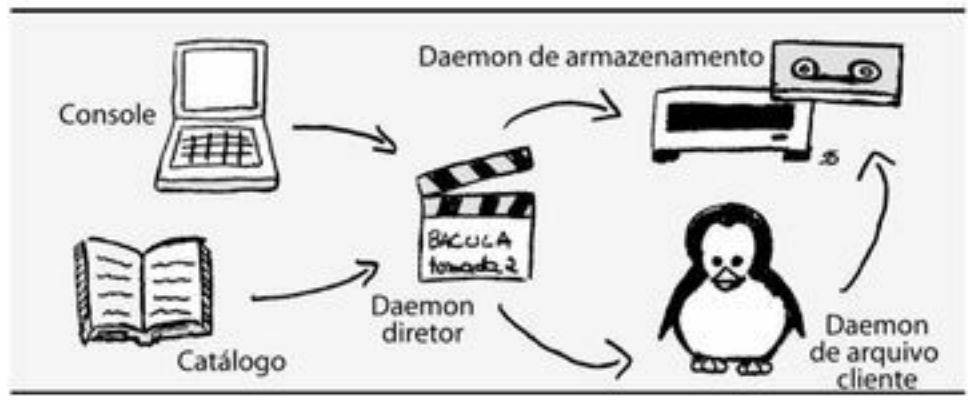
- Ele tem um projeto modular
- Ele faz backup de sistemas UNIX, Linux e Windows
- Ele suporta MySQL, PostgreSQL ou SQLite para seu banco de dados de back-end
- Ele suporta um console de linha de comando baseado em menus fácil de utilizar
- Ele está disponível sob uma licença de código-fonte aberto
- Seus backups podem abranger vários volumes de fita

- Seus servidores podem executar em várias plataformas
- Ele cria arquivos de assinatura SHA1 ou MD5 para cada arquivo cujo backup foi feito
- Ele verifica tarefas de backup
- Ele suporta bibliotecas de fita e trocadores automáticos
- Ele pode executar scripts ou comandos antes e depois das tarefas de backup
- Ele centraliza gerenciamento de backup para uma rede inteira

O modelo Bacula

Para implantar o Bacula, você deve entender seus importantes componentes. A Ilustração A ilustra a arquitetura geral do Bacula.

Ilustração A Componentes do Bacula e seus relacionamentos



O diretor do Bacula é o daemon que coordena operações de backup, restauração e verificação. Você pode enviar tarefas de backup ou restauração para o daemon de diretor utilizando o console do Bacula. Você também pode solicitar ao daemon do diretor para consultar o daemon de armazenamento do Bacula ou os daemons de serviço de arquivo localizados em computadores clientes.

Você se comunica com o daemon de diretor pelo console do Bacula, que pode ser executado como uma interface gráfica GNOME GUI ou uma ferramenta de linha de comando. O console pode ser executado em qualquer lugar; ele não precisa estar no mesmo computador em que reside o daemon de diretor.

Um daemon de armazenamento é o componente do Bacula que lê e grava fitas ou outra mídia de backup. Esse serviço deve executar na máquina que está conectada à unidade de fita ou dispositivo de armazenamento utilizado para backups, mas não precisa estar instalado no mesmo servidor em que reside o diretor (embora ele possa estar).

O daemon de arquivo do Bacula é executado em cada sistema cujo backup precisa ser feito. As implementações de daemon de arquivo para cada sistema operacional suportado enviam os dados e os atributos de arquivos apropriados para o daemon de armazenamento à medida que as tarefas de backup são executadas.

O componente final do Bacula é o catálogo, um banco de dados relacional em que o Bacula armazena informações sobre cada arquivo e volume que é colocado em backup. O catálogo torna o Bacula rápido e eficiente durante uma restauração porque a história inteira do backup está disponível on-line; o Bacula sabe quais os volumes de armazenamento necessários para restaurar um determinado conjunto de arquivos antes de ele ler uma única fita. Atualmente, o Bacula suporta três bancos de dados diferentes: MySQL, PostgreSQL e SQLite. O banco de dados de catálogo não precisa residir no mesmo servidor do diretor.

Um componente opcional adicional é o Bacula Rescue CD-ROM. Esse componente é um pacote que pode ser instalado separadamente que cria CDs de salvamento individualizados, inicializáveis para sistemas Linux utilizarem durante recuperação após desastre. Os CDs contêm uma cópia estaticamente vinculada do daemon de arquivo do sistema bem como scripts de shell personalizados que incorporam informações de configuração sobre os discos, o kernel e as interfaces de rede do sistema. Se um sistema Linux tiver uma falha catastrófica, você pode utilizar seu CD de salvamento para inicializar o sistema, reparticionar o disco e conectar-se ao diretor do Bacula para realizar uma restauração completa do sistema na rede.

Configurando o Bacula

Por causa da complexidade, conjunto de recursos avançados e projeto modular do Bacula, há muitas maneiras de configurar um esquema de backup no nível do site. Nesta seção percorremos uma configuração Bacula básica.

Em geral, cinco passos colocam o Bacula em funcionamento:

- Instale um banco de dados independente suportado e os daemons do Bacula
- Configure os daemons do Bacula
- Instale e configure os daemons de arquivo de cliente
- Inicie os daemons Bacula
- Adicione mídia a pools de mídia com o console do Bacula

O exemplo de configuração consiste em duas máquinas, harp e bull. A máquina harp é o cliente; ela executa apenas um daemon de arquivo. Os quatro componentes do Bacula restantes (o daemon de diretor, o daemon de armazenamento, o catálogo e o console) são executados no servidor bull. O dispositivo de armazenamento é uma biblioteca de fitas SureStore LTO1. A agenda de backups consiste em backups semanais completos seguidos por backups incrementais diários. O MySQL é utilizado para o catálogo.

Instalando o banco de dados e os daemons do Bacula

Antes que possamos instalar o Bacula, devemos primeiramente instalar o banco de dados de back-end para seu catálogo. Para integração fácil com o Bacula, recomendamos o MySQL com seus caminhos de configuração padrão.

A estabilidade e a confiabilidade são uma necessidade ao lidar com uma plataforma de backup, portanto, assim que você terminar de instalar o banco de dados, recomendamos fazer o download e a instalação do último código-fonte estável a partir do site do Bacula. A documentação da instalação passo a passo vem com o código-fonte no diretório **docs**. Ela também está disponível on-line em www.bacula.org, tanto no formato HTML como PDF. Tutoriais úteis e guias de desenvolvedores também podem ser encontrados nesse site.

Depois de desempacotar o código-fonte, executamos

./configure --with-mysql --prefix=/etc/bacula

seguido por **make** para compilar o binário e, por fim, **make install** para completar a instalação.

Assim que o Bacula foi instalado, o próximo passo foi criar o banco de dados MySQL real e as tabelas de dados dentro dele. Certificamo-nos de que MySQL estava executando, então aplicamos **cd** ao diretório de instalação (**/etc/bacula**) e executamos três scripts de shell que foram criados como parte do procedimento **make install**. O script **grant_mysql_privileges** concede os privilégios de acesso de usuário Bacula ao MySQL. O script **create_mysql_database** cria o banco de dados do Bacula e, por fim, o script **make_mysql_tables** preenche o banco de dados com as tabelas necessárias.

Configurando os daemons do Bacula

Antes de começarmos uma discussão mais detalhada de nossa configuração de exemplo, definimos primeiro alguns termos-chave do Bacula:

- "Tarefas" (jobs) são a unidade de atividade do Bacula. Há duas versões: backup e restauração.
- "Pools" são grupos de mídia física que armazenam tarefas. Nossa configuração de exemplo utiliza dois pools, um para backups completos e outro para backups incrementais.
- "Conjuntos de arquivos" (filesets) são listas de partições e arquivos. Os conjuntos de arquivos podem ser explicitamente incluídos, ou excluídos de, em tarefas de backup ou restauração.
- O arquivo de "bootstrap" é um arquivo de texto especial, criado pelo Bacula, que contém informações sobre os arquivos que devem ser restaurados. Os arquivos de bootstrap são criados durante o comando restore ou durante um backup se o parâmetro `Write Bootstrap` for definido para a tarefa de backup.
- "Messages" são comunicações inter-daemon (realmente entradas de log) relacionadas com o status de daemons e tarefas. As mensagens também podem ser enviadas por e-mail e gravadas em arquivos de log.

Os arquivos de configuração residem no diretório especificado com a opção **--prefix** durante a instalação, normalmente **/etc/bacula**. Nas próximas seções, abordamos as opções para o daemon de diretor (**bacula-dir.conf**), o daemon de armazenamento (**bacula-sd.conf**) e o console (**bconsole.conf**).

Não abordamos todos os possíveis parâmetros de configuração nos exemplos de arquivos de configuração. Em vez disso, começamos cada seção com uma visão geral e, então, indicamos alguns parâmetros que consideramos particularmente úteis ou complexos.

bacula-dir.conf: configuração do diretor

O arquivo **bacula-dir.conf** é o mais complexo dos arquivos de configuração do Bacula. Examinamos a seguir suas dez seções lógicas: `Director`, `Catalog`, `JobDefs`, `Job`, `FileSet`, `Schedule`, `Pool`, `Client`, `Storage` e `Messages`. Informações mais detalhadas sobre cada seção e seus parâmetros podem ser encontradas na documentação on-line.

Cada seção de recurso está entre chaves. Os comentários são introduzidos com um sinal # em todos os arquivos de configuração do Bacula.

```
# Sample Bacula director configuration file, /etc/bacula-dir.conf

Director {
    Name = bull-dir
    DIRport = 9101
    Query File = "/etc/bacula/query.sql"
    Working Directory = "/var/Bacula/working"
    Pid Directory = "/var/run"
    Maximum Concurrent Jobs = 1
    Password = "B@cul@Lik3s,fRul7"
    Messages = Standard
}
```

O recurso `Director` é mais ou menos o navio mãe no mar do Bacula. Seus parâmetros definem o nome e o comportamento básico do diretor. As opções configuram a porta de comunicação pela qual os outros daemons se comunicam com o diretor, a localização em que o diretor armazena seus arquivos temporários e o número de tarefas concorrentes que o diretor pode tratar de uma vez.

O parâmetro `Password` define a senha que o programa de console utiliza para autenticar-se com o diretor. Uma senha idêntica deve ser configurada no recurso `Director` do arquivo de configuração de console, **bconsole.conf**. Embora a senha apareça em texto simples nos arquivos de configuração, ela nunca é transmitida pela rede.

No exemplo de configuração, o diretor e o console são hospedados na mesma máquina. Entretanto, ainda é necessária uma senha em ambos os arquivos de configuração.

```
# Generic catalog service

Catalog {
    Name = ATE
    DBName = bacula;
    user = bacula;
    password = "fRul7,BOwL"
}
```

A seção de recurso `Catalog` identifica o banco de dados em que o Bacula armazena seus registros operacionais. Para simplificar, definimos um catálogo para todas as tarefas. Entretanto, o Bacula suporta vários catálogos, e esse recurso é útil se você preferir manter um banco de dados separado para cada grupo de cliente. Atualmente, todos os bancos de dados de catálogo devem residir no mesmo servidor de banco de dados físico.

O parâmetro `DBName` é o nome do banco de dados de catálogo. Esse banco de dados é configurado durante a instalação de acordo com as opções de banco de dados passadas para **configure**. Os scripts de criação de banco de dados predefinidos do Bacula podem ser encontrados no diretório **src/cats** da distribuição de código-fonte Bacula.

```
# Job definitions

Job {
    Name = "harp"
    Level = Full
    Write Bootstrap = "/atrust/admin/backups/bootstraps/harp.bsr"
    Client = harp
    File Set = harp
    Storage = SureStore
    Pool = SSFullPool
    Incremental Backup Pool = SSIncrementalPool
    Schedule = Nightly
}
```

O recurso Job define os parâmetros padrão para uma determinada tarefa de backup. Em geral, há uma definição Job por cliente. A maioria dos parâmetros é autoexplicativa, mas alguns merecem discussão adicional.

Write Bootstrap informa ao Bacula onde gravar informações de bootstrap sobre o backup para utilização durante uma restauração. Os arquivos de bootstrap listam os arquivos e volumes necessários para uma tarefa de restauração. Eles não são obrigatórios, mas altamente recomendados. Os arquivos de bootstrap são sobrescritos durante os backups completos e anexados durante os backups incrementais.

Os parâmetros Client, File Set, Storage, Pool e Schedule são referências de encaminhamento para recursos definidos posteriormente no arquivo **bacula-dir.conf**.

```
# File set definitions

FileSet {
    Name = "harp"
    Include {
         Options {
             signature=MD5
             compression=GZIP
        }
        File = "/"
        File = "/boot"
        File = "/usr"
        File = "/usr/local"
        File = "/var"
   }
   Exclude = { /proc /tmp /.journal /.fsck }
}
```

O recurso FileSet define os arquivos e diretórios a ser incluídos em, ou excluídos de, uma tarefa de backup. Cada conjunto de arquivo pode definir vários parâmetros Include e Exclude juntos com Options individuais. Por padrão, o Bacula faz backup recursivo de diretórios, mas não abrange partições. Tome cuidado ao listar todas as partições das que você quer fazer backup em parâmetros File separados.

Muitas Options adicionais são suportadas, inclusive expressões regulares e caracteres curinga. Duas opções dignas de nota são signature (configuradas como SHA1 ou MD5), que computam um valor de hash para cada arquivo copiado em backup, e compression, que compacta os dados antes de gravá-los na fita. A opção signature aumenta o tempo gasto de CPU de backups, mas pode revelar-se valiosa durante um incidente de segurança suspeito.

```
Schedule {
    Name = "Nightly"
    Run = Level=Full Pool=SSFullPool 1st-5th tue at 20:10
    Run = Level=Incremental Pool=SSIncrementalPool wed-mon at 20:10
}
```

Os recursos Schedule definem horários para tarefas de backup. O parâmetro Name e pelo menos um parâmetro Run são necessários para tarefas de backup automáticas, mas múltiplos parâmetros Run podem ser incluídos em um único recurso Schedule. Aqui, os backups completos executam na primeira e na quinta terça-feira de cada mês às 8h10 da noite e os backups incrementais executam semanalmente de quarta a segunda-feira às 8h10 da noite.

As especificações de data e hora são a única parte necessária do parâmetro Run, mas, como você pode ver nesse exemplo, é possível inserir valores de parâmetro adicionais. Esses valores então anulam os parâmetros padrão configurados na especificação Job. Aqui, configuramos o Level do backup e o Pool da mídia para utilização em cada backup.

Consulte a documentação on-line do Bacula para uma lista completa de palavras-chave de especificação de data/hora e parâmetros de tarefas.

```
# Pool definitions -- first pool is default

Pool {
     Name = SSFullPool
     Pool Type = Backup
     Recycle = yes
     Accept Any Volume = yes
}

Pool {
```

```
        Name = SSIncrementalPool
        Pool Type = Backup
        Recycle = yes
        Accept Any Volume = yes
}
```

O recurso `Pool` agrupa a mídia de backup, em geral fitas, em conjuntos que são utilizados por tarefas de backup específicas. Neste exemplo executamos backups completos semanais e backups incrementais diários, então criamos dois recursos Pool. Muitas opções se ajustam ao comportamento do recurso `Pool`; dois parâmetros particularmente úteis são `Recycle` e `Accept Any Volume`. Se o parâmetro `Recycle` estiver configurado como `yes`, o Bacula automaticamente utiliza volumes purgados cujo flag de reciclagem é configurado quando ele precisa de um volume anexável para uma tarefa de backup. O parâmetro `Accept Any Volume` especifica se o Bacula pode gravar em qualquer volume anexável dentro do pool. Se essa opção é configurada como `no`, o Bacula preenche volumes seqüencialmente e não aceita um volume anexável fora de seqüência.

```
Client {
        Name = harp
        Address = 192.168.7.2
        FDPort = 9102
        Catalog = ATE
        Password = "Ch@ch1s@Fru17"
        File Retention = 3 months
        Job Retention = 36 months
        AutoPrune = yes
}
```

Os recursos `Client` identificam os computadores a ser copiados em backup. É necessário um para cada computador. Os parâmetros `File Retention` e `Job Retention` especificam quanto tempo o arquivo e os registros de tarefa para esse cliente são mantidos no catálogo. Se o parâmetro `AutoPrune` é configurado, então os dados expirados são excluídos do catálogo. O corte (*pruning*) afeta apenas os registros de catálogo e não os arquivos reais armazenados em fitas de backup.

```
# Definition of file storage devices

Storage {
        Name = SureStore
        Address = bull.atrust.com
        SDPort = 9103
        Password = "Frult&V3ggles"
        Device = SureStoreDevice
        Autochanger = yes
        Media Type = LTO1
}
```

O recurso `Storage` descreve como se comunicar com o daemon de armazenamento, que controla os dispositivos de backup. No nosso exemplo, temos uma biblioteca de fita SureStore LTO1 conectada ao nosso servidor Bacula primário (bull.atrust.com). O daemon de armazenamento tem seu próprio arquivo de configuração que está revisado a seguir.

```
Messages {
    Name = Standard
    mailcommand = "/sbin/bsmtp -h localhost -f \"\(Bacula\)
        bacula@atrust.com\" -s \"Bacula: %t %e of %c %l\" %r"
    operatorcommand = "/sbin/bsmtp -h localhost -f \"\(Bacula\)
        bacula@atrust.com\" -s \"Bacula: Intervention needed for %j\" %r"
    mail = Bacula@atrust.com = all, !skipped
    operator = bacula-pager@atrust.com = mount
    console = all, !skipped, !saved
    append = "/var/log/bacula.log" = all, !skipped
}
```

O recurso `Messages` diz ao Bacula como tratar tipos de mensagem específicos gerados por cada daemon do Bacula. Vários recursos `Messages` podem ser definidos e, então, atribuídos a tarefas específicas em seus recursos `Job`.

Esse tipo do recurso é muito configurável; uma lista completa de variáveis e comandos pode ser encontrada na documentação on-line. A configuração anterior envia informações sobre o status de daemons e tarefas para o console e para um arquivo de log padrão, bem como as distribui por e-mail.

bacula-sd.conf: configuração de daemon de armazenamento

Os daemons de armazenamento aceitam dados de daemons de arquivo e os transferem para a mídia de armazenamento real (ou vice-versa, no caso de uma restauração). Quatro recursos devem ser definidos dentro do arquivo **bacula-sd.conf**: `Storage`, `Device`, `Messages` e `Director`. Eis um exemplo de configuração completa:

```
# Storage daemon configuration file, bacula-sd.conf

Storage {
    Name = bull-sd
    SDPort = 9103
    Working Directory = "/var/bacula/working"
    Pid Directory = "/var/run"
}

Device {
    Name = SureStoreDevice
    Media Type = LTO1
    Archive Device = /dev/nst0
    Autochanger = yes
    Changer Device = /dev/sg0
    Changer Command =  "/etc/bacula/mtx-changer %c %o %S %a %d"
    AutomaticMount = yes;
    Always Open = yes;
    Removable Media = yes;
    Random Access = no;
}

Messages {
    Name = Standard
    director = bull-dir = all
}

Director {
    Name = bull-dir
    Password = "Frult&V3ggles"
}
```

Um arquivo de configuração do daemon de armazenamento deve conter apenas um recurso `Storage`.

Esses recursos são relativamente simples e diretos. Eles definem alguns parâmetros de funcionamento básicos, como a porta de rede e o diretório funcional do daemon e identificam o daemon de diretor para o qual o daemon de armazenamento deve enviar mensagens.

O recurso `Device` caracteriza o dispositivo de backup real. Em nosso caso, é uma unidade LTO (Linear Tape-Open) com um trocador de fita automático. Você pode definir múltiplos recursos `Device`. O parâmetro `Archive Device` nomeia o arquivo de dispositivo para a unidade de fita; observe que **/dev/nst0** é um dispositivo não rebobinável que, invariavelmente, é quase o que você quer. O trocador de fita automático tem seu próprio arquivo de dispositivo; além de especificar isso, também configura o parâmetro `Autochanger`. O parâmetro `Always Open` pede para o Bacula manter o dispositivo aberto a menos que um administrador solicite uma desmontagem especificamente. Essa opção economiza tempo e *stress* de fita porque evita comandos de rebobinar e posicionar entre tarefas.

bconsole.conf: configuração de console

Você utiliza o programa de console para se comunicar com o diretor a fim de agendar tarefas, verificar o status de tarefas ou restaurar dados. Você pode iniciar o console a partir do diretório de instalação digitando **./bconsole**.

bconsole.conf informa ao console como se comunicar com o daemon de diretor do Bacula. Os parâmetros nesse arquivo devem corresponder àqueles dados no recurso `Director` no arquivo de configuração do diretor (**bacula-dir.conf**), com a exceção do parâmetro `address`.

```
# Console configuration file, bconsole.conf
```

```
Director {
    Name = bull-dir
    DIRport = 9101
    address = bull.atrust.com
    Password = "B@cul@Lik3s.fRul7"
}
```

Instalando e configurando o daemon de arquivo de cliente

O daemon de arquivo em clientes de backup comunica-se com o daemon de armazenamento do Bacula à medida que os backups e as restaurações são executados. Esse daemon deve ser instalado e configurado em cada computador que deve ser copiado durante o backup com o Bacula.

Para clientes Windows, o download dos binários pode ser feito no site do Bacula. O Bacula é excelente para fazer backup de arquivos de dados do Windows, mas não é tão bom na criação de um backup nível 0 à prova de bomba de um sistema Windows. Infelizmente, o Bacula não tem nenhum conceito de bloqueio de arquivo Windows aberto ou registro ou estado do sistema Windows, portanto, passos adicionais devem ser seguidos para assegurar que esses dados sejam salvos antes de um dump do Bacula ser realizado. A documentação on-line faz um bom trabalho ao abordar essas questões.

Nos sistemas UNIX e Linux, pode-se instalar o daemon copiando a árvore original do Bacula para cada cliente e executando

./configure --enable-client-only --prefix=/etc/bacula

Em seguida, execute **make** e **make install.**

Depois de o binário ser instalado, configure o daemon de arquivo editando o arquivo **/etc/bacula/bacula-fd.conf**:

```
# File daemon configuration for harp, bacula-fd.conf

Director {
    Name = bull-dir
    Password = "Ch@ch1s@Frul7"
}

# "Global" file daemon configuration specifications

FileDaemon {
    Name = harp
    FDport = 9102
    Working Directory = /var/bacula/working
    Pid Directory = /var/run
}

# Send all messages except skipped files back to the director

Messages {
    Name = Standard
    director = bull-dir = all, !skipped
}
```

O arquivo de configuração é dividido em três partes. A primeira consiste no recurso `Director`, que informa o daemon de arquivo que o diretor pode agendar backups do daemon de arquivo desse cliente. O recurso `Director` também inclui o parâmetro `Password`, que deve ser idêntico à senha listada no recurso `Client` dentro do próprio arquivo de configuração do diretor. A segunda parte faz o recurso `FileDaemon`, que nomeia o cliente e especifica a porta em que o daemon de arquivo escuta para comandos a partir do daemon de diretor. O componente final é o recurso `Messages`, que define como mensagens locais devem ser tratadas.

Iniciando os daemons Bacula

Agora que temos nossos daemons de servidor instalados e nosso cliente configurado, o próximo passo é disparar os daemons executando o script de inicialização no diretório de instalação do servidor (**./bacula start**). Esse mesmo comando também é utilizado em cada cliente para iniciar o daemon de arquivo. Esse script também deve ser vinculado aos scripts de inicialização de nível da execução apropriados nos diretórios **rc** e **init.d**.

Assim que os daemons do Bacula entrarem em execução, você pode utilizar o programa de console (**bconsole** no diretório de instalação) para verificar seu status, adicionar mídia a pool e executar backup e restaurar tarefas. Você pode executar **bconsole** em qualquer computador contanto que ele tenha sido adequadamente instalado e configurado.

```
$ sudo ./bconsole
Password: <senha inserida>
Connecting to Director bull:9101
1000 OK: bull-dir Version: 1.38.11 (29 June 2006)
Enter a period to cancel a command.
```

Utilize o comando **help** do console para ver uma lista completa dos comandos que ele suporta.

Adicionando mídia a pools

Antes de poder executar tarefas de backup, você precisa rotular algumas fitas e as atribuir a pools de mídia definidos no arquivo de configuração de diretor. Utilize o comando **label** do console para fazer isso.

```
* label
Automatically selected Storage: SureStore
Enter new Volume name: 003061L1
Enter slot (0 for none): 14
Defined Pools:
     1: SSFullPool
     2: SSOneOffPool
...
Select the Pool (1-5): 1
Connecting to Storage daemon SureStore at bull.atrust.com:9103
...
Sending label command for Volume "003061L1" Slot 14 ...
 ...
3001 Mounted Volume: 003061L1
3001 Device /dev/nst0 is mounted with Volume "003061L1"
```

Neste exemplo, a fita no slot 14 do trocador de fita automático era chamada 003061L1 e foi atribuída ao pool SSFullPool. Utilize o comando **list media** para verificar se a fita foi adicionada ao pool correto e marcada como anexável.

Executando um backup manual

Utilize o comando **run** do console para realizar um backup manual. Nenhum argumento é necessário; o console exibe todas as tarefas de backup definidas no arquivo de configuração do diretor. Você pode modificar qualquer opção dentro do comando **run** seguindo as perguntas baseadas nos menus do console.

O seguinte exemplo mostra um backup manual completo do servidor harp, utilizando os padrões especificados em nossos arquivos de configuração.

```
$ sudo ./bconsole
...
Connecting to Director bull:9101
1000 OK: bull-dir Version: 1.34.6 (28 July 2004)
Enter a period to cancel a command.
* run
A job name must be specified.
The defined Job resources are:
    1: harp
    2: RestoreFiles
Select Job resource (1-2): 1
Run Backup job
JobName:   harp
FileSet:   harp
Level:     Full
Client:    harp
Storage:   SureStore
Pool:      SSFullPool
When:      2006-07-08 13:14:24
Priority:  10
OK to run? (yes/mod/no): yes
```

```
Run command submitted.
```

Depois de a tarefa de backup ser enviada com sucesso para o diretor, você pode monitorar seu status com o comando **status** do console. Você também pode utilizar o comando **messages** para obter atualizações detalhadas à medida que elas chegarem. Dependendo de como você configurou os recursos `Message` do sistema, um relatório de resumo detalhado também pode ser enviado para o administrador do Bacula. A saída a seguir foi produzida pelo comando **messages** depois de a tarefa de backup ser realizada com sucesso. Ela inclui muitas informações de resumo úteis.

```
* messages
08-Jul-2006 14:21 bull-dir: Start Backup JobId 5216, Job=harp.2006-07-08_14.21.03
08-Jul-2006 14:23 bull-sd: 3301 Issuing autochanger "loaded drive 0" command.
08-Jul-2006 14:23 bull-sd: 3302 Autochanger "loaded drive 0", result is Slot 6.
08-Jul-2006 14:23 bull-sd: 3303 Issuing autochanger "unload slot 6, drive 0" command.
08-Jul-2006 14:24 bull-sd: 3304 Issuing autochanger "load slot 14, drive 0" command.
08-Jul-2006 14:25 bull-sd: 3305 Autochanger "load slot 14, drive 0", status is OK.
08-Jul-2006 14:25 bull-sd: Volume "003048L1" previously written, moving to end of data.
08-Jul-2006 14:25 bull-sd: Ready to append to end of Volume "003048L1" at file=7.
08-Jul-2006 14:54 bull-dir: Bacula 1.34.6 (28Jul04): 08-Jul-2006 14:54
JobId:                        5216
Job:                          harp.2006-07-08_14.21.03
Backup Level:                 Full
Client:                       harp
FileSet:                      "harp" 2006-01-06 20:37:06
Start time:                   08-Jul-2006 14:21
End time:                     08-Jul-2006 14:54
FD Files Written:             176,451
SD Files Written:             176,451
FD Bytes Written:             12,968,821,897
SD Bytes Written:             12,993,427,904
Rate:                         6379.2 KB/sSoftware Compression:None
Volume name(s):               003048L1
Volume Session Id:            263
Volume Session Time:          1149888981
Last Volume Bytes:            13,298,706,752
Non-fatal FD errors:          0
SD Errors: 0
FD termination status:        OK
SD termination status:        OK
Termination:                  Backup OK
08-Jul-2006 14:54 bull-dir: Begin pruning Jobs.
08-Jul-2006 14:54 bull-dir: No Jobs found to prune.
08-Jul-2006 14:54 bull-dir: Begin pruning Files.
08-Jul-2006 14:55 bull-dir: Pruned 2,206 Files from 48 Jobs for client harp from catalog.
08-Jul-2006 14:55 bull-dir: End auto prune.
```

Executando uma tarefa de restauração

Para restaurar arquivos, inicie o console e execute o comando **restore**. Como **run**, **restore** é baseado em menu. Ele começa ajudando você a identificar que tarefas precisam ser lidas para restaurar os arquivos de destino. **restore** apresenta vários métodos de especificar os IDs de tarefas relevantes. Assim que selecionar um conjunto de tarefas, você pode então selecionar os arquivos dessas tarefas para fazer a restauração.

```
* restore
To select the JobIds, you have the following choices:
    1: List last 20 Jobs run
    2: List Jobs where a given File is saved
    3: Enter list of comma separated JobIds to select
    4: Enter SQL list command
    5: Select the most recent backup for a client
    6: Select backup for a client before a specified time
    7: Enter a list of files to restore
    8: Enter a list of files to restore before a specified time
    9: Cancel
```

```
Select item:  (1-9):
```

As duas consultas mais úteis são provavelmente "Select the most recent backup for a client" (#5) e "List Jobs where a given File is saved" (#2). ("Selecione o backup mais recente de um cliente" (#5) e "Liste tarefas nas quais um determinado arquivo é salvo" (#2)). A última opção é útil para aqueles usuários irritantes que nunca parecem se lembrar do local exato em que o arquivo que eles removeram acidentalmente reside de fato. Outro comando poderoso é a opção #4, "Enter SQL list command", que permite inserir qualquer consulta SQL adequadamente formatada.

Suponha que um usuário precise de uma cópia do script **pw_expire.pl** restaurado por volta de abril de 2006; entretanto, ele não está seguro sobre qual máquina ele utilizava para desenvolvimento naquela época. Além disso, ele gostaria dos arquivos restaurados para o diretório **/tmp** da máquina original. Uma solicitação como essa teria levado muitos administradores de sistema a resmungar no bebedouro, mas, para o administrador do Bacula, isso é feito num piscar de olhos. (Infelizmente, o formato do Bacula para pesquisar resultados é tão grande que temos de diminuir a proporções quase invisíveis.)

```
* restore
To select the JobIds, you have the following choices:
    1: List last 20 Jobs run
    2: List Jobs where a given File is saved
    3: Enter list of comma separated JobIds to select
...
    9: Cancel
Select item:  (1-9): 2
Enter Filename: pw_expire.pl

+-------+--------+-------------------------------+---------------------+---------+----------+----------------+
| JobId | Client | Name                          | StartTime           | JobType | JobFiles |    JobBytes    |
+-------+--------+-------------------------------+---------------------+---------+----------+----------------+
| 4,667 | harp   | /home/seidel/pw_expire.pl     | 2006-05-09 23:55:21 | B       | 176,072  | 12,664,064,296 |
| 5,220 | bull   | /home/seidel/bin/pw_expire.pl | 2006-07-08 20:15:49 | B       |       14 |        248,386 |
| 5,216 | harp   | /home/seidel/pw_expire.pl     | 2006-07-08 14:21:05 | B       | 176,451  | 12,968,821,897 |
| 4,523 | harp   | /home/seidel/pw_expire.pl     | 2006-04-25 23:43:56 | B       | 167,435  | 12,539,537,882 |
+-------+--------+-------------------------------+---------------------+---------+----------+----------------+
```

A lista do Bacula de instâncias **pw_expire.pl** revela que a versão de abril de 2006 residia no cliente harp e cujo backup havia sido feito como parte da tarefa 4,523. O Bacula então nos retorna ao menu restore, onde podemos utilizar a opção #3 ("Enter list of comma separated JobIds to select") para focalizar essa tarefa específica.

```
Select item:  (1-9): 3
Enter JobId(s), comma separated, to restore: 4523
You have selected the following JobId: 4523
Building directory tree for JobId 4523 ...
1 Job inserted into the tree.
You are now entering file selection mode, where you add and
remove files to be restored. All files are initially added.[9]
Enter "done" to leave this mode.

cwd is: /
$ cd /home/seidel
cwd is: /home/seidel
$ dir
...
-rwxr-xr-x   1 seidel   atrust       321  2005-10-27 11:25:24  /home/seidel/pw_expire.pl
$ mark pw_expire.pl
1 files marked.
$ done
```

Note que mesmo que o Bacula exibisse originalmente o ID de tarefa com uma vírgula (4,523), teríamos de omitir a vírgula quando reinserissemos o ID; caso contrário, o Bacula interpreta 4,523 como os dois IDs de tarefa 4 e 523.

Depois que o Bacula carrega a tarefa especificada, ele inicia um modo de restauração interativo em que você pode navegar para o arquivo e utilizar o comando **mark** para selecioná-lo. Quando você terminar de marcar os arquivos, o comando **done** sai do modo interativo.

---

9 Isso não é verdade. Se você processar a fase de restauração sem selecionar nenhum arquivo, o Bacula lhe diz que os arquivos foram selecionados.

```
Bootstrap records written to /var/bacula/working/restore.bsr
The restore job will require the following Volumes:

   003048L1

1 file selected to be restored.

Defined Clients:
    1: bull
    2: harp
Select the Client (1-2): 2
```

O Bacula então grava um arquivo de bootstrap que ele utilizará para realizar a restauração, exibe os nomes dos volumes de fita que ele requer e lhe solicita a seleção de um cliente para o qual ele deve restaurar os arquivos. Neste exemplo, restauramos novamente o host original harp.

```
Run Restore job
JobName:   RestoreFiles
Bootstrap: /var/Bacula/working/restore.bsr
Where:     /scratch02/restore/
Replace:   always
FileSet:   BullHomes
Client:    harp
Storage:   SureStore
When:      2006-07-08 16:45:20
Priority:  10
OK to run? (yes/mod/no): mod
```

Para essa tarefa em particular, modificamos as configurações padrão. Especificamente, o nome de conjunto de arquivo correto é harp, que é o conjunto de arquivos padrão configurado no arquivo de configuração de daemon de diretor. Além disso, mudamos a localização dessa restauração para **/tmp** de acordo com a solicitação do usuário.

```
Select parameter to modify (1-11): 9
Please enter path prefix for restore (/ for none): /tmp
Run Restore job
JobName:   RestoreFiles
Bootstrap: /var/Bacula/working/restore.bsr
Where:     /tmp
Replace:   never
FileSet:   harp
Client:    harp
Storage:   SureStore
When:      2006-07-08 16:45:20
Priority:  10
OK to run? (yes/mod/no): yes
Run command submitted.
Restore command done.
```

Depois de fazer as alterações, enviamos a tarefa para o diretor, que a executa. Então utilizamos o comando **messages** para visualizar a saída de log da tarefa.

```
* messages
08-Jul-2006 17:06 bull-dir: Start Restore Job RestoreFiles.2006-07-08_17.06.02
08-Jul-2006 17:06 bull-sd: Ready to read from volume "003048L1" on device /dev/nst0.
08-Jul-2006 17:06 bull-sd: Forward spacing to file:block 11:0.
08-Jul-2006 17:08 bull-sd: End of Volume at file 11 on device /dev/nst0, Volume "003048L1"
08-Jul-2006 17:08 bull-sd: End of all volumes.
08-Jul-2006 17:08 harp: -rwxr-xr-x   1 seidel   atrust       321 2005-10-27 11:25:24
    /tmp/home/seidel/pw_expire.pl
08-Jul-2006 17:08 bull-dir: Bacula 1.34.6 (28Jul04): 08-Jul-2006 17:08
JobId:                      5217
Job:                        RestoreFiles.2006-07-08_17.06.02
Client:                     harp
Start time:                 08-Jul-2006 17:06
```

```
End time:                    08-Jul-2006 17:08
Files Expected:              1
Files Restored:              1
Bytes Restored:             321
Rate:                        0.0 KB/s
FD Errors:                   0
FD termination status:       OK
SD termination status:       OK
Termination:                Restore OK

08-Jul-2006 17:08 bull-dir: Begin pruning Jobs.
08-Jul-2006 17:08 bull-dir: No Jobs found to prune.
08-Jul-2006 17:08 bull-dir: Begin pruning Files.
08-Jul-2006 17:08 bull-dir: No Files found to prune.
08-Jul-2006 17:08 bull-dir: End auto prune.
```

Monitorando e depurando configurações do Bacula

Você pode utilizar o comando **status** do console para consultar os vários daemons do Bacula para obter informações. O exemplo a seguir exibe informações sobre o daemon de diretor. A saída inclui informações sobre as próximas tarefas, as tarefas atualmente em execução e as concluídas.

```
* status dir
bull-dir Version: 1.34.6 (28 July 2004) i686-redhat-linux-gnu redhat Enterprise release
Daemon started 09-Jun-06 15:36, 269 Jobs run since started.

Scheduled Jobs:
Level              Type           Scheduled              Name    Volume
=============================================================
Incremental        Backup         08-Jul-06 20:10        harp    003005L1

Running Jobs:
JobId      Level   Name                          Status
=================================================
5216       Full    harp.2006-07-08_14.21.03      is running

Terminated Jobs:
JobId      Level   Files   Bytes          Status  Finished              Name
=================================================================
5205       Incr    204     898,066,578    OK      06-Jul-06 20:36       harp
```

As tarefas do Bacula produzem um relatório que é roteado de acordo com o recurso `Message` da tarefa no arquivo de configuração do daemon de diretor. O relatório inclui informações básicas sobre os volumes utilizados, o tamanho e número de arquivos que foram copiados em backup e todos os erros que podem ter ocorrido. O relatório normalmente fornece informações suficientes para solucionar quaisquer problemas secundários.

Duas questões que parecem surgir com freqüência são daemons de arquivo de cliente que não estão funcionando e daemons de armazenamento que não podem localizar nenhum volume de fita anexável. No exemplo a seguir, o daemon de diretor informa que uma tarefa de backup terminou com um erro fatal porque não pôde se comunicar com o daemon de arquivo no host harp. Esse erro pode ser visto repetidamente no fim do relatório de resumo.

```
bull-dir: Start Backup JobId 5215, Job=harp.2006-07-08_13.19.49
bull-dir: harp.2006-07-08_13.19.49 Error: Bacula 1.34.6 (28Jul04): 08-Jul-2006 13:49

JobId:                    5215
Job:                      harp.2006-07-08_13.19.49
Backup Level:             Full
Client:                   harp
FileSet:                  "harp" 2006-01-06 20:37:06
Start time:               08-Jul-2006 13:19
End time:                 08-Jul-2006 13:49
FD Files Written:         0
SD Files Written:         0
FD Bytes Written:         0
```

```
SD Bytes Written:         0
Rate:                     0.0 KB/s
Software Compression:     None
Volume name(s):
Volume Session Id:        262
Volume Session Time:      1149888981
Last Volume Bytes:        0
Non-fatal FD errors:      0
SD Errors:                0
FD termination status:
SD termination status:    Waiting on FD
Termination:              *** Backup Error ***

bull-dir: harp.2006-07-08_13.19.49 Warning: bnet.c:666 Could not connect to File daemon on
    192.168.7.2:9102. ERR=Connection refused Retrying ...
bull-dir: harp.2006-07-08_13.19.49 Fatal error: bnet.c:672 Unable to connect to File daemon on
    192.168.7.2:9102. ERR=Connection refused
```

O exemplo a seguir mostra o daemon de armazenamento informando que nenhum volume de fita a partir do pool apropriado está disponível para realizar um backup solicitado. Você pode corrigir o problema adicionando um novo volume ao pool ou purgando e reciclando um volume existente. Não há necessidade de reiniciar a tarefa; o Bacula deve continuar a executá-la a menos que você a cancele explicitamente.

```
bull-sd: Job baikal.2006-07-04_20.10.06 waiting. Cannot find any appendable volumes.
Please use the "label"  command to create a new Volume for:
    Storage:       SureStoreDevice
    Media type:    LTO1
    Pool:          SSFullPool
```

Se alguma vez precisar de informações mais detalhadas sobre o que os daemons estão fazendo, você pode fazer os daemons enviarem muitas informações de depuração para o console acrescentando a opção **-d***nnn* ao comando de inicialização. Por exemplo,

```
$ sudo ./bacula start -d100
```

O *nnn* representa o nível de depuração. Valores típicos variam entre 50 e 200. Quanto mais alto o número, mais informações são exibidas. Você também pode ativar a depuração no console com o comando **setdebug**.

Alternativas ao Bacula

Várias outras ferramentas de backup gratuitas ou de shareware estão disponíveis para download. Os seguintes pacotes são particularmente notáveis; todos ainda estão sob desenvolvimento ativo:

- Amanda: um sistema muito popular e comprovado que faz backup de sistemas UNIX e Linux em uma única unidade de fita. Consulte www.amanda.org.
- Mondo Rescue: um utilitário que faz backup de sistemas de Linux em CD-R, DVD-R, fita ou disco rígido. Essa ferramenta é particularmente útil para recuperação "a partir do estado inicial" (*bare-metal*). Leia mais em www.mondorescue.org.
- **rsync**: uma ferramenta gratuita que faz parte de muitas instalações Linux padrão. Ela pode sincronizar arquivos de um computador para o outro e ser executada em conjunção com SSH para fazer transferência segura de dados pela Internet. Consulte a página 353 para ver mais alguma discussão do **rsync**.
- **star**: uma implementação mais rápida de **tar**; **star** é incluído com todas nossas distribuições de exemplo.

9.9 PRODUTOS COMERCIAIS PARA BACKUP

Todos gostaríamos de imaginar o Linux como o único sistema operacional existente no mundo, mas, infelizmente, esse não é o caso. Ao procurar soluções de backup comerciais, você deve considerar a capacidade delas de manipular quaisquer outros sistemas operacionais pelos quais você é responsável pelo backup. Produtos mais contemporâneos abordam questões entre diversas plataformas e permitem incluir estações de trabalho UNIX, Windows e Macintosh no seu esquema de backup Linux. Você também tem de considerar os arrays de armazenamento e servidores de arquivos não Linux.

Os laptops e outras máquinas dos usuários que não estão conectadas consistentemente à rede também devem ser protegidos contra falhas. Ao pesquisar produtos comerciais, talvez você queira perguntar se cada produto é inteligente o bastante para não fazer backups de arquivos idênticos de cada laptop. De quantas cópias de **command.com** você realmente precisa?

Já que acabamos achando o Bacula adequado para o nosso caso, não temos muita experiência com produtos comerciais. Solicitamos as impressões de alguns de nossos companheiros mais afortunados em termos de recursos financeiros em instalações comerciais sobre os sistemas que eles adotam. Seus comentários são reproduzidos aqui.

ADSM/TSM

O produto ADSM foi desenvolvido pela IBM e posteriormente adquirido pela Tivoli. É comercializado hoje como o Tivoli Storage Manager (TSM), embora o produto seja mais uma vez da IBM. O TSM é uma ferramenta para gerenciamento de dados que também trata de backups. Informações adicionais podem ser localizadas em www.ibm.com/software/tivoli.

Prós:

- Pertence à IBM; veio ao mercado para ficar
- Preço atraente e opções de leasing
- Baixa taxa de falhas
- Usa armazenamento de cache em disco; útil para backups de clientes lentos
- Lida com clientes Windows
- Excelente documentação (cobrada à parte)

Contras:

- Interface gráfica com o usuário fraca
- Cada dois arquivos equivalem a 1 K no banco de dados
- O projeto é incremental para sempre

Veritas

Houve uma fusão da Veritas com a Symantec em 2005. Elas vendem soluções de backup para uma variedade de sistemas. Ao visitar o site da Veritas (www.symantec.com), certifique-se de selecionar o produto que é adequado para você.

Prós:

- Interface gráfica com o usuário decente
- Conecta-se diretamente a dispositivos da Network Appliance
- Instalação automática (*push install*) para Linux
- Capaz de gravar fitas em formato GNU **tar**
- Banco de dados centralizado, mas capaz de suportar um sistema de backups distribuído

Contras:

- Alguns bugs
- O sistema de preços é confuso e irritante

Outras alternativas

W. Curtis Preston, autor do livro *UNIX Backup and Recovery*, da O'Reilly, mantinha uma página Web sobre tópicos relacionados a backups (produtos de espelhamento de disco, produtos de sistema de arquivos avançado, produtos de backup de sistema remoto, produtos do tipo "cofre de dados" fora do local da instalação etc.) em www.backupcentral.com. Não está claro se algumas dessas informações estão ainda sendo atualizadas, mas o material existente ainda é bem útil.

9.10 LEITURA RECOMENDADA

PRESTON, W. CURTIS. *Backup & Recovery*. Sebastopol, CA: O'Reilly Media, 2006.

9.11 EXERCÍCIOS

E9.1 Investigue o procedimento para backups utilizado em sua instalação. Que máquina(s) executa(m) os backups? Que tipos de dispositivos de armazenamento são utilizados? Onde as fitas são armazenadas? Sugira melhorias ao sistema atual.

E9.2 Que etapas são necessárias para restaurar arquivos em um sistema que utiliza o Bacula? Como localizar a fita correta?

★ E9.3 Dada a saída a seguir gerada por **df** e **/etc/dumpdates**, identifique as etapas necessárias para realizar as três restaurações solicitadas. Liste suas suposições. Suponha que a data da solicitação da restauração é 18 de janeiro.

saída **df** na máquina khaya.cs.colorado.edu:

```
/dev/hda8    256194      81103       161863      33%    /
/dev/hda1     21929       4918        15879      24%    /boot
/dev/hda6   3571696      24336       336592      41%    /local
/dev/hda10   131734       5797       119135       5%    /tmp
/dev/hda5   1815580    1113348       610004      65%    /usr
/dev/hda7    256194      17013       225953       7%    /var
```

/etc/dumpdates em khaya.cs.colorado.edu:

```
/dev/hda8   2 Tue Jan      17     22:59:23 2006
/dev/hda6   3 Tue Jan      17     22:51:51 2006
/dev/hda7   3 Tue Jan      17     22:50:24 2006
/dev/hda5   9 Tue Jan      17     22:46:25 2006
/dev/hda5   1 Thu Jan      12     22:45:42 2006
/dev/hda7   0 Thu Jan      12     23:14:47 2006
/dev/hda6   1 Thu Jan      12     23:14:32 2006
/dev/hda8   1 Thu Jan      12     23:14:17 2006
/dev/hda6   0 Tue Jan      10     22:47:31 2006
/dev/hda1   1 Sun Jan       8     22:16:05 2006
/dev/hda7   1 Sat Jan       7     22:08:09 2006
/dev/hda1   4 Tue Jan       3     22:51:53 2006
/dev/hda7   2 Sat Dec      24     22:53:52 2005
/dev/hda5   0 Thu Nov       3     22:46:21 2005
/dev/hda1   0 Wed Sep      21     22:46:29 2005
/dev/hda8   0 Wed Aug      24     23:01:24 2005
/dev/hda1   3 Fri Jul      29     22:52:20 2005
/dev/hda6   2 Fri Jul      29     23:01:32 2005
```

a) "Restaurar meu diretório inicial inteiro (**/usr/home/clements**) referente a algum momento nos últimos dias. Parece que perdi toda a base de código do meu principal projeto."

b) "Uumm, acidentalmente dei um **sudo rm -rf /*** na minha máquina khaya. Você poderia, por favor, restaurar todos os sistemas de arquivos a partir dos backups mais recentes?"

c) "Todos os meus arquivos MP3 que eu vinha colecionando a partir da BitTorrent ao longo do último mês foram para o espaço. Eles estavam armazenados em **/tmp/mp3**. Você poderia restaurá-los para mim?"

★ E9.4 Elabore um plano de backup para as seguintes situações. Suponha que cada computador tem um disco de 10 GB e que os diretórios dos usuários são armazenados localmente. Selecione um dispositivo para backup que apresente um equilíbrio entre custo *versus* necessidades de suporte e explique sua motivação. Liste todas as suposições feitas.

a) Um centro de pesquisas com 50 máquinas. Cada máquina armazena uma grande quantidade de dados importantes que se altera com freqüência.

b) Uma pequena empresa de software com 10 máquinas. Os códigos-fonte são armazenados num servidor central com 500 GB de espaço em disco. Os fontes são alterados constantemente ao longo do

dia. Os diretórios iniciais dos usuários não mudam muito. O custo não é uma preocupação e a segurança é de fundamental importância.

c) Uma rede doméstica com duas máquinas. O custo é uma questão importante.

★ E9.5 Elabore uma estratégia de restauração para cada uma das três situações descritas acima.

★ E9.6 Descreva os passos que você seguiria para realizar um **rdump** seguro por meio de um túnel SSH seguro.

★ ★ E9.7 Escreva instruções de configuração para o Bacula que implementem planos de backup que você planejou para o exercício E9.4.

10 Syslog e arquivos de log

Os daemons de sistema, o kernel e os diversos utilitários todos emitem dados que são registrados e eventualmente acabam em nossos discos de tamanho finito. A maioria desses dados tem uma vida útil limitada e precisa ser resumida, compactada, arquivada e por fim descartada.

10.1 DIRETIVAS DE REGISTRO EM LOG

As diretivas de registro em log variam de instalação para instalação. Esquemas comuns incluem os seguintes:

- Jogar fora todos os dados imediatamente
- Reinicializar arquivos de log em intervalos periódicos
- Fazer um rodízio dos arquivos de log, mantendo dados por um período fixo
- Compactar e arquivar logs em fita ou outras mídias permanentes

A escolha correta para a sua instalação dependerá de quanto espaço em disco você tem e do seu grau de preocupação com a segurança. Até mesmo instalações com uma abundância de espaço em disco têm de lidar com o crescimento canceroso dos arquivos de log.

Seja qual for o esquema adotado, você deve automatizar a manutenção de arquivos com **cron**. Consulte o Capítulo 8, *Processos periódicos*, para informações adicionais sobre o **cron**.

Descartando arquivos de log

Não recomendamos jogar fora todas as informações de registro em log. Instalações que estão sujeitas a problemas de segurança rotineiramente descobrem que os arquivos de log fornecem evidência importante de intrusão. Os arquivos de log também são importantes para ajudá-lo a alertar sobre problemas de hardware e de software. Em geral, dada uma quantidade de espaço em disco confortável, você deve manter os dados pelo menos por um mês. No mundo real, pode levar um tempo até que você descubra que sua instalação foi invadida por um hacker e que você precisa revisar os logs. Caso precise retornar ao passado, você poderá recuperar arquivos de log antigos de suas fitas de backup.

Alguns administradores permitem que os arquivos de log aumentem até um tamanho desconcertante para só então reiniciá-los do zero. Esse plano é melhor do que não manter nenhum dado em si, mas não garante que as entradas de log serão retidas por qualquer período de tempo em particular. O uso médio de disco também pode ser maior do que com outros esquemas de gerenciamento.

Em raras ocasiões, talvez se considere que certos arquivos de log têm maior probabilidade de servirem como provas em um tribunal do que a qualquer outro propósito benéfico. Um administrador de sistema nessa situação poderia manter disponível várias semanas de dados em log, mas provavelmente iria assegurar que os arquivos nunca fossem armazenados em uma mídia permanente. Um exemplo: a Microsoft, em mais de um processo recente, foi acusada de políticas de gerenciamento indevidamente destrutivas para arquivos de log e e-mail. O querelante alega que as diretivas de retenção de dados da Microsoft constituem destruição de evidência, apesar do fato de as exclusões (ou pelo menos, as políticas de exclusão) serem anteriores às ações jurídicas particulares. Infelizmente, é muito cedo para dizer como os tribunais acabarão respondendo a essas reivindicações.[1] Nos Estados Unidos, o Sarbanes-Oxley Act instituiu recentemente novos requisitos de manutenção de registros; ver a página 655.

---

1 Vale notar que a exclusão de e-mails ou arquivos de log relevantes, uma vez que um processo legal parece iminente (ou, pior ainda, durante o curso de um processo legal), é decididamente indevida. Somente uma política contínua e explícita de não-retenção tem alguma perspectiva de proteção jurídica.

Rodízio de arquivos de log

É uma prática comum (mas não universal) armazenar as informações de log de cada semana ou de cada mês em um arquivo separado. Arquivos periódicos são mantidos por um período de tempo específico e depois eliminados. Nós temos o costume de dedicar uma partição de disco (**/var/log**) em um servidor central aos arquivos de log.

No fim de cada rodízio, um script ou programa utilitário renomeia cada arquivo a fim de posicionar os dados mais antigos no final da cadeia. Por exemplo, se um arquivo de log se chama **logfile**, as cópias de backup poderiam ser chamadas de **logfile.1**, **logfile.2** e assim por diante. Se você fizer um rodízio a cada semana e mantiver oito semanas de dados, haverá um **logfile.8**, mas não um **logfile.9**. Toda semana, os dados em **logfile.8** são perdidos, uma vez que **logfile.7** os sobrescreve.

Você pode utilizar a compactação de dados para estender o tempo de retenção, quase sem nenhum prejuízo para a conveniência. Você pode executar **zgrep** para pesquisar os arquivos compactados sem desempacotá-los permanentemente.

Suponha que um arquivo necessite de atenção semanal e você queira arquivar seu conteúdo por três semanas (para fazer com que o exemplo seja curto). O script a seguir implementaria uma diretiva de rodízio apropriada:

```
#!/bin/sh
cd /var/log
mv logfile.2 logfile.3
mv logfile.1 logfile.2
mv logfile logfile.1
cat /dev/null > logfile
chmod 600 logfile
```

Informações de posse são importantes para alguns arquivos de log. Talvez seja necessário executar seu script de rodízio a partir de **cron** como o proprietário dos arquivos de log e não como root, ou talvez você precise adicionar um comando **chown** à seqüência.

A maioria das distribuições do Linux (incluindo todos os nossos exemplos) fornece um excelente utilitário de rodízio de logs chamado **logrotate**, que descrevemos a partir da página 148. É muito mais fácil (e mais confiável) do que criar os seus próprios scripts, e vale a pena procurá-lo e instalá-lo caso sua distribuição não o inclua.

Algumas instalações identificam arquivos de log por data, e não pelo número de seqüência; por exemplo, **logfile.terca** ou **logfile.2005.04.01**. Esse sistema é um pouco mais difícil de ser implementado, mas pode valer a pena o esforço caso você consulte com freqüência arquivos de logs antigos. É muito mais fácil configurar em Perl do que em **sh**. Uma variação útil que não requer nenhuma programação é:

```
mv logfile logfile.`date +%Y.%m.%d`
```

Esse esquema também apresenta a vantagem de fazer com que **ls** ordene os arquivos de log cronologicamente. (A opção **-t** para **ls** faz com que ele classifique os arquivos cronologicamente de acordo com a data/hora da última modificação em qualquer diretório, mas seria interessante se esses arquivos se auto-organizassem sem que isso tivesse de ser solicitado.)

Alguns daemons mantêm seus próprios arquivos de log abertos todo o tempo. Devido à maneira como o sistema de arquivos funciona, o script simples mostrado acima não pode ser utilizado com esses daemons. Em vez dos novos dados de log serem adicionados ao **logfile** recriado, os dados de log continuarão a ser acrescentados a **logfile.1**; a referência ativa ao arquivo original persiste mesmo depois de o arquivo ser renomeado. Para instalar um novo arquivo de log, temos de sinalizar isso para o daemon ou então anular o processo e reiniciá-lo. Cada programa tem um comportamento diferente em relação ao registro em log. Consulte o capítulo apropriado neste livro (ou seus manuais) para determinar quais procedimentos são necessários em cada caso.

Eis um exemplo um pouco mais detalhado que utiliza tanto compactação como sinais:

```
#!/bin/sh
cd /var/log
mv logfile.2.gz logfile.3.gz
mv logfile.1.gz logfile.2.gz
mv logfile logfile.1
cat /dev/null > logfile
chmod 600 logfile
kill -sinal pid
gzip logfile.1
```

sinal representa o sinal apropriado para o programa gravar o arquivo de log; *pid* é o seu ID de processo. O sinal pode ser codificado diretamente no script, mas você tem de determinar dinamicamente o PID do daemon, seja lendo um arquivo que o daemon deixou para trás para você (por exemplo, **/var/run/syslog.pid**, descrito a seguir), seja usando a variante **killall** de **kill**, que é capaz de pesquisar o PID para você na tabela de processos.

Por exemplo, o comando:

```
killall -e -HUP syslogd
```

é equivalente a:

```
kill -HUP `cat /var/run/syslogd.pid`
```

Arquivando arquivos de log

Consulte o Capítulo 9 para informações adicionais sobre backups.

A menos que você deseje evitar explicitamente deixar uma trilha em papel, você sempre deve incluir arquivos de log à sua seqüência regular de backup. Pelo fato de conterem informações que são vitais para investigar incidentes de segurança, os arquivos de log devem sofrer backup com a mais alta freqüência que sua agenda de dumps permitir. Os arquivos de log mudam freqüentemente; portanto, podem representar uma porção significativa das informações de sistema que são armazenadas em backups incrementais. Tenha em mente a interação entre sua diretiva de registro em log e sua diretiva de backups ao elaborar ambas.

Além de serem armazenados como parte de backups regulares, os logs também podem ser arquivados em uma série de fitas separadas. Fitas separadas são mais enfadonhas, mas impõem menos trabalho de documentação e não interferirão na sua capacidade de reciclar fitas de dumps. Se você utilizar fitas distintas, sugerimos que utilize o formato **tar** e crie um script para automatizar seu esquema de backup.

10.2 ARQUIVOS DE LOG NO LINUX

Os sistemas UNIX tradicionais são normalmente criticados por suas inconsistências e até mesmo por sua abordagem um tanto bizarra ao processo de registro em log. Felizmente, os sistemas Linux são, em geral, um pouco mais sensatos, embora cada distribuição tenha sua maneira própria de nomear e dividir os arquivos de log. Em sua maioria, os pacotes Linux enviam suas informações de registro em log para arquivos no diretório **/var/log**. Em algumas distribuições, alguns poucos logs também são armazenados em **/var/adm**.

*O formato do arquivo **syslog.conf** é descrito na página 149.*

A maioria dos programas nos dias de hoje na verdade envia suas entradas de log para um sistema central chamado syslog, que é descrito posteriormente neste capítulo. A configuração de syslog padrão geralmente faz um dump da maioria dessas mensagens em algum ponto de **/var/log**. Verifique o arquivo de configuração de syslog, **/etc/syslog.conf**, para descobrir as localizações específicas.

A Tabela 10.1 compila informações sobre alguns dos arquivos de log mais comuns em nossos exemplos de sistema. Especificamente, a tabela lista:

- Os arquivos de log a serem arquivados, resumidos ou truncados
- O programa que cria cada um deles
- Uma indicação de como cada nome de arquivo é especificado
- A freqüência de limpeza que consideramos razoável
- O proprietário requerido e o grupo requerido do arquivo de log
- As distribuições (entre nossos exemplos) que usam o arquivo de log
- Uma descrição do conteúdo do arquivo

Os nomes de arquivos são relativos a **/var/log** a menos que seja indicado o contrário.

O caractere na coluna Onde informa como o arquivo de log é especificado: S para programas que utilizam syslog, F para programas que utilizam um arquivo de configuração, e H se o nome do arquivo é codificado diretamente no código. A coluna Freq oferece nossa sugestão para a freqüência de limpeza. A coluna Distr lista as distribuições às quais a entrada se aplica.

Tabela 10.1 Lista dos arquivos de log

Arquivo	Programa	Onde[a]	Freq[a]	Distr[a]	Conteúdo
auth.log	**su** etc.[b]	S	M	DU	Autorizações
apache2/*	**httpd** versão 2	F	D	SDU	Logs do servidor HTTP Apache (v2)
boot.log	scripts **rc**	F[c]	M	RF	Gera saída a partir dos scripts de inicialização do sistema
boot.msg	kernel	H	–	S	Dump do buffer de mensagens do kernel
cron	**cron**	S	W	RF	Execuções e erros de **cron**

Tabela 10.1 Lista dos arquivos de log (*Continuação*)

Arquivo	Programa	Onde[a]	Freq[a]	Distr[a]	Conteúdo
cups/*	CUPS	C	W	todas	Mensagens relacionadas à impressão (CUPS)
daemon.log	vários	S	W	DU	Todas as mensagens dos recursos de daemon
debug	vários	S	D	DU	Saída de depuração
dmesg	kernel	H	–	RFDU	Dump do buffer de mensagens do kernel
dpkg.log	**dpkg**	F	M	DU	Log de gerenciamento de pacotes
faillog	**login**	H	W	SDU	Tentativas de login malsucedidas
httpd/logs/*	**httpd**	F	D	RF	Logs do servidor HTTP Apache (em **/etc**)
kern.log	kernel	S	W	DU	Todas as mensagens de recursos do kernel
lastlog	**login**	H	–	todas	Último horário de login por usuário (binário)
mail*	relacionado a correio	S	W	todas	Todas as mensagens dos recursos de correio
messages	vários	S	W	todas	Normalmente o arquivo de log do sistema principal
rpmpkgs	**cron.daily/rpm**	H	D	RF	A lista dos pacotes RPM instalados
samba/*	**smbd** etc.	C	W	–	Samba (compartilhamento de arquivos Windows/CIFS)
secure	**sshd**, **sudo** etc.	S	M	RF	Mensagens de autorização privadas
syslog	vários	S	W	DU	Normalmente o arquivo de log do sistema principal
warn	vários	S	W	S	Todas as mensagens no nível de alerta/erro
wtmp	**login**	H	M	todas	Registros de login (binários)
Xorg.*n*.log	**Xorg**	F	W	RFS	Erros de servidor X Windows
yum.log	**yum**	F	M	RF	Log de gerenciamento de pacotes

a. Onde: S=Syslog, H=Definido no código, F=Arquivo de configuração de freqüência
Freq. D=Diária, W=Semanal, M=Mensal
Distr. R = Red Hat Enterprise Linux, F = Fedora, D = Debian, S = SUSE, U = Ubuntu
b. **senha**, **login** e **desligamento** também gravados no log de autorização. Ele está em **/var/adm** nos sistemas Red Hat e Fedora.
c. Na verdade registra em log por meio do syslog, mas o recurso e o nível são configurados em **/etc/initlog.conf**.

Os arquivos de log são geralmente possuídos por root, embora as convenções para a posse e modo dos arquivos de log variem um pouco entre distribuições. Em nossa opinião, logs, na maioria das instalações, devem receber o modo 600 (leitura e gravação somente para o proprietário) porque seu conteúdo é potencialmente útil para hackers. Se seus usuários forem relativamente sofisticados, você poderá se beneficiar da capacidade de revisar os logs e, nesse caso, é razoável relaxar as permissões em um subconjunto dos arquivos de log.

Um outro meio-termo razoável é configurar o proprietário de grupo dos arquivos de log como um grupo que você cria para esse propósito e tornar o grupo de arquivos legível. Você pode adicionar seus administradores de sistemas locais a esse grupo, permitindo que eles revisem os arquivos de log sem utilizar **sudo**. Essa configuração é especialmente útil se sua instalação tiver administradores de sistemas juniores que não têm privilégios **sudo** completos.

No mínimo, os arquivos **secure**, **auth.log** e **sudo.log** devem estar fora dos limites para navegação casual. Nunca conceda permissão de gravação em qualquer arquivo de log a qualquer pessoa, exceto ao proprietário.

Vale a pena notar que muitos dos arquivos de log contidos na Tabela 10.1 são mantidos por syslog, mas a configuração de syslog padrão varia muito entre os sistemas. Com um arquivo **/etc/syslog.conf** mais consistente, os arquivos de log seriam mais semelhantes entre diferentes distribuições do Linux.

Arquivos de log especiais

A maioria dos logs são arquivos de texto em que as linhas são gravadas à medida que ocorrem eventos "interessantes". Entretanto, alguns dos logs apresentados na Tabela 10.1 possuem um contexto muito diferente.

/var/log/wtmp contém um registro dos logins e logouts de usuários, bem como entradas que registram quando o sistema foi reinicializado ou desligado. Trata-se de um arquivo de log relativamente genérico, já que novas entradas são simplesmente adicionadas ao final do arquivo. Entretanto, o arquivo **wtmp** é mantido em um formato binário. Utilize o comando **last** para decodificar as informações. Apesar de seu formato incomum, o arquivo **wtmp** deve sofrer rodízio ou ser truncado como qualquer outro arquivo de log, pois sua tendência natural é a de crescer ilimitadamente.

Consulte a nota de rodapé na página 119 para informações adicionais sobre os arquivos esparsos.

/var/log/lastlog contém informações similares a **/var/log/wtmp**, mas só registra o tempo de login de cada usuário. É um arquivo binário esparso que é indexado por UID. Ele permanecerá menor, caso seus UIDs sejam atribuídos em alguma forma de seqüência numérica, embora isso certamente não seja nada para se perder horas de sono na vida real. **lastlog** não precisa sofrer rodízio, pois seu tamanho permanece constante a menos que novos usuários façam login.

O kernel e o registro em log em tempo de inicialização

O kernel e os scripts de inicialização de sistema apresentam alguns desafios especiais no domínio do registro em log. No caso do kernel, o problema é criar um registro permanente do processo de inicialização e a operação do kernel sem criar dependências para com qualquer particular sistema de arquivos ou organização de sistema de arquivos. No caso dos scripts de inicialização, o desafio é capturar uma narrativa coerente do procedimento de inicialização sem prender permanentemente qualquer um dos daemons de sistema a um arquivo de log de inicialização, interferir no próprio registro em log de qualquer programa ou arruinar os scripts de inicialização com entradas duplicadas ou redirecionamentos de saída.

O registro em log do kernel é alcançado fazendo com que o kernel armazene suas entradas de logs em um buffer interno de tamanho limitado. O buffer é grande o suficiente para acomodar mensagens sobre todas as atividades do kernel em tempo da inicialização. Assim que o sistema tiver inicializado, um processo de usuário acessa o buffer de log do kernel e faz uma disposição final de seu conteúdo. Em geral, as distribuições fazem isso executando o comando **dmesg** e redirecionando sua saída a **/var/log/dmesg** (RHEL, Fedora, Debian e Ubuntu) ou a **/var/log/boot.msg** (SUSE). Esse é o melhor lugar para procurar detalhes sobre o ciclo mais recente de inicialização.

Daí em diante, o registro em log do kernel é tratado por um daemon chamado **klogd**. As funções de **klogd** são, na verdade, um superconjunto daquelas de **dmesg**; além de fazer o dump do log do kernel e sair, ele também pode ler mensagens do buffer do kernel à medida que são geradas e as repassar para um arquivo ou para o syslog. Em uma operação normal, **klogd** executa nesse último modo. O syslog processa as mensagens de acordo com as instruções para o recurso "kern" (elas em geral são enviadas a **/var/log/messages**).

Os scripts de inicialização de nossos exemplos de distribuição não utilizam o flag **-c** de **dmesg** ao fazerem o dump inicial das mensagens de log, de modo que o buffer de mensagens do kernel é lido, mas não reinicializado. Quando **klogd** inicializa, ele localiza o mesmo conjunto de mensagens visto por **dmesg** no buffer e as submete a syslog. Por essa razão, algumas entradas aparecem tanto nos arquivos **dmesg** ou **boot.msg** como em um outro arquivo gerenciado por syslog, por exemplo, **/var/log/messages**.

Um outro problema no registro em log do kernel é o gerenciamento apropriado do console de sistema. À medida que o sistema está inicializando, é importante para todos que a saída vá para o console. Entretanto, depois que o sistema está de pé e funcionando, as mensagens de console podem ser mais uma amolação do que uma ajuda, particularmente se o console for utilizado para logins.

Tanto **dmesg** como **klogd** permitem configurar o nível de registro em log do console do kernel com um flag de linha de comando. Por exemplo:

```
$ sudo dmesg -n 2
```

O nível 7 é o mais prolixo e inclui informações de depuração. O nível 1 inclui somente mensagens de pânico do kernel (os níveis de menor numeração são os mais graves). Todas as mensagens do kernel continuam indo para o buffer central (e para syslog), independentemente de serem encaminhadas para o console ou não.

O kernel fornece alguns arquivos de controle abaixo do diretório **/proc/sys** para permitir que inundações de mensagens de log repetidas não causem um gargalo na origem. Consulte a seção *Ajustando os parâmetros do kernel do Linux*, iniciando na página 600, para informações adicionais sobre o mecanismo geral por meio do qual os parâmetros do kernel são configurados. Os arquivos de controle específicos são **/proc/sys/kernel/printk_ratelimit**, que especificam o número mínimo de segundos que deve transcorrer entre as mensagens do kernel depois que o gargalo foi ativado (padrão 5) e **/proc/sys/kernel/printk_ratelimit_burst**, que especifica quantas mensagens agrupadas têm permissão de passar antes de ativar o gargalo (padrão 10). Esses parâmetros são recomendações; portanto, não garantem absolutamente que um fluxo pesado de mensagens será contido.

O registro em log para os scripts de inicialização de sistema não é, infelizmente, tão bem gerenciado como o registro em log do kernel. O Red Hat Enterprise Linux utiliza um comando **initlog** para capturar a saída dos comandos de inicialização e submete-os ao syslog. Infelizmente, **initlog** precisa ser mencionado explicitamente sempre que um comando é executado, assim as informações vêm junto com alguma complexidade. As mensagens por fim chegam a **/var/log/boot.log**.

Nossos outros sistemas de exemplo não fazem nenhum esforço coerente para capturar um histórico da saída dos scripts de inicialização. Algumas informações são colocadas em log por daemons e comandos individuais, mas grande parte não é registrada em log.

O Fedora anteriormente utilizava o mesmo sistema **initlog** do Red Hat, mas os comandos para submeter entradas de log foram removidos dos arquivos de inicialização. Felizmente, há uma biblioteca central das funções utilitárias, **/etc/init.d/functions**, onde você pode remover o caractere de comentário das linhas de **initlog** para reativá-las.

10.3 LOGROTATE: GERENCIANDO ARQUIVOS DE LOG

O excelente recurso **logrotate** de Erik Troan implementa várias diretivas de gerenciamento de log e é o padrão em todas as nossas distribuições de exemplo.

O arquivo de configuração **logrotate** é formado por uma série de especificações para grupos de arquivos de log a serem gerenciados. Opções que aparecem fora do contexto da especificação de um arquivo de log (como `errors`, `rotate` e `weekly` no exemplo a seguir) aplicam-se a todas as especificações a seguir; elas podem ser sobrepostas dentro da especificação para um determinado arquivo de log e também podem ser especificadas novamente no arquivo para modificar os padrões.

Eis um exemplo um tanto artificial que manipula vários arquivos de log diferentes:

```
# Opções globais
errors errors@book.admin.com
rotate 5
weekly

# Opções e definições de rodízio de arquivos de log
/var/log/messages {
      postrotate
            /bin/kill -HUP `cat /var/run/syslogd.pid`
      endscript
}

/var/log/samba/*.log {
      notifempty
      copytruncate
      sharedscripts
      postrotate
            /bin/kill -HUP `cat /var/lock/samba/*.pid`
      endscript
}
```

Essa configuração faz o rodízio de **/var/log/messages** toda semana. Ela mantém cinco versões do arquivo e notifica **syslogd** cada vez que o arquivo é reinicializado. Os arquivos de log do Samba (dos quais pode haver vários) também sofrem rodízio semanalmente, mas em vez de serem colocados de lado e reinicializados, eles são copiados e depois truncados. Os daemons do Samba recebem sinais HUP somente depois de todos os arquivos de log terem passado por rodízio.

A Tabela 10.2 apresenta as opções mais úteis de **logrotate.conf**.

Tabela 10.2 Opções de **logrotate**

Opção	Significado
`compress`	Compacta todas as versões não concorrentes do arquivo de log.
`daily`, `weekly`, `monthly`	Faz o rodízio de arquivos de log na programação especificada.
`delaycompress`	Compacta todas as versões, exceto a atual e a próxima mais recente.
`endscript`	Marca o final de um script `prerotate` ou `postrotate`.
`errors` *endereçoDeEmail*	Envia e-mails notificando erros para o *endereçoDeEmail* especificado.
`missingok`	Não reclama caso o arquivo de log não exista.
`notifempty`	Não faz o rodízio do arquivo de log se ele estiver vazio.

Opção	Significado
olddir *dir*	Especifica que as versões mais antigas do arquivo de log devem ser colocadas em *dir*.
postrotate	Introduz um script para ser executado depois de o log sofrer rodízio.
prerotate	Introduz um script para ser executado antes de quaisquer modificações serem feitas.
rotate *n*	Inclui *n* versões do log no esquema de rodízio.
sharedscripts	Executa somente scripts uma vez para todo o grupo de logs.
size=***tamanhDoLog***	Faz o rodízio caso o tamanho do arquivo de log seja maior que *tamanhDoLog* (por exemplo, 100 KB, 4 MB).

logrotate normalmente é executado a partir de **cron** uma vez por dia. Seu arquivo de configuração padrão é **//etc/logrotate.conf**, mas vários arquivos de configuração (ou diretórios contendo arquivos de configuração) podem aparecer na linha de comando de **logrotate**. Esse recurso é utilizado para dar um grande efeito aos nossos exemplos de distribuição, que definem o diretório **/etc/logrotate.d** como um lugar padrão para arquivos de configuração **logrotate**. Pacotes de software cientes de **logrotate** (e há muitos) podem inserir instruções de gerenciamento de log como parte dos seus procedimentos de instalação, simplificando significativamente a administração.

Além do **logrotate**, o Debian e o Ubuntu fornecem um programa mais simples chamado **savelog** que gerencia o rodízio para arquivos individuais. Esse programa é mais simples e fácil que **logrotate** e não utiliza (nem necessita) um arquivo de configuração. Alguns pacotes preferem utilizar suas próprias configurações **savelog** em vez do **logrotate**.

10.4 SYSLOG: O REGISTRADOR DE EVENTOS DO SISTEMA

Syslog, originalmente escrito por Eric Allman, é um sistema de registro em log abrangente. Ele possui duas funções importantes: liberar os programadores da mecânica entediante da gravação de arquivos de log e colocar os administradores no controle do processo do registro em log. Antes de syslog, todo programa era livre para praticar sua própria diretiva de registro em log. Administradores de sistemas não tinham nenhum controle sobre quais informações eram preservadas ou onde eram armazenadas.

Syslog é bastante flexível. Ele permite que mensagens sejam ordenadas pela fonte e importância ("nível de gravidade") e direcionadas para uma série de destinos: arquivos de log, terminais de usuários ou até mesmo outras máquinas. A habilidade de syslog de centralizar o registro em log para uma rede é um dos seus mais valiosos recursos.

Alternativas a syslog

Embora o syslog tenha sido por muito tempo o sistema de login dominante para o UNIX e o Linux, várias alternativas foram desenvolvidas como uma tentativa de resolver algumas das suas deficiências. Uma dessas, syslog-ng (syslog, *next generation*), agora é utilizada nos sistemas SUSE por padrão. Do ponto de vista da configuração, ele é bem diferente do syslog padrão e não o descreveremos em detalhes neste livro. Ele está disponível em www.balabit.com se quiser testá-lo em um sistema não-SUSE.

O syslog-ng adiciona recursos adicionais de configuração, filtragem baseada no conteúdo das mensagens, integridade de mensagem e melhor suporte às restrições de firewall quando as mensagens são encaminhadas pela rede.

O SDSC Secure Syslog (do San Diego Supercomputing Center) também é conhecido como syslog de alto desempenho. O syslog-ng fornece um sistema de auditoria "forensicamente perfeito" implementando as especificações da RFC3195. Ele foi projetado tendo em mente instalações de alto tráfego e contém algumas otimizações de desempenho. Você pode fazer o download do código-fonte em sourceforge.net/projects/sdscsyslog.

Arquitetura do syslog

Syslog é formado por três partes:

- **syslogd**, o daemon de registro em log (juntamente com seu arquivo de configuração, **/etc/syslog.conf**)
- **openlog** et al., rotinas de bibliotecas que submetem mensagens a **syslogd**
- **logger**, um comando em nível de usuário que submete entradas de log do shell

Na discussão a seguir, cobrimos primeiro a configuração de **syslogd** e depois mostramos como usar syslog a partir de scripts Perl.

syslogd é acionado em tempo de inicialização e executa continuamente; ele não pode ser gerenciado com **inetd**. Programas cientes de syslog gravam as entradas de log (utilizando a rotina de biblioteca **syslog**) no arquivo especial **/dev/log**, em um soquete de domínio UNIX. **syslogd** lê as mensagens a partir desse arquivo, consulta seu arquivo de configuração e envia cada mensagem ao destino apropriado.

Um sinal de suspensão (HUP, sinal 1) faz **syslogd** fechar seus arquivos de log, reler seu arquivo de configuração e iniciar o registro em log novamente. Se modificar **/etc/syslog.conf**, você terá de enviar um sinal de suspensão para **syslog** fazer com que suas alterações entrem em efeito. Um sinal TERM faz com que **syslogd** se encerre.

syslogd grava o ID do seu processo no arquivo **/var/run/syslogd.pid**. Essa convenção facilita o envio de sinais a **syslogd** a partir de um script. Por exemplo, o comando a seguir envia um sinal de suspensão:

```
# kill -HUP `/bin/cat /var/run/syslogd.pid`
```

Tentar compactar ou fazer rodízio de um arquivo de log que **syslogd** tenha aberto para gravação não é saudável e possui resultados imprevisíveis. O procedimento adequado é deixar o log antigo de lado, recriar o log com as mesmas posses e permissões e então enviar um sinal HUP a **syslogd**. Esse procedimento é facilmente implementando por meio de **logrotate**; veja um exemplo na página 148.

Configurando syslogd

O arquivo de configuração **/etc/syslog.conf** controla o comportamento de **syslogd**. Trata-se de um arquivo de texto com um formato relativamente simples. Linhas em branco e linhas com um sinal # na primeira coluna são ignoradas. O formato básico é:

```
seletor <Tab> ação
```

Por exemplo, a linha:

```
mail.info       /var/log/maillog
```

faria com que as mensagens do sistema de e-mail fossem salvas no arquivo **/var/log/maillog**.

Os seletores identificam o programa ("recurso") que está enviando uma mensagem de log e o nível de gravidade da mensagem com a sintaxe

```
recurso.nivel
```

Tanto os nomes dos recursos como os níveis de gravidade devem ser escolhidos de uma breve lista de valores definidos; programas não podem inventar seus próprios valores. Os recursos são definidos para o kernel, para grupos de utilitários comuns e para programas escritos localmente. Tudo mais é classificado sob o recurso genérico "user".

Os seletores podem conter as palavras-chave especiais `*` e `none`, significando, respectivamente, tudo ou nada. Um seletor pode incluir vários recursos separados por vírgulas. Vários seletores podem ser combinados com ponto-e-vírgulas.

Em geral, os seletores sofrem uma operação OR: uma mensagem que corresponde a um seletor qualquer estará sujeita à *ação* da linha. Entretanto, um seletor com um nível `none` exclui os recursos listados independentemente do que outros seletores na mesma linha possam dizer.

Eis alguns exemplos de maneiras para formatar e combinar seletores:

```
recurso.nivel                          ação
recurso1,recurso2.nivel                ação
recurso1.nivel1;recurso 2.nivel 2      ação
*.nivel                                ação
*.nivel;recursoRuim.none               ação
```

A Tabela 10.3 a seguir lista os nomes válidos dos recursos. Atualmente há 21 recursos.

O próprio **syslogd** produz mensagens de registro de data/hora que são registradas em log se o recurso "mark" aparecer em **syslog.conf** para especificar um destino para elas. Os registros de data/hora podem ajudá-lo a descobrir que sua máquina travou entre 3h00 e 3h20 e não somente "em algum momento durante a última noite". Essas informações podem ser uma grande ajuda ao depurar problemas que talvez ocorram regularmente (por exemplo, "travamentos misteriosos" que ocorrem quando o pessoal da faxina liga os aspiradores de pó tarde da noite, acionando os disjuntores).

Se seu sistema estiver bem ocupado, outras mensagens de log normalmente fornecerão informações cronológicas adequadas. Mas nas primeiras horas da madrugada, esse nem sempre é o caso.

A Tabela 10.4 apresenta os níveis de gravidade do syslog em ordem decrescente de importância.

O grau de gravidade de uma mensagem especifica sua importância. As distinções entre os vários níveis são às vezes confusas. Há uma diferença clara entre um informe e um alerta (e entre um alerta e um erro), mas o significado exato expresso por um alerta em oposição a algo crítico é uma questão de conjetura. A Tabela 10.7 mais à frente lista os níveis específicos utilizados por vários programas de software comuns.

Tabela 10.3 Nomes de recursos de **syslog**

Recurso	Programas que o utilizam
*	Todos os recursos, exceto "mark"
auth	Comandos relacionados com segurança e autorização
authpriv	Mensagens de autorização sigilosas/privadas [a]
cron	O daemon de **cron**
daemon	Daemons de sistema
ftp	O daemon de FTP, **ftpd**
kern	O kernel
local0-7	Oito variantes de mensagem local
lpr	O sistema de spooling de impressora de linha
mail	**sendmail** e outros programas relacionados com e-mail
mark	Registros de data/hora gerados em intervalos regulares
news	O sistema de grupos de discussão da Usenet
syslog	Mensagens internas de **syslogd**
user	Processos de usuário (o padrão se não especificado)
uucp	Obsoleto, ignore

a. Na realidade, todas as mensagens relacionadas a autorizações são sigilosas. Nem as mensagens authpriv nem as mensagens auth devem ser legíveis por todo mundo.

Tabela 10.4 Níveis de gravidade de syslog (por ordem decrescente de gravidade)

Nível	Significado aproximado
emerg	Situações de pânico
alert	Situações urgentes
crit	Condições críticas
err	Outras condições de erros
warning	Mensagens de alerta
notice	Coisas que podem merecer investigação
info	Mensagens informativas
debug	Somente para depuração

No arquivo **syslog.conf**, os níveis indicam a importância *mínima* que uma mensagem deve ter para merecer ser registrada em log. Por exemplo, uma mensagem do sistema de e-mail no nível de alerta corresponderia ao seletor `mail.warning`, bem como aos seletores `mail.notice`, `mail.info`, `mail.debug`, `*.warning`, `*.notice`, `*.info` e `*.debug`. Se **syslog.conf** especifica que mensagens `mail.info` devem ser registradas em um arquivo de log, então mensagens `mail.warning` também o serão.

Como refinamento da sintaxe básica, a versão Linux de syslog também permite que os caracteres = e ! sejam antecedidos aos níveis de prioridade para indicarem "somente essa prioridade" e "exceto essa prioridade e maiores", respectivamente. A Tabela 10.5 mostra alguns exemplos.

Tabela 10.5 Exemplos de qualificadores de nível de prioridade do Linux em syslog.conf.

Seletor	Significado
`mail.info`	Mensagens relacionadas a correio no nível de prioridade de informações e mais alto
`mail.=info`	Somente as mensagens no nível de prioridade de informações
`mail.info;mail.!err`	Somente informações, avisos e alertas de prioridades
`mail.debug;mail.!=warning`	Todas as prioridades, exceto alerta

O campo *ação* informa o que fazer com uma mensagem. As opções são listadas na Tabela 10.6.

Tabela 10.6 Ações de syslog

Ação	Significado
nomeDoArquivo	Anexa a mensagem a um arquivo na máquina local
@nomeDoHost	Encaminha a mensagem para o **syslogd** em *nomeDoHost*
@endereçoIP	Encaminha a mensagem ao **syslogd** no host *endereçoIP*
| nomeFifo	Grava a mensagem no pipe identificado *nomeFifo*[a]
usuário1,usuário2,...	Grava a mensagem na tela dos usuários se eles estiverem conectados
*	Escreve a mensagem para todos os usuários que estiverem conectados no momento

a. Veja **info mkfifo** para informações adicionais.

Se uma ação *nomeDoArquivo* (ou *nomeFifo*) for utilizada, o nome deve ser um caminho absoluto. Se especificarmos um nome de arquivo inexistente, **syslogd** criará o arquivo quando uma mensagem for direcionada pela primeira vez a ele.[2] Você pode anteceder uma ação *nomeDoArquivo* com um traço para indicar que o sistema de arquivos não deve ser sincronizado depois de cada entrada de log ter sido registrada. **sync** ajuda a preservar o máximo possível de informações de log no caso de uma pane, mas, para arquivos de log intensamente utilizados, é caro em termos de desempenho de disco.

Se for utilizado um *nomeDoHost* no lugar de um endereço IP, ele tem de ser convertido por meio de um mecanismo de tradução como DNS ou NIS.

Embora possamos utilizar vários recursos e níveis em um seletor, não há nenhuma provisão para múltiplas ações. Para enviarmos uma mensagem a dois lugares (como, por exemplo, para um arquivo local e para um host de registro em log central), podemos incluir no arquivo de configuração duas linhas com os mesmos seletores.

Pelo fato de as mensagens de syslog poderem ser utilizadas para montar um ataque de negação de serviço, o **syslogd** do Red Hat não aceitará mensagens de log de outras máquinas a menos que seja iniciado com o flag **-r**. Como padrão, o **syslogd** Red Hat também se negação a atuar como um encaminhador de mensagens de terceiros; mensagens provenientes de um host de rede não podem ser encaminhadas para um outro. Utilize o flag **-h** para sobrescrever esse comportamento. (Se quiser que essas opções permaneçam ativas o tempo todo, adicione os flags no script **/etc/sysconfig/syslog** para RHEL e Fedora ou em **/etc/init.d/sysklogd** para Debian e Ubuntu.)

Projetando um esquema de registro em log para sua instalação

Em uma instalação pequena é adequado configurar o registro em log de modo que erros de sistema e alertas importantes sejam mantidos em um arquivo em cada máquina, de maneira muito parecida com o que era feito antes de termos o syslog. O arquivo **syslog.conf** pode ser personalizado para cada host.

Em uma grande rede, o registro em log centralizado é fundamental. Ele mantém o fluxo de informações de uma forma controlada e torna os dados de auditoria indisponíveis para uma pessoa que viola a segurança de uma máquina da rede. Os hackers normalmente editam logs de sistema para encobrir seus vestígios; se as informações de log forem remetidas tão logo sejam geradas, será muito mais difícil destruí-las. Seu firewall não deveria permitir que instalações externas submetessem mensagens ao seu **syslogd**.

Esteja ciente de que qualquer um pode chamar **syslog** e falsificar entradas de log por meio de qualquer daemon ou utilitário. Syslog também utiliza o protocolo UDP, que não é garantidamente confiável; mensagens podem ser perdidas.

Consulte o Capítulo 17 para informações adicionais sobre como distribuir arquivos por uma rede.

Selecione uma máquina estável como seu servidor de registro em log, preferencialmente uma que esteja bem segura e não tenha muitos logins. Outras máquinas podem utilizar um arquivo de configuração syslog genérico que é mantido em um local centralizado. Portanto, somente duas versões do **syslog.conf** precisam ser mantidas. Essa abordagem permite que o login seja completo sem ser um pesadelo de administração.

Para segurança máxima, o servidor de syslog deve ser protegido por firewall do restante da rede, permitindo conexões somente na porta do syslog e somente a partir dos hosts que têm permissão para fazer registro em log nele, e nada mais. Dependendo do nível de paranóia existente no ambiente, conexões SSH podem ser permitidas a partir das estações de trabalho dos administradores de sistemas para tornar mais fácil para eles revisarem os logs.

Algumas instalações muito grandes podem querer adicionar mais níveis à hierarquia de registro em log. Infelizmente, o syslog retém o nome do host originário para somente um hop. Se o host "cliente" enviar algumas entradas de log ao host "servidor", que as envia ao host "mestre", o mestre interpretará os dados como provenientes do servidor, e não do cliente.

2 Note que esse comportamento é o contrário daquele da implementação original de syslog, que exigia que arquivos de log fossem criados antecipadamente.

Exemplos de arquivos de configuração

A configuração de syslog é uma área em que as distribuições Linux variam muito. Já que é relativamente fácil lermos um arquivo **syslog.conf**, não revisaremos detalhadamente os arquivos de configuração de nossos exemplos de distribuição; eles são bem diretos. Em vez disso, veremos algumas maneiras que poderíamos utilizar para configurar o registro em log caso optemos por partir do padrão do sistema ou expandi-lo.

A seguir apresentamos três exemplos de arquivos de configuração **syslog.conf** que correspondem, respectivamente, a uma máquina dedicada em uma pequena rede, uma máquina cliente em uma rede maior e um host de registro em log central na mesma rede grande. O host de registro em log central é chamado de "netloghost".[3]

Máquina isolada

Uma configuração básica para uma máquina isolada é apresentada a seguir:

```
# Arquivo syslog.conf para uma rede pequena ou máquinas isoladas

# Emergências: informa a todo mundo quem está conectado
*.emerg                                *
#  mensagens importantes
*.warning;daemon,auth.info;user.none    /var/log/messages
#  erros de impressora
lpr.debug                              /var/log/lpd-errs
```

A primeira linha não comentada envia mensagens de emergência à tela de todos os usuários atuais. Um exemplo de mensagens de nível de emergência é aquela gerada por **shutdown** quando o sistema está prestes a ser desligado.

A segunda linha envia mensagens importantes para **/var/log/messages**. O nível de informações está abaixo do nível de alerta; portanto, a instrução `daemon,auth.info` inclui registro em log adicional de **passwd**, **su** e programas daemon. A terceira linha envia mensagens de erro de impressão para **/var/log/lpd-errs**.

Cliente de rede

Em geral, um cliente de rede encaminha mensagens críticas para uma máquina central de registro em log:

```
# Arquivo syslog.conf para máquinas não mestres

# Emergências: informa a todo mundo quem está conectado
*.emerg;user.none                          *

# Encaminha mensagens importantes para o registrador de log central
*.warning;lpr,local1.none                  @netloghost
daemon,auth.info                           @netloghost

# Envia algum material local para o registrador de log central também
local2.info;local7.debug                   @netloghost

# Armazena erros de impressora localmente
lpr.debug                                  /var/log/lpd-errs

# sudo registra seu log em local2 - armazena uma cópia aqui também
local2.info                                /var/log/sudo.log

# Armazena mensagens do kernel localmente
kern.info                                  /var/log/kern.log
```

Essa configuração não mantém localmente muita informação de log. Vale a pena mencionar que se netloghost estiver inativo ou não puder ser alcançado, as mensagens de log serão irremediavelmente perdidas. Talvez você queira manter duplicatas locais de mensagens importantes para se precaver contra essa possibilidade.

Em uma instalação com software local instalado, um grande número de mensagens pode ser registrado indevidamente no recurso usuário, no nível de emergência. Neste exemplo, user/emerg foi excluído especificamente com a instrução `user.none` da primeira linha.

---

3. Mais precisamente, ele usa "netloghost" como um de seus aliases para nome de host. Isso permite que a identidade do host de logs seja modificada com um nível menor de reconfiguração. Um alias pode ser adicionado em **/etc/hosts** ou configurado com um registro em log CNAME em DNS. Consulte a página 218 para informações adicionais sobre registros DNS CNAME.

Consulte a página 35 para informações adicionais sobre **sudo**.

A segunda e terceira linhas encaminham todas as mensagens importantes para o host de registro em log central; mensagens do sistema de impressão e do sistema de acesso do campus baseado em cartão magnético englobando toda a instalação (local1) são excluídas explicitamente. A quarta linha também encaminha um subconjunto das informações de registro em log local a netloghost. As últimas três entradas mantêm cópias locais dos erros de impressora, mensagens de **sudo** e mensagens do kernel.

Host de registro em log central

Esse exemplo é para netloghost, o host de registro em log central e seguro para uma rede de tamanho razoável com cerca de 7.000 hosts.

```
# Arquivo syslog.conf para o host de registro em log master

# Emergências para o console e arquivo de log, com marcas cronológicas
*.emerg                                    /dev/console
*.err;kern,mark.debug;auth.notice          /dev/console
*.err;kern,mark.debug;user.none            /var/log/console.log
auth.notice                                /var/admlog/console.log

# Envia mensagens não-emergenciasi nos arquivos de log usuais
*.err;user.none;kern.debug                 /var/log/messages
daemon,auth.notice;mail.crit               /var/log/messages
lpr.debug                                  /var/log/lpd-errs
mail.debug                                 /var/log/mail.log

# Material de autorização local como sudo e npasswd
local2.debug                               /var/log/sudo.log
local2.alert                               /var/log/sudo-errs.log
auth.info                                  /var/log/auth.log

# Outro material local
local4.notice                              /var/log/da.log
local7.debug                               /var/log/tcp.log

# Material de usuário (o padrão se nenhum recurso for especificado)
user.info                                  /var/log/user.log
```

As mensagens provenientes dos programas locais e dos **syslogds** na rede são gravadas nos arquivos de log. Em alguns casos, a saída de cada recurso é colocada em seu próprio arquivo.

O host de registro em log central gera o registro com data/hora para cada mensagem quando ele grava a mensagem. Os registros de data/hora não refletem o horário no host originário. Se você tiver máquinas em vários fusos horários ou se os relógios de seu sistema não estiverem sincronizados, os registros de data/hora poderão ser um pouco incorretos.

Exemplo de saída de syslog

A seguir mostramos um pequeno trecho de um dos arquivos de log do host de syslog mestre do Departamento de Computação da Universidade do Colorado.

```
Dec 18 15:12:42 av18.cs.colorado.edu sbatchd[495]: sbatchd/main: ls_info() failed: LIM is down; try
    later; trying …
Dec 18 15:14:28 proxy-1.cs.colorado.edu pop-proxy[27283]: Connection from 128.138.198.84
Dec 18 15:14:30 mroe.cs.colorado.edu pingem[271]: maltese-office.cs.colorado.edu has not answered
    42 times
Dec 18 15:15:05 schwarz.cs.colorado.edu vmunix: Multiple softerrors: Seen 100 Corrected Softerrors
    from SIMM J0201
Dec 18 15:15:16 coyote.cs.colorado.edu PAM_unix[17405]: (sshd) session closed for user trent
Dec 18 15:15:48 proxy-1.cs.colorado.edu pop-proxy[27285]: Connection from 12.2.209.183
Dec 18 15:15:50 av18.cs.colorado.edu last message repeated 100 times
```

Esse exemplo contém entradas de vários hosts distintos (av18, proxy-1, schwarz, mroe e coyote) e de diversos programas: **sbatchd**, **pop-proxy**, **pingem** e da biblioteca Pluggable Authentication Modules.

Note a última linha do trecho, que reclama de uma mensagem que foi repetida 100 vezes. Para ajudar a manter os logs menores, syslog geralmente tenta fundir mensagens duplicadas e substituí-las por esse tipo de resumo. Entretanto, a máquina da qual esse exemplo foi extraído aceita entradas de muitos outros hosts, de modo que essa mensagem em particular é

um tanto enganadora. Ela se refere, na verdade, à entrada de log anterior a av18, não à entrada que a precede imediatamente no log composto.

É uma boa idéia percorrer seus arquivos de log regularmente. Determine o que é normal, de tal modo que quando ocorrer uma anomalia, você possa reconhecê-la. Melhor ainda, configure um pós-processador de logs como **swatch** para capturar esses casos automaticamente; veja a página 157.

Software que utiliza syslog

A Tabela 10.7 apresenta alguns dos programas que utilizam syslog, os recursos e níveis em que eles fazem o registro em log e uma breve descrição de cada programa.

Com todas essas informações, deve ficar perfeitamente claro quais mensagens manter e quais descartar, não? Bem, pode ser que não. Na prática, temos de saber somente quais são os níveis de registro em log úteis para o sistema. É melhor começar com um nível de registro em log excessivo e, gradualmente, separar o joio do trigo, ou seja, eliminar os casos que não desejamos. Pare de eliminar quando se sentir confortável com a taxa de dados média.

Depurando o syslog

O comando **logger** é útil para submetermos entradas de log a partir de scripts de shell. Você também pode utilizá-lo para testar alterações no arquivo de configuração de **syslogd**. Por exemplo, se você tiver adicionado a linha:

```
local5.warning          /tmp/evi.log
```

e quiser verificar se ela está funcionando, execute

```
$ logger -p local5.warning "test message"
```

Tabela 10.7 Programas que utilizam syslog

Programa	Recurso	Níveis	Descrição
cron	cron, daemon	info	Daemon de agendamento de tarefas
cups	lpr	info-err	Sistema comum de impressão UNIX
date	auth	notice	Configura a data e a hora
ftpd	daemon	depur–err	Daemon FTP
ftpd	ftp	debug-crit	Daemon FTP (**wu-ftpd**)
gated	daemon	info-alert	Daemon de roteamento
halt/reboot	auth	crit	Programas de desativação
inetd	daemon	alerta, erro	Super-daemon de Internet (Debian)
imapd	mail	info-alert	Servidor de e-mail IMAP
login/rlogind	auth	info–crit	Programas de login
login	authpriv	info-err	Programas de login
lpd	lpr	info-err	Sistema de impressão BSD
named	daemon	info-err	Servidor de nomes (DNS)
ntpd	daemon, usuário	info–crit	Daemon de tempo de rede
passwd	auth	notice, warning	Programa de configuração de senhas
popper	local0	debug, notice	Servidor de e-mail POP3
sendmail	mail	debug-alert	Sistema de transporte de e-mails
shutdown	auth	notice	Pára o sistema
ssh	auth	info	Shell seguro (logins remotos)
su	auth	aviso, crit	Comuta UIDs
sudo	local2	notice, alert	Programa **su** limitado
syslogd	syslog, mark	info-err	Erros internos, registros de data/hora
tcpd	local7	debug-err	Empacotador TCP para **inetd**
vmlinuz	kern	todas	O kernel
xinetd	*configurável*	info (padrão)	Variante de **inetd** (Red Hat, SUSE)

Uma linha contendo "test message" deve ser gravada em **/tmp/evi.log**. Se isso não acontecer, talvez você tenha se esquecido de enviar a **syslogd** um sinal de hangup.

Tenha cuidado ao fazer registros em log no dispositivo de console, **/dev/console**, ou em qualquer porta ou pseudoterminal que suporta controle de fluxo. Se alguém pressionar <Control-S> no console, a saída para ele será interrompida. Cada chamada a **syslog** será bloqueada e seu sistema ficará lento como uma lesma. Uma boa maneira de verificar essa condição degenerada é enviar uma mensagem de syslog para o console com **logger**. Se **logger** travar, você precisará descobrir qual é a porta problemática, dar um <Control-Q> e repensar sua estratégia de registro em log.

Outro inconveniente de se fazer o registro em log no console é que o fluxo de mensagens provocado por um grande problema pode tornar o console inutilizável precisamente no momento em que ele é mais necessário. Com certos tipos de drivers de console não-otimizados, na verdade a inundação de mensagens pode tornar todo o *sistema* inutilizável.

Dependendo de como o console é configurado e gerenciado (por exemplo, por meio de um servidor de console), o registro em log em console também pode apresentar algumas implicações em termos de segurança.

Utilizando syslog a partir de programas

As rotinas de biblioteca **openlog**, **syslog** e **closelog** permitem que os programas utilizem o sistema de syslog. Versões dessas rotinas de biblioteca encontram-se disponíveis para C, Perl, Python e PHP; descrevemos aqui somente a interface para Perl.

Para importar as definições das rotinas de biblioteca, inclua a seguinte linha:

```
use Sys::Syslog;
```

no início de seu script Perl.

A rotina **openlog** inicializa o registro em log, usando o nome do recurso especificado:

```
openlog(ident, opçãoDeLog, recurso);
```

As mensagens são registradas em log com as opções especificadas por *opçãoDeLog* e começam com a string de identificação *ident*. Se **openlog** não for utilizado, *ident* utiliza por padrão o nome do usuário atual, *opçãoDeLog* utiliza uma string vazia e *recurso* utiliza "user". A string *opçãoDeLog* deve conter uma lista separada por vírgulas das opções demonstradas na Tabela 10.8.

Tabela 10.8 Opções de registro em log para a rotina de **openlog**

Opção	Significado
`pid`	Inclui o PID do processo atual em cada mensagem de log
`ndelay`	Conecta-se imediatamente a **syslogd** (não espera uma mensagem ser enviada)
`cons`	Envia mensagens ao console do sistema caso **syslogd** não possa ser alcançado
`nowait`	Não espera, **wait**(3), processos-filho bifurcados enviarem mensagens para o console

Por exemplo, uma chamada razoável a **openlog** poderia ser a seguinte:

```
openlog("adminscript", "pid,cons", "local4");
```

A rotina **syslog** envia uma *mensagem* a **syslogd**, que a registra em log com a *prioridade* especificada:

```
syslog(prioridade, mensagem, ...);
```

A data, o horário, o nome do host e a string *ident* da chamada **openlog** são anexados à mensagem no arquivo de log. *mensagem* pode ser seguida por vários outros parâmetros para formar uma especificação de saída no estilo **printf** que pode incluir texto e o conteúdo de outras variáveis, por exemplo:

```
syslog("info", "Delivery to '%s' failed after %d attempts.", $user, $nAttempts);
```

O símbolo especial `%m` expande-se para uma mensagem de erro derivada do valor atual de **errno** (o código de erro mais recente).

Uma string de prioridade na forma "*nível* | *recurso*" configura o nível de gravidade e o nome do recurso. Se não chamarmos **openlog** e especificarmos uma string *ident*, a rotina **syslog** também verificará se nossa *mensagem* possui o formato de uma mensagem de erro padrão como:

```
adminscript: User "nobody" not found in /etc/passwd file.
```

Se ela tiver, a parte anterior aos dois-pontos será adotada secretamente como nossa string *ident*. Esses recursos úteis (mas não-documentados) tornam desnecessário chamar **openlog** em si; porém, mesmo assim é uma boa idéia fazê-lo. É melhor especificar o nome do recurso em um lugar (a chamada **openlog**) do que repeti-lo ao longo de todo o código.

A rotina **closelog** fecha o canal de registro em log:

```
closelog (   );
```

Você tem de chamar essa rotina se quiser reabrir o canal de registro em log com opções diferentes. É recomendável chamar **closelog** quando nosso programa se encerra, mas fazer isso não é estritamente necessário.

Eis um exemplo completo:

```
use Sys::Syslog;

openlog("adminscript", "cons,pid", "user");
syslog("warning", "Those whom the gods would destroy, they first teach Basic.");
closelog();
```

Esse pequeno script gera a seguinte entrada de log (191 é o PID de **adminscript**):

```
Dec 28 22:56:24 moet.colorado.edu adminscript[191]: Those whom the gods would destroy, they first
    teach Basic.
```

10.5 CONDENSANDO ARQUIVOS DE LOG EM INFORMAÇÕES ÚTEIS

Syslog é excelente para ordenar e encaminhar mensagens de log, mas quando tudo estiver pronto, seu produto final ainda será um monte de arquivos de log. Embora eles possam conter todo tipo de informação útil, esses arquivos não irão procurar você e informá-lo quando algo der errado. Uma outra camada de software é necessária para analisar os logs e garantir que as mensagens importantes não fiquem perdidas no meio de todo aquele palavrório.

Uma série de ferramentas gratuitas está disponível para preencher esse nicho, e a maioria delas é bastante semelhante: elas varrem entradas de log recentes, as comparam com um banco de dados de expressões regulares e processam as mensagens importantes de alguma maneira para chamar a atenção. Algumas ferramentas enviam um relatório via e-mail para nós; outras podem ser configuradas para alertar-nos de forma auditiva, para imprimirem entradas de log em cores diferentes ou para enviarem mensagens via pager. As ferramentas diferem basicamente no grau de flexibilidade e no tamanho de seus bancos de dados de padrões.

Dois dos pós-processadores de log mais comumente utilizados são **swatch** de Todd Atkins e **logcheck** de Craig Rowland. Ambos estão disponíveis em sourceforge.net (**logcheck** vem com o pacote sentrytools: sourceforge.net/projects/sentrytools).

swatch é um script em Perl que recebe suas ordens a partir de um arquivo de configuração. A sintaxe de configuração é relativamente flexível e também fornece acesso aos "poderes mágicos" de comparação de padrões do Perl. Embora **swatch** possa processar um arquivo inteiro de uma única vez, seu principal objetivo é permanecer em execução de modo que possa revisar novas mensagens à medida que elas chegam, *a la* **tail -f**. Uma desvantagem do **swatch** é que você precisa criar sua própria configuração quase a partir do zero; ele não conhece sistemas específicos e as mensagens reais de log que eles talvez gerem.

logcheck é um script mais básico escrito em **sh**. A distribuição também inclui um programa em C que **logcheck** utiliza para ajudar a registrar seu lugar dentro de um arquivo de log. **logcheck** sabe até que ponto ele leu em um arquivo de log, portanto talvez haja menor probabilidade de uma mensagem ser recebida em tempo de inicialização ou em tempo de desligamento. Além disso, **logcheck** pode executar em intervalos a partir do **cron** em vez de ser executado continuamente.

logcheck vem acompanhado de modelos de bancos de dados para várias versões do UNIX e do Linux. Mesmo que não queira utilizar o script em si, vale a pena dar uma olhada nos padrões para ver se há algum que possa lhe servir.

Ambas as ferramentas possuem a desvantagem de operarem somente em um único arquivo de log por vez. Se a sua configuração de syslog organiza as mensagens em muitos arquivos diferentes, talvez seja recomendável duplicar algumas das mensagens em um arquivo centralizado que seja freqüentemente truncado, e, então, usar esse arquivo resumido para alimentar um script de pós-processamento. É mais fácil do que configurar uma rede de scripts complicados para tratar vários arquivos.

Uma outra ferramenta que vale mencionar é **logwatch** de Kirk Bauer. Ele é mais como um "resumidor" de log do que uma ferramenta de monitoração e alerta, mas tem as vantagens de ser relativamente simples, sendo instalado por padrão no Fedora e no Red Hat e está disponível como um pacote RPM e como um pacote APT.

Um tipo diferente de ferramenta de gerenciamento de log é o SEC, Simple Event Correlator. Ele é um script Perl que lê linhas de arquivos, pipes identificados ou entrada padrão e as converte em várias classes de "eventos de entrada" comparando-as com as expressões regulares. As regras de configuração especificam então como os eventos de entrada devem ser convertidos em eventos de saída, por exemplo, a execução de um script particular ou a emissão de uma mensagem para um pipe ou arquivo especificado.

A distribuição SEC está disponível em kodu.neti.ee/~risto/sec e contém uma extensa página man com exemplos. Exemplos adicionais estão disponíveis nesse site. SEC não é do tipo "pronto para uso" como as outras ferramentas listadas anteriormente, mas ele é uma boa base para construir uma ferramenta personalizada de análise de log.

Independentemente da ferramenta que você escolher para varrer arquivos de log, há uma série de coisas que deve procurar obrigatoriamente e trazer de imediato ao conhecimento de um administrador:

- A maioria das mensagens relacionadas com segurança deve receber uma revisão imediata. Normalmente é útil monitorar logins com falha, tentativas **su** e **sudo** para capturar invasões em potencial antes que realmente aconteçam. Se alguém simplesmente esquecer sua senha (como normalmente é o caso), uma oferta imediata e preventiva de ajuda dará uma boa impressão e solidificará sua reputação de seriedade.
- Mensagens sobre discos cheios devem ser sinalizadas e devem ter uma intervenção imediata. Em geral, discos cheios paralisam o trabalho.
- Mensagens que são repetidas muitas vezes merecem atenção, mesmo que somente em nome de uma boa organização.

10.6 EXERCÍCIOS

E10.1 Quais são as principais razões para se preservar arquivos de log antigos?

E10.2 Qual é a diferença entre **lastlog** e **wtmp**? Qual seria uma diretiva de rodízio razoável para cada um deles?

E10.3 Analise e entenda a seguinte linha de **syslog.conf**:

```
*.notice;kern.debug;lpr.info;mail.crit;news.err  /var/log/messages
```

Ela parece razoável?

E10.4 Procure em seus arquivos de log entradas de **named**, o servidor de nomes DNS. Alguma máquina está tentando atualizar seus arquivos de domínio dinamicamente? Elas estão sendo bem-sucedidas em seu intento? (Pode exigir acesso de root.)

★ E10.5 Onde você encontraria o log de inicialização para sua máquina? Quais são as questões que afetam o registro em log no momento da inicialização? Como **klogd** resolve essas questões?

★ E10.6 Investigue a diretiva de registro em log em uso em sua instalação, inclusive a diretiva de rodízio de arquivos de log. Quanto espaço em disco é dedicado ao registro em log? Por quanto tempo os arquivos de log são preservados? Você é capaz de antever circunstâncias em que a diretiva aplicada em sua instalação não seria adequada? Que solução você recomendaria? (Exige acesso de root.)

★ E10.7 Algumas mensagens de log são extremamente importantes e devem ser analisadas imediatamente por um administrador. Que sistema você poderia configurar para garantir que isso aconteça o mais rápido possível?

★ E10.8 Escreva um programa ou script que submete mensagens a syslog com o recurso "usuário". (Pode exigir acesso de root.)

11 Gerenciamento de software e configuração

Embora as distribuições do Linux estejam rapidamente tornando-se mais completas em termos de recursos e mais amigáveis ao usuário, a configuração de softwares ainda é um elemento importante no ambiente administrativo. Além de realizar instalações prontas para uso, a maioria dos fornecedores UNIX vende seus computadores com softwares pré-instalados e disponibilizam procedimentos de configuração refinados para instalar correções e software de sistema. Contudo, você precisa implementar todas as personalizações que são apropriadas à sua instalação. Em geral, os administradores devem realizar todas as seguintes tarefas:

- Automatizar em massa as instalações do sistema operacional
- Personalizar os sistemas de acordo com o ambiente local
- Manter os sistemas corrigidos e atualizados
- Gerenciar pacotes de software suplementares

O processo de configurar uma distribuição ou pacote de software pronto para uso a fim de que ele possa ser adaptado às suas necessidades (e às suas convenções locais de segurança, posicionamento de arquivo e topologia de rede) é freqüentemente conhecido como "localização". Este capítulo explora algumas técnicas e aplicativos que ajudam a reduzir a dor de cabeça da instalação de software e permitem o escalonamento elegante dessas tarefas.

11.1 INSTALAÇÃO BÁSICA DO LINUX

Todas as distribuições atuais do Linux fornecem procedimentos simples e diretos para uma instalação básica. Você inicializa um CD-ROM ou disquete, responde a algumas perguntas e informa ao instalador os pacotes de software a copiar para sua unidade de disco. A maioria das distribuições também tem bons guias de instalação.

As versões Enterprise do Red Hat incluem um manual impresso de instalação. Estas também podem ser localizadas on-line (com outros manuais do Red Hat) em www.redhat.com/docs/manuals.

O Fedora tem seu próprio projeto de documentação, mas como as distribuições atuais do Fedora estão totalmente integradas ao Red Hat Enterprise Linux, parece que os colaboradores do Fedora não estão dispostos a investir tempo duplicando o conteúdo da documentação do Enterprise existente. Os manuais do Red Hat costumam ser um bom ponto de partida ao procurar respostas a perguntas sobre o Fedora.

O pacote do SUSE Linux Enterprise também inclui manuais de instalação. Esses documentos estão disponíveis on-line em:

www.novell.com/documentation/suse.html

Documentação análoga para o openSUSE pode ser encontrada aqui:

www.opensuse.org/Documentation

Você pode encontrar o manual de instalação do Debian aqui:

www.debian.org/releases/stable/installmanual

O Ubuntu resume bem as opções de instalação na forma wiki aqui:

https://wiki.ubuntu.com/Installation

Se tiver de instalar o sistema operacional em mais de um computador, rapidamente você vai atingir os limites da instalação interativa. Para início de conversa, ela não é muito reproduzível. (Será que você consegue selecionar repetidamente um conjunto consistente de pacotes de software a partir das muitas opções oferecidas em tempo de instalação?) Você pode minimizar erros de instalação com uma lista de verificação de localização, mas mesmo essa medida não evitará todas as potenciais variações. A instalação interativa também consome tempo e esforço.

Os instaladores para todas as distribuições utilizadas como exemplo neste livro podem ser programados em script, de modo que uma outra opção seria criar um disquete ou CD-ROM contendo a configuração desejada. Essa solução resolve o problema da reprodução, porém ainda não é suficientemente escalonável. Se você estiver instalando centenas de máquinas, gerar e instalar centenas de disquetes de configuração é o cúmulo do tédio.

Também é possível realizar uma instalação automática por uma rede e essa normalmente é a opção mais conveniente para instalações com centenas ou milhares de sistemas a atualizar. Todos os nossos exemplos de distribuição podem ser instalados dessa maneira, mas os sistemas prontos para uso requerem alguma configuração para que o processo ocorra tranqüilamente.

Além disso, vários projetos de código-fonte aberto estão se esforçando para produzir instaladores de rede de terceiros, por exemplo, o SystemImager. Esses sistemas são discutidos em detalhe mais adiante neste capítulo.

Inicialização de PCs via rede

A inicialização de PCs via rede permite inicializar um computador através da rede, em vez de a partir de um disco rígido, disquete ou CD-ROM. Esse tipo de inicialização é conveniente para uma instalação de software porque significa que você pode simplesmente inicializar via rede uma máquina em um outro lugar sem precisar inserir uma mídia física de inicialização.

Os fabricantes de máquinas UNIX dedicadas há muito tempo suportam a inicialização pela rede, mas só recentemente esse recurso tornou-se o padrão nos PCs. Depois de anos de hacks envolvendo PROMs de inicialização personalizadas (acesse etherboot.sourceforge.net), a Intel fez a todos nós um favor estabelecendo o padrão PXE (Pre-boot eXecution Environment). Embora o PXE não seja perfeito, um padrão rudimentar é melhor do que nenhum. (Bem, em geral.)

O PXE atua como uma miniatura do SO que reside em uma ROM na sua placa de rede. Ele expõe suas capacidades de rede por meio de uma API padronizada que a BIOS do sistema utiliza. Essa cooperação torna possível que uma única rotina de inicialização inicialize via rede o Linux em qualquer PC com o PXE ativado sem a necessidade de fornecer drivers especiais para cada placa de rede.

Consulte a página 217 para informações adicionais sobre DHCP.

A parte externa (rede) do protocolo PXE é simples e direta e é semelhante aos procedimentos de inicialização via rede usados nas outras arquiteturas. Um computador emite um broadcast de uma solicitação de "descoberta" de DHCP com a opção PXE configurada, e um servidor ou proxy DHCP responde com um pacote DHCP que inclui opções PXE (o nome de um servidor de inicialização e um arquivo de inicialização). O cliente faz o download de seu arquivo de inicialização via TFTP (ou, opcionalmente, TFTP multibroadcast) e então o executa.

Embora a especificação PXE não seja totalmente compatível com DHCP genérico, na prática muitos servidores DHCP modernos são capazes de fornecer serviço de inicialização PXE.

O grau de suporte PXE predefinido nos instaladores varia entre distribuições. Essa, porém, é uma área do desenvolvimento ativo, e sistemas baseados no PXE funcionais estão se tornando cada vez mais comuns.

Documentos que descrevem o PXE estão à disposição em uma série de fontes. Um bom local para iniciar é o *Remote Boot HOWTO* disponível em www.tdlp.org.

Configurando o PXE para Linux

Há vários sistemas inicializados pela rede e baseados no PXE, mas aquele que funciona melhor nesse momento é o PXELINUX de H. Peter Anvin, que é parte do seu conjunto SYSLINUX de rotinas de inicialização para todas as situações. Confira esse sistema em syslinux.zytor.com.

O PXELINUX fornece um arquivo de inicialização que você instala no diretório **tftpboot** do seu servidor e cuja cópia é feita no PC de inicialização quando o PXE entra em ação. O PC então executa o arquivo de inicialização e faz a cópia da sua configuração a partir do servidor; a configuração especifica qual kernel utilizar. Essa cadeia de eventos pode ocorrer sem nenhuma intervenção ou você pode criar um menu personalizado de inicialização se preferir.

O PXELINUX utiliza a API PXE para as cópias e, portanto, o processo de inicialização é completamente independente de hardware. Ele não está limitado a inicialização do Linux — o PXELINUX também pode inicializar outros SOs e pode até mesmo inicializar imagens em um disquete se você utilizar o kernel MEMDISK, que também é parte do pacote SYSLINUX.

No lado do servidor, certifique-se de que você está utilizando o servidor DHCP do ISC (Internet Systems Consortium). Se estiver utilizando um servidor diferente, talvez você precise configurar opções DHCP adicionais ou utilizar representações numéricas no arquivo de configuração do servidor.

Todo o Capítulo 14 do *Red Hat Enterprise Linux 3 System Administration Guide* é sobre o PXE e inicialização via rede. Encontre-o em www.redhat.com/docs/manuals.

O primo gratuito do Fedora, RHEL, permite acessar o sistema gráfico de configuração e inicialização via rede executando o comando **system-config-netboot**.

O SuSE não vem com nenhum pacote PXE, mas você pode emprestar aqueles do Red Hat. O YaST pode ser adaptado facilmente para operar com PXE, embora essa configuração não seja oficialmente suportada.

O instalador do Debian pode ser inicializado via rede sem problemas; basta selecionar a imagem do kernel de inicialização via rede em www.debian.org/devel/debian-installer. Existe até uma página Wiki que aponta para os recursos disponíveis:

wiki.debian.net/index.cgi?DebianInstallerNetbootPXE

Quando escrevíamos este livro, o guia de instalação do Ubuntu formal para a versão 6.06 não havia sido finalizado. Nesse interim, você pode localizar uma proposta de padrão em doc.ubuntu.com. Para um breve resumo das opções, consulte os tópicos Installation/Netboot e Installation/LocalNet disponíveis por meio da página de instalação principal do wiki:

https://wiki.ubuntu.com/Installation/Netboot

Inicialização via rede de máquinas não-PCs

O PXE é um produto da Intel e é limitado ao hardware IA-32 e IA-64. Outras arquiteturas possuem seus próprios métodos de fazer a inicialização via rede (quase sempre mais elegante do que o PXE). Um desdobramento interessante dessa história de inicialização via rede é que agora o Linux se expandiu além da arquitetura Intel, de modo que atualmente muitos desses sistemas UNIX "dedicados" têm a opção de inicializar o Linux via rede em vez de seus sistemas operacionais nativos.

A discussão das diversas diferenças entre as arquiteturas realmente está fora do escopo deste livro, mas uma série de recursos na Web poderá ajudá-lo.

Máquinas SPARC e a maioria dos computadores PowerPC utilizam o Open Firmware, que é fácil de ser inicializado via rede (digite **boot net**). O FAQ UltraLinux (www.ultralinux.org) inclui um excelente guia para inicialização do Linux via rede em processadores SPARC. Os computadores com chip Alfa geralmente utilizam o software de console SRM para inicializar. O *SRM HOWTO* descreve o uso desse recurso para iniciar o Linux. Para Macs, RS/6000s e outras máquinas baseadas no PowerPC, o procedimento para inicialização via rede é específico tanto ao hardware como à rotina de inicialização que está sendo utilizada, de modo que você deve verificar a documentação da sua rotina de inicialização. Máquinas AMD64 utilizam o PXE da mesma maneira que seus irmãos PC x86.

Kickstart: o instalador automatizado para o Enterprise Linux e o Fedora

O Kickstart é a ferramenta do Red Hat para realizar instalações automatizadas. Na verdade, ele é apenas uma interface de criação de scripts para o instalador Red Hat padrão, o Anaconda, e depende tanto da distribuição básica como dos pacotes RPM. Diferentemente do SystemImager e outros copiadores de imagens brutas, o Kickstart é flexível e bem inteligente em relação a autodetectar o hardware de um sistema.

Consulte a página 164 para informações adicionais sobre pacotes.

Se seu ambiente exigir muito trabalho de localização, você poderá incluir um script de pós-instalação que adapta o sistema aos seus padrões locais. Embora isso funcione, achamos que os scripts pós-instalação tendem a se tornar grandes e difíceis de manter ao longo do tempo. Outra opção é criar um pacote RPM contendo suas personalizações locais. Recomendamos essa rota; ela torna fácil o controle de versão, facilita atualizações posteriores e fornece suporte a dependência e todos os outros recursos úteis que vêm com um sistema de pacotes.

Definindo um arquivo de configuração do Kickstart

O comportamento do Kickstart é controlado por um único arquivo de configuração, geralmente chamado **ks.cfg**. O formato desse arquivo é simples. Se você gosta mais de utilizar interfaces gráficas para configuração, o Red Hat inclui uma ferramenta útil chamada **redhat-config-kickstart** no Enterprise Linux e **system-config-kickstart** no Fedora que lhe permite apontar e clicar para manipular o **ks.cfg**.[1]

O arquivo **ks.cfg** também é bem fácil de ser gerado programaticamente. Por exemplo, suponha que você quisesse instalar um conjunto diferente de pacotes nos servidores e nos clientes e que tivesse dois escritórios que exigissem personalizações levemente diferentes. Você poderia criar um pequeno script em Perl que usasse um conjunto mestre de parâmetros para gerar um arquivo de configuração para os servidores e clientes em cada filial. Alterar o complemento de pacotes se tornaria simplesmente uma questão de alterar esse script Perl em vez de alterar cada arquivo de configuração. Pode até haver alguns casos em que você precise gerar um arquivo de configuração individualizado para cada host. Nessa situação, certamente você iria querer que os arquivos fossem gerados automaticamente.

Um manual interessante para o Kickstart (incluindo uma lista de todas as opções disponíveis) pode ser localizado na Section II do *The Red Hat Enterprise Linux System Administration Guide*. Ele está disponível em www.redhat.com/docs/manuals.

Um pequeno alerta em relação ao arquivo de configuração do Kickstart: se você cometer um erro, o diagnóstico consiste em uma mensagem de rastreamento do Python praticamente ininteligível, que pode ou não conter uma mensagem oculta que lhe aponta diretamente o erro. Uma das principais vantagens do uso do **redhat-config-kickstart** ou **system-config-kickstart** para gerar o arquivo de configuração é que ele garante que o arquivo resultante será pelo menos sintaticamente válido.

Um arquivo de configuração do Kickstart é formado de três partes ordenadas. A primeira delas é a seção de comandos, que especifica opções como o idioma, o teclado e o fuso horário. Essa seção também especifica a origem da distribuição com a opção `url` (no exemplo a seguir, há um host chamado installserver).

---

1. Observe que o Fedora renomeou todas as ferramentas **redhat-config-*** para **system-config-***. A alteração no nome faz parte do esforço de distanciar o Fedora da marca Red Hat e ainda não reflete as diferenças técnicas subjacentes.

Eis um exemplo de uma seção de comandos completa:

```
text
lang en_US                        # idioma que é utilizado durante a instalação...
langsupport en_US                 # ... e o suporte a idioma em tempo de execução.
keyboard us                       # Utiliza o teclado norte-americano.
timezone --utc America/EST        # --utc significa que o relógio do hardware está em UTC (GMT)
mouse
rootpw qualquerUm
reboot                            # Reinicializa após a instalação. Sempre uma boa idéia.
bootloader --location=mbr         # Instala a rotina de inicialização padrão no MBR.
install                           # Instala um novo sistema em vez de atualizar.
url --url http://installserver/redhat
clearpart --all --initlabel       # Limpa todas as partições existentes
part / --fstype ext3 --size 4096
part swap --size 1024
part /var --fstype ext3 -size 1 --grow
network --bootproto dhcp
auth --useshadow --enablemd5
firewall --disabled
xconfig --defaultdesktop=GNOME --startxonboot --resolution 1280x1024 --depth 24
```

O Kickstart utiliza um modo gráfico por padrão, o que vai contra o objetivo de uma instalação sem intervenção do operador. A palavra-chave `text` no início do exemplo corrige isso.

A opção `rootpw` configura a nova senha de root da máquina. O padrão é especificar a senha em texto puro, que pode ser um tanto perigoso em termos de segurança. É possível utilizar o flag `--iscrypted` para especificar uma senha já criptografada; porém, senhas MD5 não são suportadas.

As diretivas `clearpart` e `part` especificam uma lista das partições de disco com seus respectivos tamanhos. Você pode utilizar a opção `--grow` para especificar uma das partições a expandir a fim de preencher qualquer espaço restante no disco. Esse recurso torna fácil acomodar sistemas que têm diferentes tamanhos de disco rígido.

A segunda seção é uma lista dos pacotes a serem instalados, começando com uma diretiva `%packages`. A lista pode conter pacotes individuais, coleções como `@ GNOME` ou a notação `@ Everything` para incluir todos eles. Ao selecionar pacotes individuais, especifique somente o nome do pacote, não a versão ou a extensão **.rpm**. Eis um exemplo:

```
%packages
@ Networked Workstation
@ X Window System
@ GNOME
mylocalpackage
```

Na terceira seção do arquivo de configuração do Kickstart, você pode especificar comandos de shell arbitrários para o Kickstart executar. Há dois possíveis conjuntos de comandos: um introduzido com `%pre` que é executado antes da instalação e um introduzido com `%post` que é executado depois. Ambas as seções possuem algumas restrições sobre a capacidade do sistema de determinar nomes de hosts, de modo que é mais seguro utilizar endereços IP caso queira acessar a rede. Além disso, os comandos de pós-instalação são executados num ambiente em que foi aplicado **chroot**, de maneira que eles não poderão acessar a mídia de instalação.

Construindo um servidor Kickstart

O Kickstart espera que seus arquivos de instalação estejam organizados da maneira como eles estão no CD de distribuição, com os pacotes armazenados em um diretório chamado **RedHat/RPMS** ou **Fedora/RPMS** no servidor. Você pode facilmente adicionar seus próprios pacotes a esse diretório. Mas existem algumas questões das quais precisamos estar cientes.

Primeiramente, se você solicitar ao Kickstart para instalar todos os pacotes (com um `@ Everything` na seção de pacotes do arquivo **ks.cfg**), ele instalará os pacotes adicionais em ordem alfabética depois de os pacotes básicos terem sido instalados. Se seu pacote depende de outros que não estejam no conjunto básico, você pode chamar seu pacote de algo como **zzmeupacotelocal.rpm** para se certificar de que ele seja instalado por último.

Se você quiser instalar todos os pacotes, liste seus pacotes suplementares individualmente na seção `%packages` do arquivo **ks.cfg** ou então adicione seus pacotes a uma ou mais das listas de coleções. As listas de coleções são especificadas por entradas como `@ GNOME` e significam um conjunto predefinido de pacotes cujos membros são enumerados no arquivo **RedHat/base/comps** no servidor. Infelizmente, o formato do arquivo **comps** não é bem documentado. As coleções são as linhas que começam com 0 ou 1 (o número especifica se a coleção é selecionada ou não por padrão). Geralmente, não é

uma boa idéia alterar as coleções padrão. Sugerimos que você as deixe como a Red Hat as definiu e indique explicitamente todos os seus pacotes suplementares no arquivo **ks.cfg**.

Apontando o Kickstart para seu arquivo de configuração

Depois que você criou um arquivo de configuração, há uma série de maneiras de fazer o Kickstart utilizá-lo. O método oficialmente sancionado é inicializar com um disquete ou CD-ROM e solicitar uma instalação Kickstart especificando **linux ks** no prompt `boot:` inicial. Se você não especificar argumentos adicionais, o sistema determinará o endereço de rede usando DHCP. Ele então obtém o servidor de inicialização DHCP e as opções do arquivo de inicialização, tenta montar o servidor de inicialização com NFS e usa o valor da opção do arquivo de inicialização como seu arquivo de configuração Kickstart. Se nenhum arquivo de inicialização foi especificado, o sistema procura um arquivo chamado **/kickstart/***endereçoIPdoHost***-kickstart**.

Alternativamente, o Kickstart pode ser instruído a obter o arquivo de configuração de alguma outra maneira fornecendo um caminho como um argumento para a opção **ks**. Há várias possibilidades. A instrução:

```
boot: linux ks=http:servidor:/path
```

instrui o Kickstart a utilizar HTTP para fazer a cópia do arquivo em vez de NFS. Utilizar **ks=floppy** instrui o Kickstart a procurar o **ks.cfg** na unidade de disquete local.

Para eliminar completamente o emprego da mídia de inicialização, você precisará partir para o PXE. Consulte a página 160 para informações adicionais sobre isso.

AutoYaST: a ferramenta de instalação automatizada do SUSE

O YaST2 é a ferramenta de instalação e de configuração do SUSE. Ela vem com uma ferramenta gráfica interessante e é divertido utilizá-la ao instalar um sistema individual. Distribuições do SUSE mais antigas permitiam a automação com base no YaST convencional, mas os resultados não eram aquilo que todo mundo gostaria. O SUSE 8 introduziu uma ferramenta chamada AutoYaST que automatiza instalações SUSE. Essa ferramenta é o software de instalação automatizada mais poderoso entre todas as distribuições descritas neste livro. Você pode fazer a cópia da documentação detalhada em www.suse.com/~ug.

O SUSE divide o processo da auto-instalação em três fases: preparação, instalação e configuração. A preparação inicial é realizada com o módulo AutoYaST:

```
$ /sbin/yast2 autoyast
```

Esse módulo ajuda a definir os detalhes da configuração desejada. O resultado da execução dele é um arquivo de controle XML que instrui o instalador sobre como configurar um sistema SUSE; a estrutura desse arquivo é descrita na documentação on-line mencionada anteriormente.

Alguns atalhos podem acelerar o processo de configuração. O módulo AutoYaST pode ler nos arquivos de configuração do Red Hat Kickstart para ajudá-lo a atualizar a partir de sistemas de "legado". Se quiser duplicar a configuração da máquina em que você está atualmente trabalhando, uma opção automatiza isso também.

Para realizar uma instalação real, você precisa de três serviços de rede:

- Um servidor DHCP na mesma sub-rede da máquina que você quer configurar
- Um servidor de instalação SUSE ou repositório de pacotes
- Um servidor que fornece as informações de configuração para a instalação

O último desses servidores pode fornecer os arquivos de configuração segundo sua escolha HTTP, NFS ou TFTP preferida.

Na configuração mais básica, você cria um arquivo de controle para cada máquina que você quer instalar. O AutoYaST utiliza o endereço IP do cliente para determinar qual arquivo de controle utilizar. Essa abordagem não é muito eficiente se você tiver de instalar uma série de máquinas um pouco diferentes.

Você pode criar configurações mais complexas utilizando um sistema de regras. Com base nas propriedades de sistema, como tamanho de disco, ID de host ou disponibilidade PCMCIA, diferentes arquivos de controle são comparados com o sistema alvo. O conteúdo de todos os arquivos de controle selecionados é mesclado, sendo que o último arquivo de controle sobrescreve os anteriores no caso de conflitos. (Um arquivo de controle não precisa especificar todos os aspectos da configuração de um sistema, portanto essa mesclagem faz sentido.)

Arquivos de controle também podem definir "classes" de máquinas com base nos hostnames ou em intervalos de endereço IP e cada classe pode ter ainda um outro arquivo de controle subsidiário associado com ela. Máquinas podem pertencer à classe zero, à classe um ou a múltiplas classes e suas configurações incorporarão o conteúdo de todos os arquivos de controle da classe apropriada.

Graças à sua capacidade de integrar o conteúdo de múltiplos arquivos de controle, a estrutura do AutoYaST permite que configurações complexas sejam definidas com mínima redundância. Os arquivos de controle XML são um pouco com-

plexos para leitura por humanos, mas esses arquivos são simples de processar e editar com quaisquer ferramentas de processamento XML comumente disponíveis.

O instalador do Debian e do Ubuntu

O instalador Debian (chamado, muito apropriadamente, **debian-installer**) foi completamente reformulado para a distribuição "sarge". Essa ferramenta é agora uma com a qual é muito bom trabalhar. Ela também é utilizada pelo Ubuntu, embora o Ubuntu (achamos que inteligentemente) preserve todos os nomes dos comandos que parecem ser específicos ao Debian.

O novo sistema tem um projeto modular e foi escrito com uma combinação de scripts C e scripts de shell. Você pode utilizar apenas dois disquetes em máquinas com pouca memória, tão pouco quanto 32 MB, e selecionar todos os arquivos necessários para a instalação diretamente da rede, incluindo a lista de arquivos de que ele precisa.

Todas as partes interativas do instalador do Debian utilizam o utilitário **cdebconf** para decidir quais perguntas fazer e quais respostas padrão utilizar. Fornecendo ao **cdebconf** um banco de dados com respostas pré-formuladas, é possível automatizar totalmente o instalador. Você pode gerar o banco de dados manualmente (ele é apenas um arquivo de texto) ou realizar uma instalação interativa em um sistema de exemplo e então excluir as suas respostas **cdebconf** com estes comandos:

```
# debconf-get-selections --installer > config.cfg
# debconf-get-selections >> config.cfg
```

Torne o arquivo de configuração disponível na rede e então o passe para o kernel em tempo de instalação com o argumento de kernel a seguir:

```
preseed/url=http://host/caminho/para/preseed
```

Se quiser instalar o Debian automaticamente, examine também o pacote **cdebootstrap**. Ele se baseia no mesmo código do instalador do Debian e permite criar uma imagem da instalação em um subdiretório do disco local, selecionando os pacotes requeridos diretamente de um repositório do Debian. Esse recurso pode ser muito útil para uso em scripts de configuração automatizados da forma que você planejar.

Nas distribuições anteriores do Debian, esse instalador foi a origem de inúmeras reclamações. Sem saída, muitos administradores de sistemas implementavam suas próprias instalações e sistemas de configuração, alguns dos quais estão catalogados aqui:

www.linuxmafia.com/faq/Debian/installers.html

Outra alternativa às ferramentas Debian padrão é um pacote chamado FAI (Fully Automatic Installation). O site é www.informatik.uni-koeln.de/fai.

Esses sistemas alternativos são menos necessários agora que no passado, mas mesmo assim vale a pena revisá-los se você decidir não utilizar o instalador padrão.

Embora seja uma distribuição fora da linhagem Red Hat, o Ubuntu enxertou compatibilidade dos arquivos de controle do Kickstart em seu próprio instalador subjacente. Ele também inclui a ferramenta **system-config-kickstart** para criar esses arquivos. A funcionalidade do Kickstart no Ubuntu ainda não é 100% sólida, mas parece suficientemente completa para uso. A compatibilidade com o Kickstart ainda não é compartilhada com o Debian.

Instalando a partir de um sistema mestre

Se for forçado a não utilizar o instalador padrão da sua distribuição, leve em consideração se o hardware da sua instalação é suficientemente uniforme para permitir que a distribuição seja simplesmente duplicada de uma máquina para outra. A duplicação pode ser mais rápida que a instalação tradicional e sua garantia de uniformidade é um pouco mais forte.

Talvez você ache útil a seguinte "receita" de instalação:

- Instale e configure o sistema mestre, utilizando o processo de instalação normal da distribuição ou utilizando um script como o **debootstrap** do Debian.
- Inicialize a máquina em que você quer duplicar a distribuição, utilizando um CD-ROM de inicialização como o Knoppix[2] ou um método baseado no PXE.
- Particione o disco utilizando **cfdisk** interativamente ou **sfdisk** a partir de um script.
- Crie sistemas de arquivos, inicialize o espaço de troca e monte tudo sob **/mnt**.
- Duplique a instalação mestra no disco recém-particionado utilizando **ssh** e **rsync** (consulte a página 353). Certifique-se de excluir os diretórios **/tmp**, **/proc** e **/sys**.

2. O Knoppix é uma versão do Debian que executa diretamente de um CD/DVD-ROM de inicialização; nenhuma instalação no disco rígido é necessária. Ele é ideal para uso durante a instalação ou salvamento ou ao tratar incidentes de segurança. Acesse www.knoppix.org. Várias outras distribuições (por exemplo, Ubuntu) também estão começando a se tornar executáveis a partir de um CD inicializável.

- Edite todos os arquivos que precisam de personalização (por exemplo, **/etc/modules** ou **/etc/fstab**) ou, melhor ainda, automatize esse processo.
- Torne inicializável a nova máquina instalando GRUB ou LILO no registro de inicialização mestre do disco. Consulte a página 19 para informações adicionais.

11.2 CLIENTES SEM DISCO

Na década de 1980, quando o preço de um disco rígido era alto, muitas empresas instalavam estações de trabalho que não tinham discos rígidos locais. Em vez disso, esses computadores, conhecidos como "clientes sem disco", montavam seus sistemas de arquivos a partir de um servidor de rede. Hoje, o preço dos discos caiu e quase ninguém utiliza clientes sem disco. Toda a idéia parece fora de moda, como fita de nove trilhas ou impressoras matriciais.

Mas espere! A operação sem disco continua bem possível e, na verdade, é uma opção razoável a considerar para certos tipos de instalações. Eis algumas vantagens:

- Em uma configuração sem disco, todos os dados relevantes são mantidos no servidor. Clientes sem disco nunca estão off-line no sentido de que suas informações de configuração tornam-se inacessíveis, portanto procedimentos administrativos não precisam levar em conta máquinas inoperantes como um possível caso especial.
- A quantidade de informações de configuração verdadeiramente local é mantida em um mínimo. Quase nenhuma operação administrativa exige que você esteja fisicamente presente na frente da máquina de um cliente.
- Novos clientes podem ser configurados no servidor antes de executarem ou até mesmo de estarem fisicamente presentes. Os clientes comportam-se muito mais como entidades lógicas do que entidades físicas.
- Alterações importantes na configuração de um cliente podem ser executadas no momento em que a reinicialização ocorre. Uma árvore-raiz atualizada pode residir ao lado daquela ativa, esperando o momento certo para ser ativada. Você pode fornecer um serviço mais confiável e mais rápido ao utilizar conceitos mais simples de distribuição de software e atualização.
- Para muitas aplicações, o caro e complexo array de RAID com que você pode arcar a fim de colocar no seu servidor pode ser mais rápido do que discos locais baratos, mesmo depois de a latência de rede ser levada em consideração. É possível que o servidor de rede também seja mais confiável. Discos falham, mas uma configuração de servidor redundante pode proteger uma rede inteira de clientes sem disco.
- Nenhuma regra determina que clientes sem disco não possam ter discos. Se preferir, utilize discos locais para espaço de troca ou cache. Na era moderna, a questão não é o fato de que os clientes não têm discos físicos; é o fato de que eles não utilizam esses discos para armazenar nada que requer atenção administrativa.
- Muitos aplicativos estão vinculados ao uso da CPU e não têm nenhuma dependência particular para com a velocidade de disco. Outros que são potenciais usuários que utilizam discos intensamente podem ser convertidos em uma operação de CPU com a adição de mais memória. Servidores Web, servidores de banco de dados e servidores para protocolos de rede, todos potencialmente entram nessa categoria.

Em um ambiente sem disco, os clientes em geral têm pequenos sistemas de arquivos-raiz dedicados a usos exclusivos. Clientes podem compartilhar um diretório de leitura **/usr** comum porque esse diretório não contém nenhuma informação específica da máquina e nunca gravam nele em uma operação normal.

O Red Hat Enterprise Linux é a única distribuição Linux com um sistema padronizado para configurar clientes sem disco prontos para uso. Mas mesmo sem suporte elaborado por parte do fornecedor, essa configuração não é uma ciência complexa e não é difícil de fazê-la por conta própria. Muitos sistemas têm um mecanismo para inicializar via rede terminais X Windows; você pode utilizar isso como um ponto de partida para inicializar estações de trabalho via rede.

Se a versão da sua distribuição não incluir instruções para configurar clientes sem disco, comece copiando um sistema de arquivos-raiz recém-instalado em uma máquina para seu servidor como a base da clonagem de novas máquinas. Descubra como inicializar seus sistemas pela rede com um diretório-raiz do tipo NFS. Compartilhe as partes de leitura da árvore do sistema de arquivos de uma máquina (por exemplo, **/usr**) com todos os clientes.

Ao clonar a raiz de um cliente, talvez você precise personalizar o seguinte:

- Tabela de montagem
- Suporte ao dispositivo de hardware (se sua rede não for homogênea)
- Configuração de rede (possivelmente)
- Scripts de inicialização (possivelmente)

Hardware montado com base em padrões abertos (*commodity hardware*) costuma ser comercializado com discos IDE baratos. Utilize-os para partições de espaço de troca local e de *trabalho*

Consulte a página 560 para informações adicionais sobre discos RAM.

A barata RAM de hoje é uma excelente bênção para máquinas sem disco. Verifique se seu sistema operacional suporta o uso. Considere a possibilidade de utilizar um sistema de arquivos baseado em memória RAM como um armazenamento de apoio para os diretórios **/tmp** e **/dev** (ou como um cache dos arquivos freqüentemente utilizados que você pré-carrega em tempo de inicialização).[3] Manter a árvore **/dev** na memória RAM pode aprimorar o desempenho consideravelmente porque os programas que gostam de pesquisar a árvore **/dev** não sofrem a latência do NFS.

11.3 GERENCIAMENTO DE PACOTES

Todas as distribuições do Linux utilizam alguma forma de sistema de pacotes para facilitar o trabalho do gerenciamento da configuração. Pacotes são tradicionalmente utilizados para distribuir softwares, mas eles também podem ser utilizados para empacotar arquivos de configuração e dados administrativos. Eles possuem várias vantagens em relação aos tradicionais arquivos não estruturados **.tar.gz**. Talvez o mais importante seja que eles tentam tornar o processo de instalação o mais atômico possível. Se ocorrer algum erro, o pacote poderá ser revertido ou reaplicado.

Fornecedores do UNIX têm procedimentos especiais para a distribuição de correções aos seus clientes, mas os distribuidores do Linux tiram proveito dos seus recursos padrão de gerenciamento de pacotes. A fim de emitir uma correção, o distribuidor simplesmente disponibiliza um pacote atualizado. Quando instalada pelos clientes, a nova versão substitui a antiga.

Em geral, os instaladores de pacotes estão cientes dos arquivos de configuração e normalmente não sobrescreverão as personalizações locais realizadas por um administrador de sistemas. Os instaladores de pacotes farão um backup dos arquivos de configuração existentes que eles alteram ou fornecerão arquivos de configuração de exemplo sob um nome diferente (por exemplo, **pkg.conf.rpmnew**). Caso você descubra que algum pacote recém-instalado esteja causando problemas no sistema, você pode, pelo menos em teoria, voltar atrás para restaurar o sistema ao seu estado original. Naturalmente, teoria é diferente da prática, assim não tente isso em um sistema de produção sem primeiro testá-lo.

Os sistemas de empacotamento definem um modelo de dependência que permite aos mantenedores de pacotes garantirem que todas as bibliotecas e infra-estrutura de suporte das quais seus aplicativos dependem sejam instaladas apropriadamente. Os pacotes também podem executar scripts em vários pontos durante a instalação, de modo que eles podem fazer mais do que simplesmente regurgitar arquivos novos. (Esse recurso provavelmente é responsável pela maioria das falhas observadas nos pacotes em restaurar o sistema ao seu estado original após uma desinstalação.)

Pacotes também são uma maneira interessante de distribuir suas próprias localizações. Você pode criar facilmente um pacote que, quando instalado, lê informações de localização sobre uma máquina (ou as obtém de um banco de dados centralizado) e utiliza essas informações para definir arquivos de configuração locais. Você também pode distribuir seus aplicativos locais como pacotes completos com as dependências ou criar pacotes para aplicativos de terceiros que normalmente não são distribuídos no formato de pacote. Você pode criar versões de seus pacotes e utilizar o mecanismo de dependência para atualizar máquinas automaticamente quando a nova versão de seu pacote de localização for instalada.

Você também pode utilizar o mecanismo de dependência para criar grupos de pacotes. Por exemplo, é possível criar um pacote que não instala nada próprio, porém depende de várias outras correções. Instalar o pacote com as dependências ativadas resulta na instalação de todas as correções em um único passo.

Sistemas disponíveis de gerenciamento de pacotes

Há dois formatos de pacotes sendo utilizados comumente. O Red Hat, o Fedora, o SUSE e várias outras distribuições utilizam o RPM, o Red Hat Package Manager. O Debian e o Ubuntu utilizam um formato **.deb** separado. Esses dois formatos são funcionalmente semelhantes.

É fácil converter entre as duas formas de pacotes com uma ferramenta como **alien** em kitenet.net/programs/alien. **alien** não sabe nada sobre o software dentro de um pacote, portanto, se o conteúdo não for compatível com sua distribuição, o **alien** não ajudará. Em geral, é melhor ater-se ao mecanismo de pacote nativo utilizado por sua distribuição.

Os sistemas de pacotes RPM e **.deb** agora funcionam como ferramentas completas de gerenciamento de configuração em duas camadas. No nível mais baixo estão as ferramentas que instalam, desinstalam e consultam pacotes: **rpm** para RPM e **dpkg** para **.deb**.

---

3. Mas observe que **/dev** talvez já resida em um sistema de arquivos baseado no kernel.

Por cima desses comandos, estão os sistemas que sabem como localizar pacotes na Internet, analisar dependências entre pacotes e atualizar todos os pacotes em um sistema. Os principais concorrentes nesse nível são o **yum**, que funciona com o sistema RPM, o Red Hat Network, que é específico ao Red Hat Linux e utiliza RPM, e o Advanced Package Tool (APT) do Debian, que teve origem no universo **.deb**, mas agora funciona igualmente bem com pacotes RPM e **.deb**.

Nas próximas páginas, revisaremos os comandos de baixo nível **rpm** e **dpkg**. Na seção *Sistemas de gerenciamento de pacotes de alto nível*, iniciando na página 168, discutimos os abrangentes sistemas de atualização (por exemplo, APT e **yum**) que aprimoram esses recursos de baixo nível.

rpm: gerenciando pacotes RPM

O comando **rpm** instala, verifica e consulta o status dos pacotes. Inicialmente ele também construía esses pacotes, mas essa função agora é desempenhada pelo comando **rpmbuild**. Entretanto, as opções **rpm** ainda têm interações complexas e só podem ser utilizadas conjuntamente em certas combinações. É mais útil considerar o **rpm** como se fossem vários comandos diferentes que compartilham o mesmo nome.

O modo como você instrui o **rpm** a entrar (como **--install** ou **--query**) especifica quais das múltiplas personalidades do **rmp** você espera acessar. **rpm --help** lista todas as opções divididas por modo, mas vale a pena ler a página man em detalhes se você costuma lidar com pacotes RPM.

As opções básicas são **--install**, **--upgrade**, **--erase** e **--query**. A opção de consulta é um pouco complexa pelo fato de que ela só serve para ativar outras opções; você deve fornecer um flag de linha de comando adicional para fazer uma pergunta específica. Por exemplo, o comando **rpm --query --all** lista os pacotes instalados no sistema.

Vejamos um exemplo. Suponha que você precise instalar uma nova versão de OpenSSH devido a uma correção de segurança que foi recentemente publicada. Depois de fazer a cópia do pacote para seu computador local, execute **rpm --upgrade** para substituir a versão antiga pela nova:

```
# rpm --upgrade openssh-2.9p2-12.i386.rpm
error: failed dependencies:
openssh = 2.9p2-7 is needed by openssh-askpass-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-askpass-gnome-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-clients-2.9p2-7
openssh = 2.9p2-7 is needed by openssh-server-2.9p2-7
```

Dااا... Bem, talvez não seja tão simples assim. Aqui vemos que a versão atualmente instalada do OpenSSH, 2.9p2-7, é exigida por alguns outros pacotes. O **rpm** não deixará atualizar o OpenSSH porque a alteração poderia afetar a operação desses outros pacotes. Esse tipo de conflito acontece todo o tempo e é uma motivação importante para o desenvolvimento de sistemas como o APT e o **yum**. Na prática, não tentaríamos desembaraçar as dependências manualmente, mas vamos prosseguir apenas com **rpm** para o propósito desse exemplo.

Poderíamos forçar a atualização com a opção **--force**, mas normalmente essa não é uma boa idéia. Isso não é Windows; as informações de dependência estão aí para poupar tempo e problemas, não apenas para atrapalhar você. Não há nada como um SSH quebrado em um sistema remoto para arruinar a manhã de um administrador de sistemas.

Em vez disso, também utilizaremos as versões atualizadas dos pacotes dependentes. Se fôssemos inteligentes, poderíamos ter determinado que outros pacotes dependessem do OpenSSH antes mesmo de tentarmos a atualização:

```
# rpm --query --whatrequires openssh
openssh-askpass-2.9p2-7
openssh-askpass-gnome-2.9p2-7
openssh-clients-2.9p2-7
openssh-server-2.9p2-7
```

Suponha que tivéssemos obtido cópias atualizadas de todos os pacotes. Poderíamos instalá-los um de cada vez, mas o **rpm** é suficientemente inteligente para tratá-los todos ao mesmo tempo. Se listar vários RPMs na linha de comando, o **rpm** irá classificá-los de acordo com a dependência antes da instalação.

```
# rpm --upgrade openssh-*
```

Legal! Parece que deu certo, e com certeza:

```
# rpm --query openssh
openssh-2.9p2-12
```

Observe que o **rpm** sabe sobre qual pacote estamos falando mesmo que o nome completo ou versão do pacote não fossem especificados.

dpkg: gerenciando pacotes no estilo Debian

Assim como os pacotes RPM têm um comando **rpm** do tipo "tudo em um", pacotes Debian têm o comando **dpkg**. Opções úteis incluem **--install**, **-remove** e **-l** para listar os pacotes que foram instalados no sistema. Observe que **dpkg --install** de um pacote que já está no sistema remove a versão anterior antes da instalação.

Suponha que a equipe de segurança do Debian tenha lançado recentemente uma correção para **nvi** a fim de corrigir um problema de segurança em potencial. Após pegar a correção, executamos **dpkg** para instalá-la. Como pode ser observado, é muito mais loquaz do que **rpm** e nos informa exatamente o que está fazendo.

```
# dpkg --install ./nvi_1.79-16a.1_i386.deb
(Reading database ... 24368 files and directories currently installed.)
Preparing to replace nvi 1.79-14 (using ./nvi_1.79-16a.1_i386.deb) ...
Unpacking replacement nvi ...
Setting up nvi (1.79-16a.1) ...
Checking available versions of ex, updating links in /etc/alternatives ...
(You may modify the symlinks there yourself if desired - see 'man ln'.)
Leaving ex (/usr/bin/ex) pointing to /usr/bin/nex.
Leaving ex.1.gz (/usr/share/man/man1/ex.1.gz) pointing to
    /usr/share/man/man1/nex.1.gz.
...
```

Agora podemos utilizar **dpkg -l** para verificar se a instalação funcionou. O flag **-l** aceita um padrão de pesquisa opcional, de maneira que podemos somente procurar **nvi**:

```
$ dpkg -l nvi
Desired=Unknown/Install/Remove/Purge
|Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/Name           Version              Description
+++-==========-   ==============-       ================================
ii  nvi           1.79-16a.1           4.4BSD re-implementation of vi.
```

Parece que nossa instalação funcionou perfeitamente.

11.4 SISTEMAS DE GERENCIAMENTO DE PACOTES DE ALTO NÍVEL

Sistemas de gerenciamento de metapacotes como o APT, **yum** e o Red Hat Network compartilham vários objetivos:

- Simplificar a localização e o processo de cópia dos pacotes
- Automatizar o processo de atualização ou a atualização de sistemas
- Facilitar o gerenciamento das dependências entre pacotes

Claramente, há mais para esses sistemas do que apenas comandos do lado cliente. Todos requerem que os mantenedores da distribuição organizem suas ofertas de uma maneira acordada de modo que o software possa ser acessado e racionalizado pelos clientes.

Como nenhum único fornecedor pode incluir todo o "mundo dos softwares Linux", todos os sistemas permitem a existência de múltiplos repositórios de softwares. Os repositórios podem ser locais para sua rede, portanto esses sistemas produzem uma base de primeira linha para que você crie seu próprio sistema de distribuição interno.

O Red Hat Network está bem amarrado ao Red Hat Enterprise Linux. Ele é um serviço comercial que é cobrado e oferece mais em termos de interfaces gráficas atraentes e da capacidade de automação do que o APT e o **yum** oferecem. Infelizmente, trata-se de uma misteriosa caixa-preta oculta. O lado cliente pode referenciar repositórios **yum** e APT e essa capacidade permite que distribuições como CentOS se adaptem à interface gráfica do cliente para uso não-proprietário.

O APT é mais bem documentado do que o Red Hat Network, é significativamente mais portável e é gratuito. Também é mais flexível em termos daquilo que você pode fazer com ele. O APT veio do mundo do Debian e do **dpkg**, mas ele foi estendido para incluir RPMs e há versões disponíveis que funcionam com todas as nossas distribuições de exemplo. Nesse ponto, ele é a coisa mais próxima a um padrão universal para distribuição de softwares.

yum é um análogo do APT específico ao RPM. Ele é o gerenciador de pacotes padrão para o Fedora, embora execute em qualquer sistema baseado em RPM, desde que você possa apontá-lo para os repositórios apropriadamente formatados.

Em um confronto frente a frente, no geral, o **yum** foi preferido à versão RPM do APT, mas não está claro se há alguma razão técnica sólida para isso. O **yum** tem raízes mais profundas no mundo do RPM, o que lhe garante uma possível liderança. O futuro do APT-RPM era nebuloso no início de 2005 quando o desenvolvedor original, Gustavo Niemeyer, abandonou-o para trabalhar em um sistema mais abrangente (o Smart Package Manager em labix.org; ele ainda não é o mais importante, mas é amplamente aceito como sendo a próxima grande coisa). O desenvolvimento do APT-RPM foi por fim retomado por Panu Matilainen e o projeto está atualmente sob desenvolvimento ativo mais uma vez.

Gostamos do APT e o consideramos uma escolha sólida se você quiser configurar sua própria rede automatizada para distribuição de pacotes, independentemente das distribuições atualmente em uso na sua instalação. Consulte a seção *Configurando um servidor APT interno*, na página 172, para informações adicionais.

O SUSE é como um "patinho feio" no mundo do gerenciamento de pacotes. Ele utiliza pacotes RPM, mas as distribuições anteriores só suportavam uma ferramenta própria do SUSE, YaST Online Update, para realizar atualizações de sistema. Num ímpeto de promiscuidade, o SUSE recentemente adicionou algum grau de suporte ao **yum**, APT-RPM e o próprio agente ZENworks Linux Management da Novell, com ZENworks sendo o principal gerenciador de atualizações. (Consulte **rug** para uma interface de linha de comandos.)

O ZENworks é parte de uma linha maior de produtos que incorpora a tentativa da Novell de dominar o espaço de gerenciamento de configuração para diversas plataformas. Ele é a melhor opção para você? Bem, talvez, se você for um revendedor Novell e estiver interessado em suporte pago. As instalações que querem manter seu gerenciamento de pacotes sem custos e relativamente apolíticos poderiam investigar o Smart Package Manager mencionado anteriormente.

Repositórios de pacotes

Os distribuidores do Linux mantêm repositórios de software que funcionam lado a lado com seus sistemas de gerenciamento de pacotes escolhidos. A configuração padrão para o sistema de gerenciamento de pacotes normalmente aponta para um ou mais servidores Web ou servidores FTP bem conhecidos que estão sob o controle do distribuidor.

Entretanto, não fica imediatamente óbvio o que esses repositórios devem conter. Eles devem incluir apenas os conjuntos dos pacotes tidos como distribuições fundamentais e formais? Distribuições formais mais atualizações de segurança? Versões atualizadas de todos os pacotes que existiam nas distribuições formais? Software útil de terceiros não oficialmente suportado pelo distribuidor? Código-fonte? Binários para múltiplas arquiteturas de hardware? Ao executar **apt-get upgrade** ou **yum upgrade** para atualizar o sistema, o que isso significa exatamente?

Em geral, sistemas de gerenciamento de pacotes devem responder a todas essas perguntas e devem tornar fácil para as instalações selecionarem seções cruzadas específicas que elas querem definir como seu "mundo de software". Os conceitos a seguir ajudam a estruturar esse processo:

- Uma "distribuição" é um instantâneo autoconsistente do universo de um pacote. Antes da era da Internet, distribuições identificadas para SOs eram mais ou menos imutáveis e associadas a um momento específico; correções de segurança foram disponibilizadas separadamente. Hoje em dia, uma distribuição é um conceito mais nebuloso. As distribuições evoluem ao longo do tempo à medida que os pacotes são atualizados. Algumas distribuições, como o Red Hat Enterprise Linux, são especificamente projetadas para evoluir lentamente; por padrão, apenas as atualizações de segurança são incorporadas. Outras distribuições, como as versões beta, mudam com freqüência e significativamente. Mas em todos os casos, a distribuição é a linha mestra, o alvo, a atualização com a qual quero que meu sistema pareça.
- Um "componente" é um subconjunto do software dentro de uma distribuição. As distribuições se dividem de uma maneira diferente, mas uma distinção comum é a de um software básico abençoado pelo distribuidor e um software extra disponibilizado pela comunidade mais ampla. Uma outra distinção comum no mundo do Linux é aquela entre partes gratuitas e de código-fonte aberto de uma distribuição e as partes corrompidas por algum tipo de acordo de licenciamento restritivo.

  De interesse particular do ponto de vista administrativo, são os componentes minimamente ativos que só incluem correções de segurança. Algumas distribuições permitem combinar um componente de segurança com um componente imutável e essencial para criar uma versão relativamente estável da distribuição.

- Uma "arquitetura" representa uma classe específica do hardware. A expectativa é que as máquinas dentro da classe de uma arquitetura serão suficientemente semelhantes para que possam executar o mesmo binário. As arquiteturas são instâncias específicas das distribuições, por exemplo, "Fedora Core 5 para a arquitetura i386". Como os componentes são subdivisões das distribuições, também há uma instância específica à arquitetura correspondente para cada um deles.
- Pacotes individuais são os elementos que compõem os componentes e, portanto, indiretamente, as distribuições. Os pacotes são normalmente específicos à arquitetura e suas versões são controladas independentemente da distribuição principal e de outros pacotes. A correspondência entre pacotes e distribuições é implícita na maneira como o repositório de rede é configurado.

A existência de componentes que não são mantidos pelo distribuidor (por exemplo, "contrib" do Debian ou "extras" do Fedora) levanta a pergunta de como esses componentes se relacionam com a distribuição básica do SO. Podemos realmente afirmar que eles são "um componente" de uma distribuição específica ou são alguma outra espécie completamente diferente? Da perspectiva do gerenciamento de pacotes, a resposta fica clara: extras são um verdadeiro componente. Eles estão associados a uma distribuição específica e evoluem conjuntamente. A separação do controle é interessante do ponto de vista administrativo, mas não afeta os sistemas de distribuição de pacotes.

RHN: a Red Hat Network

Com o Red Hat tendo saído do mercado Linux consumidor, o Red Hat Network tornou-se a plataforma de gerenciamento de sistemas para o Red Hat Enterprise Linux. Você adquire o direito de acesso ao Red Hat Network mediante uma assinatura. No mínimo, você pode utilizar o Red Hat Network como um poderoso portal Web e uma excelente lista de discussão. Utilizado dessa maneira, o Red Hat Network não é muito diferente das listas de discussão de notificação de correções que foram especificadas por vários fornecedores UNIX durante anos. Mas há mais recursos disponíveis caso esteja pronto para pagar por eles. Acesse rhn.redhat.com para obter as taxas atuais e informações.

O Red Hat Network dispõe de uma bela interface gráfica com o usuário para fazer a cópia de pacotes novos e fornece uma alternativa baseada em linha de comando. Ele permite até que você faça a cópia e instale pacotes novos sem qualquer intervenção humana. Assim que tiver feito o registro, suas máquinas poderão pegar todas as correções para bugs sem jamais você ter de abandonar seu jogo favorito. O inconveniente do registro automático é o fato de o Red Hat decidir de quais atualizações você necessita. Talvez você possa considerar quanto você confia no Red Hat (e nos mantenedores de software cujos produtos eles englobam) em não estragar sua instalação. Dadas algumas interessantes opções oferecidas pelo Red Hat no passado relacionadas a detalhes como qual compilador oferecer, algumas pessoas podem permanecer céticas.

Um meio-termo razoável poderia ser fazer a assinatura de atualizações automáticas para uma máquina da sua empresa. Você poderá então tirar instantâneos dessa máquina em intervalos periódicos para testar como possíveis candidatos para distribuição interna. Certifique-se, porém, de revisar os termos do acordo de licenciamento do Red Hat (disponível em www.redhat.com/licenses) antes de seguir esse caminho. Talvez você fique surpreso de saber que o Red Hat reivindica uma participação proprietária do software de código-fonte aberto distribuído por meio do Red Hat Network, sem mencionar que você concordou em permitir que o Red Hat auditasse seus sistemas à vontade.

APT: a Advanced Package Tool

O APT é um dos sistemas de gerenciamento de pacotes mais maduros. É possível atualizar um sistema inteiro cheio de softwares com um único comando **apt-get** ou mesmo (como ocorre com o Red Hat Network) atualizar suas máquinas continuamente sem intervenção humana.

Como ele se originou do universo do Debian, o APT original só suportava pacotes **.deb**. Mas o APT foi posteriormente portado para o mecanismo de pacotes RPM. Essa versão, conhecida como APT-RPM, está disponível em apt-rpm.org. O sistema **yum**, descrito a partir da página 173, oferece funcionalidades baseadas em RPM semelhantes e é diretamente suportado pelas distribuições Fedora. Ao escolher entre o **yum** e o APT-RPM, a heterogeneidade da sua instalação e as preferências predefinidas das suas distribuições são mais importantes que as distinções técnicas entre os sistemas. Escolha aquele que tiver o melhor suporte e for mais fácil para você.

A primeira regra para uso de **apt-get** em sistemas Debian (e, de fato, para todo o gerenciamento de pacotes Debian) é ignorar a existência de **dselect**, que atua como um front end para o sistema. Ele não é uma má idéia, mas a interface com o usuário é pobre. A documentação do Debian tentará guiá-lo em direção a **dselect**, mas permaneça irredutível.

Se você estiver utilizando **apt-get** para gerenciar uma instalação Debian ou Ubuntu a partir de um espelho padrão, a maneira mais fácil de ver os pacotes disponíveis é visitar a lista mestra em packages.debian.org ou packages.ubuntu.com. As duas distribuições incluem uma interessante interface de pesquisa. Se configurar seu próprio servidor **apt-get** (ver a página 172), naturalmente você saberá quais pacotes disponibilizou e você poderá listá-los da maneira que preferir.

As distribuições comumente incluem pacotes fictícios que só existem para reivindicar outros pacotes como pré-requisitos. **apt-get** faz a cópia e atualiza os pacotes de pré-requisitos conforme necessário, portanto os pacotes fictícios facilitam a instalação ou a atualização de vários pacotes como um bloco. Por exemplo, a instalação do pacote **gnome-desktop-environment** obtém e instala todos os pacotes necessários para executar a interface com o usuário GNOME.

Assim que tiver configurado o arquivo **sources.list** e souber o nome de um pacote desejado, a única coisa que resta fazer é executar **apt-get update** para atualizar o cache de informações de pacotes de **apt-get**. Depois disso, basta executar **apt-get install** *nomeDoPacote* para instalar o pacote. O mesmo comando atualiza um pacote que já foi instalado.

Suponha que queiramos instalar uma nova versão do pacote **sudo** que corrija um bug de segurança. Primeiramente, sempre é mais inteligente fazer um **apt-get update**:

```
$ sudo apt-get update
Get:1 http://http.us.debian.org stable/main Packages [824kB]
Get:2 http://non-us.debian.org stable/non-US/main Release [102B]
...
```

Agora podemos realmente recuperar o pacote. Note que estamos usando **sudo** à medida que recuperamos o novo pacote **sudo** — **apt-get** pode até atualizar pacotes que estão sendo utilizados no momento!

```
$ sudo apt-get install sudo
Reading Package Lists... Done
```

```
Building Dependency Tree... Done
1 packages upgraded, 0 newly installed, 0 to remove and 191 not upgraded.
Need to get 0B/122kB of archives. After unpacking 131kB will be used.
(Reading database ... 24359 files and directories currently installed.)
Preparing to replace sudo 1.6.1-1 (using .../sudo_1.6.2p2-2_i386.deb) ...
Unpacking replacement sudo ...
Setting up sudo (1.6.2p2-2) ...
Installing new version of config file /etc/pam.d/sudo ...
```

Configurando o apt-get

Configurar o **apt-get** é simples e direto; quase tudo o que você precisa saber pode ser encontrado no *APT HOWTO:*
www.debian.org/doc/manuals/apt-howto

O arquivo de configuração mais importante do **apt-get** é **/etc/apt/sources.list**, que informa **apt-get** sobre onde obter seus pacotes. Cada linha especifica o seguinte:

- Um tipo de pacote, atualmente `deb` ou `deb-src` para pacotes no estilo Debian ou `rpm` ou `rpm-src` para RPM
- Um URL que aponta para um arquivo, CD-ROM, servidor HTTP ou servidor FTP em que buscar pacotes
- Uma "distribuição" (na verdade, um nome de versão) que permite entregar múltiplas versões dos pacotes. Distribuidores utilizam isso para distribuições importantes, mas você pode utilizá-la como quiser para sistemas de distribuição internos
- Uma lista potencial dos componentes (categorias de pacotes dentro de uma distribuição)

A menos que você queira configurar seu próprio cache ou repositório APT, a configuração padrão geralmente funciona bem. Se tiver uma conexão de rede razoável, você deverá remover os comentários das linhas dos CD-ROMs da sua distribuição. Se você quiser fazer o download do código-fonte, remova o caractere de comentário das linhas que especificam `deb-src` ou `rpm-src`.

Enquanto estiver editando o arquivo, você deve alterar a identidade do espelho para uma que esteja mais próxima a você; uma lista completa de espelhos do Debian pode ser encontrada em:
www.debian.org/misc/README.mirrors

O Ubuntu mantém uma lista semelhante em wiki.ubuntu.com/Archive.

Para tornar as coisas ainda mais fáceis, uma ferramenta do Debian chamada **netselect-apt** gera automaticamente um arquivo **sources.list**; ela seleciona o espelho mais próximo que pode localizar com base no tempo de ping. **netselect-apt** é parte do pacote **netselect**, disponível a partir do seu espelho mais próximo. (Da maneira como é distribuída, **netselect-apt** está mais ou menos amarrada ao sistema de espelhos do Debian, mas a maioria dos pacotes também funciona bem com o Ubuntu.)

Certifique-se de que security.debian.org ou security.ubuntu.com esteja listado como uma fonte para que você tenha acesso às últimas correções de segurança.

Um exemplo de arquivo /etc/apt/sources.list

O exemplo a seguir utiliza http.us.debian.org para o repositório de arquivos estável, mas também adiciona non-us.debian.org como uma fonte para pacotes fora dos Estados Unidos (isso foi inicialmente importante para pacotes criptográficos, mas não tanto agora). Acrescentamos security.debian.org, a fonte de todas as correções referentes à segurança, e nosso servidor APT local, local-debian-server. Por fim, ativamos a cópia do código-fonte.

```
# Formato geral: tipo uri distribuição [componentes]
deb http://ftp.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free
deb http://local-debian-server/mypackages/ ./
deb-src http://http.us.debian.org/debian stable main contrib non-free
deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb-src http://security.debian.org stable/updates main contrib non-free
```

Os campos *distribuição* e *componentes* ajudam **apt-get** a navegar pela hierarquia do sistema de arquivos do repositório Debian, que tem um layout padronizado. A distribuição-raiz pode ser `stable` para a distribuição mais importante e mais recente, `unstable` (ou `testing`) para a distribuição atual do trabalho em progresso ou o nome de uma distribuição específica como `etch`. Os componentes disponíveis são em geral `main`, `contrib` e `non-free`.

Este exemplo utiliza os repositórios estáveis, que não mudam com tanta freqüência quanto outras fontes. Todos os pacotes mais recentes são incluídos na distribuição `unstable` do Debian. "Unstable" (instável) não (necessariamente) significa que os próprios pacotes são instáveis, mas, em vez disso, que a composição da distribuição inteira é volátil. Atualizações semanais de mais de 100 MB são comuns.

As linhas no arquivo **sources.list** são consultadas na ordem, assim, teoricamente, você pode colocar as linhas `unstable` e `testing` no fim do arquivo a fim de dar precedência às versões estáveis. O problema dessa abordagem é que por causa do monitoramento de dependência do APT, um pacote instável pode inserir versões instáveis atualizadas de todos os pacotes dos quais ele depende. Esses pacotes por sua vez podem inserir versões instáveis de todos os seus companheiros e assim por diante. Uma maçã ruim estraga todo o barril; não brinque com pacotes instáveis nos seus sistemas de produção.

Se precisar incorporar um pacote a partir do mundo instável ao seu ambiente de produção, a maneira correta de fazer isso é utilizar um "backport" que foi recompilado na distribuição estável para que ele conviva com as bibliotecas estáveis. Para localizar esses backports e outras delícias, confira o sistema de pesquisa APT hospedado em www.apt-get.org. Muitos dos pacotes com backports (em oposição a meros ponteiros) podem ser encontrados no site www.backports.org, de Norber Tretkowski. Os backports nesse repositório são de excelente qualidade e têm dependências externas mínimas.

Utilizando proxies para tornar o apt-get escalonável

Se você pretende utilizar **apt-get** num grande número de máquinas, provavelmente vai querer colocar pacotes em cache local — fazer a cópia de uma cópia de cada pacote para cada máquina não é um emprego sensato de largura de banda. Talvez você também precise direcionar seu **apt-get** através de um proxy caso o seu firewall exija isso.

O **apt-get** utiliza protocolos HTTP e FTP simples, assim você pode utilizar qualquer proxy Web existente instalado no seu sistema. **apt-get** honra a variável de ambiente http_proxy, mas você também pode configurar um proxy explícito com uma linha em **/etc/apt/apt.conf**:

```
Acquire::http::Proxy "http://servidorProxy:8080/
```

Uma alternativa a um proxy Web genérico é um pequeno aplicativo chamado **apt-proxy**. Apesar do nome, ele não é um verdadeiro proxy, mas sim um aplicativo que constrói um cache de pacotes aplicando **rsync** a eles por meio do verdadeiro servidor APT. **apt-proxy** está disponível em:

sourceforge.net/projects/apt-proxy

Configurando um servidor APT interno

Em vez de utilizar um proxy, você também poderá configurar o seu próprio servidor APT autônomo e apontar seus clientes internos para ele. Esse modelo permite ajustar os pacotes que você oferece a seus clientes, enviar facilmente atualizações (basta instalar novas versões no servidor), distribuir seus próprios aplicativos na forma de pacotes e, sobretudo, fornecer suas próprias versões das distribuições.

Consulte a página 499 para dicas sobre como configurar o servidor Web Apache.

Como o **apt-get** utiliza protocolos padrão (HTTP ou FTP) para fazer a cópia dos seus pacotes, configurar um servidor APT é simplesmente uma questão de estabelecer um servidor Web ou FTP que forneça conteúdo apropriado.[4] Dada a ampla disponibilidade de servidores e ferramentas relacionadas ao HTTP, esse protocolo provavelmente é a escolha mais simples para uso com o APT.

Todos os pacotes no servidor podem estar em um diretório ou serem organizados em uma hierarquia como nos espelhos do Debian e do Ubuntu.

Além de fornecer os arquivos de pacote, você deve gerar dois arquivos de resumo: **Packages.gz** e **Contents.gz**. **Packages.gz** devem ser descompactados com uma lista **gzip** dos pacotes no servidor e suas dependências. **apt-get update** utiliza essa lista para determinar o complemento disponível dos pacotes. **Contents.gz** mapeia arquivos brutos para os pacotes que os contêm; na verdade ele não é utilizado pelo **apt-get**. O comando **apt-ftparchive**, incluído no pacote **apt-utils**, gera automaticamente esses dois arquivos de resumo.

Assim que você os tiver criado, o restante é fácil. Uma linha como:

```
deb http://local-server/mypackages/ ./
```

nos arquivos **/etc/apt/sources.list** das máquinas cliente conectam **apt-get** ao seu servidor local. Execute **apt-get update** em cada cliente, depois use **apt-get** normalmente.

Para distribuir código-fonte, bem como pacotes binários, basta colocar os pacotes-fonte no servidor. Diferentemente de RPM, que possui um SRPM equivalente para pacotes-fonte, o Debian distribui os pacotes-fonte em três partes: o arquivo **.tar.gz** simples, um arquivo **.diff.gz** opcional (utilizado por empacotadores para mostrar as alterações que eles fizeram em relação à base do código original) e um arquivo **.dsc** que descreve o pacote. O código-fonte equivalente de **Packages.gz** é **Sources.gz**; ele também é gerado por **apt-ftparchive**.

A linha **sources.list** do exemplo anterior não especifica um parâmetro de distribuição. Se preferir, você poderá usar nomes de distribuição como uma forma de controle interno de versão. Basta posicionar cada conjunto de pacotes que você quer definir como uma "versão" em um subdiretório próprio e alterar `./` no arquivo **sources.list** para o nome ou número da versão.

---

4. Há outras opções também. Por exemplo, você pode configurar um servidor NFS contendo os arquivos de atualização ou gravar um DVD que você pode levar para cada sistema.

Costuma ser útil criar distribuições genericamente chamadas como "teste" e "produção" que são análogas às próprias distribuições "estáveis" e de "teste" do Debian. Use links simbólicos no servidor a fim de apontar esses nomes para distribuições específicas. Mais tarde você pode redefinir o conteúdo das distribuições alterando apenas os alvos dos links. Por exemplo, depois que tiver certeza de que uma distribuição de teste está pronta para implantação, você poderá apontar o link de produção no mesmo diretório. Os clientes se sincronizam com a distribuição de produção e, então seguida, selecionam automaticamente as alterações.

Automatizando o apt-get

É possível executar **apt-get** segundo um cronograma regular a partir de **cron**. Mesmo que você não instale pacotes automaticamente, talvez você queira executar **apt-get update** regularmente para manter os resumos de pacotes atualizados.

apt-get dist-upgrade faz a cópia e instala novas versões de quaisquer pacotes atualmente instalados na máquina local. **dist-upgrade** é semelhante a **upgrade**, mas tem um tratamento de dependência um pouco mais inteligente. **dist-upgrade** talvez queira excluir alguns pacotes que ele vê como irreconciliáveis e incompatíveis com o sistema atualizado, portanto esteja preparado para possíveis surpresas.

Se você realmente quiser mexer com fogo, faça com que as máquinas realizem a atualização de uma maneira não-supervisionada utilizando a opção **-yes**. Responde a quaisquer perguntas de confirmação que **apt-get** talvez faça com um entusiasmado "Yes!"

Provavelmente não é uma boa idéia realizar atualizações automatizadas diretamente de um espelho da distribuição. Mas, em concerto com seus próprios servidores APT, pacotes e sistemas de controle de versão, essa é uma maneira perfeita de manter clientes sincronizados. Um pequeno script de shell como o mostrado a seguir manterá um computador atualizado com seu servidor APT:

```
#!/bin/sh
apt-get update
apt-get -yes dist-upgrade
```

Chame esse script por meio de um serviço **cron** caso queira rodá-lo a cada noite. Você também pode referenciá-lo a partir de um script de inicialização de sistema para fazer a máquina atualizar-se no momento da inicialização. Veja o Capítulo 8, *Processos Periódicos*, para obter maiores informações sobre **cron**; veja o Capítulo 2, *Inicialização e Desligamento*, para informações adicionais sobre scripts de inicialização.

Se executar atualizações a partir do **cron** em muitas máquinas, uma boa idéia é utilizar randomização de tempo para certificar-se de que ninguém tente atualizar de uma só vez. O pequeno script em Perl na página 355 pode ajudá-lo nesta tarefa.

Se você não confia muito na fonte dos pacotes, considere fazer a cópia automática de todos os pacotes alterados sem instalá-los. Use a opção **--download-only** de **apt-get** para solicitar esse comportamento, depois reveja os pacotes manualmente e instale aqueles que deseja atualizar. Os pacotes obtidos via download são colocados em **/var/cache/apt**, e, ao longo do tempo, esse diretório pode acabar ficando muito grande. Elimine os arquivos que não são utilizados deste diretório com **apt-get autoclean**.

Se utilizar uma distribuição do Debian formalmente designada como estável, podemos recomendar atualizações automáticas sem muitas reservas. Alterações na distribuição estável geralmente são limitadas a atualizações de segurança, e a integração foi bem testada. O único potencial problema é que talvez você não queira atualizar automaticamente quando uma nova distribuição importante ocorre. Para evitar esse problema, especifique um nome explícito da distribuição em vez da palavra-chave `stable` em **sources.list**.

yum: gerenciamento de versão para o RPM

yum, o Yellowdog Updater, Modified, é um gerenciador de metapacotes baseado no RPM.[5] Pode ser um pouco injusto chamá-lo de clone do **apt-get**, mas ele é de fato muito semelhante tematicamente e na implementação, embora mais lento e mais limpo na prática. O **yum** é o sistema oficial de gerenciamento de pacotes para o Fedora e vem pré-instalado em muitas outras distribuições. Se necessário, você pode obter a versão mais recente em linux.duke.edu/yum.

Como ocorre com o **apt-get**, um comando do lado servidor (**yum-arch**) compila um banco de dados das informações de cabeçalho a partir de um grande conjunto de pacotes (freqüentemente uma distribuição inteira). O banco de dados de cabeçalhos é então compartilhado juntamente com os pacotes por meio de HTTP ou FTP. Os clientes utilizam o comando **yum** para buscar e instalar pacotes; **yum** descobre restrições de dependência e realiza qualquer trabalho adicional necessário para completar a instalação dos pacotes solicitados. Se um pacote solicitado depender de outro pacote, o **yum** também faz a cópia e instala esses pacotes.

---

5. Não confunda com o Yum Fish Bait da Live Prey Technology (LPT), yum3x.com.

As semelhanças entre **apt-get** e **yum** se estendem às opções de linha de comando que eles entendem. Por exemplo, **yum install foo** faz a cópia e instala a versão mais recente do pacote foo (e suas dependências, se necessário). Mas há pelo menos uma diferença traiçoeira: **apt-get update** atualiza o cache das informações de pacotes do **apt-get**, mas **yum update** atualiza cada pacote no sistema (ele é análogo a **apt-get upgrade**). Para aumentar a confusão, também há **yum upgrade**, que é o mesmo que **yum update**, mas com processamento de obsolescência ativado.

yum não compara nomes de pacote parciais a menos que você inclua caracteres de shell curingas (como * e ?) explicitamente a fim de solicitar esse comportamento. Por exemplo, **yum update 'perl*'** atualiza todos os pacotes cujo nome inicia com "perl". Lembre-se de citar os caracteres globais de modo que o shell não interfira com eles.

Diferentemente de **apt-get**, **yum** assume o padrão de validar seu cache de informações de pacote contra o conteúdo do repositório de rede toda vez que você o executa. Utilize a opção **-C** para impedir a validação e **yum makecache** para atualizar o cache local (ele demora um pouco para executar). Infelizmente, **-C** não faz muito para aprimorar o lento desempenho do **yum**.

O arquivo de configuração do **yum** é **/etc/yum.conf**. Esse arquivo inclui opções gerais e ponteiros para repositórios de pacotes. Múltiplos repositórios podem estar ativos ao mesmo tempo e cada repositório pode ser associado a múltiplos URLs.

O Fedora inclui metadados com o formato **yum** em suas árvores de distribuição padrão, assim basta os usuários do Fedora apontarem seus arquivos **yum.conf** para o espelho mais próximo e atualizarem à vontade. Nesse aspecto, o RHEL não segue o exemplo, provavelmente para encorajar as vendas de assinaturas de Red Hat Network.

11.5 CONTROLE DE VERSÃO

Erros são um fato da vida. Portanto, é importante monitorar as alterações que você faz de modo que quando essas alterações causarem problemas, você possa facilmente reverter para uma boa e conhecida configuração. Definir distribuições internas formais torna fácil fazer reversões, mas é uma solução complicada que só funciona bem em um nível grosseiro de granularidade. E se você só precisa ajustar um ou dois arquivos?

Nesta seção, discutimos algumas maneiras comuns de gerenciar alterações no nível de arquivos individuais. Esses métodos são complementares às questões maiores do controle interno de distribuição e vice-versa — escolha várias ferramentas que atendam às suas necessidades locais e à complexidade da sua instalação.

Criação de arquivos de backup

Criar cópias de backup dos arquivos que você modifica é uma tradição administrativa consagrada pelo tempo e uma que os scripts de localização e os administradores individuais devem seguir. Os arquivos de backup permitem reverter configurações para um estado anterior, mas, talvez mais importante, eles também permitem comparar, utilizando **diff**, a versão atual com a versão anterior de um arquivo a fim de descobrir as alterações que foram feitas.

Você pode criar arquivos de backup mais apropriadamente separando o arquivo original (com **mv**) e utilizando um novo nome como **nomeDoArquivo.old** ou **nomeDoArquivo.bak** para então copiá-lo de volta para seu nome original. Utilize a opção **-p** para **cp** a fim de preservar as configurações de atributo do arquivo. Depois de atualizar a versão ativa do arquivo, a data/hora da última modificação desse arquivo reflete a alteração mais recente, enquanto a data/hora da última modificação do arquivo de backup reflete a data/hora da última modificação anterior. (Se você simplesmente utilizasse **cp**, os dois arquivos teriam a mesma data/hora de modificação.) Separar o arquivo original também trata o caso em que um processo ativo tem uma referência aberta ao arquivo: as alterações que você faz na cópia agora ativa não são vistas até o arquivo ser fechado e reaberto.

Os sistemas regularmente copiados para fitas de backup ainda podem se beneficiar do uso de arquivos de backup criados manualmente. A recuperação de um arquivo de backup é mais rápida e mais fácil do que a recuperação por meio de uma fita, e os backups manuais preservam uma camada adicional de histórico.

Sistemas de controle formais de revisão

Arquivos de backup são muito úteis, mas eles tendem a ser mais práticos em pequenos sites. No próximo nível de complexidade e robustez estão os sistemas formais de controle de revisão, que são pacotes de software que monitoram, arquivam e fornecem acesso a múltiplas revisões dos arquivos. Esses pacotes se originam do mundo do desenvolvimento de softwares, mas também são bem úteis para os administradores de sistemas.

Sistemas de controle de revisão resolvem vários problemas. Primeiro, eles fornecem uma maneira organizada de rastrear a história das modificações em um arquivo de modo que essas alterações possam ser entendidas no contexto e para que versões anteriores possam ser recuperadas. Segundo, eles estendem o conceito de controle de versão além do nível de arquivos individuais. O controle de versão de grupos relacionados de arquivos pode ser feito conjuntamente de uma maneira que leva em consideração suas interdependências. Por fim, o sistema de controle de revisão coordena as atividades dos múltiplos

editores para que condições de concorrência (*race conditions*) não possam resultar na perda permanente das alterações de qualquer pessoa[6] e para que alterações incompatíveis de múltiplos editores não se tornem ativas simultaneamente.

O sistema de controle de revisão mais simples em uso comum é o RCS, apropriadamente chamado Revision Control System. Ele está disponível há décadas e vem pré-instalado em muitos sistemas.

Uma outra opção é um sistema de código-fonte aberto chamado CVS (Concurrent Versions System) que acrescenta alguma funcionalidade ao RCS. Ele suporta um modelo distribuído (para uso com um servidor remoto) e melhor suporte para o multidesenvolvimento. Uma série de instalações tem utilizado o CVS para tarefas administrativas; em particular, a capacidade cliente/servidor pode ser bem útil. Infelizmente, o CVS tem algumas deficiências conceituais que o tornam uma faca de dois gumes.

Um produto mais recente (mas já testado pelo tempo) no mercado do código-fonte aberto é o Subversion, um sistema que fornece todas as vantagens do CVS, mas que parece ter um comportamento padrão muito mais sensato. A principal desvantagem dele para administração de sistemas é que o modelo do projeto é mais ou menos centrado em diretório. Entretanto, ele continua a ser um sistema muito interessante e uma escolha razoável para uso administrativo.

Os últimos poucos anos testemunharam um "boom" dos sistemas de controle de versão de código-fonte aberto e as escolhas disponíveis aumentaram significativamente. Os principais concorrentes entre os sistemas mais recentes incluem Monotone, Darcs, Arco e Bazaar-NG. Todos são sistemas interessantes, mas parecem compartilhar uma tendência ao trabalho descentralizado: múltiplas bifurcações, múltiplos repositórios, uma grande quantidade de desenvolvimento paralelo. Em nossa opinião, o tradicional modelo de "repositório central" é mais apropriado para o controle de versão no contexto do sistema de administração.

Estão disponíveis também vários sistemas comerciais para controle de revisão. Talvez você já tenha acesso a um deles se trabalhar em uma empresa de desenvolvimento e talvez esteja tentado a adaptá-lo para dados administrativos. Seja prudente; nossa experiência é a de que esses sistemas comerciais normalmente são exagerados para uso administrativo.

Se estivesse iniciando do zero, recomendaríamos começar com o RCS para que você tenha uma idéia geral sobre o controle de revisão. Se quiser criar um repositório centralizado para informações sobre a administração de sistemas, o Subversion é sua escolha certa.

RCS: o Revision Control System

RCS (Revision Control System) é um dos mais antigos aplicativos UNIX existentes. Ele é, na verdade, um sistema relativamente simples. Ele opera no nível dos arquivos individuais e armazena o histórico de revisão de cada arquivo em um arquivo sombra separado. O nome do arquivo espelhado é o mesmo do original, mas com os caracteres **,v** anexados a ele. Se, por exemplo, você colocou **/etc/syslog.conf** sob controle do RCS, este manterá suas revisões em **/etc/syslog.conf,v**.

Para minimizar a complexidade, o RCS procura um diretório chamado **RCS** no mesmo diretório do arquivo original. Se ele existir, o RCS seqüestra o arquivo **,v** ali em vez de deixá-lo visível. As listagens de diretórios se tornam muito mais limpas dessa maneira, pois muitos arquivos podem compartilhar um diretório **RCS**. Trata-se de um recurso extraordinário e que recomendamos enfaticamente.

O modelo básico utilizado pelo RCS é que você faz um "check-out" nos arquivos antes de modificá-los. Você então faz um "check-in" nas suas alterações a fim de confirmá-las. Dessa forma, os únicos comandos RCS que você realmente precisa conhecer são **co** para fazer o check-out dos arquivos, **ci** para fazer o check-in, e **rcs**, que realiza diversas tarefas de faxina. Se utilizar o editor **emacs**, você poderá evitar completamente as ferramentas de linha de comando porque o suporte ao RCS é integrado.

Para iniciar o monitoramento RCS de um arquivo, você primeiro faz o check-in:

```
# ci -u syslog.conf
RCS/syslog.conf,v <-- syslog.conf
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> This is the syslog configuration file.
```

A opção **-u** faz com que **ci** faça imediatamente o check-out do arquivo **syslog.conf** em um estado desbloqueado (não-editável). Se você omitisse esse flag, **ci** faria o check-in do arquivo e eliminaria a cópia original, o que provavelmente não é o que você quer.

Toda vez que quiser alterar um arquivo controlado por RCS, você terá de fazer seu check-out e bloqueá-lo com **co -l**:

---

6. Suponha, por exemplo, que os administradores de sistemas Alice e Roberto tenham editado o mesmo arquivo e que cada um deles faça algumas alterações. Alice salva o arquivo primeiro. Quando Roberto salva sua cópia do arquivo, ele sobrescreve a versão de Alice. Se Alice tiver saído do editor, suas alterações serão completamente perdidas e irrecuperáveis.

```
# co -l syslog.conf
RCS/syslog.conf,v --> syslog.conf revision 1.2 (locked)
done
```

Essa operação instrui o RCS de que está em vias de modificar o arquivo. O RCS não permitirá que ninguém mais faça o check-out do arquivo até você fazer o check-in.

O RCS remove permissão de gravação dos arquivos desbloqueados como um lembrete para não editá-los até você ter feito o check-out deles adequadamente. Um erro comum é carregar um arquivo no seu editor, fazer as alterações sem perceber que você precisava fazer o check-out do arquivo até seu editor se recusar a salvar as alterações. Para corrigir isso, simplesmente pause o editor ou exiba uma outra janela de shell, execute o **co -l** apropriado e então tente novamente a operação de salvar. Você poderia utilizar **chmod** no arquivo para ele se tornar gravável, mas isso subverteria e confundiria o RCS.

Teoricamente, o bloqueio RCS impede que duas pessoas diferentes modifiquem o arquivo ao mesmo tempo. Na prática, você tem de ser root para modificar arquivos de sistema; assim, qualquer um com privilégios de **sudo** pode modificar um arquivo logo que ele sofrer check-out como root. Mas, se um segundo administrador tentar um outro **co -l**, o RCS perceberá que já existe uma versão gravável e imprimirá uma mensagem. Os administradores de sistemas devem se acostumar a sempre tentar fazer o check-out dos arquivos controlados por RCS que eles querem modificar. O fato de um arquivo já ser gravável significa: "Pare! Alguma outra pessoa já fez o check-out desse arquivo".

Ocasionalmente você poderá descobrir que alguém modificou um arquivo e o deixou bloqueado — ou até pior, subverteu o RCS e modificou o arquivo sem bloqueá-lo. Você pode rever as alterações feitas pelo perpetrador com **rcsdiff**, que é uma versão de **diff** que reconhece RCS: Por exemplo:

```
# rcsdiff syslog.conf
===================================================================
RCS file: RCS/syslog.conf,v
retrieving revision 1.3
diff -r1.3 syslog.conf 4c4
< define(LOGHOST,moonbase)
---
> define(LOGHOST,spacelounge)
```

Como último recurso, você pode desfazer o bloqueio com o comando **rcs -u** *nomeDoArquivo*. Esse comando solicita a introdução de uma explicação sobre seus atos e envia um e-mail para o usuário que havia anteriormente bloqueado o arquivo (em geral root, infelizmente).

Quando estiver satisfeito com suas alterações em um arquivo que sofreu check-out, faça o check-in com **ci -u**. Você será solicitado a fornecer um comentário que descreve o que você acabou de fazer. Não pule esse passo, mas também não escreva um romance. Daqui a alguns anos quando você estiver tentando descobrir por que fez uma alteração particular, os comentários úteis podem salvar sua vida.

```
# ci -u syslog.conf
RCS/syslog.conf,v <-- syslog.conf
new revision: 1.3; previous revision: 1.2
enter log message, terminated with single '.' or end of file:
>> Started logging debug messages to track down SSH problem
>> .
done
```

Você pode inspecionar o histórico de revisão de um arquivo com o comando **rlog**:

```
# rlog syslog.conf
RCS file: RCS/syslog.conf,v
Working file: syslog.conf
head: 1.3
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 3; selected revisions: 3
description:
----------------------------
revision 1.3
date: 2002/01/10 00:44:58; author: adam; state: Exp; lines: +1 -0
```

```
Started logging debug messages to track down problem
----------------------------
revision 1.2
date: 2000/07/19 08:23:10; author: evi; state: Exp; lines: +2 -0
Changed log destination to new logmaster
----------------------------
revision 1.1
date: 1998/03/14 11:13:00; author: matthew; state: Exp;
Initial revision
=============================================
```

Se quiser verificar o aspecto do arquivo antes de você ter mudado para o novo logmaster, você pode fazer o check-out da revisão 1.2 do arquivo com a opção **-r** do comando **-co**:

```
# co -r1.2 syslog.conf
RCS/syslog.conf,v --> syslog.conf
revision 1.2
done
```

Esse comando substitui o arquivo **syslog.conf** atual pela versão antiga; portanto, certifique-se de fazer regularmente um **co** quando tiver terminado; caso contrário, você (e **syslogd**) poderá ficar muito confuso. Uma outra opção seria **co -r1.2 -p syslog.conf**, que envia o conteúdo da versão solicitada para a saída padrão; infelizmente, não há como fazer o check-out do arquivo sob um nome de arquivo diferente. Jamais faça o check-out de cópias bloqueadas de revisões mais antigas (via **co -l**), pois essa operação criará ramificações na árvore de versões. Ramificações de uma versão são ocasionalmente úteis para o código-fonte, mas quase nunca são utilizadas para o trabalho de um administrador de sistemas; para tornar sua vida mais simples, simplesmente ignore toda a documentação do RCS que trata delas.

Informações adicionais sobre o RCS podem ser encontradas em

www.cs.purdue.edu/homes/trinkle/RCS/

CVS: o Concurrent Versions System

O RCS tem algumas fraquezas significativas. Você deve ter cuidado para não pisar no pé dos seus colegas ao editar arquivos. Você precisa seguir um conjunto relativamente específico de procedimentos. Você não pode modificar de uma maneira segura vários arquivos de uma só vez sem expor o sistema a suas modificações não-concluídas. Essas deficiências (e outras) motivaram o projeto de última geração do Concurrent Versions System, que atualmente é o sistema de controle de revisão mais amplamente utilizado nos sistemas UNIX e Linux.

Uma das principais idéias por trás do CVS é que todos os arquivos de projeto (e suas versões históricas) são armazenados em uma localização central. Como ocorre com o RCS, você faz um check-out das cópias dos arquivos para trabalhar neles e então faz o check-in nos arquivos quando tiver terminado. O aspecto interessante sobre o CVS é que ele não tem o conceito de check-outs "bloqueados" e "desbloqueados"; várias pessoas podem fazer o check-out e modificar os mesmos arquivos ao mesmo tempo (daí "concorrente").

Como CVS não impede que múltiplas pessoas modifiquem um arquivo, ele precisa fornecer uma maneira para que as alterações sejam integradas quando é feito o check-in nos arquivos. Essa mesclagem normalmente ocorre automaticamente, mas só funciona para arquivos de texto e somente se as várias alterações forem compatíveis. Se não forem, o CVS conta com a pessoa que executa o check-in para resolver os conflitos manualmente. No caso de alterações incompatíveis em um arquivo de texto, o CVS insere comentários úteis que mostram onde o conflito ocorreu.

Muita teoria. Parece legal, e de fato é — mesmo a mesclagem funciona bem. Mas o CVS ainda apresenta alguns problemas:

- Ele não suporta "confirmações atômicas". Se duas pessoas tentarem fazer o check-in de uma grande modificação que afeta múltiplos arquivos, as duas versões podem acabar sendo aceitas pela metade, com uma seleção aleatória de conflitos a resolver atribuída a cada operador. Nada bom.
- Para renomear um arquivo, você deve copiá-lo sob o novo nome e excluir o arquivo original, perdendo assim todo o histórico passado do arquivo. De maneira semelhante, não é possível renomear diretórios, exceto copiando-os e removendo as versões originais.
- Atributos de arquivo não estão sob controle de revisão. Eles permanecem fixos nas configurações que um arquivo tinha quando foi feito o check-in pela primeira vez.

Contudo, o CVS é utilizado por muitos projetos de software de código-fonte aberto. Não tanto porque ele é ilimitadamente maravilhoso, mas mais por causa da falta anterior de alternativas viáveis. Essa situação, porém, mudou à medida que vários grupos trabalham em substitutos do CVS. Hoje, há inúmeras escolhas e é possível que ocorram "guerras santas" sobre qual sistema de controle de revisão será mais bem-sucedido que o CVS. O site Web *Better SCM Initiative,* de Shlomi

Fish (better-scm.berlios.de), introduz a maioria dos candidatos e apresenta uma comparação sistemática (embora sem data) dos seus recursos.

A seguir há uma breve visão geral dos comandos CVS mais importantes do ponto de vista do usuário. O primeiro passo para modificar um projeto é efetuar o login no servidor e verificar o módulo em que você quer trabalhar. Aqui trabalhamos em um módulo chamado **sort**.

```
$ cvs -d :pserver:nomeDoUsuário@nomeDoServidor:/caminho/para/o/repositório login
CVS password: <senha>
$ cvs -d :pserver:nomeDoUsuário@nomeDoServidor:/caminho/para/o/repositório co sort
```

Aqui, **pserver** é o método de acesso utilizado para contatar o repositório, que nesse caso é um servidor de senha CVS dedicado. A operação **login** verifica a senha com o servidor e cria uma cópia dela para uso em transações posteriores. A operação **co** é diretamente análoga àquela do RCS.

Você agora pode inserir sua cópia local do diretório **sort** e editar os arquivos. Quando for o momento de fazer o check-in nos arquivos para o repositório CVS, você não precisa utilizar a opção **-d**, porque o CVS já criou uma cópia de todas as informações necessárias de localização no subdiretório **sort/CVS**.

```
$ cd sort
$ vi foo.c
$ cvs commit foo.c -m "Added more efficient sort routine"
```

Se você estiver trabalhando temporariamente na sua cópia do módulo e quiser atualizar suas cópias locais dos arquivos que outras pessoas modificaram (e das quais fizeram o check-in) a partir do seu check-out, utilize o comando **cvs update**. A opção **-d** significa incluir todos os subdiretórios e **-P** solicita ao CVS a remoção de quaisquer diretórios vazios:

```
$ cvs update -dP
```

Tenha em mente que embora você não faça o check-in nos seus arquivos para o repositório central, esse é um tipo de integração. Sempre há a possibilidade de as alterações dos outros usuários entrarem em conflito com as suas. O CVS deixará que você saiba se há conflitos que você precisa resolver.

Subversion: o CVS feito corretamente

Embora o CVS seja atualmente o sistema predominante de controle de revisão, sugerimos que os administradores que estão começando a utilizar o RCS pulem diretamente para o Subversion. Esse pacote foi escrito por Karl Fogel e Jim Blandy, que em 1995 iniciaram a Cyclic Software para vender contratos de suporte ao CVS. Em 2000, eles foram contratados pela CollabNet para escrever um substituto de código-fonte aberto para o CVS. O Subversion é o resultado; depois de uma longa gestação, a versão 1.0 foi distribuída em fevereiro de 2004.

O Subversion tem todos os recursos "ausentes" mencionados anteriormente, mas não sacrifica a clareza e a usabilidade. Como no CVS, um repositório centralizado armazena todos os arquivos de controle de versão. O Subversion pode tratar arquivos binários e é mais rápido que o CVS.

Por padrão, o servidor Subversion é um módulo no servidor Web Apache 2.x. Excelente para o desenvolvimento de software distribuído, mas talvez não tão bom para usos administrativos. Felizmente, o pessoal do Subversion fornece um segundo tipo de servidor na forma de um daemon chamado **svnserve**. Você pode executar o **svnserve** a partir do seu diretório inicial ao testar o Subversion, mas no uso de produção ele deve ter sua própria conta de usuário e é executado a partir de **inetd**.

A distribuição inicial do Subversion utilizava o banco de dados Berkeley DB como o armazenamento de apoio ao seu repositório, mas o Subversion 1.1 adiciona suporte a um sistema alternativo conhecido como FSFS. Os dois armazenamentos têm vantagens e desvantagens. Um das principais distinções é que o Berkeley DB conta com uma semântica de E/S de memória mapeada e, portanto, tem fobia do NFS.[7] Os repositórios do Subversion que utilizam o Berkeley DB devem ser locais à máquina em que o servidor Subversion executa. Repositórios FSFS não têm essa limitação. Acesse subversion.tigris.org para outras vantagens e desvantagens.

Configurar o repositório é fácil. Por exemplo, os seguintes passos criam um novo repositório do Subversion chamado **admin**:

```
# cd /home/svn
# mkdir repositories
# cd repositories
# svnadmin create admin
# chmod 700 admin
```

7. Visite www.sleepycat.com para informações adicionais sobre essa questão.

Se quiser usar o formato FSFS para seu repositório em vez do Berkeley DB padrão, adicione uma opção **--fs-type=fsfs** ao comando **svnadmin create**. Faça uma escolha inteligente; o formato de repositório é difícil de alterar depois do fato.

Se examinar dentro do diretório **admin**, você descobrirá uma estrutura bem organizada de repositórios, incluindo um arquivo **README**. O arquivo de configuração **svnserve.conf** pode ser encontrado no subdiretório **conf**. Esse arquivo instrui o daemon de servidor a fornecer acesso ao novo repositório. Eis uma configuração de exemplo apropriada para arquivos administrativos:

```
[general]
anon-access = none
auth-access = write
password-db = passwd
realm = The Sysadmin Repository
```

Como um dos objetivos de projeto do Subversion foi facilitar a colaboração entre as pessoas em diferentes instalações, ele tem um modelo de controle de acesso que é separado daquele do sistema operacional. O arquivo **passwd** (no mesmo diretório) contém uma lista dos usuários e suas senhas em texto simples (!). A senha em texto simples é realmente um problema, mas elas nunca são transmitidas pela rede. Elas também nunca são digitadas de memória pelos usuários, portanto você também pode atribuir senhas que são suficientemente longas e aleatórias para serem seguras. Por exemplo:

```
[users]
tobi = lkadslfkjasdljkhe893Buhau7623rhkdfndf
evi = 09uqalkhlkasdgfprghkjhsdfjj83yyouhfuhe
fritz = kd939hjahkjaj3hkuyasdfaadfk3ijdkjhf
```

Naturalmente, as permissões no arquivo **passwd** devem ser configuradas restritivamente.

Tudo o que resta a fazer é iniciar o servidor no novo repositório:

```
# svnserve --daemon --root /home/svn/repositories
```

Como um usuário não-privilegiado, você agora verifica os respositórios de arquivos **admin** a partir de qualquer lugar na rede:

```
$ svn checkout --username tobi svn://server.atrust.com/admin checkout
Authentication realm: <svn://server.atrust.com:3690> The Sysadmin Repository
Password for 'tobi': <senha>
```

Ao inserir a senha pela primeira vez, o Subversion armazena uma cópia em um diretório **.subversion** que ele cria no seu diretório inicial. Para adicionar ou mover arquivos dentro da sua cópia local do projeto, use o comando **svn**:

```
$ cd checkout
$ vi foo.c
$ svn add foo.c
```

Depois de terminar, confirme suas alterações no repositório:

```
$ svn commit -m "Initial checkin; added foo.c"
```

Não é necessário listar os arquivos alterados que você quer confirmar, embora você possa fazer isso se desejar; o **svn** descobrirá esses arquivos por conta própria. Se omitir a opção **-m**, o **svn** inicia um editor para você poder editar a mensagem de confirmação.

Para obter as últimas atualizações a partir do repositório, execute **svn update** dentro do projeto. Como no CVS, o Subversion realiza uma operação de mesclagem em quaisquer arquivos modificados tanto na sua cópia local do projeto como no repositório mestre. Os arquivos com conflitos insolúveis são marcados como "conflicted" e o Subversion não permite que você faça o check-in até corrigir os problemas e informar o Subversion de que os conflitos foram resolvidos:

```
$ svn resolved foo.c
```

Se quiser saber quem alterou determinadas linhas em um arquivo, você pode solicitar que o Subversion aponte o culpado:

```
$ svn blame bar.c
```

Esse comando imprime uma versão anotada do arquivo que mostra quando e por quem cada linha foi modificada pela última vez. (Aqueles com uma natureza mais tolerante e otimista podem utilizar o sinônimo **svn praise**.) Também é fácil obter diffs relativos a uma data ou versão particular. Por exemplo, se quiser saber o que mudou em **foo.c** desde 2 de junho de 2006, o comando a seguir o informará disso:

```
$ svn diff -r "{2006-06-02}" foo.c
```

Você pode fazer a cópia da versão mais recente do Subversion em subversion.tigris.org. A documentação padrão é o livro *Version Control with Subversion*, publicado pela O'Reilly. O texto integral está disponível on-line em svnbook.red-bean.com.

O Subversion também tem uma excelente interface gráfica Windows; acesse tortoisesvn.tigris.org. Utilizamos essa interface para gerenciar os arquivos-fonte deste livro.

11.6 LOCALIZAÇÃO E CONFIGURAÇÃO

Adaptar computadores ao seu ambiente local é um dos principais campos de batalha da administração de sistemas: informar o sistema sobre todas as impressoras disponíveis na rede, iniciar o daemon de licenciamento especial, adicionar a tarefa **cron** que limpa o diretório **/scratch** uma vez por semana, integrar o suporte ao scanner especial que eles utilizam no departamento gráfico etc. Cuidar dessas questões de uma maneira estruturada e reproduzível é o objetivo central do pensamento arquitetônico.

Mantenha em mente estes pontos:

- Usuários não têm privilégios de root. Qualquer necessidade de privilégios de root durante operações normais é um fato suspeito e provavelmente indica que há algo suspeito na sua configuração local.
- Usuários não destroem o sistema intencionalmente. Projete a segurança interna de modo que ela proteja contra erros não-intencionais e contra a disseminação dos privilégios administrativos.
- Usuários com mau comportamento em aspectos menores devem ser entrevistados antes de serem punidos. Usuários costumam responder a procedimentos administrativos ineficientes tentando subvertê-los, portanto é prudente considerar a possibilidade de que a não-obediência é uma indicação de problemas arquitetônicos.
- Seja centralizado no cliente. Converse com os usuários e pergunte quais tarefas eles acham difícil. Descubra maneiras de tornar essas tarefas mais simples.
- Suas preferências pessoais são apenas suas. Deixe que os usuários tenham suas próprias preferências. Ofereça escolhas onde quer que possível.
- Quando suas decisões administrativas afetam a experiência dos usuários do sistema, esteja ciente das razões para suas decisões. Deixe que suas razões tornem-se conhecidas.
- Mantenha sua documentação local em dia e facilmente acessível. Consulte a página 638 para informações adicionais sobre esse tópico.

Organizando sua localização

Se sua instalação tiver milhares de computadores e cada um deles tiver uma configuração própria, você gastará a maior parte de seu tempo tentando descobrir por que um computador apresenta determinado problema enquanto outro não. Claramente, a solução é fazer com que todos os computadores tenham a mesma configuração, certo? Mas restrições do mundo real e as diferentes necessidades dos seus usuários em geral tornam isso impossível.

Há uma grande diferença em administrar *múltiplas* configurações e *inúmeras* configurações. O truque é dividir sua configuração em partes gerenciáveis. Você descobrirá que algumas partes da localização se aplicam a todos os hosts gerenciados, outras se aplicam a apenas alguns e ainda outras são específicas a componentes individuais.

Além de fazer instalações a partir do zero, você também precisará distribuir atualizações continuamente. Tenha em mente que hosts individuais têm necessidades diferentes para o formato da moeda corrente, estabilidade e tempo de funcionamento.

Um administrador de sistemas prudente jamais deveria distribuir novas versões de software em massa. Em vez disso, as implantações devem ser em estágios de acordo com um plano gradual que acomode as necessidades de outros grupos e dê tempo para a descoberta de problemas enquanto seu potencial de provocar danos ainda for limitado. Jamais atualize servidores críticos até ter certa confiança nas alterações que você está contemplando.[8]

Entretanto, se você projeta seu sistema de localização, certifique-se de que todos os dados originais sejam mantidos em um sistema de controle de revisão. Essa precaução permite monitorar quais alterações foram completamente testadas e estão prontas para implantação. Além disso, ela permite identificar o originador de quaisquer alterações problemáticas. Quanto mais pessoas estão envolvidas nesse processo, mais importante essa última consideração se torna.

É vantajoso separar a distribuição do SO básico, da distribuição de localização. Dependendo das necessidades de estabilidade do seu ambiente, você pode utilizar distribuições locais menores somente para correção de bugs. Mas achamos que a adição de novos recursos em pequenas doses resulta em uma operação mais tranqüila do que enfileirar as alterações em distribuições "cavalares" que impõem o risco de uma ruptura significativa do serviço.

---

8. Correções de segurança são uma possível exceção a essa regra. É importante tapar brechas de segurança tão logo sejam descobertas. As correções de segurança, por outro lado, acabam algumas vezes introduzindo bugs.

Costuma ser uma boa idéia especificar um número máximo de "distribuições" que você quer disponibilizar em um dado momento qualquer. Alguns administradores acreditam que não há razões para corrigir um software que não está quebrado. Eles afirmam que atualizar sistemas sem nenhuma razão específica custa tempo e dinheiro, e que quase tudo aquilo que se propaga como "essencial" freqüentemente acaba se revelando como "essencialmente inútil". As pessoas que colocam esses princípios em prática devem estar dispostas a coletar um catálogo extenso das distribuições ativas.

Em oposição, o pessoal mais ortodoxo aponta para a complexidade inerente das distribuições e para a dificuldade de compreender (para não mencionar o gerenciar) uma coleção aleatória de distribuições que remonta há vários anos no passado. A exceção para esse pessoal são as correções de segurança que, em geral, devem ser aplicadas universalmente e em curto prazo. Corrigir versões ultrapassadas do sistema operacional é freqüentemente impraticável, portanto os administradores se confrontam com a escolha entre pular atualizações em alguns computadores ou atualizar, correndo o risco de eles travarem com a nova versão interna. Nada bom.

Nenhuma dessas perspectivas é comprovadamente correta, mas tendemos a estar do lado daqueles que preferem um número limitado de versões. É melhor realizar suas atualizações de acordo com sua própria agenda em vez de daquela ditada por uma emergência externa.

Testes

É importante testar alterações antes de liberá-las para o mundo. No mínimo, isso significa que você precisa testar suas próprias alterações na configuração local. Mas você também deve testar o software que seu fornecedor distribui. Um fornecedor UNIX importante uma vez distribuiu uma correção que, quando aplicada de certa maneira, realizava um **rm -rf /**. Imagine instalar essa correção por toda sua empresa sem antes testá-la.

Os testes são uma questão especialmente pertinente para correções fornecidas pelos desenvolvedores se você utiliza um serviço como **apt-get** ou o Red Hat Network que oferece uma capacidade de correção automática. Sistemas essenciais a uma missão nunca devem ser conectados diretamente a um serviço de atualização mantido pelo fornecedor. Identifique uma máquina sacrificial a ser conectada ao serviço e distribua as alterações a partir dessa para outras máquinas na sua instalação somente depois de um teste apropriado. Desative as atualizações durante a fase de testes; caso contrário, enquanto você testa, as alterações podem inverter seu caminho e entrar de uma maneira prematura e sorrateira nos seus sistemas de produção.

Consulte página 640 para informações adicionais sobre o rastreamento de problemas.

Caso anteveja a ocorrência de problemas ou alterações visíveis pelo usuário em associação com uma atualização planejada, notifique os usuários antecipadamente e lhes dê uma chance de se comunicar com você se eles tiverem preocupações com as alterações ou sincronizações que você pretende fazer. Certifique-se de que os usuários tenham à disposição uma maneira fácil de informar bugs.

Se sua organização for geograficamente distribuída, certifique-se de que outras filiais ajudem no processo de testes. A participação internacional é particularmente valiosa em ambientes multilingüísticos. Se, por exemplo, ninguém na unidade norte-americana fala japonês, é melhor você fazer a unidade de Tóquio testar qualquer coisa que possa vir a afetar o suporte ao kanji. Um número surpreendente de parâmetros de sistema varia com a localização. A unidade nos Estados Unidos testará as alterações na infra-estrutura de impressão com papel A4, ou as unidades fora dos Estados Unidos simplesmente terão uma surpresa?

Compilação local

Nos velhos tempos do UNIX, quando havia muitas arquiteturas diferentes, geralmente os programas eram distribuídos na forma de repositórios de arquivos-fonte, normalmente arquivos **.tar.Z** que você descompactaria e então compilaria. Assim que o programa fosse gerado, seria instalado o software em uma localização como **/usr/local**. Hoje, o emprego de sistemas de gerenciamento de pacotes significa que um número menor de programas precisa ser instalado dessa maneira. Isso também significa que os administradores tomam menos decisões, uma vez que os pacotes especificam onde seus conteúdos devem ser instalados.

Mesmo com o fácil gerenciamento de pacotes, algumas pessoas ainda preferem compilar seus próprios softwares.[9] Executar sua própria construção fornece mais controle sobre as opções de compilação do software. Ela também lhe permite ser mais paranóico, pois você pode inspecionar o código-fonte que você está compilando. Algumas pessoas parecem pensar que essa inspeção é importante, mas a menos que você tenha o tempo e conhecimento para inspecionar cada linha de um pacote de software de 20.000 linhas, suspeitamos que o valor de segurança agregado é provavelmente mínimo.

Como nem todo software no mundo foi empacotado para todas as versões de distribuição Linux, é possível que você encontre pelo menos alguns programas que você mesmo precisa instalar, especialmente se seus computadores não são sis-

9. Aqueles que são irredutíveis e preferem fazer suas próprias compilações devem conferir a distribuição Gentoo Linux, que é projetada para ser recompilada a partir do zero no sistema de destino.

temas Intel de 32 bits. Mais ainda, se a sua instalação tem como objetivo o desenvolvimento de sistemas, você terá de considerar onde colocar o seu próprio software desenvolvido localmente para a instalação.

Historicamente, a localização mais comum para software local tem sido **/usr/local** e essa convenção ainda é amplamente seguida hoje em dia. O padrão FHS (Filesystem Hierarchy Standard, padrão de hierarquia de sistema de arquivos) do UNIX/Linux especifica que **/usr/local** esteja presente e vazio após a instalação inicial do sistema operacional e muitos pacotes esperam ser instalados neste local.

Um número deprimentemente grande de outros pacotes (particularmente aplicativos comerciais) esperam ser instalados em **/usr**, o que geralmente é uma má idéia fora do contexto de um sistema de gerenciamento de pacotes.[10] Nossa experiência é a de que embora muitos aplicativos tentem ser instalados em **/usr**, eles geralmente funcionam sem problemas quando instalados em um outro lugar. No raro caso de um pacote mal-comportado, freqüentemente é possível corrigir as coisas adicionando links simbólicos para **/usr** que apontam para um diretório de instalação em outro lugar. Uma outra possível solução é verificar se o aplicativo mal comportado recorre a uma variável de ambiente para localizar seu diretório de instalação; normalmente esse recurso está documentado, mas nem sempre.

Embora **/usr/local** seja tradicional, muitas instalações o classificam como inapropriado. A maneira tradicional como ele é disposto (basicamente a mesma que **/usr**, com binários em **/usr/local/bin**, páginas de manual em **/usr/local/man** e assim por diante) cria uma série de problemas em alguns ambientes: é difícil ter várias versões instaladas do mesmo software, os diretórios podem ficar grandes, é problemático gerenciar várias arquiteturas etc.

Distribuindo localizações

O sistema de localização da sua empresa deve tratar tanto a instalação inicial como as atualizações incrementais. As atualizações podem ser particularmente difíceis. A eficiência pode ser uma preocupação fundamental desde que você não queira repetir toda a novela da localização para atualizar as permissões de um único arquivo. Mesmo que o processo seja automatizado, o de reconstrução a partir do zero torna as atualizações um processo caro e demorado.

Uma maneira simples e escalonável de organizar as localizações é manter os arquivos em uma estrutura de árvore que simula o sistema de arquivos de uma máquina de produção. Um script dedicado de instalação pode copiar a árvore para a máquina de destino e realizar todas as edições adicionais necessárias.

Esse tipo de configuração tem várias vantagens. Você pode manter um número qualquer de árvores de localização necessárias para implementar seu esquema administrativo local. Algumas árvores serão alternativas, com cada máquina obtendo somente uma das escolhas disponíveis. Outras árvores serão sobreposições que podem ser copiadas para a parte superior das árvores que vieram antes delas. As árvores de localização podem sobrescrever arquivos, se necessário, ou podem estar completamente desassociadas. Cada árvore potencialmente instalada de uma maneira independente deve ser representada por um projeto separado de controle de revisão.

A abordagem de sobreposição de árvores permite flexibilidade na implementação. Se utilizar um sistema de empacotamento para distribuir suas personalizações locais, as sobreposições podem ser simplesmente colocadas em pacotes independentes. Os scripts apropriados de personalização podem ser incluídos no pacote e configurados para executarem como parte do processo de instalação.

Uma outra boa idéia de implementação é utilizar **rsync** para fazer com que as máquinas de destino sigam as regras das suas árvores de sobreposição. **rsync** copia somente os arquivos obsoletos, portanto ele pode ser bem eficiente para distribuição de alterações incrementais. Esse comportamento é difícil de simular apenas com um sistema de empacotamento. Consulte a página 353 para informações adicionais sobre o **rsync**.

Resolvendo questões de agendamento

Algumas instalações atualizam todos os seus sistemas de acordo com uma agenda regular. Esse plano é potencialmente intrusivo, mas tem a vantagem de definir um limite superior quanto ao obsoletismo dos sistemas cliente. Outras instalações atualizam seu sistema em tempo de inicialização. Essa é uma opção relativamente segura, mas pode significar que muito tempo transcorre entre as atualizações. Algumas instalações com usuários tecnicamente sofisticados permitem que os próprios usuários escolham quando atualizar suas máquinas. Esse é um plano cooperativo e amigável, mas tende a exigir um dos outros esquemas como um adjunto porque alguns usuários nunca atualizarão.

As atualizações podem ser selecionadas do servidor de atualização (normalmente por meio do **cron**) ou elas podem ser adicionadas pelos clientes individuais. Um sistema de extração dá aos clientes melhor controle sobre seus próprios cronogramas de atualização. Por exemplo, uma empresa com operações no mundo todo pode achar mais fácil implementar a atualização todas as noites; uma atualização à meia-noite nos Estados Unidos é uma atualização no meio do expediente na Ásia.

---

10. A questão em manter software aleatório fora de **/usr** e de outros diretórios de sistema é segregar personalizações locais do software de sistema. Sob o gerenciamento de pacotes, os próprios pacotes fornecem uma trilha de auditoria, portanto é mais sensato que os pacotes modifiquem os diretórios de sistema.

Dependendo de quantas máquinas você gerencia e da extensão de uma área geográfica que elas cobrem, você poderia configurar um único servidor de distribuição ou uma hierarquia de servidores. Por exemplo, você poderia fazer com que um servidor mestre distribuísse para um escravo em cada edifício e esse escravo por sua vez poderia distribuir diretamente para os clientes. Para instalações geograficamente dispersas, esse arranjo pode reduzir significativamente o consumo da largura de banda da rede remota.

11.7 FERRAMENTAS DE GERENCIAMENTO DE CONFIGURAÇÃO

Os sistemas de localização tendem a ser feitos em casa. A razão disso é que todas as instalações são diferentes e cada instalação tem suas próprias peculiaridades. Entretanto, a síndrome NIH ("not invented here") também contribui significativamente. Talvez a falta de uma ferramenta de código-fonte aberto dominante para realizar o gerenciamento de configuração nos condicionou a pensar nesse problema como residindo fora do domínio das ferramentas padronizadas.

Contudo, essas ferramentas existem e vale a pena você revisá-las, mesmo que seja apenas para lhe dar alguma clareza sobre por que você optou por não utilizá-las. As seções a seguir delineiam os sistemas mais populares mais ou menos na ordem de popularidade e semelhança.

cfengine: sistema imune a computador

Uma das ferramentas de localização mais bem conhecidas é a **cfengine**, de Mark Burgess. Ela foi considerada como um tipo de "sistema imune a computador" que baseia sua operação em um modelo de como o sistema deve ser configurado. Quando ela detecta uma discrepância entre o modelo e a realidade, **cfengine** segue os passos apropriados para fazer com que o sistema obedeça ao padrão. Por causa desse modelo subjacente, **cfengine** é realmente útil para manutenção de configuração em progresso.

cfengine pode criar cópias de backup dos arquivos que ela modifica e pode manter um log detalhado das alterações. Essa ferramenta também pode ser executada em um modo sem ação real em que ela descreve as alterações que ela faria sem realmente implementá-las.

Você utiliza linguagem especial da própria **cfengine** para descrever a maneira como você quer configurar seus computadores. Você pode especificar regras como: "O arquivo **xyz** deve existir em **/etc**, ter permissões 644 e pertencer a root". Você também pode escrever regras com relação ao conteúdo dos arquivos individuais. Por exemplo, você pode especificar que **/etc/hosts** deve conter a linha "192.168.0.1 de roteador". **cfengine** então adiciona essa linha se ela estiver ausente.

A linguagem de configuração da **cfengine** permite ativar regras individuais dependendo de fatores como o hostname, o SO ou a sub-rede. Esse recurso facilita escrever um único arquivo de configuração que abrange as necessidades de todas as máquinas no seu domínio administrativo.

O seguinte é um exemplo simples do mundo UNIX. Ele certifica-se de que **/bin** é um link simbólico para **/usr/bin** em Suns, realiza alguma verificação adicional de link nos componentes OSF de legado e remove tudo de **/var/scratch** mais antigo que sete dias:

```
control:
actionsequence = ( links tidy )
links:
 sun4::
 /bin -> /usr/bin
 # other links
 osf::
 # some osf specific links
tidy:
 /var/scratch pattern=* age=7 recurse=inf
```

Acesse a home page do **cfengine** em www.iu.hio.no/cfengine para informações adicionais.

LCFG: um sistema de configuração em larga escala

O LCFG foi originalmente desenvolvido em 1993, por Paul Anderson, na Universidade de Edinburgh. Na sua encarnação mais recente, conhecida como LCFG(ng), ganhou alguns usuários fora das universidades. O LCFG é projetado principalmente para o gerenciamento de grandes instalações Red Hat, mas versões para o Mac OS X e o Solaris estão a caminho. O site Web do LCFG é www.lcfg.org.

Como ocorre com **cfengine**, o LCFG define uma linguagem de configuração especializada. As configurações de todas as máquinas gerenciadas são armazenadas em um servidor central em um conjunto de arquivos de configuração mestre. A partir desses arquivos, o LCFG gera arquivos XML personalizados que descrevem a configuração de cada host gerenciado.

Um daemon no servidor central monitora os arquivos de configuração mestre para alterações e regenera os arquivos XML conforme necessário.

Os arquivos XML são publicados em um servidor Web interno a partir do qual os clientes podem então selecionar suas próprias configurações. Os clientes utilizam uma variedade de scripts de componente para autoconfigurarem de acordo com os XML.

O Arusha Project (ARK)

O Arusha Project foi fundado por Will Parain da Universidade de Glasgow. A motivação por trás do Arusha é o insight de que a maioria dos administradores de sistemas faz essencialmente as mesmas coisas, mas nem sempre. Ou como Will coloca: "Quero que meu serviço seja o mesmo que o deles, mas com essas pequenas diferenças...". O site Web Arusha Project em ark.sf.net entra em mais detalhes no que diz respeito à estrutura básica subjacente filosófica desse sistema.

O Arusha Project está baseado em uma linguagem orientada a objetos chamada ARK. Ela é expressa em XML e permite que o administrador defina objetos como hosts, licenças, sites Web, lista de discussão etc. Você pode descrever as dependências entre esses objetos com ARK e pode fornecer métodos e dados que esclarecem seus inter-relacionamentos. Como no desenvolvimento de software orientado a objetos, é possível reutilizar e dividir objetos em subclasses, sobrescrevendo apenas as partes que devem ser alteradas para que elas funcionem no seu ambiente local.

Template Tree 2: o auxiliary do cfengine

O Template Tree 2 foi criado no Swiss Federal Institute of Technology por Tobias Oetiker. Ele é um sistema baseado em componentes e guiado por uma configuração central. Ele reduz a complexidade seguindo uma abordagem de dois níveis para definir a configuração de uma instalação e pode lidar com os diretórios-raiz realocados das máquinas sem disco.

No nível mais baixo, esse sistema consiste em alguns "pacotes de recursos". Um pacote de recursos é uma coleção de arquivos acompanhados por um arquivo **META** que descreve a maneira como esses arquivos devem ser instalados no sistema alvo. Um recurso pode ser qualquer coisa, desde a configuração de uma rede até a última versão do OpenSSH. Os recursos podem expor parâmetros configuráveis que podem ser definidos no arquivo de configuração mestre.

O nível superior da configuração é um arquivo de configuração mestre de instalação em que você seleciona os recursos conjuntamente e associa-os a máquinas ou a grupos de máquinas. Nesse nível, você deve especificar os valores para os parâmetros de configuração não-vinculados expostos por cada recurso. Por exemplo, um dos parâmetros para um recurso de servidor de correio poderia ser o nome do domínio de correio.

O Template Tree 2 combina as informações do arquivo de configuração mestre com os recursos individuais dos arquivos **META** para gerar um arquivo de configuração **cfengine** para a instalação inteira. Como cada recurso deve conter a documentação sobre seu propósito e uso, o Template Tree 2 também pode gerar documentação composta.

DMTF/CIM: o Common Information Model

O Distributed Management Task Force (DMTF), uma coalizão com "mais de 3.000 participantes ativos", tem trabalhado desde 1992 para desenvolver seu Common Information Model (CIM) como uma tentativa de criar padrões para um sistema de gerenciamento para diversas plataformas orientado a objetos.

Nas palavras do próprio DMTF, CIM é "um esquema de gerenciamento (...) fornecido para estabelecer uma estrutura conceitual comum no nível de uma topologia fundamental tanto em relação com a classificação e a associação, como em relação com o conjunto básico das classes destinado a estabelecer uma estrutura comum para uma descrição do ambiente gerenciado". Ou seja o que for.

Todos os principais fornecedores, da Microsoft à Sun, são membros do DMTF. Infelizmente, os padrões que eles produziram demonstram um domínio impressionante das artes do esoterismo e do cultivo dos termos na moda. As empresas envolvidas parecem ávidas em demonstrar sua boa vontade de padronizar não importa o quê. Os padrões se concentram em XML e na orientação a objeto. Entretanto, ainda estamos por ver um produto razoável construído sobre eles.

Como o esforço de padronização está sendo principalmente patrocinado pelos nossos velhos amigos fornecedores de código-fonte fechado, os resultados são mais ou menos aquilo que poderíamos esperar. A Sun, por exemplo, distribuiu um sistema para gerenciamento de clientes sem disco com base nessas tecnologias com o Solaris 8. Um exame mais minucioso revelou um servidor baseado em Java que foi desencadeado por um cliente baseado em Java para produzir um script de shell que então fez o trabalho real. Quando ocorram problemas, o manual instruía os usuários a revisar o arquivo de log, que era uma estrutura binária só acessível por meio de um aplicativo visualizador de log baseado em Java.

Se há uma vantagem nesse lamaçal, é que os esforços do DMTF pelo menos exigem que os fornecedores disponibilizem programaticamente interfaces de configuração acessíveis aos seus sistemas com base em um padrão aberto. Para ambientes UNIX e Linux, isso não tem nada de novo, mas o DMTF não é uma criatura UNIX. Ele inclui a Cisco, Microsoft, Symantec e muitas outras empresas com um pouco histórico de fornecer maneiras sensatas de programar seus sistemas. Dar a esses produtos uma API de configuração é uma coisa boa, mesmo se as implementações ainda estiverem faltando.

11.8 COMPARTILHANDO SOFTWARE PELO NFS

Onde o software extra deveria ser realmente instalado: em clientes individuais ou em um servidor de arquivos central a partir do qual ele poderia ser compartilhado via NFS? A resposta Linux padrão é "nos clientes", mas a solução NFS torna as atualizações mais rápidas (é mais rápido e mais confiável atualizar dez servidores NFS do que 1.000 clientes) e economiza espaço em disco nos clientes (não que isso importe muito no mundo dos discos de 400 GB).

Na verdade, a pergunta se reduz à gerenciabilidade *versus* confiabilidade. Acesso baseado em sistemas de arquivos pela rede é centralizado e mais fácil de gerenciar no dia-a-dia; além de permitir disponibilizar instantaneamente a correção de bugs e novos pacotes em todos os clientes. Mas executar pela rede pode ser um pouco mais lento do que acessar um disco local. Além disso, o modelo de servidor de rede adiciona dependências à rede e ao servidor de arquivos, não apenas porque ele adiciona potenciais pontos de falha, mas também porque exige que clientes e servidores concordem sobre coisas como bibliotecas compartilhadas que estarão disponíveis e a versão dessas bibliotecas que será instalada. A questão é que bibliotecas de software NFS são uma técnica administrativa avançada e só devem ser tentadas em ambientes que permitem um alto grau de coordenação central.

Se trabalhar em um ambiente com redes eficientes e puder arcar com um array RAID rápido para o servidor central, talvez você descubra que o servidor de rede realmente tem um desempenho mais rápido que um disco IDE local. Qualquer que sejam as condições, o impacto sobre o desempenho possivelmente será pequeno; considerações sobre o desempenho não devem dominar essa decisão arquitetônica particular.

Em geral, redes de sistemas heterogêneos tiram mais proveito dos repositórios de software compartilhados. Se sua instalação padronizou um sistema operacional e se esse sistema operacional fornecer recursos razoáveis de gerenciamento de pacotes, você estará mais bem posicionado atendo-se ao sistema nativo.

Espaços de nomes de pacotes

O UNIX tradicional pulveriza o conteúdo de novos pacotes por múltiplos diretórios. As bibliotecas são colocadas em **/usr/lib**, binários em **/usr/bin**, documentação em **/usr/share/docs** e assim por diante. O Linux herda mais ou menos o mesmo sistema, embora Filesystem Hierarchy Standard ajude a tornar as localizações um pouco mais previsíveis. (Acesse www.pathname.com/fhs para informações adicionais sobre o FHS.)

A vantagem dessa convenção é que os arquivos são exibidos em lugares bem conhecidos — desde que sua variável de ambiente PATH aponte para **/usr/bin** e para os outros diretórios binários padrão, por exemplo, programas recém-instalados estarão prontamente disponíveis.

As desvantagens são que as origens dos arquivos precisam ser monitoradas explicitamente (por meio dos sistemas de gerenciamento de pacotes) e que os arquivos dispersos são difíceis de compartilhar em uma rede. Felizmente, administradores de sistemas dispostos a realizar algum esforço extra têm uma maneira razoável de resolver esse dilema: espaços de nomes de pacotes.

O principal ponto desse esquema é instalar cada pacote em diretório-raiz próprio e separado. Por exemplo, você poderia instalar **gimp** em **/tools/graphics/gimp**, com o binário sendo localizado em **/tools/graphics/gimp/bin/gimp**. Você então pode recriar um diretório binário agregado para sua coleção de ferramentas posicionando os links simbólicos em um diretório como **/tools/bin**:

```
/tools/bin/gimp -> /tools/graphics/gimp/bin/gimp
```

Os usuários poderiam então adicionar o diretório **/tools/bin** às suas variáveis PATH com a certeza de que selecionarão todas as ferramentas compartilhadas.

Há várias opções para estruturar o diretório **/tools**. Uma abordagem hierárquica (por exemplo, **/tools/graphics**, **/tools/editors** etc.) facilita a navegação e melhora o desempenho. Talvez você queira incluir a versão de software, arquitetura de hardware, sistema operacional ou as iniciais da pessoa responsável nas suas convenções para atribuição de nomes a fim de permitir que a mesma coleção de ferramentas seja disponibilizada para muitos tipos de clientes. Por exemplo, usuários do Solaris poderiam incluir **/tools/sun4/bin** a suas PATHs e os usuários de Fedora poderiam incluir **/tools/fedora/bin**.

Ao instalar uma nova versão de uma ferramenta importante, uma boa idéia é manter as versões mais antigas disponíveis indefinidamente, em particular quando os usuários investiram tempo e esforços significativos nos projetos que utilizam essa ferramenta. Idealmente, novas versões das ferramentas seriam para retrocompatibilidade com arquivos de dados e softwares antigos, mas, na prática, desastres são comuns. É bom exigir que os usuários passem por alguns problemas de configuração ao acessarem uma versão mais antiga de um pacote; não é bom simplesmente quebrar o trabalho existente deles e fazer com que eles lidem com as conseqüências.

Gerenciamento de dependências

Alguns pacotes dependem de bibliotecas ou de outros pacotes de software. Ao instalar um software localmente por meio de um sistema de gerenciamento de pacotes, você obtém uma grande quantidade de ajuda para resolver essas questões. Mas ao construir seu próprio repositório de softwares de rede no nível global da instalação, você deve resolver essas questões explicitamente.

Se gerenciar bibliotecas da mesma maneira como gerencia aplicativos, você pode compilar suas ferramentas para utilizar bibliotecas a partir de dentro do diretório **/tools** compartilhado. Essa convenção permite manter múltiplas versões de uma biblioteca ativa simultaneamente. Como aplicativos dependentes são vinculados contra versões específicas da biblioteca, a configuração permanece estável mesmo quando novas versões da biblioteca são distribuídas. A desvantagem é que esse tipo de configuração pode ser bem complicado para utilizar e manter ao longo do tempo.

Resista à tentação de vincular contra um diretório **/tools/lib** global que contém links genericamente identificados para bibliotecas comuns. Se alterar esses links, você pode se deparar com problemas inesperados e difíceis de ser diagnosticados. Sistemas de bibliotecas compartilhadas são projetados para resolver algumas dores de cabeça potenciais, mas faz sentido ter mais cuidado em uma configuração complexa.

Os passos exatos necessários para fazer o linker utilizar uma versão específica de uma biblioteca compartilhada variam entre sistemas. Sob o Linux, você pode configurar a variável de ambiente LD_LIBRARY_PATH ou utilizar a opção **-R** do linker.

Scripts empacotadores

Infelizmente, a compatibilidade no nível de uma biblioteca é apenas metade da história. O fato de que as ferramentas invocam umas às outras diretamente levanta outra oportunidade para conflitos. Por exemplo, suponha que o utilitário chamado **foo** faça uso freqüente do utilitário chamado **bar**. Se atualizar a versão padrão de **bar**, você poderá descobrir que **foo** repentinamente pára de funcionar. Nesse caso, você pode concluir que **foo** dependia de algum comportamento de **bar** que não mais é suportado (ou pelo menos não é mais o padrão).

Se seu repositório de software suportar múltiplas versões (por exemplo, **/tools/util/bar-1.0** e **/tools/util/bar-2.0**), você pode corrigir esse problema movendo a versão original de **foo** para **foo.real** e substituindo-a por um pequeno script empacotador:

```
#!/bin/sh
# certifica-se de que o programa pode encontrar qualquer arquivo empacotado junto com ele
# primeiro mesmo se ele não utilizar um caminho explícito.
PATH=/tools/util/bar-1.0/bin:$PATH
export PATH
exec /tools/util/foo-1.0/bin/foo.real "$@"
```

Agora **foo** será carregado com uma variável de ambiente PATH personalizada e chamará a versão antiga de **bar** em preferência à nova.

Empacotadores são uma ferramenta poderosa que podem resolver não apenas dependências de pacotes, mas também questões como segurança, arquitetura ou dependência de SO e monitoramento de uso. Algumas instalações empacotam todos os binários compartilhados.

Ferramentas de implementação

A manutenção dos repositórios de software compartilhados é um problema comum na administração de sistemas e os administradores desenvolveram várias soluções de software para ajudar a facilitar esse processo. Como ocorre com a localização, havia uma tendência do predomínio de soluções internas, mas, como sempre, vale a pena conferir os sistemas disponíveis para ver se algum deles atende às suas necessidades.

Para alguns dos clássicos, pesquise no google por CMU Depot, o CERN ASIS e o CITES Encap. Alguns desses projetos continuam vivos e ativos, enquanto outros foram abandonados sob a pressão dos componentes Windows e a disponibilidade onipresente das atualizações dos pacotes RPM. Há também vários projetos relativamente novos:

- O GNU Stow (www.gnu.org/software/stow) mantém links a partir de um diretório binário central para os binários reais presentes em diretórios específicos a pacotes.
- O Swpkg, de Chrisophe Kalt (web.taranis.org/swpkg), é uma coleção de ferramentas que abordam a maioria dos passos na construção de um repositório de software compartilhado.
- O Pack Management Project (pack.sunsite.dk), de Peter Kristensen, começou com a audaciosa ambição de fornecer não apenas um sistema para instalar softwares, mas também o próprio software pré-empacotado (com foco no Solaris). A parte do pré-empacotamento foi desde então abandonada, mas o trabalho continua nas ferramentas de software.
- O sistema SEPP, criado no Swiss Federal Institute of Technology (www.sepp.ee.ethz.ch), implementa a maioria das idéias discutidas anteriormente.

11.9 SOFTWARE RECOMENDADO

Algumas ferramentas são tão importantes que as consideramos obrigatórias para todas as máquinas. Outros pacotes de que gostamos talvez só sejam necessários ou úteis em máquinas selecionadas. Entretanto, o preço do espaço em disco hoje

em dia quase torna mais fácil instalar tudo em todos os locais e manter a consistência da árvore local de software. A Tabela 11.1 mostra nossa lista dos pacotes obrigatórios.

Tabela 11.1 Pacotes de software de terceiros essenciais

Pacote	Descrição e comentários
ssh/scp	Secure shell – utiliza criptografia, não revela senhas
sudo	Substituto do **su** – adiciona controle e registro em log
sendmail ou **postfix**	As opções para programas de transporte de correio
traceroute	Rastreamento de rota de rede – necessário para depurar problemas de rede
tcpdump ou Wireshark	As opções para *sniffers* de rede – utilizado para analisar o tráfego de rede
gzip	Compactador de dados GNU – necessário para desempacotar downloads
mozilla ou **firefox**	As opções para navegadores Web – você precisa ter
RCS/CVS/Subversion	Sistemas de controle de revisão – úteis tanto para os administradores de sistema como para os usuários
Perl	Linguagem de criação de scripts – para solução de problemas de uso geral

A Tabela 11.2 mostra nossas escolhas de programas interessantes, mas não essenciais; dividimos a tabela em uma seção para administradores de sistemas e uma seção geral para usuários.

Todos os programas listados nas tabelas 11.1 e 11.2 são gratuitos. A maioria deles está disponível na Web (tente freshmeat.net), mas alguns ainda só são distribuídos por meio de FTP. Se eles não estão incluídos na sua distribuição, utilize um sistema de pesquisa para localizá-los ou verifique se um ponteiro é fornecido em outra parte neste livro (consulte o índice para ver se um comando é discutido em uma outra parte).

Tabela 11.2 Pacotes de software úteis de terceiros

	Pacote	Descrição e comentários
Ferramentas para administradores	**gcc**	Compilador C/C++ – um compilador de alta qualidade do GNU
	BIND	Ferramentas de serviço de nomes – obtenha a versão atual (por razões de segurança)[a]
	npasswd	substituto de **passwd** – força os usuários a escolher boas senhas
	xntpd	Daemon de data/hora – mantém todos os relógios das máquinas corretos e sincronizados
	Samba	Windows SMB – compartilha arquivos/impressoras com sistemas Windows
	Apache	Servidor Web
	Squid	Proxy Web e software de cache
	imapd/**procmail**	Ferramentas e correio – para acessar e filtrar e-mail
	mrtg, RRDtool	Ferramentas de monitoramento – para tráfego de rede e outros dados
Ferramentas para usuários	Acrobat Reader	Exibe arquivos PDF, uma interessante ferramenta (gratuita) da Adobe[b]
	gimp	Editor de imagens de mapa de bits
	xfig	Programa de desenho X Windows simples
	GnuPG	Sinaliza, verifica e criptografa mensagens
	nvi/**vim**	Editores de textos do tipo **vi** – recomendado para administradores de sistemas
	emacs	Editor de textos/sistema operacional – bom para usuários avançados
	pico, nedit	Editores de texto – uma boa opção para iniciantes
	enscript, **mpage**	Utilitários de impressão – pretty-printer e N-up formatter
	pine	Leitor de e-mail apropriado para iniciantes
	thunderbird	Leitor de correio para aqueles que preferem uma interface gráfica
	mh/**exmh**	Leitor de correio para processar grandes quantidades de e-mails (não muito bem mantido)
	glimpse	Ferramenta de indexação – indexa seus arquivos e realiza pesquisas rápidas
	gs, **gv**, **ghostview**	Ferramentas para visualização e impressão de documentos PostScript

a. O pacote BIND inclui **dig**, **host** e **nslookup**.
b. O pacote Evince gratuito é um substituto razoável ao Acrobat Reader sob o GNOME.

11.10 LEITURA RECOMENDADA

INTEL CORPORATION AND SYSTEMSOFT. *Preboot Execution Environment (PXE) Specification, Version 2.1*. 1999. Disponível em: <www.pix.net/software/pxeboot/archive/pxespec.pdf>.

MINTHA, JIM E KRUL, PIETER. *UltraLinux FAQ*. Disponível em: <www.ultralinux.org/faq.html>.

PXELINUX QUESTIONS. Disponível em: <syslinux.zytor.com/pxe.php>.

RODIN, JOSIP. *Debian New Maintainers' Guide*. Disponível em: <www.debian.org/doc/maint-guide>.

Este documento contém boas informações sobre pacotes **.deb**.

SILVA, GUSTAVO NORONHA. *APT HOWTO*. Disponível em: <www.debian.org/doc/manuals/apt-howto>.

HOHNDEL, DIRK E HERSCHEL, FABIAN. *Automated Installation of Linux Systems Using YaST*. Disponível em: <www.usenix.org/events/lisa99/full_papers/hohndel/hohndel_html>.

STÜCKELBERG, MARC VUILLEUMIER E CLERC, DAVID. *Linux Remote-Boot mini-HOWTO:* Configuring Remote-Boot Workstations with Linux, DOS, Windows 95/98 and Windows NT. 1999. Disponível em: tldp.org/HOWTO/Remote-Boot.html

THE RED HAT ENTERPRISE LINUX SYSTEM ADMINISTRATION GUIDE. Disponível em: <www.redhat.com/docs>.

WACHSMANN, Alf. *How to Install Red Hat Linux via PXE and Kickstart*. Disponível em: <www.stanford.edu/~alfw/PXE-Kickstart/PXE-Kickstart.html>.

BURGESS, MARK. *Cfengine:* USENIX Computing Systems, A Site Configuration Engine. vol. 8, n. 3. 1995. Disponível em: <www.cfengine.org>.

OETIKER, TOBIAS. *SEPP:* Software Installation and Sharing System. Boston: LISA. 1998. Disponível em: <people.ee.ethz.ch/oetiker/sepp>.

11.11 EXERCÍCIOS

E11.1 Aponte as diferenças entre o Kickstart e o AutoYaST. Quando você procuraria uma alternativa aos instaladores específicos a distribuições?

E11.2 Instale uma cópia do Subversion a partir de subversion.tigris.org. Configure **svnserve** e crie um repositório. Como você pode tornar o repositório utilizável a partir de algum lugar na rede local e ainda manter uma segurança razoável?

E11.3 Analise a maneira como o software local é organizado em sua instalação. O sistema pode ser ampliado? Ele é fácil de ser utilizado? Discuta.

★ E11.4 Descubra os passos necessários para criar um pacote RPM. Utilize esse procedimento para empacotar um produto de software da sua escolha.

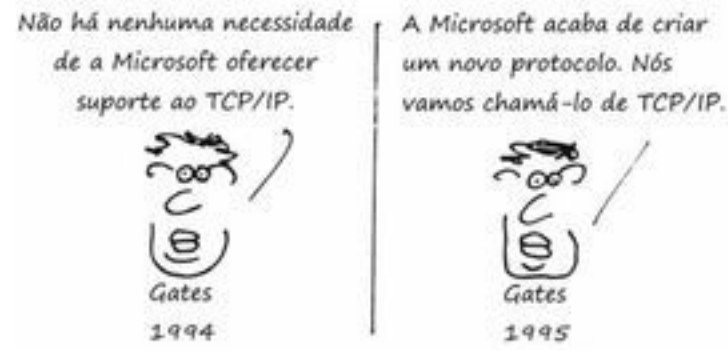
★ E11.5 Repita o exercício anterior, mas crie um pacote **.deb** no formato do Debian.

★ ★ E11.6 Configure o instalador de rede de sua preferência e instale uma máquina nova utilizando o seu servidor. Descreva todas as etapas necessárias para realizar essa tarefa. Quais são alguns dos empecilhos encontrados? Quais são alguns dos problemas de escalabilidade que você descobriu com o instalador escolhido?

PARTE 2
REDES

12 Redes TCP/IP

Seria difícil aumentar ainda mais a importância das redes para a computação moderna, embora isso pareça não impedir as pessoas de ficarem tentando. Em muitas instalações, os acessos à Web e ao e-mail agora são as principais atividades nas quais os computadores são usados. Em 2006, a estimativa é de que a Internet tenha mais de um bilhão de usuários, um aumento de 55% em relação à de 2001. A manutenção de redes locais, conexões Internet, sites Web e softwares relacionados a redes é a rotina da maioria dos administradores de sistemas.



O TCP/IP é o conjunto de protocolos de rede mais comumente usado no Linux/UNIX, Mac OS, Windows e muitos outros sistemas operacionais. Ele também é a linguagem nativa da Internet. TCP quer dizer Transmission Control Protocol, ou seja, protocolo de controle de transmissão, e IP significa Internet Protocol, ou seja, protocolo Internet.

Dispositivos que "falam" o protocolo TCP/IP são capazes de trocar dados (interoperar) apesar das várias diferenças existentes entre eles. IP, o protocolo de remessa subjacente desse conjunto, é o burro de carga da Internet. TCP e UDP (User Datagram Protocol) são protocolos de transporte construídos sobre o IP para entrega de pacotes para aplicativos específicos.

O TCP é um protocolo orientado a conexões que facilita a conversação entre dois programas. Ele funciona de forma muito parecida com uma ligação telefônica: as palavras que você diz chegam à pessoa para quem você ligou e vice-versa. A conexão persiste mesmo quando nenhuma das partes está falando. O TCP fornece entrega confiável, controle de fluxo e controle de congestionamento.

O UDP é um serviço orientado a pacotes. Ele é análogo a enviar uma carta pelo e-mail. Ele é um protocolo sem conexão, não tem nenhuma forma de controle de congestionamento e não garante que os pacotes serão entregues na mesma ordem em que foram enviados (nem mesmo que serão entregues).

O TCP é um protocolo educado que força os usuários que competem pelos mesmos recursos a compartilhar largura de banda e geralmente a se comportar de modo adequado à produtividade da rede como um todo. Por outro lado, o UDP despeja pacotes o mais rápido que ele puder.

À medida que a Internet se torna cada vez mais popular e mais congestionada, precisamos que a maior parte do tráfego seja TCP para evitar congestionamento e compartilhar efetivamente a largura de banda disponível. Hoje, o TCP é responsável pela maior parte do tráfego da Internet, com o UDP e o ICMP em um distante segundo e terceiro lugar, respectivamente. Aplicativos UDP como jogos, música, voz e vídeo estão marcando suas presenças, mas são completamente sobrepujados pela Web e programas como BitTorrent que são verdadeiros comilões de largura de banda, mas que utilizam TCP em vez de UDP para transporte.

Este capítulo introduz os protocolos TCP/IP nos contextos técnico e político da Internet. Infelizmente, mesmo os conceitos básicos de rede são muito extensos para serem vistos em um único capítulo. Outros capítulos relacionados à rede neste livro incluem o Capítulo 13, *Roteamento*, o Capítulo 15, *DNS: o Domain Name System*, o Capítulo 19, *Gerenciamento e depuração de redes*, e o Capítulo 20, *Segurança*.

As próximas seções abrangem material básico sobre os protocolos e a política da Internet e são bem contumazes e introdutórias. Pule para a página 198 se quiser ir diretamente às entranhas do IP ou para a página 215 para ir diretamente a informações de configuração específicas das distribuições.

12.1 O TCP/IP E A INTERNET

O TCP/IP e a Internet compartilham uma história que remonta a várias décadas. O sucesso técnico da Internet se deve em grande parte ao design elegante e flexível do TCP/IP e ao fato de que ele é um conjunto de protocolos aberto e não proprietário. Por outro lado, os benefícios fornecidos pela Internet ajudaram o TCP/IP, prevalecer sobre vários suites de protocolo concorrentes que foram favorecidos em um ou outro momento por razões políticas e comerciais.

Um breve histórico

Contrário à crença popular, a Internet não é um produto da Microsoft que debutou em 1995, nem é a criação de um vice-presidente norte-americano. O progenitor da Internet moderna foi uma rede chamada ARPANET estabelecida em

1969 pela DARPA (Defense Advanced Research Project Agency), o braço de P&D (Pesquisa e Desenvolvimento) do Departamento de Defesa dos Estados Unidos. A ARPANET acabou tornando-se o backbone NSFNET que conectava instalações com supercomputadores e redes regionais.

No final dos anos 80, a rede não era mais um projeto de pesquisa e chegara a hora da National Science Foundation pular fora do negócio das redes. A transição para a Internet comercial foi feita ao longo de um período de vários anos; a NSFNET foi desativada em abril de 1994. O backbone da Internet atual é um conjunto de redes privadas pertencentes a ISPs (Internet Service Providers, provedores de serviço Internet) que se interconectam em vários pontos denominados peer.

Em meados dos anos 80 a Internet era formada, basicamente, por instalações ARPANET originais e um punhado de universidades com computadores VAX da Digital rodando UNIX Berkley em redes Ethernet a 10 Mb/s e conectadas por linhas telefônicas digitais de 56 Kb alugadas. A cada mês de setembro, quando os estudantes retornavam às aulas, a Internet sofria o que acabou se tornando conhecido como colapso de congestionamento. Van Jacobson, então pesquisador do Network Research Group do Lawrence Berkeley Labs, analisaria o comportamento dos protocolos sobre condições de carga e os corrigiria. Os algoritmos que hoje conhecemos como *partida lenta*, *prevenção de congestionamento*, *retransmissão rápida* e *recuperação rápida* surgiram desse contexto.

A lei de Moore (a regra prática de que a velocidade dos equipamentos dobram a cada 18 meses) e a pressão do mercado aceleraram em muito o desenvolvimento da Internet. Desde o final dos anos 80, quando os algoritmos TCP atuais foram estabilizados, a velocidade das interfaces de rede foi multiplicada por 1.000 (de 6% de eficiência das primeiras redes Ethernet com velocidade de 10 Mb/s a aproximadamente 100% de eficiência em redes Ethernet com velocidades na casa dos gigabits), a velocidade dos circuitos alugados multiplicou-se por 12.000 e o número total de hosts foi multiplicado por um fator de 80.000.

Qualquer um que tivesse projetado um sistema de software e visse que ele havia se tornado obsoleto em relação à próxima geração de hardware ou à próxima versão de um sistema operacional sabe quão maravilhoso é o fato de nossa Internet ainda estar viva e resistindo, utilizando praticamente o mesmo conjunto de protocolos TCP/IP que foi elaborado há 30 anos para uma Internet muito diferente da atual. Tiramos o chapéu para Bob Kahn, Vint Cerf, Jon Postel e todas as demais pessoas que tornaram isso possível.

Como a Internet é gerenciada atualmente

O desenvolvimento da Internet sempre foi um empreendimento aberto e cooperativo. Agora que ela é uma força comercial mobilizadora na economia mundial, vários setores estão preocupados e têm a impressão de que a Internet está nas mãos de um punhado de geeks e com talvez uma pequena orientação por parte do governo norte-americano. Agrade ou não, um controle sobre a Internet está se aproximando.

Várias organizações estão envolvidas:

- ICANN (Internet Corporation for Assigned Names and Numbers):

  se alguém pode ser visto como o administrador da Internet, esse grupo é o tal (www.icann.org).
- ISOC, the Internet Society: a ISOC é uma organização membro que representa a comunidade de usuários Internet (www.isoc.org).
- IETF, o Internet Engineering Task Force: esse grupo supervisiona o desenvolvimento e a padronização dos aspectos técnicos da Internet. Ele é o fórum aberto no qual qualquer um pode participar (www.ietf.org).

Desses grupos, ICANN tem o trabalho mais árduo: estabelecer-se como a autoridade a cargo da Internet, desfazendo os erros do passado e antevendo o futuro.

Além dessas organizações, um grupo internacional formado por pesquisadores acadêmicos, órgãos governamentais e líderes da indústria criou um consórcio de rede chamado Internet2. Essas organizações se uniram para contribuir com idéias e desenvolver tecnologias que são fundamentais ao fomento do progresso da Internet. Diferentemente do que a imprensa afirma, a Internet2 não é uma rede distinta da Internet. Embora ela utilize um backbone de rede privada chamado Abilene para pesquisa de rede, ela não substituirá a Internet como nós a conhecemos hoje em dia. Aprenda mais sobre a Internet2 em www.internet2.edu.

Documentação e padrões de rede

As atividades técnicas da comunidade Internet são sintetizadas em documentos conhecidos como RFCs; uma RFC (Request for Comments) é uma solicitação de comentários. Os padrões de protocolos, as alterações propostas e os boletins informativos acabam tornando-se RFCs. As RFCs começam suas vidas como Internet Drafts e, após vários e-mails terem circulado de lá para cá e reuniões do IETF, elas acabam tendo um de dois destinos: ou morrem ou são promovidas à categoria de RFC. Qualquer um que tenha algum comentário a respeito de um draft ou RFC proposto é incentivado a participar. Além de padronizar os protocolos Internet, o mecanismo RFC algumas vezes simplesmente documenta ou explica aspectos da prática existente.

As RFCs são numeradas seqüencialmente; hoje em dia, há por volta de 4.000. As RFCs também têm títulos descritivos (por exemplo, *Algorithms for Synchronizing Network Clocks* — algoritmos para sincronizar clocks de rede), mas, para evitar ambigüidades, elas normalmente são citadas pelo número. Uma vez distribuídas, o conteúdo de uma RFC jamais muda. As atualizações são distribuídas como novas RFCs com seus próprios números de referência. Por convenção, RFCs atualizadas contêm todo o material que permanece relevante, de forma que RFCs novas substituem completamente as antigas, pelo menos teoricamente.

O próprio processo pelo qual as RFCs são publicadas está documentado na RFC intitulada *Internet Official Protocol Standards*. Essa RFC também inclui ponteiros para as RFCs mais atualizadas para vários padrões de protocolos. Como as informações mudam com freqüência, essa RFC é emitida novamente a cada 100 RFCs. O processo de padrões Internet em si é detalhado na RFC2026. Outra meta-RFC útil é a RFC2555, *30 Years of RFCs*, que descreve parte do contexto cultural e técnico por trás do sistema de RFCs.

Não se assuste com a opulência de detalhes técnicos encontrados nas RFCs. A maioria contém introduções, sumários e princípios úteis para os administradores de sistemas. Algumas são escritas especificamente como visões gerais ou introduções genéricas. As RFCs podem não ser a maneira mais suave de aprender um tópico, mas são autorizadas, concisas e gratuitas.

Nem todas as RFCs são áridas e cheias de detalhes técnicos maçantes. Algumas de nossas favoritas do tipo mais suave (normalmente escritas em 1.º de abril) são as RFCs 1118, 1149, 1925, 2324 e 2795:

- RFC1118 — *The Hitchhiker's Guide to the Internet*
- RFC1149 — *A Standard for the Transmission of IP Datagrams on Avian Carriers*[1]
- RFC1925 — *The Twelve Networking Truths*
- RFC2324 — *Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)*
- RFC2795 — *The Infinite Monkey Protocol Suite (IMPS)*

Elas se constituem em uma excelente leitura e nos dão alguma idéia de quem são as pessoas desenvolvendo e construindo nossa Internet.

Além de receber seu próprio número de série, uma RFC também pode receber um número FYI (For Your Information), um número BCP (Best Current Practice) e um número STD (Standard). FYIs, STDs e BCPs são subséries das RFCs que incluem documentos de interesse ou importância especiais.

As FYIs são documentos introdutórios ou informativos destinados ao público em geral. Elas normalmente são um excelente local para iniciar pesquisa sobre um tópico não familiar. As STDs documentam protocolos Internet que completaram os processos de revisão e de teste do IETF e foram adotados formalmente como padrões. As BCPs documentam procedimentos recomendados para sites Internet; elas são formadas por sugestões em termos de administração, e para os administradores de sistemas normalmente formam a série mais valiosa das subséries RFC.

RFCs, FYIs, STDs e BCPs são numeradas seqüencialmente dentro de suas próprias séries de modo que um documento pode originar vários números de identificação diferentes. Por exemplo, a RFC1635, *How to Use Anonymous FTP*, também é conhecida como FYI0024.

As RFCs são encontradas em diversas fontes. Há uma lista de espelhos RFC mantidos ativamente em www.rfc-editor.org, que é a central de despacho para questões relacionadas com RFCs.

12.2 GUIA DE REDES

Agora que já fornecemos a você um pouco de contexto, vamos examinar os protocolos TCP/IP em si. TCP/IP é uma "família de protocolos", um grupo de protocolos de rede desenvolvido para funcionamento em conjunto de forma estável. Ele inclui vários componentes, cada um dos quais definido por uma RFC única ou por uma série delas:

- IP (Internet Protocol), que direciona pacotes de dados de uma máquina para uma outra (RFC791)
- ICMP (Internet Control Message Protocol), que fornece vários tipos de suporte de baixo nível para o IP, incluindo mensagens de erros, assistência de roteamento e ajuda para depuração (RFC792)
- ARP (Address Resolution Protocol), que traduz endereços IP em endereços de hardware (RFC823)[2]
- UDP (User Datagram Protocol) e o TCP (Transmission Control Protocol), que entregam dados para aplicativos específicos na máquina de destino. O UDP dispõe de transporte não verificado e dentro das possibilidades para mensagens

1. Um grupo de entusiastas Linux do BLUG (Bergen [Noruega] Linux Group) até implementou o CPIP (Carrier Pigeon Internet Protocol, protocolo Internet pombo-e-mail), conforme especificado na RFC1149. Para maiores detalhes, veja o site www.blug.linux.no/rfc1149!
2. Essa é, na verdade, uma mentirinha. O ARP não faz parte realmente do TCP/IP e pode ser usado com outros conjuntos de protocolos. Entretanto, ele é parte integral da maneira pela qual o TCP/IP funciona na maioria das mídias para LANs.

individuais, ao passo que o TCP garante uma conversação confiável, full duplex, com controle de fluxo e com correção de erros entre os processos nos dois hosts. (RFCs 768 e 793)

O TCP/IP foi projetado em torno do esquema de divisão em camadas mostrado na Tabela 12.1.

Tabela 12.1 Modelo de rede TCP/IP

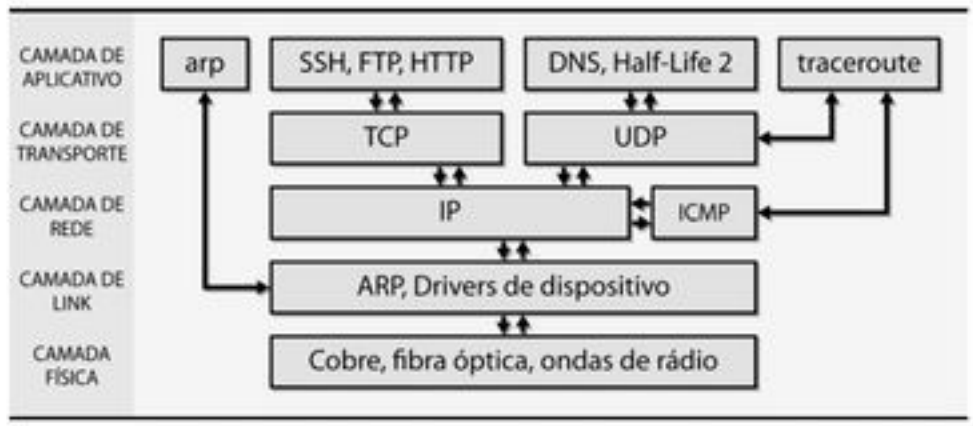
Camada	Função
Camada de aplicativo	Programas aplicativos para o usuário final
Camada de transporte	Entrega de dados aos aplicativos[a]
Camada de rede	Comunicação, endereçamento e roteamento básicos
Camada de link	Hardware de rede e drives de dispositivos
Camada física	O cabo ou a mídia física em si

a. Opcionalmente questões relacionadas à confiabilidade de endereçamento e controle de fluxo.

Depois de o TCP/IP ter sido implementado e empregado, a ISO (International Standards Organization) produziu seu próprio conjunto de protocolos de sete camadas chamado OSI. Ele era um elefante branco projetado por comitê consumado e nunca realmente popularizou-se devido à sua complexidade e ineficiência. Alguns imaginam que uma camada financeira e uma camada política poderiam ser adicionadas ao modelo original OSI de sete camadas.[3]

A Ilustração A ilustra como os vários componentes e clientes do TCP/IP se encaixam nessa arquitetura geral e esquema de camadas.

Ilustração A Uma grande e feliz família TCP/IP



12.3 PACOTES E ENCAPSULAMENTO

O Linux pode suportar uma série de redes físicas, entre as quais Ethernet, FDDI, Token Ring, ATM (Asynchronous Transfer Mode), Ethernet sem fio e sistemas baseados em linha serial. O hardware é gerenciado dentro da camada de link da arquitetura TCP/IP, e protocolos de nível mais alto não sabem o hardware específico que está sendo usado, ou não dão importância a isso.

Os dados percorrem uma rede sob a forma de *pacotes*, seqüências de dados com um comprimento máximo imposto pela camada de link. Cada pacote é formado por um cabeçalho e um conteúdo (*payload*). O cabeçalho informa de onde o pacote veio e para onde vai. Ele também pode incluir somas de verificação, informações específicas a um protocolo ou outras instruções de manipulação. O conteúdo é formado pelos dados a serem transferidos.

O nome da unidade de dados primitiva depende da camada de protocolos. Na camada de link ela é chamada de *frame*, na camada IP de *pacote* e na camada TCP de *segmento*. Aqui, usamos pacote como um termo genérico que engloba todos esses casos.

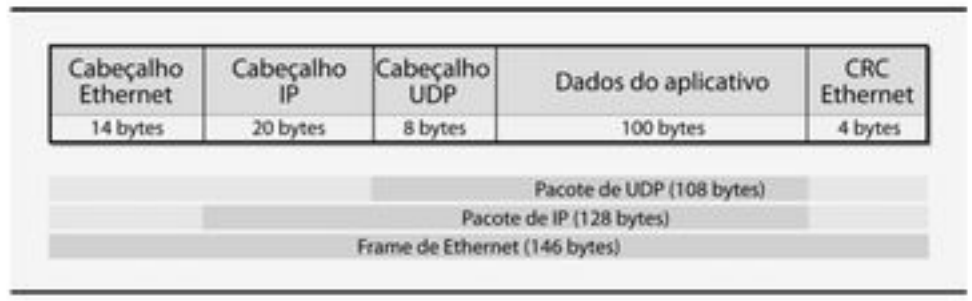
À medida que o pacote desce pela pilha de protocolos (do transporte TCP ou UDP para IP, daí para Ethernet e então o cabo físico) em preparação para ser enviado, cada protocolo adiciona sua própria informação de cabeçalho. Cada pacote finalizado do protocolo se torna parte do conteúdo do pacote gerado pelo próximo protocolo. Esse aninhamento é conheci-

3. Na realidade, uma camiseta que mostra esse modelo estendido de nove camadas está disponível no site Internet Systems Consortium, www.isc.org.

do como encapsulamento. Na máquina receptora, o encapsulamento é invertido à medida que o pacote percorre de volta para cima pela pilha de protocolos.

Por exemplo, um pacote UDP sendo transmitido por rede Ethernet contém três empacotadores ou envelopes diferentes. Na rede Ethernet, ele recebe um "frame" com um cabeçalho simples que lista a fonte e os endereços de hardware de destino do próximo salto, o comprimento do frame e a soma de verificação do frame (CRC). O conteúdo do frame Ethernet é um pacote IP, o conteúdo do pacote IP é um pacote UDP e o conteúdo do pacote UDP é formado pelos próprios dados que estão sendo transmitidos. A Ilustração B mostra os componentes desse frame.

Ilustração B Um pacote de rede típico



Utilizamos o termo "byte" para nos referirmos a uma unidade de dados de 8 bits. No passado, "byte" era um termo mais genérico, de modo que freqüentemente você verá o termo "octeto" utilizado em seu lugar nas RFCs.

A camada de link

Nesta seção, abordamos vários tópicos que ligam o intervalo entre as camadas mais baixas do software de rede e o hardware de rede em si.

Padrões de framing Ethernet

Uma das principais tarefas da camada de link é adicionar cabeçalhos a pacotes e colocar separadores entre eles. Os cabeçalhos contêm as informações de endereçamento da camada de link e as somas de verificação dos pacotes, e os separadores garantem que os receptores consigam informar onde um pacote termina e onde o próximo começa. O processo de adicionar esses trechos de informação é conhecido genericamente como *framing*.

Hoje, há um único padrão para framing Ethernet de 10 MB/s em uso comum: DIX Ethernet II.[4] Historicamente, um outro padrão conhecido como IEEE 802.2 LLC/SNAP era comum para redes baseadas em Novell. Alguns administradores podem encontrar remanescentes dessa versão do framing em suas redes.

O framing que uma máquina utiliza é determinado tanto pela sua placa de rede quanto pelo driver da placa de rede. Em PCs executados em Windows você pode escolher o estilo de framing que você quer, mas no Linux normalmente isso não é possível. Ambos os tipos de framing operam perfeitamente entre si sob a perspectiva do Linux. Por outro lado, as máquinas Windows que utilizam framing distinto na mesma rede não conseguem se comunicar entre si. Como administrador de sistemas, você não precisa se preocupar com discordâncias de framing a menos que esteja efetuando uma depuração de baixo nível em uma rede mista.

Cabeamento Ethernet e padrões de sinalização

As opções de cabeamento para as diversas velocidades Ethernet (10 MB/s, 100 MB/s, 1 GB/s e agora 10 GB/s) normalmente são especificadas como parte da iniciativa de padronização do IEEE. Normalmente, um único tipo de cabo com limites de distância curta será aprovado à medida que uma nova tecnologia emergir. Tipos de mídia mais baratos e limites mais generosos serão acrescentados posteriormente.

Consulte o Capítulo 14, *Hardware de rede*, para informações adicionais sobre os diversos tipos de padrão. Outra referência útil aos prós e contras da Ethernet é o site www.ethermanage.com/ethernet/ mantido por Charles Spurgeon.

Redes sem fio

O padrão IEEE 802.11 tenta definir os padrões de framing e de sinalização para redes sem fio. Infelizmente, ele era originalmente um tanto vago e incluía vários parâmetros e opções que não foram especificados por completo. Um problema de interoperabilidade que talvez você precise prestar atenção é aquele da "tradução" *versus* "encapsulamento".

---

4. A camada de link é, na verdade, dividida em duas partes: MAC, a subcamada Media Access Control, e a LLC, a subcamada Link Layer Control. A camada MAC lida com a mídia e coloca os pacotes nos cabos. A camada LLC trata do framing.

A tradução converte um pacote de um formato em outro; o encapsulamento envolve o pacote com o formato desejado. Os sistemas Windows tendem a utilizar como padrão o encapsulamento e os sistemas Linux, a tradução; as estações-base sem fio devem ser configuradas explicitamente. Se você estiver implantando uma rede sem fio, deverá ter certeza de que suas estações base e as estações de trabalho com as quais elas conversam estejam todas operando no mesmo modo.

Unidade de transferência máxima

O tamanho dos pacotes em uma rede pode ser limitado tanto pelas especificações de hardware quanto pelas convenções dos protocolos. Por exemplo, o conteúdo de um frame Ethernet padrão não pode ultrapassar os 1.500 bytes. O limite de tamanho está associado ao protocolo da camada de link e é chamado de MTU (Maximum Transfer Unit). A Tabela 12.2 ilustra valores típicos para o MTU.

Tabela 12.2 MTUs para diversos tipos de camada de link de rede

Tipo de rede	Unidade de transferência máxima
Ethernet	1.500 bytes (1.492 com framing 802.2)
FDDI	4.470 bytes (4.352 para IP/FDDI)
Token ring	Configurável[a]
Link de modem PPP	Configurável, normalmente 512 ou 576 bytes
Pilhas de PC	Configurável, normalmente assume o padrão de 512
Links WAN ponto a ponto (T1, T3)	Configurável, normalmente 1.500 ou 4.500 bytes

a. Valores comuns são 552; 1.064; 2.088; 4.508 e 8.232. Às vezes 1.500 para corresponder com a Ethernet.

No conjunto TCP/IP, a camada IP divide pacotes para atender ao MTU de um determinado link de rede. Se um pacote é roteado por diversas redes, uma das redes intermediárias pode ter um MTU menor do que a rede de origem. Nesse caso, o roteador que encaminha o pacote para a rede de menor MTU subdividirá ainda mais o pacote em um processo denominado fragmentação. A fragmentação é uma tarefa malvista por um roteador atarefado.

O protocolo TCP é capaz de determinar o menor MTU ao longo do caminho para o destino e utilizar esse tamanho desde o princípio. O UDP não é tão bom e fica satisfeito em passar esse trabalho extra para a camada IP. No IPv6, um novo protocolo traça seu caminho ao longo do processo de padronização, os roteadores intermediários não podem mais executar a fragmentação; é necessário descobrir o MTU.

Problemas de fragmentação podem ser insidiosos. Embora a descoberta MTU de caminho deva resolver conflitos no MTU, às vezes um administrador precisa intervir. Se estiver, por exemplo, utilizando uma arquitetura com túneis para uma rede privada virtual, você terá de examinar o tamanho dos pacotes que estão atravessando o túnel. Normalmente eles têm 1.500 bytes só para começar, mas assim que o cabeçalho de tunelamento for adicionado, eles passarão a ter por volta de 1.540 bytes e deverão ser fragmentados. Configurar o MTU do link com um valor menor evitará a fragmentação e aumentará o desempenho global do tunelamento da rede. Veja a página man de **ifconfig** para ver como configurar o MTU de uma interface.

Endereçamento de pacotes

Assim como as cartas ou mensagens de e-mail, os pacotes de rede têm de ser endereçados adequadamente para chegarem a seus destinos. São utilizados vários esquemas de endereçamento combinados:

- Endereços MAC (*medium access control*) para hardware
- Endereços IP para software
- Nomes de host para as pessoas

A interface de rede de um host em geral tem um endereço MAC na camada de link que o distingue de outras máquinas na mesma rede física, um endereço IP que a identifica na Internet global e um nome de host que é usado pelos seres humanos.

O nível mais baixo de endereçamento é ditado pelo hardware da rede. Por exemplo, dispositivos Ethernet recebem um endereço de hardware exclusivo de 6 bytes no momento da fabricação.[5] Interfaces Token Ring possuem um endereço similar que também tem 6 bytes de comprimento. Algumas redes ponto a ponto (como PPP, descritas na página 223) não precisam de nenhum endereço de hardware; a identidade do destino é especificada quando o link é estabelecido.

5. Exclusivos pelo menos na teoria. Certa vez a 3Com duplicou números Ethernet entre as suas placas com diferentes tipos de conectores de rede; eles supuseram que os clientes fariam pedidos apenas de um único tipo. Esse atalho gerou grande tumulto e problemas em instalações que estavam numa fase de transição entre mídias e chegou até mesmo a provocar problemas na própria rede interna da 3Com. Conflitos de endereços MAC são fatais na mesma rede, porém são aceitáveis em redes separadas por um roteador.

Um endereço Ethernet de 6 bytes é dividido em duas partes: os três primeiros bytes identificam o fabricante do hardware e os três últimos formam um número de série exclusivo atribuído pelo fabricante. Os administradores de sistemas normalmente conseguem identificar pelo menos a marca do dispositivo que está congestionando a rede por meio de observação do identificador de 3 bytes em uma tabela de IDs de fornecedores. Há uma tabela dessas atualizada no endereço a seguir:

www.iana.org/assignments/ethernet-numbers

Essas informações costumavam ser publicadas regularmente nas séries RFC, mas não são mais distribuídas desta maneira. A RFC1700 (1994) foi a última RFC sobre *Assigned Numbers*. O repositório oficial de todos os números mágicos da Internet agora é www.iana.org/numbers.htm.

Endereços de hardware de interfaces Ethernet devem ser atribuídos permanentemente e de maneira imutável; infelizmente, algumas placas de rede permitem que especifiquemos o endereço de hardware. Placas sem fio são especialmente problemáticas nesse aspecto. Não atribua valores no intervalo de endereços multibroadcast (segundo dígito ímpar) nem use outros valores especiais.

O Linux permite que alteremos o endereço de hardware da interface Ethernet, mas não faça isso; você poderá desmontar firewalls e algumas implementações DHCP. Entretanto, esse recurso pode ser útil se você tiver de substituir uma máquina ou placa de rede danificada e, por alguma razão, precisar utilizar o endereço MAC antigo (por exemplo, todos os seus switches o filtram ou seu servidor DHCP distribui endereços baseados em endereços MAC).

No próximo nível acima do hardware, usa-se o endereçamento Internet (mais comumente conhecido como endereçamento IP). Tipicamente, é atribuído um[6] IP de 4 bytes a cada interface de rede. Os endereços IP são globalmente exclusivos[7] e dependentes de hardware. Continuamos falando sobre endereços IP na próxima seção.

Veja a página 207 para informações adicionais sobre ***ARP****.*

O mapeamento entre endereços IP e endereços de hardware é implementado na camada de link do modelo TCP/IP. Em redes que suportam broadcast (por exemplo, redes que permitem que pacotes sejam endereçados a "todos os hosts nesta rede física"), o protocolo ARP permite que esses mapeamentos sejam descobertos automaticamente, sem a ajuda de um administrador de sistemas.

Como os endereços IP são longos, números aparentemente aleatórios, eles são difíceis de lembrar. Os sistemas Linux permitem que um ou mais nomes de host sejam associados a um endereço IP, de modo que os usuários podem digitar yahoo.com em vez de 216.115.108.245. Esse mapeamento pode ser configurado de várias maneiras, desde um arquivo estático (**/etc/hosts**) até o sistema de banco de dados NIS ou DNS (Domain Name System), o sistema mundial para nomes de domínios. Tenha em mente que os nomes de host são, na realidade, apenas uma abreviatura conveniente dos endereços IP.

Portas

Os endereços IP identificam máquinas, ou mais precisamente, interfaces de rede em uma máquina. Eles não são específicos o bastante para endereçar determinados processos ou serviços. O TCP e o UDP estendem os endereços com um conceito conhecido como "porta". Uma porta é um número de 16 bits que suplementa um endereço IP para especificar um determinado canal de comunicação. Serviços padrão como e-mail, FTP e a Web se associam a portas "bem conhecidas" definidas em **/etc/services**. Para ajudar a evitar a apropriação indevida desses serviços, os sistemas Linux restringem os programas de servidor de se vincularem a números de porta abaixo de 1.024, a menos que eles estejam rodando como root. (Mas qualquer um pode se comunicar com um servidor que está executando em um número de porta baixo; a restrição só se aplica para assumir o controle da porta.)

Tipos de endereço

Tanto a camada IP quanto a camada de link definem vários tipos diferentes de endereços:

- Unicast — endereços que se referem a um único host (interface de rede, na verdade)
- Multicast — endereços que identificam um grupo de hosts
- Broadcast — endereços que incluem todos os hosts da rede local

O endereçamento multicast facilita aplicativos como videoconferência no qual o mesmo conjunto de pacotes tem de ser enviado a todos os participantes. O IGMP (Internet Group Management Protocol, protocolo de gerenciamento de grupos Internet) constrói e gerencia conjuntos de hosts que são tratados como um único destino multicast. Endereços IP multicast iniciam com um byte no intervalo 224 a 239.

---

6. As interfaces de rede podem, na verdade, ter mais do que um endereço IP associado a elas, porém, se trata de uma configuração especial que é usada somente em algumas circunstâncias específicas. Consulte a seção *Configurando interfaces virtuais* na página 501 para obter informações adicionais.
7. Essa é uma pequena mentira que é verdade na maioria dos casos. Leia a discussão sobre o NAT que se inicia na página 203 para obter as informações exatas sobre endereços IP não exclusivos.

Os endereços broadcast alcançam todos os hosts na rede local utilizando uma forma especial de curinga do endereço em que a representação binária da parte do host (definida a seguir) é inteiramente de 1s.

Multicast praticamente não é utilizado na Internet de hoje em dia, mas vários projetos de pesquisa ativos estão em andamento. Consulte Internet2.org para obter informações adicionais.

12.4 ENDEREÇOS IP: OS MÍNIMOS DETALHES

Um endereço IP ou Internet tem quatro bytes de comprimento e é dividido em uma parte rede e uma parte host. A parte rede identifica uma rede lógica à qual o endereço se refere e a parte host identifica uma máquina nesta rede.

Por convenção, os endereços IP são escritos como números decimais, um para cada byte, separados por pontos. Por exemplo, o endereço IP para nossa máquina "boulder" é escrito como 128.138.240.1. O byte mais à esquerda é o mais significativo e sempre faz parte da parte rede.

Quando 127 é o primeiro byte de um endereço, ele representa a "rede loopback", uma rede fictícia que não possui nenhuma interface de hardware real e um único host. O endereço loopback 127.0.0.1 sempre se refere ao host atual. Seu nome simbólico é "localhost".

O endereço IP de uma interface e outros parâmetros são configurados pelo comando **ifconfig**. Pule para a página 209 para obter uma descrição detalhada de **ifconfig**.

Classes de endereço IP

Historicamente, os endereços IP eram agrupados em "classes", dependendo dos primeiros bits do byte mais à esquerda. A classe determinava quais bytes do endereço se encontravam na parte rede e quais na parte host. Hoje, os sistemas de roteamento utilizam uma máscara explícita para especificar a parte rede e são capazes de definir limites entre quaisquer dois bits, não somente nos limites dos bytes. Entretanto, as classes tradicionais ainda são utilizadas como padrão quando nenhuma divisão explícita é especificada.

As classes A, B e C representam endereços IP regulares. As classes D e E são utilizadas para endereços de multicast e de pesquisa. A Tabela 12.3 descreve as características de cada uma das classes. A parte rede de um endereço é representada por N e a parte host, por H.

Tabela 12.3 Classes de endereços Internet históricas

Classe	1º byte[a]	Formato	Comentários
A	1-126	N.H.H.H	As primeiras redes ou reservadas para DoD
B	128-191	N.N.H.H	Grandes instalações, normalmente com sub-redes, eram difíceis de ser conseguidas
C	192-223	N.N.H.H	Fáceis de ser conseguidas, normalmente obtidas em conjuntos
D	224-239	–	Endereços multicast, atribuídos de uma forma não permanente
E	240-255	–	Endereços experimentais

a. O valor 0 é especial e não é utilizado como primeiro byte de endereços IP comuns. 127 é reservado ao endereço de loopback.

Sub-redes e máscaras de rede

É raro uma única rede física ter mais do que uma centena de computadores conectados a ela. Portanto, os endereços classe A e classe B (que permitem, respectivamente, 16.777.214 e 65.534 hosts por rede) são de fato estúpidos e um desperdício. Por exemplo, as 126 redes classe A utilizam até metade do espaço de endereços disponível.

Instalações com esses endereços usam um refinamento do esquema de endereçamento, chamado sub-redes, no qual parte da parte host de um endereço é "emprestada" para estender a parte rede. Por exemplo, os quatro bytes de um endereço classe B normalmente seriam interpretados como N.N.H.H. Se fosse empregado o recurso de sub-rede para atribuir o terceiro byte ao número de rede em vez do número de host, o endereço seria interpretado como N.N.N.H. O uso de sub-redes transforma um único endereço de rede classe B em 256 redes distintas tipo classe C, cada uma das quais capaz de suportar até 254 hosts.

Consulte a página 209 para informações adicionais sobre o ifconfig.

Essa nova atribuição é efetuada pelo emprego do comando **ifconfig** para associar uma "máscara de sub-rede" explícita a uma interface de rede. Cada bit da máscara de rede que corresponde à parte rede de um endereço IP é configurado em 1 e os bits host são configurados em 0. Por exemplo, a máscara de rede para a configuração N.N.N.H seria 255.255.255.0 em decimal ou 0xFFFFFF00 em hexadecimal. **ifconfig** normalmente utiliza a classe inerente de um endereço para descobrir que bits estão na parte rede. Ao configurar uma máscara explícita, simplesmente sobrescreva esse comportamento.

A divisão entre a parte rede e a parte host não precisa cair em um limite de byte. Entretanto, os bits de rede têm de ser contíguos e aparecer no lado da ordem mais alta do endereço.[8]

As máscaras de sub-rede que não terminam em um limite de byte podem ser irritantes para serem decodificadas e normalmente são escritas como /XX, onde XX é o número de bits na parte rede do endereço. Isso algumas vezes é chamado de notação CIDR (Classless Inter-Domain Routing, roteamento entre domínios sem classes). Por exemplo, o endereço de rede 128.138.243.0/26 refere-se à primeira de quatro redes cujos primeiros bytes são 128.138.243. As outras três redes têm 64, 128 e 192 como seus primeiros quatro bytes. A máscara de sub-rede associada a essas redes é 255.255.255.192 ou 0xFFFFFFC0; em binário, são 26 números 1 seguidos por 6 zeros. A Ilustração C mostra os relacionamentos entre esses números com um pouco mais de detalhe.

Ilustração C Conversão de base de máscara de sub-rede

Endereço IP	128 .	138 .	243 .	0
Máscara decimal	255 .	255 .	255 .	192
Submáscara hexadecimal	f f .	f f .	f f .	c 0
Submáscara binária	1111 1111 .	1111 1111 .	1111 1111 .	1100 0000

Uma rede /26 possui 6 bits deixados (32 – 26 = 6) para numerar hosts. 2^6 é 64, de modo que a rede possui 64 endereços de host possíveis. Entretanto, ela é capaz de acomodar efetivamente apenas 62 hosts, pois os endereços de host formados somente com 0s ou somente com 1s são reservados (eles são, respectivamente, os endereços de rede e de broadcast).

É confuso fazer toda essa conversão de bits de cabeça, porém alguns truques poderão torná-la mais simples. O número de hosts por rede e o valor do último byte na máscara de rede sempre totalizam 256:

último byte da máscara de sub-rede = 256 – tamanho da rede

Por exemplo, 256 – 64 = 192, que é o byte final da máscara de sub-rede do exemplo anterior. Um outro fato aritmético é que o último byte de um verdadeiro endereço de rede (ao contrário de uma máscara de sub-rede) tem de ser divisível exatamente pelo número de hosts por rede. Vemos isso em ação no presente exemplo, em que os últimos bytes são 0, 64, 128 e 192 — todos divisíveis exatamente por 64.

Em nosso exemplo, os dois bits extras do endereço de rede obtidos por aplicação de sub-redes podem assumir os valores 00, 01, 10 e 11. A rede 128.138.243.0/24 foi dividida então em quatro redes /26.

- 128.138.243.0/26 (0 em decimal é **00**000000 em binário)
- 128.138.243.64/26 (64 em decimal é **01**000000 em binário)
- 128.138.243.128/26 (128 em decimal é **10**000000 em binário)
- 128.138.243.192/26 (192 em decimal é **10**000000 em binário)

Os bits em negrito do último byte de cada endereço são os bits pertencentes à parte rede deste byte.

Dado um endereço IP (digamos, 128.138.243.100), não podemos dizer sem o emprego da máscara de rede associada quais serão os endereços de rede e de broadcast. A Tabela 12.4 ilustra as possibilidades para /16 (o padrão para um endereço classe B), /24 (um valor sensível) e /26 (um valor realista caso o espaço de endereços seja apertado).

Tabela 12.4 Exemplo de decodificação de endereços IP

Endereço IP	Máscara de rede	Rede	Broadcast
128.138.243.100/16	255.255.0.0	128.138.0.0	128.138.255.255
128.138.243.100/24	255.255.255.0	128.138.243.0	128.138.243.255
128.138.243.100/26	255.255.255.192	128.138.243.64	128.138.243.127

Um script on-line útil chamado IP Calculator de Krischan Jodies (disponível em www.jodies.de/ipcalc) ajuda na aritmética binário/hexadecimal/máscara. O **ipcalc** exibe tudo o que talvez você precise conhecer sobre um endereço de rede e sua máscara de rede, endereço broadcast, hosts etc. Há um tarball disponível além da versão on-line; o Debian e o Ubuntu incluem por padrão a versão **ipcalc** de linha de comando.

8. Configurações como N.N.H.N já foram permitidas no passado, mas eram incomuns; hoje elas não são mais permitidas.

Eis uma amostra da saída gerada pelo IP Calculator, um pouco modificada para ajudar na formatação:

```
Address:   24.8.175.69          00011000.00001000.10101111   .01000101
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111   .00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000   .11111111
=>
Network:   24.8.175.0/24        00011000.00001000.10101111   .00000000 (Class A)
Broadcast: 24.8.175.255         00011000.00001000.10101111   .11111111
HostMin:   24.8.175.1           00011000.00001000.10101111   .00000001
HostMax:   24.8.175.254         00011000.00001000.10101111   .11111110
```

A saída fornece uma versão fácil de ser compreendida dos endereços e uma versão para "cortar e colar". Legal.

O Red Hat inclui um programa, também chamado **ipcalc**, que é muito incompleto e, para a maioria dos cálculos, supõe que os endereços IP estejam nas classes A, B ou C.

A RFC original sobre a aplicação de sub-redes em endereços IP (RFC950) não permite o emprego da primeira ou da última sub-redes (tudo 0 e tudo 1). Em nosso exemplo com as redes /26, essa regra eliminaria metade das sub-redes: a sub-rede 0 e a sub-rede 192. Todo o mundo ignorou a RFC, exceto a Novell e a Cisco. (Em versões antigas do sistema operacional IOS da Cisco, tínhamos de ativar explicitamente a sub-rede 0 com o comando `ip subnet zero`. Nas versões 12.0 e posteriores, a sub-rede 0 é disponível como padrão.)

A RFC está enganada, embora tenha boas intenções. A sub-rede 0 não foi permitida, pois se pensava que poderia surgir confusão caso o endereço de sub-rede fosse indistinguível de um endereço de rede em que não tivesse sido aplicado à sub-rede. Entretanto, esse temor provou-se sem sentido e as sub-redes tudo 0/tudo 1 são usadas comumente hoje em dia. É a parte host que não deveria ser tudo 0 ou tudo 1.

O endereço de rede e o endereço de broadcast roubam dois hosts de cada rede, de modo que pareceria que a menor rede significativa teria quatro hosts possíveis: dois hosts reais — usualmente em uma das extremidades de um **link** ponto a ponto — e os endereços de rede e de broadcast. Ter quatro valores para hosts requer dois bits na parte host, de modo que uma rede destas seria uma rede /30 com máscara de rede 255.255.255.252 ou 0xFFFFFFFC. Entretanto, uma rede /31 é tratada como um caso especial (veja a RFC3021) e não possui nem endereço de rede nem endereço de broadcast; seus dois endereços são utilizados para hosts e sua máscara de rede é 255.255.255.254.

Embora os hosts em uma rede possam concordar que estão utilizando endereços nos quais se aplicou sub-rede, o resto do mundo não sabe a esse respeito e continua a tratar endereços de acordo com suas classes implícitas.[9] Em vez de anunciar cada sub-rede para o mundo externo, em nosso exemplo 128.138.243.100 você precisaria anunciar somente uma única rede classe B. Assim que um pacote chega dentro da área de sub-rede, seu endereço de destino é reinterpretado com máscaras de rede locais, a rede real alvo "descoberta", e o pacote roteado para seu destino exato.

A crise dos endereços IP

A comunidade da Internet, mais ou menos em 1992, percebeu que o esquema original de alocação de endereços levantava três problemas fundamentais.

- Primeiramente, iríamos ficar sem endereço classe B — os mais desejáveis para empresas moderadamente grandes — em meados de 1995.
- Em segundo lugar, as tabelas de roteamento de sites backbone Internet estavam crescendo a tal ponto de não caberem mais na memória dos roteadores disponíveis na época.
- Por fim, os endereços IP eram alocados de um modo em que o primeiro que chegasse era o primeiro a ser atendido sem localidade de referência; isto é, endereços numericamente adjacentes poderiam estar na mesma organização ou em continentes diferentes. Imagine a confusão resultante se os códigos postais fossem atribuídos dessa maneira ao acaso.

Para resolver o problema, foram apresentadas duas soluções associadas: uma para o futuro próximo e outra a longo prazo.

A solução de curto prazo, Classless Inter-Domain Routing (CIDR), é uma maneira diferente de gerenciar o existente espaço de endereçamento de 4 bytes, a saber, CIDR utiliza os endereços disponíveis mais eficientemente e simplifica as tabelas de roteamento levando em consideração as adjacências numéricas. Discutimos o CIDR em mais detalhes na próxima seção.

A solução de longo prazo, IPv6, é uma revisão do protocolo IP. O IPv6 expande o espaço de endereçamento para 16 bytes e incorpora várias outras lições aprendidas do uso do IP nos últimos 25 anos. Ele elimina vários recursos do IP que a experiência demonstrou ser de pouca valia, tornando o protocolo potencialmente mais rápido e mais fácil de ser implementado. Ele também integra recursos de segurança e de autenticação no protocolo básico e elimina a fragmentação em roteadores intermediários.

---

9. Outra mentira no nome de uma descrição simples, embora ainda incompleta; veja a discussão sobre CIDR (Classless Inter-Domain Routing) na página 201 para se inteirar sobre o verdadeiro escopo.

O IPv6 ainda se encontra nos estágios iniciais de desenvolvimento, porém o CIDR está totalmente operacional há anos. O CIDR é suportado e utilizado pelo backbone Internet e pela maioria dos fabricantes de equipamentos para roteamento. O NAT (Network Address Translation), um esquema para reutilizar endereços IP que é visto na página 289, também desempenhou um grande papel na redução da demanda por endereços IP.

A complexidade do IPv6, a eficiência do CIDR e do NAT e a inércia de uma Internet que já funciona muito bem, tudo isso combinado sugere que levará tempo antes de passarmos para o IPv6. Embora muitos aplicativos e sistemas operacionais já apresentem suporte nativo ao IPv6 e, como muitas pessoas nos Estados Unidos pensam, o custo por si só é proibitivo. Contudo, extensos esforços por parte dos distribuidores e dos desenvolvedores voluntários asseguraram que, quando a rede mudar para o IPv6, o Linux estará pronto.

A escassez de endereços IPv4 é sentida mais na arena internacional, e foi aí que o IPv6 recebeu as boas-vindas mais calorosas. Nos Estados Unidos, só um aplicativo arrasador poderia impulsionar o IPv6, por exemplo, uma nova geração de celulares que mapeia um endereço IPv6 para um número de telefone. (Sistemas de voz sobre IP também se beneficiariam de uma correspondência mais próxima entre números de telefone e endereços IPv6.)

Alguns detalhes adicionais sobre endereçamento IPv6 são dados na página 204.

CIDR: Classless Inter-Domain Routing

O CIDR (Classless Inter-Domain Routing), definido na RFC1519 (setembro de 1993), elimina o sistema de classes que outrora determinava a parte rede de um endereço IP. Da mesma forma que sub-redes, das quais ele é uma extensão direta, ele depende de uma máscara de rede explícita para definir o limite entre as partes rede e host de um endereço. Mas diferentemente da sub-rede, ele permite, para fins de roteamento, que a parte rede seja *menor* do que seria implícito pela classe implícita de um endereço. Utilizar uma máscara de rede mais curta tem o efeito de agregar diversas redes. Portanto, o CIDR é algumas vezes chamado de super-rede.

Com o CIDR, várias redes classe C podem ser alocadas a uma instalação sem exigir que a Internet tenha entradas de tabelas de roteamento distintas para cada uma delas. A instalação também poderia ter alocado um subespaço de um endereço classe A ou B. Suponha, por exemplo, que tenha sido concedido a uma instalação um bloco de 8 endereços classe C numerados de 192.144.0.0 a 192.144.7.0 (na notação CIDR, 192.144.0.0/21). Internamente, a instalação poderia utilizá-los como:

- 1 rede de comprimento /21, 2.046 host,[10] máscara de rede 255.255.248.0
- 8 redes de comprimento /24, 254 hosts cada, máscara de rede 255.255.255.0
- 16 redes de comprimento /25, 126 hosts cada, máscara de rede 255.255.255.128
- 32 redes de comprimento /26, 62 hosts cada, máscara de rede 255.255.255.192

e assim por diante. Também é possível misturarmos e combinarmos regiões de comprimentos de sub-rede diferentes, desde que todas as partes se juntem sem sobreposição. Isto é chamado de sub-rede de comprimento variável. Por exemplo, um ISP com a alocação 192.144.0.0/21 poderia definir cerca de /30 redes para clientes PPP discados, cerca de /24 redes para clientes grandes e cerca de /27 redes para clientes menores.

Quando você mistura e combina assim, todos os hosts de uma determinada rede terão de ser configurados com a mesma máscara de rede. Não é possível informar a um host da rede que ele é um /24 e a outro host nessa mesma rede que ele é um /25.

A beleza e o valor do CIDR estão no fato de que, sob a perspectiva da Internet, não é necessário ter 256, 128 ou até mesmo 32 entradas de tabelas de roteamento para esses endereços. Todos eles se referem à mesma organização e todos os pacotes precisam ir para o mesmo lugar. Uma única entrada de roteamento para o endereço 192.144.0.0/21 já basta. Além disso, o CIDR facilita a alocação de partes de endereço classe A e B e, conseqüentemente, aumenta em muitas vezes o número de endereços disponíveis.

Com o advento do CIDR, os administradores de sistemas se tornaram bons em aritmética binária e hexadecimal ou então descobriram que o utilitário **bc** do Linux é capaz de realizar cálculos matemáticos em qualquer base, utilizando as diretivas **ibase** e **obase**.[11] Você pode utilizar a Tabela 12.5 como uma referência.

Quando o CIDR foi introduzido em 1993, as tabelas backbone continham aproximadamente 20.000 rotas. O crescimento lento, mas constante da Internet desde a implosão do "dot com" aumentou o tamanho da tabela de roteamento para algo em torno de 250.000 em 2006.[12]

Uma região desagregada do espaço de endereços, chamada de pântano 192 (e pântanos menores nos intervalos 199 e 205), é formada por endereços classe C antigos cujos proprietários não os podem agregar e não querem entregá-los e depois

---

10. O padrão Ethernet original em cabos coaxiais RG-11 permitia no máximo 1.024 hosts em uma única rede; com os switches atuais é possível (mas não muito sensato) realmente construirmos redes imensas.
11. Mas tenha cuidado para não se perder… Esse quebra-cabeça é deixado como um exercício para o leitor.
12. Consulte bgp.potaroo.net para obter informações atuais sobre a tabela de roteamento BGP.

ter de renumerá-los. Os Estados Unidos são particularmente ruins nesse aspecto. A Europa e a Ásia, que começaram um pouco mais tarde, aprenderam com os erros dos norte-americanos e fizeram um trabalho de alocação de endereços muito melhor. Instalações com uma rede 192 desagregada deveriam devolvê-la para o ARIN (American Registry for Internet Numbers, registro americano para números Internet) e receber um novo bloco de seus ISPs. Infelizmente, o custo de renumeração (no espaço IPv4, pelo menos) impede que a maioria das instalações faça isso.

Tabela 12.5 Configurações para vários comprimentos de máscara de rede

Comprimento[a]	Bits de host	Hosts/rede[b]	Máscara de rede decimal	Máscara de sub-rede hexadecimal
/20	12	4094	255.255.240.0	0xFFFFF000
/21	11	2046	255.255.248.0	0xFFFFF800
/22	10	1022	255.255.252.0	0xFFFFFC00
/23	9	510	255.255.254.0	0xFFFFFE00
/24	8	254	255.255.255.0	0xFFFFFF00
/25	7	126	255.255.255.128	0xFFFFFF80
/26	6	62	255.255.255.192	0xFFFFFFC0
/27	5	30	255.255.255.224	0xFFFFFFE0
/28	4	14	255.255.255.240	0xFFFFFFF0
/29	3	6	255.255.255.248	0xFFFFFFF8
/30	2	2	255.255.255.252	0xFFFFFFFC

a. O comprimento de rede + o número de bits de host é sempre 32, uma vez que estamos dividindo a "torta" de tamanho fixo em um endereço IP de 32 bits.

b. Os matemáticos perceberão que o número de hosts por rede é igual a 2^#bitshost – 2; o –2 reflete o fato de os endereços de host tudo 0 e tudo 1 serem especiais.

Embora o CIDR tenha sido imaginado apenas como uma solução interina, ele demonstrou ser sólido o bastante para lidar com os problemas de crescimento da Internet num futuro visível. De fato, a combinação do CIDR e do NAT tem funcionado tão bem que não é claro se realmente precisamos de um novo protocolo IP. Uma enorme quantidade de trabalho de engenharia foi consumida pela especificação IPv6 e em implementações e ambiente de produção do IPv6. Seria uma pena perder todo esse trabalho, mas o emprego no atacado do IPv6 provavelmente irá exigir uma nova aplicação "matadora" escrita somente para o IPv6, um brinquedo feito para "executivos" que use o endereçamento IPv6 ou então uma decisão por parte da Microsoft em tornar obsoleto o IPv4.

Alocação de endereços

Nos primórdios da Internet, sites individuais requisitavam espaço de endereço à InterNIC (Network Information Center). O ARIN e o LACNIC agora substituem o InterNIC nas Américas. Somente ISPs que alocam quantidades significativas de espaço de endereçamento por ano são elegíveis para se cadastrarem no ARIN para espaço de endereço IP. Todos os demais sites têm de fazer solicitação a seus ISPs.

Apenas números de rede são atribuídos formalmente; os sites têm que definir seus próprios números de host para formarem endereços IP completos. Você pode subdividir o espaço de endereços dado a você em sub-redes caso prefira.

Administrativamente, o ICANN (Internet Corporation for Assigned Names and Numbers) delegou os blocos de endereços a cinco registros Internet regionais e essas autoridades regionais são responsáveis por distribuir sub-blocos a ISPs dentro de suas regiões (consulte a Tabela 12.6). Esses ISPs por sua vez dividem seus blocos e liberam trechos a clientes individuais. Apenas grandes ISPs devem lidar diretamente com um dos registros de endereços patrocinados pelo ICANN.

Tabela 12.6 Registros regionais da Internet

Nome	Endereço Web	Região abrangida
ARIN	www.arin.net	América do Norte, parte do Caribe
APNIC	www.apnic.net	Região da Ásia/Pacífico
AfriNIC	www.afrinic.net	África
LACNIC	www.lacnic.net	América Central e do Sul, parte do Caribe
RIPE	www.ripe.net	Europa e adjacências

A delegação do ICANN para registros regionais e para ISPs nacionais ou regionais permitiu maior agregação das tabelas de roteamento ao backbone. Clientes de ISP que receberam espaço de endereços dentro do bloco do ISP não precisam de entradas de roteamento individuais no backbone. Uma única entrada para o bloco agregado que aponta para o ISP já basta.

Originalmente, o espaço de endereços não era alocado de maneira muito razoável. O governo norte-americano reservou cerca de metade do espaço de endereços para ele e deu blocos relativamente pequenos para a Europa e Ásia. Mas a Europa e a Ásia gerenciaram seus espaços de endereço de modo muito mais inteligente do que os Estados Unidos. O mapa de espaço de endereços em

www.caida.org/analysis/learn/ipv4space

ilustra esse fato efetivamente. Mostra o espaço de endereço IP como um todo, as partes que foram alocadas, as partes que são roteadas (e, portanto, acessíveis) e os endereços em que o tráfego foi observado em alguns pontos de troca importantes nos Estados Unidos.

Endereços privados e o NAT

Outra solução temporária para o esgotamento do espaço de endereços é o emprego de espaços de endereços IP privados, descrito na RFC1918 (fevereiro de 1996). Na era do CIDR, os sites normalmente obtinham seus endereços IP de seus ISPs (Internet Service Providers). Se um site quisesse mudar de ISP, isso poderia ser equivalente ao pagamento de um resgate, dado o alto custo para renumeração de suas redes. O ISP cedia o espaço de endereço enquanto ele permanecesse cliente. Se o site então quiser optar por um outro ISP, ele tem de convencer o antigo ISP a permitir que ele continue com os endereços e também convencer o novo ISP a fazer com que o roteamento funcione corretamente para a nova localização com os antigos endereços. Tipicamente, os ISPs não querem esquentar a cabeça com esses tipos de problema e exigirão que o cliente renumere.

Uma alternativa ao emprego de endereços atribuídos por ISPs é utilizar endereços privados que nunca são mostrados a seu ISP. A RFC1918 deixa de lado uma rede classe A, 16 redes classe B e 256 redes classe C que nunca serão alocadas globalmente e poderão ser usadas internamente por qualquer site. O ponto negativo é que nunca se deve permitir que pacotes pertencentes a esses endereços escapem para a Internet. Você deve filtrá-los em seu roteador de fronteira somente por garantia. Se alguns pacotes escaparem, você deve rastrear as configurações incorretas que possibilitaram que eles escapassem.

A Tabela 12.7 mostra números de rede reservados para endereçamento privado. (A coluna "Intervalo CIDR" mostra o intervalo para cada classe na notação CIDR mais compacta; ela não acrescenta nenhuma informação adicional.)

Tabela 12.7 Endereços IP reservados para uso privado

Classe IP	De	Para	Intervalo CIDR
Classe A	10.0.0.0	10.255.255.255	10.0.0.0/8
Classe B	172.16.0.0	172.31.255.255	172.16.0.0/12
Classe C	192.168.0.0	192.168.255.255	192.168.0.0/16

Os sites poderão optar dentre esse conjunto pelo tamanho de rede que melhor atenda às necessidades de suas organizações.

Para permitir que hosts que utilizam endereços privados se comuniquem com a Internet, o roteador de fronteira do site executa um sistema denominado NAT (Network Address Translation). O NAT intercepta pacotes endereçados com esses endereços de uso somente interno e reescrevem seus endereços de origem, utilizando um endereço IP externo real e talvez um número de porta de origem diferente. Ele também mantém uma tabela dos mapeamentos que ele fez entre pares de endereço/porta de origem interno e externo para que a tradução possa ser executada ao contrário quando os pacotes de resposta chegarem da Internet.

O emprego de mapeamento de número de porta por parte do NAT possibilita que várias conversas possam ser multiplexadas no mesmo endereço IP para que um único endereço externo possa ser compartilhado por vários hosts internos. Em alguns casos, um site pode dar conta com apenas um endereço IP "real".

Um site que utiliza o NAT ainda assim tem de solicitar espaço de endereços de seu ISP, mas a maioria dos endereços então obtidos é usada para mapeamentos NAT e não é atribuída a hosts individuais. Se, mais para frente, o site quiser escolher um outro ISP, somente o roteador de borda e sua configuração NAT precisarão ser alterados e não as configurações dos hosts individuais.

O NAT é amplamente utilizado pelos consumidores e também pelas corporações. Também é possível ter um computador com Linux executando a função NAT, embora muitos sites prefiram delegar essa tarefa a seus roteadores ou dispositivos de conexão de rede. Consulte as seções específicas por fornecedor mais adiante neste capítulo para obter maiores detalhes. Por alguma razão, o mundo Linux costumava chamar o NAT de "mascaramento de IP". Entretanto, com a introdução do kernel 2.4, o pessoal do Linux começou a chamá-lo de NAT também.

Uma configuração NAT incorreta pode permitir que pacotes do espaço de endereços privados escapem para a Internet. Os pacotes chegarão a seus destinos, mas pacotes de resposta não serão capazes de retornar. CAIDA[13], uma organização que mede tudo a respeito de redes backbone, acredita que cerca de 0,1% a 0,2% dos pacotes no backbone possuam endereços privados ou verificações de soma incorretas. Isso parece ser uma porcentagem muito pequena, e realmente é, porém ela representa cerca de 20.000 pacotes a cada 10 minutos em um circuito movimentadíssimo em MAE-West (uma das maiores centrais telefônicas públicas na qual vários ISPs se encontram para intercâmbio de tráfego). Consulte o site www.caida.org para obter outros dados estatísticos interessantes e ferramentas de medição de redes.

Uma desvantagem do NAT (ou quem sabe uma vantagem) é que um host arbitrário na Internet não pode se conectar diretamente às máquinas internas de sua instalação. Algumas implementações (por exemplo, Linux e Cisco PIX) permitem configurar "túneis" que suportam conexões diretas para determinados hosts.

Um outro problema é que alguns aplicativos embutiam endereços IP na parte dados dos pacotes; esses aplicativos são despistados ou confundidos pelo NAT. Entre os exemplos estão alguns protocolos de roteamento, programas de streaming, como o RealVideo e SHOUTcast, alguns comandos FTP, como PORT e PASV, o aplicativo de mensagens instantâneas ICQ e muitos jogos. O NAT algumas vezes também interfere com VPNs (Virtual Private Networks).

Grandes corporações que adotam o NAT e endereços RFC1918 têm de instituir alguma maneira de coordenação centralizada de forma que todos os hosts, independentemente de seus grupos administrativos ou departamentais, tenham endereços IP únicos. A situação pode se tornar mais complicada quando uma empresa que utiliza espaço de endereços RFC1918 adquire ou se funde com uma outra que esteja fazendo a mesma coisa. Partes da organização combinada normalmente têm de ser renumeradas.

O NAT oculta a estrutura interna. Esse sigilo parece ser um ganho em termos de segurança, mas o pessoal especializado em segurança de sistemas diz que o NAT realmente não ajuda em nada em termos de segurança e certamente não substitui a necessidade de um firewall. O NAT também enfatiza as tentativas de medir o tamanho ou a topologia da Internet.

Pelo menos uma ferramenta foi desenvolvida como uma tentativa de identificar hosts que utilizam o NAT. O algoritmo do utilitário correlaciona valores TTL de sistema operacional bem conhecidos a valores TTL esperados na rede. Quando eles não se correspondem, o endereço IP de origem é identificado como um dispositivo NAT. Teoricamente, a idéia é eficaz, mas, na realidade, sua implementação é cara e incômoda. Aprenda mais sobre esse método em www.sflow.org/detectNAT.

Endereçamento IPv6

Um endereço IPv6 tem 128 bits de comprimento. Esses endereços longos destinavam-se originalmente a solucionar o problema da exaustão de endereços IP. Agora que estão aí, eles estão sendo explorados para ajudar em questões de roteamento, mobilidade e localidade de referência.

Os endereços IP nunca foram agrupados geograficamente da maneira como é feita com os números de telefone ou CEPs. Agora, com a segmentação proposta do espaço de endereços IPv6, eles pelo menos estarão agrupados com ISPs. A fronteira entre a parte rede e a parte host de um endereço IPv6 é fixada em /64; o limite entre a topologia pública e a topologia local de um site é fixado em /48. A Tabela 12.8 ilustra as diversas partes de um endereço IPv6.

Tabela 12.8 As partes de um endereço IPv6

Bits	Acrônimo	Tradução
1-3	FP	Prefixo do formato; o tipo de endereço, por exemplo, unicast
4-16	ID TLA	ID de agregação de alto nível, como o ISP backbone
17-24	RES	Reservado para uso futuro
25-48	ID NLA	ID de agregação do próximo nível, por exemplo, ISPs regionais e ID do site
49-64	ID SLA	ID de agregação em nível de site, como a sub-rede local
65-128	ID INTERFACE	Identificador de interface (endereço MAC mais preenchimento)

13. CAIDA é a Cooperative Association for Internet Data Analysis, sediada no San Diego Supercomputer Center no campus da UCSD (www.caida.org).

Dessas partes, somente o ID SLA e o ID INTERFACE pertencem ao host e seu site. As demais partes são fornecidas pelo ISP no sentido oposto do fluxo. O SLA especifica uma sub-rede local. O ID de interface de 64 bits identifica a interface de rede do host. Ele contém, tipicamente, o endereço MAC de 48 bits com os dígitos hexadecimais 0xFFFE no meio. Um bit especial no endereço MAC (bit 6 do primeiro byte, numerando os bits da esquerda para a direita, iniciando em 0) chamado de bit universal/local tem de ser complementado (consulte a RFC2373). Esse esquema permite que os hosts sejam numerados automaticamente, o que é um recurso interessante para o administrador de sistemas, já que somente a sub-rede precisa ser gerenciada.

No IPv6, o endereço MAC é visto na camada de IP, uma situação com implicações boas e ruins. A marca e o modelo da placa de rede são codificados na primeira metade do endereço MAC, de modo que hackers de posse do código para uma determinada arquitetura serão beneficiados. A visibilidade dessas informações também preocupou alguns advogados especializados na área. O pessoal do IPv6 respondeu apontando que não era realmente exigido o emprego de endereços MAC por parte dos sites; eles eram livres para utilizarem o que quisessem como endereço host. Um esquema para incluir um identificador aleatório na parte local do endereço também foi proposto. Muitos bits para se lidar!

Por outro lado, atribuir endereço IPv6 deve ser mais fácil do que atribuir endereço IPv4, uma vez que você precisa apenas controlar o endereço de sub-rede. Os hosts são capazes de se autoconfigurarem — ou pelo menos, na teoria.

O prefixo de formato identifica o tipo de endereço IPv6: unicast, multicast ou qualquer tipo de cast. Endereços unicast configuram FP em 001 (binário). Os IDs TLA e NLA identificam sua companhia telefônica de backbone IP de nível mais alto e os ISPs locais até a cadeia para o seu provedor de backbone.

Hoje em dia, a maioria dos fornecedores está pronta para o IPv6. Quase todas as distribuições vêm com suporte IPv6 nativo e o hardware de rede tem suporte IPv6 há quase uma década.

O ARIN geralmente aloca espaço IPv6 somente para grandes ISPs ou para registros Internet locais que planejam distribuir grandes partes de espaço de endereçamento no futuro próximo. Essas empresas podem então alocar subespaços para seus clientes. As taxas são estruturadas com base em um pagamento mínimo de US$ 1.250/ano e no máximo US$ 36.000/ano. ARIN renunciou as taxas dos membros leais.

Eis algumas fontes úteis sobre IPv6:

- www.ipv6tf.net — um portal de informações sobre o IPv6
- www.ipv6.org — FAQs e informações técnicas
- www.ipv6forum.com — para o pessoal de marketing e propaganda do IPv6

Uma vantagem importante do IPv6 é que ele foi projetado para resolver a questão da renumeração. No mundo do IPv4, os ISPs alocam espaço de endereçamento aos clientes, mas os endereços não são portáveis; quando os clientes saem de um ISP, eles precisam retornar seus endereços e renumerá-los com endereços do seu novo ISP. Com o IPv6, o ISP novo fornece um prefixo de endereço que você simplesmente pré-anexa às partes locais de seus endereços, provavelmente no roteador de borda. Esse esquema é semelhante àquele do NAT para endereçamento IPv4, mas sem nenhum dos pequenos problemas do NAT.

Foram propostos diversos esquemas para facilitar a transição do sistema IPv4 para o IPv6, inclusive o emprego do NAT para ocultar endereços IPv6 enquanto pacotes são tunelados por meio da infra-estrutura IPv4 existente..

12.5 ROTEAMENTO

Roteamento é o processo de direcionar um pacote por meio do emaranhado de redes que se interpõe entre a sua origem e o seu destino. No sistema TCP/IP, ele é similar a pedir informações num país que não conhecemos bem. A primeira pessoa com que você falar pode lhe indicar a cidade correta. Assim que você estivesse um pouco mais próximo de seu destino, a próxima pessoa poderia lhe informar como chegar à rua correta. Você acabaria chegando perto o bastante para que uma outra pessoa fosse capaz de identificar o prédio que você está procurando.

As informações de roteamento TCP/IP assumem a forma de regras ("rotas") como, por exemplo: "Para atingir a rede A, envie pacotes por meio da máquina C". Também pode haver uma rota padrão que informa o que fazer com pacotes destinados a uma rede para a qual não há rota explícita.

As informações de roteamento são armazenadas em uma tabela no kernel. Cada entrada de tabela possui vários parâmetros, entre os quais uma máscara de rede para cada rede listada (outrora opcional, porém agora exigida se a máscara de rede padrão não for correta). Para rotear um pacote para um determinado endereço, o kernel escolhe a rota mais específica daquelas que atendem aos critérios especificados (isto é, aquela com a máscara de rede mais longa). Se o kernel não encontrar nenhuma rota relevante e nenhuma rota padrão, então ele retorna um erro ICMP "network unreachable" (rede inatingível) para o emissor.

A palavra "roteamento" é usada comumente com dois significados:

- Pesquisar um endereço de rede na tabela de roteamento para encaminhar um pacote em direção a seu destino

- Construir a tabela de roteamento primária.

Nesta seção examinaremos a função de encaminhamento e veremos como as rotas podem ser adicionadas ou eliminadas manualmente da tabela de roteamento. Postergamos o tópico mais complicado de protocolos de roteamento que constroem e mantêm a tabela de roteamento para o Capítulo 13, *Roteamento*, a seguir.

Tabelas de roteamento

Você pode examinar a tabela de roteamento de uma máquina com **netstat –r**. Use **netstat -rn para** evitar pesquisas de DNS e apresentar todas as informações numericamente. Discutiremos o comando **netstat** de maneira mais detalhada a partir da página 449, porém eis um pequeno exemplo para lhe dar uma idéia melhor sobre o aspecto das rotas. Esse host possui duas interfaces de rede: 132.236.227.93 (eth0) na rede 132.236.227.0/24 e 132.236.212.1 (eth1) na rede 132.236.212.0/26.

```
$ netstat -rn
Kernel IP routing table
Destination       Genmask          Gateway            Fl    MSS    Iface
132.236.227.0     255.255.255.0    132.236.227.93     U     1500   eth0
default           0.0.0.0          132.236.227.1      UG    1500   eth0
132.236.212.0     255.255.255.192  132.236.212.1      U     1500   eth1
132.236.220.64    255.255.255.192  132.236.212.6      UG    1500   eth1
127.0.0.1         255.255.255.255  127.0.0.1          U     3584   lo
```

O campo de destino normalmente é um endereço de rede; o gateway tem de ser um endereço de host. Por exemplo, a quarta rota diz que para alcançar a rede 132.236.220.64/26, os pacotes têm de ser enviados para o gateway 132.236.212.6 pela interface eth1. A segunda entrada é a rota padrão; pacotes não endereçados explicitamente a qualquer uma das três redes listadas (ou para a máquina em si) serão enviados para o host gateway padrão, 132.236.227.1. Os hosts são capazes de rotear somente para máquinas gateway que estão conectadas diretamente às mesmas redes.

Consulte a página 212 para informações adicionais sobre o ***route****.*

As tabelas de roteamento podem ser configuradas estática, dinamicamente ou com uma combinação dos dois métodos. Uma rota estática é uma que introduzimos explicitamente por meio do comando **route**. Rotas estáticas devem permanecer na tabela de roteamento enquanto o sistema estiver no ar; normalmente elas são configuradas em tempo de inicialização a partir de um dos scripts de inicialização do sistema. Por exemplo, os comandos Linux

```
route add -net 132.236.220.64 netmask 255.255.255.192 gw 132.236.212.6 eth1
route add default gw 132.236.227.1 eth0
```

acrescentariam a quarta e a segunda rotas exibidas pelo comando **netstat -rn** acima. (A primeira e a terceira rotas nessa tela foram adicionadas por **ifconfig** quando as interfaces eth0 e eth1 foram configuradas.)

A rota final também é acrescentada em tempo de inicialização. Ela configura um pseudodispositivo chamado interface loopback. A interface loopback impede que pacotes enviados do host para si mesmo saiam da rede; em vez disso, eles são transferidos diretamente da fila de saída da rede para a fila de entrada da rede dentro do kernel.

Em uma rede local estável, o roteamento estático é uma solução eficiente. Ele é fácil de ser gerenciado e é confiável. Porém, requer que o administrador de sistemas conheça minuciosamente a topologia da rede em tempo de inicialização e que a topologia não mude com freqüência.

A maioria das máquinas em uma rede local tem apenas um caminho para atingir o restante da rede e, portanto, o problema do roteamento é simples. Uma rota padrão adicionada em tempo de inicialização já é suficiente para apontar o caminho a seguir. Hosts que utilizam DHCP (consulte a página 217) para obter seus endereços IP também podem obter uma rota padrão por meio de DHCP.

Para topologias de rede mais complicadas, é necessário o emprego de roteamento dinâmico. O roteamento dinâmico é executado tipicamente por um processo daemon que mantém e modifica a tabela de roteamento. Os daemons de roteamento em hosts distintos se comunicam para descobrir a topologia da rede e para descobrir como atingir destinos distantes. Há disponíveis vários daemons de roteamento.

Redirecionamentos ICMP

Embora o IP em geral não se preocupe propriamente com o gerenciamento das informações de roteamento, ele define efetivamente um pequeno recurso de controle de danos chamado de redirecionamento ICMP. Quando um roteador encaminha um pacote a uma máquina na mesma rede da qual o pacote foi originalmente recebido, com certeza algo está errado. Como o remetente, o roteador e o roteador do próximo salto se encontram todos na mesma rede, o pacote poderia ter sido encaminhado em um salto em vez de dois. O roteador pode concluir que as tabelas de roteamento do remetente são imprecisas ou incompletas.

Nessa situação, o roteador pode notificar o emissor de seu problema enviando o pacote de redirecionamento ICMP. Basicamente, um redirecionamento diz: "Você não deve ficar enviando pacotes para o host *xxx* para mim, mas sim para o host *yyy*". O protocolo ICMP permite que sejam enviados redirecionamentos tanto para endereços de host individuais como para redes inteiras. Entretanto, muitas implementações geram apenas redirecionamentos para hosts; os redirecionamentos para redes são muito menos úteis hoje em dia, pois se aplicam apenas a redes classes A, B ou C.

Após receber um redirecionamento, um emissor inteligente atualiza sua tabela de roteamento de modo que pacotes futuros associados àquele destino adotarão o caminho mais direto. Nos primórdios da multicast, alguns poucos sistemas geravam redirecionamentos de roteamento ICMP em resposta a pacotes multicast. Os sistemas modernos não apresentam esse problema.

A situação ICMP padrão não contém nenhuma etapa de autenticação. Seu roteador recebe um redirecionamento que diz pertencer a outro roteador bem respeitado e o direciona a fim de enviar o tráfego para algum outro lugar. Você deve dar ouvidos a ele? Dar atenção a redirecionamentos cria realmente certo tipo de problema de segurança. Os redirecionamentos geralmente são ignorados pelo Linux (por questões de segurança) e por roteadores Cisco (pois eles são roteadores). Não é uma boa idéia permitir que hosts não confiáveis modifiquem suas tabelas de roteamento.

No Linux, a variável **accept_redirects** na hierarquia **/proc** controla a aceitação de redirecionamentos ICMP. Consulte a página 219 para obter instruções sobre como examinar e reinicializar essa variável.

12.6 ARP: O PROTOCOLO DE CONVERSÃO DE ENDEREÇO

Muito embora os pacotes IP normalmente sejam interpretados em termos de endereços IP, os endereços de hardware têm de ser utilizados para transportarem efetivamente dados por meio da camada de link de uma rede.[14] O ARP, Address Resolution Protocol, descobre o endereço de hardware associado a um determinado endereço IP. Ele pode ser utilizado em qualquer tipo de rede que suporte broadcasting, mas é descrito comumente em termos de redes Ethernet.

Se o host A quer enviar um pacote para o host B na mesma rede Ethernet, ele utiliza o ARP para descobrir o endereço de hardware de B. Se B não se encontrar na mesma rede que A, o host A utiliza o sistema de roteamento para determinar o roteador do próximo salto juntamente com a rota para B e depois utiliza o ARP para descobrir o endereço de hardware deste roteador. Como o ARP utiliza pacotes de transmissão ampla, que não são capazes de cruzar redes,[15] ele só pode ser utilizado para descobrir os endereços de hardware das máquinas conectadas diretamente à rede local do host emissor.

Toda máquina mantém uma tabela na memória chamada cache ARP contendo os resultados de consultas ARP recentes. Sob circunstâncias normais, muitos dos endereços de que um host necessita são descobertos logo após a inicialização, de modo que o ARP não é responsável por grande parte do tráfego de rede.

O ARP funciona enviando por broadcast [16] um pacote do tipo: "Alguém por acaso conhece o endereço de hardware para 128.138.116.4?". A máquina que está sendo procurada reconhece o seu próprio endereço IP e retorna uma resposta: "Sim, esse é o endereço IP atribuído a uma das minhas interfaces de rede e o endereço Ethernet correspondente é 8:0:20:0:fb:6a".

A consulta original inclui os endereços IP e Ethernet do solicitante para que a máquina que está sendo procurada possa responder sem enviar uma consulta ARP própria. Conseqüentemente, as duas máquinas tomam conhecimento dos mapeamentos ARP de cada uma delas utilizando apenas um intercâmbio de pacotes. Outras máquinas que "grampeiam" a broadcast do solicitante também são capazes de registrar o seu mapeamento de endereços. Essa inspeção passiva de tráfego ARP é chamada algumas vezes de *snooping*.

O comando **arp** examina e manipula o cache ARP do kernel , adiciona ou exclui entradas e esvazia ou mostra a tabela. O comando **arp -a** exibe o conteúdo do cache ARP. Por exemplo:

```
redhat$ /sbin/arp -a
sprint-gw (192.168.1.254) at 00:02:4B:5B:26:45 [ether] on eth0
inura-local.toadranch.com (192.168.1.101) at 00:04:76:37:AE:7E [ether] on eth0
```

O comando **arp** é geralmente útil apenas para depuração e situações que envolvam um hardware especial. Alguns dispositivos não são espertos o bastante para compreenderem ARP (por exemplo, impressoras conectadas à rede ou monitores gráficos de propósito especial). Para suportar tais dispositivos, talvez seja necessário configurar uma outra máquina como servidor ARP proxy para o seu hardware incapaz. Isso normalmente também é feito com o comando **arp**. (usando o flag **-s**). Por exemplo:

```
# /sbin/arp -s printer.toadranch.com 00:50:04:ce:ef:38
```

---

14. Exceto em links ponto a ponto, no qual a identidade do destino algumas vezes já está implícita.
15. Os roteadores normalmente podem ser configurados para inundar pacotes de broadcast em outras redes; não faça isso.
16. O ARP usa as convenções de broadcast da camada de link, e não a de broadcast IP.

Os kernels Linux versão 2.4 e posteriores não suportam serviço ARP proxy para uma sub-rede inteira, porém, atuarão automaticamente como um servidor ARP proxy quando existir uma rota e a interface for configurada para encaminhar pacotes.

Se dois hosts em uma rede estiverem utilizando o mesmo endereço IP, um deles tem a entrada correta na tabela ARP e o outro, a errada. Você pode utilizar o comando **arp** para monitorar a máquina problemática.

Algumas vezes, endereços de hardware precisam ser convertidos em endereços IP. Uma série de equipamentos (por exemplo, estações de trabalho sem disco, computadores de rede, impressoras) precisa fazer essa conversão em tempo de inicialização. Em vez de ter um endereço IP definido diretamente em um arquivo de configuração, uma máquina pode consultar um servidor central para descobrir o seu próprio endereço.

Isso encerra nossa cobertura sobre fundamentos de rede. Nas seções a seguir, tratamos das questões envolvidas na configuração de máquinas Linux para sua máquina local e Internet.

12.7 ADIÇÃO DE UMA MÁQUINA A UMA REDE

Somente alguns poucos passos estão envolvidos na adição de uma máquina nova a uma rede local existente, porém alguns fornecedores ocultam os arquivos que você tem de modificar e geralmente dificultam essa tarefa. Outros fornecem um script de configuração que solicita parâmetros de rede necessários, que é bom até o momento em que você precisar desfazer algo ou transferir uma máquina. Antes de ativar uma máquina nova em uma rede que está conectada à Internet, você deve executar procedimentos de segurança nesta máquina (Capítulo 20, *Segurança*) para que você não convide inadvertidamente hackers para sua rede local.

Os passos básicos para se adicionar uma nova máquina a uma rede local são:

- Atribuir um nome de host e um endereço IP exclusivos
- Definir o novo host para configurar suas interfaces de rede em tempo de inicialização
- Configurar uma rota padrão e talvez roteamento mais complexo
- Apontar para um servidor de nomes DNS, para possibilitar o acesso ao restante da Internet

Obviamente, você também poderia acrescentar uma etapa de depuração a essa seqüência. Após qualquer modificação que possa vir a afetar a inicialização, você sempre deve reinicializar para verificar se a máquina é acionada corretamente. Seis meses depois quando a alimentação falhar e a máquina se recusar a inicializar, será difícil de se lembrar que alteração você fez que poderia estar causando o problema. (Consulte também o Capítulo 19, *Gerenciamento e depuração de redes*.)

Se sua rede utiliza DHCP (Dynamic Host Configuration Protocol), o servidor DHCP fará essas tarefas para você. Consulte a seção sobre DHCP que inicia na página 217 para obter informações gerais sobre o DHCP e as especificidades de como configurar as distribuições adotadas neste livro a fim de utilizar o DHCP em tempo de inicialização.

Primeiro, abrangemos o esquema geral desses passos e então retornaremos aos detalhes sobre a magia das distribuições especiais de cada fornecedor em uma série de seções específicas a fornecedores. Os comandos envolvem a camada de rede do kernel do Linux e são os mesmos em várias distribuições diferentes. Porém, cada distribuição estabeleceu seus próprios arquivos de configuração para configuração de rede automática em tempo de inicialização, conforme sintetizado na Tabela 12.9.

Tabela 12.9 Arquivos de configuração de rede

Sistema	Arquivo	O que é configurado aí
Red Hat, Fedora	**/etc/sysconfig/network**	Nome do host, rota padrão
	network-scripts/ifcfg-*ifname*	endereço IP, máscara de rede, endereço de broadcast
SUSE	**/etc/rc.config**	Nome do host, endereço IP, máscara de rede e outros
	/etc/route.conf	Rota-padrão
Debian, Ubuntu	**/etc/hostname**	Nome de host
	/etc/network/interfaces	Endereço IP, máscara de rede, rota padrão

O processo de projetar e instalar uma rede física é visto no Capítulo 14, *Hardware de Rede*. Caso esteja lidando com uma rede existente e tenha uma noção geral de como ela está configurada, talvez não seja necessário você ler muito sobre os aspectos físicos de redes a menos que planeje estender a rede existente.

Descrevemos o processo de configuração de rede em termos de Ethernet; outras tecnologias são basicamente similares.

Atribuição de endereços IP e de hostname

Consulte o Capítulo 15 para informações adicionais sobre DNS.

Os administradores têm várias teorias sobre como o mapeamento entre nomes de host (hostnames) e endereços IP é mais bem mantido em uma instalação local: por meio do arquivo **hosts**, do NIS, do sistema DNS ou, quem sabe, alguma combinação destas. Se tiver vários sistemas, você também precisará ter um plano sensato sobre como esses sistemas combinados devem funcionar. Os valores conflitantes são escalabilidade e manutenção *versus* um sistema que seja suficientemente flexível para permitir que máquinas inicializem quando nem todos os serviços estarão disponíveis (e flexíveis o bastante para lidar com a heterogeneidade de sua rede).

Outra questão a longo prazo que pode ser relevante é a da renumeração. Se sua instalação mudar de ISP, talvez você precise devolver os seus antigos endereços IP e fazer uma renumeração de acordo com endereços atribuídos pelo novo ISP. Esse processo torna-se bastante desalentador se você tiver de visitar cada host da rede para reconfigurá-lo. Para agilizar tal processo de renumeração, use nomes de host nos arquivos de configuração e faça a conversão nome de host-endereço IP somente nos arquivos de bancos de dados DNS. Por outro lado, utilizar endereços IP nos arquivos de configuração reduz as dependências durante a inicialização quando nem todos os serviços estarão disponíveis. Ruim se você faz, ruim se você não faz.

O arquivo **/etc/hosts** é a maneira mais antiga e simples de mapear nomes para endereços IP. Cada linha começa com um endereço IP e prossegue com os diversos nomes simbólicos pelos quais esse endereço é conhecido.

Eis um arquivo **/etc/hosts** típico para o host lollipop:

```
127.0.0.1          localhost
192.108.21.48      lollipop.xor.com lollipop loghost
192.108.21.254     chimchim-gw.xor.com chimchim-gw
192.108.21.1       ns.xor.com ns
192.225.33.5       licenses.xor.com license-server
```

Uma versão minimalista só teria as duas primeiras linhas. localhost costuma ser a primeira entrada no arquivo **/etc/hosts**.

Pelo fato de **/etc/hosts** conter somente mapeamentos locais, a maioria dos sistemas modernos o utiliza apenas para mapeamentos necessários em tempo de inicialização. O DNS é então consultado para localizar mapeamentos para o restante da rede local e para todos os demais. Você também pode usar **/etc/hosts** para especificar mapeamentos que você não quer que outras pessoas conheçam e, portanto, não publicar em DNS.

/etc/hosts já foi importante durante o processo de inicialização, pois o DNS ainda não se encontrava disponível e os nomes de host eram algumas vezes utilizados em arquivos de configuração em vez de endereços IP. As distribuições Linux modernas realmente não precisam do arquivo **/etc/hosts**, mas provavelmente devem conter ao menos os mapeamentos para o próprio host e o endereço loopback. Mapeamentos para a máquina gateway padrão e um servidor de nomes provavelmente também seriam úteis. Muitas instalações colocam todos os seus hosts, servidor e gateways realmente importantes no arquivo **/etc/hosts**. Outros colocam somente o host em si e a interface loopback; outros ainda colocam todos os hosts locais e seus servidores de nomes de backup fora do local de trabalho.

Nossos sistemas Linux de exemplo instalam uma miscelânea de arquivos **/etc/hosts**. Os arquivos **/etc/hosts** do Debian e do Ubuntu contêm localhost em 127.0.0.1, o hostname do sistema em 127.0.1.1 e algumas informações sobre o IPv6. Os arquivos **/etc/hosts** do Red Hat e do Fedora definem o localhost e a própria máquina. O do SUSE contém endereços do localhost e endereços IPv6 para localhost e alguns nomes IPv6 especiais.

O comando **hostname** atribui um nome de host a uma máquina. **hostname** é normalmente executado em tempo de inicialização a partir de um dos scripts de inicialização, que obtém o nome a ser atribuído a partir do arquivo de configuração. Obviamente, cada fornecedor dá nomes diferentes a esse arquivo de configuração. Consulte as seções específicas a cada fornecedor, a partir da página 215, para obter maiores informações sobre sua distribuição específica. A maioria dos sistemas hoje em dia atribui um nome totalmente qualificado (isto é, um nome que inclui tanto o nome do host quanto o nome de domínio DNS, por exemplo, anchor.cs.colorado.edu).

Consulte a página 361 para obter maiores informações sobre LDAP.

Em uma instalação pequena, você pode distribuir facilmente nomes de host e endereços IP manualmente. Mas, quando várias redes e vários grupos administrativos distintos estão envolvidos, ajuda ter alguma espécie de coordenação central para garantir exclusividade. Para parâmetros de rede atribuídos dinamicamente, o DHCP cuida das questões de exclusividade. Algumas instalações utilizam agora bancos de dados LDAP para gerenciarem suas atribuições de nomes de host e endereços IP.

ifconfig: configurando interfaces de rede

ifconfig ativa ou desativa uma interface de rede, configura o seu endereço IP e máscara de sub-rede e configura várias outras opções e parâmetros. Ele é normalmente executado em tempo de inicialização (com parâmetros de linha de comando extraídos de arquivos de configuração), mas também pode fazer alterações com o sistema em andamento. Tome cuidado se estiver fazendo alterações **ifconfig** e estiver conectado remotamente; muitos administradores de sistemas ficaram bloqueados desta maneira e tiveram de se dirigir até o local para corrigir o problema.

Um comando **ifconfig** mais comumente assume a seguinte forma:

```
ifconfig interface endereço opções … up…
```

por exemplo:

```
ifconfig eth0 192.168.1.13 netmask 255.255.255.0 up
```

interface identifica a interface de hardware a qual o comando se aplica. Em sistemas UNIX esse normalmente é um nome de dispositivo de dois a três caracteres (derivados do chipset utilizado na placa de rede) seguido de um número, mas, para o Linux, quase sempre é algo do tipo eth0.[17] A identidade real do hardware e o mapeamento para um driver de dispositivo apropriado são armazenados no arquivo **/etc/modprobe.conf** em uma linha `alias`. A interface loopback chama-se lo.

ifconfig *interface* exibe as configurações atuais para *interface* sem alterá-las. Muitos sistemas interpretam **-a** como "all interfaces", ou seja, todas as interfaces, e **ifconfig -a** pode, conseqüentemente, ser utilizado para descobrir que interfaces estão presentes no sistema. Se o seu sistema não compreende **ifconfig -a**, tente **netstat -i** para descobrir os nomes das interfaces.

O parâmetro *endereço* especifica o endereço IP da interface. Muitas versões de **ifconfig** também aceitam um nome de host para o parâmetro de endereço. Preferimos utilizar o endereço IP real; se **ifconfig** receber um nome de host (ou a saída do comando **hostname**), o potencial para surgimento de problemas na hora da inicialização aumenta. Se houver um problema na determinação do nome de host, a máquina não inicializará ou inicializará em um estado no qual não possa ser acessada da rede, exigindo que você vá pessoalmente até a máquina para depurar o problema. Consultas DNS que não são capazes de ser completadas demoram para expirar, fazendo parecer que a máquina está travada. Por outro lado, se alguma vez você tiver de renumerar a sua rede, localizar todos esses endereços IP definidos diretamente e ocultos nos arquivos de configuração pode se tornar um pesadelo.

A palavra-chave **up** ativa a interface; **down** a desativa. Quando um comando **ifconfig** atribui um endereço IP a uma interface, como no exemplo anterior, o parâmetro **up** está implícito e não precisa ser mencionado pelo nome.

ifconfig reconhece muitas outras opções. Abordaremos somente as mais comuns; como sempre, consulte suas páginas man para a palavra final sobre o seu sistema em particular. Algumas opções exigem um argumento, que poderia ser colocado imediatamente após o nome da opção.

A opção **netmask** configura a máscara de sub-rede para a interface e é necessária caso a rede não esteja dividida em sub-redes de acordo com sua classe de endereços (A, B ou C). A máscara pode ser especificada na notação ponto decimal ou na forma de um número hexadecimal de quatro bytes começando com **0x**. Em qualquer um dos casos, bits configurados como 1 fazem parte do número da rede e bits configurados como 0 fazem parte do número do host.

A opção **broadcast** especifica o endereço IP de broadcast para a interface, expresso tanto em notação hexadecimal como na que usa pontos. O endereço de broadcast correto é um no qual a parte host esteja configurada toda como 1s, e a maioria dos sistemas assume como padrão esse valor; eles utilizam a máscara de rede e o endereço IP para calcular o endereço de broadcast.

No Linux, você pode configurar o endereço de broadcast para qualquer endereço IP que seja válido para a rede na qual o host está conectado. Alguns sites optaram por valores estranhos como endereços de broadcast na esperança de evitarem determinados tipos de ataques de rejeição de serviços que se baseiam em pings de broadcast. Não gostamos dessa abordagem por diversas razões.

Primeiramente, ela requer que você reinicialize o endereço de broadcast em cada host da rede local, o que pode consumir muito tempo em uma rede extensa. Segundo, exige que você esteja absolutamente seguro de que cada host foi reconfigurado; do contrário, podem acontecer congestionamentos de transmissões (*broadcast storms*), nas quais os pacotes viajam de máquina para máquina até seus TTLs expirarem.

Os congestionamentos de transmissões ocorrem porque o mesmo endereço de broadcast da camada de link tem de ser utilizado para transportar pacotes independentemente de como o endereço IP de broadcast foi configurado. Suponha, por exemplo, que a máquina X imagina que o endereço de broadcast seja A1 e que a máquina Y imagine que ele seja A2. Se X enviar um pacote para o endereço A1, Y receberá o pacote (pois o endereço de destino da camada de link é o endereço de broadcast), verá que o pacote não é para ele mesmo e nem para o endereço de broadcast (pois Y pensa que o endereço de broadcast é A2) e então encaminhará[18] o pacote de volta para a rede. Se duas máquinas estiverem no estado de Y, o pacote circulará até que o seu TTL expire. Congestionamentos de transmissões podem acabar com sua largura de banda, especialmente em uma grande rede comutada.

---

17. Você pode atribuir mais de um endereço IP a uma interface fazendo uso do conceito de "interfaces de rede virtuais" ou "aliases de IP". Os administradores de sistemas fazem isso com freqüência para permitirem que uma máquina hospede vários sites Web. Em sistemas Linux, as interfaces virtuais são denominadas eth0:0, eth0:1 e assim por diante. Você não precisa declarar as interfaces com antecedência, apenas aplicar-lhes o comando **ifconfig** para configurá-las. Consulte a página 501 para obter informações adicionais.

18. A máquina Y deve ser configurada com `ip_forwarding` ativado para que isso aconteça.

Uma forma melhor de evitar problemas com pings de broadcast é impedir que seus roteadores de borda os encaminhem e informar hosts individuais para não responderem a eles. Consulte a página 220 para obter instruções de como implementar essas restrições.

No exemplo de **ifconfig** no início desta seção, o endereço de broadcast é 192.168.1.255, pois a rede é do tipo /24, conforme especificado pelo valor de máscara de rede de 255.255.255.0.

Executar **ifconfig** gera o seguinte:

```
redhat$ /sbin/ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:02:B3:19:C8:86
          inet addr:192.168.1.13  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:206983 errors:0 dropped:0 overruns:0 frame:0
          TX packets:218292 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:7 Base address:0xef00
```

A falta de colisões na interface Ethernet deste exemplo pode indicar uma rede pouco carregada ou, mais provavelmente, uma rede comutada. Em uma rede compartilhada (construída com hubs em vez de switches), devemos verificar esse número para garantir que ele esteja abaixo de cerca de 5% dos pacotes de saída. Um grande número de colisões indica uma rede carregada que precisa ser observada e possivelmente dividida em várias sub-redes ou então migrada para uma infraestrutura comutada.

Vejamos alguns exemplos completos.

```
# ifconfig lo 127.0.0.1 up
```

Esse comando configura a interface loopback, que normalmente não requer a configuração de nenhuma opção. Você jamais deve alterar a configuração padrão de seu sistema para essa interface. A máscara de rede implícita de 255.0.0.0 é correta e não precisa ser sobrescrita manualmente.

```
# ifconfig eth0 128.138.243.151 netmask 255.255.255.192
    broadcast 128.138.243.191 up
```

Esse é um típico exemplo de uma interface Ethernet. Os endereços IP e de broadcast são configurados, respectivamente, como 128.138.243.151 e 128.138.243.191. A rede é classe B (isso pode ser visto a partir do primeiro byte do endereço), mas foi dividida em sub-redes por 10 bits adicionais em uma rede /26. O 192 na máscara de rede é 11000000 em binário e, portanto, adiciona 2 bits extras aos 24 contidos nos três octetos 255. O 191 no endereço de broadcast é 10111111 em binário, que configura todos os 6 bits de host em 1 e indica que essa interface faz parte da terceira rede (os dois primeiros bits são 10) no grupo de 4 extraídos do 4º octeto. (Esse é o tipo de situação em que uma calculadora IP é útil!)

Agora que já sabe como configurar manualmente uma interface de rede, você precisa descobrir como os parâmetros de **ifconfig** são configurados quando a máquina é inicializada e precisa certificar-se de que os novos valores sejam introduzidos corretamente. Em geral, você faz isso editando um ou mais arquivos de configuração; consulte as seções específicas a cada fornecedor a partir da página 215 para informações adicionais.

mii-tool: configurando autonegociação e outras opções específicas da mídia

Ocasionalmente, o hardware de rede tem opções configuráveis que são específicas ao seu tipo de mídia. Um exemplo extremamente comum disto é a Ethernet dos dias atuais, em que uma placa pode suportar 10, 100 ou até mesmo 1000 MB/s tanto em half-duplex quanto em full-duplex. A maioria dos equipamentos assume como padrão o modo de autonegociação, no qual tanto a placa como a conexão acima (normalmente uma porta de um switch) tentam adivinhar o que o outro deseja utilizar.

Historicamente, a autonegociação funcionava tão bem quanto um vaqueiro de olhos vendados tentando laçar um bezerro. Mais recentemente, os fornecedores de dispositivos de rede começaram a trabalhar em conjunto, mas a autonegociação continua sendo uma fonte comum de falhas. Altas taxas de perdas de pacotes (especialmente de pacotes grandes) normalmente são resultado de autonegociação que falhou.

A melhor maneira de evitar essa armadilha é fixar a velocidade da interface e o modo duplex tanto nos servidores quanto nas portas de switch nas quais eles estão conectados. A autonegociação é útil para portas em áreas públicas em que laptops de usuários em percurso podem dar uma parada para uma visita, mas não é de nenhuma utilidade para hosts conectados estaticamente. Se estiver tendo problemas com perdas misteriosas de pacotes, desative a autonegociação em todos os pontos como primeira medida.

No Linux, o comando **mii-tool** consulta e configura parâmetros específicos de mídia como velocidade do link e modo duplex. Você pode verificar o estado de uma interface por meio do flag **-v**. Por exemplo, a interface eth0 a seguir tem a autonegociação ativada:

```
$ mii-tool -v eth0
eth0: negotiated 100baseTx-FD flow-control, link ok
  product info:    vendor 00:10:5a, model 0 rev 0
  basic mode:      autonegotiation enabled
  basic status:    autonegotiation complete, link ok
  capabilities:    100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:     100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
  link partner:    100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

Para fixar essa interface em modo full-duplex 100 MB/s, utilize o comando

```
# mii-tool -force=100BaseTx-FD eth0
```

Adicione esse comando a um script de inicialização de sistema para torná-lo permanente. Após isso, a consulta de estado retorna

```
$ mii-tool -v eth0
eth0: 100 Mbit, full duplex, link ok
  product info:    vendor 00:10:5a, model 0 rev 0
  basic mode:      100 Mbit, full duplex
  basic status:    link ok
  capabilities:    100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
  advertising:     100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow-control
```

route: configurando rotas estáticas

O comando **route** define roteamentos estáticos, entradas explícitas na tabela de roteamento que nunca mudam (assim esperamos), mesmo se você executar um daemon de roteamento. Ao adicionar uma nova máquina a uma rede local, normalmente você só precisa especificar uma rota padrão; consulte a próxima seção para maiores detalhes.

A discussão sobre roteamento deste livro é dividida entre esta seção e o Capítulo 13, *Roteamento*. Embora grande parte das informações básicas sobre roteamento e o comando **route** esteja aqui, talvez você ache interessante ler as primeiras seções do Capítulo 13 se precisar de mais informações.

O roteamento é realizado na camada IP. Quando chega um pacote vinculado a algum outro host, o endereço IP de destino do pacote é comparado com as rotas na tabela de roteamento do kernel. Se ela coincidir total ou parcialmente com uma rota da tabela, o pacote é encaminhado para o endereço IP do "gateway de próximo salto" associado a essa rota.

Há dois casos especiais. Primeiramente, um pacote pode ser destinado a algum host em uma rede diretamente conectada. Nesse caso, o endereço de "gateway de próximo salto" na tabela de roteamento é uma das próprias interfaces do host local, e o pacote é enviado diretamente a seu destino. Esse tipo de rota é adicionado à tabela de roteamento pelo comando **ifconfig** ao configurarmos uma interface.

Em segundo lugar, pode ser que nenhuma rota coincida com o endereço de destino. Nesse caso, se existir uma rota padrão, ela é invocada. Caso contrário, será retornada uma mensagem ICMP "network unreachable" ou "host unreachable" (rede ou host inatingíveis) ao remetente. Muitas redes locais possuem apenas um caminho, e sua rota padrão aponta para ele. No backbone Internet, os roteadores não possuem rotas padrão — o negócio pára por aí mesmo. Se eles não têm uma entrada de roteamento para um destino, esse destino não pode ser atingido.

Cada comando **route** acrescenta ou elimina uma rota. O formato normalmente é

```
route [op] [tipo] destino gw gateway [métrica] [dev interface]
```

O argumento *op* deve ser **add** para adicionar uma rota, **del** para eliminar uma rota, e deve ser omitido para exibir as tabelas de roteamento. *destino* pode ser um endereço de host (tipo **-host**), um endereço de rede (tipo **-net**) ou então a palavra-chave **default**. Se *destino* for um endereço de rede, você também deve especificar uma máscara de rede.

gateway é a máquina para a qual os pacotes devem ser encaminhados. Ela *deve obrigatoriamente* estar em uma rede diretamente conectada; o encaminhamento pode ser realizado somente um salto por vez. O Linux permite especificar uma interface em vez do (ou juntamente com o) *gateway*. A palavra-chave **dev** na especificação interface é opcional e pode ser omitida.

métrica é o número de encaminhamentos (a contagem de saltos) necessário para se atingir o destino. O Linux não requer nem utiliza a contagem de saltos, porém mantém o valor nas tabelas de roteamento se você a ativar, para que os protocolos de roteamento possam utilizá-la.

O argumento opcional *tipo* suporta rotas de host, que se aplicam a um endereço IP completo (um host específico) em vez de um endereço de rede. Os valores **-net** e **-host** são aceitos para o parâmetro *tipo*. Se não for especificado um tipo (*type*), **route** verificará a parte host do endereço de destino para ver se ela é zero. Se a parte do host for 0 ou o endereço for uma rede definida no arquivo **/etc/networks** (se seu sistema tiver um), assumimos então que a rota é uma rota normal de

rede.[19] Como **route** não pode saber em um passe de mágica que números de rede foram divididos em sub-redes, freqüentemente temos de utilizar o campo *tipo* para instalar certas rotas. Por exemplo, o endereço 128.138.243.0 se refere a uma rede classe B dividida em sub-rede em nossa instalação, porém, para o comando **route**, ela se parece com um endereço classe B de 128.138 com uma parte host 243.0; você deve especificar a opção **-net** para eliminar essa confusão feita por **route**. Geralmente, é uma boa prática fornecer um tipo (*type*) explícito para todas as rotas que envolvam sub-redes.

route del *destino* elimina uma entrada específica da tabela de roteamento. Outros sistemas UNIX possuem uma opção para **route**, usualmente **-f** ou **-flush**, que descarregam completamente as tabelas de roteamento e reiniciam. O Linux não suporta essa opção, de forma que talvez você se depare com muitos **route dels** para limpar uma grande tabela de roteamento — certifique-se de estar conectado localmente ou você pode acabar não terminando o serviço e desconectado!

Para inspecionar rotas existentes, utilize o comando **netstat -nr** ou **netstat -r** se quiser ver os nomes em vez de números. Em geral, números são melhores se você estiver depurando, uma vez que a pesquisa de nome pode ser o item problemático.

```
redhat$ netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS    Window  irtt   Iface
192.168.1.0     0.0.0.0         255.255.255.0   U         0     0          0   eth0
127.0.0.0       0.0.0.0         255.0.0.0       U         0     0          0   lo
0.0.0.0         192.168.1.254   0.0.0.0         UG        0     0          0   eth0

redhat$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS    Window  irtt   Iface
192.168.1.0     *               255.255.255.0   U         0     0          0   eth0
127.0.0.0       *               255.0.0.0       U         0     0          0   lo
default         sprint-gw       0.0.0.0 UG      0         0     0              eth0
```

Genmask é a máscara de rede associada ao destino. Flags especificam o estado da rota, como ela foi descoberta e outros parâmetros. Finalmente, Iface é a interface pela qual os pacotes utilizando essa rota são enviados. Esses exemplos são de um sistema Red Hat, porém o SUSE e o Debian são idênticos, exceto pelo fato de o Debian não exibir a rota loopback como padrão.

Rotas padrão

Uma rota padrão faz com que todos os pacotes cuja rede de destino não é encontrada na tabela de roteamento do kernel sejam enviados para o gateway indicado. Para configurar uma rota padrão, basta adicionar a seguinte linha a seus arquivos de inicialização:

```
route add default gw endereço IP do gateway
```

Em vez de colocar um endereço IP explícito definido diretamente nos arquivos de inicialização, a maioria dos fornecedores faz com que seus sistemas obtenham o endereço IP de gateway a partir de um arquivo de configuração. A maneira pela qual informações de roteamento locais são integradas na seqüência de inicialização infelizmente é diferente para cada um de nossos sistemas Linux (vamos logo, LSB, corrija essa síndrome não inventada aqui!). A Tabela 12.10 sintetiza as mágicas necessárias.

Tabela 12.10 Como configurar a rota padrão

Sistema	Arquivo a ser alterado	Variável a ser alterada
Red Hat, Fedora	/etc/sysconfig/network	`GATEWAY`
SUSE	/etc/route.conf	add line: `default` *IP-addr mask interface*
Debian, Ubuntu	/etc/network/interfaces	`gateway`

Configuração de DNS

Para configurar uma máquina como um cliente DNS, basta editar um ou dois arquivos: todos os sistemas requerem que **/etc/resolv.conf** seja modificado e alguns exigem que você também modifique um arquivo de "service switch".

19. **/etc/networks** pode mapear nomes para números de rede, quase como o arquivo **/etc/hosts** mapeia hostnames para completar endereços IP. Muitos comandos que esperam um número de rede podem aceitar um nome caso ele esteja listado no arquivo **/etc/networks** (ou em DNS).

O arquivo **/etc/resolv.conf** lista os domínios DNS que devem ser pesquisados para determinar os nomes que estão incompletos (isto é, não totalmente qualificados, como anchor em vez de anchor.cs.colorado.edu) e os endereços IP dos servidores de nomes a contatar para pesquisas de nomes. A seguir veja uma amostra; para mais detalhes, consulte a página 291.

```
search cs.colorado.edu colorado.edu
nameserver 128.138.242.1
nameserver 128.138.243.151
nameserver 192.108.21.1
```

/etc/resolv.conf deve listar primeiramente o servidor de nomes estável "mais próximo", pois o servidor na primeira posição será contatado primeiro. Você pode ter até três entradas `nameserver`. Se possível, você deve sempre ter mais de uma. O período de expiração de uma consulta DNS para um determinado servidor de nomes parece demorado, de modo que se o primeiro servidor de nomes não responder, seus usuários perceberão.

Algumas vezes você verá a linha `domain` em vez de uma linha `search`. Essa linha indica um arquivo **resolv.conf** antigo que não foi atualizado para que possa utilizar a diretiva `search` ou um conversor antigo que não entende `search`. `domain` define apenas o domínio atual, enquanto `search` aceita até seis domínios diferentes para consulta. Portanto, `search` é preferível.

Consulte o Capítulo 17 para informações adicionais sobre o NIS.

Além disso, alguns sistemas antigos não utilizam o DNS por padrão, mesmo se houver um arquivo **resolv.conf** adequadamente configurado. Esses sistemas possuem um arquivo "service switch" que determina que mecanismos serão utilizados para determinar os mapeamentos nome de host para endereço IP. A priorização de fontes de informação é abordada de forma mais detalhada a partir da página 358, mas mencionamos o tópico aqui também, já que ele algumas vezes frustra suas tentativas de configurar uma máquina antiga.

O arquivo service switch permite especificar a ordem na qual DNS, NIS e **/etc/hosts** devem ser consultados. Na maioria dos casos, você também pode excluir certas fontes de dados. Sua escolha da ordem tem um impacto sobre a capacidade da máquina de inicializar e sobre a maneira como a inicialização interage com o conteúdo do arquivo **/etc/hosts**.

Se o DNS for escolhido como a primeira fonte de dados a ser consultada, talvez seja necessário ter um servidor de nomes na rede local e ter o seu nome de host e o endereço IP no arquivo **hosts** para tudo funcionar em tempo de inicialização.

A Tabela 12.11 apresenta a localização dos arquivos de configuração relevantes e a configuração padrão para pesquisas de host em cada um de nossos sistemas escolhidos como exemplo.

Tabela 12.11 Arquivos service switch por sistema

Sistema	Arquivo switch	Padrão para pesquisas de nomes de host
Ubuntu	/etc/nsswitch.conf	`files dns mdns`[a]
	/etc/host.conf	`hosts, bind`
Outros	/etc/nsswitch.conf	`files dns`
	/etc/host.conf	`hosts, bind`

a. mdns = multicast DNS, um protocolo pouco comum que permite a resolução de nomes como um DNS em pequenas redes que não tenham servidores DNS.

A estrutura de rede do Linux

Consulte a página 501 para mais informações sobre interfaces.

A estrutura de rede no kernel 2.2 e nas versões superiores do Linux suporta interfaces de rede virtual e reconhecimentos seletivos (ou SACKs, como são chamados). Os kernels 2.4 e superiores implementam uma notificação explícita de congestionamento (*explicit congestion notification* – ECN).

A ECN marca os pacotes TCP para informar o peer remoto sobre problemas de congestionamento em vez de deixar que pacotes descartados sirvam como a única indicação de que algo saiu errado. O ECN foi especificado originalmente na RFC2481 (janeiro de 1999) e agora é um padrão proposto, documentado na RFC3168. A RFC2884 (julho de 2000) incluiu uma avaliação do desempenho da ECN. Essa RFC descobriu que a ECN se beneficiava de uma variedade de transações de rede.

O Linux sempre é uma das primeiras estruturas de rede a incluir novos recursos. O pessoal do Linux às vezes é tão rápido que o restante da infra-estrutura de rede não pode interoperar. Por exemplo, o recurso ECN do Linux (que é ativo como padrão) colidiu com ajustes padrão de configuração incorretos num produto firewall da Cisco mais antigo, fazendo com que todos os pacotes com o bit ECN ativo fossem perdidos. Opa!

Os desenvolvedores Linux adoram experimentar e eles freqüentemente implementam recursos e algoritmos que ainda não são padrões aceitos. Um exemplo é a adição na versão 2.6.13 do Linux dos algoritmos de controle de congestionamento plugáveis. As várias opções incluem variações para redes com perda, redes remotas de alta velocidade com uma grande quantidade de perda de pacotes, links de satélite e mais. O mecanismo "reno" TCP padrão (inicialização lenta, contenção de congestionamento [*congestion avoidance*], retransmissão rápida e recuperação rápida) ainda é utilizado por padrão, mas uma variante pode ser mais adequada ao seu ambiente.

12.8 CONFIGURAÇÃO DE REDE ESPECÍFICA DO FORNECEDOR

O Capítulo 2 descreve os detalhes dos procedimentos de inicialização das distribuições apresentadas como exemplo neste livro. Nas próximas seções, simplesmente sintetizamos as tarefas que estão relacionadas à configuração de uma rede. Os sistemas aqui apresentados configuram a interface loopback automaticamente; você nunca precisará modificar essa parte da configuração. Além disso, cada sistema difere um pouco um do outro.

Quatro arquivos são comuns a cada um dos nossos sistemas abordados: **/etc/hosts**, **/etc/resolv.conf**, **/etc/nsswitch.conf** e **/etc/host.conf**. Estes foram abrangidos nas seções anteriores sobre configuração genérica de rede e, exceto para **resolv.conf** e possivelmente **hosts**, normalmente não precisam ser modificados quando você adiciona uma máquina à rede.

Após qualquer modificação em um arquivo que controla a configuração de rede em tempo de inicialização, talvez você também precise reinicializar ou então desativar a interface de rede e depois ativá-la novamente para que a alteração entre em vigor. Em todas as nossas distribuições de exemplo você pode utilizar os comandos **ifup** e **ifdown**.

Configuração de rede para o Red Hat e o Fedora

A Tabela 12.12 mostra os arquivos de configuração de rede do Red Hat e do Fedora.

Tabela 12.12 Arquivos de configuração de rede do Red Hat e do Fedora

Arquivo em /etc/sysconfig	O que é configurado aí
network	Nome do host, rota padrão
static-routes	Rotas estáticas
network-scripts/ifcfg-*ifname*	Parâmetros por interface: Endereço IP, máscara de rede etc.

Você configura o hostname da máquina em **/etc/sysconfig/network**, que também contém linhas que especificam o domínio DNS e o gateway padrão da máquina. Por exemplo, eis um arquivo **network** para um host que possui uma única interface Ethernet:

```
NETWORKING=yes
HOSTNAME=redhat.toadranch.com
DOMAINNAME=toadranch.com  ### optional
GATEWAY=192.168.1.254
```

Dados específicos a uma interface que são armazenados em **/etc/sysconfig/network-scripts/ifcfg-***ifname*, onde *ifname* é o nome da interface de rede. Esses arquivos de configuração nos permitem configurar o endereço IP, as máscaras de rede e os endereços de rede e de broadcast para cada interface. Eles também incluem uma linha que especifica se a interface deve ser configurada como "ativa" em tempo de inicialização.

Tipicamente, arquivos para uma interface Ethernet (eth0) e para a interface loopback (lo) estão presentes. Por exemplo,

```
DEVICE=eth0
IPADDR=192.168.1.13
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
```

e

```
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

são os arquivos **ifcfg-eth0** e **ifcfg-lo** para a máquina redhat.toadranch.com descrita no arquivo **network** anteriormente nesta seção.

Alguns scripts úteis facilitam gerenciamento de interface. **ifup** e **ifdown** aceitam o nome de uma interface de rede como um argumento e ativam ou desativam a interface especificada. Após alterar as informações de rede em qualquer um dos diretórios **/etc/sysconfig** , certifique-se de executar **ifdown** *ifname* seguido de **ifup** *ifname*. Melhor ainda, reinicialize o sistema para certificar-se de que suas alterações não provocarão nenhum tipo de problema sutil. Não há nenhuma página man

de **ifup** e **ifdown**, porém eles são scripts de shell (mantidos em **/sbin**), de modo que você pode dar uma olhada e verificar o que eles fazem detalhadamente.

Se você precisar gerenciar todas as interfaces de uma só vez, execute o script **/etc/rc.d/init.d/network** que aceita os argumentos **start**, **stop**, **restart** e **status**. Esse script é chamado em tempo de inicialização com o argumento **start**.

Os scripts de inicialização também podem configurar rotas estáticas. Qualquer rota adicionada ao arquivo **/etc/sysconfig/static-routes** é inserida na tabela de roteamento em tempo de inicialização. As entradas fornecem argumentos a **route add**, embora em uma ordem misturada (a interface está em primeiro lugar em vez de último):

```
eth0 net 130.225.204.48 netmask 255.255.255.248 gw 130.225.204.49
eth1 net 192.38.8.0 netmask 255.255.255.224 gw 192.38.8.129
```

A interface é especificada primeiro, seguida por argumentos para **route**: o tipo de rota (`net` ou `host`), a rede de destino, a máscara de rede associada a essa rede e, finalmente, o gateway do próximo salto. A palavra-chave `gw` é exigida. Os kernels Linux atuais não usam o parâmetro métrico (`metric`) para **route**, mas permitem que ele seja introduzido e mantido na tabela de roteamento para que os daemons de roteamento o usem. O exemplo **static-routes** anterior produziria os seguintes comandos **route**:

```
route add -net 130.225.204.48 netmask 255.255.255.248 gw 130.225.204.49 eth0
route add -net 192.38.8.0 netmask 255.255.255.224 gw 192.38.8.129 eth1
```

Configuração de rede para o SUSE

A Tabela 12.13 a seguir mostra os arquivos de configuração de rede utilizados pelo SUSE.

SUSE tem um esquema único de configuração de rede. Com as exceções dos parâmetros de DNS e do nome de host de sistema, o SUSE configura a maioria das opções de configuração de rede nos arquivos **ifcfg-***interface* no diretório **/etc/sysconfig/network**. Um arquivo deve estar presente para cada interface no sistema.

Tabela 12.13 Arquivos de configuração de rede do SUSE em /etc/sysconfig/network

Arquivo	O que é configurado aí
ifcfg-*interface*	Nome do host, endereço IP, máscara de rede e outros
ifroute-*interface*	Definições de rota específicas às interfaces
routes	Rota padrão e rota estática para todas as interfaces
config	Uma grande quantidade de variáveis de rede menos utilizadas comumente

Para uma interface de rede real (isto é, não a loopback), o nome do arquivo tem a forma estendida **ifcfg-***interface***-id-***MAC*, onde *MAC* é o endereço de hardware da interface de rede. (**ifcfg-eth-id-00:0c:29:d4:ea:26** é um exemplo.)

Além de especificar o endereço IP, gateway e informações de transmissão para uma interface, os arquivos **ifcfg-*** podem ajustar muitos outros parâmetros de rede; o arquivo **ifcfg.template** é uma visão geral bem comentada dos possíveis parâmetros.

A ferramenta YaST do SUSE inclui uma interface fácil de usar pronta para configurar a rede. Ela funciona bem e a recomendamos para gerenciar os arquivos **ifcfg-*** sempre que possível. Se precisar configurar a rede manualmente, eis um modelo simples com nossos comentários:

```
BOOTPROTO='static'                # Static está implícito, mas não custa informar explicitamente
IPADDR='192.168.1.4/24'           # O /24 define as váriaveis rede e mascara
NAME='AMD PCnet - Fast 79C971'    # Usado para parar e inicar a interface
STARTMODE='auto'                  # Inicia automaticamente no boot
USERCONTROL='no'                  # GUI Desativa o controle na interface kinternet/cinternet
```

Informações globais de roteamento estático para um sistema SUSE (incluindo a rota padrão) são armazenadas no arquivo **routes**. Cada linha nesse arquivo é como um comando **route** com o nome do comando omitido: destino, gateway, máscara de rede, interface e parâmetros extras opcionais a ser armazenados na tabela de roteamento para uso pelos daemons de roteamento. Para o host configurado anteriormente, que tem apenas uma rota padrão, o arquivo **routes** contém a linha

```
default 192.168.1.1 - -
```

Rotas únicas para interfaces específicas são mantidas nos arquivos **ifroute-***interface*, onde a nomenclatura do componente *interface* é a mesma dos arquivos **ifcfg-***. O conteúdo tem o mesmo formato do arquivo **routes**.

Configuração de rede para o Debian e o Ubuntu

Como mostrado na Tabela 12.14, o Debian e o Ubuntu configuram a rede principalmente em **/etc/hostname** e **/etc/network/interfaces**, com uma pequena ajuda do arquivo **/etc/network/options**.

O hostname é configurado em **/etc/hostname**. O nome nesse arquivo deve ser completamente qualificado; seu valor é utilizado em vários contextos, alguns dos quais exigem isso. Porém, a instalação Debian padrão deixa um nome curto aí.

Tabela 12.14 Os arquivos de configuração de rede do Debian e do Ubuntu

Arquivo	O que é configurado aí
/etc/hostname	Nome de host
/etc/network/interfaces	Endereço IP, máscara de rede, rota padrão
/etc/network/options	Opções de rede de baixo nível (encaminhamento IP etc.)

O endereço IP, a máscara de rede e o gateway padrão são configurados no arquivo **/etc/network/interfaces**. Uma linha que inicia com a palavra-chave `iface` introduz cada interface. A linha `iface` pode ser seguida por linhas recuadas que especificam parâmetros adicionais. Por exemplo:

```
iface lo inet loopback
iface eth0 inet static
    address 192.168.1.102
    netmask 255.255.255.0
    gateway 192.168.1.254
```

Os comandos **ifup** e **ifdown** lêem esse arquivo e ativam ou desativam as interfaces chamando comandos de nível mais baixo (como **ifconfig**) com os parâmetros apropriados. A palavra-chave `inet` na linha `iface` é a família de endereço; essa sempre será `inet`. A palavra-chave `static` é chamada de "método" e especifica que o endereço IP e a máscara de rede para eth0 são atribuídos diretamente. As linhas `address` e `netmask` são necessárias para configurações estáticas; as primeiras versões do kernel do Linux também exigiam que fosse especificado o endereço de rede, mas agora o kernel é mais inteligente e capaz de descobrir o endereço de rede pelo endereço IP e pela máscara de rede. A linha `gateway` especifica o endereço do gateway padrão e é usada para instalar uma rota padrão.

O arquivo **options** permite configurar variáveis de rede em tempo de inicialização. Como padrão, o Debian deixa o encaminhamento IP desativado, a proteção contra falsificação (*spoofing*) ativada e a proteção contra syn cookies desativada.

12.9 DHCP: O DYNAMIC HOST CONFIGURATION PROTOCOL

O DHCP é definido nas RFCs 2131 e 2132.

Os hosts Linux exigiram historicamente que a configuração manual fosse adicionada a uma rede. Ao conectar um Mac ou PC em uma rede, ele simplesmente funciona. Por que o Linux não pode fazer o mesmo? O DHCP (Dynamic Host Configuration Protocol) torna essa expectativa razoável vários passos mais próximo da realidade.

O protocolo ativa um cliente DHCP para locar uma variedade de parâmetros de rede e administrativos de um servidor central que está autorizado a distribuí-los. O paradigma da locação é particularmente conveniente para PCs que são desativados quando não estão em uso e para ISPs que possuem clientes discados intermitentes.

Entre os parâmetros locáveis temos:

- Endereços IP e máscaras de rede
- Gateways (rotas padrão)
- Servidores de nomes DNS
- Hosts para syslog
- Servidores WINS, servidores X, servidores proxy, servidores NTP
- Servidores TFTP (para carregar uma imagem de inicialização)

e muitos outros (consulte a RFC2132). Entretanto, o emprego no mundo real de parâmetros mais exóticos é raro. Em muitos casos, um servidor DHCP fornece somente parâmetros de rede básicos como endereços IP, máscaras de rede, gateways padrão e servidores de arquivos de nomes.

Clientes têm de se reportar periodicamente ao servidor DHCP para renovar as suas locações. Se uma locação não for renovada, ela acaba expirando. O servidor DHCP então fica livre para atribuir o endereço (ou seja lá o que estiver sendo locado) a um cliente diferente. O período de locação é configurável, mas normalmente é bem longo (horas ou dias).

O DHCP é capaz de poupar um bom tempo e sofrimento de um administrador de sistemas outrora desafortunado. Assim que o servidor estiver ativo e em funcionamento, os clientes poderão utilizá-lo para obter automaticamente suas configurações em tempo de inicialização. Sem frescuras, sem confusão.

Software DHCP

As distribuições Linux são historicamente distribuídas com vários servidores e clientes DHCP diferentes . Hoje em dia, todas elas foram mais ou menos padronizadas na implementação de referência do Internet Systems Consortium, ISC. O servidor ISC também entende o protocolo BOOTP, que é semelhante em conceito ao DHCP, mas mais antigo e menos sofisticado. O software de cliente DHCP é instalado por padrão em todas as distribuições modernas, mas às vezes você precisa instalar pacotes adicionais a fim de que o servidor ISC e o agente de retransmissão sejam ativados e colocados em execução.

Os clientes DHCP iniciam uma conversa com o servidor DHCP utilizando o endereço de broadcast genérico todo preenchido somente com 1s — os clientes não conhecem ainda suas máscaras de sub-rede e, conseqüentemente, não podem utilizar o endereço de broadcast de sub-rede.

O servidor DHCP do ISC fala o protocolo de atualização dinâmica de DNS. O servidor não só dá ao seu host o seu endereço IP e outros parâmetros de rede, como também pode atualizar o banco de dados DNS com o mapeamento nome de host-endereço IP correto. Consulte a página 312 para informações adicionais sobre atualizações DNS dinâmicas.

Nas seções a seguir, discutiremos brevemente o protocolo DHCP, explicaremos como configurar o servidor ISC que o implementa e, a seguir, discutiremos algumas questões de configuração de clientes.

Funcionamento do DHCP

O DHCP é uma extensão do BOOTP compatível com versões anteriores, um protocolo que era originalmente destinado a permitir que estações de trabalho UNIX sem disco fossem capazes de fazer a inicialização. BOOTP fornece aos clientes seu endereço IP, máscara de rede, gateway padrão e informação sobre a inicialização do TFTP. O DHCP generaliza os parâmetros que podem ser fornecidos e acrescenta o conceito de "locação".

Um cliente DHCP inicia sua interação com um servidor DHCP transmitindo a todos uma mensagem de "Socorro! Quem sou eu?". Se houver um servidor DHCP na rede local, ele negocia com o cliente para locar-lhe um endereço IP e fornecer-lhe outros parâmetros de rede (máscara de rede, informações de servidor de nomes e gateway padrão). Se não houver nenhum servidor DHCP na rede local, os servidores em sub-redes diferentes também poderão receber a mensagem inicial de um proxy chamado de "agente relay".

Quando a locação do cliente estiver próxima da expiração, ele deve renovar a locação enviando uma outra mensagem REQUEST se quiser continuar utilizando o endereço alocado. Se um cliente decidir terminar sua locação, ele faz isso com uma mensagem RELEASE.

Quando o tempo de locação do cliente estiver pela metade, ele renovará a locação. O servidor é obrigado a controlar os endereços que ele manipulou e essas informações devem persistir entre reinicializações. Os clientes também devem manter seus estados de locações entre reinicializações, embora muitos não o façam. O objetivo é maximizar a estabilidade na configuração da rede.

A propósito, o DHCP normalmente não é utilizado para configurar interfaces PPP discadas. Em geral,próprio PPPCP (PPP Control Protocol) do PPP desempenha esse papel.

Servidor DHCP do ISC

Para configurar o servidor DHCP, **dhcpd**, edite o arquivo **dhcpd.conf** modelo do diretório **server** e instale-o em **/etc/dhcpd.conf**.[20] Você também deve criar um arquivo vazio de banco de dados de locação chamado **/var/db/dhcp.leases/**; utilize o comando **touch**. Certifique-se de que o **dhcpd** seja capaz de efetuar gravações nesse arquivo. Para configurar o arquivo **dhcpd.conf**, você precisará das seguintes informações:

- As sub-redes para as quais o **dhcpd** deve dirigir endereços IP e os intervalos de endereços a ser doados
- As durações inicial e máxima da locação, medidas em segundos
- Configurações para clientes BOOTP se você tiver algum (eles têm endereços IP estáticos e também devem ter listados seus endereços de hardware no nível MAC)
- Quaisquer outras opções que o servidor precisar passar para clientes DHCP: máscara de rede, rota padrão, domínio DNS, servidores de nomes etc.

A página man do **dhcpd** revisa o processo de configuração. A página man do **dhcpd.conf** abrange a sintaxe exata do arquivo de configuração. Ambas se encontram no subdiretório **server** da distribuição. Algumas distribuições incluem um modelo de arquivo **dhcpd.conf** no diretório **/etc**; modifique-o para ficar de acordo com a configuração de rede de sua instalação local.

---

20. Tome cuidado: o formato do arquivo **dhcpd.conf** é um pouco frágil. O simples fato de omitir um ponto-e-vírgula já é o bastante para você receber uma mensagem de erro vaga e inútil.

dhcpd deve ser iniciado automaticamente em tempo de inicialização. Talvez você ache interessante fazer a inicialização do daemon condicionada à existência de **/etc/dhcpd.conf**.

Eis um exemplo de arquivo **dhcpd.conf** de um computador Linux com duas interfaces: uma interna e uma que se conecta à Internet. Essa máquina realiza conversões NAT para a rede interna e também faz a locação de um intervalo de 10 endereços IP nesta rede. O arquivo **dhcpd.conf** contém uma entrada fictícia para a interface externa (obrigatória) e uma entrada `host` para uma determinada máquina que precisa de um endereço fixo.

```
# dhcpd.conf
#
# global options
option domain-name "synack.net";
option domain-name-servers gw.synack.net;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.51 192.168.1.60;
    option broadcast-address 192.168.1.255;
    option routers gw.synack.net;
}

subnet 209.180.251.0 netmask 255.255.255.0 {
}

host gandalf {
   hardware ethernet 08:00:07:12:34:56;
   fixed-address gandalf.synack.net;
}
```

Consulte o Capítulo 15 para informações adicionais sobre DNS.

Endereços atribuídos pelo DHCP podem eventualmente entrar em conflito com o conteúdo do banco de dados DNS. Os sites normalmente atribuem um nome genérico a cada um dos endereços dinamicamente locados (por exemplo, dhcp1.synack.net) e permitem que os nomes de máquinas particulares "flutuem" com os seus endereços IP. Se estiver utilizando uma versão recente do BIND que ofereça suporte a atualizações dinâmicas, você também poderá configurar **dhcpd** para atualizar o banco de dados DNS à medida que ele manipula endereços. A solução de atualização dinâmica é mais complicada, mas tem a vantagem de preservar o nome de host de cada máquina.

dhcpd registra cada transação de locação no arquivo **dhcp.leases**. Ele também faz periodicamente um backup desse arquivo renomeando-o para **dhcpd.leases~** e recriando o arquivo **dhcp.leases** a partir do seu banco de dados na memória. Se **dhcpd** travasse durante essa operação, você poderia acabar apenas com um arquivo **dhcp.leases~**. Nesse caso, **dhcpd** se recusará a iniciar e você terá de renomear o arquivo antes de reiniciá-lo. *Não* crie simplesmente um arquivo **dhcp.leases** vazio. Caso contrário, o caos se instalará, já que os clientes acabarão ficando com endereços duplicados.

O cliente DHCP realmente não requer configuração. Ele armazena arquivos para cada conexão no diretório **/var/lib/dhcp** ou **/var/lib/dhclient**. Os arquivos recebem nomes conforme as interfaces que eles descrevem. Por exemplo, **dhclient-eth0.leases** conteria todos os parâmetros de rede que **dhclient** configurou para a interface eth0.

12.10 RECONFIGURAÇÃO DINÂMICA E AJUSTE

O Linux tem uma maneira particularmente própria de ajustar o kernel e os parâmetros de rede. Em vez de fornecer um arquivo de configuração normal que é lido para determinar os valores apropriados, o Linux coloca uma representação de cada variável que pode ser ajustada no sistema de arquivos virtual/**proc**. As variáveis de rede se encontram em **/proc/sys/net/ipv4**:

```
$ cd /proc/sys/net/ipv4; ls -F
conf/                           ip_local_port_range   tcp_mem
icmp_echo_ignore_all            ip_nonlocal_bind      tcp_moderate_rcvbuf
icmp_echo_ignore_broadcasts     ip_no_pmtu_disc       tcp_no_metrics_save
icmp_errors_use_inbound_ifaddr  neigh/                tcp_orphan_retries
icmp_ignore_bogus_error_responses route/              tcp_reordering
icmp_ratelimit                  tcp_abc               tcp_retrans_collapse
icmp_ratemask                   tcp_abort_on_overflow tcp_retries1
igmp_max_memberships            tcp_adv_win_scale     tcp_retries2
```

```
igmp_max_msf                    tcp_app_win           tcp_rfc1337
inet_peer_gc_maxtime            tcp_congestion_control tcp_rmem
inet_peer_gc_mintime            tcp_dma_copybreak     tcp_sack
inet_peer_maxttl                tcp_dsack             tcp_stdurg
inet_peer_minttl                tcp_ecn               tcp_synack_retries
inet_peer_threshold             tcp_fack              tcp_syncookies
ip_autoconfig                   tcp_fin_timeout       tcp_syn_retries
ip_default_ttl                  tcp_frto              tcp_timestamps
ip_dynaddr                      tcp_keepalive_intvl   tcp_tso_win_divisor
ip_forward                      tcp_keepalive_probes  tcp_tw_recycle
ipfrag_high_thresh              tcp_keepalive_time    tcp_tw_reuse
ipfrag_low_thresh               tcp_low_latency       tcp_window_scaling
ipfrag_max_dist                 tcp_max_orphans       tcp_wmem
ipfrag_secret_interval          tcp_max_syn_backlog
ipfrag_time                     tcp_max_tw_buckets
```

Muitas dessas variáveis com **rate** e **max** em seus nomes são utilizadas para impedir ataques de recusas de serviços (DoS). O subdiretório **conf** contém as variáveis que são configuradas por interface individual. Ele contém subdiretórios **all** e **default** e um subdiretório para cada interface (inclusive a loopback). Cada subdiretório contém o mesmo conjunto de arquivos.

```
$ cd conf/default; ls -F
accept_redirects          bootp_relay           log_martiansrp_filter
accept_source_route       disable_policy        mc_forwardingsecure_redirects
arp_announce              disable_xfrm          medium_idsend_redirects
arp_filter                force_igmp_version    promote_secondariesshared_media
arp_ignore                forwarding            proxy_arptag
```

Se você alterar algo no subdiretório **all**, suas alterações se aplicarão a todas as interfaces. Se você modificar a mesma variável no, digamos, subdiretório **eth0**, somente essa interface é afetada. O subdiretório **defaults** contém os valores padrão conforme sai de fábrica.

O diretório **neigh** também contém um subdiretório para cada interface. Os arquivos em cada subdiretório controlam o gerenciamento de tabelas ARP e a descoberta de vizinhos IPv6 para essa interface. Eis a lista de variáveis; aquelas iniciadas por **gc** (de garbage collection, ou seja, coleta de lixo) determinam como as entradas de tabelas ARP atingem o seu tempo de expiração e como são descartadas.

```
$ cd neigh/default; ls -F
anycast_delay             gc_interval    locktimeretrans_time_ms
app_solicit               gc_stale_time  mcast_solicitucast_solicit
base_reachable_time       gc_thresh1     proxy_delayunres_qlen
base_reachable_time_ms    gc_thresh2     proxy_qlen
delay_first_probe_time    gc_thresh3     retrans_time
```

Para observar o valor de uma variável, utilize **cat**; para configurá-la, utilize **echo** redirecionado para o nome de arquivo apropriado. Por exemplo, o comando:

```
$ cat icmp_echo_ignore_broadcasts
0
```

mostra que o valor dessa variável é 0, o que significa que os pings de broadcast não são ignorados. Para configurá-la como 1 (e evitar ser presa em ataques de negação de serviço), execute

```
$ sudo sh -c "echo 1 > icmp_echo_ignore_broadcasts"[21]
```

no diretório **/proc/sys/net**. Normalmente você estará conectado na mesma rede em que está ajustando essas variáveis; portanto, tome cuidado! Você pode bagunçar as coisas de tal maneira a exigir uma reinicialização no console para poder se recuperar, o que poderia ser inconveniente se, por acaso, o sistema estivesse em Point Barrow, Alasca, e no pico do inverno. Faça ajustes finos e testes dessas variáveis em seu sistema desktop antes mesmo de pensar em mexer em uma máquina de produção.

---

21. Se você tentar esse comando na forma **sudo echo 1 > icmp_echo_ignore_broadcasts**, você receberá uma mensagem "permission denied"— seu shell tenta abrir um arquivo de saída antes que ele inicie o **sudo**. Você quer que o **sudo** aplique para ambos o comando **echo** e o redirecionamento. Você deve criar um subshell root no qual executar o comando inteiro.

Para alterar um desses parâmetros permanentemente (ou mais exatamente, redefini-los toda vez que o sistema é inicializado), adicione as variáveis apropriadas a **/etc/sysctl.conf**, que é lido pelo comando **sysctl** em tempo de inicialização. O formato do arquivo **sysctl.conf** é *variável=valor* em vez de **echo value > variable**, uma vez que você executaria a partir do shell para alterar a variável manualmente. Os nomes de variáveis são nomes de caminhos relativos a **/proc/sys**; se preferir, você também pode utilizar pontos em vez de barras. Por exemplo, qualquer uma das linhas abaixo

```
net.ipv4.ip_forward=0
net/ipv4/ip_forward=0
```

no arquivo **/etc/sysctl.conf** faria com que o encaminhamento IP fosse desativado (para esse host).

O documento **/usr/src/linux/Documentation/proc.txt**, escrito pelo pessoal do SUSE, é uma boa introdução sobre ajustes de kernels com **/proc**.[22] Ele informa qual o verdadeiro significado das variáveis e, algumas vezes, fornece sugestões para valores. O arquivo **proc.txt** está um pouco obsoleto — as pessoas que escrevem código Linux parecem escrever mais rápido que aquelas que o documentam.

12.11 QUESTÕES DE SEGURANÇA

Trataremos do tópico segurança em um capítulo à parte (Capítulo 20), mas várias questões de segurança relevantes a redes IP merecem uma discussão aqui. Nesta seção, examinaremos brevemente alguns poucos recursos de rede que adquiriram má fama por provocarem problemas de segurança e recomendações para minimizar seu impacto. Os detalhes do comportamento padrão das distribuições Linux vistas neste livro em relação a esses problemas (e métodos apropriados para modificá-los) são vistos posteriormente nesta seção.

Encaminhamento IP

Uma máquina Linux que tem o encaminhamento IP ativado pode atuar como um roteador. A menos que seu sistema tenha várias interfaces de rede e realmente deva atuar como um roteador, é recomendável inibir esse recurso. Hosts que encaminham pacotes podem, algumas vezes, ser coagidos a comprometer a segurança do sistema fazendo com que pacotes externos pareçam ter sido provenientes de dentro da rede. Esse subterfúgio pode ajudar pacotes nocivos a escapar de scanners de rede e de filtros de pacotes.

Redirecionamentos ICMP

Redirecionamentos ICMP podem ser utilizados maliciosamente para redirecionar tráfego e desorganizar suas tabelas de roteamento. A maioria dos sistemas operacionais os ouve e segue suas instruções como padrão. Seria ruim se todo o seu tráfego de informações fosse redirecionado para a rede de um concorrente por algumas horas, especialmente quando backups estão sendo feitos! Recomendamos que você configure seus roteadores (e hosts que atuam como roteadores) para ignorarem e talvez registrarem em log redirecionamentos ICMP.

Roteamento de origem

O mecanismo de roteamento de origem do IP permite especificar explicitamente uma série de gateways pelos quais um pacote deve transitar no trajeto a seu destino. O roteamento de origem prevalece sobre o algoritmo de roteamento de próximo salto que é normalmente executado em cada gateway para determinar como um pacote deve ser encaminhado.

O roteamento de origem fazia parte da especificação IP original; ele era destinado a, basicamente, facilitar testes. Ele pode criar problemas de segurança, pois os pacotes normalmente são filtrados conforme suas origens. Se alguém for capaz de direcionar engenhosamente um pacote para fazer com que ele pareça ter se originado dentro de sua rede e não da Internet, ele poderia passar pelo seu firewall. Recomendamos que você não aceite nem encaminhe pacotes roteados pela origem.

Pings de broadcast e outras formas de broadcast direcionado

Pacotes ping destinados ao endereço de broadcast de uma rede (em vez de um endereço de host em particular) normalmente serão entregues a todos os hosts da rede. Tais pacotes têm sido utilizados em ataques de negação de serviços; por exemplo, os chamados ataques smurf. A maioria dos hosts tem uma maneira de desativar pings de broadcast — isto é, o host pode ser configurado para não responder nem encaminhar pings de broadcast. O seu roteador Internet também pode filtrar pings de broadcast antes de chegarem à sua rede interna. É uma boa idéia utilizar medidas de segurança tanto no nível de host quanto no de firewall, se você puder.

---

22. Para disponibilizar uma cópia do **proc.txt**, você precisa instalar o código-fonte do kernel.

Os pings de broadcast são uma espécie de "broadcast direcionado" em que eles são pacotes enviados ao endereço de broadcast de uma rede distante. A manipulação padrão desses pacotes tem mudado gradualmente. Por exemplo, o IOS da Cisco até a versão 11.x encaminhava pacotes de broadcast direcionados por padrão, ao passo que a partir da versão 12.0 isso não ocorre mais. Normalmente é possível convencer sua estrutura TCP/IP a ignorar pacotes de broadcast provenientes de tão longe, mas, já que esse comportamento pode ser configurado em cada interface, essa pode ser uma tarefa incomum em uma instalação grande.

Falsificação de IP

O endereço de origem de um pacote IP normalmente é preenchido pela implementação TCP/IP do kernel e o endereço IP do host a partir do qual o pacote foi enviado. Entretanto, se o software que cria o pacote utilizar um soquete bruto, ele poderá preencher qualquer endereço de origem que ele quiser. Isso é chamado de falsificação de IP e normalmente está associado a algum tipo de comportamento malicioso da rede. A máquina identificada pelo endereço IP de origem falsificado (se for um endereço real) em geral é a vítima no esquema. Pacotes retornados e de erros podem corromper ou inundar as conexões de rede da vítima.

Você deve impedir a falsificação de IP em seu roteador de borda bloqueando pacotes de saída cujos endereços de origem não se encontram dentro do seu espaço de endereços. Essa precaução é especialmente importante se sua instalação for uma universidade em que os alunos gostam de fazer experiências e freqüentemente têm um sentimento de retaliação em relação àqueles considerados "idiotas" nos seus canais favoritos de bate-papo.

Ao mesmo tempo, se você estiver utilizando internamente espaço de endereços privados, poderá fazer uma filtragem para capturar quaisquer endereços internos que estejam escapando para a Internet. Esses pacotes nunca podem ser respondidos (graças à falta de uma rota de backbone) e normalmente indicam que sua instalação possui algum erro de configuração interno.

Com os firewalls baseados no Linux, descritos na próxima seção, você pode implementar essa filtragem individualmente por host. Entretanto, a maioria das instalações prefere implementar esse tipo de filtragem em seus roteadores de borda e não em cada host. Essa é a abordagem que também recomendamos. Descrevemos os firewalls baseados em hosts apenas por uma questão de completude e para situações especiais.

Você também tem de se proteger contra um hacker forjando o endereço de origem em pacotes externos para lograr o seu firewall e fazê-lo pensar que se originaram em sua rede interna. O parâmetro de kernel **rp_filter** (configurável no diretório **/proc/sys/net/ipv4/conf/***ifname*) pode ajudá-lo a detectar tais pacotes; **rp** significa reverse path, ou seja, caminho inverso. Se você configurar essa variável como 1, o kernel descartará pacotes que chegam a uma interface diferente daquela na qual eles deveriam partir se o endereço de origem fosse o de destino. Esse comportamento fica ativado por padrão.

Se sua instalação tiver várias conexões com a Internet, pode ser perfeitamente razoável que rotas de chegada e de saída sejam diferentes. Nessa situação, configure **rp_filter** como 0 para que seu protocolo de roteamento funcione adequadamente. Se a instalação tiver apenas uma saída para a Internet, então configurar **rp_filter** como 1 é normalmente seguro e apropriado.

Firewalls baseados em hosts

O Linux inclui software para filtragem de pacotes (também conhecido como firewall). Apesar de descrevermos esse software posteriormente neste capítulo (página 223) e também no Capítulo 20, *Segurança* (página 484), realmente não recomendamos o uso de uma estação de trabalho como um firewall. A segurança do Linux (especialmente aquela fornecida pelos nossos amigáveis fornecedores) é fraca e a segurança do NT é pior ainda. Sugerimos que você adquira uma solução de hardware dedicada para utilizar como firewall. Até mesmo uma solução de software sofisticada como o produto Firewall-1 da Checkpoint (que é executada em um host Solaris) não é tão boa quanto a de um hardware dedicado como o equipamento PIX da Cisco — e seu preço é praticamente o mesmo!

Uma discussão mais aprofundada sobre questões relacionadas a firewalls está na página 484.

Redes privadas virtuais (VPNs)

Muitas empresas que têm escritórios em várias partes do mundo gostariam que todas essas localidades se conectassem a uma grande rede privada. Infelizmente, o custo de locar uma linha de dados transoceânica ou até mesmo transcontinental pode se tornar proibitivo. Essas empresas podem, na verdade, utilizar a Internet como se fosse uma linha de dados privada estabelecendo uma série de "túneis" criptografados e seguros entre suas diversas localidades. Uma rede "privada" que inclui esses túneis é conhecida como rede virtual privada ou VPN (Virtual Private Network).

Consulte a página 489 para informações adicionais sobre IPsec.

Algumas VPNs utilizam o protocolo IPsec, que foi padronizado pelo IETF em 1998. Outras utilizam soluções proprietárias que normalmente não interoperam umas com as outras. Se precisar de funcionalidade VPN, sugerimos que você examine produtos como o roteador 3660 da Cisco ou o Watchguard Firebox, ambos capazes de fazer tunelamento e criptografia. O dispositivo Watchguard utiliza PPP para uma porta serial para fins de gerenciamento. Um administrador de sistemas pode discar para a máquina para configurá-la ou para acessar a VPN para testes.

Para uma solução VPN para orçamento de baixo custo, veja o exemplo da página 229 que utiliza PPP sobre uma conexão **ssh** para implementar uma rede privada virtual.

Variáveis do kernel relacionadas com segurança

A Tabela 12.15 ilustra o comportamento padrão do Linux em relação a diversas questões sensíveis de rede. Para uma breve descrição das implicações desses comportamentos, consulte a página 220. Recomendamos que você altere os valores dessas variáveis para que não responda a pings de broadcast, não dê ouvidos a redirecionamentos de rotas e não aceite pacotes roteados pela origem.

Tabela 12.15 Comportamentos de rede relacionados com segurança no Linux.

Recurso	Host	Gateway	Arquivo de controle (em /proc/sys/net)
Encaminhamento IP	desativado	ativado	**ipv4/ip_forward** para todo o sistema **ipv4/conf/***interface***/forwarding** por interface[a]
Redirecionamentos ICMP	obedece	ignora	**ipv4/conf/***interface***/accept_redirects**
Roteamento de origem	ignora	obedece	**ipv4/conf/***interface***/accept_source_route**
Ping de broadcast	responde	responde	**ipv4/icmp_echo_ignore_broadcasts**

a. *interface* pode ser um nome de interface específico ou **all**.

12.12 LINUX NAT

O Linux tradicionalmente implementa apenas uma forma limitada do NAT (Network Address Translation), que é mais apropriadamente chamada de PAT (Port Address Translation). Em vez de utilizar um intervalo de endereços IP como uma verdadeira implementação NAT faria, o PAT multiplexa todas as conexões em um único endereço. Para aumentar ainda mais a confusão, documentos do Linux freqüentemente não se referem ao recurso como NAT ou PAT, mas como "mascaramento de IP". Os detalhes e diferenças não são de muita importância prática, portanto nos referiremos à implementação Linux como NAT para fins de consistência.

iptables implementa não só o NAT, mas também a filtragem de pacotes. Nas primeiras versões do Linux isso era um pouco confuso, mas **iptables** faz uma distinção muito mais clara entre os recursos NAT e de filtragem.

Os recursos da filtragem de pacotes são abrangidos em mais detalhes no Capítulo 20, *Segurança*, que começa na página 484. Se utilizar o NAT para permitir que hosts locais acessem a Internet, você *deverá* utilizar um complemento completo de filtros de firewall ao executar o NAT. O fato de o NAT "não ser realmente um roteamento de IP" não torna um gateway NAT do Linux mais seguro que um roteador do Linux. Para simplificar, só descrevemos a configuração NAT real aqui; mas isso é apenas uma pequena parte de uma configuração completa.

Para que o NAT funcione, você precisa ativar o encaminhamento IP no kernel configurando a variável de kernel **/proc/sys/net/ipv4/ip_forward** como 1. Além disso , você deve inserir os módulos de kernel apropriados:

```
$ sudo /sbin/modprobe iptable_nat
$ sudo /sbin/modprobe ip_conntrack
$ sudo /sbin/modprobe ip_conntrack_ftp
```

O comando **iptables** para rotear pacotes utilizando o NAT é da forma

```
$ sudo iptables -t nat -A POSTROUTING -o eth1 -j SNAT --to 63.173.189.1
```

Neste exemplo, eth0 é a interface conectada à Internet e seu endereço IP é o que aparece como o argumento para **--to**. A interface eth1 está a conectada à rede interna.

Para hosts Internet, parece que todos os pacotes dos hosts na rede interna têm o endereço IP da eth0. O host realizando o NAT recebe pacotes entrantes, pesquisa seus destinos verdadeiros, regrava-os com o endereço IP apropriado da rede interna e envia-os.

12.13 PPP: O POINT-TO-POINT PROTOCOL

PPP, o protocolo ponto a ponto (Point-to-Point Protocol), tem a distinção de ser utilizado tanto nos links Internet mais lentos quanto nos mais velozes. Em sua forma síncrona, ele é o protocolo de encapsulamento utilizado em circuitos de alta velocidade que possuem roteadores robustos em ambas as pontas. Em sua forma assíncrona, ele é um protocolo de encapsulamento de linhas seriais que especifica como os pacotes IP devem ser codificados para transmissão em uma linha serial lenta (e normalmente não confiável). As linhas seriais apenas transmitem fluxos de bits e não têm noção alguma do ponto de início ou final de um pacote. O driver de dispositivo PPP cuida da codificação e decodificação de pacotes na linha serial; ele acrescenta um cabeçalho em nível de link e marcadores que separam os pacotes.

O PPP é algumas vezes utilizado com as tecnologias domésticas mais modernas, como DSL e modems a cabo, mas esse fato é normalmente omitido de um administrador de sistemas. O encapsulamento é tipicamente executado pelo dispositivo de interface, e o tráfego faz uma ponte com Ethernet. Você só vê uma conexão Ethernet.

Projetado por comitê, o PPP é o protocolo de encapsulamento "tudo o que se possa imaginar *e* mais um pouco". Ele se inspirou nos protocolos SLIP (Serial Line IP) e CSLIP (Compressed SLIP) desenvolvidos por, respectivamente, Rick Adams e Van Jacobson. O PPP difere desses sistemas pelo fato de permitir a transmissão de vários protocolos em um único link. Ele é especificado na RFC1331.

Resolvendo problemas de desempenho no PPP

O PPP fornece toda a funcionalidade do padrão Ethernet, mas a velocidades *muito* menores. As LANs empresariais comuns operam até 10 MB/s ou 100 MB/s — isto é, de 10.000 KB/s a 100.000 KB/s. Uma conexão discada opera a aproximadamente de 28 KB/s a 56 KB/s. Para relacionar melhor esses números, leva aproximadamente 5 minutos para transferirmos um arquivo de um megabyte em uma rede PPP discada. A velocidade é aceitável para e-mail ou navegação Web com imagens desativadas, mas sites Web muito enfeitados o levarão à loucura. Para melhorar o desempenho interativo, você poderá configurar o MTU do link ponto a ponto com um valor bem baixo. Ele normalmente adota como padrão 512 bytes; tente 128 se estiver realizando muitas tarefas interativas. Se estiver utilizando PPP na Ethernet, utilize **tcpdump** para ver os tamanhos dos pacotes trafegando pela rede e configure o MTU de acordo. O MTU de redes Ethernet é de 1500, mas o encapsulamento PPP torna valores ligeiramente menores mais eficientes. Por exemplo, **pppoe** sugere 1412 bytes para hosts por trás da conexão PPP e 1492 no link PPP. Certamente você não quer que cada pacote seja fragmentado porque você configurou o seu MTU padrão com um valor muito alto.

Consulte o Capítulo 16 para informações adicionais sobre NFS.

Executar NFS em um link PPP pode ser terrivelmente lento. Você deve considerá-lo somente se puder executar NFS via TCP em vez de UDP.

O protocolo X Window System utiliza TCP, então é possível executar aplicativos X sobre um link PPP. Programas como **xterm** funcionam bem, contudo evite aplicativos que utilizem fontes extravagantes ou imagens com mapas de bits.

Conectando-se a uma rede com o PPP

Para conectar um host a uma rede com o PPP, você deve atender a três pré-requisitos:

- O kernel do host deve ser capaz de enviar pacotes IP por uma linha serial conforme especificado pelo padrão de protocolo PPP
- Você deve ter um programa em nível de usuário que permita estabelecer e manter conexões PPP
- Um host no outro lado da linha serial deve entender o protocolo que você está utilizando

Fazendo um host falar PPP

Consulte a página 209 para informações adicionais sobre o ifconfig.

Para que uma conexão PPP seja estabelecida, o host precisa ser capaz de enviar e receber pacotes PPP. Em sistemas Linux, o PPP é um módulo de kernel carregável que coloca pacotes de rede na fila de saída de dispositivo serial e vice-versa. Esse módulo normalmente simula ser mais uma interface de rede, desse modo, pode ser manipulado com ferramentas de configuração padrão como **ifconfig**.

Controlando links PPP

A seqüência exata dos eventos envolvidos no estabelecimento de uma conexão PPP depende do sistema operacional e do tipo de servidor para o qual você está discando. As conexões podem ser iniciadas tanto manual quanto dinamicamente.

Para estabelecer uma conexão PPP manualmente, você executa um comando que disca um modem, faz o login em um host remoto e inicia o mecanismo de protocolo PPP remoto. Se esse procedimento for bem-sucedido, a porta serial é então configurada como uma interface de rede. Essa opção normalmente deixa o link ativo por um tempo longo, o que a torna mais indicada para uma linha telefônica dedicada à conectividade IP.

Em uma configuração dinâmica, um daemon observa suas interfaces de "rede" seriais para ver quando o tráfego foi enfileirado para elas. Quando alguém tenta enviar um pacote, o daemon disca um modem automaticamente para estabelecer a conexão, transmite o pacote e, se a linha retorna ao estado de ocioso, desconecta a linha após um período de tempo razoável. A discagem dinâmica é normalmente utilizada se uma linha telefônica transporta tanto dados quanto voz ou se a conexão envolve tarifas de tempo de conexão e de chamadas interurbanas.

O software para implementar esses dois esquemas de conexão é incluso na maioria das versões do PPP.

Atribuindo um endereço

Consulte a página 209 para informações adicionais sobre a atribuição de endereços IP.

Da mesma forma que você deve atribuir um endereço IP a um novo host em uma rede Ethernet, você precisa atribuir um endereço IP a cada interface PPP. Existem várias maneiras de atribuir endereços a esses links (inclusive não atribuir nenhum endereço). Discutimos somente o método mais simples aqui.

Imagine um link PPP como uma rede de sua propriedade. Isto é, uma rede formada exatamente por dois hosts, freqüentemente denominada de rede ponto a ponto. Você precisa atribuir um número de rede ao link exatamente como faria para atribuir um número de rede a um novo segmento em uma rede Ethernet, utilizando seja lá quais forem as regras em vigor em sua instalação. Você pode escolher dois endereços de host quaisquer dessa rede e atribuir um a cada extremidade do link. Siga também outros costumes locais, por exemplo, padrões de sub-redes. Cada host torna-se então um "gateway" para a rede ponto a ponto no que diz respeito ao restante do mundo. (Na realidade, você normalmente não controla ambas as extremidades do link; seu ISP fornece o endereço IP que você deve utilizar na sua extremidade.)

O DHCP também pode atribuir o endereço de IP à extremidade de um link PPP. Alguns ISPs oferecem serviços domésticos que utilizam o DHCP e serviços comerciais que são mais caros, mas incluem um conjunto de endereços estáticos.

Roteamento

Consulte o Capítulo 13 para informações adicionais sobre roteamento.

Como o PPP requer que o servidor remoto atue como um roteador IP, você precisa se preocupar com o roteamento IP exatamente como faria com um gateway "real", tal como uma máquina que conecta duas redes Ethernet. O objetivo do roteamento é o de direcionar pacotes por meio de gateways de modo que eles possam atingir os seus destinos finais. O roteamento pode ser configurado de diversas maneiras.

Uma conexão host-cliente PPP comum deve ter uma rota padrão que encaminha pacotes ao servidor PPP. Da mesma maneira, o servidor precisa ser conhecido pelos outros hosts em sua rede como o gateway para a máquina na extremidade (folha).

A maioria dos pacotes PPP cuida dessas tarefas de roteamento de modo automático.

Garantindo a segurança

Consulte o Capítulo 20 para mais informações sobre segurança.

Toda vez que você adiciona um host a uma rede surgem problemas de segurança. Como um host conectado via PPP é um membro fidedigno da rede, precisamos tratá-lo como tal: verificar se o sistema não tem contas sem senhas ou com senhas inseguras, se todas as correções de segurança dos diversos fornecedores foram instaladas e assim por diante. Consulte a seção *Questões de segurança* na página 221 para algumas especificidades sobre a segurança de rede. PPP no Linux suporta dois protocolos de autenticação: PAP (Password Authentication Protocol) e o CHAP (Challenge Handshake Authentication Protocol).

Utilizando "chat scripts"

A implementação PPP de linha serial do Linux utiliza um "chat script" para conversar com o modem e também para conectar-se à máquina remota e inicializar um servidor PPP. Um chat script é formado por uma seqüência de strings a enviar e strings a esperar em retorno, com uma forma limitada de instrução condicional que pode expressar conceitos como "espere receber a string 'Login', mas se você não a receber, envie um retorno de linha e aguarde por ela novamente".

A idéia de um chat script originou-se com o sistema armazene-e-encaminhe UUCP de antigamente. Nos anos 80, as máquinas chamavam uma à outra no meio da noite, faziam o login por meio de chat scripts e trocavam arquivos entre si. Apesar da reivindicação popular, o UUCP ainda não está completamente morto: o usuário UUCP é o proprietário do grupo de arquivos de dispositivos seriais no SUSE, e você precisa ser um membro do grupo UUCP para poder utilizar um modem discado para PPP.

A maioria das implementações PPP tem modelos de chat scripts aos quais você pode adaptar ao seu próprio ambiente. Você precisa editar os scripts para configurar parâmetros como o número de telefone a ser chamado e o comando a ser executado após um login bem-sucedido. A maioria dos chat scripts contém uma senha em texto não criptografado; configure as permissões apropriadamente.

Configurando o PPP do Linux

Os modems (juntamente com as impressoras) sempre foram uma pedra no sapato dos administradores de sistemas. E não é nenhuma surpresa, quando o software para configurar uma conexão PPP em um modem aleatório tem mais de 125 opções possíveis — de longe um número excessivo para se avaliar e configurar cuidadosamente.

Todas as nossas distribuições, exceto o Debian, incluem o pacote PPP de Paul Mackerras na instalação padrão. Esse pacote utiliza um daemon chamado **pppd** e mantém a maioria dos seus arquivos de configuração em **/etc/ppp**. Execute o comando **pppd --version** para ver qual versão do pacote PPP foi instalada em sua distribuição em particular. Utilize **apt-get install ppp** para instalar esse pacote no Debian.

Nossos sistemas de referência incluem uma versão do PPP da Roaring Penguin Software que é projetada para uso em redes Ethernet (por exemplo, em uma conexão DSL para um ISP local). Os sistemas de referência também incluem suporte PPP a conexões ISDN. Os arquivos de **configuração** para essas mídias adicionais são colocados com aquelas para PPP via links seriais no diretório **/etc/ppp**. Os nomes dos arquivos normalmente são similares, mas com o acréscimo de um **oe** de "over Ethernet" ou **i** de ISDN. A Tabela 12.16 a seguir mostra as localizações dos comandos relevantes e os arquivos de configuração.

Tabela 12.16 Comandos e arquivos de configuração relacionados ao PPP por sistema

Sistema	Comandos ou arquivos de configuração	Descrição
Todos	/usr/sbin/pppd	Programa daemon PPP
	/usr/sbin/chat	Conversa com o modem
	/usr/sbin/pppstats	Mostra dados estatísticos do link PPP
	/usr/sbin/pppdump	Transforma pacotes PPP em ASCII legível
	/etc/ppp/options	Arquivo de configuração para **pppd**
Debian, Ubuntu	/usr/bin/pon	Inicia uma conexão PPP
	/usr/bin/poff	Encerra uma conexão PPP
	/usr/bin/plog	Mostra o final de **ppp.log**
	/usr/sbin/pppconfig	Configura **pppd**
	/etc/ppp/peers/provider	Opções para **pon** para contatar seu ISP
	/etc/chatscripts/provider	Chat script para **pon** conversar com o ISP
Red Hat (DSL)	/usr/sbin/pppoe	Cliente PPP em rede Ethernet
	/usr/sbin/pppoe-server	Servidor PPP em rede Ethernet
	/usr/sbin/pppoe-sniff	Analisador que depura peculiaridades do provedor
	/usr/sbin/adsl-connect	Script que gerencia links
	/usr/sbin/adsl-setup	Script que configura **pppoe**
	/usr/sbin/adsl-start	Script que aciona o link **pppoe**
	/usr/sbin/adsl-stop	Script que desativa link **pppoe**
	/usr/sbin/adsl-status	Mostra o status do link **pppoe**
	/etc/ppp/pppoe.conf	Arquivo de configuração utilizado por **adsl-***
	/etc/ppp/pppoe-server-options	Arquivo com opções extras para servidor
SUSE (DSL)	/usr/sbin/pppoed	PPP em cliente Ethernet
	/etc/pppoed.conf	Arquivo de configuração para **pppoed**
All (ISDN)	/usr/sbin/ipppd	PPP no daemon ISDN
	/usr/sbin/ipppstats	Mostra estatísticas ISDP PPP
	/etc/ppp/ioptions	Opções para **ipppd**

Consulte a página 586 para mais informações sobre os nomes das portas seriais.

Nos nossos exemplos dos arquivos de configuração, **/dev/modem** é nosso nome para a porta serial que tem um modem anexado a ela. Na verdade, algumas distribuições têm um arquivo **/dev/modem** que é um link para uma das portas seriais do sistema (normalmente **/dev/ttyS0** ou **/dev/ttyS1**), mas essa prática agora está obsoleta. Substitua o arquivo de dispositivo apropriado à sua situação.

Além do software PPP, cada distribuição inclui o programa **wvdial** para discar o número de telefone e estabelecer uma conexão.

Discutimos anteriormente as portas de modem e o software discador; agora examinaremos como configurar **pppd** para utilizá-las. As opções globais são configuradas no arquivo **/etc/ppp/options** e opções para conexões particulares podem ser armazenadas nos diretórios **/etc/ppp/peers** e **/etc/chatscripts** (no Debian e Ubuntu). O Red Hat, o Fedora e o SUSE tendem a colocar chat scripts no diretório **/etc/ppp** com nomes como **chat.***hostRemoto*. Alternativamente, no Red Hat, o arquivo **/etc/sysconfig/network-scripts/ifcfg-***ttyname* pode incluir opções específicas à conexão para uma determinada interface PPP.

Como padrão, **pppd** consulta primeiramente o arquivo **options**, depois o arquivo de inicialização pessoal do usuário, **~/.ppprc**, a seguir o arquivo específico a uma determinada conexão **options.***nomeDeTty* (se existir um) e, finalmente, seus argumentos de linha de comando.

Um truque prático sugerido por Jonathan Corbet, um veterano do Linux, é o de definir mais de uma interface PPP: uma para casa, uma para hotéis enquanto estiver viajando etc. Essa configuração facilita a troca de contextos.

wvdial é mais inteligente do que **chat** e possui um comportamento padrão sensível se parâmetros forem deixados sem especificação. **wvdial** obtém suas informações de configuração de **/etc/wvdial.conf**: detalhes de modems, nome de login,

senha, número de telefone etc. Você pode fornecer informações para vários destinos em um único arquivo de configuração. Utilize o programa **wvdialconf** para descobrir as características de seu modem e crie um arquivo **wvdial.conf** inicial para ele.

Os arquivos de configuração a seguir são extraídos de várias configurações PPP diferentes. O primeiro arquivo, **/etc/ppp/options**, configura opções globais para **pppd**. As opções ativas para cada distribuição conforme saem de fábrica são mostradas a seguir:

/etc/ppp/options no Red Hat e no Fedora:

```
lock
```

/etc/ppp/options no SUSE:

```
noipdefault
noauth
crtscts
lock
modem
asyncmap 0
nodetach
lcp-echo-interval 30
lcp-echo-failure 4
lcp-max-configure 60
lcp-restart 2
idle 600
noipx
file /etc/ppp/filters
```

/etc/ppp/options no Debian e no Ubuntu:

```
asyncmap 0
auth
crtscts
lock
hide-password
modem
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
noipx
```

Gostamos de utilizar este arquivo **options**:

```
# Global PPP options
lock                    # Sempre bloqueia o dispositivo que você está utilizando
asyncmap 0x00000000     # Por padrão, não escapa nada
crtscts                 # Utiliza controle de fluxo de hardware
defaultroute            # Adiciona rota padrão por meio da interface ppp
mru 552                 # MRU/MTU 512 (data) + 40 (header)
mtu 552
```

O arquivo **/etc/sysconfig/network-scripts/ifcgf-ppp0** a seguir é proveniente de um sistema Red Hat. Esse arquivo esqueleto foi construído pelo utilitário **linuxconf**.

```
PERSIST=yes
DEFROUTE=yes
ONBOOT=no
INITSTRING=ATZ
MODEMPORT=/dev/modem
LINESPEED=115200
ESCAPECHARS=no
DEFABORT=yes
HARDFLOWCTL=yes
DEVICE=ppp0
PPPOPTIONS=
```

```
DEBUG=yes
PAPNAME=remote
REMIP=
IPADDR=
BOOTPROTO=none
MTU=
MRU=
DISCONNECTTIMEOUT=
RETRYTIMEOUT=
USERCTL=no
```

Eis um exemplo de script de chat (**chat-ppp0**) correspondente ao arquivo **ifcfg-ppp0** anterior (com toda a sua sintaxe concisa e ligeiramente bizarra):

```
'ABORT' 'BUSY'
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'ABORT' 'Invalid Login'
'ABORT' 'Login incorrect'
'''ATZ'
'OK' 'ATDT número de telefone'
'CONNECT' '      '
'TIMEOUT' '120'
'ogin:' 'conta'
'ord:' 'senha'
'TIMEOUT' '5'
'~--' '   '
```

Várias linhas desse script de chat contêm um parâmetro nulo indicado por um par de aspas simples, que se parecem com aspas duplas nesta fonte.

Normalmente podemos adaptar um script de chat existente ao nosso ambiente sem nos preocuparmos muito com o seu funcionamento exato. Aqui, as linhas iniciais configuram algumas condições gerais nas quais o script deve abortar. As linhas seguintes inicializam o modem e discam o telefone, e as linhas restantes esperam por uma conexão e introduzem o nome de usuário e senha apropriados.

O tempo de expiração no script de chat algumas vezes precisa ser ajustado para lidar com situações de discagem complicadas como aquelas em hotéis ou estabelecimentos comerciais com comutadores telefônicos locais ou para lidar com o sinal de voice mail que algumas empresas utilizam antes de darem um tom de discagem real. Na maioria dos modems, uma vírgula no número de telefone indica uma pausa na discagem. Talvez sejam necessárias várias vírgulas se você tiver de discar um determinado dígito e depois aguardar um segundo tom de discagem antes de prosseguir.

Logins PPP em nossa instalação são apenas nomes de usuários com um P na frente deles. Essa convenção facilita memorizar a quem uma determinada máquina PPP pertence.

A associação entre **ifcfg-ppp0** e **chat.ppp0** é feita pelo comando **ifup**, que será executado automaticamente durante a inicialização uma vez que o arquivo **ifcfg** existe. Você também pode chamar **pppd** explicitamente com um arquivo de opções específicas a uma determinada conexão como um argumento, desde que o arquivo contenha uma linha `connect` que liste o nome de arquivo chat correspondente.

Nosso próximo exemplo de discagem é de um sistema Debian. Ele utiliza o diretório **peers**, coloca seu script de chat no diretório **/etc/chatscripts** e utiliza o mecanismo de autenticação PAP em vez de armazenar a senha no script de chat. Primeiramente, as opções para essa conexão, **/etc/ppp/peers/my-isp**:

```
/dev/modem                ### preencha com a porta serial do seu modem
debug
crtscts
name nomeDeUsuário        ### nome de usuário no meuISP
remotename meuISP
noauth
noipdefault
defaultroute
connect '/usr/sbin/chat -v -f /etc/chatscripts/my-isp'
```

/etc/chatscripts/my-isp contém as seguintes entradas:

```
'ABORT' 'BUSY'
```

```
'ABORT' 'ERROR'
'ABORT' 'NO CARRIER'
'ABORT' 'NO DIALTONE'
'''ATZ'
'OK''ATDT número De Telefone'
'CONNECT'''
'TIMEOUT' 15
'~--'''
```

O arquivo de autenticação utilizado para se conectar ao ISP, **/etc/ppp/pap-secrets**, precisa conter a seguinte linha:

```
login-name meuISP senha
```

onde *meuISP* é o valor da variável `remotename` nas opções anteriores. Para ativar a conexão nessa situação, utilize o comando **pppd call my-isp**.

Eis um exemplo que utiliza PPP por meio da conectividade Internet genérica existente, mas que se associa a **ssh** para criar uma conexão segura por meio de uma VPN (Virtual Private Network). Mostramos tanto as configurações de servidor como as de cliente.

O arquivo **/etc/ppp/options** do servidor:

```
noauth
logfile pppd.log
passive
silent
nodetach
```

Cada conexão também possui um arquivo **/etc/ppp/options.***nome_do_tty* que contém as atribuições de endereços IP para a conexão:

```
endereço_IP_local:endereço_IP_remoto
```

O shell do usuário PPP é configurado como **/usr/sbin/pppd** no servidor de modo que o daemon de servidor é iniciado automaticamente. Todas as chaves de autenticação têm de ser configuradas antecipadamente com **ssh-agent** para que não seja solicitada nenhuma senha. No lado cliente, a configuração é feita no diretório **/etc/ppp/peers** com um arquivo nomeado para o servidor — chamemos a configuração de "my-work". O arquivo **/etc/ppp/peers/my-work** do cliente conteria:

```
noauth
debug
logfile pppd.log
passive
silent
pty "ssh -t usuário@host_remoto"
```

Para conectar-se ao escritório a partir de casa em uma conexão PPP segura, o usuário teria apenas de digitar **pppd call my-work**.

Finalmente, incluímos um exemplo que utiliza o comando **wvdial** e sua fácil configuração para evitar toda aquela mágica do chat script que parece ser necessária:

/etc/wvdial.conf:

```
[Dialer Defaults]
Phone = número_de_telefone
Username = nome_de_login
Password = senha
Modem = /dev/ttyS1

[Dialer creditcard]
Phone = código_de_acesso_de_longa_distância,,,número de telefone,,número_cc
```

Se **wvdial** for chamado sem o emprego de argumentos, ele usará a seção padrão do discador do arquivo **/etc/wvdial.conf** em **~/.wvdialrc** para fazer a chamada e iniciar o PPP. Se for chamado com um parâmetro (por exemplo, **wvdial creditcard**), ele usará a seção apropriada do arquivo de configuração para sobrescrever todos os parâmetros especificados na seção padrão.

Para interromper uma conexão PPP, é melhor utilizar **ifdown** do que simplesmente eliminar o daemon **pppd**. Se interrompermos **pppd** diretamente, o Linux perceberá e o reiniciará.

```
$ sudo ifdown ppp0
```

Se sua máquina for portátil e algumas vezes utilizar Ethernet em vez de PPP, pode haver uma rota padrão pela interface Ethernet antes de **pppd** ser inicializado. Infelizmente **pppd** é muito educado para "arrancar" a rota e instalá-la por conta própria, que é o comportamento que você quer realmente. Para corrigir esse problema, execute **ifdown** na interface apropriada para eliminar a rota.

Eis o aspecto da tabela de roteamento e da configuração da interface PPP depois de a conexão PPP ter sido estabelecida:

```
$ ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.0.0.56  P-t-P:10.0.0.55  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:125 errors:0 dropped:0 overruns:0 frame:0
          TX packets:214 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
 RX bytes:11446 (11.1 Kb)  TX bytes:105586 (103.1 Kb)

$ netstat -nr
Kernel IP routing table
Destination     Gateway     Genmask         Flags  MSS    Window irtt   Iface
10.0.0.55       0.0.0.0     255.255.255.255 UH     40     0      0      ppp0
0.0.0.0         10.0.0.55   0.0.0.0         UG     40     0      0      ppp0
```

Você pode obter dados estatísticos sobre a conexão PPP e os pacotes que ele transferiu usando o comando **pppstats**:

```
$ pppstats
    IN      PACK  VJCOMP  VJUNC   VJERR  |        OUT  PACK   VJCOMP  VJUNC  NON-VJ
 11862       133       8     96       0  |     110446  226       27     89     110
```

A coluna VJCOMP conta pacotes que utilizam a compactação de cabeçalho TCP de Van Jacobson, e a coluna VJUNC conta aqueles que não a utilizam. Consulte a RFC1144 para detalhes.

Depurar uma conexão PPP pode realmente ser uma dor de cabeça, pois há muitas peças em jogo. **pppd** envia registros de log ao recurso daemon por meio de syslog em sistemas Red Hat e Debian e para o recurso local2 no SUSE. Você pode aumentar o nível de registro em log utilizando o flag **debug** na linha de comando de **pppd** ou então solicitando mais registros em log no arquivo de opções. **pppd** também fornece códigos de saída detalhados em caso de falha, de forma que se você tentar executar **pppd** e ele se recusar, execute **echo $status** (antes de fazer qualquer outra coisa) para recuperar o código de saída e depois pesquisar esse valor na página man do **pppd**.

O SUSE tende a incluir arquivos de configuração de exemplo para cada subsistema; esses arquivos são principalmente comentários que explicam o formato e o significado das opções disponíveis. Os arquivos no diretório **/etc/ppp** do SUSE não são exceção; eles são bem documentados e contêm valores sugeridos razoáveis para muitos parâmetros.

O Debian também tem modelos de arquivo de configuração bem documentados para PPP. Ele possui um subdiretório, **/etc/chatscripts**, dedicado a scripts de chat. Para ativar uma interface com PPP, você pode incluí-la no arquivo **/etc/network/interfaces** com o método `ppp` e a opção `provider` para amarrar o nome de seu provedor (em nosso caso, my-isp) a um nome de arquivo no diretório **/etc/peers** (**/etc/peers/my-isp**). Por exemplo:

```
iface eth0 inet ppp
   provider my-isp
```

Nesse caso, os comandos **pon** e **poff** específicos ao Debian gerenciam a conexão.

12.14 IDIOSSINCRASIAS DE REDE DO LINUX

Diferentemente da maioria dos kernels, o Linux leva em consideração os bits TOS (Type-Of-Service) nos pacotes IP e dá prioridade aos pacotes que estão rotulados como interativos (baixa latência). Legal! Infelizmente, problemas mentais por parte da Microsoft fazem com que tenhamos de desativar esse comportamento perfeitamente razoável.

Todos os pacotes que se originam de sistemas Windows mais antigos são rotulados como sendo interativos, independentemente dos seus propósitos. Por outro lado, os sistemas UNIX normalmente não marcam nenhum pacote como interativo. Se seu gateway Linux atende a uma rede mista de sistemas UNIX e Windows, os pacotes do Windows receberão incondicionalmente tratamento preferencial. Se trabalhar em um ambiente com algumas tecnologias mais antigas, o impacto sobre o desempenho para o UNIX pode ser bem significativo.

Você pode desativar a classificação de pacotes baseados em TOS ao compilar o kernel do Linux. Basta dizer não à opção "IP: utilize o valor TOS como chave de roteamento".

Quando o mascaramento IP (NAT) estiver ativado, ele informa o kernel para remontar fragmentos de pacotes em um pacote completo antes de encaminhá-los, mesmo que o kernel tenha de fragmentar o pacote novamente, de imediato, para

poder enviá-lo. Essa remontagem pode significar um número razoável de ciclos de CPU, mas hoje em dia as CPUs são rápidas o bastante de modo que isso realmente não se constitui em problema para as máquinas modernas.

O Linux permite que alteremos os endereços em nível MAC de certos tipos de interfaces de rede:

```
redhat$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:02:B3:19:C8:87
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:7 Base address:0xee80

redhat$ sudo ifconfig eth1 hw ether 00:02:B3:19:C8:21

redhat$ ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:02:B3:19:C8:21
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:7 Base address:0xee80
```

Trata-se de um recurso perigoso que tende a danificar as coisas. Pode ser útil, mas ele só o utiliza como um último recurso.

12.15 LEITURA RECOMENDADA

STEVENS, W. RICHARD. *TCP/IP Illustrated, Volume One:* The Protocols. Reading, MA: Addison-Wesley, 1994.

WRIGHT, GARY R. E STEVENS, W. RICHARD. *TCP/IP Illustrated, Volume Two:* The Implementation. Reading, MA: Addison-Wesley, 1995.

Estes dois livros são um excelente e completo guia para a estrutura de protocolos TCP/IP. Um pouco obsoletos, mas ainda sólidos.

STEVENS, W. RICHARD E FENNER, BILL E RUDOFF, ANDREW M. *UNIX Network Programming, Volume 1, The Sockets Networking API. 3. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2003.

STEVENS, W. RICHARD. *UNIX Network Programming, Volume 2:* Interprocess Communications. *2. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 1999.

STEVENS, W. RICHARD. *UNIX Network Programming.* Upper Saddle River, NJ: Prentice Hall, 1990.

Esses livros são as bíblias do aluno nas aulas de rede que envolvam programação. Se só precisar da interface de soquetes BERKELEY, a edição original é uma boa referência. Se também precisar da interface STREAMS, a terceira edição, que inclui o IPv6, é uma boa aposta. Todos os três são muito bem escritos no estilo típico de Rich Stevens.

TANENBAUM, ANDREW. *Computer Networks. 4. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2003.

Esse foi o primeiro texto sobre redes e continua um clássico. Ele contém uma descrição completa de todos os detalhes práticos que acontecem nas camadas de links físicos da estrutura de protocolos. A última edição inclui cobertura das redes sem fio, Ethernet de gigabit, redes não-hierárquicas, voz sobre IP e mais.

SALUS, PETER H. *Casting the Net, From ARPANET to INTERNET and Beyond.* Reading, MA: Addison-Wesley Professional, 1995.

Essa é uma história encantadora sobre a ARPANET à medida que ela cresceu e tornou-se a Internet, escrita por um historiador que há muito tempo acompanha os passos da comunidade UNIX!

COMER, DOUGLAS. *Internetworking with TCP/IP Volume 1:* Principles, Protocols, and Architectures. *5. ed.* Upper Saddle River, NJ: Pearson Prentice Hall, 2006.

A série *Internetworking with TCP/IP* de Comer Doug foi por muito tempo a referência padrão para os protocolos TCP/IP. Esses livros foram projetados como textos universitários e são uma boa fonte introdutória a material informativo.

HEDRICK, CHARLES. *Introduction to the Internet Protocols.* Rutgers University, 1987.

Esse documento é uma dócil introdução ao TCP/IP. Ele parece não ter um site permanente, mas é amplamente distribuído na Web; procure-o.

HUNT, CRAIG. *TCP/IP Network Administration. (3. ed.).* Sebastopol, CA: O'Reilly Media, 2002.

Como outros livros da série *Nutshell*, esse se destina a administradores de sistemas UNIX. Metade do livro é sobre TCP/IP e o restante trata de tópicos do UNIX de nível mais alto como e-mail e login remoto.

Uma excelente coleção de documentos sobre a história da Internet e suas várias tecnologias pode ser encontrada em www.isoc.org/internet/history.

12.16 EXERCÍCIOS

E12.1 Como o atendimento a (isto é, a obediência a) redirecionamentos ICMP permite que um usuário não-autorizado comprometa a rede?

E12.2 O que é MTU de um link de rede? O que acontece se o MTU para um dado link for configurado com um valor muito alto? E muito baixo?

★ E12.3 Explique o conceito de sub-redes e explique a razão de sua utilidade. O que são máscaras de rede? Como as máscaras de rede se relacionam com a divisão das partes rede e host de um endereço IP?

★ E12.4 A rede 134.122.0.0/16 foi subdividida em /19 redes.

a) Quantas redes /19 existem? Enumere-as. Quais são suas máscaras de rede?

b) Quantos hosts poderia haver em cada rede?

c) Determine à qual rede o endereço IP 134.122.67.124 pertence.

d) Qual é o endereço de broadcast para cada rede?

★ E12.5 O host 128.138.2.4 na rede 128.138.2.0/24 deseja enviar um pacote para o host 128.138.129.12 na rede 128.138.129.0/24. Suponha o seguinte:

- O host 128.138.2.4 tem uma rota padrão por meio de 128.138.2.1.
- O host 128.138.2.4 acaba de ser inicializado e não enviou nem recebeu nenhum pacote.
- Todas as outras máquinas na rede já estão executando por um longo período de tempo.
- O roteador 128.138.2.1 tem um link direto com 128.138.129.1, o gateway para a sub-rede 128.138.129.0/24.

a) Liste todas as etapas necessárias para enviar o pacote. Mostre os endereços Ethernet e IP de origem e de destino de todos os pacotes transmitidos.

b) Se a rede fosse 128.138.0.0/16, a sua resposta mudaria? Como ou por que não?

c) Se a rede 128.138.2.0 fosse uma rede /26 em vez de uma /24, a sua resposta mudaria? Como ou por que não?

★★ E12.6 Após instalar um sistema Linux novo, como você poderia solucionar as questões relativas à segurança mencionadas neste capítulo? Verifique se algum desses problemas de segurança foi solucionado nos seus sistemas Linux. (Pode ser que seja necessário acesso root.)

★★ E12.7 Quais os passos necessárias para acrescentar uma máquina nova à rede de seu ambiente? Na resposta, utilize parâmetros apropriados para a sua rede e situação local. Suponha que a nova máquina já execute o Linux.

★★ E12.8 Mostre o arquivo de configuração necessário para configurar um servidor DHCP que atribui endereços no intervalo 128.138.192.[1-55]. Utilize um tempo de locação de duas horas e garanta que o host com endereço Ethernet 00:10:5A:C7:4B:89 sempre receba o endereço IP 128.138.192.55.

13 Roteamento

Controlar para onde o tráfego de rede deve fluir não é uma tarefa fácil. O Capítulo 12 fez uma breve introdução ao encaminhamento de pacotes IP. Neste capítulo, examinamos o processo de encaminhamento com mais detalhes e investigamos vários protocolos de rede que permitem que os roteadores descubram automaticamente rotas eficientes. Os protocolos de roteamento não somente reduzem o encargo administrativo do dia-a-dia de manter informações de roteamento, como também permitem que o tráfego de rede seja redirecionado rapidamente caso um roteador ou rede falhe.

É importante fazer a distinção entre o processo de efetivamente encaminhar pacotes IP e o gerenciamento da tabela de roteamento que dirige esse processo, ambos comumente denominados de "roteamento". O encaminhamento de pacotes é simples, enquanto o cálculo da rota é complexo; assim, o segundo significado é utilizado com mais freqüência na prática. Este capítulo descreve somente o roteamento unicast; o roteamento multicast envolve uma série de problemas muito diversos e está além do escopo deste livro.

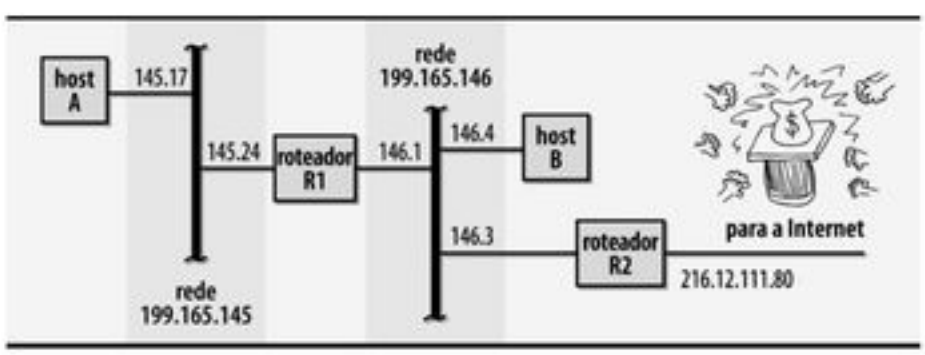
Para a vasta maioria dos casos, as informações abordadas no Capítulo 12, *Redes TCP/IP*, é tudo que você precisa saber sobre roteamento. Se a infra-estrutura de rede apropriada já está instalada, você pode configurar um único roteamento estático (como descrito na seção *Roteamento* que começa na página 205) e *voilà*, tem informações suficientes para alcançar praticamente todos os lugares da Internet. Se você deve sobreviver dentro de uma topologia de rede complexa ou se estiver utilizando um sistema Linux para parte da infra-estrutura de rede, então as informações deste capítulo sobre protocolos e ferramentas de roteamento dinâmico podem ser úteis.

Convencionalmente diz-se que o roteamento IP é excepcionalmente difícil, compreendido apenas por alguns poucos hippies cabeludos que vivem nos dutos do sistema de aquecimento sob o campus dos Laboratórios Lawrence Berkeley na Califórnia. Na realidade, esse não é o caso, desde que você entenda a premissa básica de que o roteamento IP é o roteamento do "próximo hop". A qualquer dado momento, você somente precisa determinar o *próximo* host ou roteador na jornada de um pacote até o seu destino final. Essa é uma abordagem diferente daquela de muitos protocolos mais antigos que determinam o caminho exato que um pacote percorrerá antes de deixar seu host original, um esquema conhecido como roteamento de origem.[1]

13.1 Encaminhamento de pacotes: uma visão mais detalhada

Antes de pularmos para o gerenciamento de tabelas de roteamento, vejamos com mais detalhes a maneira como as tabelas são utilizadas. Consideremos a rede mostrada na Ilustração A.

Ilustração A Exemplo de redes



1 Os pacotes de IP também podem ser roteados pela fonte, mas isso quase nunca é feito. O recurso não é amplamente aplicado devido a questões de segurança.

O roteador R1 conecta duas redes; e o roteador R2 conecta uma das redes ao mundo externo. (Por enquanto, assumiremos que R1 e R2 são computadores com Linux, e não roteadores dedicados.) Vejamos algumas tabelas de roteamento e alguns cenários específicos de encaminhamento de pacotes. Primeiramente, apresentamos a tabela de roteamento do host A:

```
A$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS    Window irtt   Iface
199.165.145.0   0.0.0.0         255.255.255.0   U        0    0         0   eth0
127.0.0.0       0.0.0.0         255.0.0.0       U        0    0         0   lo
0.0.0.0         199.165.145.24  0.0.0.0         UG       0    0         0   eth0
```

Consulte a página 209 para informações adicionais sobre o ifconfig.

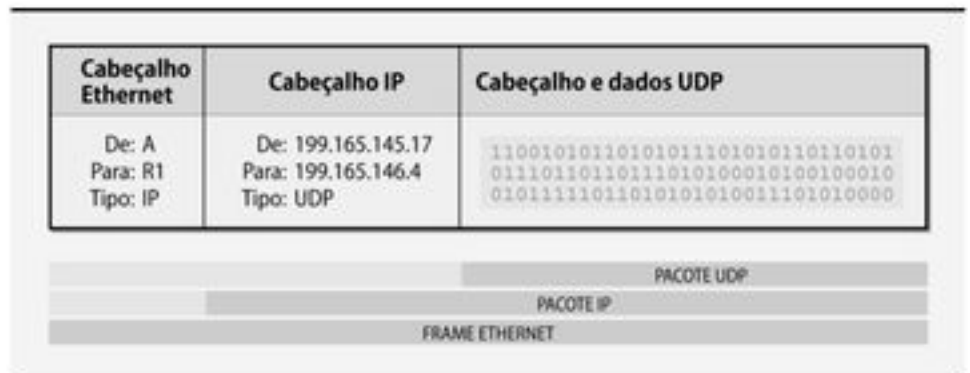
O host A possui a configuração de roteamento mais simples das quatro máquinas. As duas primeiras rotas descrevem as interfaces de rede da máquina em termos de roteamento padrão. Essas entradas existem para que o encaminhamento a redes conectadas diretamente não precise ser tratado como um caso especial. eth0 é a interface Ethernet do host A e lo é a interface de loopback, uma interface de rede virtual emulada por software. Entradas como essas são normalmente adicionadas de forma automática pelo **ifconfig** quando a interface de rede é configurada.

A rota padrão no host A encaminha todos os pacotes não destinados ao endereço de loopback ou para a rede 199.165.145 para o roteador R1, cujo endereço nesta rede é 199.165.145.24. O flag G indica que essa rota vai para um gateway, não para uma das interfaces locais de A. Os gateways devem estar no máximo a um hop de distância.

Consulte a página 196 para informações adicionais sobre endereçamento.

Suponha que um processo em A envie um pacote para B, cujo endereço é 199.165.146.4. A implementação do IP procura uma rota para a rede de destino, 199.165.146, mas nenhuma delas corresponde às especificações. A rota padrão é chamada e o pacote é encaminhado para R1. A ilustração B ilustra o pacote que sai efetivamente da Ethernet (os endereços contidos no cabeçalho Ethernet são os endereços MAC das interfaces de A e de R1 na rede 145).

Ilustração B Pacote Ethernet



O endereço de hardware de destino Ethernet é aquele do roteador R1, mas o pacote IP oculto dentro do frame Ethernet simplesmente não menciona R1. Ao inspecionar o pacote recebido, R1 verá pelo endereço IP do destino que não se trata do último destino do pacote. Ele então utilizará sua própria tabela de roteamento para encaminhar o pacote para o host B sem regravar o cabeçalho IP, de forma que ele ainda mostra o pacote proveniente de A.

Eis a tabela de roteamento para o host R1:

```
R1$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS    Window irtt   Iface
127.0.0.0       0.0.0.0         255.0.0.0       U        0    0         0   lo
199.165.145.0   0.0.0.0         255.255.255.0   U        0    0         0   eth0
199.165.146.0   0.0.0.0         255.255.255.0   U        0    0         0   eth1
0.0.0.0         199.165.146.3   0.0.0.0         UG       0    0         0   eth1
```

Essa tabela é similar àquela do host A, exceto pelo fato de mostrar duas interfaces de rede físicas. A rota padrão nesse caso aponta para R2, já que é o gateway por meio do qual a Internet pode ser alcançada. Pacotes associados a qualquer uma das redes 199.165 podem ser encaminhados diretamente.

Da mesma maneira que o host A, o host B possui apenas uma interface de rede real. Entretanto, B precisa de uma rota adicional para funcionar corretamente porque tem conexões diretas a dois roteadores diferentes. O tráfego para a rede 199.165.145 tem de passar por R1, enquanto outros dados têm de sair para a Internet por R2.

```
B$ netstat -rn
Kernel IP routing table
Destination       Gateway        Genmask        Flags  MSS    Window irtt   Iface
127.0.0.0         0.0.0.0        255.0.0.0      U      0      0         0   lo
199.165.145.0     199.165.146.1  255.255.255.0  U      0      0         0   eth0
199.165.146.0     0.0.0.0        255.255.255.0  U      0      0         0   eth0
0.0.0.0           199.165.146.3  0.0.0.0        UG     0      0         0   eth0
```

Consulte página 206 para uma explicação sobre os redirecionamentos ICMP.

Você pode configurar o host B com conhecimento inicial de apenas um gateway, contando assim com a ajuda de redirecionamentos ICMP para eliminar hops extras. Por exemplo, eis uma possível configuração inicial para o host B:

```
B$ netstat -rn
Kernel IP routing table
Destination       Gateway        Genmask        Flags  MSS    Window irtt   Iface
127.0.0.0         0.0.0.0        255.0.0.0      U      0      0         0   lo
199.165.146.0     0.0.0.0        255.255.255.0  U      0      0         0   eth0
0.0.0.0           199.165.146.3  0.0.0.0        UG     0      0         0   eth0
```

Se B enviar então um pacote para o host A (199.165.145.17), nenhuma rota corresponderá e o pacote será encaminhado para R2 para entrega. R2 (que, por ser um roteador, provavelmente tem informações completas sobre a rede) envia o pacote para R1. Já que R1 e B se encontram na mesma rede, R2 também envia um aviso de redirecionamento ICMP para B, e B insere uma rota de host para A na sua tabela de roteamento:

```
199.165.145.17    199.165.146.1  255.255.255.255  UGHD  0       0        0    eth0
```

Essa rota envia todo o tráfego futuro para A diretamente por R1. Entretanto, ela não afeta o roteamento para os demais hosts na rede de A, todos os quais tem de ser roteados por redirecionamentos distintos a partir de R2.

Algumas instalações optaram pelos redirecionamentos ICMP como "protocolo" de roteamento principal, imaginando que essa abordagem seria dinâmica. Infelizmente, assim que o kernel aprende uma rota a partir de um redirecionamento, a rota tem de ser manualmente excluída, ou a máquina tem de ser reinicializada se essas informações mudarem. Devido a esse problema e a várias outras desvantagens de redirecionamentos (aumento de carga na rede, carga aumentada em R2, desorganização completa na tabela de roteamento, dependência de servidores extras), não recomendamos o emprego de redirecionamentos para configurações como essas. Em uma rede configurada apropriadamente, os redirecionamentos jamais devem aparecer na tabela de roteamento.

13.2 DAEMONS DE ROTEAMENTO E PROTOCOLOS DE ROTEAMENTO

Em redes simples como aquela mostrada na Ilustração A, é perfeitamente razoável configurar o roteamento manualmente. Em algum momento, porém, as redes se tornam muito complexas para serem gerenciadas desse modo (possivelmente devido à sua taxa de crescimento). Em vez de ter de informar explicitamente cada computador em cada uma das redes como atingir cada um dos demais computadores e redes, seria interessante se os computadores pudessem simplesmente unir suas inteligências e descobrir tudo isso. Essa é a tarefa dos protocolos de roteamento e dos daemons que os implementam.

Os protocolos de roteamento possuem uma grande vantagem em relação a sistemas de roteamento estáticos já que eles reagem e se adaptam a mudanças das condições da rede. Se um link cair, os daemons de roteamento poderão descobrir e propagar rapidamente rotas alternativas para as redes que o link servia, se existir uma rota assim.

Os daemons de roteamento coletam informações de três fontes: arquivos de configuração, tabelas de roteamento existentes e daemons de roteamento em outros sistemas. Essas informações são mescladas para calcular um conjunto de rotas otimizado, e a tabela de roteamento do sistema (e possivelmente outros sistemas por meio de um protocolo de roteamento) é então realimentada com as novas rotas. Pelo fato de as condições de rede mudarem ao longo do tempo, os daemons de roteamento devem consultar uns aos outros periodicamente para se certificarem de que suas informações de roteamento ainda estão válidas.

A maneira exata como as rotas são calculadas depende do protocolo de roteamento. Dois tipos de protocolos estão comumente em uso: protocolos de vetor de distância e protocolos de estado de link.

Protocolos de vetor de distância

Os protocolos de vetor de distância (também conhecidos como "fofoqueiros") se baseiam na idéia geral:"Se o roteador X se encontra a cinco hops de distância da rede Y e eu me encontro em uma posição adjacente ao roteador X, então tenho de estar a seis hops da rede Y.". Você anuncia a que distância se encontra das redes sobre as quais tem conhecimento. Se seus vizinhos não tiverem conhecimento de uma maneira melhor de atingir cada uma das redes, eles marcam você como o

melhor gateway. Se já conhecerem uma rota mais curta, eles ignoram seus anúncios.[2] Com o tempo, supostamente as tabelas de roteamento devem convergir para um estado estável.

Trata-se realmente de uma idéia muito elegante. Se ela funcionasse da forma anunciada, o roteamento seria relativamente simples. Infelizmente esse tipo de algoritmo não é capaz de lidar bem com mudanças na topologia. Em alguns casos, loops infinitos (por exemplo, o roteador X recebe informações do roteador Y e as envia para o roteador Z que, por sua vez, as envia de volta para o roteador Y) podem impedir que as rotas convirjam de fato. Os protocolos de vetor de distância do mundo real devem evitar esses problemas introduzindo heurística complexa ou forçando restrições arbitrárias como a noção do RIP (Routing Information Protocol) em que toda rede com mais de 15 hops de distância é inatingível.

Mesmo em casos não patológicos, pode levar vários ciclos de atualização para que todos os roteadores atinjam um estado estável. Portanto, para garantir que o roteamento não ficará empacado por um longo período, o tempo de ciclo tem de ser curto, e, por essa razão, os protocolos de vetor de distância como um todo tendem a ser prolixos. Por exemplo, o RIP requer que os roteadores façam um broadcast de todas as suas informações de roteamento a cada 30 segundos. O IGRP e o EIGRP enviam atualizações a cada 90 segundos.

Por outro lado, o BGP (Border Gateway Protocol) transmite toda a tabela de uma só vez e depois transmite as modificações à medida que elas forem ocorrendo. Essa otimização reduz substancialmente o potencial para tráfego "de bate-papo" (e, em sua maioria, desnecessário).

A Tabela 13.1 lista os protocolos de vetor de distância que estão em uso comum hoje.

Tabela 13.1 Protocolos de roteamento de vetor de distância comuns

Protocolo	Nome por extenso	Aplicativo
RIP	Routing Information Protocol	LANs internas
IGRP	Interior Gateway Routing Protocol (obsoleto)	Pequenas WANs
EIGRP	Enhanced Interior Gateway Routing Protocol	WANs, LANs corporativas
BGP	Border Gateway Protocol	roteamento de backbone Internet

Protocolos de estado de link

Os protocolos de estado de link distribuem informações de uma maneira relativamente não processada. Os registros negociados entre os roteadores são normalmente do tipo "O roteador X é adjacente ao roteador Y e o link está ativo". Um conjunto completo de tais registros forma um mapa de conectividade da rede por meio do qual cada roteador pode computar sua própria tabela de roteamento. A principal vantagem dos protocolos de estado de link em relação aos protocolos de vetor de distância é a capacidade de convergir rapidamente para uma solução de roteamento operacional após a ocorrência de alguma pane. A desvantagem é que manter um "mapa" completo da rede em cada nó requer memória e capacidade de CPU que não seriam necessários em um sistema de roteamento de vetor de distância.

Como as comunicações entre os roteadores em um protocolo de estado de link não fazem parte do algoritmo de cálculo das rotas em si, elas podem ser implementadas de tal maneira que não ocorram loops de transmissão. As atualizações para o banco de dados de topologia se propagam por meio da rede de modo eficiente, a um custo menor em termos de largura de banda da rede e de tempo de CPU.

Os protocolos de estado de link tendem a ser mais complicados do que os protocolos de vetor de distância, mas essa complexidade pode ser explicada em parte pelo fato de os protocolos de estado de link facilitarem a implementação de recursos avançados como o roteamento de tipos de serviço e rotas múltiplas para o mesmo destino. Nenhum desses recursos é suportado em sistemas Linux comuns; você tem de utilizar roteadores dedicados para tirar proveito deles.

Os protocolos de estado de link comuns são mostrados na Tabela 13.2.

Tabela 13.2 Protocolos de roteamento de estado de link comuns

Protocolo	Nome completo	Aplicação
OSPF	Open Shortest Path First	LANs internas, WANs pequenas
IS-IS	Intermediate System to Intermediate System	Experiências de laboratório, asilos de loucos

2 Na verdade, não é tão simples assim, já que há regras para manipulação de alterações na topologia que podem aumentar a distância de rotas existentes. Alguns protocolos de vetor de distância como o EIGRP mantêm informações sobre várias rotas possíveis de modo que elas sempre têm um plano de emergência. Os detalhes exatos não são importantes.

Métrica de custo

Para que um protocolo de roteamento determine qual caminho para uma dada rede é o mais curto, ele tem de definir o que significa "é mais curto"[3]. É o caminho envolvendo o menor número de hops? O caminho de menor latência? A maior largura de banda intermediária mínima? O custo financeiro mais baixo?

Para o roteamento, a qualidade de um link é representada por um número chamado de métrica de custo. Um custo de caminho é a soma dos custos de cada link no caminho. Nos sistemas mais simples, cada link tem um custo de 1, o que torna a contagem de hops uma métrica de caminho. Mas qualquer uma das considerações mencionadas pode ser convertida em uma métrica de custo numérica.

Os especialistas em redes trabalharam arduamente e por longo tempo para tornar a definição de métrica de custo flexível, e alguns protocolos modernos permitem até que sejam utilizadas diferentes métricas para diferentes tipos de tráfego de rede. Não obstante, em 99% dos casos, todo esse árduo trabalho pode ser ignorado com segurança. A métrica padrão funciona perfeitamente bem para a maioria dos sistemas.

Você poderá se deparar com situações em que o verdadeiro caminho mais curto para um determinado destino talvez não seja uma boa rota padrão por questões políticas. Para tratar esses casos, você pode aumentar artificialmente o custo de links críticos para fazer com que eles pareçam menos atraentes. Deixe como está o restante da configuração de roteamento.

Protocolos internos e externos

Um "sistema autônomo" é um grupo de redes sob o controle político e administrativo de uma única entidade. A definição é vaga; sistemas autônomos do mundo real podem ser grandes como uma rede corporativa mundial ou pequenos como uma rede em um prédio ou de um único departamento acadêmico. Tudo depende de como você quer gerenciar o roteamento. A tendência geral é fazer os sistemas autônomos terem o maior tamanho possível. Essa convenção simplifica a administração e torna o roteamento o mais eficiente possível.

O roteamento no interior de um sistema autônomo é um tanto diverso do roteamento entre sistemas autônomos. Os protocolos para roteamento entre sistemas autônomos (protocolos "externos") normalmente têm de lidar com rotas de várias redes e eles têm de lidar de forma graciosa com o fato de que os roteadores vizinhos estarem sob o controle de uma outra pessoa. Os protocolos externos não revelam a topologia dentro de um sistema autônomo, no sentido de que eles podem ser imaginados como um segundo nível de hierarquia de roteamento que lida com conjuntos de redes em vez de cabos ou hosts individuais.

Na prática, instalações de pequeno e médio porte raramente precisam executar um protocolo externo a menos que eles estejam conectados a mais do que um ISP. Com vários ISPs, a fácil divisão de redes em domínios locais e Internet se reduz e os roteadores têm de decidir qual rota para a Internet é a melhor para um determinado endereço. (Entretanto, isso não quer dizer que *cada* um dos roteadores tenha de conhecer essas informações. A maioria dos hosts pode permanecer ignorante e rotear os seus pacotes (rota padrão) por meio de um gateway que esteja mais bem informado.)

Embora os protocolos externos não sejam tão diferentes de seus equivalentes internos, este capítulo se concentra nos protocolos internos e nos daemons que os suportam. Se sua instalação também tiver de utilizar um protocolo externo, consulte a lista de leitura recomendada no final deste capítulo para algumas referências sugeridas.

13.3 VISÃO GERAL DOS PROTOCOLOS

Vários protocolos de roteamento internos estão sendo utilizados hoje em dia. Nesta seção, apresentaremos os principais concorrentes e resumiremos suas principais vantagens e fraquezas.

RIP: Routing Information Protocol

O RIP, definido na RFC1058, é um antigo protocolo da Xerox que foi adaptado para redes IP. Ele é o protocolo utilizado por **routed**. O RIP é um protocolo de vetor de distância simples que utiliza contagem de hops como métrica de custo. Pelo fato de ter sido desenvolvido em uma era em que um único computador custava centenas de milhares de dólares e as redes eram relativamente pequenas, o RIP considera qualquer host a 15 ou mais hops de distância como inatingível. Portanto, grandes redes locais que possuam mais de 15 roteadores ao longo de qualquer trajeto não poderão utilizar o protocolo RIP.

Embora o RIP seja grande consumidor de recursos devido ao seu uso abusivo de broadcasting, ele faz um excelente trabalho quando uma rede é modificada com freqüência ou quando a topologia de redes remotas não é conhecida. Entretanto, sua estabilização pode ser lenta após a queda de um link.

---

3 Felizmente, ele não tem de definir qual é o significado de "é".

Muitas instalações utilizam **routed** em seu modo **-q** ("quiet"), no qual ele gerencia a tabela de roteamento e ouve atualizações de roteamento na rede, mas não faz broadcast de informações próprias. Nessas instalações, os cálculos de rotas em si são normalmente executados com um protocolo mais eficiente como o OSPF (veja a seguir). As rotas computadas são convertidas em atualizações RIP para consumo por parte de máquinas não roteadoras. **routed** não consome muitos recursos (no modo **-q**) e é suportado universalmente; portanto, a maioria das máquinas pode desfrutar dos benefícios do roteamento dinâmico sem qualquer configuração especial.

O RIP é amplamente implementado em outras plataformas que não o Linux. Uma série de dispositivos comuns que vão desde impressoras a componentes de rede gerenciados por SNMP são capazes de ouvir anúncios RIP para tomar conhecimento de possíveis gateways. Além disso, **routed** está disponível para todas as versões do Unix e Linux, de modo que o RIP é de fato o menor denominador comum em termos de protocolo de roteamento. Muitas vezes o RIP é utilizado para roteamento de LANs, e um protocolo com mais recursos é utilizado para conectividade de redes remotas.

RIP-2: Routing Information Protocol, versão 2

Consulte a página 201 para informações sobre o endereçamento sem classe, conhecido como CIDR.

O RIP-2 é uma revisão moderada do RIP que acrescenta suporte a alguns poucos recursos que faltavam no protocolo original. A alteração mais importante é que o RIP-2 distribui máscaras de rede junto com endereços do próximo hop, portanto seu suporte a redes em sub-redes e CIDR é melhor do que o do RIP. Um vago gesto na tentativa de aumentar a segurança do RIP também foi incluído, mas a definição de um sistema de autenticação específico foi deixada para desenvolvimento futuro.

O RIP-2 fornece vários recursos que parecem destinados a esse ambiente multiprotocolo. As atualizações de "próximo hop" permitem aos broadcasters anunciarem rotas para as quais eles não são o gateway real, e "tags de rotas" permitem que rotas descobertas externamente sejam propagadas por meio do RIP.

O RIP-2 pode ser executado em um modo de compatibilidade que preserva a maioria dos novos recursos de RIP-2 sem abandonar inteiramente os receptores RIP convencionais. Na maioria dos aspectos, o RIP-2 é idêntico ao RIP e deve ser preferido a este se for suportado pelos sistemas que você estiver utilizando. Entretanto, as distribuições Linux geralmente não o suportam da maneira como ele vem implementado inicialmente.

OSPF: Open Shortest Path First

O OSPF é definido na RFC2328. Trata-se de um protocolo de estado de link. "Shortest path first" ("o caminho mais curto primeiro") refere-se ao algoritmo matemático utilizado para calcular as rotas; "open" ("aberto") é utilizado no sentido de "não-proprietário".

O OSPF foi o primeiro protocolo de roteamento de estado de link a ser amplamente utilizado e ainda é o mais popular. Sua adoção em larga escala foi impelida em grande parte pelo seu suporte no **gated**, um daemon de roteamento multiprotocolo popular sobre o qual falaremos a respeito mais adiante. Infelizmente, o protocolo em si é muito complexo e, portanto, só vale a pena em instalações de tamanho significativo (em que o comportamento do protocolo de roteamento realmente faz diferença).

A especificação de protocolo OSPF não impõe nenhuma métrica de custo particular. A implementação da Cisco utiliza contagens de hop por padrão e também pode ser configurada para usar a largura de banda de rede como uma métrica de custo.

O OSPF é um protocolo que funciona bem para grandes e complexas topologias. Ele oferece várias vantagens em relação ao RIP, inclusive a capacidade de controlar vários caminhos para um único destino e a capacidade de dividir a rede em seções ("áreas") que compartilham somente informações de roteamento de alto nível.

IGRP e EIGRP: Interior Gateway Routing Protocol

O IGRP e seu sucessor aprimorado, o EIGRP, são protocolos de roteamento proprietários que rodam somente em roteadores Cisco. O IGRP foi criado para atender algumas das deficiências do RIP antes da existência de padrões robustos como o OSPF. Hoje se tornou obsoleto em favor do EIGRP, embora ainda esteja em uso em muitas instalações. O EIGRP é configurado de forma semelhante ao IGRP, embora seja realmente bem diferente em seu projeto de protocolo subjacente. O IGRP trata somente anúncios de roteamento que respeitam fronteiras de classes de endereços IP tradicionais, enquanto o EIGRP entende máscaras de rede CIDR arbitrárias.

Tanto o IGRP quanto o EIGRP são protocolos de vetor de distância, mas eles são projetados para evitar os problemas de looping e de convergência encontrados em outros sistemas de vetor de distância. O EIGRP em particular é amplamente considerado um modelo de perfeição do roteamento de vetor de distância. Para a maioria dos propósitos, EIGRP e OSPF são igualmente funcionais.

Em nossa opinião, é melhor ater-se a um protocolo de roteamento estabelecido, não-proprietário e amplamente implementado como o OSPF. Mais pessoas estão utilizando e trabalhando com o OSPF do que com o EIGRP e várias implementações estão disponíveis.

IS-IS: o "padrão" ISO

IS-IS, o Intra-domain Intermediate System to Intermediate System Routing Protocol, é a resposta da International Organization for Standardization ao OSPF. Ele foi projetado originalmente para gerenciar "roteamento" para protocolos de rede OSI e foi posteriormente estendido para manipular roteamento IP.

Tanto IS-IS como OSPF foram desenvolvidos no início dos anos de 1990 quando os protocolos ISO estavam politicamente em voga. Uma atenção inicial por parte da IETF ajudou a atribuir ao IS-IS um verniz de legitimidade para IP, mas ele parece se distanciar cada vez mais do OSPF em termos de popularidade. Hoje, a utilização do IS-IS é extremamente rara fora de ambientes de teste de certificação de fornecedor. O protocolo em si é emaranhado com grande quantidade de "bagagem" ISO e, geralmente, deve ser evitado.

MOSPF, DVMRP e PIM: protocolos de roteamento multicast

MOSPF (Multicast OSPF, OSPF multicast), DVMRP (Distance Vector Multicast Routing Protocol) e PIM (Protocol Independent Multicast) são protocolos projetados para oferecer suporte ao multicasting de IP, uma tecnologia ainda não amplamente implementada. Você poderá encontrar indicadores para informações adicionais sobre esses protocolos em www.mbone.com.

Router Discovery Protocol

O RDP (Router Discovery Protocol) utiliza mensagens ICMP enviadas para o endereço de multicast IP 224.0.0.1 para anunciar e tomar conhecimento de outros roteadores em uma rede. Infelizmente, nem todos os roteadores fazem esses anúncios atualmente e nem todos os hosts os reconhecem. A esperança é que algum dia esse protocolo se torne mais popular.

13.4 ROUTED: CRIANDO SUAS PRÓPRIAS ROTAS

Talvez você não seja rico. Pode ser que você não tenha uma boa aparência. Mas certamente você sempre usou **routed**. **routed** foi por um bom tempo um daemon padrão de roteamento no UNIX e ainda é fornecido com a maioria das versões do UNIX e Linux.

O **routed** do Linux só fala RIP. Se você planeja utilizar RIP-2, o Nexus Routing Daemon encontra-se disponível em sourceforge.net/projects/nx-routed é uma opção fácil de configurar. O RIP-2 é essencial apenas se você tiver sub-redes com máscaras não em um limite de byte.

routed pode ser executado no modo servidor (**-s**) ou no modo silencioso (**-q**). Ambos os modos ouvem broadcasts, mas somente servidores distribuem suas próprias informações. Em geral, somente máquinas com múltiplas interfaces devem ser servidores. Se nem **-s** nem **-q** for especificado, **routed** executa no modo silencioso em hosts com uma interface e no modo de servidor em hosts com mais de uma interface.

*Consulte a página 212 para informações adicionais sobre **route**.*

routed acrescenta as rotas por ele descobertas à tabela de roteamento do kernel. As rotas têm de ser ouvidas novamente pelo menos a cada quatro minutos ou serão removidas. Entretanto, **routed** sabe quais rotas ele acrescentou e não elimina rotas estáticas que foram instaladas com o comando **route**.

routed -t pode ser utilizado para depurar o roteamento. Essa opção faz com que **routed** seja executado em primeiro plano e imprima todos os pacotes que ele envia ou recebe.

routed em geral descobre dinamicamente informações de roteamento e não requer configuração. Contudo, se a sua instalação contiver gateways para a Internet ou para outros sistemas autônomos, pode ser que você tenha de acrescentar alguns passos adicionais para fazer com que esses links funcionem com **routed**.

Se tiver apenas um único gateway externo, você pode anunciá-lo como uma rota padrão global executando seu **routed** com o flag **-g**. Isso é análogo a configurar a rota padrão em uma única máquina, exceto que ela é propagada por toda a rede.

routed também suporta um arquivo de configuração, **/etc/gateways**, que foi desenvolvido para fornecer informações estáticas sobre gateways para "pré-carga" na tabela de roteamento à qual foi aplicado **routed**.

13.5 GATED: O FIM DE UM BOM SOFTWARE LIVRE

gated era uma estrutura de roteamento fantástica gratuitamente disponível por meio da qual muitos protocolos de roteamento diferentes podiam ser utilizados simultaneamente. Com o **gated**, você podia controlar precisamente rotas anunciadas, endereços de broadcast, diretivas de confiança e métrica. **gated** compartilhava rotas entre vários protocolos, permitindo que gateways de roteamento fossem construídos entre áreas que tinham se padronizado com base em diferentes sistemas de roteamento. **gated** também tinha uma das mais elegantes interfaces administrativas e projeto de arquivo de configuração entre todos os softwares administrativos para Linux.

gated nasceu como um software livremente distribuído, mas em 1992 foi privatizado e se voltou para o Merit GateD Consortium. Versões comerciais de **gated** só estavam disponíveis aos membros desse consórcio. O consórcio foi por fim desmembrado e os direitos do **gated** foram adquiridos pela NextHop, um desenvolvedor de software de rede. Essa transferência encerrou a vida do **gated** no mundo do código-fonte aberto, deixando para trás uma trilha de histórias amargas.

Um novo e promissor projeto, o XORP (a eXtensible Open Router Platform), surgiu para ajudar a preencher o vazio criado quando o **gated** foi comprado. Embora esteja neste exato momento em beta-teste, o XORP está sendo finalizado para utilização em produção e estamos esperando que ele cresça para preencher o nicho inicial do **gated**. Confira os últimos progressos em www.xorp.org.

Nesse ínterim, os sites que precisarem de roteamento multiprotocolo podem considerar o uso do GNU Zebra (www.zebra.org), um pacote de roteamento prático e funcional que executa na maioria das plataformas Linux. Infelizmente, ele não tem a maioria dos recursos, as facilidades e a documentação detalhada necessárias para gerenciar o roteamento dinâmico em um ambiente de produção. Esse pode ser um caso em que comprar um roteador dedicado (como aqueles fabricados pela Juniper ou pela Cisco) é a melhor utilização de seus recursos.

13.6 CRITÉRIOS PARA SELEÇÃO DE UMA ESTRATÉGIA DE ROTEAMENTO

O roteamento de uma rede pode ser gerenciado essencialmente em quatro níveis de complexidade:

- Nenhum roteamento
- Somente rotas estáticas
- A maioria das rotas estáticas, mas os clientes ouvem atualizações RIP
- Roteamento dinâmico em toda parte

A topologia da rede como um todo possui um efeito dramático sobre as necessidades de roteamento de cada segmento individual. Redes diferentes podem precisar de níveis muito diversos de suporte a roteamento. As regras práticas a seguir podem ajudá-lo a escolher uma estratégia:

- Uma rede isolada não requer nenhum roteamento.
- Se uma rede tiver apenas uma saída, clientes (máquinas não-gateway) nessa rede devem ter uma rota padrão estática para o gateway isolado. Não é necessária nenhuma configuração adicional, exceto no gateway em si.
- Um gateway com um pequeno número de redes de um lado e um gateway para "o mundo" no outro lado pode ter rotas estáticas explícitas apontando para o primeiro e uma rota padrão para o último. Entretanto, o roteamento dinâmico é aconselhável se ambos os lados tiverem mais de uma escolha de roteamento.
- Se utilizar o RIP e estiver preocupado com a carga sobre rede e o sistema que isso requer, evite utilizar **routed** no modo ativo — ele transmite tudo que sabe (correto ou não) a pequenos intervalos. Para fazer clientes ouvirem passivamente as atualizações de roteamento sem enviar suas próprias informações, utilize **routed -q**.
- Muitas pessoas irão lhe informar que RIP é um protocolo horrível, terrível e que **routed** é uma criação do diabo. Não é necessariamente assim. Se funcionar e você ficar feliz com o desempenho, vá em frente e utilize-o. Você não ganha nada gastando tempo em uma estratégia de roteamento superprojetada.
- **routed** escuta todo mundo e acredita em tudo que ouve. Mesmo que seu site utilize RIP, talvez você queira gerenciar a troca de roteamento de dados com um roteador dedicado (como um Cisco) e executar **routed** apenas em hosts de clientes.
- Use roteamento dinâmico em pontos nos quais as redes cruzam fronteiras políticas ou administrativas.
- Em redes roteadas dinamicamente contendo loops ou caminhos redundantes, utilize OSPF se possível.
- Roteadores conectados à Internet com vários provedores para o backbone têm de usar BGP. Entretanto, a maioria dos roteadores conectados à Internet tem apenas um caminho para o backbone e, portanto, pode utilizar uma rota padrão estática simples.

Uma boa estratégia de roteamento para instalações de porte médio com estrutura local relativamente estável e conexão com a rede de qualquer outro usuário é utilizar uma combinação de roteamento dinâmico e estático. Máquinas dentro da estrutura local que não possuem um gateway para redes externas podem utilizar roteamento estático, encaminhando todos os pacotes desconhecidos para uma máquina padrão que compreenda o mundo externo e realize o roteamento dinâmico.

Uma rede que é muito complicada de ser gerenciada com esse esquema deveria fiar-se no roteamento dinâmico. Rotas estáticas padrão ainda podem ser utilizadas em redes das extremidades, mas máquinas em redes com mais de um roteador devem executar **routed** no modo passivo.

13.7 ROTEADORES CISCO

Os roteadores produzidos pela Cisco Systems, Inc., são um padrão de fato para roteamento Internet hoje em dia. Tendo capturado mais de 70% do mercado de roteadores, os produtos da Cisco são bem conhecidos e pessoal que saiba operá-los é relativamente fácil de ser encontrado. Antes da Cisco, os computadores UNIX com várias interfaces de rede eram freqüentemente utilizados como roteadores. Hoje, os roteadores são o mecanismo preferido para colocação em gabinetes de comunicação de dados e por cima de forros onde os cabos se juntam. Eles são mais baratos, mais rápidos e mais seguros do que seus equivalentes UNIX ou Linux.

A maioria dos produtos para roteamento da Cisco roda em um sistema operacional chamado Cisco IOS, que é proprietário e não-relacionado com o Linux. Seu conjunto de comandos é bem grande; a documentação completa chega a ocupar 1,5 m de espaço em prateleira. Jamais poderíamos cobrir completamente o Cisco IOS aqui, mas conhecer alguns fundamentos pode levá-lo para bem longe.

O IOS define dois níveis de acesso (usuário e privilegiado), ambos os quais são protegidos por senha. Por padrão, podemos simplesmente enviar um **telnet** para um roteador Cisco para entrar no modo usuário.[4] A senha de acesso em nível de usuário será solicitada:

```
$ telnet acme-gw.acme.com
Connected to acme-gw.acme.com.
Escape character is '^]'.

User Access Verification
Password:
```

Depois de introduzir a senha correta, você receberá um prompt do interpretador de comandos EXEC da Cisco.

```
acme-gw.acme.com>
```

Nesse prompt, você pode introduzir comandos como **show interfaces** para ver as interfaces de rede do roteador ou **show?** para obter ajuda sobre as demais coisas que podem ser vistas.

Para entrar no modo privilegiado, digite **enable** e introduza a senha privilegiada quando ela for solicitada. Assim que tiver atingido o nível privilegiado, seu prompt terminará em um #:

```
acme-gw.acme.com#
```

CUIDADO — você poderá fazer qualquer coisa com esse prompt, inclusive apagar as informações de configuração do roteador e seu sistema operacional. Quando estiver em dúvida, consulte os manuais da Cisco ou um dos abrangentes livros publicados pela Cisco Press.

Você poderá digitar **show running** para ver a configuração atual em execução do roteador e **show config** para visualizar a configuração não-volátil atual. Na maior parte do tempo, esses dados são iguais.

Eis uma configuração típica:

```
acme-gw.acme.com# show running
Current configuration:
version 12.1
hostname acme-gw
enable secret xxxxxxxx
ip subnet-zero

interface Ethernet0
 description Acme internal network
 ip address 192.108.21.254 255.255.255.0
 no ip directed-broadcast
interface Ethernet1
 description Acme backbone network
 ip address 192.225.33.254 255.255.255.0
 no ip directed-broadcast

ip classless
line con 0
```

4 Uma série de métodos de acesso pode ser configurada. Se sua instalação já utiliza roteadores Cisco, entre em contato com o administrador da sua rede para descobrir que métodos foram habilitados.

```
transport input none
line aux 0
 transport input telnet
line vty 0 4
 password xxxxxxxx
 login

end
```

A configuração do roteador pode ser modificada de várias maneiras. A Cisco oferece ferramentas gráficas que são executadas em algumas versões do UNIX/Linux e do Windows. Verdadeiros administradores de redes jamais as utilizam; o prompt de comando é sempre a "aposta segura". Também é possível transferir um arquivo de configuração via **tftp** de ou para um roteador para você poder editá-lo em seu editor preferido.[5]

Para modificar a configuração por meio do prompt de comando, digite **config term**:

```
acme-gw.acme.com# config term
Enter configuration commands, one per line.  End with CNTL/Z.
acme-gw(config)#
```

A seguir, você pode digitar novos comandos de configuração exatamente como quer que eles apareçam na saída de **show running**. Por exemplo, se você quisesse alterar o endereço IP da interface Ethernet0 do exemplo acima, poderíamos digitar o seguinte:

```
interface Ethernet0
ip address 192.225.40.253 255.255.255.0
```

Ao terminar de introduzir comandos de configuração, pressione <Control-Z> para retornar para o prompt de comando normal. Se estiver satisfeito com a nova configuração, digite **write mem** para salvar a configuração em memória não-volátil.

Eis algumas dicas para ser bem-sucedido no uso de roteadores Cisco:

- Atribua um nome ao roteador por meio do comando **hostname**. Essa precaução ajuda a impedir acidentes causados por alterações de configuração no roteador errado. O nome do host sempre aparece no prompt de comandos.
- Sempre mantenha uma configuração de backup do roteador à mão. Você pode criar um pequeno script **expect** que transfere via **tftp** a configuração em execução para um computador Linux a cada noite por segurança.
- Controle o acesso à linha de comandos do roteador colocando listas de acesso nas VTYs do roteador (VTYs são como PTYs em um computador Linux). Essa precaução evita que pessoal não-autorizado tente invadir o seu roteador.[6]
- Controle o tráfego que flui entre suas redes (e, possivelmente, para o mundo externo) com listas de acesso em cada interface. Consulte a seção *Firewalls para filtragem de pacotes* na página 484 para informações adicionais sobre como configurar listas de acesso.
- Mantenha os roteadores fisicamente seguros. É fácil reinicializar a senha privilegiada se você tiver acesso físico a um equipamento Cisco.

13.8 LEITURA RECOMENDADA

PERLMAN, RADIA. *Interconnections:* Bridges, Routers, Switches, and Internetworking Protocols. *2. ed.* Reading, MA: Addison-Wesley, 2000.

Esse é o trabalho definitivo sobre esse tópico. Se comprar apenas um livro sobre fundamentos de rede, deve ser esse. Além disso, nunca perca uma oportunidade de falar com a Radia — ela é muito divertida e tem um espantoso conhecimento sobre o assunto.

HUITEMA, CHRISTIAN. *Routing in the Internet. 2. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2000.

Esse livro é uma introdução clara e bem escrita ao roteamento, dos princípios básicos aos avançados. Ele cobre a maioria dos protocolos mais comumente utilizados e também alguns tópicos avançados como multicasting.

MOY, JOHN T. *OSPF:* Anatomy of an Internet Routing Protocol. Reading, MA: Addison-Wesley, 1998.

Uma exposição completa sobre o OSPF pelo autor do padrão desse protocolo.

---

5 Dica quente: o Microsoft Word não é a melhor opção para essa aplicação.

6 Versões modernas do IOS suportam o protocolo SSH. Você deve utilizar isso em vez da interface TELNET padrão se ela estiver disponível no ambiente.

STEWART, JOHN W. *BGP4 Inter-domain Routing in the Internet*. Reading, MA: Addison-Wesley, 1999.

Há muitas RFCs relacionadas a roteamento. As principais são mostradas na Tabela 13.3.

Tabela 13.3 RFCs relacionadas com roteamento

RFC	Título	Autores
2328	OSPF Version 2	John T. Moy
1058	Routing Information Protocol	C. Hedrick
2453	RIP Version 2	Gary Scott Malkin
1256	ICMP Router Discovery Messages	Stephen E. Deering
1142	OSI IS-IS Intra-domain Routing Protocol	David R. Oran
1075	Distance Vector Multicast Routing Protocol	D. Waitzman et al.
4632	CIDR: an Address Assignment and Aggregation Strategy	Vince Fuller et al.
4271	A Border Gateway Protocol 4 (BGP-4)	Yakov Rekhter et al.

13.9 EXERCÍCIOS

E13.1 Investigue o comando **route** do Linux e escreva uma breve descrição sobre o que ele faz. Utilizando **route**, como você:

a) Adiciona uma rota padrão a 128.138.129.1 utilizando a interface eth1?

b) Elimina uma rota para 128.138.129.1?

c) Determina se uma rota foi adicionada por um programa como **routed** ou um redirecionamento de ICMP? (Note que esse método também funciona com a saída de **netstat -rn**.)

E13.2 Comparando o roteamento estático com o roteamento dinâmico, liste várias vantagens e desvantagens de cada um deles. Descreva situações em que cada um deles seria apropriado e explique por quê.

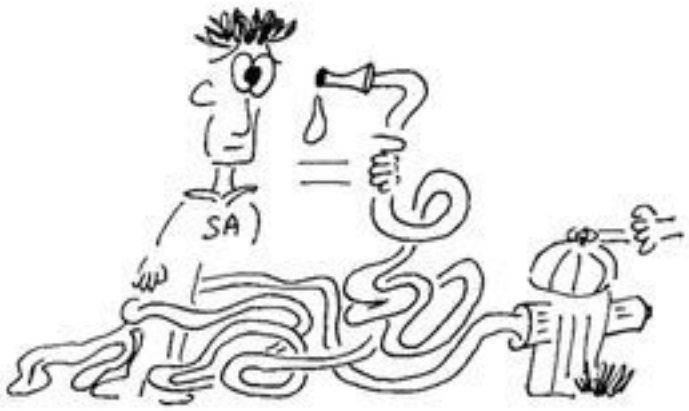
★ E13.3 Considere a saída gerada por **netstat -rn** abaixo. Descreva as rotas e descubra a configuração da rede. Qual rede, 10.0.0.0 ou 10.1.1.0, está mais próxima da Internet? Que processo adicionou cada rota?

```
Destination  Gateway     Genmask       Flags MSS  Window irtt Iface
10.0.0.0     0.0.0.0     255.255.255.0 U      40  0         0 eth1
10.1.1.0     0.0.0.0     255.255.255.0 U      40  0         0 eth0
0.0.0.0      10.0.0.1    0.0.0.0       UG     40  0         0 eth1
```

★★ E13.4 Descubra o esquema de roteamento que é utilizado em sua instalação. Quais protocolos estão em uso? Que máquinas se conectam diretamente à Internet? Você pode utilizar **tcpdump** para procurar pacotes de atualização de roteamento na rede local e **traceroute** para explorar além da rede local. (Exige acesso de root.)

★★ E13.5 Se você fosse um ISP de porte médio que fornecesse contas discadas e hospedagem virtual, que tipo de configuração de roteamento você utilizaria? Certifique-se de considerar não somente o(s) roteador(es) do gateway entre o backbone Internet e sua própria rede, como também quaisquer roteadores internos que possam estar em uso. Desenhe um diagrama da rede que descreva sua arquitetura de roteamento.

14 Hardware de rede



Sejam imagens de vídeo provenientes de um lugar qualquer no planeta ou o som da voz do seu filho em um local específico, praticamente tudo no mundo em que vivemos é tratado de uma forma digital. Mover dados rapidamente de um lugar para outro está na mente de todas as pessoas. Por trás de toda essa loucura temos o hardware de rede elaborado e — adivinhe! — uma quantidade enorme de coisas que se originou nas profundas e escuras cavernas do UNIX. Se há uma área em que a tecnologia UNIX mexeu com a vida das pessoas, essa área é a realização prática do transporte de dados em larga escala via pacotes.

Acompanhar todo esse avanço em ritmo alucinante é um desafio. Naturalmente a velocidade e a confiabilidade da sua rede têm um efeito direto sobre a produtividade da sua empresa, mas as redes atuais são tão penetrantes que seu estado pode afetar nossa capacidade de realizar várias interações humanas básicas, como fazer uma chamada telefônica. Uma rede mal projetada é um constrangimento pessoal e profissional que pode ter efeitos sociais catastróficos. Pode até ser muito caro corrigi-la.

Pelo menos três fatores principais contribuem para uma instalação bem-sucedida:

- Desenvolvimento de um projeto de rede razoável
- Seleção de hardware de alta qualidade
- Documentação e instalação apropriadas
- Competentes operações e manutenções

As primeiras seções deste capítulo discutem as mídias comumente utilizadas para redes locais e de longa distância, inclusive Ethernet, ATM, frame relay, wireless e DSL. A seguir abordamos questões de projeto com as quais você provavelmente irá se deparar em qualquer rede, seja ela nova ou antiga.

14.1 LAN, WAN ou MAN?

De certo modo, temos sorte que o TCP/IP pode ser transportado facilmente em uma série de mídias. Na realidade, porém, o mercado de hardware de rede é dividido em uma série de classificações confusas.

As redes que existem dentro de um prédio ou grupo de prédios geralmente são conhecidas como redes locais (Local Area Networks – LANs). Prevalecem conexões de baixo custo e alta velocidade. As redes de longa distância (Wide Area Networks – WANs) são redes nas quais as extremidades estão geograficamente dispersas, talvez separadas por milhares de quilômetros. Nessas redes, alta velocidade normalmente implica um alto custo, mas praticamente não há limites para as instalações que podemos incluir na rede (Bruges, Bélgica a Sitka, Alasca!). MAN é um termo de marketing de telecomunicações para Metropolitan Area Network, que significa uma mídia de custo moderado e de alta velocidade usada dentro de uma cidade ou de um grupo de cidades. Neste capítulo, exploramos algumas das tecnologias utilizadas para implementar essas feras.

14.2 Ethernet: a LAN comum

Tendo capturado mais de 90% do mercado mundial de LANs, a Ethernet pode ser encontrada praticamente em toda parte em suas várias formas. Ela começou como tese de Ph.D. de Bob Metcalfe no MIT. Bob formou-se e foi para a Xerox PARC; junto com a DEC e a Intel, a Xerox acabou por transformar a Ethernet em um produto. Ela foi um dos primeiros casos em que empresas de informática concorrentes aliaram forças em um projeto técnico.[1]

---

1. Bob Metcalfe também articulou a "Lei de Metcalfe", que afirma que o valor da rede se expande exponencialmente à medida que o número de usuários aumenta.

O padrão Ethernet foi originalmente especificado para uma velocidade de 3 Mb/s (mega*bits* por segundo), porém passou para 10 Mb/s quase imediatamente. Em 1994, a Ethernet chamou atenção à medida que era padronizada a 100 Mb/s. Somente depois de completar 19 anos, em 1998, ela estava pronta para enfrentar uma nova guerra a 1 Gb/s. Agora, madura com mais ou menos 20 anos, a Ethernet está disponível em fibra óptica a 10 Gb/s, tendo ofuscado todos os seus rivais. Um padrão de 10 Gb/s para fio de cobre (802.3an) foi aprovado pelo IEEE em julho de 2006. A Tabela 14.1 a seguir destaca a evolução dos vários padrões Ethernet.[2]

Como a Ethernet funciona

A Ethernet pode ser descrita como um jantar refinado em que os convidados (computadores) não interrompem uns aos outros, mas sim aguardam uma calmaria na conversação (sem tráfego na rede) antes de falarem. Se dois convidados começarem a conversar ao mesmo tempo (uma colisão), ambos param, se desculpam, aguardam um pouco e depois um deles começa a falar novamente.

O termo técnico para esse esquema é CSMA/CD:

- Carrier Sense: você pode dizer se alguém está ou não conversando
- Multiple Access: todo o mundo pode conversar
- Collision Detection: você sabe quando interrompe alguém

Tabela 14.1 A evolução da Ethernet

Ano	Velocidade	Nome comum	IEEE#	Distância	Mídia[a]
1973	3 Mb/s	Xerox Ethernet	–	?	Coaxial
1976	10 Mb/s	Ethernet 1	–	500 m	Coaxial RG-11
1982	10 Mb/s	DIX Ethernet (Ethernet II)	–	500 m	Coaxial RG-11
1985	10 Mb/s	10 Base5 ("Thicknet")	802.3	500 m	Coaxial RG-11
1985	10 Mb/s	10 Base2 ("Thinnet")	802.3	180 m	Coaxial RG-58
1989	10 Mb/s	10 BaseT	802.3	100 m	Cobre UTP categoria 3
1993	10 Mb/s	10 BaseF	802.3	2 km	Fibra MM
				25 km	Fibra SM
1994	100 Mb/s	100 BaseTX ("100 meg")	802.3u	100 m	Cobre UTP categoria 5
1994	100 Mb/s	100 BaseFX	802.3u	2 km	Fibra MM
				20 km	Fibra SM
1998	1 Gb/s	1000 BaseSX	802.3z	260 m	Fibra MM 62,5-µm
				550 m	Fibra MM 50-µm
1998	1 Gb/s	1000 BaseLX	802.3z	440 m	Fibra MM 62,5-µm
				550 m	Fibra MM 50-µm
				3 km	Fibra SM
1998	1 Gb/s	1000 BaseCX	802.3z	25 m	Twinax
1999	1 Gb/s	1000 BaseT ("Gigabit")	802.3ab	100 m	Cobre UTP cat 5E e 6
2002	10 Gb/s	10 GBase-SR	802.3ae	300 m	Fibra MM
		10 GBase-LR		10 km	Fibra SM
2006	10 Gb/s	10 GBase-T	802.3an	100 m	Cobre UTP categoria 7
2006[b]	100 Gb/s	TBD	TBD	TBD	Fibra
2008[b]	1 Tb/s	TBD	TBD	TBD	Fibra CWDM
2010[b]	10 Tb/s	TBD	TBD	TBD	Fibra DWDM

a. MM = Multimode, SM = Single-mode, UTP = Unshielded twisted pair, CWDM = Coarse wavelength division multiplexing, DWDM = Dense wavelength division multiplexing.

b. Projeção da indústria.

2. Omitimos alguns padrões Ethernet que não se tornaram populares, como o 100BaseT4 e o 100BaseVG-AnyLAN.

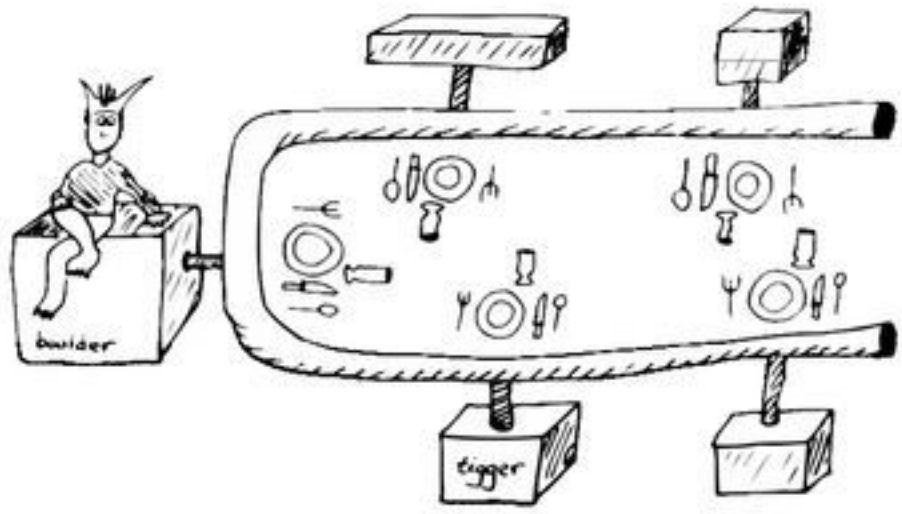
O atraso real após a detecção de colisão é um tanto aleatório. Essa convenção impede a situação na qual dois hosts transmitem simultaneamente para a rede, detectam a colisão, aguardam a mesma quantidade de tempo e depois começam a transmitir novamente, inundando, conseqüentemente, a rede com colisões. Isso nem sempre é verdade!

Topologia Ethernet

A topologia Ethernet é um barramento ramificado sem loops; existe apenas uma maneira de um pacote percorrer um caminho entre dois hosts quaisquer na mesma rede. Três tipos de pacotes podem ser trocados em um segmento: unicast, multicast e broadcast. Pacotes unicast são destinados apenas a um único host. Os pacotes multicast são destinados a um grupo de hosts. Os pacotes de broadcast são entregues a todos os hosts de um segmento.

Um "domínio de broadcast" é o conjunto de hosts que recebe pacotes destinados para o endereço de broadcast de hardware e há exatamente um domínio de broadcast para cada segmento Ethernet. Nos primeiros padrões e mídias Ethernet (como 10Base5), os segmentos físicos e lógicos eram exatamente os mesmos, uma vez que todos os pacotes trafegavam em um único grande cabo com as interfaces de hosts conectadas na lateral dele.[3]

Ilustração A Um refinado "jantar" Ethernet



Com o advento dos switches, os segmentos lógicos atuais normalmente consistem em muitos (possivelmente dezenas ou centenas) segmentos físicos (ou, em alguns casos, segmentos sem fio) aos quais apenas dois dispositivos são conectados: a porta do switch e o host. Os switches são responsáveis por acompanharem pacotes multicast e unicast até os segmentos físicos nos quais os destinatários pretendidos residem; o tráfego de broadcast é encaminhado para todas as portas em um segmento lógico.

Um único segmento lógico pode ser formado de segmentos físicos (ou sem fio) operando em velocidades diversas (10 Mb/s, 100 Mb/s, 1 Gb/s ou 10 Gb/s); portanto, os switches devem ter recursos de bufferinge timing para eliminar possíveis conflitos.

Par trançado não blindado

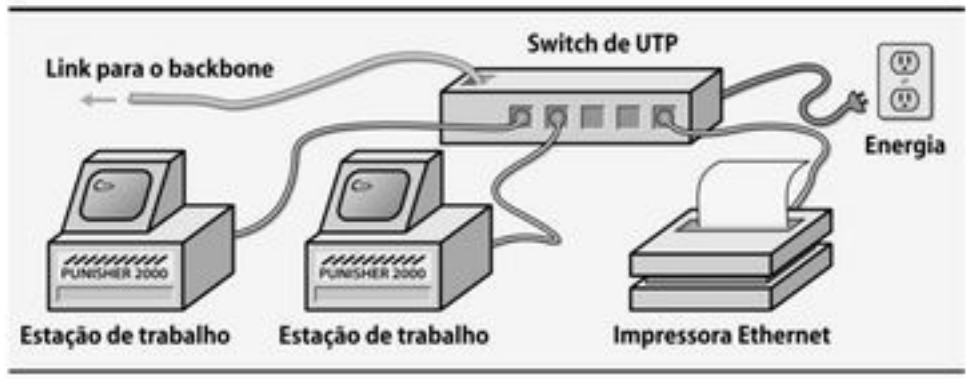
O par trançado não blindado (*unshielded twisted pair* – UTP) é o tipo de mídia de cabo preferida para o padrão Ethernet. Ele se baseia em uma topologia estrela e possui várias vantagens em relação a outras mídias:

- Ele usa fio de cobre barato e fácil de encontrar. (Algumas vezes, pode ser usada a fiação telefônica existente.)
- A fiação UTP é muito mais fácil de instalar e depurar do que fibra ou cabo coaxial. Comprimentos sob medida são obtidos facilmente
- O UTP usa conectores RJ-45, que são baratos, confiáveis e fáceis de serem instalados
- O link com cada máquina é independente (e privado!), de modo que um problema de cabeamento em um link dificilmente afetará outros hosts da rede

O "formato" genérico de uma rede UTP é demonstrado na Ilustração B

3. Não é brincadeira! Conectar um novo computador envolvia o monótono processo de fazer um furo no revestimento externo do cabo com uma broca para se poder atingir o condutor central. Um "mordedor-vampiro", que morde o condutor externo, era então fixado via parafusos.

Ilustração B Uma instalação UTP



A fiação UTP adequada para uso em LANs modernas é comumente subdividida em oito classes. O sistema de avaliação de desempenho foi introduzido pela primeira vez pela Anixter, um grande fabricante de cabos. Esses padrões foram formalizados pela TIA (Telecommunications Industry Association) e são conhecidos hoje como Categorias 1 a 7, com uma Categoria 5E especial no meio.

A ISO (International Organization for Standardization) também entrou no excitante e rentável mundo da classificação de cabos e promove padrões exatamente equivalentes ou aproximadamente equivalentes às categorias TIA com numeração mais alta. Por exemplo, o cabo Categoria 5 da TIA é equivalente à Classe D da ISO. Para os *geeks* de plantão, a Tabela 14.2 ilustra as principais diferenças entre as diversas classificações modernas de hoje em dia. São informações interessantes de serem memorizadas para você poder impressionar seus amigos em festas.

Tabela 14.2 Características de cabos UTP

Parâmetro[a]	Categoria 5 Classe D[b]	Categoria 5E	Categoria 6 Classe E	Categoria 7 Classe F
Intervalo de freqüência	100 MHz	100 MHz	250 MHz	600 MHz
Atenuação	24 dB	24 dB	21 .7 dB	20 .8 dB
NEXT	27 .1 dB	30 .1 dB	39 .9 dB	62 .1 dB
ACR	3 .1 dB	6 .1 dB	18 .2 dB	41 .3 dB
ELFEXT	17 dB	17 .4 dB	23 .2 dB	?[c]
Perda de retorno	8 dB	10 dB	12 dB	14 .1 dB
Atraso de propagação	548 ns	548 ns	548 ns	504 ns

a. NEXT = Near-end crosstalk, ACR = Attenuation-to-crosstalk ratio, ELFEXT = Equal level far-end xtalk.

b. Inclui os requisitos TIA e ISO adicionais, TSB95 e FDAM 2, respectivamente.

c. Atualmente não especificado até mais estudos.

Na prática, os cabos Categoria 1 e 2 são adequados apenas para aplicações de voz (se tanto). Cabo Categoria 3 é o mínimo exigido para uma LAN; ele é o padrão para redes 10BaseT a 10 Mb/s. Cabo Categoria 4 é uma espécie de órfão, não adequado exatamente a nenhum tipo de aplicação em particular. Ele é usado ocasionalmente para redes Token Ring UTP a 16 Mb/s ou para instalações 10BaseT peculiares. Cabo Categoria 5 é capaz de suportar 100 Mb/s e é o padrão mais comumente usado hoje para cabeamento de dados. Cabeamento Categoria 5E e Categoria 6 suportam 1 GB/s. A Categoria 7 destina-se ao uso de 10 GB/s assim que o padrão de cobre sobre Ethernet de 10 Gb/s estiver pronto.

Consulte a página 254 para informações adicionais sobre cabeamento.

Conexões 10BaseT exigem dois pares de cabos Categoria 3, e cada link está limitado a um comprimento de 100 metros; conexões 100BaseTX têm a mesma limitação de comprimento, mas requerem dois pares de cabos Categoria 5. Há cabos com revestimento de PVC ou Teflon. Sua opção de revestimento deve se basear no ambiente no qual o cabo será instalado. Tubulações fechadas que passam pelo sistema de ventilação do prédio ("sistemas de retorno de ar") exigem, normalmente, Teflon.[4] PVC é mais barato e mais fácil de trabalhar.

4. Fale com a brigada de incêndio de seu prédio para determinar as exigências de sua instalação.

Conectores RJ-45 interligados com os pinos 1, 2, 3 e 6 são utilizados para estabelecer as conexões. Embora somente dois pares de fios sejam necessários para uma conexão funcional a 10 Mb/s ou 100 Mb/s, recomendamos que, ao instalar uma nova rede, você use fiação de quatro pares Categoria 5E e conecte todos os oito pinos do conector RJ-45.

Consulte a página 579 para informações adicionais sobre o padrão R-232.

Para terminação de cabo UTP de quatro pares em painéis de interligação e tomadas de parede RJ-45, sugerimos que você utilize o padrão de fiação TIA/EIA-568A RJ-45. Esse padrão, que é compatível com outros empregos do RJ-45 (por exemplo, RS-232), é uma maneira conveniente de manter consistente a fiação em ambas as extremidades da conexão, independentemente de você poder ou não acessar facilmente os pares de cabos em si. O padrão 568A é detalhado na Tabela 14.3.

Tabela 14.3 Padrão TIA/EIA-568A para fiação UTP de quatro pares para um conector RJ-45

Par	Cores	Conectados aos	Par	Cores	Conectados aos
1	Branco/Azul	Pinos 5/4	3	Branco/Verde	Pinos 1/2
2	Branco/Laranja	Pinos 3/6	4	Branco/Marrom	Pinos 7/8

Fiação já existente no prédio pode ou não ser adequada para emprego em redes, dependendo de como e quando ela foi instalada. Muitos prédios antigos foram reformados com cabos novos nos anos 50 e 60. Infelizmente, esse tipo de cabo normalmente não oferecerá suporte nem mesmo a uma velocidade de 10 Mb/s.

Conectando e expandindo redes Ethernet

As redes Ethernet podem estar conectadas logicamente em diversos pontos no modelo de rede OSI de sete camadas. Na camada 1, a camada física, você pode usar tanto conectores quanto repetidores físicos (comumente denominados hubs nos tempos modernos). Eles transferem o sinal diretamente, de forma muito parecida a duas latas conectadas por um barbante.

Na camada 2, a camada de link de dados, são utilizados switches. Comuta frames de transferência de acordo com a origem de hardware e com o destino dos endereços, quase como entregar uma mensagem em uma garrafa mostrando somente o rótulo na parte externa da garrafa.

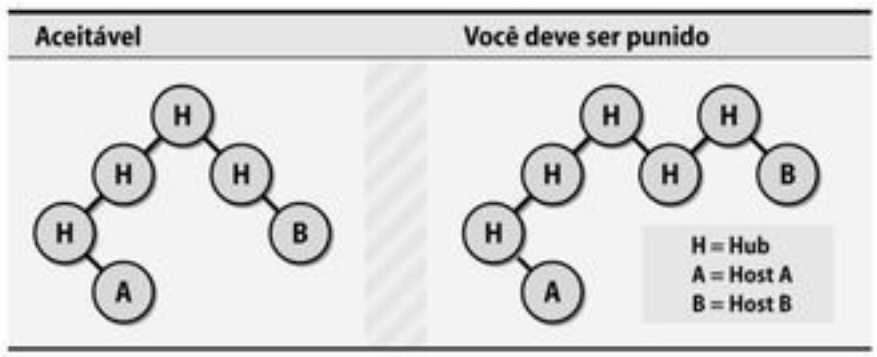
Na camada 3, a camada de rede, são utilizados roteadores. Os roteadores transferem mensagens para o próximo salto de acordo com a localização do receptor final, mais exatamente como ler a mensagem dentro de uma garrafa para determinar a quem ela realmente foi destinada.

Hubs

Hubs (também conhecidos como concentradores) são dispositivos ativos que conectam segmentos físicos em redes Ethernet UTP. Eles requerem alimentação de energia externa. Atuando como um repetidor, um hub ajusta novamente o tempo e reconstitui frames Ethernet, mas não os interpreta; ele não tem a mínima idéia para onde os pacotes estão indo ou que protocolo eles estão usando.

Os dois pontos mais distantes da rede nunca devem estar separados por mais de quatro hubs. O padrão Ethernet versões 1 e 2 especificavam no máximo dois hubs em série por rede. O padrão IEEE 802.3 estendeu esse limite para quatro em redes Ethernets de 10 Mb/s. Redes Ethernet de 100 Mb/s permitem o emprego de dois hubs, redes Ethernet 1000BaseT apenas um e redes 10Gb/s não permitem o emprego de hubs. A Ilustração C ilustra tanto uma configuração correta quanto uma incorreta para uma rede de 10 Mb/s.

Ilustração C Contagem de hubs



Os hubs exigem ocasionalmente atenção por parte de um administrador de sistemas, de modo que não devem ficar em lugares pouco iluminados e de difícil alcance. Reciclar a alimentação de energia (ou seja, desligando e ligando novamente) em geral permite que eles se recuperem de um estado em que ocorreu travamento.

Switches

Os switches conectam redes Ethernet na camada de link de dados (camada 2) do modelo OSI. O propósito deles é o de conectar duas redes físicas diversas de uma maneira em que elas se pareçam com uma grande rede física. Switches são o padrão da indústria para conectar dispositivos Ethernet hoje em dia.

Switches recebem, regeneram e retransmitem pacotes no hardware.[5] A maioria dos switches utiliza um algoritmo de aprendizagem dinâmica. Eles percebem quais endereços de origem provêm de uma determinada porta e quais provêm de uma outra. Eles encaminham pacotes entre portas somente quando necessário. Inicialmente todos os pacotes são encaminhados, mas em poucos segundos o switch já toma conhecimento das localizações da maioria dos hosts e poderá ser mais seletivo.

Como nem todos os pacotes são encaminhados entre redes, cada segmento do cabo fica menos saturado com tráfego do que ficaria se todas as máquinas estivessem no mesmo cabo. Dada que a maior parte da comunicação tende a ser localizada, o aumento na largura de banda aparente pode ser dramático. E como o modelo de rede não é afetado pela presença de um switch, há poucas conseqüências administrativas na instalação de um.

Os switches podem, algumas vezes, ficar confusos caso a rede contenha loops. A confusão é originada pelo fato de os pacotes de um único host aparecerem em duas (ou mais) portas do switch. Uma rede Ethernet simples não pode conter loops, mas como você conecta várias redes Ethernet com roteadores e switches, a topologia poderá incluir vários caminhos para um host. Alguns switches são capazes de lidar com essa situação reservando rotas alternativas no caso de a rota principal apresentar problemas. Eles executam uma operação de expurgo na rede até que as seções remanescentes apresentem somente um caminho para cada nó da rede. Alguns switches também são capazes de duplicar links entre as mesmas duas redes, e o tráfego de rotas é feito seguindo um modelo "round robin".

Os switches estão cada vez mais inteligentes à medida que mais recursos são embutidos em seus firmwares. Alguns podem ser utilizados para monitorar a segurança da rede. Eles registram qualquer endereço Ethernet externo que virem, conseqüentemente, detectando e relatando máquinas recém-conectadas. Como operam na camada Ethernet, os switches são independentes de protocolo e podem lidar com qualquer combinação de tipos de pacotes de alto nível (por exemplo, IP, AppleTalk ou NetBEUI).

Os switches têm de varrer cada pacote para determinar se devem ser encaminhados. O desempenho dos switches é normalmente medido tanto pela taxa de varredura de pacotes quanto pela taxa de encaminhamento de pacotes. Muitos fornecedores não mencionam os tamanhos dos pacotes nos números que divulgam; conseqüentemente, o desempenho real pode ser menor do que aquele anunciado.

Embora o hardware de switches Ethernet esteja se tornando cada vez mais veloz, ele ainda não é uma tecnologia razoável para conexão de mais do que uma centena de hosts em um único segmento lógico. Problemas como "tempestades de broadcast" normalmente afetam redes com muitos switches, já que o tráfego de broadcast deve ser encaminhado a todas as portas em um segmento com switch. Para solucionar esse problema, use um roteador para isolar o tráfego de broadcast entre segmentos com switches, criando, desse modo, mais de uma rede Ethernet lógica.

Grandes instalações podem se beneficiar dos switches capazes de repartir suas portas (por meio de configuração de software) em subgrupos chamados VLANs (Virtual Local Area Networks). Uma VLAN é um grupo de portas que pertence ao mesmo segmento lógico, como se as portas estivessem conectadas aos seus próprios switches dedicados. Tal repartição aumenta a capacidade do switch de isolar tráfego, e essa capacidade possui efeitos benéficos tanto em termos de segurança quanto de desempenho.

O tráfego entre VLANs é tratado por um roteador ou, em alguns casos, por um módulo de roteamento ou camada de software de roteamento dentro do switch. Uma extensão desse sistema conhecida como "entroncamento VLAN" (como a especificada pelo protocolo IEEE 802.1Q) permite que switches separados fisicamente sirvam a portas na mesma VLAN lógica.

Escolher um switch pode ser difícil. O mercado de switches é um segmento altamente competitivo da indústria de informática e é infestado por anúncios de marketing que não são nem mesmo parcialmente verdadeiros. Ao selecionar um fornecedor de switches, você deve contar com avaliações independentes (como aqueles que aparecem nas comparações em uma revista) em vez dos dados fornecidos pelos próprios fornecedores. Em anos recentes, tem sido comum para um fabri-

5. Pelo fato de os pacotes serem regenerados e recalculados em termos de tempo, as redes totalmente comutadas não sofrem das limitações de "contagem de repetidores" exibidas na Ilustração C.

cante ter o "melhor" produto por alguns meses, e depois destruir completamente seu desempenho ou confiabilidade ao tentar realizar melhorias, fazendo com que um outro concorrente assuma a liderança.

De qualquer modo, certifique-se de que a velocidade do backplane do switch é adequada — esse é o número que é efetivamente considerado no final de um longo dia. Um switch bem projetado deve ter uma velocidade de backplane que exceda a soma das velocidades de todas as suas portas.

Roteadores

Os roteadores são computadores dedicados em um gabinete contendo duas ou mais interfaces de rede; eles direcionam tráfego na Camada 3 da pilha de protocolo OSI (a camada de rede). Eles enviam pacotes a seus destinos finais de acordo com as informações contidas nos cabeçalhos de protocolos TCP/IP. Além de simplesmente transferirem pacotes de um lugar para outro, os roteadores também podem executar outras funções como filtragem de pacotes (por questões de segurança), priorização (por questões de qualidade de serviço) e descoberta de topologia de rede como um todo. Veja os detalhes de como realmente funciona o roteamento no Capítulo 13, *Roteamento*.

Em um único roteador podemos encontrar interfaces de hardware de vários tipos diferentes (por exemplo, SONET, Ethernet e ATM). Em termos de software, alguns roteadores também são capazes de tratar de tráfego não IP como IPX ou AppleTalk. Nessas configurações, o roteador e suas interfaces têm de ser configurados para cada protocolo que você deseja que ele trate. Hoje em dia, geralmente uma boa idéia é evitar esses protocolos de legado e suportar bem apenas o TCP/IP.

Os roteadores assumem uma de duas formas: configuração fixa e modular. Roteadores de configuração fixa têm interfaces de rede específicas de fábrica instaladas permanentemente. Eles normalmente são adequados para aplicativos pequenos e especializados. Por exemplo, um roteador com uma interface T1 e uma interface Ethernet poderia ser uma ótima opção para conexão de uma pequena empresa à Internet.

Os roteadores modulares têm uma arquitetura de barramento ou slots à qual as interfaces podem ser acrescentadas pelo usuário final. Embora essa forma normalmente seja cara, ela garante muito mais flexibilidade no futuro.

Dependendo de suas necessidades de confiabilidade e carga de tráfego esperada, um roteador dedicado pode ou não ser mais barato do que um sistema Linux configurado para atuar como um roteador. Mas o roteador dedicado normalmente resulta em melhor desempenho e confiabilidade. Essa é uma área do projeto de redes na qual normalmente é aconselhável gastar um pouco mais no início para evitar dores de cabeça depois.

14.3 REDES SEM FIO: A LAN DO NÔMADE

Rede sem fio é uma área em crescimento acelerado e há produtos disponíveis a preços razoáveis. Considerando os recentes avanços na tecnologia de redes com fio, as velocidades das redes sem fio (variando normalmente de 2 Mb/s a 54 Mb/s) podem parecer um tanto inadequadas para o uso corporativo. De fato, essas velocidades são perfeitamente boas para muitos propósitos. Uma rede sem fio de 11 Mb/s em uma residência ou pequeno escritório pode ser o sonho de qualquer administrador de sistemas. A 54 Mb/s, uma rede sem fio pode ser aceitável em um ambiente corporativo. Além disso, o acesso sem fio para feiras, aeroportos e outros locais públicos pode realmente transformar um dia em que se fica incomunicável em um dia hiperconectado para várias pessoas.

Os padrões sem fio mais promissores hoje são as especificações IEEE 802.11g e 802.11a. A 802.11g opera na freqüência de banda de 2.4 GHz e fornece acesso do tipo rede local em até 54 Mb/s. Intervalos operacionais variam de 100 metros a 40 quilômetros, dependendo do equipamento e do terreno. A 802.11a também fornece até 54 Mb/s de largura de banda, mas utiliza a freqüência de banda de 5.4 GHz. Alguns equipamentos atuais podem agregar dois canais para fornecer 108 Mb/s de largura de banda. Espera-se que o padrão 802.11n, com boas perspectivas de êxito, forneça mais que 200 Mb/s de largura de banda no final de 2007.

Embora tanto a 802.11g como a 802.11a afirmem operar a 54 Mb/s, a largura de banda pretendida e efetiva dessa especificação pode ser bem diferente. A 802.11g tem por objetivo principalmente o mercado consumidor. Ela é em geral mais barata que a 802.11a e fornece três canais de dados sem sobreposição *versus* doze da 802.11a. Os canais são muito parecidos a pistas em uma rodovia: quanto maior o número de canais disponível, maior o número de clientes que pode perceber sua potencial largura de banda total.

Para pequenas instalações, um desses padrões é provavelmente aceitável. Instalações maiores ou implantações em universidades talvez queiram considerar a 802.11a por causa da sua quantidade maior de espectro. Na realidade, a maioria dos equipamentos de rádio sem fio pode ser utilizada com um dos tipos de rede.

Hoje, redes 802.11b (11 Mb/s), 802.11g e 802.11a são bem comuns. As placas são baratas e disponíveis para (ou embutidas) a maioria dos laptops e PCs desktops. Como ocorre com a Ethernet cabeada, a arquitetura mais comum para uma rede 802.11 utiliza um hub (chamado de "ponto de acesso" na terminologia sem fio) como o ponto de conexão para múltiplos clientes. Os pontos de acesso podem ser conectados a redes com fio tradicionais ou sem fio a outros pontos de acesso (uma configuração conhecida como "malha sem fio" [*wireless mesh*]).

Você pode configurar uma máquina Linux para que ela atue como um ponto de acesso 802.11a/b/g se você tiver o hardware e o driver corretos. Conhecemos pelo menos um chipset que suporta essa configuração, o Intersil Prism II. Uma excelente estação-base sem fio 802.11b/g independente para uso doméstico ou pequenos escritórios é o AirPort Express da Apple, um produto do tipo "wall-wart" que é barato (em torno de US$ 150) e altamente funcional.[6] Compre um hoje!

Há literalmente dezenas de fornecedores comercializando pontos de acesso sem fio. Você pode comprá-los em lojas de informática. Previsivelmente, o ditado "o barato sai caro" se aplica. Pontos de acesso baratos (aqueles na faixa de US$ 50) provavelmente terão um desempenho ruim ao tratar grandes transferências de arquivos ou mais de um cliente ativo.

Depurar uma rede sem fio é algo como magia negra. Você deve considerar um amplo espectro de possíveis variáveis ao lidar com problemas. Se estiver instalando uma rede sem fio em uma escala corporativa, você provavelmente precisará investir em um analisador de rede sem fio. Recomendamos veementemente os produtos de análise fabricados pela AirMagnet.

Segurança sem fio

A segurança das redes sem fio é tradicionalmente muito ruim. O Wired Equivalent Privacy (WEP) é um protocolo usado junto com as redes 802.11b para permitir criptografia de 40 bits, 104 bits ou de 128 bits para pacotes que percorrem ondas aéreas. Infelizmente, esse padrão contém uma falha de projeto fatal que o torna nada mais que um gargalo de velocidade para bisbilhoteiros. Alguém próximo de seu prédio ou casa poderá acessar sua rede diretamente e de modo imperceptível.

Mais recentemente, os padrões de segurança Wi-Fi Protected Access (WPA) trouxeram mais confiança à segurança sem fio. Hoje, o WPA deve ser utilizado em vez do WEP em todas as instalações novas. Sem o WPA, redes sem fio — tanto com o WEP como sem ele — devem ser consideradas completamente inseguras.

O 802.11i, conhecido como WPA2, é uma alternativa mais recente ao WPA. Ele adiciona mais mecanismos de autenticação a redes sem fio corporativas.

Switches sem fio

Quase da mesma maneira como hubs Ethernet evoluíram para tornarem-se switches Ethernet, produtos sem fio estão passando por um aprimoramento gradual para uso em grandes empresas. Diversos fornecedores (como a Airespace) estão produzindo agora "switches sem fio" que funcionam em conjunção com uma série de pontos de acesso instalados por todo um campus. A teoria é que inúmeros pontos de acesso baratos possam ser implantados e então gerenciados centralmente por um switch "inteligente". O switch mantém as informações de configuração das WAPs e suporta facilmente autenticação e roaming.

Se precisar fornecer serviços sem fio por toda uma empresa de pequeno a médio porte, com certeza essa categoria de produtos merece ser avaliada. Esses produtos não só diminuem o tempo de gerenciamento, como também incluem um meio de monitorar e gerenciar a qualidade do serviço entregue aos usuários.

Um truque particularmente interessante é instalar uma rede 802.11a/b/g por toda a instalação e utilizá-la para suportar dispositivos VoIP de mão (tipo *hand-held*) para os funcionários. É como uma rede de celulares gratuita!

14.4 FDDI: UMA LAN DECEPCIONANTE, CARA E ULTRAPASSADA

Em velocidades de 10 Mb/s, as redes Ethernet dos anos 80 não ofereciam largura de banda suficiente para algumas necessidades de rede, como conectar grupos de trabalhos por meio de um backbone corporativo (ou campus). Em uma tentativa de oferecer opções com largura de banda maiores, o comitê ANSI X3T9.5 produziu a FDDI (Fiber Distributed Data Interface) como alternativa a redes Ethernet.[7] Desenvolvida e comercializada como um Token Ring de 100 Mb/s, a FDDI parecia ser uma solução fácil para as necessidades de largura de banda de muitas empresas.

Infelizmente, a FDDI foi uma decepção total e, nesse momento, não deve ser considerada em nenhuma área de produção. Incluímos informações sobre FDDI aqui somente para uma perspectiva histórica (e na eventualidade de você ainda encontrá-la em uso em algum canto obscuro da sua rede).

Consulte a página 196 para mais sobre MTUs.

Para bom desempenho, a FDDI precisa de uma MTU muito maior do que o padrão que é ajustado para redes Ethernet. Um valor de MTU igual a 4.352 (configurado com **ifconfig**) é razoável.

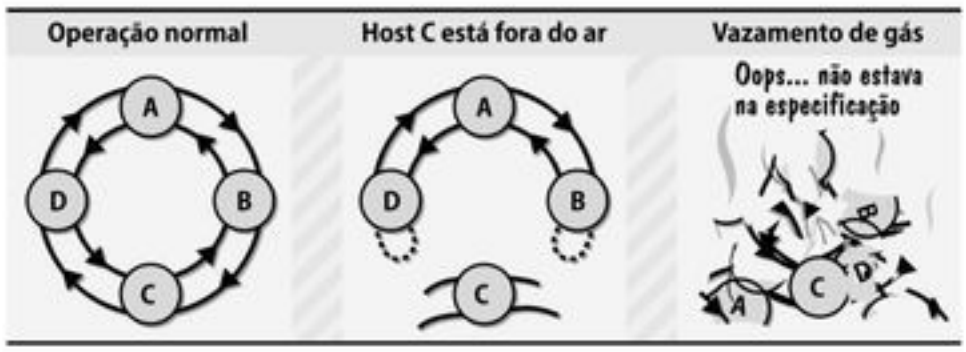
O padrão FDDI especifica uma LAN anel dual usando fibra óptica como meio de transmissão, com passagem de identificadores (tokens) a 100 Mb/s, conforme mostrado na Ilustração D. A arquitetura de anel dual fornece um anel primário que é usado para transmissão de dados e um anel secundário que é utilizado como backup no caso de o anel ser cortado (tanto física como eletronicamente).

Os hosts podem ser conectados a ambos os anéis (eles são então chamados de hosts classe A ou hosts "conectados dualmente") ou simplesmente ao anel primário (hosts classe B ou hosts "conectados de forma simples"). Mais comumente,

6. Na realidade, ele também pode ser conectado ao estéreo para tocar música no seu PC ou laptop sem o uso de fios ou cabos.
7. A FDDI também foi aceita como um padrão ISO.

os concentradores e roteadores de backbone são conectados dualmente e as estações de trabalho conectadas de forma simples, em geral por meio de um "concentrador", uma espécie de hub para fibra óptica.

Ilustração D Token ring dual FDDI



Uma vantagem dos sistemas token ring é que o acesso à rede é controlado por um protocolo determinístico. Não há colisões, então o desempenho da rede não degrada sob carga intensa, como acontece com a Ethernet. Muitos sistemas token ring são capazes de operar de 90% a 95% de sua capacidade calculada ao servir vários clientes.

Para mídia física, o padrão FDDI sugere dois tipos de fibra: modo simples e multimodo. Os "modos" são, basicamente, feixes de raios luminosos que entram na fibra em um determinado ângulo. Fibras modo simples permitem que exatamente uma freqüência de luz percorra seu trajeto e, conseqüentemente, requerem um laser como fonte emissora.[8] Fibras multimodo possibilitam vários trajetos e normalmente utilizam LEDs mais baratos e menos perigosos do que o laser. A fibra modo simples pode ser usada em distâncias maiores do que as fibras multimodo. Na prática, fibras multimodo 62,5 µm são mais comumente utilizadas para FDDI.

Vários padrões de conectores de fibra são utilizados com a FDDI e eles variam de fabricante para fabricante. Independentemente de quais conectores você utiliza, tenha em mente que uma conexão de fibra adequada é essencial para operação confiável. Embora existam disponíveis "kits faça você mesmo" para terminações de fibras, sugerimos que, sempre que possível, você empregue uma empresa especializada em cabeamento para instalar as terminações em trechos de fibra óptica.

14.5 ATM: A LAN PROMETIDA (MAS FRACASSADA)

ATM significa Asynchronous Transfer Mode, mas alguns administradores insistem em denominá-la Another Technical Mistake ("outro erro técnico"). Um porta-voz dos fabricantes do mercado de comunicação de dados a descreveu como "uma tentativa da companhia telefônica de transformar o seu problema de rede em algo que eles sabem como tarifar".

A ATM é tecnicamente "especial", pois promove a filosofia de que pequenos pacotes de tamanho fixo (chamados "células") são a maneira mais eficiente de implementar redes na casa dos gigabits. A ATM também promete recursos que tradicionalmente não foram prometidos por outras mídias, inclusive garantias de qualidade de serviço e reserva de largura de banda.

Acima das células de 53 bytes da ATM, são descritas cinco AALs (ATM Adaptation Layers) para transporte de células. O propósito de cada camada de adaptação está resumido na Tabela 14.4.

Tabela 14.4 Camadas de adaptação ATM

AAL	Aplicativo
1	Aplicativos com taxa de transmissão constante, como voz (requer atraso vinculado)
2	Aplicativos com taxa de transmissão variável que requerem atraso vinculado
3	Aplicativos de dados orientados a conexões
4	Aplicativos de dados sem conexão
5	Transporte de dados genérico (especialmente tráfego IP, substitui 3 e 4)

Não está claro como a AAL 2 seria utilizada na realidade. Atualmente, não existe nenhum padrão definido para ela. As AALs 3 e 4 tornaram-se muito similares e foram combinadas. Um grupo de fabricantes que teve de implementar a ATM

8. A maioria dos lasers (mas não todos) utilizados na rede de fibra óptica são dispositivos da classe 1, o que pode significar "seguro para seus olhos" ou "não seguro para seus olhos, porém, o dispositivo foi projetado para impedir a exposição durante uso normal". Infelizmente, "uso normal" talvez não inclua mexer com cabos cortados, portanto não há realmente nenhuma garantia de segurança. Não aponte o laser para olhos, mesmo se todas as crianças pareçam fazer isso.

estava insatisfeito com as AALs 3 e 4 devido à alta carga de trabalho imposto por elas. Eles desenvolveram uma solução própria, a Simple and Efficient Adaptation Layer (SEAL), que logo se tornou a AAL 5.

A ATM foi amplamente divulgada como uma mídia de rede com switches "tudo em um" que poderia ser utilizada para requisitos de LAN, WAN e MAN. Hoje, a ATM está praticamente morta, preservada apenas em ambientes de rede remota em que grandes empresas de telecomunicações ainda tentam tirar proveito de investimentos mal-aplicados em hardware ATM.

Fornecedores de switch ATM continuam a comercializar agressivamente seus produtos e é possível comprar um circuito ATM em muitas localidades. Entretanto, provavelmente é uma boa idéia considerar outras tecnologias além da ATM para novas implantações de rede.

14.6 FRAME RELAY: A WAN SACRIFICIAL

O frame relay é uma tecnologia WAN que oferece serviços de dados com comutação de pacotes, normalmente por um custo razoável. Embora a afirmação não seja 100% precisa, normalmente diz-se que o frame relay é um X.25 maquiado, uma assustadora tecnologia de comutação de pacotes dos meados dos anos 70. Felizmente, ela é tão amplamente usada que os equipamentos, software e a equipe que fornece suporte evoluíram para um estado de robustez e de bom desempenho.

Tradicionalmente, os usuários que desejavam conectar-se a sites remotos comprariam um circuito dedicado de uma companhia telefônica, por exemplo, uma linha DDS de 56 Kb/s ou uma linha T1. Esses são circuitos de dados ponto a ponto que ficam conectados 24 horas por dia. Infelizmente, esse tipo de conexão é, em geral, caro, uma vez que exige que a companhia telefônica dedique equipamento e largura de banda ao link.

Por outro lado, o frame relay é uma abordagem de "economia de escala". A companhia telefônica cria uma rede (normalmente conhecida como "nuvem"[9]) que estabelece a conexão com suas centrais. Os usuários enviam os dados em pequenos pacotes para sites remotos. A companhia telefônica comuta os pacotes por meio das centrais apropriadas, em última instância entregando-os aos seus respectivos destinos. Nesse modelo, você e a companhia telefônica estão apostando que, a cada segundo, a quantidade total de tráfego não excederá a largura de banda da rede (uma condição conhecida eufemisticamente como "estar subscrita em excesso").

Um roteador encapsula o tráfego IP nas conexões de frame relay. Os pacotes são comutados por meio de "circuitos virtuais permanentes" (*permanent virtual circuits* – PVCs) invisíveis, que permitem que os pacotes trafeguem somente nos sites para os quais você pagou para alcançar. Esses PVCs oferecem algum grau de proteção de privacidade a partir de outros sites conectados à rede frame relay.

A maior vantagem do frame relay é normalmente o seu baixo custo. Mas no mundo do "barato sai caro", você pode descobrir que o desempenho de redes frame relay é, algumas vezes, sofrível. As conexões frame relay têm custo adicional de comutação de pacotes, e a velocidade do link pode degradar em períodos de uso intenso.

14.7 ISDN: A WAN NATIVA

O ISDN (Integrated Services Digital Network) é uma oferta das companhias telefônicas que assume diversas formas. Em sua forma mais comum e útil, chamada de Basic Rate Interface (BRI) ISDN, é essencialmente uma linha telefônica totalmente digital que fornece dois canais "B" discados de 64 Kb/s e um único canal "D" de sinalização de 16 Kb/s. Cada canal B pode ser utilizado tanto para dados como para voz (uma linha de voz pode ser transportada em um único canal de 64 Kb/s).

O ISDN oferece uma linha digital de velocidade relativamente alta a um custo razoável (Nos Estados Unidos, US$ 30–US$ 150 por mês, dependendo de onde se resida). Dispositivos chamados adaptadores de terminais convertem a linha telefônica em uma interface mais familiar como a RS-232. Eles são utilizados (e cobrados) de maneira muito parecida com os modems. A maioria dos adaptadores pode agregar os dois canais B, levando um canal de dados de 128 Kb/s.

O ISDN pode ser utilizado no lugar da rede discada comum e também como tecnologia de rede WAN que utiliza um roteador ou ponte para conectar sites remotos por meio da linha.

Embora muitas companhias telefônicas norte-americanas tenham instalado switches compatíveis com ISDN, elas ainda não descobriram como comercializá-los ou suportá-los.[10] Apenas em poucas regiões podemos ligar para a companhia telefônica e solicitar uma linha ISDN. Algumas dicas: certifique-se de tratar com o departamento da companhia telefônica que cuida de serviços comerciais, uma vez que é desta maneira como o ISDN é classificado no momento. Em muitas regiões você terá de vencer várias ondas de parasitas antes de conseguir falar com alguém que conheça ISDN, mesmo se o serviço estiver realmente disponível.

---

9. Um nome bastante apropriado — nunca fica muito claro qual será a previsão do tempo em uma rede de frame relay. Tempestade? Chuva? Neve? Granizo?
10. Daí a seguinte interpretação: It Still Does Nothing, ou seja, ele ainda não faz nada.

14.8 DSL E MODEMS A CABO: A WAN PÚBLICA

É fácil transferir grandes quantidades de dados entre empresas e outras grandes instalações de dados. Tecnologias oferecidas por companhias telefônicas como T1, T3, SONET, ATM e frame relay fornecem canais relativamente simples para transferência de dados de um lugar para o outro. Entretanto, essas tecnologias não são opções realistas para conectar residências e escritórios domésticos. Elas custam muito e a infra-estrutura necessária não se encontra largamente disponível.

O Digital Subscriber Line (DSL) utiliza cabos telefônicos de cobre comuns para transmissão de dados a velocidades de até 7 Mb/s (embora conexões DSL típicas levem a velocidades entre 256 Kb/s e 3 Kb/s). Uma vez que a maioria das residências já tem cabos telefônicos instalados, o DSL é uma maneira de completar a "última milha" de conectividade da companhia telefônica até a residência. As conexões DSL normalmente terminam em uma "caixa" que atua como um roteador TCP/IP e conecta-se a outros dispositivos dentro da residência em uma rede Ethernet. O DSL é normalmente mais barato e mais rápido do que o ISDN, de modo que se constitui hoje na tecnologia preferida para usuários domésticos.

Diferentemente de conexões POTS (Plain Old Telephone Service) comuns e conexões ISDN, que requerem que você "disque" para um destino final, a maioria das implementações DSL fornece um serviço de rede dedicado que está sempre conectado. Esse recurso torna a DSL ainda mais atraente, pois não há nenhuma configuração ou atraso de conexão quando um usuário quer transferir dados.

A DSL tem várias formas e como resultado freqüentemente é conhecida como xDSL, com o x representando uma subtecnologia específica, por exemplo, A para assimétrico, S para simétrico, H para velocidade alta, RA para taxa adaptativa e I para DSL-over-ISDN (útil para localizações muito distantes da estação central para suportar formas mais rápidas da DSL). As variantes de tecnologia e taxas de transferência de dados exatas disponíveis em sua região dependem dos equipamentos que as companhias telefônicas optaram por empregar.

A corrida pela conectividade da "última milha" para centenas de milhões de residências é uma corrida disputada. Ela é também altamente politizada, bem capitalizada e superdivulgada. A abordagem DSL alavanca a infra-estrutura de cobre que é comum entre os Incumbent Local Exchange Carriers (ILECs), que preferiram altas margens de lucro em vez de investimentos em infra-estrutura conforme a revolução das redes dos anos 80 e 90 passou por elas.

As empresas de TV a cabo, que já têm infra-estrutura de fibra óptica na maioria dos bairros, estão promovendo suas próprias soluções de "última milha", que entregam conexões de banda larga similares (embora assimétricas) para as residências. A indústria de modems a cabo recentemente se tornou esclarecida sobre padrões de dados e está promovendo no momento o padrão Data Over Cable Service Interface Specification (DOCSIS). Esse padrão define as especificações técnicas tanto para modems a cabo quanto para os equipamentos utilizados na empresa de TV a cabo e permite que várias marcas de equipamentos operem entre si.

Em suma, a briga entre as tecnologias de modem a cabo e DSL em grande parte se resume a "meu orçamento para marketing é maior do que o seu". A DSL tem uma vantagem inerente pelo fato de que, na maioria dos casos, cada conexão é privada para o cliente particular; modems a cabo em uma largura de banda vizinhança compartilham largura de banda e às vezes podem bisbilhotar o tráfego de uma ou de outra.

14.9 PARA ONDE AS REDES ESTÃO CAMINHANDO?

Ao observar de perto as tecnologias descritas até aqui, você verá uma coisa em comum: as mais simples e mais baratas estão sendo bem-sucedidas, enquanto as mais complexas e mais caras estão morrendo rapidamente. Onde isso nos coloca no futuro?

O padrão Ethernet deu uma surra em seus rivais, pois ele é incrivelmente barato. Sua implementação é tão simples que hoje podemos até comprar fornos de microonda com interfaces Ethernet. O padrão Ethernet cresceu bem: em muitas empresas, a infra-estrutura Ethernet de 10 Mb/s do início dos anos 80 ainda está em uso produtivamente conectada a trechos de 100 Mb/s e de 1 Gb/s. Ethernet de 10 GB/s em fibra já está disponível e em breve veremos seu uso disseminado nos cabos de cobre. Esperamos que essa tendência continue, com hardwares de switch cada vez mais rápidos para conectar isso tudo.

No front da "conectividade residencial", a DSL dá novo ânimo às velhas e cansadas instalações de cabos de cobre. A proliferação de modems a cabo trouxe acesso de alta velocidade (e sérios problemas de segurança) ao alcance de milhões de residências.

O que é fantástico em relação a todos esses novos desenvolvimentos é que independentemente da mídia ou de sua velocidade, o TCP/IP é compatível com eles.

14.10 TESTE E DEPURAÇÃO DE REDES

A principal vantagem da migração em larga escala para a Ethernet (e outras tecnologias baseadas em UTP) é a facilidade de depuração das redes. Como essas redes podem ser analisadas link por link, problemas de rede normalmente podem ser isolados em segundos em vez de dias.

O segredo para depurar uma rede é subdividi-la em suas partes componentes e testar cada parte até isolar o dispositivo ou cabo problemático. As "luzes idiotas" dos switches e hubs (como as de "status de link" e as de "tráfego de pacotes") normalmente nos dão dicas imediatas para chegar à fonte do problema. Documentação de primeira categoria sobre seus esquemas de fiação é essencial para fazer com que essas luzes indicadoras trabalhem a seu favor.

Como acontece com a maioria das tarefas, ter as ferramentas corretas para o serviço é meio caminho andado para concluir o trabalho de forma correta e sem atrasos. O mercado oferece dois tipos principais de ferramentas para depuração de problemas em redes (embora ambas estejam crescendo rapidamente).

A primeira delas é o analisador de cabos portátil. Esse dispositivo é capaz de medir as características elétricas de um determinado cabo, inclusive seu comprimento (com uma tecnologia maravilhosa chamada reflectometria no domínio de tempo (*time domain reflectrometry* – TDR). Normalmente, esses analisadores também podem apontar falhas simples como cabos rompidos ou mal conectados. Nosso produto favorito para análise de cabos LANs é o Fluke LanMeter. Trata-se de um analisador "tudo em um" capaz de realizar até mesmo pings IP pela rede. Versões topo de linha têm seu próprio servidor Web capaz de exibir históricos estatísticos. Para circuitos WAN (telco), o analisador de linha T-BERD é o "miado" do gato. O T-BERD e o produto topo de linha para testes de LANs que o acompanha, o FIREBERD, são produzidos pela TTC (www.acterna.com).

O segundo tipo de ferramenta de depuração é o analisador (sniffer) de redes. Esse dispositivo desmonta pacotes de rede em busca de erros de protocolo, configurações incorretas e desordens em geral. Há analisadores comerciais disponíveis, mas achamos que o programa livremente disponível Wireshark (www.wireshark.org) executando em um laptop potente é, em geral, a melhor opção.[11]

14.11 CABEAMENTO

Independentemente de você estar em uma rede Ethernet na casa dos gigabits ou apenas com cabos seriais, recomendamos que use cabos da melhor qualidade possível. Isso aumentará as chances de você ainda utilizar a mesma fiação por dez anos. É mais barato empregar fiação/cabeamento no prédio inteiro de uma só etapa em vez de fazê-lo conexão por conexão.

Opções de cabeamento UTP

O fio categoria 5E em geral oferece o preço melhor *versus* relação de desempenho no mercado atual. Seu formato normal é o de quatro pares por revestimento, que é apropriado para uma série de conexões de dados, desde RS-232 até Ethernet na casa dos gigabits.

As especificações da categoria 5E requerem que o trançado seja mantido dentro do intervalo de meia polegada da conexão ao bloco punchdown. Isso implica que qualquer fiação com mais de quatro pares por revestimento terá de ser amarrada com fita ou presa para manter o trançado, uma vez que alimenta mais do que uma conexão.

Você tem de usar terminadores Categoria 5E além dos cabos Categoria 5E. Tivemos mais sorte utilizando componentes fabricados pela Siemon.

Conexões para escritórios

Uma conexão por escritório certamente não é o suficiente. Mas você deve usar duas ou quatro? Recomendamos quatro, por diversas razões:

- Elas podem ser utilizadas para conexões seriais (modem, impressora etc.)
- Elas podem ser utilizadas com telefones
- Elas podem ser utilizadas para acomodar visitantes ou máquinas de demonstração
- O custo dos materiais gira em torno de somente 5% a 10% do custo total
- Sua melhor estimativa multiplicada por dois é uma boa estimativa
- É muito mais barato fazer isso de uma só vez do que ficar acrescentando fiação depois
- Quando o número de portas não é suficiente, as pessoas acrescentam hubs a 10 MB/s comprados em uma loja de informática qualquer e então reclamam com o balconista da velocidade de conexão

Se você estiver prestes a implementar o cabeamento em todo o seu edifício, você poderia considerar a instalação de algumas tomadas nos corredores, salas de conferência, salas de almoço, banheiros e, naturalmente, tetos para pontos de acesso sem fio. Mas não se esqueça da segurança e posicione portas publicamente acessíveis em uma VLAN de "convidado" que não tenha acesso aos recursos da sua rede interna.

---

11. Como ocorre com vários programas populares, o Wireshark está freqüentemente sujeito a ataques de hackers. Certifique-se de ter a versão atualizada.

Padrões de cabeamento

Os prédios modernos normalmente requerem uma infra-estrutura de fiação grande e complexa para suportar todas as diversas atividades que ocorrem internamente. Entrar em um gabinete de telecomunicações mediano normalmente é uma experiência chocante para alguém de estômago fraco, uma vez que fios sem identificação e da mesma cor em geral e cobrem as paredes.

Em uma tentativa de aumentar a capacidade de rastreamento e padronizar a fiação predial, a Telecommunications Industry Association (TIA) lançou o TIA/EIA-606 Administration Standard para infra-estrutura de telecomunicações em prédios comerciais em fevereiro de 1993. O padrão EIA-606 especifica as exigências e as diretrizes para a identificação e documentação de infra-estrutura de telecomunicações.

Os itens abrangidos pelo EIA-606 são:

- Hardware para terminações
- Cabos
- Caminhos de cabos
- Espaços para equipamentos
- Codificação de cores para infra-estrutura
- Símbolos para componentes padrão

Em particular, ele especifica cores padrão a ser utilizadas para fiação. Os detalhes ocultos são revelados na Tabela 14.5.

Tabela 14.5 Tabela de cores EIA-606

Tipo de terminação	Cor	Código[a]	Comentários
Ponto de demarcação	Laranja	150C	Terminações das centrais
Conexões de rede	Verde	353C	Também usado para terminações de circuitos auxiliares
Equipamento comum[b]	Roxo	264C	Principais terminações para equipamentos de switches/dados
Backbone de primeiro nível	Branco	–	Terminações de cabos
Backbone de segundo nível	Cinza	422C	Terminações de cabos
Estação	Azul	291C	Terminações de cabos horizontais
Backbone entre edifícios	Marrom	465C	Terminações de cabos de um campus
Variado	Amarelo	101C	Manutenção, alarmes etc.
Sistemas telefônicos de teclas	Vermelho	184C	–

a. De acordo com o Pantone Matching System®

b. PBXes, hosts, LANs, muxes etc.

A Pantone agora comercializa um software para mapear entre os sistemas Pantone para tinta sobre papel, tinturas têxteis e plástico colorido. Ei, você poderia coordenar as cores da fiação, os uniformes dos instaladores e a documentação sobre fiação! Pensando melhor …

14.12 QUESTÕES DE PROJETO DE REDE

Esta seção trata do projeto lógico e físico de uma rede. Ela é destinada a instalações de médio porte. As idéias aqui apresentadas serão indicadas para algumas centenas de computadores, mas são um exagero para três máquinas e insuficientes para milhares de máquinas. Também supomos que você tenha um orçamento adequado e está começando do zero, o que provavelmente só seja verdadeiro em parte.

A maior parte de um projeto de rede consiste na especificação do seguinte:

- Tipos de mídia que serão utilizados
- A topologia e o roteamento de cabos
- O uso de switches e roteadores

Uma outra questão fundamental no projeto de redes é o controle de congestionamento. Por exemplo, protocolos de compartilhamento de arquivos como o NFS e o CIFS impõem tráfego pesado na rede e, portanto, o arquivo disponibilizado em um cabo de backbone é indesejável.

As questões apresentadas nas seções a seguir são tipicamente aquelas que devem ser consideradas no projeto de qualquer rede.

Arquitetura da rede *versus* arquitetura da edificação

A arquitetura de rede normalmente é mais flexível do que a arquitetura predial, porém as duas têm de coexistir. Se você tiver sorte suficiente para poder especificar a rede antes de o prédio ser construído, seja pródigo. Para a maioria de nós, tanto o prédio quanto o departamento de gerenciamento de instalações já existem e são um tanto rígidos.

Em prédios existentes, a rede tem de usar a arquitetura do prédio, e não brigar contra ela. Os prédios modernos normalmente contêm dutos para passagem de cabos de dados e telefônicos, além daqueles da fiação elétrica de alta voltagem e encanamentos de água e gás. Normalmente eles utilizam forros, uma bênção para instaladores de rede. Muitas empresas têm dutos subterrâneos que facilitam a instalação de redes.

A integridade das paredes corta-fogo[12] deve ser preservada; se você passar um cabo por uma parede/porta corta-fogo, o furo tem de ser justo e preenchido com uma substância não combustível. Respeite os sistemas de retorno de ar na sua escolha dos cabos. Se for pego violando a legislação de incêndio, você poderá ser multado e solicitado a corrigir os problemas criados, mesmo se isso significar inutilizar a rede inteira e refazê-la corretamente.

O projeto lógico de sua rede tem de estar de acordo com as restrições físicas impostas pelos prédios que ele atende. Ao especificar uma rede, tenha em mente que é fácil desenhar uma solução logicamente boa e depois descobrir que é fisicamente difícil senão impossível de implementá-la.

Redes existentes

As redes de computadores são o foco desta discussão, mas muitas empresas já têm redes CATV e redes telefônicas capazes de transmitir dados. Normalmente, essas incluem links por fibra óptica. Se sua empresa estiver pronta para instalar um sistema telefônico novo, compre grandes quantidades de fibra a mais e as instale ao mesmo tempo.

Tivemos a oportunidade, há vários anos, de perguntar aos empreiteiros se eles passariam alguns cabos de fibra óptica para nós. "Claro, sem cobrar nada extra", e ficaram um pouco zangados quando mostramos a tonelada de cabos de fibra óptica que eles teriam de instalar.

Expansão

É muito difícil prever necessidades daqui a dez anos, especialmente nos campos da computação e de suas redes. Portanto, é importante projetar a rede tendo em mente fatores como expansão e aumento na largura de banda. À medida que o cabo for sendo instalado, especialmente em lugares fora do caminho e de difícil acesso, puxe de três a quatro vezes o número de pares de que você realmente precisa no momento. Lembre-se: a maior parte do custo da instalação reside na mão-de-obra, não nos materiais.

Mesmo que você não tenha planos imediatos para usar fibra óptica, é sensato instalar certa extensão ao fazer a fiação de seu prédio, especialmente se for difícil de instalar cabos mais tarde. Puxe tanto fibra modo simples quanto multimodo; certamente o tipo de que você precisará no futuro sempre será aquele que você não instalou.

Congestionamento

Uma rede é como uma corrente: só é tão boa quanto o seu elo mais fraco ou mais lento. O desempenho de redes Ethernet, como muitas outras arquiteturas de rede, degrada não linearmente à medida que a rede é carregada.

Switches sobrecarregados, interfaces cujos tipos são discordantes e links de baixa velocidade são fatores que levam ao congestionamento de uma rede. Vale a pena isolar tráfego local criando sub-redes e utilizando dispositivos de interconexão como roteadores. Sub-redes também podem ser utilizadas para isolar máquinas empregadas em testes. É difícil conduzir um experimento que envolve várias máquinas se não houver uma maneira fácil de isolar essas máquinas tanto física como logicamente do restante da rede.

Manutenção e documentação

Descobrimos que a capacidade de manutenção de uma rede está altamente correlacionada com a qualidade de sua documentação. Documentação acurada, completa e atualizada é absolutamente indispensável.

Os cabos devem ser rotulados em todos os pontos de terminação e também em intervalos regulares de alguns metros para ser facilmente identificados quando descobertos em um forro ou parede.[13] É uma boa idéia deixar cópias dos diagramas de cabeamento local em gabinetes informativos para que possam ser atualizados no local quando alterações forem feitas. Depois, em intervalos regulares de algumas semanas, alguém deve copiar essas alterações para entrada em um banco de dados eletrônico.

---

12. Esse tipo de compartimento é feito de concreto, tijolo ou material que retarda as chamas impedindo-as de se propagarem e queimarem o prédio. Embora um tanto diferente de um firewall para segurança de rede, provavelmente ele tem a mesma importância.

13. Alguns fabricantes de cabos já identificam previamente rolos de cabos em intervalos de alguns metros.

As junções entre os principais centros populacionais na forma de switches ou roteadores podem facilitar a depuração possibilitando que partes da rede sejam isoladas e depuradas separadamente. Também é útil colocar junções entre domínios políticos e administrativos.

14.13 QUESTÕES DE GERENCIAMENTO

Se a rede deve funcionar corretamente, algumas coisas precisam ser centralizadas, outras distribuídas e outras ainda devem ser locais. Há necessidade de formular e de haver acordo sobre regras concretas razoáveis e de "política de boa vizinhança".

Um ambiente típico incluiria o seguinte:

- Uma rede backbone entre os prédios
- Sub-redes departamentais conectadas ao backbone
- Sub-redes de grupos dentro de um departamento
- Conexões com o mundo externo (por exemplo, Internet ou escritórios externos)

Várias facetas do projeto e da implementação de redes devem ter controle, responsabilidade, manutenção e financiamento que abranja toda a instalação. As redes com algoritmos de tarifação segmentada para cada conexão crescem de modo muito bizarro, porém muito previsível à medida que os departamentos tentam minimizar seus próprios custos locais. Os principais alvos para controle centralizado são:

- O projeto da rede, incluindo o emprego de sub-redes, roteadores, switches etc.
- A rede backbone em si, inclusive as conexões para ele
- Endereços IP de hosts, nomes de host e nomes de subdomínios
- Protocolos, basicamente para garantir interoperabilidade
- Política de roteamento para a Internet

Os nomes de domínios, endereços IP e nomes de rede são, a certo ponto, já controlados por autoridades como ARIN, ICANN e NIC-BR (registro.Br), que faz o registro de domínios no Brasil. Entretanto, o uso desses itens por parte de sua instalação também deve ser coordenado localmente.

Uma autoridade central possui uma visão global da rede: seu projeto, capacidade e crescimento esperado. Ela é capaz de arcar com equipamentos de monitoramento próprios (e o pessoal para comandá-los) e de manter a saúde da rede backbone. Ela pode insistir em projeto de rede correto, mesmo quando isso significar dizer a um departamento para comprar um roteador e criar uma sub-rede para conexão com a rede backbone do campus. Tal decisão poderia ser necessária para que uma nova conexão não tivesse impacto negativo sobre a rede existente.

Se uma rede atende a vários tipos de máquinas, sistemas operacionais e protocolos, é quase essencial ter um roteador muito inteligente (por exemplo, Cisco) como um gateway entre redes..

14.14 FORNECEDORES RECOMENDADOS

Ao longo de mais de 15 anos instalando redes ao redor do mundo, acabamos nos dando mal algumas vezes por utilizarmos produtos que não atendiam às especificações ou foram mal representados, eram muito caros ou de alguma forma deixaram de atender às nossas expectativas. Apresentamos a seguir uma lista de fornecedores como referência, que ainda confiamos, recomendamos e nós próprios utilizamos hoje em dia.

Cabos e conectores

AMP (agora parte da Tyco)
(800) 522-6752
www.amp.com

Anixter Newark Electronics
(800) 264-9837
www.anixter.com

Belden Cable
(800) 235-3361
(765) 983-5200
www.belden.com

Black Box Corporation
(724) 746-5500 e (11) 5515.4001
www.blackbox.com

(800) 463-9275
www.newark.com

Siemon
(860) 945-4395
www.siemon.com

Equipamento de teste

Fluke
(800) 443-5853 e (11) 3759-7600
www.fluke.com

Acterna
(866) 228-3762
www.acterna.com

Siemon
(860) 945-4395
www.siemon.com

Roteadores/switches

Cisco Systems
(415) 326-1941 e 0800 702 4726
www.cisco.com

14.15 Leitura recomendada

BARNETT, DAVID E GROTH, DAVID E McBEE, JIM. *Cabling:* The Complete Guide to Network Wiring. *3. ed.* San Francisco: Sybex, 2004.

SEIFERT, RICH. *Gigabit Ethernet:* Technology and Applications for High Speed LANs. Reading, MA: Addison-Wesley, 1998.

ANSI/TIA/EIA-568-A, *Commercial Building Telecommunications Cabling Standard*, e ANSI/TIA/EIA-606, *Administration Standard for the Telecommunications Infrastructure of Commercial Buildings.* Veja também www.tiaonline.org.

SPURGEON, CHARLES. "Guide to Ethernet." Disponível em: <www.ethermanage.com/ethernet>

14.16 Exercícios

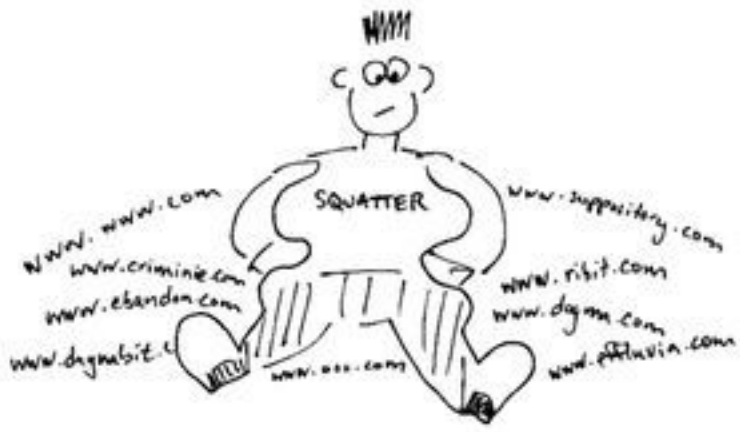
E14.1 Hoje, a maioria dos edifícios para escritórios abriga redes de computadores e essas são conectadas via UTP Ethernet. Algumas combinações de hubs e switches são necessárias para suportar essas redes. Em muitos casos, os dois tipos de equipamentos são intercambiáveis. Liste as vantagens e as desvantagens de cada um deles.

★ E14.2 Desenhe uma ilustração de rede simples e imaginária que conecte uma máquina em sua rede experimental à Amazon.com. Inclua componentes LAN, MAN e WAN. Mostre que tecnologia é utilizada para cada um desses componentes. Ilustre alguns hubs, switches e roteadores.

★ E14.3 Pesquise o protocolo Temporal Key Integrity do WPA. Detalhe as vantagens que ele tem em relação ao WEP e os tipos de ataques que ele evita.

★ ★ E14.4 TTCP é uma ferramenta que mede o desempenho de redes TCP e UDP (procure-a em www.rpmfind.net). Instale o TTCP em duas máquinas ligadas em rede e meça o desempenho do link entre elas. O que acontece com a largura de banda se você aumentar ou diminuir os tamanhos dos buffers? Como se comparam os números obtidos em relação à capacidade teórica da mídia física?

15 DNS: o Domain Name System



Zilhões de computadores (hosts) estão conectados à Internet. Como controlar todos eles quando pertencem a tantos países, redes e grupos administrativos diferentes? Duas peças fundamentais da infra-estrutura juntam todas as coisas: o Domain Name System (DNS), que monitora de quem são os hosts, e o sistema de roteamento Internet, que monitora a maneira como eles são conectados.

Este capítulo (alguns diriam um minilivro) é sobre o DNS. Embora o DNS tenha surgido para atender a diversos propósitos, sua tarefa principal é a de associar nomes de hosts a endereços IP. Usuários e programas no nível de usuário adoram se referir a máquinas pelo nome, mas o software de rede de baixo nível entende somente números. O DNS fornece o elo que faz com que todo o mundo fique contente. O DNS também desempenha um papel importante no roteamento de e-mail, acesso de servidor Web e muitos outros serviços.

O DNS é um banco de dados distribuído. "Distribuído" significa que minha instalação armazena os dados sobre meus computadores, a sua instalação armazena dados sobre os seus computadores e, de alguma forma, nossas instalações cooperam e compartilham dados automaticamente entre si quando uma instalação precisa procurar dados do outro.

Nossa discussão sobre o DNS pode ser dividida em três seções fundamentais:

- O sistema DNS em geral: sua história, dados e protocolos
- O BIND, uma implementação específica do sistema DNS
- A operação e a manutenção dos servidores BIND, incluindo tópicos relacionados, como a segurança

Se precisar configurar ou manter os servidores DNS da sua instalação e se você já tiver uma idéia geral de como o DNS funciona, sinta-se livre para avançar.

Antes de começar a discutir as generalidades do DNS, vamos primeiro parar por um momento e responder a uma pergunta feita com mais freqüência por todo mundo: como adicionar uma nova máquina a uma rede que já utiliza o BIND? O que vem a seguir é uma receita passo a passo que não define nem explica terminologia alguma e que talvez não se encaixe precisamente nas diretivas e procedimentos do seu administrador de sistemas local. Tome cuidado.

15.1 DNS PARA O IMPACIENTE: ACRESCENTANDO UMA NOVA MÁQUINA

Se sua rede está configurada para utilizar o Dynamic Host Configuration Protocol (DHCP), talvez você não precise realizar uma configuração manual para o DNS. Quando um novo computador é adicionado, o servidor DHCP informa quais servidores DNS ele deve utilizar nas consultas. Mapeamentos de hostnames para endereços IP utilizados pelo mundo externo são especificados quando o servidor DHCP é configurado e inserido automaticamente por meio do recurso de atualização dinâmica do DNS.

Para redes que não utilizam o DHCP, a seguinte receita mostra como atualizar a configuração do DNS, copiando e modificando os registros de um computador semelhante.

Passo 1: Escolha um hostname e um endereço IP para a nova máquina em conjunção com os administradores de sistemas locais ou com o ISP em que você está conectado.

Passo 2: Identifique uma máquina similar na mesma sub-rede. Usaremos os registros dessa máquina como modelo para nossas máquinas novas. Neste exemplo, usaremos uma máquina chamada templatehost.my.domain como o modelo.

Passo 3: Faça o login na máquina principal ou mestre de servidor de nomes.

Passo 4: Examine o arquivo de configuração do servidor de nomes, normalmente **/etc/named.conf**:

- Dentro da instrução `options`, localize a linha `directory` que informa onde arquivos de dados de zona são mantidos em sua instalação (consulte a página 295). Os arquivos de zona contêm o host em si e os dados de endereço IP.

- A partir das instruções zone, localize os nomes de arquivo para o arquivo de zona direta e para o arquivo de zona reversa apropriados ao seu novo endereço IP (página 300).
- Verifique nas instruções `zone` se esse servidor é de fato o servidor mestre para o domínio. A instrução da zona direta deve ser semelhante a:

```
zone "my.domain" {
type master;
...
```

Passo 5: Vá até o diretório do arquivo de zona e edite o arquivo da zona direta. Localize os registros para o host modelo que você identificou antes. Eles serão parecidos com o seguinte:

```
templatehost IN     A       128.138.243.100
                            IN      MX      10      mail-hub
                            IN      MX      20      templatehost
```

Sua versão talvez não inclua as linhas MX, que são utilizadas para roteamento de correio.

Passo 6: Duplique esses registros e altere-os de maneira apropriada para o seu novo host. O arquivo de zonas pode estar ordenado segundo o nome de host; siga a convenção existente. Certifique-se também de alterar o número de série no registro SOA no início do arquivo (ele é o primeiro dos cinco números no registro SOA). O número de série só deve aumentar; adicione 1 se sua instalação usar um número de série arbitrário ou configure o campo com a data atual se sua instalação usar essa convenção.

Passo 7: Edite o arquivo de zonas reversas,[1] duplique o registro para o host modelo e atualize-o. Ele deve ser parecido com o seguinte:

```
100                 IN      PTR     templatehost.my.domain.
```

Observe que há um ponto final depois do nome de host; não o omita. Você também tem de atualizar o número de série no registro SOA do arquivo de zonas reversas.

Se o seu arquivo de zonas reversas apresentar mais do que o último byte do endereço IP de cada host, você terá de introduzir os bytes na ordem inversa. Por exemplo, o registro:

```
100.243             IN      PTR     templatehost.my.domain.
```

corresponde ao endereço IP 128.138.243.100 (neste caso, a zona reversa é relativa a 138.128.in-addr.arpa, e não a 243.138.128.in-addr.arpa).

Passo 8: Ainda conectado à máquina mestre do servidor de nomes, execute **rndc reload**, ou, se for um servidor ocupado, simplesmente recarregue os domínios (ou visualizações) que você alterou:

```
# rndc reload nome-da-zona-de-encaminhamento
# rndc reload nome-da-zona-reversa
```

Passo 9: Verifique a configuração usando **dig**; consulte a página 330. Você também poderá tentar **ping** ou **traceroute** para o nome de seu novo host, mesmo que o novo host ainda não tenha sido configurado. Uma mensagem "host unknown" (máquina desconhecida) indica que você errou; "host not responding" (máquina não respondendo) significa que provavelmente tudo está OK.

Os erros mais comuns são:

- esquecer de atualizar o número de série e reiniciar o servidor de nomes; e
- esquecer de acrescentar um ponto no final do nome de host na entrada reversa PTR.

15.2 A HISTÓRIA DO DNS

O DNS foi formalmente especificado por Paul Mockapetris nas RFCs 882 e 883 (1983) e atualizado nas RFCs 1034 e 1035 (1987). Ele continha dois conceitos-chave: nomes de host hierárquicos e responsabilidade distribuída.

1. A zona reversa poderia ser mantida em qualquer outro lugar (por exemplo, na instalação de seu provedor de acesso). Se isso acontecer, a entrada reversa terá de ser inserida lá.

Implementações BIND

A implementação UNIX original foi feita por quatro alunos graduados na Berkeley (Douglas Terry , Mark Painter, David Riggle e Songnian Zhou), em 1984. Ela foi então adicionada à distribuição Berkeley UNIX por Kevin Dunlap em meados da década de 1980 e tornou-se conhecida como BIND, o sistema Berkeley Internet Name Domain. Paul Vixie e o ISC, Internet Systems Consortium (www.isc.org, conhecido como Internet Software Consortium antes de 2004), mantém atualmente o BIND. Ele é um projeto de código-fonte aberto. Em 2000 e 2001, o ISC desenvolveu uma versão totalmente nova do BIND — o BIND 9 — com financiamento de vários fornecedores, órgãos governamentais e outras organizações.

O ISC também fornece vários tipos de apoio a esses produtos, inclusive ajuda na configuração, classes para BIND e DNS e até mesmo programação personalizada. Esses serviços são uma bênção para instalações que são obrigadas a ter um contrato de suporte antes de poder usar o código-fonte aberto. Várias empresas utilizam contratos de serviço como uma maneira de contribuir com o ISC — elas adquirem contratos caros, mas jamais pedem ajuda.

Graças a uma adaptação da feita pela Nortel, o BIND está disponível para o Windows e para o UNIX/Linux. Como o protocolo DNS está padronizado, todas as implementações DNS do UNIX e de não-UNIX podem interoperar e compartilhar dados. Muitas instalações executam servidores UNIX para fornecer serviço DNS a seus desktops Windows; esta combinação funciona bem.

As RFCs 1034 e 1035 ainda são consideradas a especificação-base para o DNS, porém mais de 40 outras RFCs as suplantaram e acrescentaram diversos aspectos ao protocolo e registros de dados ao longo da última década (veja a lista no final deste capítulo). Atualmente, nenhum padrão ou RFC única consegue juntar todas as partes em único local. Historicamente, o DNS foi mais ou menos definido como "o que o BIND implementa", embora isso esteja se tornando menos preciso à medida que surgem outros servidores DNS.

Outras implementações DNS

No início, o BIND era a única implementação de DNS em uso disseminado. Hoje há várias, tanto de código-fonte aberto como comerciais. Muitas não implementam todas as especificações definidas pelas várias RFCs relacionadas ao DNS que ainda não foram padronizadas. A Tabela 15.1 lista as implementações DNS mais populares e mostra onde procurar informações adicionais.

Tabela 15.1 Algumas implementações DNS populares

Nome	Autor	Fonte	Comentários
BIND	ISC	isc.org	Autorizado ou de cache
NSD	NLnet Labs	www.nlnetlabs.nl	Autorizado somente
PowerDNS	PowerDNS BV	www.powerdns.com	Autorizado somente
djbdns [a]	Dan Bernstein	tinydns.org	Viola algumas RFCs
DNS da Microsoft	Microsoft	microsoft.com	Culpada de vários pecados
ANS, CNS	Nominum	www.nominum.com	Autorizado ou de cache

a. Também conhecido como tinydns, que é o componente de servidor do pacote djbdns.

A pesquisa de domínio em progresso do ISC monitora as várias implementações de DNS e o número dos servidores de nomes utilizado em cada uma. Para ver a atual demografia populacional, acesse www.isc.org, clique em ISC Internet Domain Survey, em Latest Survey Results e, por fim, em Domain Server Software Distribution. Dispositivos de DNS como o Infoblox (www.infoblox.com) são utilizados por algumas instalações grandes, mas ainda não aparecem no fingerprinting da pesquisa.

Neste livro discutimos apenas o BIND, que é considerado a implementação DNS de referência. Ele é de longe a implementação mais amplamente utilizada e é uma escolha apropriada para a maioria das instalações. O BIND monitora todos os padrões e os padrões propostos pelo IETF, freqüentemente implementando recursos antes de as suas especificações estarem completas. Isso é bom porque alguns recursos de monitoramento de padrões revelaram-se falhos — a inclusão destes no BIND permite que problemas sejam reconhecidos e corrigidos antes de serem incorporados "oficialmente".

NSD, o daemon de servidor de nomes, foi desenvolvido em 2003 pelos NLnet Labs em Amsterdã. Ele fornece um apropriado servidor de nomes rápido, seguro e autorizado para uso por root e servidores de domínio mais importantes. (Um servidor autorizado é apropriado para fornecer as respostas a consultas sobre hosts no seu domínio, mas ele não pode responder às consultas dos usuários sobre outros domínios.)

O PowerDNS é um servidor de nomes autorizado de código-fonte aberto que fornece um sistema de back-end flexível único. Os dados sobre o DNS podem ser provenientes de arquivos ou de uma longa lista de outras fontes: MySQL, Oracle (8i e 9i), DB2 da IBM, PostgreSQL, SQL Server da Microsoft, LDAP, ODBC, XDB ou mesmo um pipe UNIX.

O djbdns é um pacote alternativo de servidor de nomes que consiste em um servidor autorizado chamado tinydns e um servidor de cache chamado dnscache. Ele diz ser seguro e muito rápido, embora alguns dos dados sobre a medição que conhecemos sejam inconclusivos. Sua principal desvantagem é que ele viola os padrões DNS freqüente e intencionalmente tornando difícil a interoperação com outros servidores DNS.

A Microsoft fornece um servidor DNS para Windows, mas a implementação da Microsoft possui suas peculiaridades e diferenças próprias. Ele interage com o BIND, mas também tende a congestionar a rede com pacotes desnecessários e inválidos.

A Nominum, empresa contratada que escreveu a versão inicial do BIND 9 para o ISC, vende seus próprios servidores de nomes e ferramentas de gerenciamento de rede. Os servidores da Nominum são bem rápidos e incluem a maioria dos padrões propostos atualmente.

15.3 QUEM PRECISA DO DNS?

O DNS define:

- Um espaço de nomes hierárquico para hosts e endereços IP
- Um banco de dados distribuído das informações sobre o nome de host e de endereço
- Um "conversor" para consultar esse banco de dados
- Encaminhamento de e-mails otimizado
- Um mecanismo para localizar serviços em uma rede
- Um protocolo para intercâmbio de informações de atribuição de nomes

Para serem verdadeiros cidadãos da Internet, os sites precisam do DNS. Manter um arquivo **/etc/hosts** local com mapeamentos para cada host que você imagina querer entrar em contato é impossível.

Cada site mantém um ou mais blocos do banco de dados distribuído que forma o sistema DNS mundial. Seu banco de dados consiste em arquivos de texto que contêm registros para cada um dos seus hosts. Cada registro é uma linha única formada por um nome (normalmente um nome de host), um tipo de registro e alguns valores de dados. O campo de nome pode ser omitido se seu valor for o mesmo da linha anterior.

Por exemplo, as linhas a seguir:

```
bark        IN      A       206.168.198.209
            IN      MX      10 mailserver.atrust.com.
```

no arquivo "direto" (forward) e

```
209         IN      PTR     bark.atrust.com.
```

no arquivo "reverse" associam "bark.atrust.com" com o endereço IP 206.168.198.209 e roteiam novamente o e-mail endereçado a esta máquina para o host mailserver.atrust.com.

O DNS é um sistema cliente/servidor. Servidores ("servidores de nomes") carregam os dados de seus arquivos DNS para a memória e os utilizam para responder a consultas tanto de clientes internos como de clientes externos e de outros servidores na Internet. Todos os seus hosts devem ser clientes DNS, mas um número relativamente pequeno precisa ser servidor DNS.

Se sua organização for pequena (alguns hosts em uma única rede), você poderá executar um servidor em um host ou solicitar a seu ISP para fornecer serviço DNS em seu nome. Um site de porte médio com várias sub-redes deve manter vários servidores DNS para reduzir a latência de consultas e aumentar a confiabilidade. Um site muito grande poderá dividir seu domínio DNS em subdomínios e manter vários servidores para cada subdomínio.

15.4 O ESPAÇO DE NOMES DNS

O espaço de nome DNS é organizado de acordo com aquilo que os matemáticos chamam árvore; cada nome de domínio corresponde a um nó na árvore. Um ramo da árvore de nomes associa nomes de hosts a endereços IP e um segundo ramo associa endereços IP de volta a seus nomes de host. O primeiro ramo é chamado de "mapeamento direto" e os arquivos de dados BIND associados a ele são chamados de "arquivos de zonas diretas". O ramo endereço-nome de host é chamado de "mapeamento reverso" e seus arquivos de dados são chamados de "arquivos de zonas reversas". Infelizmente, muitos sites não mantêm seus mapeamentos reversos.

Cada domínio representa um trecho diferente do espaço de nomes e é controlado de maneira livre por uma única entidade administrativa. A raiz da árvore é chamada de "." ou ponto, e abaixo dele se encontram os domínios de primeiro nível (ou domínios em nível-raiz).

Por razões históricas, dois tipos de nomes de domínio de primeiro nível estão sendo utilizados hoje em dia. Nos Estados Unidos, os domínios de primeiro nível originalmente descreviam entidades organizacionais e políticas e recebiam nomes de três letras como *com* e *edu*. Alguns desses domínios (principalmente com, org e net) também são utilizados fora dos Estados Unidos; eles são chamados de domínios genéricos de primeiro nível ou gTLDs (*generic top-level domains*), de forma abreviada.

Os domínios de primeiro nível eram relativamente fixos no passado, mas o ICANN aprovou sete novos no final de 2000: biz, info, name, pro, museum, aero e coop.[2] Mais recentemente, "jobs" foi adicionado à lista gTLD. Esses domínios são agora operacionais e disponíveis para uso. Biz, info e nomes do domínio são chamados gTLDs "não patrocinados" e são abertos a qualquer pessoa; museum, aero, jobs, pro e coop são TLDs "patrocinados" limitados a tipos específicos de quem os registra.

A Tabela 15.2 lista os gTLDs mais importantes juntamente com seus objetivos originais. Quando bons nomes no domínio com se tornaram escassos, os órgãos de registro começaram a oferecer nomes nos domínios org e net sem levar em conta as restrições originais destes domínios. Os domínios da coluna esquerda da Tabela 15.2 são os originais, que remontam a aproximadamente 1988; a coluna da direita inclui os novos domínios acrescentados em 2001.

Tabela 15.2 Domínios genéricos de primeiro nível

Domínio	A que se destina	Domínio	A que se destina
com	Empresas comerciais	aero	Segmento de transporte aéreo
edu	Instituições educacionais norte-americanas	biz	Negócios
gov	Órgãos governamentais norte-americanos	coop	Cooperativas
mil	Órgãos militares norte-americanos	info	Uso irrestrito
net	Provedores de rede	jobs	Pessoal de recursos humanos
org	Organizações sem fins lucrativos	museum	Museus
int	Organizações internacionais	name	Indivíduos
arpa	Árvore de endereços IP	pro	Contadores, advogados etc.

Para os domínios fora dos Estados Unidos, são utilizados códigos de países ISO de duas letras. Esses domínios são conhecidos como ccTLDs ou "domínios de primeiro nível para o código de países". Tanto TLDs geográficos como organizacionais coexistem dentro do mesmo espaço de nomes global. A Tabela 15.3 mostra alguns códigos comuns para países.

Tabela 15.3 Códigos de países comuns

Código	País	Código	País	Código	País
au	Austrália	fi	Finlândia	hk	Hong Kong
ca	Canadá	fr	França	ch	Suíça
br	Brasil	jp	Japão	mx	México
de	Alemanha	se	Suécia	hu	Hungria

Alguns países, salvo os Estados Unidos, criam uma hierarquia organizacional com domínios de segundo nível. As convenções para atribuição de nomes variam. Por exemplo, uma instituição acadêmica poderia ser um edu nos Estados Unidos e um ac.jp no Japão[3].

O domínio de primeiro nível "us" também é utilizado algumas vezes nos Estados Unidos, basicamente com domínios de localidade; por exemplo, bvsd.k12.co.us, a Boulder Valley School District no Estado de Colorado. O domínio "us" nunca é combinado com um domínio educativo — não há nenhum "edu.us" (ainda). A vantagem dos nomes de domínios "us" se baseia no fato de apresentarem registros gratuitos ou baratos; veja www.nic.us para obter detalhes. As restrições nos domínios de segundo nível abaixo de "us" (que inicialmente eram limitados aos estados norte-americanos) foram relaxadas e nomes de domínio como evi-nemeth.us podem ser utilizados.

Os piratas de domínios em alguns casos compram o espaço de nomes de um país inteiro. Por exemplo, o domínio para a Moldávia, "md", agora está sendo comercializado para médicos e residentes do Estado de Maryland (MD) nos Estados Unidos. Um outro exemplo é Tuvalu, para o qual o código de país é "tv". A primeira dessas vendas foi Tonga ("to"), a mais

2. ICANN (Internet Corporation for Assigned Names and Numbers): consulte a página 192 para mais informações sobre o ICANN.
3. N. R. T: No Brasil, uma lista da categoria de domínios pode ser encontrada em http://registro.br/info/dpn.html e engloba mais de uma dezena de tipos.

ativa no momento é Niue ("nu") e, quem sabe, a mais atraente seja "tm" de Turcomenistão. Essas negociatas algumas vezes foram razoáveis para o país com o código de duas letras desejado e outras vezes não.

A grilagem de domínios é amplamente praticada: o pessoal registra nomes que eles acham que serão solicitados no futuro e depois os revendem para as empresas cujos nomes afanaram. Anos atrás, todas as estações de esqui do Colorado foram registradas com o nome do mesmo indivíduo, que fez um bom dinheiro revendendo-os para várias estações de esqui individuais na época em que estes quiseram ter seus sites na Web.

O preço médio para um bom nome de domínio "com" é de milhares a milhões de dólares. Ofereceram-nos US$ 50.000 pelo nome admin.com, que conseguimos anos atrás quando sysadmin.com já havia sido adquirido pela revista /Sys/Admin. O preço mais alto até agora (ou, pelo menos, o mais alto conhecido) foi de US$ 7,5 milhões pagos pelo domínio business.com durante o auge do "boom" das ações das empresas de tecnologia. US$ 20.000 a US$ 100.000 é um faixa mais comum hoje em dia, mas vendas multimilionárias ainda estão ocorrendo; um exemplo disso é a venda em julho de 2004 do CreditCards.com por US$ 2,75 milhões.

Esperava-se que Dan Parisi, empresário da Internet, recebesse milhões de dólares pelo ex-site pornô, whitehouse.com, que foi colocado à venda no início de 2004. Várias empresas utilizam esse nome desde então, mas os termos exatos e os detalhes financeiros nunca se tornaram público.

Atualmente, nomes de domínio válidos consistem apenas em letras, números e traços. Cada componente de um nome não pode conter mais de 63 caracteres e os nomes devem ter menos que 256 caracteres. A internacionalização do sistema e suporte de DNS para conjuntos de caracteres não ASCII irá por fim mudar todas as regras de atribuição de nomes, mas, por enquanto nomes de outros conjuntos de caracteres são mapeados de volta para o ASCII; ver a página 270.

Os nomes de domínios não diferenciam as letras maiúsculas de minúsculas. "Colorado" é o mesmo que "colorado", que é o mesmo que "COLORADO" no que se refere a DNS. Implementações atuais de DNS devem ignorar a distinção entre maiúsculas e minúsculas ao fazer comparações, porém devem propagar essa distinção quando ela é fornecida. No passado era comum utilizar letras maiúsculas para domínios de primeiro nível e uma letra maiúscula inicial para domínios de segundo nível. Nos dias de hoje, os dedos já estão cansados de digitar e é norma colocar tudo em letra minúscula.[4]

O nome de um host Internet totalmente qualificado é formado anexando-se seu nome de domínio a seu nome de host. Por exemplo, boulder.colorado.edu é o nome totalmente qualificado para o host boulder na Universidade de Colorado. Outros sites podem usar o nome de host boulder sem entrar em conflito, pois os nomes totalmente qualificados serão diferentes.

Dentro do sistema DNS, os nomes totalmente qualificados terminam com um ponto, por exemplo, "boulder.colorado.edu.". A falta de um ponto final pode indicar um endereço relativo. Dependendo do contexto em que um endereço relativo é utilizado, componentes extras poderiam ser adicionados. A convenção de ponto final geralmente permanece oculta dos usuários comuns do DNS. De fato, alguns sistemas (como e-mail) travarão se você fornecer o ponto por conta própria.

É comum um host ter mais de um nome. O host boulder.colorado.edu também poderia ser conhecido como www.colorado.edu ou ftp.colorado.edu se quiséssemos fazer com que seu nome refletisse os serviços por ele fornecidos. De fato, é uma boa prática tornar os nomes de host de serviços como www "móveis", de modo que você poderá deslocar servidores de uma máquina para outra sem alterar o nome primário de qualquer máquina. Você pode atribuir nomes extras utilizando a construção CNAME; consulte a página 277.

Quando recebemos o nome colorado.edu, nos foi garantido que colorado era exclusivo dentro do domínio edu. Dividimos ainda mais esse domínio em subdomínios de acordo com os departamentos. Por exemplo, o host anchor no Departamento de Ciências da Computação chama-se anchor.cs.colorado.edu na Internet.

A criação de cada subdomínio tem de ser coordenada com os administradores do domínio acima para garantir exclusividade. As entradas nos arquivos de configuração para o domínio pai delegam autoridade para o espaço de nomes ao subdomínio.

Mestres de seus domínios

O gerenciamento de domínios de primeiro nível com, org, net e edu, nos Estados Unidos, era antigamente coordenado pela Network Solutions, Inc., sob contrato da National Science Foundation. Essa situação de monopólio agora mudou e outras organizações têm permissão para registrarem nomes de domínios nesses gTLDs. Outros domínios de primeiro nível, como aqueles para cada país, são mantidos por organizações regionais.

Houve várias propostas para permitir que empresas privadas operassem seus próprios domínios de primeiro nível, e é provável que domínios de primeiro nível adicionais estejam disponíveis em um futuro próximo. Consulte www.icann.org para obter informações atualizadas.

---

4. O BIND conserva a distinção entre maiúsculas e minúsculas, mas algumas implementações (por exemplo, implementações da Microsoft e djbdns) alteram a distinção entre maiúsculas e minúsculas de acordo com suas próprias preferências. Isto é o bastante para o assunto padrões rigorosos.

A maioria dos ISPs oferece serviços de registro de nomes de domínios tarifados. Eles lidam com a autoridade de domínios de primeiro nível em seu nome e configuram seus servidores DNS para tratarem das pesquisas de nomes dentro de seu domínio. A desvantagem de depender dos servidores de um ISP é o fato de você perder o controle direto sobre a administração de seu domínio.

Consulte a página 201 para informações adicionais sobre CIDR.

Para gerenciar seus próprios serviços de DNS, ainda é necessário coordenar isso com seu ISP. A maioria dos ISPs fornece mapeamentos DNS reversos para endereços IP dentro de seus blocos CIDR. Se você assumir o gerenciamento de seus endereços DNS, certifique-se de que seu ISP desativa seu serviço para esses endereços e delegue essa responsabilidade a você.

Os mapeamentos direto e reverso de um domínio devem ser gerenciados no mesmo local sempre que possível. Alguns ISPs ficam contentes em deixar sob sua responsabilidade o gerenciamento dos arquivos diretos, mas são relutantes em abandonar o controle de mapeamentos reversos. Esse gerenciamento dividido pode levar a problemas de sincronização. Consulte a página 278 para uma alternativa elegante que faz com que a delegação funcione até mesmo para pequenos trechos de espaço de endereços.

Os domínios DNS devem (são obrigados, na realidade; consulte a RFC1219) ser servidos por pelo menos dois servidores. Um arranjo comum é aquele em que um site opera o seu próprio servidor mestre e deixa que os servidores do ISP atuem como escravos. Depois de o sistema ser configurado, os servidores do ISP fazem automaticamente o download dos dados sobre o host a partir do servidor mestre. Alterações feitas na configuração DNS se refletem nos servidores escravos sem qualquer intervenção explícita por parte dos administradores de ambos os sites.

Não coloque todos os seus servidores DNS na mesma rede. Quando o DNS pára de operar, a rede, efetivamente, pára aos olhos de seus usuários. Distribua seus servidores DNS de modo que você não acabe ficando com um sistema frágil com um único ponto de falha. O DNS é bem robusto se configurado de maneira cuidadosa.

Selecionando um nome de domínio

Nosso conselho costumava ser para que os nomes fossem os mais curtos e fáceis de serem digitados e que identificassem a organização que os usava. Nos dias de hoje, a realidade é que todos os nomes bons e curtos já foram utilizados por alguém, pelo menos no domínio com. É tentador jogar a culpa nos grileiros, mas, na realidade, a maioria dos bons nomes está efetivamente em uso. Em 2004, aproximadamente 60% dos nomes registrados estavam em uso; historicamente, menos da metade dos nomes registrados era de fato utilizada.

O inchamento dos domínios

O DNS foi desenvolvido para associar o nome de domínio de uma organização a um servidor de nomes para essa organização. Desse modo ele precisa ser capaz de ser ampliado ao número de organizações existentes no mundo. Entretanto, agora que a Internet se tornou um conduíte de cultura de massa, os nomes de domínios estão sendo aplicados a todos os produtos, filmes, eventos esportivos, substantivos em inglês etc. Nomes de domínios como twinkies.com não estão (diretamente) relacionados à empresa que fabrica o produto; eles simplesmente estão sendo utilizados como propagandas. Não está claro que o DNS é capaz de continuar a crescer dessa forma exponencial. O problema real aqui é que a árvore de atribuição de nomes DNS é uma estrutura de dados mais eficiente quando ela tem alguma forma de hierarquia e não é totalmente plana. Com cada empresa atribuindo centenas ou até mesmo milhares de produtos no primeiro nível da árvore, a hierarquia está condenada.

O que realmente precisamos é de um serviço de diretórios que associe marcas e nomes-fantasia a empresas, deixando o DNS livre para lidar com os endereços IP. Outra solução possível é forçar a hierarquia no sistema; por exemplo, twinkies.hostess-foods.com. Mas isso nunca ocorrerá — já fomos muito longe na trilha do marketing.

A Sony faz as coisas certas segundo a perspectiva do DNS — todos os seus produtos são subdomínios de sony.com. Pode significar um ou dois cliques a mais para localizar os produtos desejados, mas o DNS aprecia a hierarquia.

Registrando um nome de domínio de segundo nível

Para obter um nome de segundo nível, você tem de solicitar o domínio de primeiro nível apropriado a um serviço de registros. O ICANN credencia várias agências para fazerem parte de seu projeto de escritórios de registro compartilhado para se registrar nomes nos gTLDs. Na época em que escrevíamos este livro, havia algo em torno de 500 opções de agências de registro. Vá ao endereço www.icann.org para obter uma lista definitiva.

Para o registro de um nome ccTLD na Europa, entre em contato com o Council of European National Top-level Domain Registries em www.centr.org para identificar seu registro local e requerer um nome de domínio. Para a região Ásia/Pacífico, o órgão responsável é o Asia-Pacific Network Information Center, www.apnic.net.[5]

---

5. N. R. T.: No Brasil isso está a cargo da NIC.Br em registro.Br

Para completar os formulários de registro de domínios, você deve identificar uma pessoa para contatos técnicos, uma para contatos administrativos e pelo menos dois hosts que serão os servidores para o seu domínio.

Criando seus próprios subdomínios

O procedimento para criação de um subdomínio é similar àquele para criação de um domínio de segundo nível, exceto que a autoridade central agora é local (ou, mais precisamente, dentro de sua própria organização). Especificamente, os passos são os seguintes:

- Escolha um nome que seja exclusivo no contexto local
- Identifique dois ou mais hosts como servidores de seu novo domínio
- Coordene-se com o administrador do domínio pai

Domínios pai devem fazer uma verificação para garantir que os servidores de nomes de um domínio filho estejam ativos e funcionais antes de fazer a delegação. Se os servidores não estiverem funcionando, resultará em uma "delegação defeituosa" e pode ser que você receba um e-mail desagradável solicitando a correção do seu procedimento DNS. A página 331 aborda as delegações defeituosas em mais detalhes.

15.5 FUNCIONAMENTO DO DNS

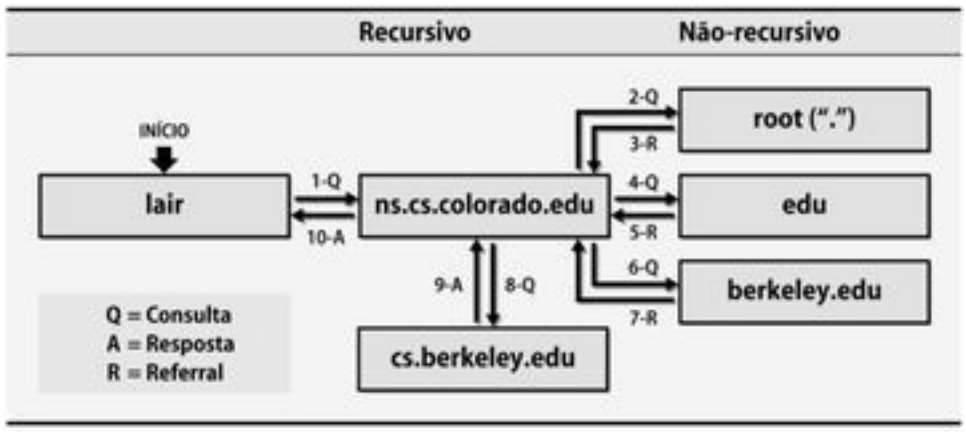
Cada host que usa DNS é um cliente do sistema ou, simultaneamente, um cliente e um servidor. Se você não planeja executar nenhum servidor DNS, não é essencial ler algumas seções a seguir (pule para *Configuração do conversor* na página 291), embora elas ajudem a desenvolver um entendimento mais sólido da arquitetura do DNS.

Delegação

Todos os servidores de nomes lêem as identidades dos servidores-raiz a partir de um arquivo de configuração local. Por sua vez, os servidores-raiz têm conhecimento dos domínios com, org, edu, fi, de e outros domínios de primeiro nível. Mais abaixo na cadeia, edu tem conhecimento de colorado.edu, com sabe da existência de admin.com e assim por diante. Cada zona é capaz de delegar autoridade sobre seus subdomínios para outros servidores.

Vejamos um exemplo real. Suponha que queiramos pesquisar o endereço da máquina vangogh.cs.berkeley.edu na máquina lair.cs.colorado.edu. O host lair solicita seu servidor de nomes local, ns.cs.colorado.edu, para encontrar a solução. A Ilustração A demonstra os eventos subseqüentes.

Ilustração A Processo de consultas DNS para vangogh.cs.berkeley.edu



Os números nas setas entre os servidores mostram a ordem dos eventos e uma letra indica o tipo de transação (consulta, referral ou resposta). Fizemos a suposição de que nenhuma das informações necessárias fosse armazenada em cache antes da consulta, exceto pelos nomes e endereços IP dos servidores do domínio-raiz.

O servidor de nomes local não conhece o endereço; além disso, ele não sabe nada a respeito de cs.berkeley.edu ou berkeley.edu ou até mesmo de edu. Ele conhece alguns servidores para o domínio-raiz, porém, e uma vez que ele é um servidor recursivo, ele consulta um servidor-raiz em vangogh.cs.berkeley.edu e recebe uma referral aos servidores para edu.

O servidor de nomes local envia então sua consulta a um servidor edu (solicitando, como sempre, vangogh.cs.berkeley.edu) e recebe uma referral aos servidores para berkeley.edu. Ele então repete a consulta no domínio berkeley.edu. Se o servidor Berkeley não tiver a resposta armazenada em cache, ele retornará uma referral para cs.berkeley.edu. O servidor cs.berkeley.edu é responsável pelas informações solicitadas e retorna o endereço de vangogh.

Quando a poeira abaixa, ns.cs.colorado.edu armazena em cache o endereço de vangogh. Esse servidor também tem dados armazenados em cache nos servidores para edu, berkeley.edu e cs.berkeley.edu.

Armazenamento em cache e eficiência

O armazenamento em cache aumenta a eficiência das pesquisas: uma resposta armazenada em cache é praticamente livre e normalmente está correta porque em geral os mapeamentos de nome de host para endereço raramente mudam. Uma resposta é salva por um período de tempo chamado "Time to Live" (TTL), que é especificado pelo proprietário do registro de dados em questão. A maior parte das consultas é para hosts locais e pode ser solucionada rapidamente. Os usuários também ajudam inadvertidamente a eficiência, pois repetem muitas consultas; após a primeira ocorrência de uma consulta, o restante fica "livre".

Por um longo tempo, o armazenamento em cache era aplicado somente a respostas positivas. Se o endereço ou nome de um host não podia ser localizado, esse fato não era salvo. Um esquema para armazenamento em cache DNS negativo foi descrito na RFC1034, mas ele era incompleto e não amplamente implementado. Um esquema melhor foi delineado na RFC2308 em 1998. Esse esquema foi implementado no BIND 8.2 como um recurso opcional e agora é obrigatório no BIND 9.

Uma medida no servidor-raiz RIPE na Europa indicava que 60% das consultas DNS eram para dados inexistentes (muitas consultas eram para 127.in-addr.arpa ou para serviços Microsoft como nomes de hosts). O armazenamento em cache dessas informações mais abaixo na árvore DNS deveria reduzir drasticamente a carga nos servidores-raiz.

O armazenamento em cache negativo salva respostas dos seguintes tipos:

- Nenhum host ou domínio atende ao nome consultado
- O tipo de dados solicitado não existe para esse host
- O servidor a ser consultado não está respondendo
- O servidor não pode ser alcançado devido a problemas na rede

Os dois primeiros tipos de dados negativos são, por padrão, armazenados em cache por 10 minutos. Você pode aumentar essa duração para três horas com um parâmetro no registro SOA, discutido na página 273, e para uma semana com uma opção BIND.

A maioria das implementações não realiza os dois últimos tipos de armazenamento em cache negativo. Entretanto, o BIND penaliza servidores irresponsáveis e não irá consultá-los desde que outras escolhas estejam disponíveis. Se todos os servidores de uma zona não conseguirem responder, o BIND não armazena em cache esse fato.

Respostas não-autorizadas *podem* ser armazenadas em cache; respostas negativas autorizadas devem ser *obrigatoriamente* armazenadas em cache. O BIND segue essas diretrizes com base nas RFCs, mas os computadores com Windows parecem implementar as TTLs de forma seletiva, pelo menos no que tange ao armazenamento negativo. Eles utilizam o valor padrão correto (o mínimo do registro SOA) na primeira vez em que uma consulta retorna NXDOMAIN (no such domain), depois reiniciam o TTL para 15 minutos e deixam que ele se expire normalmente a partir daí.

Um servidor de nomes recebe freqüentemente múltiplos registros DNS em resposta a uma consulta. Por exemplo, uma consulta para os servidores de nomes do domínio-raiz receberia uma resposta que listaria todos os 13 servidores-raiz. Qual deles seu servidor deveria consultar?

Quando o servidor de nomes BIND precisa optar entre vários servidores remotos, todos os quais autorizados para um dado domínio, ele determina primeiramente o tempo de percurso de ida e volta (Round Trip Time – RTT) para cada servidor. A seguir ele ordena os servidores em "baldes" de acordo com seus RTTs e seleciona um servidor do balde mais rápido. Servidores dentro de um balde são tratados como equivalentes e são utilizados em uma forma *round robin*.

Você pode obter uma forma primitiva, mas eficiente, de balanceamento de carga atribuindo um único nome de host a vários endereços IP (que, na realidade, são máquinas diferentes):

```
www         IN      A       192.168.0.1
            IN      A       192.168.0.2
            IN      A       192.168.0.3
```

Na verdade, servidores Web ocupados como o Yahoo! ou o Google não são uma única máquina; eles são apenas um único nome no DNS.[6] Um servidor de nomes que tem múltiplos registros para o mesmo nome e o mesmo tipo de registro retorna todos eles ao cliente, mas na forma de rodízio. Por exemplo, a ordem round robin para os registros A acima seria 1, 2, 3 para a primeira consulta; 2, 3, 1 para a consulta seguinte; 3, 1, 2 para a terceira e assim por diante.

6. A última vez que verificamos, o Google tinha mais de 400.000 máquinas Linux (a empresa não irá informar, mas conseguimos uma estimativa fazendo uma pesquisa no Google) e o Yahoo!, mais de 100.000 máquinas FreeBSD.

O protocolo DNS estendido

A definição do protocolo DNS original data do final dos anos 80 e utiliza tanto UDP quanto TCP na porta 53. O UDP é utilizado tipicamente para consultas e respostas e o TCP para transferências de zonas entre servidores mestres e servidores escravos. Infelizmente, o tamanho máximo do pacote que garantidamente funcionará em todas as implementações do UDP é de 512 bytes, que é muito pequeno para alguns dos novos recursos de segurança DNS (por exemplo, DNSSEC) que devem incluir assinaturas digitais em cada pacote.

A restrição de 512 bytes também afeta o número e os nomes dos servidores-raiz. Para fazer com que todos os dados de servidor-raiz caibam em um pacote UDP de 512 bytes, o número de servidores-raiz é limitado a 13, e cada servidor é nomeado apenas com uma única letra do alfabeto.

Muitos conversores enviam primeiramente uma consulta UDP, depois, se receberem uma resposta truncada, eles enviam novamente a consulta via TCP. Esse procedimento contorna o limite de 512 bytes, porém, é ineficiente. Você poderia imaginar que o DNS deve simplesmente descartar o UDP e utilizar o TCP a todo o momento, mas as conexões TCP são muito mais custosas. Uma troca de servidor de nomes UDP pode ser tão curta quanto dois pacotes: uma consulta e uma resposta. Uma troca de informações TCP envolve pelo menos sete pacotes: um handshake de três vias para iniciar a conversa, uma consulta, uma resposta e um handshake final para fechar a conexão.

15.6 NOVIDADES NO DNS

Os desenvolvimentos mais recentes na área de DNS estão relacionados a domínios políticos em vez de domínios técnicos. A VeriSign, a empresa de registro que costumava ter o monopólio do registro de nomes de domínio e que é atualmente responsável pelas zonas com e net, adicionou um registro de endereços curingas a essas zonas. Isso fez com que cada usuário que digitasse erroneamente um nome de domínio fosse direcionado a um site mantido por um dos anunciantes da VeriSign. O serviço era conhecido como Site Finder.

A comunidade da Internet reclamou da injustiça de tudo isso, assim o ISC adicionou uma opção `delegation-only` ao BIND que reconhecia os resultados desses curingas e retornava uma reposta "no such domain" mais exata em vez dos endereços abençoados pela VeriSign. Essa correção era razoável na maioria dos domínios de primeiro nível, mas não em todos, assim uma cláusula de exceções foi posteriormente adicionada para fornecer controle mais refinado. Abordamos essas novas opções BIND na página 298. Após mais ou menos um mês de reclamações, a VeriSign removeu os registros curingas e desativou o serviço. Depois da mediação dos seus advogados, é possível que ela o ative novamente. O IETF pode acabar reforçando as especificações para impedir completamente o uso de registros curingas.

Várias mudanças significativas foram feitas no DNS ao longo dos últimos poucos anos. Em particular, os padrões relacionados ao DNS para o IPv6 e segurança DNS foram radicalmente alterados pelo IETF, tornando a discussão desses tópicos nas edições anteriores deste livro totalmente errada. A Tabela 15.4 lista as principais alterações e fornece um guia para as páginas em que elas são discutidas detalhadamente.

Tabela 15.4 Desenvolvimentos recentes no DNS e no BIND

Página	RFCs	Descrição
388	3492	Nomes de domínios internacionalizados via Punycode
389	2671	EDNS0, mudanças e extensões de protocolo
394	1996	Notificação assíncrona de mudanças de zonas
400	2317	Delegação in-addr sem classes (o hack do CNAME)
402	2782, 3958	Registros SRV para localização de serviços
404	–	Registros AAAA para endereços IPv6 (o A6 está obsoleto)
405	2672–3	Registros DNAME foram abandonados
405	–	ip6.arpa para mapeamentos IPv6 reversos; ip6.int abandonado
–	3596, 3646	Suporte IPv6
447	1995	Transferências de zonas incrementais
448	2136	Atualização dinâmica (para sites que utilizam DHCP)
453	2845, 2930, 3645	Assinaturas de transações TSIG/TKEY e intercâmbio de chaves
456	3225-6, 4033-5	DNSSEC, autenticação e segurança para dados de zona[a]

a. Totalmente refeito em 2004.

Algumas dessas novas características são projetos enormes que o IETF ainda não terminou de padronizar. Os grupos de trabalho que estão criando os padrões têm bons escritores, porém carecem de batalhadores vigilantes em termos de codifi-

cação; algumas das especificações mais recentes podem ser difíceis ou até mesmo impossíveis de serem implementadas. A versão atual do BIND (9.4) inclui a maioria dos novos recursos.

O IPv6 é descrito em mais detalhes no Capítulo 12.

Dois novos recursos maciços, o suporte IPv6 e o DNSSEC, garantem algum comentário. O IPv6 aumenta o comprimento dos endereços IP de 32 bits para 128 bits. Se alguma vez ele for totalmente implementado, terá um impacto enorme sobre a Internet. O BIND 9 suporta os trechos do IPv6 que já foram padronizados até então, mas parece pouco provável que o IPv6 chegue a ser utilizado amplamente durante o período de vida deste livro. Portanto, nossa abordagem sobre o suporte IPV6 do BIND 9 será breve. Há bastante neste capítulo para que você sinta o gostinho, mas não o suficiente para lhe garantir uma migração de seu site para o IPv6 e para configuração do DNS para ele.

O padrão DNSSEC adiciona dados de autenticação ao banco de dados do DNS e aos seus servidores. Ele utiliza criptografia de chave pública para verificar a origem e a integridade dos dados DNS e utiliza o DNS para distribuir chaves, bem como dados de host.

Também foram introduzidos mecanismos de autenticação mais simples, tais como suporte para autenticação por meio do emprego de um "código secreto compartilhado". Contudo, o código secreto compartilhado tem de ser distribuído a cada par de servidores que quiser executar autenticação mútua. Embora seja aceitável um site local com um punhado de servidores, o mesmo não é verdade no nível da Internet. O BIND 9 implementa tanto sistema de chaves públicas DNSSEC quanto o sistema de código secreto compartilhado TSIG (assinaturas transacionais).

Versões do BIND iniciando com 9.3 incluíam as novas especificações para o DNSSEC. Entretanto, à medida que as pessoas começaram a experimentar zonas assinadas há alguns anos, elas perceberam que o sistema DNSSEC original não era prático. Sob o sistema original, uma zona pai assinava a chave de uma zona filha, e as cópias da chave assinadas eram mantidas nas duas zonas. Se a zona filha quisesse alterar sua chave, ela tinha de negociar com a zona pai e solicitar que sua nova chave fosse assinada. Tudo bem. Mas se a zona pai quisesse alterar sua chave, ela tinha de atualizar todas as chaves filhas armazenadas tanto dentro da sua própria zona como em todas as suas zonas filhas. Essa operação revelou-se ingerenciável para grandes zonas, por exemplo, com, porque algumas zonas filhas seriam invariavelmente inacessíveis durante uma atualização. As chaves dessas zonas sairiam de sincronia e impossibilitariam o DNS de verificar as assinaturas.

A solução atual é fazer com que a chave assinada de cada filha resida unicamente na zona filha, mas introduzir um novo registro de recurso na zona pai: DS (*delegation signer*), o assinante da delegação. Abrangeremos o DNSSEC em detalhes no começo da página 318.

A introdução de nomes de domínio internacionalizados, que permite o uso de caracteres não ingleses, prossegue por meio de um hack que mapeia caracteres Unicode de volta para ASCII. Um sistema chamado Punycode realiza o mapeamento única e revogavelmente utilizando um algoritmo conhecido como Bootstring; consulte a RFC3492 para detalhes. Em 2005, órgãos de registro começaram a publicar os nomes Punycode e a maioria dos navegadores implementa alguma forma desse sistema. Infelizmente, algumas questões de segurança e de spoofing relacionadas ao Punycode se manifestaram. Além disso, nomes de domínio internacionalizados reduzem efetivamente o comprimento máximo (tanto por componentes individuais como totais) permitido para nomes de DNS.

O esquema de internacionalização atual contornou uma questão fundamental, o antialiasing, pois ele é muito difícil de resolver. O antialiasing envolve resolver as inconsistências no mapeamento entre caracteres do idioma asiático e o Unicode codificado pelo Punycode que representa esses caracteres no DNS. Se um caractere pode significar uma de 3, 4 ou 10 coisas diferentes no Unicode, especialistas em idiomas devem então concordar com os padrões de tradução, e os caracteres exibidos na tela de um computador devem ser projetados para diferenciar entre os vários significados.

Cada uma destas três grandes questões (IPv6, DNSSEC e internacionalização) aumenta significativamente o tamanho dos registros de dados DNS e, portanto, torna mais provável que o DNS acabe chegando aos limites de tamanho dos pacotes UDP.

Na metade dos anos 90, o protocolo DNS foi reformulado para incluir transferências incrementais de zonas (como um **diff** entre arquivos de zonas antigos e novos, inspirado no programa **patch** de Larry Wall), notificações assíncronas (para informar aos escravos quando os arquivos de dados do mestre foram atualizados) e atualizações dinâmicas (para hosts DHCP). Essas modificações adicionaram recursos, mas não resolveram realmente a questão fundamental de transporte.

No final dos anos 90, o EDNS0 (Extended DNS, versão 0) abordou alguns dos problemas do protocolo DNS na Internet atual. Ele permite aos transmissores anunciarem tamanho de buffer remontado, opções suportadas e versões de protocolo compreendidas. Se o servidor de nomes responder com uma mensagem de erro, o emissor retorna ao protocolo DNS original. O BIND 9 implementa o EDNS0 tanto no servidor como no conversor.

15.7 O BANCO DE DADOS DNS

O banco de dados DNS de um domínio é um conjunto de arquivos de texto mantidos pelo administrador de sistemas no servidor de nomes mestre do domínio. Esses arquivos de texto são normalmente chamados de arquivos de zonas. Eles

contêm dois tipos de entrada: comandos do analisador sintático (coisas como `$ORIGIN` e `$TTL`) e "registros de recursos", ou RRs como são chamados algumas vezes. Apenas os registros de recursos realmente fazem parte do banco de dados; os comandos do analisador sintático simplesmente fornecem algumas maneiras reduzidas para introduzir registros.

Começamos esta seção descrevendo os registros de recursos DNS definidos nas RFCs 1035, 1183, 1876, 2230, 2782, 2930, 3596 e 3658. Postergamos a discussão dos comandos do analisador sintático para a página 282.

Registros de recursos

Cada zona da hierarquia DNS tem um conjunto dos registros de recursos associados a ela. O formato básico de um registro de recurso é

```
[name] [ttl] [classe] tipo dados
```

Os campos são separados por espaço em branco (tabulações ou espaços) e podem conter os caracteres especiais mostrados na Tabela 15.5.

Tabela 15.5 Caracteres especiais utilizados nos RRs

Caractere	Significado
;	Introduz um comentário
@	O nome da zona atual
( )	Permite que os dados se estendam sobre linhas
*	Caractere curinga[a] (somente para o campo *nome*)

a. Consulte a página 277 para algumas instruções de advertência.

O campo *nome* identifica a entidade (normalmente um host ou domínio) que o registro descreve. Se vários registros consecutivos referirem-se à mesma entidade, o nome poderá ser omitido depois do primeiro registro contanto que os registros subseqüentes iniciem com espaço em branco. Se estiver presente, o campo *nome* deve iniciar na coluna um.

Um nome pode ser tanto relativo quanto absoluto. Os nomes absolutos terminam com um ponto e são completos. Internamente, o software lida apenas com nomes absolutos; ele anexa o domínio atual e um ponto a qualquer nome que ainda não termine em um ponto. Esse recurso permite que os nomes sejam mais curtos, mas ele também é um convite a erros.

Por exemplo, no domínio cs.colorado.edu, o nome "anchor" seria interpretado como "anchor.cs.colorado.edu.". Se o nome fosse inserido como "anchor.cs.colorado.edu", a ausência de um ponto final ainda implicaria um nome relativo e o domínio padrão seria anexado, resultando no nome "anchor.cs.colorado.edu.cs.colorado.edu.". Esse é um erro muito comum.

O campo *ttl* (*time to live*) especifica o tempo, em segundos, que o item de dados pode ser armazenado em cache e ainda ser considerado válido. Normalmente ele é omitido, exceto no arquivo hints do servidor-raiz. Seu padrão é o valor configurado pela diretiva `$TTL` logo no início do arquivo de dados para a zona. No BIND 9, a diretiva `$TTL` é exigida. Se não existir nenhuma diretiva `$TTL` no BIND 8, o *ttl* assumirá o padrão de um valor por zona configurado no registro SOA da zona.

Consulte o Capítulo 17 para informações adicionais sobre NIS.

Aumentar o valor do parâmetro *ttl* para cerca de uma semana reduz substancialmente o tráfego na rede e a carga DNS. Entretanto, assim que os registros tiverem sido armazenados em cache fora de sua rede local, você não poderá forçar para que eles sejam descartados. Se você planeja uma renumeração massiva, configure a `$TTL` com um valor baixo (por exemplo, uma hora) para que os registros velhos que foram armazenados em cache em algum ponto na Internet expirem rapidamente.

Alguns sites configuram o TTL nos registros para servidores da Internet como um valor baixo de modo que, se um servidor tiver problemas (falha de rede, falha de hardware, ataque de negação de serviço etc.), os administradores possam responder alterando o mapeamento de nome para endereço IP do servidor. Como os TTLs originais eram baixos, os novos valores se propagarão rapidamente. Por exemplo, o nome google.com tem um TTL de cinco minutos, mas os servidores de nomes do Google têm um TTL de quatro dias (345.600 segundos):

```
google.com.        300        IN    A     216.239.37.99
google.com.        345600     IN    NS    ns1.google.com.
ns1.google.com.    345600     IN    A     216.239.32.10
```

Utilizamos o comando **dig** (**dig @ns1.google.com google.com**) para recuperar esses dados; a saída está truncada aqui.

O BIND 9 impõe um conceito conhecido como harmonização TTL, que força todos os registros em um RRset (isto é, todos os registros do mesmo tipo pertencentes a um único nó) a ter o mesmo TTL. O valor efetivamente utilizado é aquele do primeiro registro de recurso para o par nó/tipo.

classe especifica o tipo de rede. São reconhecidos três valores:

- IN de Internet
- HS de Hesiod
- CH de ChaosNet

O valor padrão para class é IN. Em geral, ele é especificado explicitamente nos arquivos de dados de zona, muito embora ele seja o padrão. Hesiod, desenvolvido no MIT, é um serviço de banco de dados construído sobre o BIND. ChaosNet é um protocolo de rede obsoleto utilizado anteriormente por máquinas Symbolics Lisp.

Hoje, apenas dois fragmentos dos dados de identificação são normalmente guardados na classe ChaosNet: o número de versão do software do servidor de nomes e o nome do host em que o servidor está executando. Essas unidades de dados podem ser extraídas com **dig**, como mostrado na página 285. Os administradores utilizam o número da versão do servidor de nomes para identificar os servidores que precisam de atualizações e eles utilizam a identificação do host para depurar servidores de nomes que são replicados pelo uso de roteamento anycast. Disponibilizar essas informações por meio da classe CH era originalmente um recurso (alguns poderiam dizer um "hack") das implementações BIND, mas agora isso está sendo padronizado pelo IETF como parte do DNS adequado.[7]

São definidos muitos tipos diferentes de registros DNS, mas menos de 10 são os comumente utilizados; o IPv6 acrescenta alguns outros. Dividimos os registros de recursos em quatro grupos:

- Registros de zona – identificam domínios e seus servidores de nomes
- Registros básicos – associam nomes a endereços e encaminham e-mail
- Registros de segurança – acrescentam autenticação e assinaturas aos arquivos de zonas
- Registros opcionais – fornecem informações adicionais sobre hosts ou domínios

O conteúdo do campo *dados* depende do tipo de registro. A Tabela 15.6 lista os tipos de registro comuns.

Tabela 15.6 Tipos de registros DNS

	Tipo	Nome	Função
Zona	SOA	Start Of Authority	Define uma zona DNS
	NS	Name Server	Identifica servidores de zonas, delega subdomínios
Básicos	A	IPv4 Address	Conversão de nome para endereço
	AAAA[a]	IPv6 Address	Conversão de nome para endereço
	PTR	Pointer	Conversão de endereço para nome
	MX	Mail Exchanger	Controla o roteamento de e-mail
Segurança	DS	Delegation Signer	Hash da chave de assinatura de chave da zona filha assinada
	DNSKEY	Public Key – Chave Pública	Chave pública para um nome DNS
	NSEC	Next Secure	Usado com DNSSEC para respostas negativas
	RRSIG	Signature	Conjunto de registros de recursos assinados e autenticados
Opcionais	CNAME	Canonical Name	Apelidos ou aliases para um host
	LOC	Location	Localização e extensão geográfica
	SRV	Services	Dá as localizações de serviços bem conhecidos
	TXT	Text	Comentários ou informações não transcritas

a. Os registros dos endereços IPv6 AAAA e A6 foram rivais no IETF nos últimos anos. AAAA por fim ganhou e passou de obsoleto para o padrão. O A6 agora é rotulado como experimental.

Alguns tipos de registros são obsoletos, experimentais ou não são utilizados amplamente. Consulte a documentação do BIND para uma lista completa. A maioria dos registros é mantida manualmente (editando arquivos de texto), mas os registros dos recursos de segurança exigem processamento criptográfico e, portanto, devem ser gerenciados com ferramentas de software. Esses registros são descritos no começo da seção sobre o DNSSEC, na página 318.

A ordem dos registros de recursos é praticamente arbitrária. O registro SOA para uma zona tinha de ser inicialmente o primeiro, mas esse requisito agora foi relaxado. O SOA é em geral seguido pelos registros NS. Os registros para cada host são normalmente agrupados. É uma prática comum classificar pelo campo *name*, embora alguns sites classifiquem pelo endereço IP para que seja mais fácil identificar endereços não utilizados.

7. Infelizmente, há disputas em relação ao nome sob o qual esses dados devem ser arquivados. Ele deveria ser version.bind, hostname. bind, id-servidor ou…

Como descrevemos cada tipo de registro de recurso em detalhes nas próximas seções, examinaremos algumas amostras de registros nos arquivos de dados do cs.colorado.edu. O domínio padrão nesse contexto é "cs.colorado.edu.", assim um host especificado como "anchor" realmente significa "anchor.cs.colorado.edu.".

O registro SOA

Um registro SOA marca o início de uma zona, um grupo de registros de recursos localizados no mesmo local dentro do espaço de nomes DNS. Esse nó da árvore DNS também é chamado de ponto de delegação ou corte de zona. Conforme discutiremos detalhadamente na página 275, os dados para um domínio DNS normalmente incluem pelo menos duas zonas: uma para converter os nomes de hosts em endereços IP e outras que associam na direção inversa. A árvore DNS tem uma ramificação direta organizada pelo nome e uma ramificação invertida organizada por endereço IP.

Cada zona tem exatamente um registro SOA. O registro SOA inclui o nome da zona, um contato técnico e vários valores de tempo-limite. Um exemplo:

```
; Start of authority record for cs.colorado.edu

@                IN     SOA    ns.cs.colorado.edu. hostmaster.cs.colorado.edu. (
                                       2004111300     ; Serial number
                                       7200           ; Refresh      (2 hours)
                                       1800           ; Retry        (30 minutes)
                                       604800         ; Expire       (1 week)
                                       7200 )         ; Minimum      (2 hours)
```

Nesse caso, o campo *nome* contém o símbolo `@`, que é uma abreviatura para o nome da zona atual. Nesse exemplo, "cs.colorado.edu." poderia ter sido utilizado em seu lugar. O valor de `@` é o nome de domínio especificado na instrução `zone` no arquivo de configuração de servidor de nomes; ele pode ser alterado de dentro do arquivo de zona por meio de uma diretiva do analisador sintático `$ORIGIN` (consulte a página 282).

Esse exemplo não tem campo *ttl*. A classe é IN de Internet, o tipo é SOA e os itens restantes formam o campo *dados*.

"ns.cs.colorado.edu." é o servidor de nomes mestre da zona.

"hostmaster.cs.colorado.edu." é o endereço de e-mail do contato técnico no formato "*usuario.host.*" e não no formato padrão *usuario@host*. Simplesmente substitua esse primeiro ponto por um @ e elimine o ponto final se precisar enviar um e-mail para o administrador de um domínio. Os sites normalmente utilizam um alias como admin ou hostmaster no lugar de um nome de login verdadeiro. O administrador de sistemas responsável pelas tarefas de hostmaster pode mudar e é mais fácil mudar uma entrada no arquivo **aliases** (consulte a página 377) do que ter de alterar todos os seus arquivos de zonas quando precisar atualizar a pessoa para contato.

Os parênteses continuam no registro SOA em várias linhas. Seu posicionamento não é arbitrário no BIND 4 ou 8 — tentamos reduzir a primeira linha dividindo-a antes do endereço para contato, porém isso fez com que BIND não reconhecesse o registro SOA. Em algumas implementações, os parênteses só são reconhecidos nos registros SOA e TXT. O BIND 9 tem um analisador sintático melhor e os parênteses podem ser utilizados em qualquer lugar.

O primeiro parâmetro numérico é o número serial dos dados de configuração da zona. O número serial é utilizado pelos servidores escravos para determinar quando obter dados atualizados. Ele pode ser qualquer inteiro de 32 bits e deve ser incrementado toda vez que o arquivo de dados para a zona for modificado. Muitos sites codificam a data de modificação do arquivo no número serial. Por exemplo, 2004111300 é a primeira modificação para a zona em 13 de novembro de 2004.

Números seriais não precisam ser contínuos, mas devem aumentar monotonicamente. Se, sem querer, você configurar um valor realmente grande no servidor mestre e se esse valor for transferido aos escravos, então corrigir o número serial mestre não funcionará. Os escravos solicitam dados novos somente se o número serial do mestre for maior do que o deles.

Há três maneiras de corrigir esse problema; somente as duas primeiras funcionam no BIND 9.

- Uma terceira maneira de contornar o problema é explorar propriedades do espaço de seqüência no qual os números seriais residem. Esse procedimento envolve acrescentar um valor grande (2^{31}) para o número serial inchado, permitindo que todos os servidores escravos transfiram os dados e depois configurem o número serial para aquilo que você deseja. Essa estranha aritmética, com exemplos explícitos, é vista com detalhes no livro *DNS*, da O'Reilly; a RFC1982 descreve o espaço de seqüências.
- Uma forma furtiva, porém tediosa de corrigir o problema é modificar o número serial do mestre, eliminar os servidores escravos, eliminar os arquivos de dados de backup dos escravos para que eles sejam forçados a recarregar a partir do mestre e reiniciar os escravos. Elas não funcionam apenas para remover e recarregar os arquivos; você precisa eliminar e reiniciar os servidores escravos.

- O BIND 4.9 e o BIND 8 incluem um truque que lhe permite configurar o número serial em zero para um intervalo de atualização e depois reiniciar a numeração. Zero sempre provoca uma recarga; portanto, não se esqueça de configurá-lo para um valor real após cada um dos escravos ter recarregado a zona com o número serial 0.

É um erro comum alterar os arquivos de dados e esquecer de atualizar o número serial. Seu servidor de nome o punirá não propagando suas alterações para os servidores escravos.

As próximas quatro entradas no registro SOA são valores de tempo-limite, em segundos, que controlam por quanto tempo os dados podem ser armazenados em cache em diversos pontos ao longo do banco de dados DNS global. Datas/horas também podem ser expressas em unidades de minutos, horas, dias ou semanas adicionando um sufixo de `m`, `h`, `d` ou `w`, respectivamente. Por exemplo, `1h30m` significa 1 hora e 30 minutos. Os valores de tempo-limite representam um compromisso entre eficiência (é mais barato usar um valor antigo do que buscar um novo) e precisão (valores novos devem ser mais precisos).

Eis uma outra cópia daquele mesmo registro SOA de exemplo, só para que você não tenha de voltar à página anterior:

```
; Start of authority record for cs.colorado.edu

@                IN     SOA    ns.cs.colorado.edu. hostmaster.cs.colorado.edu. (
                                       2004111300     ; Serial number
                                       7200           ; Refresh      (2 hours)
                                       1800           ; Retry        (30 minutes)
                                       604800         ; Expire       (1 week)
                                       7200 )         ; Minimum      (2 hours)
```

O primeiro deles é o tempo-limite *atualizar (refresh)*, que especifica com que freqüência os servidores escravos devem fazer uma verificação no mestre para ver se o número serial da configuração da zona foi alterado. Toda vez que a zona é alterada, os escravos devem atualizar suas cópias dos dados da zona. O escravo compara os números seriais e, se o número serial do mestre for maior, requisita uma transferência de zona para atualizar os dados. Valores comuns para o tempo-limite *atualizar* variam de uma hora a seis horas (3.600 segundos a 21.600 segundos).

Em vez de simplesmente ficar esperando de modo passivo que os servidores escravos expirem, os servidores BIND agora notificam seus escravos toda vez que uma zona for alterada, a menos que o parâmetro `notify` seja especificamente desativado no arquivo de configuração. Os escravos que entendem a notificação se atualizam imediatamente. É possível que uma notificação de atualização seja perdida devido ao congestionamento de rede, assim o tempo-limite de atualização deve ser sempre configurado como um valor razoável.

Se um servidor escravo tentar verificar o número serial do mestre, mas o mestre não responder, o escravo tentará novamente depois que o período de tempo-limite *nova_tentativa (retry)* expirar. Nossa experiência indica que 20 minutos a 60 minutos (de 1.200 segundos a 3.600 segundos) é um bom valor.

Se um servidor mestre estiver inativo por um longo tempo, os escravos tentarão atualizar seus dados muitas vezes, mas sempre falharão. Cada escravo deve acabar decidindo se o mestre nunca responderá e se seus dados certamente estão desatualizados. O parâmetro *expire (expirar)* determina por quanto tempo os escravos continuarão a servir de forma autorizada os dados do domínio na ausência de um mestre. O sistema deve ser capaz de sobreviver se o servidor mestre ficar fora do ar por uma semana, de modo que esse parâmetro deve ter um valor mais longo. Recomendamos de uma semana a um mês.

O parâmetro *minimum (mínimo)* no registro SOA configura o tempo de sobrevida de respostas negativas que são armazenadas em cache.[8] O padrão para respostas positivas (isto é, registros reais) é especificado logo no início do arquivo de zonas pelo uso da diretiva `$TTL`. A experiência sugere valores entre várias horas e alguns dias para `$TTL` e entre algumas horas e um dia para o *minimum*. O valor `$TTL` deve ser maior ou igual a *minimum*.

Os parâmetros `$TTL`, *expire* e *minimum* acabam forçando todos que utilizam DNS a descartar valores de dados antigos. O projeto inicial do DNS baseou-se no fato de que dados de host eram relativamente estáveis e não mudavam com freqüência. Entretanto, hosts móveis e DHCP mudaram essas regras. O BIND está tentando desesperadamente acompanhar esse ritmo fornecendo os mecanismos de transferências de zonas incrementais e de atualização dinâmica descritos a partir da página 311. Para informações adicionais sobre TTLs e um conceito chamado harmonização TTL, Consulte a página 271.

Registros NS

Registros NS (Name Server) identificam os servidores autorizados para uma determinada zona (isto é, todos os servidores mestres e escravos) e delegam subdomínios a outras organizações. Os registros NS normalmente seguem o registro SOA.

O formato é

8. Anterior ao BIND 8.2, o parâmetro *minimum* configurava o tempo padrão de sobrevida para registros de recursos. Ele foi incluído com cada registro e utilizado para expirar os registros armazenados em cache em servidores não autorizados.

```
zona [ttl] IN NS Nome de host
```

Por exemplo:

```
cs.colorado.edu.   IN     NS     ns.cs.colorado.edu.
cs.colorado.edu.   IN     NS     anchor.cs.colorado.edu.
cs.colorado.edu.   IN     NS     ns.cs.utah.edu.
```

Como o nome de zona é o mesmo que o campo *nome* do registro SOA que precede esses registros NS, ele pode ser deixado em branco. Portanto, as linhas

```
                   IN     NS     ns.cs.colorado.edu.
                   IN     NS     anchor.cs.colorado.edu.
                   IN     NS     ns.cs.utah.edu.
```

logo após o registro SOA para cs.colorado.edu são equivalentes.

Para tornar-se visível ao mundo externo, um servidor autorizado de cs.colorado.edu deve ser listado tanto no arquivo de zona para cs.colorado.edu como no arquivo para a zona pai, colorado.edu. Servidores somente de cache não podem ser autorizados; não liste esses servidores. Nenhum parâmetro nos registros NS especifica se um servidor é um mestre ou um escravo. Essas informações são especificadas no arquivo de configuração do servidor de nomes.

BIND utiliza os registros NS de uma zona para identificar servidores escravos quando ele quer enviar notificações para a zona. Esses mesmos registros NS dentro da zona pai (colorado.edu) definem o subdomínio cs e delegam sua autoridade aos servidores de nomes apropriados. Se a lista de servidores de nomes na zona pai não for mantida atualizada com aquelas da zona em si, qualquer servidor novo que for adicionado se tornará um servidor invisível e não será utilizado para responder a consultas provenientes do mundo externo. Essa configuração ocorre às vezes por meio de planejamento e às vezes por esquecimento. Isso não é muito errado contanto que o pai tenha pelo menos um registro NS válido para a zona filha.

Consulte a página 283 para informações adicionais sobre delegação.

Uma rápida análise em nossas delegações revelou um servidor principal para colorado.edu sobre o qual o domínio edu não sabia nada. Faça como eu digo, não como eu faço: verifique as delegações com **dig** para certificar-se de que especificam um conjunto apropriado de servidores (veja a página 330).

Registros A

Os registros A (address) são o cerne do banco de dados DNS. Eles fornecem o mapeamento entre nomes de hosts e endereços IP que foi anteriormente especificado no arquivo **/etc/hosts**. Um host deve ter um registro A para cada uma de suas interfaces de rede. O formato é

```
Nome de host [ttl] IN A ipaddr
```

Por exemplo:

```
anchor             IN     A      128.138.243.100
```

Uma máquina com várias interfaces de rede pode utilizar um único nome de host associado a todas as interfaces ou ter nomes de host distintos para cada uma das interfaces.

Registros PTR

Os registros PTR (pointer) realizam o mapeamento reverso de endereços IP para nomes de hosts. Como acontece com os registros A, um host deve ter um registro PTR para cada interface de rede. Porém, antes de descrevermos os registros PTR, precisamos desviar um pouco do assunto e falar a respeito de um domínio de primeiro nível especial chamado in-addr.arpa.

Nomes de hosts totalmente qualificados podem ser vistos como uma notação na qual a "parte mais significativa" se encontra à direita. Por exemplo, no nome anchor.cs.colorado.edu, anchor se encontra em cs, cs em colorado e colorado em edu. Por outro lado, os endereços IP têm a "parte mais significativa" à esquerda. No endereço 128.138.243.100, o host 100 se encontra na sub-rede 243, que faz parte da rede 128.138.

O domínio in-addr.arpa foi criado para permitir que um conjunto de módulos de software e uma árvore de atribuição de nomes mapeassem endereços IP para nomes de hosts, bem como de nomes de hosts para endereços IP. Domínios abaixo de in-addr.arpa são nomeados como endereços IP com seus bytes invertidos. Por exemplo, a zona para nossa sub-rede 243 é 243.138.128.in-addr.arpa.

O formato genérico de um registro PTR é

```
addr [ttl] IN PTR Nome de host
```

Por exemplo, o registro PTR na zona 243.138.128.in-addr.arpa correspondente ao registro A de anchor acima é

```
100        IN PTR    anchor.cs.colorado.edu.
```

O nome 100 não termina em um ponto e, portanto, é relativo. Mas relativo a quê? Não a "cs.colorado.edu.". Para esse registro de exemplo ser preciso, o domínio padrão tem de ser "243.138.128.in-addr.arpa.".

Você pode configurar o domínio colocando os registros PTR para cada sub-rede em seu próprio arquivo, como neste exemplo. O domínio padrão associado com o arquivo é especificado no arquivo de configuração do servidor de nomes. Uma outra forma de fazer mapeamentos reversos é incluir registros como

```
100.243    IN  PTR   anchor.cs.colorado.edu.
```

com um domínio padrão igual a 138.128.in-addr.arpa. Alguns sites colocam todos os registros invertidos no mesmo arquivo e utilizam diretivas `$ORIGIN` para especificar a sub-rede. Note que o nome de host anchor.cs.colorado.edu deve terminar com um ponto para impedir que 138.128.in-addr.arpa seja anexado ao seu nome.

Como cs.colorado.edu e 243.138.128.in-addr.arpa são regiões diferentes do espaço de nomes DNS, elas formam duas zonas distintas. Cada zona deve ter o seu próprio registro SOA e seus próprios registros de recursos. Além de definir uma zona in-addr.arpa para cada rede real, você também deve definir uma responsável pela rede de loopback, 127.0.0.0.

Tudo isso funciona bem se as sub-redes estiverem dentro dos limites de byte. Mas como tratar dos mapeamentos reversos para uma sub-rede como 128.138.243.0/26? Um truque elegante definido na RFC2317 explora os registros de recursos CNAME para realizar esse feito; consulte a página 278.

Os mapeamentos reversos fornecidos por registros PTR são utilizados por qualquer programa que autentique tráfego que chega pela rede. Por exemplo, **sshd** pode permitir logins remotos sem uma senha se a máquina de origem estiver listada, por nome, em um arquivo **~/.shosts** de um usuário. Quando o host de destino receber uma requisição de conexão, ele saberá qual é a máquina de origem somente pelo seu endereço IP. Ele utiliza o DNS para converter o endereço IP em um hostname, que então é comparado com o arquivo apropriado. **netstat**, **tcpd**, **sendmail**, **sshd**, X Windows e **ftpd** todos realizam mapeamentos reversos para obter os nomes de host dos endereços IP.

É importante que registros A coincidam com seus registros PTR correspondentes. Registros PTR discordantes e ausentes provocam falhas de autenticação que podem deixar o seu sistema lento como uma lesma. Esse problema é irritante por si só; ele também pode facilitar os ataques de negação de serviço contra qualquer aplicativo que requer que o mapeamento reverso corresponda com o registro A.

Registros MX

O sistema de e-mails usa registros de servidores de e-mail (MX) para rotear e-mails de modo mais eficiente. Um registro MX interpõe-se como destino de uma mensagem, na maioria dos casos direcionando-a para um centralizador de e-mails no site do receptor em vez da estação de trabalho do próprio receptor.

O formato de um registro MX é

```
nome [ttl] IN MX preferência host …
```

Dois exemplos são ilustrados a seguir, um para um host que, a menos que esteja inativo, recebe suas próprias mensagens de e-mail e outro para um host que não é capaz de receber e-mails:

```
piper      IN     MX     10 piper
           IN     MX     20 mailhub
           IN     MX     50 boulder.colorado.edu.
xterm1     IN     MX     10 mailhub
           IN     MX     20 anchor
           IN     MX     50 boulder.colorado.edu.
```

Hosts com valores preferenciais baixos são tentados em primeiro lugar: 0 é o mais desejável, e 65.535 é o pior. Nesse exemplo, e-mails endereçados a bob@xterm1 poderiam ser enviados a mailhub se ele estivesse acessível, para anchor como segunda opção e se tanto mailhub quanto anchor estivessem fora do ar, para boulder. Note que o nome de boulder deve ser totalmente qualificado, porque ele não é um membro do domínio padrão (nesse caso, "cs.colorado.edu.").

A lista de preferências e de hosts pode estar inteiramente na mesma linha, mas linhas separadas facilitam a leitura. Deixe "espaço" numérico entre os valores preferenciais para que você não tenha de renumerar se precisar encaixar um novo destino.

Os registros MX são úteis em várias situações:

- Quando se tem um servidor de e-mail centralizado
- Quando o host de destino estiver fora do ar
- Quando o host de destino não estiver diretamente acessível da Internet
- Quando o host de destino não falar SMTP
- Quando o administrador local souber para onde o e-mail deve ser enviado melhor do que fazem seus correspondentes

Na primeira dessas situações, o e-mail é roteado para o centralizador de e-mail, a máquina em que a maioria dos usuários lê seus e-mails. No segundo caso, o e-mail é roteado para um host próximo e encaminhado quando o destino retorna.

Hosts que não estão diretamente acessíveis da Internet (pública) ainda podem ter registros MX. Esses hosts somente com MX poderiam ser máquinas protegidas por um firewall, nomes de domínio hospedados por um ISP, um serviço de hospedagem ou máquinas que não são ligadas todo o tempo. **sendmail** não consegue conectar-se ao host de destino, mas pode receber o correio em local mais próximo se conectando a um dos hosts MX do destino.

A razão final e mais importante de utilizarmos registros MX é que os administradores de sistemas locais provavelmente conhecem a arquitetura de e-mail muito melhor do que seus correspondentes. Eles precisam ter a palavra final sobre como o seu site canaliza o seu fluxo de e-mail.

Cada host que o mundo externo conhece deve ter registros MX. Para hosts secundários, uma ou duas alternativas são suficientes. Um host principal deve ter vários registros. Por exemplo, o conjunto de registros a seguir poderia ser apropriado para um site no qual cada host envia e recebe seus próprios e-mails:

- Um para o próprio host, como primeira opção
- Um centralizador de e-mail departamental como segunda opção
- Um centralizador de e-mail central para o domínio ou domínio pai como backup

O próprio domínio deve ter um registro MX para uma máquina centralizadora de e-mail para que e-mails para *usuario@dominio* funcionem. Obviamente, essa configuração de fato requer que os nomes de usuários sejam exclusivos em todas as máquinas desse domínio. Por exemplo, para sermos capazes de enviar e-mails para evi@cs.colorado.edu, precisamos de uma máquina chamada cs, registros MX em cs.colorado.edu, ou talvez de ambos.

```
cs    IN  MX      10 mailhub.cs.colorado.edu.
      IN  MX      20 anchor.cs.colorado.edu.
      IN  MX      50 boulder.colorado.edu.
```

Uma máquina que aceite enviar e-mails para outro host deve listar esse outro host em seus arquivos de configuração **sendmail**; consulte a página 397 para uma discussão sobre o recurso `use_cw_file` do **sendmail** e do arquivo **local-host-names**.

Registros MX curingas também são algumas vezes vistos no banco de dados DNS:

```
*     IN  MX      10 mailhub.cs.colorado.edu.
```

À primeira vista, esse registro parece capaz de poupar bastante digitação e adicionar um registro MX padrão para todos os hosts. Mas os registros curinga não funcionam bem como esperamos. Eles concordam com qualquer coisa no campo *nome* de um registro de recursos que já *não* esteja listado como um nome explícito em outro registro de recursos.

Portanto, você *não pode* utilizar um asterisco para configurar um valor padrão para todos os seus hosts. Mas, de modo contrário, você pode utilizá-lo para configurar um valor padrão para nomes que não são seus hosts. Essa configuração faz com que muitas mensagens de e-mail sejam enviadas para o seu centralizador somente para ser rejeitadas, pois o nome de host que concorda com o asterisco realmente não pertence ao seu domínio. Portanto, evite registros MX curinga.

Registros CNAME

Os registros CNAME atribuem nomes adicionais a um host. Esses apelidos são comumente utilizados tanto para associar uma função a um host como para reduzir um nome de host longo. O nome real algumas vezes é chamado de nome canônico (daí, "CNAME").

Alguns exemplos:

```
ftp   IN   CNAME   anchor
kb    IN   CNAME   kibblesnbits
```

O formato de um registro CNAME é

```
apelido [ttl] IN CNAME Nome de host
```

Quando o software DNS encontra um registro CNAME, ele interrompe sua consulta em busca do apelido e muda para o nome real. Se um host tiver um registro CNAME, outros registros (A, MX, NS etc.) para esse host devem se referir ao seu nome real, não ao seu apelido.[9] Por exemplo, as linhas a seguir estão certas:

9. Essa regra para CNAMEs foi explicitamente relaxada para o DNSSEC, que adiciona assinaturas digitais a cada conjunto de registros de recursos DNS. O registro RRSIG para o CNAME refere-se ao apelido.

```
colo-gw     IN      A       128.138.243.25
moogie      IN      CNAME   colo-gw
www         IN      CNAME   moogie
```

Entretanto, atribuir uma prioridade de endereço ou de e-mail (com um registro A ou MX) para www ou para moogie nesse exemplo seria incorreto.

Os registros CNAME podem ser aninhados em até oito níveis no BIND. Isto é, um registro CNAME pode apontar para um outro CNAME e esse CNAME pode apontar para um terceiro CNAME e assim por diante, até no máximo sete vezes; o oitavo destino deve ser o nome de host real com um registro A.

Em geral, você pode evitar completamente CNAMEs usando apenas registros A para o nome real do host e seus apelidos.

O hack do CNAME

Consulte a página 201 para informações adicionais sobre CIDR.

Os CNAMEs também são utilizados para forçar a semântica existente do DNS a suportar zonas reversas para redes que não estão em sub-redes em um limite de byte. Antes de o endereçamento CIDR ser um lugar comum, a maioria das atribuições de sub-rede era em limites de byte ou dentro da mesma organização e as delegações reversas eram fáceis de ser gerenciadas. Por exemplo, se a rede 128.138, classe B, fosse dividida em sub-redes para um conjunto de redes do tipo classe C, cada sub-rede criaria um pequeno pacote para o domínio in-addr.arpa. A zona reversa para a sub-rede 243 seria 243.138.128.in-addr.arpa.

Mas o que acontece se a sub-rede 243 for ainda mais dividida, digamos, em quatro trechos como uma rede /26? Se todos os quatro trechos forem atribuídos à mesma organização, não há realmente nenhum problema. As quatro sub-redes ainda poderão compartilhar um único arquivo contendo todos os seus registros PTR. Porém, se a sub-rede 243 for atribuída a um ISP que quer delegar cada rede /26 a um cliente diferente, se faz necessária uma solução mais complexa. O ISP deve manter os registros reversos em nome de cada cliente ou então deve encontrar uma maneira de pegar o terceiro octeto do endereço IP (243 nesse caso) e dividi-lo em quatro trechos diferentes que podem ser delegados independentemente.

Quando um limite administrativo localiza-se no meio de um byte, você terá de ser habilidoso. Você também deverá trabalhar em sintonia com o domínio acima ou abaixo de você. O truque é o seguinte: para cada endereço de host possível na zona in-addr.arpa natural, acrescente um CNAME que desvie a procura para uma zona controlada pelo proprietário da sub-rede adequada. Esse esquema produz arquivos de zonas confusos no pai, mas permite que você delegue autoridade aos verdadeiros usuários de cada sub-rede.

Eis o esquema nos mínimos detalhes. A organização pai (em nosso caso, o ISP) cria registros CNAME para cada endereço IP possível com um componente dissimulador adicional (trecho separado por ponto) que representa a sub-rede. Por exemplo, na situação da rede /26 descrita há pouco, o primeiro quarto dos endereços teria um componente "0-63", o segundo quarto teria um componente "64-127" e assim por diante. Eis como ele ficaria:

```
$ORIGIN 243.138.128.in-addr.arpa.
1           IN      CNAME   1.0-63
2           IN      CNAME   2.0-63
...
63          IN      CNAME   63.0-63
64          IN      CNAME   64.64-127
65          IN      CNAME   65.64-127
...
```

Para delegarmos o trecho 0-63 da zona reversa ao cliente que foi designado àquela sub-rede, adicionamos os seguintes registros NS:

```
0-63        IN      NS      ns1.customer1.com.
0-63        IN      NS      ns2.customer1.com.
...
```

O site de customer1.com teria um arquivo de zonas que conteria os mapeamentos reversos para a zona 0-63.243.138.128.in-addr.arpa.

Por exemplo,

```
1           IN      PTR     host1.customer1.com.
2           IN      PTR     host2.customer1.com.
...
```

Adicionando um componente extra, criamos um novo "corte" no qual podemos realizar a delegação. Quando, por exemplo, alguém procurar o mapeamento reverso para 128.138.243.1, o registro CNAME em 1.243.138.128.in-addr.arpa redirecionará a pesquisa para o nome 1.0-63.243.138.128.in-addr.arpa, e esse nome será controlado pelo cliente.

Os arquivos do cliente são organizados; somente o ISP é que deve lidar com aquela confusão de configuração. Mas as coisas podem se tornar ainda mais complicadas. O próprio Customer1 poderia ser um ISP que quisesse subdividir ainda mais seus endereços. Tudo bem: o BIND suporta cadeias de CNAME de até 8 links de profundidade e, como um byte tem apenas oito bits, nunca poderemos ultrapassar esse limite. O emprego de cadeias CNAMEs não é encorajado, mas isso não é proibido pelas RFCs; elas realmente desaceleram o processo de determinação de nomes, pois cada link em uma cadeia CNAME faz com que o link seja seguido e uma nova consulta para o destino seja iniciada.

Logo que o hack do CNAME começou a ser utilizado, o comando `$GENERATE` foi adicionado ao repertório do BIND para facilitar a criação de registros de recursos na zona pai. Por exemplo, estas linhas produzem os registros para a primeira sub-rede:

```
$ORIGIN 243.138.128.in-addr.arpa.
$GENERATE 0-63 $ CNAME $.0-63
0-63       NS  ns1.customer1.com.
0-63       NS  ns2.customer1.com.
```

O `$` no comando `$GENERATE` varia de 0 a 63 e cria 64 registros CNAME diferentes. As outras três redes /26 seriam manipuladas de maneira similar.

Registros LOC

Registros LOC estão definidos na RFC1819.

Um registro LOC descreve a localização geográfica e, opcionalmente, o tamanho físico (diâmetro) de um objeto DNS. Os registros LOC não têm atualmente nenhum efeito sobre a operação técnica da Internet e nenhum software padrão os pesquisa. Entretanto, uma série de empregos potenciais interessantes para as informações foi sugerida, inclusive rastreamento e otimização de rotas, mapeamento automatizado e pesquisa de redes.

O formato é

```
nome [ttl] IN LOC lat lon [alt [tamanho [ph [pv]]]]
```

A latitude e a longitude são fornecidas como graus, minutos e segundos separados por espaços e seguidos por `N`, `S`, `E` ou `W`. Os segundos podem ser omitidos; se forem, os minutos também poderão ser omitidos.

Os demais campos são todos especificados em centímetros (sem sufixo) ou metros (`m`). *alt* é a altitude do objeto, *tamanho* é o diâmetro da esfera que circunda o objeto, *ph* é a precisão horizontal da medida e *pv* é a precisão vertical. O tamanho padrão é de um metro, e as precisões horizontal e vertical padrão são de, respectivamente, 10 metros e 10 quilômetros.

Eis um exemplo para caida.org em San Diego, Califórnia:

```
caida.org. IN     LOC    32 53 01 N 117 14 25 W 107m 30m 18m 15m
```

Muitas das ferramentas para visualização gráfica escritas pelo CAIDA (Cooperative Association for Internet Data Analysis) requerem dados de latitude e de longitude e os sites são encorajados a incluir essas informações em seus DNSs. Entretanto, se você for paranóico e executar um servidor de alta visibilidade ou ISP, talvez não queira que o público em geral saiba a localização exata de suas máquinas. Em tais situações, recomendamos que você utilize valores inexatos com um grande parâmetro de precisão horizontal. Registros LOC imprecisos ainda são de valia para o pessoal de pesquisa em redes, mas oferecem algum anonimato.

Registros SRV

Um registro SRV especifica a localização de serviços dentro de um domínio. Por exemplo, o registro SRV permite que você consulte um domínio remoto diretamente e pergunte pelo nome de seu servidor FTP. Até agora, grande parte é adivinhação. Para entrar em contato com o servidor FTP de um domínio remoto, você tinha de torcer que os administradores remotos seguissem o costume atual e adicionassem um CNAME para "ftp" a registros DNS de seus servidores.

Os registros SRV fazem mais sentido do que os CNAMEs para essa aplicação e são, certamente, uma maneira melhor para administradores de sistemas deslocarem serviços e controlarem seus empregos. Entretanto, os registros SRV devem ser pesquisados explicitamente e analisados sintaticamente pelos clientes, de modo que levará um tempo antes de seus efeitos realmente serem sentidos. Eles são utilizados extensamente pelo Windows.

Os registros SRV lembram registros MX genéricos com campos que permitem ao administrador DNS local dirigir e balancear a carga de conexões do mundo externo. O formato é

```
serviço.proto.nome [ttl] IN SRV alvo da porta pri wt
```

onde *serviço* é um serviço definido no banco de dados de números atribuídos IANA (consulte www.iana.org/numbers.htm), *proto* é `tcp` ou `udp`, *nome* é o domínio para o qual o registro SRV se refere, *pri* é uma prioridade estilo MX, *wt* é um peso utilizado para balanceamento de cargas entre diversos servidores, *porta* é a porta na qual o serviço é executado e *alvo* é o nome de host do servidor que fornece esse serviço. O registro A do destino normalmente é retornado de maneira automática com a resposta a uma consulta SRV. Um valor 0 para o parâmetro *wt* significa que não precisa ser feito nenhum balanceamento de carga especial. Um valor de "." para o alvo significa que o serviço não é executado nesse site.

Eis um exemplo extraído das RFCs 2052 e 2782 (em que o SRV é definido) e adaptado ao domínio cs.colorado.edu:

```
_ftp._tcp          SRV    0  0  21  ftp-server.cs.colorado.edu.

; don't allow finger anymore (target = .)
_finger._tcp       SRV    0  0  79  .

; 1/4 of the connections to old box, 3/4 to the new one
_ssh._tcp          SRV    0  1  22  old-slow-box.cs.colorado.edu.
                   SRV    0  3  22  new-fast-box.cs.colorado.edu.

; main server on port 80, backup on new box, port 8000
_http._tcp         SRV    0  0  80  www-server.cs.colorado.edu.
                   SRV    10  0  8000  new-fast-box.cs.colorado.edu.

; so both http://www.cs.colo… and http://cs.colo… work
_http._tcp.www     SRV    0  0  80  www-server.cs.colorado.edu.
                   SRV    10  0  8000  new-fast-box.cs.colorado.edu.

; block all other services (target = .)
*._tcp             SRV    0  0   0  .
*._udp             SRV    0  0   0  .
```

Esse exemplo ilustra o emprego tanto do parâmetro de peso (para SSH) quanto do parâmetro de prioridade (HTTP). Ambos os servidores SSH serão utilizados, com o trabalho dividido entre eles. O servidor HTTP de backup será utilizado somente quando o servidor principal não estiver disponível. O serviço **finger** não é incluso, assim como não são outros serviços não mencionados explicitamente. O fato de que o daemon **finger** não aparece no DNS não significa que ele não esteja em execução, mas sim que você não foi capaz de localizar o servidor por meio do DNS.

WKS (Well-Known Services) era um antigo registro DNS relacionado com serviços que não pegou. Em vez de apontar você para o host que fornecia um determinado serviço para um domínio, ele listava os serviços fornecidos por um determinado host. O WKS parecia ser um tanto inútil e também era considerado um risco em termos de segurança. Não foi amplamente adotado.

Registros TXT

Um registro TXT adiciona texto arbitrário aos registros DNS de um host. Temos, por exemplo, um registro TXT que identifica nosso site:

```
 IN TXT    "University of CO, Boulder Campus, CS Dept"
```

Esse registro vem logo após os registros SOA e NS para a zona "cs.colorado.edu." e também herda o campo *nome* deles.

O formato de um registro TXT é o seguinte:

```
name [ttl] IN TXT informação …
```

Todos os itens *informação* têm de ser colocados entre aspas. Você pode utilizar uma string entre aspas ou então várias strings, cada uma entre aspas. Certifique-se de utilizar aspas nas duas extremidades — aspas ausentes provocam uma devastação nos seus dados DNS porque todos os registros entre as aspas ausentes e a próxima ocorrência de uma aspa desaparecem misteriosamente.

Muitos administradores utilizam registros TXT para publicar os nomes das máquinas nos seus sites a partir das quais os e-mails legítimos se originam. Outros sites podem utilizar os chamados registros SPF (Sender Policy Framework) para ajudar a identificar e eliminar spam. As informações SPF residem dentro da parte *info* dos registros TXT; não há nenhum tipo separado de registro DNS para o SPF.

Eis alguns exemplos:

```
sendmail.com.      IN      TXT     "v=spf1 ip4:209.246.26.40 ip4:63.211.143.38 ip4:209.246.26.36
    ip4:209.246.26.12 ip4:209.246.26.18 ip4:209.246.26.10 ~all"
example.com.       IN      TXT     "v=spf1 MX PTR -all"
```

Na primeira linha, o domínio sendmail.com está listando os endereços IP dos servidores de correio. Se o correio supostamente se originar do sendmail.com, mas não estiver sendo enviado de uma máquina cujo endereço IP está na lista de registros SPF, o e-mail é forjado e deve ser descartado. A segunda linha exige que a máquina enviando o e-mail tenha um registro MX correspondente e um registro PTR correspondente no DNS a fim de ser validado.

A cláusula `v=spf1` refere-se à versão do protocolo SPF; atualmente apenas a versão 1 está definida ou implementada. Vários agentes de transporte de correio (incluindo **sendmail**, Postfix e **exim**) suportam processamento SPF. Há muitos outros penduricalhos que podem ser invocados por meio de registros SPF; consulte a página 414 para uma discussão mais completa.

Os registros TXT não possuem nenhuma ordem intrínseca. Se você usar vários deles para adicionar um parágrafo de informações ao seu DNS, eles podem se misturar inteiramente quando **named** e UDP já os tiverem utilizado.

Registros de recursos no IPv6

Consulte o Capítulo 12 para uma discussão mais detalhada sobre o IPv6.

O IPv6 é uma nova versão do protocolo IP. Foram gastos mais de 10 anos no processo da especificação e ela ainda não foi concluída.[10] A razão original do IPv6 foi impulsionada pela necessidade de mais endereços IP de rede. Entretanto, as soluções "tapa-buraco" para esse problema — CIDR, NAT e controle mais estrito dos endereços — foram tão bem-sucedidas que uma migração em massa para o IPv6 revelou-se não tão essencial quanto pensada originalmente. A adoção do IPv6 agora acontece na Ásia, onde endereços IPv4 estão mais concentrados. A próxima geração de celulares, que podem ter endereços IP, também pode ajudar a favorecer o uso do IPv6.

As primeiras propostas para suporte IPv6 no DNS passaram por dificuldades a fim de suportarem posse compartilhada de endereços IPv6 com os registros de recursos A6 e DNAME. Embora esses tipos de registros tornem a renumeração dos endereços mais fácil, eles eram tão complicados que agora o IETF retirou seu apoio ao plano original, favorecendo os registros AAAA muito mais simples para o encaminhamento de mapas e o domínio ip6.arpa para mapeamentos reversos. Não vamos mais descrever registros A6, strings de bits ou registros DNAME neste livro. Se estiver curioso, consulte as edições anteriores ou as RFCs para detalhes sobre como eles devem funcionar.

Registros de encaminhamento IPv6

O formato de um registro AAAA é

```
Nome de host [ttl] IN AAAA  ipaddr
```

Por exemplo:

```
anchor     IN     AAAA   3ffe:8050:201:9:a00:20ff:fe81:2b32
```

Cada trecho separado por dois-pontos corresponde a 4 dígitos hexadecimais, com zeros não significativos normalmente omitidos. Dois-pontos adjacentes significam "zeros suficientes para preencher os 128 bits para um endereço IPv6 completo". Um endereço pode conter no máximo um destes dois-pontos duplos.

Registros reversos no IPv6

Consulte a página 275 para uma discussão sobre a versão IPv4 dos registros PTR.

No IPv4, os mapeamentos reversos residem no domínio in-addr.arpa e os mapeamentos diretos residem nas demais ramificações da árvore de domínios (sob com ou edu, por exemplo). No IPv6, as informações de mapeamento reverso correspondentes a um registro de endereço AAAA são um registro PTR no domínio de primeiro nível ip6.arpa[11].

O formato esmiuçado inverte um registro de endereço AAAA, expandindo cada trecho de endereço separado por dois-pontos para os 4 dígitos hexadecimais completos e depois invertendo a ordem destes dígitos e anexando ip6.int ao final. Por exemplo, o registro PTR correspondente ao nosso exemplo de registro AAAA para anchor seria

```
2.3.b.2.1.8.e.f.f.f.0.2.0.0.a.0.9.0.0.0.1.0.2.0.0.5.0.8.e.f.f.3.ip6.arpa PTR
    anchor.cs.colorado.edu.
```

Certamente não é agradável para um administrador de sistemas ter de digitar ou depurar ou até mesmo ler. Obviamente, em seus arquivos de zonas DNS reais, a instrução `$ORIGIN` ocultaria parte da complexidade.

O IPv6 ainda é jovem, pelo menos sob a perspectiva da implantação. Os escritórios de registros estão apenas começando a atribuir endereços e o processo se tornará mais fluente à medida que for acumulando experiência. Alguns servidores de nomes-raiz começaram recentemente (2004) a anunciar endereços IPv6. Ainda há questões sobre o comportamento padrão.

---

10. Tony Li, um membro ativo da comunidade IETF, descreveu o IPv6 como "muito pequeno, muito recente".
11. A ramificação reversa do IPv6 da árvore para atribuição de nomes era originalmente chamada ip6.int.

Por exemplo, se um nome tiver um endereço IPv4, mas nenhum endereço IPv6, e for consultado por um registro AAAA, ele deve informar "no such record" ou retornar o registro A do IPv4?

Registros relacionados à segurança

Os registros de recursos relacionados aos DNSSEC (DNSKEY, DS, RRSIG e NSEC) abrangem um tópico próprio importante. Discutiremos esses registros na seção sobre segurança do DNS que começa na página 314. Esses registros são bem diferentes da maioria pelo fato de que eles em geral são gerados com ferramentas de software em vez de digitados manualmente.

Comandos em arquivos de zonas

Agora que vimos todos os registros de recursos básicos, vamos examinar os comandos que podem ser embutidos em um arquivo de zona. Na verdade, esses comandos são apenas diretivas do analisador de sintaxe que ajudam a tornar a manutenção dos arquivos de zona mais legível e mais fácil de manter. Os comandos influenciam a maneira como o analisador de sintaxe interpreta os registros subseqüentes ou eles se expandem em múltiplos registros DNS. Depois de um arquivo de zona ter sido lido e interpretado, nenhum desses comandos permanece com uma parte dos dados da zona (pelo menos, não nas suas formas originais).

Há quatro comandos:

```
$ORIGIN nomeDeDomínio
$INCLUDE nomeDeArquivo [original]
$TTL padrãottl
$GENERATE muitosArgumentos
```

Os comandos têm *obrigatoriamente* que começar na primeira coluna e cada um deles deve ocupar uma linha exclusiva. Os comandos `$ORIGIN` e `$TTL` são especificados nas RFCs e devem ser entendidos por todos os servidores de nome; `$INCLUDE` e `$GENERATE` eram originalmente específicos ao BIND, mas são utilizados por algumas outras implementações de DNS.

À medida que o servidor de nomes lê um arquivo de zonas, ele adiciona o domínio padrão (ou "origem") a qualquer nome que não esteja ainda completamente qualificado. A origem é inicialmente configurada como o nome de domínio especificado na instrução `zone` correspondente no arquivo de configuração do servidor de nome. Entretanto, você pode configurar a origem manualmente dentro de um arquivo de zona utilizando a diretiva `$ORIGIN`.

O emprego de nomes relativos em que se espera nomes totalmente qualificados poupa uma grande quantidade de digitação e torna a leitura dos arquivos de zonas muito mais fácil. Por exemplo, os registros reversos para um site classe B subdividido em redes poderiam estar todos em um único arquivo de zonas, com as instruções `$ORIGIN` configurando o contexto para cada sub-rede. Uma instrução como

```
$ORIGIN 243.138.128.in-addr.arpa
```

poderia anteceder os registros para a sub-rede 243.

Muitos sites utilizam a diretiva `$INCLUDE` em seus arquivos de bancos de dados de zonas para separar registros extras de registros de dados, para separar partes lógicas de um arquivo de zonas ou para manter chaves criptográficas em um arquivo com permissões restritas. A sintaxe da diretiva `$INCLUDE` é:

```
$INCLUDE nomeDeArquivo [origem]
```

O arquivo especificado é lido no banco de dados no ponto da diretiva `$INCLUDE`; se uma *origem* for especificada, uma diretiva `$ORIGIN` irá preceder o conteúdo do arquivo sendo lido. Se *nomeDeArquivo* não for um caminho absoluto, ele será interpretado em relação ao diretório inicial do servidor de nomes em execução.

A diretiva `$TTL` configura um valor padrão para o tempo de sobrevida dos registros que o seguem. Ele deve preceder o registro SOA para a zona. As unidades padrão para o valor `$TTL` são segundos, mas você também pode qualificar números com `h` para horas, `m` para minutes, `d` para dias ou `w` para semanas. Por exemplo, todas as linhas a seguir configuram `$TTL` como um dia:

```
$TTL 86400
$TTL 24h
$TTL 1d
```

`$GENERATE`, uma construção relativamente nova, fornece uma maneira simples de gerar uma série de registros similares. Ela serve basicamente para ajudar na geração de associações in-addr.arpa sem classes no estilo RFC2317 (o hack do CNAME para arquivos de zonas reversas), para casos em que os limites da autoridade administrativa não concordem com os limites dos bytes em um endereço IP.

O formato da diretiva `$GENERATE` é

```
$GENERATE início-fim/[passo] lhs tipo rhs [comentário]
```

e as linhas geradas são na forma

```
lhs tipo rhs
```

Os campos *início* e *fim* especificam o intervalo de valores para um único iterador numérico. Uma linha é gerada para cada valor no intervalo. O valor de iterador é incorporado a *le* (lado esquerdo) e *ld* (lado direito) com o caractere `$`. Se você também especificar um passo (*passo*), a iteração se dará em incrementos equivalentes a esse *passo*. *tipo* é o tipo de registro. O BIND 9 suporta `$GENERATE` para os tipos de registro CNAME, PTR, NS, DNAME, A e AAAA. Consulte a página 279 para um exemplo.

Registros "cola": links entre zonas

Cada zona é auto-suficiente com seu próprio conjunto de arquivos de dados, servidores de nomes e clientes. Mas as zonas precisam estar conectadas para que formem uma hierarquia coerente: cs.colorado.edu é uma parte de colorado.edu, e precisamos de alguma linkagem DNS entre eles.

Como as delegações DNS ocorrem somente de domínios pai para domínios filhos, não é necessário que um servidor de nomes saiba qualquer coisa a respeito dos domínios (ou, mais precisamente, sobre as zonas) acima dele na hierarquia DNS. Entretanto, os servidores de um domínio pai devem conhecer os endereços IP dos servidores de nomes para todos os seus subdomínios. Na realidade, *somente* os servidores de nomes conhecidos por parte da zona pai podem ser retornados como referrals em resposta a consultas externas.

Em termos DNS, a zona pai precisa conter os registros NS para cada zona delegada. Pelo fato de os registros NS serem escritos em termos de nomes de hosts e não de endereços IP, o servidor pai também deve ter uma maneira de descobrir os nomes de hosts, seja por meio de uma consulta DNS normal (se isso não cria um loop de dependência), seja por meio de cópias dos registros A apropriados.

Há duas maneiras pelas quais você pode atender a essa exigência: incluindo os registros necessários diretamente ou utilizando zonas stub.

Com o primeiro método, você pode simplesmente incluir os registros NS e A necessários na zona pai. Por exemplo, o arquivo de zonas colorado.edu poderia conter os seguintes registros:

```
; subdomain information

cs          IN      NS      ns.cs.colorado.edu.
            IN      NS      piper.cs.colorado.edu.
            IN      NS      ns.atrust.com.
ee          IN      NS      ns.ee.colorado.edu.
            IN      NS      ns.cs.colorado.edu.

; glue records

ns.cs       IN      A       128.138.243.151
piper.cs    IN      A       128.138.204.4
ns.ee       IN      A       128.138.200.1
```

Os registros A "estrangeiros" são chamados de registros de adesão, pois realmente não pertencem a essa zona. Eles são reproduzidos aqui somente para conectar o novo domínio à árvore de atribuição de nomes. Registros de adesão ausentes ou incorretos deixarão parte de seu espaço de nomes inacessível, e os usuários que tentarem atingi-lo se depararão com erros "host unknown" (host desconhecido).

É um erro comum incluir registros de adesão para nomes de hosts que não precisem deles. Por exemplo, ns.atrust.com no exemplo anterior pode ser descoberto com uma consulta DNS comum. Inicialmente, um registro A seria desnecessário, mas posteriormente ele poderia tornar-se bem enganoso se o endereço de ns.atrust.com mudasse. A regra prática é incluir registros A somente para hosts que se encontrem dentro do domínio atual ou qualquer um de seus subdomínios. As versões atuais do BIND ignoram registros de adesão desnecessários e registram em log a presença deles na forma de erro.

O esquema recém-descrito é a maneira padrão de se conectar zonas, mas requer que o filho permaneça em contato com o pai e informe o pai sobre quaisquer alterações ou acréscimos ao seu conjunto de servidores de nomes. Como as zonas pai e filho normalmente são mantidas por sites diferentes, as atualizações normalmente se transformam em uma operação manual tediosa que requer coordenação entre os diversos níveis administrativos. Um corolário é que, no mundo real, esse tipo de configuração normalmente se encontra desatualizado.

A segunda maneira de manter links é utilizar zonas stub. Uma zona stub é basicamente a mesma coisa que uma zona escrava, porém ela só inclui os registros NS da zona. Atualizar automaticamente a zona stub elimina a necessidade da comunicação entre a zona pai e a filha. Uma dica importante é que zonas stub devem ser configuradas de uma maneira

idêntica tanto nos servidores mestre como nos servidores escravos *da zona pai*. A comunicação com seu domínio pai talvez seja mais fácil de manter manualmente e verificar sua configuração pelo menos algumas vezes no ano (especialmente se ela for local).

Você pode usar o comando **dig** para verificar quais de seus servidores seu domínio pai está anunciando no momento. Primeiramente execute

```
$ dig domínio pai ns
```

para determinar os servidores de nomes de seu domínio pai. Escolha um e execute

```
$ dig @servidor-de-nomes.domínio pai domínio filho ns
```

para ver a sua lista de servidores de nomes públicos. Eis um exemplo real da saída gerada por **dig** onde parte dela foi eliminada para fins de concisão:

```
$ dig colorado.edu ns
;;      ...
;; ANSWER SECTION:
colorado.edu.           5h9m22s IN NS   ns1.westnet.net.
colorado.edu.           5h9m22s IN NS   boulder.colorado.edu.
colorado.edu.           5h9m22s IN NS   cujo.colorado.edu.

$ dig @boulder.colorado.edu cs.colorado.edu ns
;;; ANSWER SECTION:
cs.colorado.edu.        6H IN NS        cs.colorado.edu.
cs.colorado.edu.        6H IN NS        huizil.cs.colorado.edu.
cs.colorado.edu.        6H IN NS        anyns.pch.net.
cs.colorado.edu.        6H IN NS        pacifier.com.
```

Apenas quatro servidores para o domínio cs.colorado.edu são visíveis do mundo externo. Um comando **dig** realizado de dentro do departamento resultaria em uma lista diferente:

```
;; ANSWER SECTION:
cs.colorado.edu.        2H IN NS        cs.colorado.edu.
cs.colorado.edu.        2H IN NS        moet.cs.colorado.edu.
cs.colorado.edu.        2H IN NS        piper.cs.colorado.edu.
cs.colorado.edu.        2H IN NS        anchor.cs.colorado.edu.
cs.colorado.edu.        2H IN NS        vulture.cs.colorado.edu.
```

Observe que os valores TTL variam (2 horas *versus* 6 horas) dependendo de onde vem uma consulta, de dentro ou fora do departamento. Isso ocorre porque a instrução `view` do BIND foi utilizada para definir visualizações internas e externas dos dados; consulte a página 304. (Os dois valores utilizam curtos períodos de tempo; entre alguns dias e uma semana seria uma escolha melhor.)

Uma situação na qual as zonas stub são muito úteis é aquela em que sua rede interna utiliza o espaço de endereços IP privados RFC1918 e você precisa manter as delegações RFC1918 em sincronia. O exemplo do isc.org que começa na página 309 utiliza zonas stub extensamente.

Algumas sutilezas das zonas stub (sutilezas?) são dignas de nota:

- As zonas stub não são cópias autorizadas dos dados da zona, e os servidores stub não devem ser listados entre os registros NS da zona.
- Como os servidores stub não são listados em registros NS, eles não são notificados automaticamente quando os dados da zona mudam. Servidores stub simplesmente esperam que a zona seja atualizada no final do intervalo de atualização especificado no registro SOA da zona. Se esse intervalo for longo, ele potencialmente poderá resultar em delegações deficientes (consulte a página 331).
- Teoricamente, não tem valor algum para um servidor de nomes ter cópias dos registros NS de uma zona se ele também não puder obter os registros A concordantes. Mas o próprio servidor de nomes pode se auto-inicializar utilizando o endereço IP do mestre a partir do seu arquivo de configuração (a cláusula `masters` da instrução `zone`, consulte a página 301).
- Por que se limitar aos registros NS? Por que não simplesmente ser um servidor secundário para os subdomínios? Isso também funciona. Entretanto, se cada servidor do domínio pai também for um servidor de um domínio filho, então referral nunca será feita para os servidores abaixo. Os servidores do domínio pai fornecerão todo o serviço DNS para o subdomínio. Talvez isso seja o que você deseja, talvez não.

Agora abordamos a maioria das informações aplicadas ao Domain Name System em geral e ao seu banco de dados. Na próxima seção, continuamos a discussão sobre o DNS com detalhes de configuração específicos à implementação BIND.

15.8 O SOFTWARE BIND

O BIND, o sistema Berkeley Internet Name Domain, é um pacote de software de código-fonte aberto do ISC, o Internet Systems Consortium, que implementa o protocolo DNS para sistemas Linux, UNIX, Mac OS e Windows.

Versões do BIND

Havia três versões principais do BIND: BIND 4, BIND 8 e BIND 9. O BIND 4 está no mercado desde o final dos anos 80 (correspondendo grosseiramente ao lançamento das RFCs 1034 e 1035). O BIND 8 foi lançado em 1997 e o BIND 9, em meados de 2000. Não há nenhum BIND 5, 6 ou 7; o BIND 8 foi uma atualização tão significativa que os autores sentiram que ela merecia um número de versão que fosse o dobro da anterior.[12] Bem, não foi realmente assim … o BIND 8 foi lançado com o 4.4BSD (a Berkeley Software Distribution do UNIX), para o qual todos os números de versão foram elevados para 8. **sendmail** também pulou alguns números e foi para a versão 8 na mesma época.

O BIND 8 incorporou numerosos avanços técnicos que aumentaram a eficiência, robustez e segurança. O BIND 9 aumenta ainda mais a aposta com suporte de multiprocessamento, operação segura com threads, segurança efetiva (criptografia de chaves públicas), suporte IPv6, transferências incrementais de zonas e uma enormidade de outros recursos. Uma nova estrutura de dados (pelo menos nova para o BIND), a árvore vermelha-preta, armazena dados de zonas na memória. O BIND 9 é uma reimplementação e reprojeto completos. Ele isola as partes do código específicas ao sistema operacional, tornando mais fácil portar o BIND para sistemas não-UNIX. As partes internas do BIND 9 são significativamente diferentes, mas seu procedimento de configuração permanece o mesmo. Abordaremos apenas o BIND 9 neste livro.

Descobrindo que versão você possui

Normalmente não parece ocorrer aos fabricantes documentar que versão de um pacote de software externo eles incluíram em seus sistemas, portanto, você pode precisar fazer uma investigação para descobrir exatamente com que versão de software está trabalhando. No caso do BIND, podemos algumas vezes determinar o número de versão com uma consulta furtiva usando **dig**, um comando que vem com o BIND. O comando:

```
$ dig @servidor version.bind txt chaos
```

retorna o número de versão a menos que alguém tenha decidido restringir essa informação alterando-a no arquivo de configuração do BIND. Por exemplo, o comando funciona em isc.org:

```
$ dig @ns-ext.isc.org version.bind txt chaos
version.bind.           0S CHAOS TXT    "9.4.0a0"
```

Mas não funciona em cs.colorado.edu:

```
$ dig @mroe.cs.colorado.edu version.bind txt chaos
version.bind.           0S CHAOS TXT "wouldn't you like to know…"
```

Alguns sites configuram o BIND para esconder seu número de versão baseado na teoria que isso fornece algum grau de "segurança por meio de obscuridade". Realmente não endossamos essa prática, mas isso poderia ajudá-lo a rechaçar alguns dos scripts maliciosos. Consulte a página 295 para uma discussão mais detalhada sobre esse tópico.

A saída dessa consulta inclui "0S". O 0 é o dígito zero e representa o TTL. O S significa segundos, mas esse é o padrão para o TTL e normalmente não é impresso. Talvez devêssemos informar isso como um bug — à primeira vista ele se parece com OS de "*operating system*".

O IETF ocupa-se em padronizar os dados nessa esquisita zona de classe CHAOS e os generaliza para que outras implementações também possam utilizar essa convenção. Muitos já fizeram isso e deixam que o nome da variável permaneça version.bind; outros não gostam que o nome inclua "bind". Como o IETF continua discutindo quais devem ser os nomes das variáveis, o ISC implementou todos os candidatos na sua versão atual. Por exemplo, você também pode consultar na classe CHAOS hostname.bind ou o id.server mais genérico.

Consulte o Capítulo 10 para informações adicionais sobre syslog.

Normalmente, você também pode determinar a versão BIND que você tem inspecionando os arquivos de log em **/var/log**. O daemon de servidor BIND, **named**, registra em log o número da sua versão para o syslog ("daemon" de recurso) à medida que ele é inicializado. **grep** para linhas como esta:

12. Quem disse que os departamentos de marketing e de engenharia não podem conviver pacificamente?

```
Feb 23 00:25:13 senna named[433]: starting BIND 9.4.0a0 -c /var/named/named.ns-ext.conf
```

Se **named** estiver instalado, mas seu sistema normalmente não o inicia em tempo de inicialização, basta executar **named -v** e ele irá gerar a saída do número de versão e encerrar.

A Tabela 15.7 mostra as versões do BIND incluídas nas distribuições de sistemas usadas como exemplo neste livro. No Debian e no Ubuntu, temos a escolha de um pacote **bind** que instala o BIND 8 e um pacote **bind9** que instala o BIND 9. As versões anteriores à 9.3.1 ou 8.4.6 tinham problemas de segurança; esqueça o BIND 4. Usar as versões atuais é a opção mais segura.

Tabela 15.7 Versões do BIND nos sistemas utilizados como exemplo neste livro

Sistema	Versão do sistema operacional	Versão do BIND
ISC	–	9.4.0b
RHEL	4.3	9.2.4 via rpm
Fedora	FC5	9.3.2
SUSE	10.2	9.3.2
Debian	3.2b	8.4.6 ou 9.2.4
Ubuntu	6.06	8.4.6 ou 9.3.2

A maioria dos distribuidores Linux fornece correções de segurança para versões mais antigas.

Componentes do BIND

O sistema BIND tem três componentes:

- Um daemon de servidor de nomes chamado **named** que responde a consultas
- Rotinas de biblioteca que solucionam consultas host contatando os servidores do banco de dados DNS distribuído
- Interfaces de linhas de comandos para o DNS: **nslookup**, **dig** e **host**

No linguajar DNS, um daemon como **named** (ou a máquina na qual ele é executado) é chamado de "servidor de nomes", e o código cliente que o contata é chamado de "conversor". Discutimos brevemente a função de cada componente a seguir, porém adiamos a real configuração de BIND até a página 292.

named: o servidor de nomes BIND

named responde a consultas sobre nomes de hosts e endereços IP. Se **named** não souber a resposta a uma consulta, ele questiona outros servidores e armazena suas respostas em cache. **named** também executa "transferências de zonas" para copiar dados entre servidores de um domínio. (Lembre-se de que uma "zona" é essencialmente um domínio menos seus subdomínios. Os servidores de nomes lidam com zonas, mas "domínio" é usado onde normalmente se quer dizer "zona".)

Os servidores de nomes podem operar em diversos modos diferentes. As distinções entre eles recaem sobre vários eixos, de modo que a categorização final normalmente não é muito ordenada. Para tornar tudo mais confuso, um único servidor é capaz de desempenhar diversos papéis em relação a diferentes zonas. A Tabela 15.8 lista alguns dos adjetivos utilizados para descrever os servidores de nomes. Entradas recuadas são vagamente classificadas sob seus cabeçalhos não-recuados.

Tabela 15.8 Uma taxonomia de servidores de nomes

Tipo de servidor	Descrição
autorizado	Um representante oficial de uma zona
mestre	O servidor principal de uma zona; obtém dados de um arquivo em disco
escravo	Copia seus dados do mestre (também chamado de servidor secundário)
stub	Semelhante a um escravo, mas só copia dados de servidor de nomes (e não dados de host)
distribuição	Um servidor que é visível[a] somente de dentro de um domínio ("stealth server")
não-autorizado [b]	Responde a uma consulta no cache; não sabe se os dados ainda são válidos
cache	Armazena em cache dados de consultas anteriores; normalmente não tem zonas locais
encaminhador	Executa consultas em nome de diversos clientes; constrói um grande cache
recursivo	Consultas em seu nome até ele retornar uma resposta ou um erro
não-recursivo	Refere-se a um outro servidor se ele não conseguir responder a uma consulta

a. Um servidor de distribuição pode ser visível a qualquer um que conheça o seu endereço IP.
b. Estritamente, "não-autorizado" é um atributo de uma resposta a uma consulta DNS, não um servidor.

Essas categorizações se baseiam na fonte de dados de um servidor de nomes (autorizado, cache, mestre, escravo), no tipo de dados salvos (stub), no caminho de consulta (encaminhadores), no tipo de respostas manipuladas (recursivos, não-recursivos) e, finalmente, na visibilidade do servidor (distribuição). As seções a seguir fornecem alguns detalhes adicionais sobre as diferenças mais importantes; as demais diferenças são descritas em outros pontos deste capítulo.

Servidores autorizados e servidores somente para cache

Servidores mestre, escravo e somente para cache são distinguidos por duas características: de onde provêm os dados e se o servidor é autorizado para o domínio.

Cada zona tem um servidor de nomes mestre. O servidor mestre mantém a cópia oficial dos dados da zona em disco. O administrador de sistemas altera os dados da zona editando os arquivos de dados do servidor mestre.

Consulte a página 311 para informações adicionais sobre as transferências de zonas.

Um servidor escravo obtém seus dados do servidor mestre por meio de uma operação de "transferência de zonas". Uma zona pode ter vários servidores de nomes escravos e deve *obrigatoriamente* ter pelo menos um. Um servidor stub é um tipo especial de escravo que carrega somente os registros NS (Name Server, servidor de nomes) do servidor mestre. Consulte a página 301 para uma explicação da razão pela qual você pode desejar esse tipo de comportamento. Uma máquina pode ser ao mesmo tempo um servidor mestre para suas zonas e um servidor escravo para outras zonas. Tal cooperação normalmente gera bons vizinhos DNS.

Um servidor de nomes somente para cache carrega os endereços dos servidores para o domínio-raiz a partir de um arquivo de inicialização e acumula o restante de seus dados armazenando em cache respostas às consultas que ele soluciona. Um servidor de nomes somente para cache não possui nenhum dado próprio e não é autorizado para nenhuma zona, exceto talvez a zona de host local.

Uma resposta autorizada de um servidor de nomes é "garantidamente" acurada; uma resposta de um servidor não-autorizado pode estar desatualizada. Entretanto, uma porcentagem muito grande das respostas desse tipo de servidor é perfeitamente correta. Os servidores mestre e escravo são autorizados para suas próprias zonas, mas não para informações acerca de outros domínios que eles armazenaram em cache. A verdade seja dita, mesmo respostas autorizadas podem ser imprecisas se um administrador de sistemas alterar os dados do servidor mestre e se esquecer de propagar as alterações (por exemplo, não alterar o número serial dos dados).

O servidor mestre deve ser localizado em uma máquina estável, sem muitos usuários, relativamente segura e conectada a um no-break. Deve haver ao menos dois escravos, um dos quais fora da instalação. Escravos na instalação devem residir em redes diferentes e em circuitos de alimentação diferentes. Quando o serviço de nomes pára, todo o acesso normal à rede também pára.

Embora não sejam autorizados, os servidores somente para cache podem reduzir a latência vista por seus usuários e a quantidade de tráfego DNS em suas redes internas. Considere a possibilidade de colocar um servidor somente para cache em cada sub-rede. Na maioria dos sites, as máquinas desktop normalmente passam por um servidor de cache para solucionar consultas sobre hosts na Internet.

No BIND 4 e no BIND 8, não era uma boa idéia usar um único servidor de nomes como seu servidor autorizado para suas próprias zonas e como servidor de cache para seus usuários. Cada **named** era executado com um único banco de dados na memória e poderia ocorrer contaminação cruzada se a memória fosse reduzida e os dados em cache misturados com dados autorizados. O BIND 9 eliminou esse problema, assim você pode fazer a mistura. Mas a segurança e a higiene geral do DNS ainda exigem a separação das funções entre disponibilizar seus dados autorizados ao mundo e disponibilizar os dados mundiais aos seus usuários.

Servidores recursivos e não-recursivos

Os servidores de nomes são recursivos ou não-recursivos. Se um servidor não-recursivo tiver a resposta a uma consulta armazenada em cache a partir de uma transação anterior ou for autorizado para o domínio ao qual a consulta pertence, ele fornece uma resposta apropriada. Caso contrário, em vez de retornar uma resposta verdadeira, ele retornará uma referral aos servidores autorizados de outro domínio que têm maior probabilidade de conhecer a resposta. Um cliente de um servidor não-recursivo tem de ser preparado para aceitar e atuar baseado em referrals.

Embora os servidores não-recursivos possam parecer preguiçosos, eles normalmente têm uma boa razão para não fazer trabalho extra. Os servidores-raiz e de domínio de primeiro nível são todos não-recursivos, porém a 10.000 consultas por segundo podemos desculpá-los por usar a lei do menor esforço.

Um servidor recursivo só retorna respostas reais e mensagens de erros. Ele próprio segue referrals, liberando o cliente dessa responsabilidade. Em outro sentido, o procedimento básico para resolver uma consulta é essencialmente o mesmo. Por razões de segurança, os servidores de nomes externamente acessíveis de uma empresa sempre devem ser não-recursivos.

As bibliotecas de conversores (*resolvers*) *não* entendem referências; qualquer servidor de nomes local para o qual os clientes apontam deve ser recursivo.

Um efeito colateral de fazer com que um servidor de nomes siga referrals é que o seu cache adquire informações sobre domínios intermediários. Em uma rede local, esse cache normalmente é o comportamento desejado, uma vez que ele permite que pesquisas subseqüentes em qualquer host da rede se beneficiem do trabalho anterior do servidor de nomes. Por outro lado, o servidor para um domínio de primeiro nível, como com ou edu, não deve salvar informações solicitadas por um host vários domínios abaixo dele.

As primeiras versões de BIND exigiam mudanças no código-fonte e uma recompilação para modificar a recursividade de um servidor. Essa opção então foi movida para um flag (**-r**) de linha de comando e agora é um parâmetro no arquivo de configuração. Um servidor pode até mesmo ser configurado como recursivo para seus próprios clientes e não-recursivo para os de fora.

Os servidores de nomes geram referrals de maneira hierárquica. Por exemplo, se um servidor não for capaz de fornecer um endereço para o host lair.cs.colorado.edu, ele se refere aos servidores de cs.colorado.edu, colorado.edu, edu ou ao domínio-raiz. Uma referral deve incluir endereços para os servidores dos domínios referidos, de modo que a escolha não é arbitrária; o servidor deve se referir a um domínio para o qual ele já conhece os servidores.

Normalmente é retornado o domínio mais longo conhecido. Se o endereço de lair não fosse conhecido, porém os servidores de nomes para cs.colorado.edu fossem conhecidos, então seriam retornados os endereços para esses servidores. Se cs.colorado.edu não fosse conhecido, mas colorado.edu fosse, então os endereços dos servidores de nomes para colorado.edu seriam retornados e assim por diante.

Os servidores de nomes carregam previamente seus caches a partir de um arquivo de hints que lista os servidores para o domínio-raiz. É sempre possível fazer alguma referral, mesmo que simplesmente "vá perguntar ao servidor-raiz".

A biblioteca de conversão

Os clientes procuram mapeamentos de nomes de hosts chamando a família de rotinas de bibliotecas **gethostbyname**. A implementação original de **gethostbyname** pesquisava nomes no arquivo **/etc/hosts**. Para que mapeamentos de hosts sejam fornecidos pelo DNS, essas rotinas devem utilizar a biblioteca de conversão, que sabe como localizar e se comunicar com os servidores de nomes. O conversor normalmente é integrado nas bibliotecas padrão contra as quais os aplicativos são compilados.

As implementações de **gethostbyname** feitas pelos sistemas Linux podem extrair informações de várias fontes diferentes: arquivos simples (como **/etc/hosts**), DNS e talvez um sistema de banco de dados administrativo local como o NIS. Um arquivo de chaves possibilita um controle administrativo detalhado sobre quais fontes são pesquisadas e em que ordem. Consulte a página 333 ou a seção *Priorizando fontes de informações administrativas,* na página 358, para especificidades. As seções específicas a fornecedores e à distribuição da nossa abordagem do DNS apresentam tratamentos detalhados desse tópico, uma vez que ele pertence a pesquisas de hosts; elas começam na página 333.

Os aplicativos que utilizam a rede em geral estão vinculados a uma biblioteca de conversão stub que envia consultas DNS a um servidor de nomes de cache local. O suporte IPv6 torna as coisas mais complicadas, mas o BIND 9 fornece uma "biblioteca de conversão leve" e um daemon conversor, **lwresd**, para sites que não precisam falar o IPv6. O termo "peso leve" pode parecer um oxímoro no contexto do DNS, mas, nessa instância, diz respeito ao protocolo utilizado entre a biblioteca de conversores e o daemon conversor. Se a instrução `lwres` for incluída no arquivo de configuração de **named**, o servidor de nomes também atuará como um conversor leve. Atualmente o daemon conversor não utiliza o arquivo de comutação de serviço de nomes mencionado anteriormente: ele só examina no DNS a conversão de nomes.

Interfaces de shell para o DNS

A distribuição de software BIND inclui os comandos **dig**, **host** e **nslookup**, que fornecem interfaces de linha de comando para execução de consultas DNS. Elas são úteis como ferramentas de depuração e como ferramentas para extração de informações do DNS. Embora os comandos sejam similares em função, eles são um pouco diferentes em termos de design. Consulte a página 330 para informações adicionais.

15.9 PROJETANDO SEU AMBIENTE DE DNS

Muitos fatores afetam o projeto de um sistema DNS robusto e eficiente para seu ambiente particular: o tamanho da sua empresa, se você utiliza endereços IP privados da RFC1918 na sua rede local, se utiliza o DHCP, se utiliza o diretório ativo da Microsoft, se sua rede interna é roteada ou comutada e qual é a posição do seu firewall em relação aos seus servidores DNS, para citar alguns. Você pode achar útil dividir o problema em três partes:

- gerenciar a hierarquia do espaço de nomes: subdomínios, múltiplos níveis etc.;
- servir os dados autorizados sobre seu site para o mundo externo; e
- fornecer pesquisas de nomes para seus usuários.

Gerenciamento do espaço de nomes

Se o site for pequeno e independente, o uso de subdomínios não é necessário nem desejável a menos que seu gerenciamento necessite deles por alguma razão não técnica. Por outro lado, em uma empresa de médio porte com vários grupos independentes de administradores de sistemas, os subdomínios podem reduzir a necessidade de coordenação no nível global do site. (Subdomínios divididos ao longo de linhas geográficas ou departamentais são mais comuns.) Uma grande empresa tem poucas esperanças de impor nomes únicos por todo o seu site e, portanto, precisa de subdomínios, talvez em vários níveis.

A criação de subdomínios requer comunicação e cooperação entre os administradores de sistemas responsáveis pelo domínio pai e aqueles responsáveis pelo subdomínio. No momento em que o subdomínio é delegado e configurado, não deixe de escrever uma nota sobre quem contatar se você quiser adicionar, alterar ou excluir servidores. Certifique-se de que seu firewall não bloqueia acesso aos servidores do subdomínio se você quiser que o subdomínio seja acessível de fora da empresa.

Se utiliza subdomínios para gerenciar seu espaço de nomes, execute a ferramenta **doc** (*domain obscenity control*) a partir do **cron** uma vez por semana para certificar-se de que as delegações estão sincronizadas e de que você não cria inadvertidamente delegações defeituosas. A seção sobre ferramentas DNS (página 324) descreve **doc** e várias outras ferramentas que o ajudam a manter o DNS saudável.

Servidores autorizados

As especificações do DNS requerem pelo menos dois servidores autorizados para cada domínio. Servidores mestres e escravos são autorizados; servidores stub e de cache não são. Idealmente, um site tem múltiplos servidores autorizados, cada um em uma rede separada e em um circuito de energia diferente. Muitos sites mantêm um servidor autorizado fora da instalação, freqüentemente hospedados pelos seus ISPs. Se seu ISP não oferecer esse serviço, você pode comprá-lo de um provedor de serviço de DNS ou negociar com uma firma local (idealmente, não uma concorrente) ou uma universidade.

Recentemente, a Microsoft foi pega violando a regra das redes separadas. Ela tinha os seus três servidores autorizados na mesma sub-rede e, quando o roteador que conectava essa sub-rede à Internet falhou, os servidores tornaram-se inacessíveis. Duas horas depois, à medida que os registros armazenados em cache expiraram, o microsoft.com e todos os seus outros domínios ficaram fora do ar. O número de consultas para nomes relacionados à Microsoft nos servidores-raiz aumentou para 25% da carga total (10.000 consultas/segundo), até seu valor típico de 0,000001%. Os problemas persistiram por alguns dias. Quando a poeira assentou, a Microsoft havia corrigido o roteador e terceirizado seu serviço DNS!

Servidores autorizados mantêm seus dados sincronizados utilizando transferências de zona. Utilize chaves TSIG para autenticar e controlar as transferências de zona do seu servidor mestre aos seus servidores escravos. Consulte a página 453 para informações sobre a configuração do TSIG.

Talvez você queira que as respostas das consultas fornecidas pelos seus servidores autorizados dependam até certo ponto de quem está perguntando. Uma consulta de fora da sua rede poderia receber uma resposta, enquanto a mesma consulta que se origina de dentro da sua empresa receberia uma resposta diferente (mais completa). Essa configuração é chamada "DNS dividido" e é implementada no nível da zona, não no nível do servidor.

Cada versão da zona é chamada "visualização", depois da instrução `view` com que é configurado no arquivo de configuração do BIND. Funcionários externos vêem uma visualização dos dados e funcionários internos vêem outra. Esse recurso é comumente utilizado para ocultar a existência das máquinas internas de olhos bisbilhoteiros e assegurar que as máquinas utilizando endereços IP privados da RFC1918 não vazem esses endereços para a Internet. As visualizações são difíceis de depurar, mas as extensas capacidades de registro em log do BIND, juntamente com o uso inteligente do comando **dig**, podem ajudar; consulte a página 324 para algumas dicas.

Servidores de cache

Servidores recursivos de cache respondem às consultas dos usuários locais sobre sites na Internet. Cada computador na sua instalação deve ter acesso pronto a um servidor local de cache, preferivelmente um que esteja na mesma sub-rede.

Empresas de determinado porte devem considerar o uso de uma hierarquia em que uma ou mais máquinas são designadas como "encaminhadoras" por meio das quais os servidores de cache das sub-redes locais passam suas consultas. As encaminhadoras desenvolvem assim um cache que é comum para múltiplas sub-redes.

Dependendo do tamanho do seu site, encaminhadoras podem ser independentes ou organizadas em uma hierarquia. Por exemplo, os servidores de sub-rede poderiam encaminhar para um servidor departamental que, por sua vez, encaminha para um gateway no nível global do site. Os encaminhadores no nível global do site armazenam todos os nomes solicitados

pelos usuários por todo o site. Essa configuração minimiza a largura de banda externa usada para serviço de nomes e permite que todas as máquinas locais compartilhem um único e grande cache. Os encaminhadores são vistos no item sobre configuração que começa na página 297.

Se um servidor de cache morrer, essencialmente a rede pára de funcionar para todos os usuários que eram clientes desse servidor. (E seu telefone começa a tocar.) Inicie seus servidores de nome de cache com um script que os reinicia depois de alguns segundos se eles morrerem. Eis um exemplo de um script keep-running (mantenha-se ativo) a partir de uma máquina que executa duas instâncias separadas de **named**. Esse script recebe um único argumento para identificar o arquivo de log e o arquivo de configuração do servidor de nome a utilizar:

```
#!/bin/sh

PATH=/usr/local/sbin:/usr/sbin:/sbin:$PATH
export PATH

trap "    " 1
while :: do
   named -f -c /var/named/named.$1.conf >> /var/log/named.$1 2>&1 < /dev/null
   logger "named ($1) restart"
   sleep 15
done
exit
```

O script espera 15 segundos (um valor arbitrário) antes de reiniciar **named** e executa o comando **logger** para submeter uma entrada de syslog sempre que ele tem de reiniciar o servidor.

Segurança

A segurança DNS é discutida em uma seção própria, que inicia na página 314. Não iremos reproduzir essa discussão aqui, exceto para lembrá-lo de que caso você utilize um firewall, se certifique de que seu sistema DNS não emita consultas para as quais seu firewall bloqueia as respostas. Isso essencialmente significa que seus administradores de DNS devem ter comunicação constante com seus administradores de segurança e de rede. Por padrão, o DNS usa UDP com portas de origem não-privilegiadas aleatórias (>1023) para consultas; as respostas são pacotes UDP endereçados a essas mesmas portas.

Resumindo

A Ilustração B demonstra a estrutura recomendada nos parágrafos anteriores. Ele mostra uma hierarquia de encaminhamento em dois níveis, que é demasiado para a maioria dos sites. Ajuste o balanço entre os servidores que manipulam consultas que vão para fora e os servidores que atendem a consultas que chegam de fora para que nenhum dos grupos seja muito sobrecarregado.

Note também o uso do servidor escravo fora da instalação, que é altamente recomendado. Empresas com vários escritórios podem designar uma localidade como mestre; essa localidade se torna "fora da instalação" em relação ao restante da empresa.

Ilustração B Arquitetura de servidor DNS

Uma taxonomia das tarefas DNS/BIND

Consulte o Capítulo 17 para informações adicionais sobre como distribuir arquivos por uma rede.

A Tabela 15.9 resume o que deve ser feito, para quem e a freqüência com que você utiliza o BIND e o DNS. Uma entrada na coluna "Com que freqüência" que inclua a palavra "distribuir" significa que você o faz uma vez só por sub-rede ou arquitetura e depois copia o resultado para os hosts apropriados com uma ferramenta do tipo **rdist** ou **rsync**. É aplicado um sombreamento em linhas alternadas para facilitar a leitura.

15.10 QUESTÕES RELACIONADAS A CLIENTES NO BIND

Como cada host da rede tem de ser um cliente BIND, comecemos nossa discussão detalhada com tarefas do lado cliente.

Configuração do conversor

Cada host Linux na rede tem um arquivo chamado **/etc/resolv.conf** que lista os servidores DNS que o host deve consultar. Se seu host obtiver o seu endereço IP e parâmetros de rede a partir de um servidor DHCP, o arquivo **/etc/resolv.conf** deve ser configurado para você de forma automática. Do contrário, você precisa editar o arquivo manualmente. O formato é

```
search nomeDeDominio ...
nameserver ipaddr
```

Tabela 15.9 Tarefas de instalação e de manutenção no BIND.

Tarefa	Para quem	Com que freqüência
Obter o nome do domínio	Site	Somente uma vez
Escolher servidores de nomes	Site	Uma vez ou mais
Obter distribuição BIND	Site	Somente uma vez, mas mantê-la atualizada
Configurar conversor	Cliente	Somente uma vez e distribuir
Configurar conversor eficiente	Cliente	A cada sub-rede e distribuir
Configurar comutação de serviços	Cliente	A cada arquitetura e distribuir
Iniciar **named** em tempo de inicialização	Servidor	A cada servidor de nomes
Configurar o arquivo de configuração **named**	Servidor	A cada tipo de servidor
Configurar o arquivo de hints	Servidor	Somente uma vez[a] e distribuir aos servidores
Configurar arquivos de zonas	Mestre	Somente uma vez
Atualizar arquivos de zonas	Mestre	Conforme necessário
Revisar arquivos de log	Host de log	Pelo menos uma vez por semana
Instruir os usuários	Todos os hosts	Contínua e repetidamente

a. Mas tem de ser refeita se os servidores-raiz mudarem.

Podem ser listados até três servidores de nomes. Eis um exemplo completo:

```
search cs.colorado.edu colorado.edu ee.colorado.edu
nameserver 128.138.243.151    ; ns
nameserver 128.138.204.4      ; piper
nameserver 128.138.240.1      ; anchor
```

Os comentários nunca foram definidos para o arquivo **resolv.conf**. Eles são mais ou menos suportados pelo fato de que qualquer coisa que não é reconhecida é ignorada. É seguro colocar comentários no final das linhas do `nameserver`, pois o analisador sintático só procura um endereço IP e ignora o restante da linha. Pelo fato de a linha `search` poder conter vários argumentos, os comentários ali poderiam provocar problemas.

A linha `search` lista os domínios a serem consultados caso um nome de host não seja totalmente qualificado. Se um usuário emitir o comando **ssh foo**, por exemplo, o conversor completa o nome com o primeiro domínio na lista de pesquisa (no **resolv.conf** anterior, cs.colorado.edu) e procura foo.cs.colorado.edu. Se nenhum desses nomes puder ser localizado, o conversor também tenta foo.colorado.edu e foo.ee.colorado.edu. O número de domínios que pode ser especificado em uma diretiva `search` é específico ao conversor; a maioria permite entre seis e oito.

Os servidores listados em **resolv.conf** devem ser recursivos uma vez que o resolver não entende referências. Os servidores nas linhas `nameserver` são contatados em ordem; enquanto o primeiro deles continuar a responder consultas, os demais

serão ignorados. Se ocorrer um problema, a consulta é expirada e o próximo servidor de nomes será tentado. Cada servidor é tentado alternadamente, até quatro vezes. O intervalo de tempo-limite aumenta a cada falha.

A maioria dos conversores permite que sejam listados no máximo três servidores de nomes. Se for fornecido um número maior, eles serão silenciosamente ignorados. Se o próprio host for um servidor de nomes, ele deve ser listado primeiramente em seu próprio arquivo **resolv.conf**. Se nenhum servidor de nome estiver listado, o host local será utilizado.

As primeiras versões do BIND usavam a diretiva `domain` no arquivo **resolv.conf** em vez da diretiva `search`. Ela especificava um único domínio para adicionar nomes que não eram totalmente qualificados. Recomendamos substituir as diretivas `domain` por diretivas `search`. As duas diretivas são mutuamente exclusivas, de modo que apenas uma deve estar presente. Se você tiver um conversor mais antigo e incluir ambas as diretivas no arquivo **resolv.conf**, será usada a última delas.

Testes de conversores

A maioria das distribuições Linux tentará pesquisar hostnames no DNS por padrão contanto que haja uma linha `nameserver` em **/etc/resolv.conf**. O ordenamento das várias fontes de informações sobre o hostname, incluindo o DNS, é configurado no arquivo **/etc/nsswitch.conf**; consulte a página 333.

Após configurar **/etc/resolv.conf** (e supondo que sua conexão de rede local esteja ativa e funcionando corretamente), você deve ser capaz de se referir a outras máquinas pelo nome em vez de seu endereço IP. Se você tentar alcançar uma outra máquina local e o comando simplesmente travar, tente se referir à máquina pelo seu endereço IP. Se isso funcionar, então sua configuração DNS é o problema. Verifique se os endereços IP do servidor de nomes em **/etc/resolv.conf** estão corretos e se os servidores para os quais você aponta permitem consultas a partir da sua rede (consulte a página 297). **dig** a partir de uma máquina em funcionamento pode responder a essas perguntas.

Impacto sobre o restante do sistema

A mudança de tabelas de hosts estáticas para DNS cria algumas dependências potenciais na inicialização e configuração contra as quais você deve se proteger.

Quando um host faz sua inicialização, as referências a nomes de host presentes nos arquivos de inicialização talvez não sejam conversíveis se forem encontrados antes de a rede estar ativa. Os comandos nos arquivos de inicialização tentarão, em vão, contatar o DNS. Graças à robustez do conversor, eles tentarão várias vezes em vários servidores distintos, aumentando o respectivo período de tempo-limite a cada tentativa frustrada. Alguns minutos depois, o comando que precisava do nome de host finalmente falhará.

Para solucionar o problema, use somente endereços IP literais nos estágios iniciais do processo de inicialização. Ou então, como o Linux suporta o emprego simultâneo de DNS e de **/etc/hosts** por meio do arquivo de switch, você poderá instalar um arquivo **hosts** contendo os endereços de servidor necessários em tempo de inicialização. Certifique-se de que o arquivo **hosts** é verificado antes do DNS para não ter de esperar o DNS atingir seu período de tempo-limite.

15.11 Configuração de servidores BIND

Nesta seção, partimos da suposição de que suas tarefas políticas foram completadas. Isto é, supomos que você tem um nome de domínio (possivelmente um subdomínio), que coordenou com o administrador de DNS do domínio pai e que delegou seu espaço de endereçamento na árvore reversa in-addr.arpa. Escolheu seu servidor mestre de nomes e alguns secundários e instalou o BIND.

Exigências de hardware

O BIND é um consumidor voraz de memória. Seu banco de dados é mantido em memória, de modo que à medida que o cache cresce, o mesmo acontece com o processo **named**. Alguns dos novos recursos do BIND 9 também consomem muitos recursos de CPU, mais notadamente DNSSEC. Para ajudar a reduzir toda essa carga, o BIND 9 é multithread e capaz de fazer pleno uso dos sistemas multiprocessados. O BIND 9 também contém opções de configuração que controlam o emprego de recursos por parte de **named**.

A melhor maneira de determinar se um servidor de nomes possui memória suficiente é executá-lo por um instante e observar o tamanho do processo **named**. Leva de uma a duas semanas até se chegar a um tamanho estável no qual registros de cache antigos devem expirar praticamente na mesma taxa que os novos estão sendo inseridos. Uma vez estabilizado, o sistema não deve fazer swapping e suas taxas de paginação devem ser razoáveis.

Se o seu servidor de nomes é executado numa máquina dedicada, uma ótima regra é a máquina ter o dobro de memória consumido por **named** depois de ele ter sido executado por uma semana. Os comandos **top** e **vmstat** mostram o emprego de memória; consulte a seção *Analisando o emprego de memória*, na página 557 para mais detalhes.

Arquivos de configuração

A configuração completa para **named** consiste no arquivo de configuração, no arquivo hints do servidor de nomes-raiz, e, para servidores mestres, nos arquivos de dados de zona que contêm mapeamentos de endereço para cada host. O arquivo de configuração do **named** tem um formato próprio; todos os outros arquivos são coleções de registros de dados DNS individuais formatadas de acordo com a especificação DNS. Discutimos o arquivo de configuração nas próximas duas seções. O formato de registros de dados DNS é descrito no começo da página 271.

O arquivo de configuração do **named**, **named.conf**, especifica os papéis (mestre, escravo, stub ou somente de cache) desse host e a maneira como ele deve obter sua cópia dos dados para cada zona que ele serve. Ele também é o lugar em que as opções são especificadas — tanto as opções globais relacionadas à operação geral do **named** como as opções específicas ao servidor ou à zona que só se aplicam a uma parte do tráfego DNS.

O arquivo de configuração é formado por uma série de instruções, cada uma das quais terminada por um ponto-e-vírgula. Os tokens são separados por espaço em branco, que podem incluir caracteres de uma nova linha. As chaves são algumas vezes usadas para agrupamento, mas somente em contextos específicos. O formato é bem frágil e a ausência de um ponto-e-vírgula pode provocar uma devastação.

Felizmente, o BIND 9 inclui algumas ferramentas úteis para verificar a sintaxe do arquivo de configuração (**named-checkconf**) e os arquivos de dados de zona (**named-checkzone**). Eles saem em busca de erros e omissões. Por exemplo, **named-checkzone** informará se você se esqueceu de incluir uma diretiva `$TTL`. Infelizmente ele não captura tudo. Por exemplo, registros de adesão ausentes (consulte a página 283) não são informados e provocam pesadas cargas nos servidores-raiz e gTLD.

Comentários podem aparecer em qualquer ponto em que o espaço em branco for indicado. São válidos e compreendidos os comentários em C, C++ e tipo shell:

```
/* This is a comment and can span lines. */
// Everything to the end of the line is a comment.
# Everything to the end of the line is a comment.
```

Cada instrução começa com uma palavra-chave que identifica o tipo de instrução. Pode haver mais de uma instância de cada tipo de instrução, exceto para `options` e `logging`. Instruções e partes de instruções também podem ser omitidas, invocando o comportamento padrão para os itens ausentes. A Tabela 15.10 lista as instruções disponíveis; a coluna Página indica nossa discussão sobre cada instrução nas próximas seções. Linhas alternadas foram sombreadas para aumentar a legibilidade.

Tabela 15.10 Instruções utilizadas em named.conf

Instrução	Página	Função
`include`	423	Interpola um arquivo (por exemplo, chaves confiáveis legíveis somente por **named**)
`options`	423	Configura opções e padrões de configuração de servidor de nomes globais
`acl`	429	Define listas de controle de acesso
`key`	430	Define informações de autenticação
`trusted-keys`	430	Usa chaves pré-configuradas
`server`	431	Especifica opções por servidor
`masters`	432	Define uma lista dos mestres para inclusão nas zonas stub e nas zonas escravas
`logging`	432	Especifica categorias de registro em log e seus destinos
`zone`	432	Define uma zona de registros de recursos
`controls`	436	Define canais utilizados para controlar o servidor de nomes com rndc
`view`	438	Define uma visualização dos dados de zona
`lwres`	–	Especifica se o servidor de nomes também deve ser um conversor

Antes de descrevermos essas instruções e a maneira pelas quais elas são usadas para configurar **named**, precisamos descrever uma estrutura de dados que é utilizada em muitas das instruções: a lista de correspondência de endereços. Uma lista desse tipo é uma generalização de um endereço IP que pode incluir os seguintes itens.

- Um endereço IP, v4 ou v6 (por exemplo, 199.165.145.4)

- Uma rede IP especificada com uma máscara de rede CIDR[13] (por exemplo, 199.165/16)
- O nome de uma lista de controle de acesso definida previamente (consulte a página 298)
- O nome de uma chave criptográfica de autenticação
- O caractere ! para negar coisas

As listas de correspondência de endereços são utilizadas como parâmetros para várias instruções e opções. Alguns exemplos:

```
{ ! 1.2.3.13; 1.2.3/24; };
{ 128.138/16; 198.11.16/24; 204.228.69/24; 127.0.0.1; };
```

A primeira dessas listas exclui o host 1.2.3.13, mas inclui o restante da rede 1.2.3/24; uma segunda define as redes atribuídas à Universidade de Colorado. As chaves e o ponto-e-vírgula final não são realmente parte das listas de correspondência de endereços, mas foram incluídas para ilustração; elas seriam parte das instruções de inclusão das quais as listas de correspondência de endereços são uma parte.

Quando uma rede ou endereço IP é comparado a uma lista de concordância, a lista é pesquisada em ordem até que seja encontrada uma concordância. Esse algoritmo de "primeira concordância" torna importante o ordenamento das entradas. Por exemplo, a primeira lista de correspondência de endereços acima não teria o efeito desejado se as duas entradas fossem invertidas, pois 1.2.3.13 seria bem-sucedido na concordância com 1.2.3/24, mas a entrada invertida jamais seria encontrada.

Agora, vamos às instruções! Algumas são curtas e doces; outras quase precisam de um capítulo inteiro só para elas.

A instrução `include`

Para subdividir ou organizar melhor uma configuração grande, você pode colocar diferentes trechos da configuração em arquivos separados. Arquivos subsidiários são trazidos para o arquivo **named.conf** pela instrução `include`:

```
include "caminho";
```

Se *caminho* for relativo, então ele será interpretado em relação ao diretório especificado na opção `directory`. Um emprego comum da instrução `include` é o de trazer chaves criptográficas que não deveriam ser legíveis por todos. Em vez de fechar o acesso de leitura ao arquivo **named.conf** inteiro, alguns sites mantêm chaves em arquivos com permissões restritas que somente **named** é capaz de ler. Esses arquivos são então incluídos em **named.conf**.

A instrução `options`

A instrução `options` especifica opções globais, algumas das quais podem ser posteriormente anuladas para determinadas zonas ou servidores. O formato genérico é:

```
options {
    opção;
    opção;
 …
};
```

Se não estiver presente nenhuma instrução `options` em **named.conf**, então serão utilizados valores padrão.

O BIND 8 tinha aproximadamente 50 opções e o BIND 9 tem mais ou menos 100. Para uma lista completa, consulte a documentação do BIND[14] ou o *DNS and BIND* da O'Reilly, de Paul Albitz e Cricket Liu (a quarta edição abrange tanto o BIND 8 como o 9). Reduzimos a cobertura dessas opções e discutimos somente aquelas cujo uso recomendamos. (Também solicitamos sugestões por parte dos desenvolvedores do BIND sobre quais opções deveríamos focar e aceitamos os seus conselhos.) Os valores padrão são enumerados entre colchetes ao lado de cada opção. Para a maioria dos sites, os padrões são OK. As opções não são listadas em ordem particular.

```
directory "caminho";      [diretório em que o servidor foi iniciado]
```

A instrução `directory` faz com que **named** mude para o diretório (**cd**) especificado. Sempre que nomes de caminhos relativos aparecerem nos arquivos de configuração de **named**, eles serão interpretados de forma relativa a esse diretório. O *caminho* deve ser um caminho absoluto. Quaisquer arquivos de saída (depuração, informações estatísticas etc.) também são gravados nesse diretório.

---

13. As máscaras de rede CIDR são descritas a partir da página 201.
14. O arquivo **doc/misc/options** na distribuição contém a sintaxe para a linguagem de configuração e inclui todas as opções.

Recomendamos colocar todos os arquivos de configuração relacionados com BIND (além de **named.conf** e **resolv.conf**) em um subdiretório abaixo de **/var** (ou onde quer que você mantenha seus arquivos de configuração para outros programas). Utilizamos **/var/named**.

```
version "string";          [número real da versão do servidor]
hostname "string";         [nome de host real do servidor]
server-id "string";        [nenhum]
```

A string de versão identifica a versão do software do servidor de nomes em execução no servidor; a string de nome de host identifica o próprio servidor assim como ocorre com a string de ID de servidor. Essas opções permitem mentir sobre os valores verdadeiros. Cada uma coloca os dados nos registros TXT da classe CHAOS em que os espectadores curiosos irão procurá-los.

Há duas linhas de pensamento sobre a questão de ocultar o número de versão do software do servidor de nomes. Alguns sites acreditam que seus servidores serão mais vulneráveis a ataques se os hackers conseguirem saber que versão está sendo executada neles. Outros acreditam que ocultar as informações sobre versão é contraproducente, pois os hackers tentarão a sorte de qualquer maneira, e a maioria dos bugs recentemente descoberta está presente em todas as versões do software.

Recomendamos que você não redefina a string de versão. É muito prático ser capaz de consultar seus servidores de nomes e descobrir que versão eles estão executando (por exemplo, caso você queira saber se o seu fornecedor está entregando uma versão atual ou caso você precise verificar se de fato atualizou todos os seus servidores para a última revisão).

Os parâmetros de nome de host e de ID de servidor são adições recentes motivadas pelo uso do roteamento anycast para duplicar instâncias dos servidores-raiz e dos servidores gTLD. Por exemplo, há 20 instâncias do f.root-servers.net no mundo todo e, como um usuário ou administrador de sistemas tentando depurar as coisas, você não tem nenhuma idéia de qual desses 20 servidores responderam (ou não conseguiram responder) a sua consulta. Consultar hostname.bind na classe CHAOS informaria qual instância do servidor-raiz F estava respondendo (contanto que o roteamento não mudasse nesse ínterim). "Servidor-id" é um nome politicamente correto para hostname.bind, uma vez que nem todos os servidores de nomes executam o BIND.

```
notify yes | no;                                    [yes]
also-notify servidores_endereçosIP;                 [vazio]
```

Se `notify` for configurado como `yes` e esse **named** for o servidor mestre para uma ou mais zonas, **named** notificará automaticamente os servidores escravos dessas zonas sempre que o banco de dados de zonas correspondente mudar. Os servidores escravos podem então se encontrar com hora marcada com o mestre para atualizar suas cópias dos dados de zona. Você pode utilizar a opção `notify` tanto como uma opção global como uma opção específica à zona. Ela faz com que os arquivos de zonas convirjam muito mais rapidamente depois que você fizer alterações.

Consulte a página 301 para informações adicionais sobre zonas de stub.

named normalmente descobre quais máquinas são servidores escravos de uma zona analisando os registros NS desta zona. Se `also-notify` for especificada, um conjunto de servidores adicionais que não são anunciados com registros NS também pode ser notificado. Esse ajuste fino algumas vezes é necessário quando seu site tiver servidores internos. Não use a opção `also-notify` em servidores stub; eles estão interessados somente nos registros NS da zona e podem aguardar o ciclo de atualização regular. O alvo de uma `also-notify` deve ser uma lista de endereços IP. A zona reversa de host local é um bom lugar para desativar a notificação.

```
recursion yes | no;                                 [yes]
allow-recursion { lista_de_correspondência_de_endereços };    [todos os hosts]
```

A opção `recursion` especifica se **named** consulta outros servidores de nomes em nome dos clientes, conforme descrito na página 287. É relativamente incomum configurar um servidor de nomes com a recursão desativada. Entretanto, pode ser que você queira permitir recursão para os seus próprios clientes, mas não para consultas externas.

A recursão pode ser controlada com uma precisão mais fina pela opção `allow-recursion` e uma lista de endereços que inclua os hosts e redes em cujo nome você pretende fazer consultas recursivas.

```
transfer-format one-answer | many-answers;      [many-answers]
```

Essa opção afeta a maneira como os registros de dados DNS são replicados dos servidores mestres aos seus servidores escravos. A transmissão em si de registros de dados costumava acontecer um registro de cada vez, que é uma receita para lentidão e ineficiência. Uma opção para combinar muitos registros em um pacote (`many-answers`) foi adicionada ao BIND 8.1; ela é o padrão no BIND 9. Se tiver um ambiente misto, você poderá especificar um formato de transferência nas instruções `server` individuais para sobrescrever a opção global. Sua mescla de servidores determinará se você deve optar por `many-answers` globalmente e sobrescrevê-la para servidores específicos ou vice-versa.

```
transfers-in número;            [10]
transfers-out número;           [10 (V9 somente)]
```

```
transfers-per-ns número;          [2]
transfer-source Endereço-IP;      [dependente de sistema]
```

Um site grande — um que atenda a uma zona muito ampla (por exemplo, com, que hoje está além dos dois gigabytes) ou um que atenda a milhares de zonas — pode precisar de ajuste em algumas dessas opções de transferência de zonas.

As opções `transfers-in` e `transfers-out` limitam o número de transferências de zonas de dentro para fora ou de fora para dentro que podem acontecer concorrentemente.[15] A opção `transfers-per-ns` configura o número máximo de transferências de zonas de dentro para fora que pode estar em execução concorrente no mesmo servidor remoto. Grandes sites talvez precisem aumentar `transfers-in` ou `transfers-out`; tenha cuidado para não ficar sem descritores de arquivo para o processo **named**. Você deve aumentar `transfers-per-ns` se todos os servidores mestres pretenderem manipular mais de duas transferências de zonas simultâneas. Modificá-lo em uma base servidor por servidor utilizando a estrutura `transfers` da instrução `server` é uma forma melhor de ajustar a convergência de zonas escravas.

A opção `transfer-source` permite especificar o endereço IP da interface que você deseja utilizar para transferências que chegam. Ela deve atender ao endereço especificado na instrução `allow-transfer` do mestre.

Como acontece com qualquer parâmetro que modificamos drasticamente, você deve observar as coisas com atenção após modificar um desses valores reguladores para garantir que a máquina não apresenta problemas. Os arquivos de log são nossos aliados.

```
files número;                     [ilimitado]
datasize número;                  [valor padrão do SO, dependente de sistema]
```

A opção `files` configura o número máximo de arquivos que o servidor poderá ter aberto ao mesmo tempo. O valor padrão significa o mais próximo possível do número de arquivos abertos que o kernel é capaz de suportar. (Para modificar o número de arquivos aberto que o kernel pode suportar, configure o valor de **/proc/sys/fs/file-max**. Consulte a página 600.)

A opção `datasize` configura a quantidade máxima de memória para os dados que o servidor tem permissão de usar. As unidades são bytes, mas **named** entende G para gigabytes, M para megabytes etc. Essa opção não deve ser utilizada para tentar controlar o tamanho do cache; ela é um limite rígido e, dependendo de onde é encontrada no código, pode fazer com que o servidor de nomes trave ou pare. Ela é normalmente utilizada para aumentar o limite máximo padrão para processos individuais na memória do sistema operacional. Para controlar o tamanho do cache, utilize as opções `max-cache-size` e `recursive-clients`.

```
max-cache-size número;            [ilimitado]
recursive-clients número;         [1000]
```

A opção `max-cache-size` limita a quantidade de memória que o servidor pode utilizar para armazenamento em cache de respostas a consultas. Se o tamanho do cache aumentar muito, os registros serão removidos do cache antes dos seus TTLs expirarem para assegurar que o uso de cache de memória permaneça dentro do limite especificado.

A opção `recursive-clients` limita o número de consultas recursivas simultâneas que o servidor processará. Cada consulta recursiva consome uma parte da memória no servidor; servidores que limitam o uso de memória talvez precisem configurar esse limite com um valor mais baixo do que o valor padrão.

```
query-source address endereçoIP port porta_ip;                    [aleatório]
listen-on port lista_de_correspondência_de_endereços_de_portas_ip;  [53 any]
```

A opção `query-source` especifica a interface e a porta que **named** utiliza para consultar outros servidores de nomes. A opção `listen-on` especifica as portas e interfaces de rede nas quais o processo **named** fica na escuta esperando consultas. Os valores dessas opções têm como padrão o comportamento padrão de **named**: ouvir na porta 53 em todas as interfaces e utilizar uma porta UDP aleatória de numeração elevada e qualquer interface para consultas.

Se seu site tiver um firewall, você poderá usar a opção `query-source` para atribuir a consultas DNS externas um perfil específico e reconhecível. Algum sites configuram **named** para que ele utilize a porta 53 como a porta de origem (bem como a porta em que ouvir) de modo que o firewall possa reconhecer o tráfego DNS externo como sendo pacotes confiáveis provenientes de um dos seus servidores de nomes. Entretanto, essa convenção dificulta o filtro de pacotes em fazer a distinção entre o tráfego DNS interno e externo. Você pode usar uma outra porta em vez da 53 desde que o firewall saiba quais são essas portas.

Não configure a porta de origem como zero — trata-se de uma porta ilegal, e **named** registrará em log a consulta como um erro e não responderá a ela. Um grande ISP possui um administrador de sistemas que gosta da porta 0 e tornou muitos de seus servidores de nomes ineficientes pelo uso da cláusula `query-source`. É curioso que seus clientes não percebam e não reclamem.

---

15. O código BIND impõe um limite definido internamente de 20 para o parâmetro `transfers-in`.

```
avoid-v4-udp-ports { lista_de_portas };          [vazio]
avoid-v6-udp-ports { lista_de_portas };          [vazio]
```

Se você não utilizar a opção `query-source`, em geral o BIND 9 utilizará qualquer interface de saída e atribuirá aleatoriamente a porta de origem para uma consulta a partir do intervalo de portas não privilegiadas (>1023). Isso será um problema se o firewall do seu site bloquear certas portas nesse intervalo. Por exemplo, alguns firewalls consideram incorretamente a porta 1024 como parte do intervalo de portas privilegiadas ou bloqueiam corretamente a porta 2049 (SunRPC) do mundo externo. Quando seu servidor de nome envia uma consulta e utiliza uma das portas bloqueadas como sua origem, o firewall bloqueia a resposta e o servidor de nomes conseqüentemente desiste de esperar e envia a consulta novamente. Isso não é fatal, mas irritante para o usuário pego no fogo cruzado.

Para evitar esse problema, utilize a opção `avoid-v4-udp-ports` para fazer com que o BIND permaneça longe das portas bloqueadas. Quaisquer portas UDP com números altos bloqueadas pelo seu firewall devem ser incluídas na lista.[16] Se atualizar seu firewall em resposta a algum ataque ameaçador, não deixe também de atualizar a lista de portas aqui.

```
forwarders { no_endereço; no_endereço; … };      [lista vazia]
forward only | first;                            [first]
```

Em vez de fazer com que cada servidor de nomes execute suas próprias consultas externas, você pode designar um ou mais servidores como *encaminhadores*. Um servidor comum é capaz de pesquisar em seu cache e nos registros para os quais ele é autorizado; se ele não encontrar a resposta que está procurando, poderá então enviar a consulta para um host encaminhador. Dessa maneira, os encaminhadores constroem caches que beneficiam o site inteiro. A designação é implícita — não há nada no arquivo de configuração do encaminhador que diga: "Ei, você é um encaminhador".

A opção `forwarders` lista os endereços IP dos servidores que você quer usar como encaminhadores. Eles são consultados alternadamente. O emprego de um encaminhador impede que o procedimento DNS normal inicie em um servidor-raiz e siga a cadeia de referrals. Tome cuidado para não criar loops de encaminhamento.

Um servidor somente para encaminhamento (`foward only`) coloca em cache valores e encaminhadores de consultas, mas ele nunca consulta mais ninguém. Se os encaminhadores não responderem, as consultas falharão. Um servidor do tipo encaminhe primeiro (`forward first`) prefere lidar com encaminhadores, mas processará consultas diretamente, se for necessário.

Como a opção `forwarders` não tem nenhum valor padrão, o encaminhamento não ocorre a menos que ele tenha sido configurado especificamente. Você pode ativar o encaminhamento de forma global ou então dentro das instruções `zone` individuais.

```
allow-query { lista_de_correspondência_de_endereços };        [todos os hosts]
allow-transfer { lista_de_correspondência_de_endereços };     [todos os hosts]
blackhole { lista_de_correspondência_de_endereços };          [vazio]
```

Essas opções especificam quais hosts (ou redes) podem consultar seu servidor de nomes e requisitam transferências de blocos dos seus dados de zona. A lista de endereços `blackhole` identifica servidores com os quais você nunca desejará falar; **named** não aceitará consultas desses servidores e não irá requisitar respostas deles.

```
sortlist { lista_de_correspondência_de_endereços };           [elas devem morrer, não as utilize]
```

Mencionamos a opção `sortlist` somente para alertá-lo para que fique longe dela. Seu propósito era o de ajudar os conversores primitivos que não ordenavam conjuntos de registros de forma apropriada. Ela permite que você especifique a ordem na qual várias respostas são retornadas e funciona contra a inteligência interna dos BINDs atuais.

Outras opções do BIND que se metem na ordem das coisas são a instrução `rrset-order`, que especifica se devem ser retornadas várias respostas em uma ordem cíclica (round robin), fixa ou aleatória, e a instrução `topology`, que tenta adivinhar para que seleção de servidores remotos enviar a consulta. Na maioria dos casos não há necessidade de usar essas instruções.

```
lame-ttl número;          [600 (10 minutos)]
max-ncache-ttl número;    [10800 (3 horas)]
max-cache-ttl número;     [604800 (1 semana)]
```

As opções TTL configuram limites sobre os valores de tempo de vida que controlam o armazenamento em cache dos resultados de consultas positivas, negativas e de não-resposta. Todos os registros de recursos entrantes chegam com TTLs próprios, mas essas opções permitem que o servidor local configure seus próprios limites como um esforço para reduzir a memória consumida pelos dados armazenados em cache. As opções `max-` armazenam os valores TTL e nunca irão aumentá-los. As especificações DNS requerem que o TTL para respostas negativas ("no such domain", por exemplo) seja menor que o TTL para os dados reais.

---

16. Alguns firewalls têm informações de estado e podem ser suficientemente inteligentes para reconhecer a resposta DNS como associada a consulta correspondente de um segundo atrás. Esses firewalls não precisam da ajuda dessa opção.

`lame-ttl` especifica o número de segundos para armazenar em cache uma indicação de servidor defeituoso. O valor máximo é 1.800 segundos ou 30 minutos. A maioria dos servidores defeituosos (consulte a página 331) permanece defeituosa indefinidamente porque seus proprietários não sabem que eles não estão em sincronização com suas zonas pai. Configurar esse parâmetro como o valor máximo deve ser suficiente. `max-ncache-ttl` configura o limite TTL para cache negativo. Ele assume o padrão de três horas e tem um valor máximo de uma semana. `max-cache-ttl` configura o TTL máximo para uma resposta positiva a uma consulta.

```
enable-dnssec yes | no;   [no]
```

Essa opção configura o suporte a DNSSEC. O padrão atual é `no`, mas esperamos que em um ou dois anos possa ser modificado para `yes`.

```
edns-udp-size número;     [4096]
```

O tamanho padrão antigo de pacotes UDP é 512 bytes; todas as máquinas na Internet precisam ser capazes de remontar um pacote UDP fragmentado de 512 bytes ou menos. Embora esse limite fizesse sentido na década de 1980, ele não faz mais. Esse limite agora afeta seriamente programas como servidores de nomes que utilizam UDP para consultas e respostas.

O BIND 9 configura o tamanho padrão dos pacotes UDP como 4.096 bytes (o máximo permitido) a fim de acomodar novos recursos como DNSSEC, IPv6 e nomes de domínio internacionalizados. Entretanto, alguns firewalls (falhos) não permitirão pacotes UDP maiores que 512 bytes; se utilizar um desses firewalls, você precisará dessa opção como uma maneira de contornar o problema até corrigir o firewall. Valores válidos variam de 512 a 4.096 bytes.

```
ixfr-from-differences yes | no;  [no]
```

Transferências incrementais de zona (*incremental zone transfers* – IXFRs) eram inicialmente suportadas apenas por zonas que utilizavam atualizações dinâmicas (descritas a partir da página 312). Com essa opção, as zonas que são mantidas manualmente (isto é, com um editor de textos) agora também tiram proveito das transferências incrementais de zona. As IXFRs economizam largura de banda de rede à custa de um pouco de CPU e de memória para classificar os registros de zonas e calcular e aplicar os diffs. Essa opção utiliza o mesmo sistema de journaling robusto empregado pelas atualizações dinâmicas.

```
root-delegation-only exclude { lista_de_nomes}; [vazio]
zone "name" { type delegation-only; };
```

Consulte a página 269 para informações sobre o Site Finder.

A ferramenta Site Finder, da VeriSign, tenta redirecionar erros de digitação de todas as pessoas para seus próprios servidores utilizando um registro DNS curinga. Há tentativas de resolver essa forma de seqüestro por meios políticos, mas, nesse ínterim, o BIND 9 implementou um tipo de zona `delegation-only`, uma opção de zona `delegation-only` e a opção global `root-delegation-only` como medidas temporárias. Essas opções restringem o fornecimento de informações de delegação (registros de servidor de nomes e endereços dos servidores de nomes) pelos servidores-raiz e servidores TLD e os proíbe de fornecer dados sobre o host. Com elas ativadas, seus servidores locais de cache retornarão "no such domain" como a resposta para consultas digitadas erroneamente.

A cláusula `exclude` não foi adicionada porque nem todos os TLDs são zonas somente de delegação. Por exemplo, "museum", "us" e "de" não são, mas "com", "net" e "org" devem ser. O trecho a seguir a partir dos arquivos de zona do domínio cs.colorado.edu ilustra o uso desse novo tipo de zona para neutralizar o Site Finder.

```
// to deal with VeriSign Site Finder crap
zone "com" { type delegation-only; };
zone "net" { type delegation-only; };
```

A instrução `acl`

Uma lista de controle de acesso é simplesmente uma lista de correspondência de endereços com um nome:

```
acl nome_acl {
     lista_de_correspondência_de_endereços
};
```

Você pode utilizar uma lista de controle de acesso em qualquer lugar em que uma lista de correspondência de endereços é necessária.

Uma `acl` deve ser uma instrução de primeiro nível em **named.conf**, portanto, não tente colocá-la furtivamente no meio das demais declarações de opções. **named.conf** é lido de uma só vez, de modo que as listas de controle de acesso devem ser definidas antes de utilizadas. São predefinidas quatro listas:

- `any` — todos os hosts

- `localnets` — todos os hosts na rede local
- `localhost` — a máquina em si
- `none` — nada

A lista `localnets` inclui todas as redes às quais o host está conectado diretamente. Em outras palavras, ela é uma lista do módulo de endereços de rede da máquina em relação às suas máscaras de rede.

A instrução `key`

A instrução `key` define uma chave criptográfica identificada para ser usada para autenticação com determinado servidor. São dadas informações sobre o suporte BIND à autenticação criptográfica na seção *Questões de segurança* a partir da página 314. Aqui, discutimos brevemente os mecanismos do processo.

Para criarmos um registro de chave, especificamos tanto o algoritmo criptográfico que desejamos usar como um "segredo compartilhado" (isto é, a senha) representados na forma de uma string codificada base 64:

```
key id-da-chave {
    algorithm string;
    secret string;
};
```

Como ocorre com as listas de controle de acesso, o *id-da-chave* deve ser definido com uma instrução `key` antes de ser utilizado. Para associar a chave a um servidor, basta incluir o *id-da-chave* na cláusula `keys` da instrução `server` desse servidor. A chave é usada tanto para verificar requisições desse servidor como para assinar as respostas a essas requisições.

A instrução `trusted-keys`

A instrução `trusted-keys` para segurança DNSSEC está especificada na RFC2535. Cada entrada é um conjunto de cinco valores que identifica o nome de domínio, flags, protocolo, algoritmo e chave necessários para verificar os dados nesse domínio. O formato é:

```
trusted-keys {
    chave algoritmo protocolo flags domínio;
    chave algoritmo protocolo flags domínio;
    ...
}
```

Cada linha representa a chave confiável para determinado domínio. *Flags*, *protocolo* e *algoritmo* são inteiros não-negativos. *Chave* é uma string codificada na base 64.

A construção `trusted-keys` é destinada a ser usada quando uma zona tem assinatura, mas sua zona pai não, de modo que você não pode ter certeza de que a chave pública para a zona que obtém do DNS é realmente autêntica. Inserir uma chave confiável com uma instrução `trusted-keys` (usando métodos desvinculados) garante que você realmente tem a chave apropriada para o domínio em questão.

DNSSEC é visto detalhadamente a partir da página 318.

A instrução `server`

named tem potencial para conversar com vários servidores, dos quais nem todos estão executando a versão mais recente de BIND e dos quais nem mesmo todos são nominalmente sãos. A instrução `server` informa **named** sobre as características de seus parceiros remotos.

```
server endereço_ip {
    bogus yes | no;                              [no]
    provide-ixfr yes | no;                       [yes]
    request-ixfr yes | no;                       [yes]
    edns yes | no;                               [yes]
    transfers número;                            [2]
    transfer-format one-answer | many-answers;   [many-answers]
    keys { id-chave; id-chave; … };
    transfer-source endereço-IP [porta]
    transfer-source-v6 endereço-ipv6 [porta]
};
```

Você pode usar uma instrução `server` para sobrescrever os valores das opções de configuração relacionadas ao servidor. Liste apenas aquelas para as quais você quer um comportamento não padronizado.

Se você marcar um servidor como `bogus`, **named** não enviará nenhuma consulta em sua direção. Essa diretiva deve ser reservada aos servidores que são realmente fictícios. `bogus` difere da opção global `blackhole` pelo fato de que ela só suprime consultas externas. Em contraste, a opção `blackhole` elimina completamente todas as formas de comunicação com os servidores listados.

Um servidor de nomes BIND 9 atuando como mestre para uma zona realizará transferências incrementais de zona se `provide-ixfr` estiver configurado como `yes`. Da mesma forma, um servidor atuando como escravo requisitará transferências de zonas incrementais do mestre se `request-ixfr` estiver configurado como `yes`.

A cláusula `edns` determina se o servidor local tentará utilizar o protocolo DNS estendido ao entrar em contato com o servidor remoto. Muitos dos recursos novos do BIND (IPv6 e DNSSEC, por exemplo) geram pacotes maiores do que 512 bytes e, portanto, requerem o emprego do protocolo EDNS para negociar um tamanho de pacote UDP maior.

Os servidores escravos permanecem sincronizados com seus servidores mestres recebendo uma transferência de zona quando os dados mudam (ver a página 311 para mais informações sobre transferências de zona). A cláusula `transfers` limita o número de transferências de zonas de fora para dentro concorrentes do servidor remoto. Ela é uma versão específica do servidor da instrução `transfers-in`, mas, como se aplica somente a um servidor, atua como uma sobrescrição por servidor da opção `transfers-per-ns`. O nome é diferente para preservar compatibilidade com o BIND 8.

As cláusulas `transfer-format` são as formas específicas a servidor das opções discutidas na página 295. Utilize `transfer-format` se você conversa com servidores BIND 8/9 e servidores BIND 4 ou se tiver servidores de nomes Microsoft em sua rede. O Windows NT não pode tratar nada, exceto o formato de uma resposta; o Windows 2000 e superiores são bons com qualquer formato, embora versões anteriores não possam lidar com mensagens DNS maiores que 16K.

A cláusula `keys` identifica um ID chave que foi previamente definido em uma instrução `key` para emprego com assinaturas de transação TSIG (consulte a página 316). Quaisquer requisições enviadas para o servidor remoto são assinadas com essa chave. Requisições originadas no servidor remoto não precisam ser assinadas, mas, se forem, a assinatura será verificada.

As cláusulas `transfer-source` fornecem o endereço IPv4 ou IPv6 da interface (e, opcionalmente, a porta) que deve ser usado como um endereço de origem (porta) para requisições de transferência de zona. Essa cláusula só é necessária quando o sistema tem múltiplas interfaces e o servidor remoto definiu um endereço IP específico na sua cláusula `allow-transfer`.

A instrução `masters`

Como pode haver mais de um mestre? Consulte a página 301.

A instrução `masters` permite nomear um conjunto de um ou mais servidores mestres especificando seus endereços IP e suas chaves criptográficas. Você pode então utilizar esse nome na cláusula `masters` das instruções `zone` em vez de repetir os endereços IP e as chaves.

O recurso `masters` pode ser útil quando múltiplas zonas escravas ou zonas stub recebem seus dados dos mesmos servidores remotos. Se os endereços ou chaves criptográficas dos servidores remotos mudarem, você poderá atualizar a instrução `masters` que os introduz em vez de alterar muitas instruções `zone` diferentes.

A sintaxe é:

```
masters nome| endereço_ip [port porta_ip] [key chave] ; ... | ;
```

A instrução `logging`

named é o atual detentor do "sistema de registro em log mais configurável da face da Terra". Syslog coloca a priorização das mensagens de log nas mãos do programador e a administração dessas mensagens, nas mãos do administrador de sistemas. Mas, para uma dada prioridade, o administrador de sistemas não tinha nenhuma maneira de dizer: "Eu me importo com essa mensagem, mas não com aquela". O BIND 8 adicionou categorias que classificam mensagens de log por tipo e canais que aumentam as opções para a administração das mensagens. As categorias são determinadas pelo programador e os canais pelo administrador de sistemas.

Como a questão de registro em log é um tanto tangencial (especialmente dada a quantidade de explicação exigida), a discutiremos no tópico sobre depuração a partir da página 324.

A instrução `zone`

Instruções `zone` são o coração do arquivo **named.conf**. Elas informam **named** sobre as zonas para as quais ele é autorizado e configuram as opções apropriadas para gerenciar cada zona. Uma instrução `zone` também é usada para pré-carregar os hints do servidor-raiz — os nomes e os endereços dos servidores-raiz, que disparam o processo de pesquisa DNS.

O formato exato de uma instrução `zone` varia, dependendo do papel que **named** desempenha com relação a essa zona. Para ser específico, os possíveis tipos de zona são `master`, `slave`, `hint`, `forward`, `stub` e `delegation-only`. Descrevemos o tipo `delegation-only` na página 298. Os outros são descritos nas breves seções a seguir.

Muitas das opções globais abordadas anteriormente podem se tornar parte de uma instrução `zone` e sobrescrever os valores definidos previamente. Não repetiremos essas opções aqui, exceto algumas utilizadas mais freqüentemente.

Configurando o servidor mestre para uma zona

Eis o formato de que você precisa para uma zona da qual esse processo **named** é o servidor mestre:

```
zone "nome_de_domínio" {
    type master;
    file "caminho";
};
```

nome_de_domínio em uma especificação de zona deve sempre estar entre aspas duplas.

Os dados da zona são mantidos em disco em um arquivo de formato legível (e editável). Como não há nenhum padrão para o nome de arquivo, você deve fornecer uma instrução `file` ao declarar uma zona mestre. Um arquivo de zonas é simplesmente um conjunto de registros de recursos DNS; a descrição de formato começa na página 271.

Outros atributos específicos ao servidor também costumam ser especificados dentro da instrução `zone`. Por exemplo:

```
allow-query { lista_de_correspondência_de_endereços };        [any]
allow-transfer { lista_de_correspondência_de_endereços };     [any]
allow-update { lista_de_correspondência_de_endereços };       [nenhum]
zone-statistics yes | no                                      [no]
```

As opções de controle de acesso não são exigidas, porém é uma boa idéia utilizá-las. Se forem utilizadas atualizações dinâmicas para essa zona, a cláusula `allow-update` deve estar presente com uma lista de correspondência de endereços que limita os hosts com base nas quais as atualizações podem ocorrer. As atualizações dinâmicas só se aplicam às zonas mestres; a cláusula `allow-update` não pode ser utilizada para uma zona escrava. Certifique-se de que essa cláusula inclua somente seus servidores DHCP locais e não toda a Internet.[17]

A opção `zone-statistics` faz com que **named** controle as informações estatísticas de consultas/respostas como o número e porcentagem de respostas que eram referrals, recursão demandada ou que resultaram em erros. Consulte os exemplos da página 330.

Com todas essas opções específicas a uma determinada zona (e cerca de 30 outras que não apresentamos aqui), a configuração está começando a ficar complicada. Entretanto, uma declaração de zonas mestres formada por nada além de um nome de caminho para o arquivo de zonas é perfeitamente razoável. No BIND 4, isto é tudo o que você poderia especificar. Eis um exemplo, que modificamos um pouco, da documentação BIND:

```
zone "example.com" {
    type master;
    file "forward/example.com";
    allow-query { any; };
    allow-transfer { my-slaves; };
}
```

Aqui, `my-slaves` seria uma lista de controle de acesso definida anteriormente.

Configurando um servidor escravo para uma zona

A instrução `zone` para um servidor escravo é muito semelhante àquela de um servidor mestre:

```
zone "nome_de_domínio" {
    type slave | stub;
    file "caminho";
    masters { endereço_ip[port porta_ip] [key nome_da_chave]; … };     [nenhum padrão]
    allow-query { lista_de_correspondência_de_endereços };             [any]
    allow-transfer { lista_de_correspondência_de_endereços };          [any]
};
```

Servidores escravos normalmente mantêm uma cópia completa do banco de dados de suas zonas. Porém, se `type` for configurado como `stub` em vez de `slave`, somente registros NS (Name Server) serão transferidos. Zonas stub permitem que os processos **named** para a zona pai descubram automaticamente que máquinas fornecem serviços DNS para suas zonas filhas delegadas, na eventualidade do administrador da zona filha não estar ciente da necessidade de informar a zona pai sobre as alterações. A zona pai precisa dessas informações para fazer referrals apropriadas ou consultas recursivas.

---

17. Você também precisa de filtragem de entrada no firewall; consulte a página 484. Melhor ainda, use o TSIG para autenticação.

A instrução `file` especifica um arquivo local no qual o banco de dados replicado pode ser armazenado. Cada vez que o servidor busca uma nova cópia da zona, ele salva os dados nesse arquivo. Se o servidor travar e reinicializar, o arquivo pode então ser recarregado no disco local sem ser transferido pela rede.

Você não deve editar esse arquivo de cache, porque ele é mantido por **named**. Entretanto, pode ser interessante examiná-lo se suspeitar que você cometeu um erro no arquivo de dados do servidor mestre. O arquivo de disco do escravo nos mostra como **named** interpretou os dados de zona originais; nomes relativos e diretivas de origem foram expandidos. Se você vir um nome no arquivo de dados que se pareça com um destes:

128.138.243.151.cs.colorado.edu.
anchor.cs.colorado.edu.cs.colorado.edu.

pode estar certo de que acabou esquecendo algum ponto final em algum lugar.

A instrução `masters` lista os endereços IP de uma ou mais máquinas das quais o banco de dados de zonas pode ser obtido. Também pode conter o nome de uma lista de masters definida com uma instrução `masters` anterior.

Dissemos que somente uma máquina pode ser o mestre para uma zona, então por que é possível listar mais de um endereço? Duas razões. Em primeiro lugar, a máquina mestra poderia ter mais de uma interface de rede e, conseqüentemente, mais de um endereço IP. É possível para uma interface se tornar inatingível (devido a problemas de rede ou de roteamento) enquanto outras ainda continuem acessíveis. Portanto, é uma boa prática listar todos os endereços topologicamente distintos do servidor mestre.

Em segundo lugar, **named** realmente não se importa de onde provêm os dados. Ele pode extrair igualmente o banco de dados de um servidor escravo como de um servidor mestre. Você poderia utilizar esse recurso para permitir que um servidor escravo bem conectado servisse como uma espécie de mestre backup, uma vez que os endereços IP seriam tentados em ordem até que um servidor operacional fosse encontrado. Teoricamente, você também pode configurar a hierarquia de servidores, com um mestre atendendo a vários servidores de segundo nível que, por sua vez, servem vários servidores de terceiro nível.

Configurando hints no servidor-raiz

Uma outra forma de instrução `zone` aponta **named** em direção a um arquivo a partir do qual ela pode pré-carregar seu cache com os nomes e endereços dos servidores de nomes-raiz:

```
zone "." {
    type hint;
    file "caminho";
};
```

As "hints" são um conjunto de registros DNS que listam servidores para o domínio-raiz. Elas são necessárias para dar a **named** um ponto de partida para busca de informações sobre os domínios dos demais sites. Sem elas, **named** saberia apenas sobre os domínios que ele efetivamente atende e seus subdomínios.

Quando **named** é inicializado, ele recarrega as hints de um dos servidores-raiz. Portanto, tudo estará bem desde que seu arquivo hints contenha ao menos um servidor-raiz válido, alcançável.

O BIND 9 tem hints de servidor-raiz compiladas em seu código, de modo que nenhuma configuração da zona-raiz é realmente necessária. Entretanto, se você fornecer um arquivo hints, o BIND 9 o usará. Recomendamos que você forneça hints explícitas; a política invadiu a arena do DNS, tornando os servidores de nomes-raiz e seus endereços IP mais voláteis.

O arquivo hints é freqüentemente chamado **root.cache**. Ele contém a resposta que você receberia se consultasse em um servidor-raiz os registros do servidor de nomes no domínio-raiz. De fato, você pode gerar o arquivo hints exatamente dessa maneira executando **dig**:

```
$ dig @f.root-servers.net . ns > root.cache
```

Lembre-se do ponto. Se f.root-servers.net não estiver respondendo, você pode executar a consulta sem especificar um determinado servidor:

```
$ dig . ns > root.cache
```

A saída será semelhante; mas, você obterá a lista de servidores-raiz do cache de um servidor de nomes local, não de uma fonte autorizada. Essa deve ser excelente — mesmo se você não tiver reinicializado o servidor de nomes há um ou dois anos, ele vem atualizando seus registros de servidor-raiz periodicamente à medida que seus TTLs expiram.

Eis com que o arquivo de cache se parece (resumido a partir de **dig @f.root-servers.net . ns**):

```
; <<>> DiG 9.3.0rc1 <<>> @f.root-servers.net . ns
;; global options:  printcmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28797
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13

;; QUESTION SECTION:
;.                          IN    NS

;; ANSWER SECTION:
.               518400      IN    NS      I.ROOT-SERVERS.NET.
.               518400      IN    NS      J.ROOT-SERVERS.NET.
.               518400      IN    NS      K.ROOT-SERVERS.NET.
.               518400      IN    NS      L.ROOT-SERVERS.NET.
...
;; ADDITIONAL SECTION:
I.ROOT-SERVERS.NET. 3600000       IN      A       192.36.148.17
J.ROOT-SERVERS.NET. 3600000       IN      A       192.58.128.30
K.ROOT-SERVERS.NET. 3600000       IN      A       193.0.14.129
L.ROOT-SERVERS.NET. 3600000       IN      A       198.32.64.12
...
;; Query time: 3 msec
;; SERVER: 192.5.5.241#53(f.root-servers.net)
;; WHEN: Wed Jul  7 13:54:26 2004
;; MSG SIZE  rcvd: 436
```

Note os pontos que iniciam o primeiro conjunto de registros; eles não são aleatórios, mas sim definem o domínio (raiz) ao qual os registros NS se aplicam.

Configurando uma zona direta

Uma zona do tipo `forward` sobrescreve as configurações de encaminhamento global de **named** (descritas na página 297) para determinado domínio:

```
zone "nome_de_dominio" {
    type forward;
    forward only | first;
    forwarders { endereço_ip; endereço_ip; ... };
};
```

Você poderia utilizar uma zona `forward` se sua organização tivesse um acordo estratégico em vigência com algum outro grupo ou empresa e você quisesse afunilar o tráfego diretamente para os servidores de nomes daquela empresa, driblando o caminho de consultas padrão.

A instrução `controls`

A instrução `controls` especifica como **rndc** controla um processo **named** em execução. **rndc** pode iniciar e parar **named**, fazer um dump do seu estado, colocá-lo no modo de depuração etc. **rndc** é um programa de rede e, com uma configuração imprópria, talvez permita que qualquer pessoa na Internet desorganize seu servidor de nomes. A sintaxe é:

```
controls {
    inet endereço_ip port porta-ip allow { lista_de_correspondência_de_endereços | keys {
    lista_de_chaves}};
}
```

A porta que **rndc** utiliza para conversar com **named** assume por padrão a porta 953 se ela não for especificada com a cláusula `port`.

Permitir que o seu servidor de nomes seja controlado remotamente parece algo prático, porém, ao mesmo tempo, perigoso. Autenticação forte por meio de uma entrada `key` na cláusula `allow` é exigida; chaves na lista de correspondência de endereços são ignoradas e devem ser declaradas explicitamente na cláusula `keys` da instrução `controls`.

No BIND 9, você pode utilizar o comando **rndc-confgen** para gerar uma chave de autenticação destinado ao uso entre **rndc** e **named**. Há essencialmente duas maneiras de configurar isso: você pode fazer com que **named** e **rndc** consultem o mesmo arquivo de configuração para que conheçam a chave (**/etc/rndc.key**) ou você pode incluir a chave tanto nos arquivos de configuração **rndc** como nos arquivos **named** (**/etc/rndc.conf** para **rndc** e **/etc/named.conf** para **named**). A última opção é mais complicada, porém necessária quando **named** e **rndc** executam em computadores diferentes.

Quando nenhuma instrução `controls` está presente, o BIND 9 assume o padrão do endereço de loopback para a lista de correspondência de endereços e procura a chave em **/etc/rndc.key**. Como a autenticação forte é obrigatória no BIND 9,

você não pode utilizar o comando **rndc** para controlar **named** se não houver nenhuma chave. Isso pode parecer um tanto draconiano, mas considere o seguinte: mesmo se **rndc** funcionasse somente a partir de (127.0.0.1) e esse endereço fosse bloqueado do mundo externo no seu firewall, você ainda estaria confiando que todos os usuários locais não desorganizariam seu servidor de nomes. Qualquer usuário poderia enviar um **telnet** para a porta de controle e digitar "stop" — um ataque de negação de serviços bastante efetivo.

Eis um exemplo da saída (para padronizar) do **rndc-confgen** quando uma chave de 256 bits é requisitada. Escolhemos 256 porque ela cabe na página. Você normalmente escolheria uma chave mais longa e redirecionaria a saída para **/etc/rndc.conf**. Os comentários na parte inferior da saída mostram as linhas que precisam ser adicionadas a **named.conf** para fazer com que **named** e **rndc** reproduzam conjuntamente.

```
$ ./rndc-confgen -b 256
# Start of rndc.conf
key "rndc-key" {
     algorithm hmac-md5;
     secret "orZuz5amkUnEp52zlHxD6cd5hACldOGsG/elP/dv2IY=";
};

options {
     default-key "rndc-key";
     default-server 127.0.0.1;
     default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as needed:
# key "rndc-key" {
#       algorithm hmac-md5;
#       secret "orZuz5amkUnEp52zlHxD6cd5hACldOGsG/elP/dv2IY=";
# };
#
# controls {
#       inet 127.0.0.1 port 953
#               allow { 127.0.0.1; } keys { "rndc-key"; };
# };
# End of named.conf
```

O modo "automático", utilizado para gerar um arquivo de configuração compartilhado, é mais simples. Executar como root, **rndc-confgen -a -b 256** produz o arquivo **/etc/rndc.key**, que contém

```
key "rndc-key" {
     algorithm hmac-md5;
     secret "laGbZj2CobycOm/jFVNCu8OJzsLKNH+CCb2JCWY6yJw=";
};
```

O arquivo possui modo 600 e seu proprietário deve ser o usuário **named** ou o root. Ele pode então ser incorporado ao arquivo **named.conf** com a instrução `include`.

DNS dividido e a instrução `view`

Muitas instalações querem que a vista interna de suas redes seja diferente daquela observada externamente por quem as acessa via Internet. Por exemplo, você poderia revelar todos os hosts de uma zona a usuários internos, mas restringir a vista externa a alguns poucos servidores bem conhecidos. Ou, então, você poderia expor o mesmo conjunto de hosts em ambas as vistas, mas fornecer registros adicionais (ou diferentes) para usuários internos. Por exemplo, os registros MX para roteamento de e-mail poderiam apontar para uma única máquina centralizadora de e-mail (mail hub) de fora do domínio, mas apontar para estações de trabalho individuais sob a perspectiva de usuários internos.

Consulte a página 203 para informações adicionais sobre espaços de endereçamento privados.

Uma configuração de DNS dividido é especialmente útil para sites que utilizam endereços IP privados da RFC1918 em suas redes internas. Por exemplo, uma consulta buscando o nome de host associado ao endereço IP 10.0.0.1 nunca poderá ser respondida pelo sistema DNS global, mas faz sentido dentro do contexto da rede local. Das consultas que chegam aos servidores de nomes-raiz, 4% a 5% são *de* um endereço IP em um dos intervalos de endereços privados ou *sobre* um destes endereços. Nenhuma delas pode ser respondida; ambas são resultantes de uma configuração incorreta, seja do DNS dividido do BIND ou por parte dos "domínios" Microsoft.

A instrução `view` empacota uma lista de acesso que controla quais clientes têm quais vistas, algumas opções que se aplicam a todas as zonas da vista e, finalmente, as zonas em si. A sintaxe é:

```
view nome-da-vista {
     match-clients { lista_de_correspondência_de_endereços } ;
     opção_de_vista; …
     instrução_zone; …
} ;
```

A cláusula `match-clients` controla quem pode ver a vista. As vistas são processadas em ordem; portanto, coloque as vistas mais restritas em primeiro lugar. Zonas nas diferentes visualizações podem ter os mesmos nomes, mas selecionam seus dados a partir de diferentes arquivos. As vistas são uma proposição ou tudo ou nada; se você utilizá-las, todas as instruções `zone` contidas em seu arquivo **named.conf** deverão aparecer no contexto de uma `view`.

Eis um exemplo da documentação BIND 9. As duas vistas definem a mesma zona, porém com dados diferentes.

```
view "internal" {
     match-clients { nossas_redes; };    // only internal networks
     recursion yes;                      // internal clients only
     zone "example.com" {                // complete view of zone
           type master;
           file "example-internal.db";
     };
};

view "external" {
     match-clients { any; };             // allow all queries
     recursion no;                       // but no recursion
     zone "example.com" {                // only "public" hosts
           type master;
           file "example-external.db";
     }
     };
```

Se a ordem das vistas fosse invertida, ninguém jamais teria a vista interna. Os hosts internos corresponderiam ao valor `any` na cláusula `match-clients` da vista externa antes de atingirem a vista interna.

Nosso segundo exemplo da configuração DNS a partir da página 306 fornece alguns exemplos adicionais das vistas.

15.12 Exemplos de configuração BIND

Agora que exploramos as maravilhas de **named.conf**, vamos examinar alguns exemplos completos. Nas seções a seguir, discutimos várias configurações de exemplo:

- A zona localhost
- Uma pequena empresa especializada em segurança de sistemas que usa DNS dividido
- Os especialistas: isc.org, o Internet Systems Consortium

A zona localhost

O endereço 127.0.0.1 refere-se a um host próprio e deve ser mapeado para o nome "localhost".[18] Alguns sites mapeiam o endereço para "localhost.*domínioLocal*." e outros mapeiam para os dois. O endereço IPv6 correspondente é ::1.

Se você se esquecer de configurar a zona localhost, seu site pode acabar consultando os servidores-raiz em busca dessas informações. Os servidores-raiz recebem tantas dessas consultas que os operadores estão considerando a possibilidade de adicionar um mapeamento genérico entre localhost e 127.0.0.1 no nível-raiz. Em medições no servidor-raiz chamado F em São Francisco em março de 2004, o localhost era o terceiro domínio mais popular consultado, atrás apenas do arpa e do com. Essa é uma grande quantidade de consultas inúteis para um servidor de nomes ocupado.

O mapeamento direto para o nome localhost.*domínioLocal* pode ser definido no arquivo de zona direta para o domínio ou em seu próprio arquivo. Cada servidor, mesmo um servidor de cache, normalmente é o mestre para o próprio domínio localhost reverso.

---

18. Na verdade, toda a rede 127/8 classe A refere-se a localhost, mas a maioria das pessoas simplesmente utiliza 127.0.0.1.

Eis as linhas em **named.conf** que configuram o localhost:

```
zone "localhost" { // localhost forward zone
    type master;
    file "for/localhost";
    allow-update { none; };
};
zone "0.0.127.in-addr.arpa" {// localhost reverse zone
    type master;
    file "rev/127.0.0";
    allow-update { none; };
};
```

O arquivo de zona direta correspondente, **for/localhost**, contém

```
$TTL 30d
; localhost.
@          IN      SOA     localhost. postmaster.localhost. (
                                   1998050801     ;serial
                                   3600           ;refresh
                                   1800           ;retry
                                   604800         ;expiration
                                   3600 )         ;minimum

                   NS      localhost.
                   A       127.0.0.1
```

e o arquivo reverso, **rev/127.0.0**:

```
$TTL 30d
; 0.0.127.in-addr.arpa
@          IN      SOA     localhost. postmaster.localhost. (
                                   1998050801     ;serial
                                   3600    ;refresh
                                   1800    ;retry
                                   604800  ;expiration
                                   3600 )  ;minimum

           NS      localhost.
1          PTR     localhost.
```

O mapeamento reverso para o endereço localhost (127.0.0.1) nunca muda, de modo que os tempos-limite podem ser grandes. Observe o número serial, que codifica a data; o arquivo foi alterado pela última vez em 1998. Note também que apenas o servidor de nomes mestre é listado para o domínio localhost. O significado de @ aqui é "0.0.127.in-addr.arpa.".

Uma pequena empresa especializada em segurança de sistemas

Nosso primeiro exemplo é o de uma pequena empresa especializada em consultoria sobre segurança. Eles são executados no BIND 9 em uma versão recente do Red Hat Enterprise Linux e utilizam vistas para implementar um sistema DNS dividido em que usuários internos e externos vêem diferentes dados sobre o host. Eles também utilizam espaço de endereços privado internamente; consultas sobre esses endereços nunca devem escapar para a Internet para bagunçar o sistema DNS global. Eis o arquivo **named.conf**, reformatado e com alguns comentários:

```
options {
    directory var/domain";
    version "root@atrust.com";
    allow-transfer {82.165.230.84; 71.33.249.193; 127.0.0.1; };
    listen-on { 192.168.2.10; 192.168.2.1; 127.0.0.1; 192.168.2.12; };
};

include "atrust.key";                    // defn of atkey in mode 600 file

controls {
    inet 127.0.0.1 allow { 127.0.0.1; } keys { atkey; };
};
```

```
view "internal" {                       // internal view

    match-clients { 192.168.0.0/16; 206.168.198.192/28; 172.29.0.0/24; };
    recursion yes;

    zone "." IN {                       // root hints zone
            type hint;
            file "named.cache";
    };
    zone "localhost" IN {               // localhost forward zone
            type master;
            file "localhost.forw";
            allow-update { none; };
    };
    zone "0.0.127.in-addr.arpa" IN {    // localhost reverse zone
            type master;
            file "localhost.rev";
            allow-update { none; };
    };
    zone "atrust.com" IN {              // internal forward zone
            type master;
            file "internal/atrust.com";
    };
    zone "1.168.192.in-addr.arpa" IN {  // internal reverse zone
            type master;
            file "internal/192.168.1.rev";
            allow-update { none; };
    };                                  // lots more reverse zones omitted
    zone "atrust.us" {                  // slave in internal view
            type slave;
            file "internal-slave/atrust.us";
            masters { 127.0.0.1; };
    };
    zone "atrust.org" {
            type slave;
            file "internal-slave/atrust.org";
            masters { 127.0.0.1; };
    };                                  // lots more zones omitted
};  // end of internal view

view "world" {    // external view

    match-clients { any; };
    recursion no;

    // zone statements for dot and localhost as above, omitted for brevity

    zone "atrust.com" {                 // external forward zone
            type master;
            file "world/atrust.com";
            allow-update { none; };
    };
    zone "189.173.63.in-addr.arpa" {    // external reverse zone
            type master;
            file "world/63.173.189.rev";
            allow-update { none; };
    };
    zone "atrust.us" {                  // master in world view
                    type master;
                    file "world/appliedtrust.com-aggregate";
                    allow-update { none; };
    };
```

```
    zone "atrust.org" {                 // note that the file is the same
            type master;
            file "world/appliedtrust.com-aggregate";
            allow-update { none; };
    };                                  // lots more atrusty zones omitted
    zone "admin.com" {                  // master zones only in world view
            type master;
            file "world/admin.com";
            allow-update { none; };
    };                                  // lots more master zones omitted
    zone "eldoraskipatrol.org" {        // slave zones only in world view
            type slave;
            file "world-slave/eldoraskipatrol.org";
            masters { 192.231.139.1; };
    };                                  // lots more slave zones omitted
};// end of external view
```

O arquivo **atrust.key** contém a definição da chave chamada "atkey".

```
key "atkey" {
     algorithm hmac-md5;
     secret "chave secreta compartilhada entra aqui";
};
```

As zonas são organizadas de acordo com suas vistas e com o tipo (mestre/escravo) e a convenção para atribuição de nomes de arquivos de dados de zona reflete esse mesmo esquema. Esse servidor é recursivo para a vista interna, o que inclui todos os hosts locais, incluindo muitos que utilizam endereçamento privado. O servidor não é recursivo para a vista externa, que só contém hosts selecionados em atrust.com e as zonas externas para as quais eles fornecem serviço de DNS mestre ou escravo.

Os trechos dos arquivos **internal/atrust.com** e **world/atrust.com** são mostrados a seguir. Em primeiro lugar, o arquivo interno:

```
; atrust.com - internal file
$TTL 86400
$ORIGIN com.
atrust                    3600   SOA    ns1.atrust.com. trent.atrust.com. (
                                                2004012900 10800 1200 3600000 3600 )
                          3600   NS     NS1.atrust.com.
                          3600   NS     NS2.atrust.com.
                          3600   MX     10 mailserver.atrust.com.
                          3600   A      66.77.122.161

$ORIGIN atrust.com.
ns1                              A      192.168.2.11
ns2                              A      66.77.122.161
www                              A      66.77.122.161
mailserver                       A      192.168.2.11
exchange                         A      192.168.2.100
secure                           A      66.77.122.161
bark                             A      192.168.2.10
superg                           A      192.168.1.249
at-dmz-gw                        A      192.168.1.254
at-external-gw                   A      192.168.2.254
at-external-outside-gw           A      206.168.198.220
indras-gw                        A      206.168.198.222
; dhcp host range
dhcp-0-hardwire           IN     A      192.168.1.64
dhcp-1-hardwire           IN     A      192.168.1.65
; ...

; booklab "subdomain", note that this is a subdomain but not a subzone

redhat.booklab            IN     A      192.168.10.1
redhat-ent.booklab        IN     A      192.168.10.2
debian.booklab            IN     A      192.168.10.3
```

```
fedora.booklab          IN      A       192.168.10.4
freebsd.booklab         IN      A       192.168.10.5
suse.booklab            IN      A       192.168.10.6
macos.booklab           IN      A       192.168.10.7
solaris.booklab         IN      A       192.168.10.8
```

Eis a vista externa desse mesmo domínio em **world/atrust.com**:

```
; atrust.com - external file
$TTL 57600
$ORIGIN .
atrust.com              SOA     ns1.atrust.com. trent.atrust.com. (
                                        2004020400 10800 1200 3600000 3600 )
                        NS      NS1.atrust.com.
                        NS      NS2.atrust.com.
                        MX      10 mailserver.atrust.com.
                        A       66.77.122.161
ns1.atrust.com.         A       206.168.198.209
ns2.atrust.com.         A       66.77.122.161

$ORIGIN atrust.com.
www                     A       66.77.122.161
mailserver              A       206.168.198.209
bark                    A       206.168.198.209
secure                  A       66.77.122.161

; reverse maps
exterior1               A       206.168.198.209
209.198.168.206         PTR     exterior1.atrust.com.
exterior2               A       206.168.198.213
213.198.168.206         PTR     exterior2.atrust.com.
exterior3               A       206.168.198.220
220.198.168.206         PTR     exterior3.atrust.com.
exterior4               A       206.168.198.210
210.198.168.206         PTR     exterior4.atrust.com.
```

Note que quando são necessários vários nomes para a mesma máquina, eles são apresentados como registros A adicionais em vez de registros CNAME. Poucos hosts são realmente visíveis na vista externa.

O TTL nesses arquivos de zona é configurado como 16 horas (57.600 segundos). Para zonas internas, o TTL é um dia (86.400 segundos). A maioria dos registros individuais nos arquivos de zonas não recebe um valor TTL explícito. O TTL é opcional; ele pode ser adicionado a qualquer linha individual no arquivo de zonas antes do tipo de registro.

Os bizarros registros PTR no final do arquivo externo permitem ao ISP de atrust.com delegar o mapeamento reverso de um trecho muito pequeno do espaço de endereços. Os registros CNAME no site do ISP fazem com que essa variação do uso de CNAME funcione; consulte a página 278 para informações adicionais.

O Internet Systems Consortium, isc.org

ISCs são os autores e mantenedores do BIND bem como o operador do servidor de nomes-raiz F. Eles também são um servidor TLD que serve cerca de 60 domínios de primeiro nível. Essa é a razão pela qual os chamamos de especialistas!

A seguir estão os trechos dos seus arquivos de configuração. Observe que eles utilizam tanto o IPv4 como o IPv6. Eles também usam a criptografia TSIG para autenticação entre servidores mestres e servidores escravos para transferências de zona. As opções `transfer-source` asseguram que os endereços IP de origem para requisições enviadas de transferência de zona obedeçam às instruções `allow-transfers` nos servidores mestres.

O arquivo **named.conf**:

```
// isc.org TLD name server

options {
    directory "/var/named";
    datasize 1000M;
    listen-on { 204.152.184.64; };
    listen-on-v6 { 2001:4f8:0:2::13; };
    recursion no;
```

```
    transfer-source 204.152.184.64;
    transfer-source-v6 2001:4f8:0:2::13;
};

// rndc key
key rndc_key {
    algorithm hmac-md5;
    secret "<segredo>";
};
// TSIG key for name server ns-ext
key ns-ext {
    algorithm hmac-md5;
    secret "<código secreto>";
};

server 204.152.188.234 { keys { ns-ext; }; };

controls {
    inet 204.152.184.64 allow { any; } keys { rndc_key; };
};

include "inf/named.zones";       // root, localhost, 127.0.0.1, ::1
include "master.zones";          // zones we master
include "slave.zones";           // lots of slaves
```

Essas instruções `include` mantêm o arquivo **named.conf** organizado e curto. Se você servir a uma grande quantidade de zonas, pense em dividir sua configuração em partes menores como esta. Mais importante, configure a hierarquia do seu sistema de arquivos de modo que você não tenha um diretório com milhares de arquivos de zona. Sistemas de arquivos Linux modernos tratam grandes diretórios eficientemente, mas gerenciar esses sistemas pode ser bastante trabalhoso.

Eis mais do arquivo **master.zones**:

```
zone "isc.org" {   // ISC
    type master;
    file "master/isc.org";
    allow-update { none; };
    allow-transfer { none; };
};

zone "sfo2.isc.org" {     // ISC
    type master;
    file "master/sfo2.isc.org";
    allow-update { none; };
    allow-transfer { none; };
};

// lots of zones truncated
```

E do **slaves.zones**:

```
zone "vix.com" {
    type slave;
    file "secondary/vix.com";
    masters { 204.152.188.234; };
};

zone "cix.net" {
    type slave;
    file "secondary/cix.net";
    masters { 204.152.188.234; };
};
```

Com as informações sobre a configuração do DNS e do BIND já discutidas, agora passaremos para as questões mais operacionais da execução de um serviço de nomes no seu site, incluindo a manutenção dos arquivos de zona, questões de segurança, teste e depuração.

15.13 INICIANDO NAMED

A existência do BIND nas distribuições do Linux depende tanto da distribuição como das opções selecionadas quando a distribuição foi instalada. Se parece que o BIND não foi instalado, você terá de obter e instalar o pacote BIND. Esse pacote também instala um script de inicialização para **named** que é executado por meio de **init**: **/etc/init.d/named** para RHEL, Fedora e SUSE e **/etc/init.d/bind9** para Debian e Ubuntu.

named é inicializado em tempo de inicialização e executado continuamente. Para controlar uma cópia em execução de **named**, utilize a interface de linha de comandos **rndc**, descrita com mais detalhes a partir da página 329.

Às vezes, os scripts **init** que iniciam **named** fornecem pontos de entrada extras (por exemplo, **reload**) que são concebidos para uso pelos administradores de sistemas. Entretanto, a implementação em algumas distribuições de sistemas é questionável e é mais fácil utilizar **rndc** de qualquer maneira. Sugerimos deixar os scripts de inicialização para seu sistema operacional e utilizar **init** e **rndc** para controle depois que **named** foi iniciado.

*Consulte a página 609 para obter informações adicionais sobre o **inetd**.*

named utiliza syslog e, conseqüentemente, **syslogd** deve ser inicializado antes de **named**. Não utilize **inetd** ou **xinetd** para controlar **named**; ele irá reiniciar **named** toda vez que for necessário, aumentando os tempos de resposta e impedindo que qualquer cache útil seja desenvolvido. Alguns sites utilizam um script *keep-running* que coloca o comando de inicialização **named** em um loop infinito para se proteger contra a morte do **named**; consulte a página 290 para um exemplo. A distribuição BIND inclui um script chamado **nanny** que serve a um propósito semelhante.

15.14 ATUALIZANDO ARQUIVOS DE ZONAS

Ao efetuar uma alteração em um domínio (como adicionar ou excluir um host), os arquivos de dados no servidor mestre devem ser atualizados. Você também deve incrementar o número serial no registro SOA para a zona e então executar **rndc reload** para sinalizar **named** a selecionar as alterações. Você também pode matar e reiniciar o processo **named**, mas essa operação faz com que dados armazenados em cache provenientes de outros domínios sejam descartados.

Versões mais antigas do BIND utilizavam sinais e o comando **kill** do Linux para controlar **named**, mas somente quando os desenvolvedores começaram a ficar sem números de sinalização, o **rndc** entrou em cena e corrigiu tudo. O material histórico sobre sinais foi removido do BIND 9 (exceto para o sinal HUP para reler o arquivo de configuração, recarregar zonas e reiniciar **named** e os sinais TERM e INT para eliminar **named**), assim recomendamos que utilize **rndc**.

Dados de zona atualizados são propagados para os servidores escravos imediatamente porque a opção `notify` está ativada por padrão. Se, inadvertidamente, você desativar essa opção, seus servidores escravos não coletarão as alterações até se passar númeroDeSegundos para *refresh*, conforme configurado no registro SOA da zona (tipicamente de uma hora a seis horas depois). Se você quiser uma atualização mais oportuna quando a opção `notify` estiver desativada, o **rndc reload** em um escravo faz com que ele verifique com o mestre, veja que dados foram modificados e solicite uma transferência de zonas.

Não se esqueça de modificar tanto as zonas diretas como as reversas ao modificar o nome de um host ou endereço IP. Esquecer os arquivos reversos deixa erros sorrateiros: alguns comandos funcionam e outros não.

Alterar os arquivos de dados, mas se esquecer de alterar o número serial faz com que as alterações entrem em vigor no servidor mestre (após uma recarga), mas não nos escravos.

É inadequado editar arquivos de dados pertencentes aos servidores escravos. Esses arquivos são mantidos por **named**; os administradores de sistemas não devem se meter com eles. Não há nenhum problema em verificar os arquivos de dados desde que você não faça alterações.

O BIND permite que sejam realizadas alterações em zonas por meio de uma API de programação, conforme especificado na RFC2136. Esse recurso, chamado de atualizações dinâmicas, é necessário para suportar protocolos de autoconfiguração como o DHCP. O mecanismo de atualização dinâmica é descrito na página 312.

Transferências de zonas

Os servidores DNS são sincronizados por um mecanismo chamado transferência de zonas. A especificação DNS original (e BIND 4) exigia que todos os dados de zona fossem transferidos de uma só vez. Atualizações incrementais foram por fim definidas na RFC1995 e implementadas no BIND 8.2. Transferências de zonas originais e do tipo incremental são algumas vezes chamadas de, respectivamente, AXFR e IXFR. Uma vez configuradas, elas são supostamente equivalentes.

Um escravo que quer atualizar seus dados solicita uma transferência de zona do servidor mestre e faz uma cópia backup dos dados de zona em disco. Se os dados no mestre não tiverem sido modificados, conforme determinado por uma comparação dos números seriais (não os dados em si), não ocorrerá nenhuma atualização de dados e os arquivos de backup serão somente tocados (isto é, o horário de modificação deles será configurado como o horário atual).

As transferências de zona utilizam o protocolo TCP na porta 53 e registram em log as informações com a categoria "transfer-*". IXFR, conforme especificado pelo IETF, pode utilizar TCP ou UDP, mas BIND só o implementou em TCP.

Tanto o servidor que está enviando como o que está recebendo permanecem disponíveis para responder a consultas durante uma transferência de zona. Apenas depois de a transferência ter sido completada o escravo começa a usar os novos dados. Na verdade, o BIND 8 chama um programa **named-xfer** separado para realizar a transferência, mas o processo **named** do BIND 9 trata diretamente das transferências.

Quando as zonas são imensas (como com) ou atualizadas dinamicamente (veja a próxima seção), as alterações são tipicamente menores em relação ao tamanho de toda a zona. Com o IXFR, somente as alterações são enviadas (a menos que elas sejam maiores do que a zona inteira, caso em que é feita uma transferência AXFR comum). O mecanismo IXFR é como o programa **patch** no sentido de que ele aplica diferenças a um banco de dados antigo para sincronizá-lo com um novo banco de dados.

No BIND 9, IXFR é o padrão para quaisquer zonas configuradas para atualização dinâmica e **named** sempre mantém um log das transações chamado *NomeDaZona*.**jnl**. Você pode configurar as opções `provide-ixfr` e `request-ixfr` nas instruções `server` para peers individuais. `provide-ixfr` ativa ou desativa o serviço IXFR das zonas para as quais esse servidor é o mestre. `request-ixfr` solicita IXFRs para as zonas em que esse servidor é um escravo.

```
provide-ixfr yes          :# in BIND 9 server statement
request-ixfr yes          :# in BIND 9 server statement
```

Começando com o BIND 9.3, **named** agora pode fornecer IXFRs para servidores escravos a fim de transferir edições que são feitas manualmente. Utilize a opção de zona `ixfr-from-differences` para ativar esse comportamento; consulte a página 298.

IXFR requer que o arquivo de zona seja classificado em uma ordem canônica. **named** cuida dessa tarefa para você, mas o requisito torna IXFRs uma relação de troca ou compensação. Mais memória e CPU são utilizados nos servidores em troca para tráfego de rede reduzido.

Foram investidos muitos esforços para assegurar que uma queda de servidor durante um IXFR não deixe zonas com dados inúteis. Uma requisição IXFR a um servidor que não o suporte automaticamente retorna a uma transferência de zona AXFR padrão.

Atualizações dinâmicas

O sistema DNS se baseia na premissa de que as associações de nome para endereço são relativamente estáveis e não são alteradas com freqüência. Mas, um site que usa DHCP para atribuir endereços IP de modo dinâmico enquanto as máquinas são inicializadas e se conectam à rede quebra essa regra constantemente. Há duas soluções clássicas: adicionar entradas genéricas ao banco de dados DNS ou editar continuamente os arquivos DNS. Para muitos sites, nenhuma das soluções é satisfatória.

A primeira solução deve ser familiar para qualquer um que tenha utilizado um ISP discado. A configuração DNS ficará parecida com a seguinte:

```
dhcp-host1.domain.        IN     A      192.168.0.1
dhcp-host2.domain.        IN     A      192.168.0.2
...
```

Embora essa seja uma solução simples, isso significa que nomes de hosts são associados permanentemente a determinados endereços IP e que os computadores, conseqüentemente, mudam nomes de hosts sempre que eles receberem um novo endereço IP. Registro em log baseado em nomes de hosts ou medidas de segurança se tornam muito difíceis nessa situação.

O recurso de atualização dinâmica em versões recentes do BIND fornece uma solução alternativa. Ele permite que o daemon DHCP notifique o BIND sobre as atribuições de endereços feitas, conseqüentemente, atualizando o conteúdo do banco de dados DNS em tempo real. As atualizações dinâmicas podem adicionar, excluir ou modificar registros de recursos. Enquanto atualizações dinâmicas estão ocorrendo, um arquivo de diário (*NomeDaZona*.**jnl**) é mantido para proteger contra a ocorrência de uma queda de servidor.

Você não pode editar manualmente uma zona atualizada dinamicamente sem antes parar o fluxo da atualização dinâmica. **rndc freeze** *zone* ou **rndc freeze** *zone class view* fará o truque. Esses comandos sincronizam o arquivo journal para o arquivo de zona mestre no disco e, então, excluem o arquivo journal. Você poderá então editar manualmente o arquivo de zonas. Naturalmente, a formatação original do arquivo de zona foi destruída pela alteração feita por **named** — o arquivo será parecido com aqueles mantidos pelo **named** para servidores escravos.

Tentativas de atualização dinâmica serão recusadas enquanto a zona está congelada; utilize **rndc thaw** com os mesmos argumentos com os quais você congelou para recarregar o arquivo de zona a partir do disco e reativar as atualizações dinâmicas.

O programa **nsupdate** fornecido juntamente com o BIND 9 fornece uma interface de linha de comando para fazer atualizações dinâmicas. Ela é executada em modo lote, extraindo comandos do teclado ou de um arquivo. Uma linha em branco ou de envio sinaliza o fim de uma atualização e envia as alterações ao servidor. Duas linhas em branco significam o fim da entrada. A linguagem de comando inclui uma instrução primitiva if para expressar construções do tipo: "se esse

nome de host não existir no DNS, adicione-o". Como predicados para uma ação **nsupdate**, você pode requerer que um nome exista ou não, ou requerer que um conjunto de registros de recursos exista ou não.

Eis, por exemplo, um script **nsupdate** simples que adiciona um novo host e também um apelido para um host existente se o apelido já não estiver em uso. O prompt de sinal de menor e maior é produzido por **nsupdate** e não faz parte do script de comando.

```
$ nsupdate
> update add newhost.cs.colorado.edu 86400 A 128.138.243.16
>
> prereq nxdomain gypsy.cs.colorado.edu
> update add gypsy.cs.colorado.edu CNAME evi-laptop.cs.colorado.edu
```

Realizar atualizações dinâmicas no DNS é amedrontador. Elas têm o potencial de fornecer acesso de gravação incontrolado para dados de sistemas importantes. Não tente utilizar endereços IP para controle de acesso — eles são facilmente forjados. A autenticação TSIG com uma chave secreta compartilhada é mais indicada; ela se encontra disponível e é fácil de ser configurada. O BIND 9 suporta os dois:

```
$ nsupdate -k dirDaChave:ArquivoDaChave
```

ou

```
$ nsupdate -y NomeDaChave:senha
```

Como a senha é inserida na linha de comando, qualquer pessoa executando **w** ou **ps** no momento certo pode vê-la. Por essa razão, a forma **-k** é preferível. Para mais detalhes sobre o TSIG, consulte a seção que começa na página 316.

As atualizações dinâmicas para uma zona são ativadas em **named.conf** com uma cláusula `allow-update` ou `update-policy`. `allow-update` concede permissão para atualizar quaisquer registros de acordo com o endereço IP de origem ou autenticação baseada em chave. `update-policy` é uma extensão do BIND 9 que permite controle fino para atualizações de acordo com o nome de host ou tipo de registro. Ela requer autenticação baseada em chave. Ambas são opções de zona.

Use `update-policy` para permitir que os clientes atualizem seus registros A ou PTR, mas não alterem o registro SOA, registros NS ou registros KEY. Você também pode usá-la para permitir que um host atualize apenas seus próprios registros. Os parâmetros permitem que você expresse nomes explicitamente, na forma de um subdomínio, na forma de um curinga ou na forma da palavra-chave `self`, que configura uma política genérica para acesso das máquinas a seus próprios registros. Os registros de acesso são identificados por classe e tipo. A sintaxe de uma regra `update-policy` é:

```
update-policy { grant | deny } identidade TipoDeNome nome [tipos] ;
```

identidade é o nome da chave criptográfica necessária para autorizar a atualização. *TipoDeNome* tem um de quatro valores: `name`, `subdomain`, `wildcard` ou `self`. *nome* é a zona a ser atualizada e *tipos* são os tipos de registro de recursos que podem ser atualizados. Se nenhum tipo for especificado, todos os tipos, exceto SOA, NS, RRSIG e NSEC podem ser atualizados. Eis um exemplo:

```
update-policy { grant dhcp-key subdomain dhcp.cs.colorado.edu A } ;
```

Essa configuração permite que qualquer pessoa que conhece a chave `dhcp-key` atualize os registros de endereço no subdomínio dhcp.cs.colorado.edu. Essa instrução apareceria no arquivo **named.conf** do servidor mestre sob a instrução `zone` para dhcp.cs.colorado.edu. Haveria também uma instrução `key` para definir `dhcp-key`.

O trecho abaixo do arquivo **named.conf** no Departamento de Ciência da Computação da Universidade de Colorado utiliza a instrução `update-policy` para permitir que estudantes das aulas de administração de sistema atualizem seus próprios subdomínios, mas não desorganizem o resto do ambiente DNS.

```
// saclass.net
zone "saclass.net" in {
    type master;
    file "saclass/saclass.net";
    update-policy {
    grant feanor_mroe. subdomain saclass.net.;
    grant mojo_mroe. subdomain saclass.net.;
    grant dawdle_mroe. subdomain saclass.net.;
    grant pirate_mroe. subdomain saclass.net.;
    // and lots more
 };
...
```

15.15 QUESTÕES DE SEGURANÇA

Nos velhos tempos, a Internet era pequena, amigável e utilizada em grande parte por geeks. Hoje ela é um ambiente hostil, bem como uma parte crucial da infra-estrutura. Nesta seção, abrangemos vários tópicos relacionados com a segurança de uma maneira que pode parecer um pouco paranóica. Infelizmente, esses tópicos e as precauções são necessários na Internet de hoje em dia.

O DNS começou como um sistema inerentemente aberto, porém ele foi crescendo de forma mais segura — ou, pelo menos, em condições de tornar-se seguro. Como padrão, qualquer um na Internet pode investigar seu domínio com consultas individuais usando ferramentas como **dig**, **host** ou **nslookup**. Em alguns casos, elas podem fazer um dump completo de seu banco de dados DNS.

Para abordar essas vulnerabilidades, o BIND agora suporta vários tipos de controle de acesso baseado em endereços de host e de rede ou em autenticação criptografada. A Tabela 15.11 resume os recursos de segurança que são configurados em **named.conf**. A coluna Página mostra onde neste livro procurar informações adicionais.

Tabela 15.11 Recursos de segurança em named.conf

Recurso	Instruções	Página	O que especifica
`acl`	various	429	Listas de controle de acesso
`allow-query`	options, zone	428	Quem pode consultar uma zona ou servidor
`allow-recursion`	options	425	Quem pode fazer consultas recursivas
`allow-transfer`	options, zone	428	Quem pode solicitar transferências de zonas
`allow-update`	zone	433	Quem pode realizar atualizações dinâmicas
`blackhole`	options	428	Que servidores ignorar completamente
`bogus`	server	431	Que servidores jamais devem ser consultados
`update-policy`	zone	449	Que atualizações são permitidas

named é capaz de ser executado em um ambiente em que foi aplicado **chroot** sob um UID não privilegiado para minimizar riscos de segurança. Ele pode usar assinaturas de transação para controlar atualizações dinâmicas e, obviamente, também suporta todo o emaranhado do DNSSEC. Esses tópicos são vistos nas próximas seções.

Revendo as listas de controle de acesso

As ACLs são listas de correspondência de endereços identificadas que podem aparecer como argumentos em instruções como `allow-query`, `allow-transfer` e `blackhole`. A sintaxe básica das ACLs foi descrita na página 298. Elas podem ajudar a aprimorar a segurança do DNS de várias maneiras.

Cada site deveria ter ao menos uma ACL para endereços não genuínos e uma ACL para endereços locais.

Por exemplo:

```
acl bogusnets {            // ACL for bogus networks
    0.0.0.0/8 ;            // default, wild card addresses
    1.0.0.0/8 ;            // reserved addresses
    2.0.0.0/8 ;            // reserved addresses
    169.254.0.0/16 ;       // link-local delegated addresses
    192.0.2.0/24 ;         // sample addresses, like example.com
    224.0.0.0/3 ;          // multicast address space
    10.0.0.0/8 ;           // private address space (RFC1918)[19]
    172.16.0.0/12 ;        // private address space (RFC1918)
    192.168.0.0/16 ;       // private address space (RFC1918)
} ;

 acl cunets {              // ACL for University of Colorado networks
    128.138.0.0/16 ;       // main campus network
    198.11.16/24 ;
    204.228.69/24 ;
};
```

19. Não torne endereços privados falsos se você os utilizar e estiver configurando seus servidores de DNS internos!

Na seção `options` global de seu arquivo de configuração, você poderia então incluir:

```
allow-recursion { cunets; } ;
blackhole { bogusnets; } ;
```

Também é uma boa idéia restringir transferências de zonas para legitimar servidores escravos. Uma ACL facilita e organiza as coisas.

```
acl ourslaves {
    128.138.242.1 ;        // anchor
 ...
} ;
acl measurements {
    198.32.4.0/24 ;        // bill manning's measurements, v4 address
    2001:478:6:0::/48 ;    // bill manning's measurements, v6 address
} ;
```

A real restrição é implementada por uma linha como:

```
allow-transfer { ourslaves; measurements; } ;
```

As transferências são limitadas aos nossos próprios servidores escravos e às máquinas de dois projetos de medidas Internet que percorrem a árvore DNS ao contrário para determinar o tamanho da Internet e a porcentagem de servidores mal configurados. Limitar transferências dessa maneira torna impossível que outros sites façam o dump dos seus bancos de dados inteiros com uma ferramenta como **dig** (ver a página 330).

Por exemplo, o caida.org utiliza uma ACL para limitar transferências a apenas seus servidores secundários; utilizando **dig** *@server domain* o **axfr** é bem-sucedido na máquina em jungle.caida.org, mas falha em gypsy.cs.colorado.edu:

```
jungle$ dig @rommie.caida.org caida.org axfr

; <<>> DiG 8.3 <<>> @rommie.caida.org caida.org axfr
; (1 server found)
$ORIGIN caida.org.
@                   4H     IN      SOA     @ postmaster (
                                           200406300      ; serial
                                           1H             ; refresh
                                           30M            ; retry
                                           1W             ; expiry
                                           )
...

gypsy$ dig @rommie.caida.org caida.org axfr

; <<>> DiG 9.2.4 <<>> @rommie.caida.org caida.org axfr
;; global options:  printcmd
; Transfer failed.
```

Você ainda pode proteger sua rede em um nível mais baixo por meio de listas de controle de acesso de roteador e medidas de higienização de segurança normais em cada host. Se essas medidas não forem possíveis, você poderá recusar pacotes DNS, exceto para uma máquina de gateway que você monitora de perto.

Confinando named

Para confinar o dano que alguém poderia causar caso ele comprometesse seu servidor, você pode executar **named** em um ambiente onde foi aplicado **chroot** e/ou executá-lo como usuário não privilegiado, ou ambos. O flag **-t** especifica o diretório em que armazenar **chroot** e o flag **-u** para especificar o UID em que **named** deve executar. Por exemplo, o comando:

```
# named -u 53 -t /var/named
```

inicia **named** com UID 53 e um diretório-raiz de **/var/named**.

Se hackers comprometerem seu **named**, eles poderão ter acesso ao sistema sob o pretexto de serem o usuário que **named** executa. Se esse usuário for root e você não usar um ambiente no qual se aplicou **chroot**, tal brecha pode ser bastante

destrutiva. Muitos sites não se importam em utilizar os flags **-u** e **-t**, mas, quando uma nova vulnerabilidade é anunciada, esse sites deverão ser mais rápidos para atualizar do que os hackers são para atacar.

O diretório **chroot** não pode ser um diretório vazio já que tem de conter todos os arquivos que **named** normalmente requer para ser executado: **/dev/null**, os arquivos de zona, **named.conf**, arquivos alvo syslog e o soquete de domínio UNIX syslog, **/var** etc. O **named** do BIND 9 faz a chamada de sistema **chroot** depois que todas as bibliotecas foram carregadas, portanto ele não é mais necessário para copiar bibliotecas compartilhadas para o diretório **chroot**.

O Fedora instala **named** em um diretório **chroot** por padrão.

Comunicação segura entre servidores com TSIG e TKEY

Enquanto o DNSSEC (discutido na próxima seção) estava sendo desenvolvido, o IETF desenvolvia um mecanismo mais simples chamado TSIG (RFC2845) para possibilitar comunicação segura entre servidores por meio do emprego de "assinaturas de transação". O controle de acesso baseado em assinaturas de transação é mais seguro do que o controle de acesso baseado em endereços de origem IP. TSIG pode proteger as transferências de zona entre um servidor mestre e seus escravos e pode implementar atualizações dinâmicas seguras.

As assinaturas TSIG assinam as mensagens e as respostas entre servidores, não entre servidores e conversores. A assinatura de transação autentica o peer (par) e verifica se os dados não foram adulterados. As assinaturas são verificadas no momento em que um pacote é recebido e são depois descartadas; elas não são armazenadas em cache e não passam a fazer parte dos dados DNS.

TSIG utiliza um esquema de criptografia simétrica. Ou seja, a chave de criptografia é a mesma que a chave de decifrar. Essa chave única é chamada chave de segredo compartilhado. Embora a especificação TSIG permita múltiplos métodos de criptografia, o BIND implementa somente um, o algoritmo HMAC-MD5.

Você deve utilizar uma chave diferente para cada par de servidores que quer se comunicar de forma segura. O TSIG é muito menos caro computacionalmente do que a criptografia de chaves públicas, mas é apropriado somente para uma rede local na qual o número de pares de servidores de comunicação é pequeno. Ele não pode ser expandido para a Internet global.

O utilitário **dnssec-keyge**[20] do BIND gera uma chave para um par de servidores. Por exemplo, para gerar uma chave de segredo compartilhado do host para dois servidores, master e slave1, utilize:

```
# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST master-slave1
```

O flag **-b 128** instrui **dnssec-keygen** a criar uma chave de 128 bits. São gerados dois arquivos: **Kmaster-slave1.+157+09068.private** e **Kmaster-slave1.+157+09068.key**. 157 significa o algoritmo HMAC-MD5 e o 09068 é um número aleatório[21] utilizado como identificador de chaves caso você tenha várias chaves para o mesmo par de servidores. O arquivo **.private** se parece com este:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: jxopbeb+aPc71Mm2vc9R9g==
```

e o arquivo **.key** com este:

```
master-slave1. IN KEY 512 3 157 jxopbeb+aPc71Mm2vc9R9g==
```

Esses dois arquivos devem ter o modo 600 e devem ser possuídos pelo usuário **named**.

Observe o ponto que foi adicionado depois da string de argumento master-slave1 tanto nos nomes dos arquivos como no conteúdo do arquivo **.key**. O propósito dessa convenção é que, em outros contextos, os nomes das chaves devem ser nomes de domínio completamente qualificados e, portanto, devem terminar em um ponto.

Na verdade, você não precisa do arquivo **.key** — ele foi produzido porque o programa **dnssec-keygen** também gera pares de chave pública em que a chave pública (arquivo **.key**) é inserida no arquivo de zona DNS como um registro de recursos KEY. O 512 no registro KEY não é o comprimento da chave, mas um bit de flag que especifica que o registro é um registro de chave DNS.

Depois dessa complicação toda, talvez você esteja decepcionado de saber que a chave gerada é de fato apenas um longo número aleatório. Você poderia gerar a chave manualmente escrevendo uma string ASCII do comprimento exato (divisível por 4) e fazendo com que essa se passasse por uma codificação de base 64 de alguma coisa ou utilizando **mmencode** para codificar uma string aleatória. A maneira pela qual você cria a chave não é importante; ela apenas tem de existir em ambas as máquinas.

---

20. Esse comando chama-se **dnskeygen** no BIND 8.
21. Não é realmente aleatório, nem mesmo pseudoaleatório; é um hash do registro de recursos DNSKEY.

scp é parte do conjunto SSH. Consulte a página 481 para detalhes.

Copie a chave tanto para master como para slave1 usando **scp** ou então recortando-a e colando-a. *Não* utilize **telnet** ou **ftp** para copiar a chave; até redes internas podem ser inseguras. A chave deve ser incluída nos arquivos **named.conf** de ambas as máquinas. Como **named.conf** é, em geral, mundialmente legível e as chaves não devem ser, coloque a chave em um arquivo separado que esteja incluído em **named.conf**. Por exemplo, você poderia colocar o trecho:

```
key master-slave1. {
    algorithm hmac-md5 ;
    secret "segredo-compartilhado-que-você-gerou" ;
} ;
```

no arquivo **master-slave1.tsig**. O arquivo deve ter o modo 600 e seu proprietário deve ser o UID de **named**. No arquivo **named.conf**, adicione a seguinte linha:

```
include "master-slave1.tsig" ;
```

próximo do início do arquivo.

Essa parte da configuração simplesmente define as chaves. Para que eles realmente sejam utilizados para assinar e verificar atualizações, o mestre precisa exigir a chave para transferências e o escravo precisa identificar o mestre com uma instrução `server` e uma cláusula `keys`. Você poderia, por exemplo, adicionar a linha:

```
allow-transfer { key master-slave1. ;} ;
```

à instrução `zone` do servidor mestre, e a linha:

```
server endereço-IP-do-mestre { keys { master-slave1. ; } ; } ;
```

ao arquivo **named.conf** do escravo. Se o servidor mestre permitir atualizações dinâmicas, ele também poderá utilizar a chave na sua cláusula `allow-update` na instrução `zone`.

Já utilizamos os nomes genéricos master e slave1 para identificar os servidores e a chave. Se utilizar chaves TSIG para muitas zonas, você também deverá incluir a zona no seu esquema de atribuição de nomes a fim de ajudá-lo a manter tudo correto.

Para testar sua configuração TSIG, execute **named-checkconf** para verificar se você tem a sintaxe correta. Então utilize **dig** para tentar uma transferência de zona (**dig** *@master* **axfr**) a partir tanto de slave1 como de alguma outra máquina. A primeira deve ser bem-sucedida e a segunda deve falhar com o diagnóstico "Transfer failed". Para ter total certeza de que tudo está correto, remova a cláusula `allow-transfer` e tente os comandos **dig** mais uma vez. Dessa vez, as duas devem ser bem-sucedidas. (Não se esqueça de colocar `allow-transfer` de volta!) Como um teste final, aumente o número serial para a zona no servidor mestre e observe o arquivo de log no escravo para ver se ele seleciona a alteração e transfere a zona.

Ao começar a utilizar assinaturas de transação, execute **named** no nível de depuração 1 (consulte a página 324para informações sobre o modo de depuração) por algum tempo para ver que mensagens de erro são geradas. Versões mais antigas de BIND não compreendem mensagens assinadas e reclamam delas, algumas vezes chegando ao ponto de se recusarem a carregar a zona.

TKEY é um mecanismo do BIND 9 que permite que dois hosts gerem uma chave de segredo compartilhado automaticamente sem chamadas telefônicas ou cópias seguras para distribuir a chave. Ele utiliza um algoritmo chamado intercâmbio de chaves Diffie-Hellman no qual cada lado forma um número aleatório, aplica o mesmo algoritmo sobre ele e envia o resultado para o outro lado. A seguir, cada lado combina matematicamente seu próprio número com a transmissão que ele recebeu para chegar à mesma chave. Um espião talvez ouça a transmissão sem ser percebido, mas não conseguirá inverter a matemática.[22] Infelizmente, o código para tornar TKEY realmente útil e evitar armazenar a chave TSIG no arquivo de configuração nunca foi implementado; esperamos que ele esteja na lista "to-do".

Consulte a página 619 para informações adicionais sobre NTP.

Ao utilizar chaves TSIG e assinaturas de transação entre os servidores mestre e escravos, você deve manter os relógios dos servidores sincronizados com NTP. Se os horários nos relógios forem muito diferentes (mais de 5 minutos no BIND 9), a verificação de assinatura não funcionará. Esse problema pode ser muito difícil de depurar.

Atualmente não é possível usar chaves TSIG para manter seguras comunicações entre servidores BIND e Microsoft. A Microsoft usa TSIG de maneira não-padrão (GSS-TSIG) e não compartilhará os detalhes do seu esquema fora do contexto de um acordo de não-revelação de segredos industriais. As negociações para resolver esse conflito estão em andamento.

---

22. Os cálculos envolvidos são chamados de problema de logaritmo distinto e depende do fato de que, para aritmética modular, calcular potências é fácil, mas calcular logaritmos para desfazer as potências é quase impossível.

O SIG(0) é um outro mecanismo para assinar transações entre servidores ou entre atualizadores dinâmicos e o servidor mestre. Ele utiliza criptografia de chave pública; consulte as RFCs 2535 e 2931 para detalhes.

DNSSEC

DNSSEC é um conjunto de extensões DNS que autenticam a origem dos dados de zonas e verificam sua integridade usando criptografia de chaves públicas. Isto é, as extensões permitem que clientes DNS façam as seguintes perguntas: "Esses dados DNS realmente provêm do proprietário da zona?" e "Esses são realmente os dados enviados por esse proprietário?".

O DNSSEC é baseado em uma cadeia em cascata de confiabilidade: os servidores-raiz fornecem informações de validação para os domínios de primeiro nível, estes fornecem informações de validação para os domínios de segundo nível e assim por diante. A opção de configuração `trusted-keys` do BIND permite inicializar o processo e manter seguras as partes da árvore de DNS antes de os domínios-raiz e de primeiro nível serem protegidos.

Sistemas de criptografia de chaves públicas utilizam duas chaves: uma para criptografar (assinar) e uma outra diferente para decifrar (verificar). Os publicadores assinam seus dados com uma chave secreta "privada". Qualquer um pode verificar a validade de uma assinatura com uma chave "pública" correspondente que é amplamente distribuída. Se uma chave pública decifra corretamente um arquivo de zonas, então a zona só pode ter sido criptografada com a chave privada correspondente. O truque é ter certeza de que as chaves públicas que você utiliza para verificação são autênticas. Os sistemas de chaves públicas permitem que uma entidade assine a chave pública de uma outra atestando assim a legitimidade da chave; daí o termo "cadeia de confiabilidade".

Os dados em uma zona DNS são muito volumosos para serem cifrados com criptografia de chaves públicas — a criptografia seria muito lenta. Em vez disso, já que os dados não são secretos, um hash (por exemplo, uma soma de verificação MD5) é executado nos dados e os resultados do hash são assinados (cifrados/encriptados) pela chave privada da zona. Os resultados do hash são como uma impressão digital dos dados e são chamados assinatura digital. As assinaturas são acrescentadas aos dados que elas autenticam como registros RRSIG no arquivo de zona assinada.

Para verificar a assinatura, você a decifra com a chave pública do assinante, passa os dados pelo mesmo algoritmo hash seguro e compara o valor hash calculado com o valor hash decifrado. Se eles forem coincidentes, você autenticou o assinante e verificou a integridade dos dados.

No sistema DNSSEC, cada zona tem suas próprias chaves pública e privada. A chave privada assina cada RRset (ou seja, cada conjunto de registros do mesmo tipo para o mesmo host). A chave pública verifica as assinaturas e é incluída nos dados da zona na forma de um registro de recursos DNSKEY.

Zonas pai assinam as chaves públicas de suas zonas filhas. **named** verifica a autenticidade do registro DNSKEY de uma zona filha comparando-a com a assinatura da zona pai. Para verificar a autenticidade da chave da zona pai, **named** é capaz de verificar o pai do pai e assim por diante até chegar à raiz. A chave pública para a zona-raiz seria incluída no arquivo-raiz hints.

Antes de passarmos para os mecanismos de geração de chaves e de assinatura de zonas, precisamos ser honestos em relação ao estado atual do DNSSEC e seu impacto sobre administradores de sistemas. Administradores de sistemas podem precisar lidar com DNSSEC dentro de um a dois anos, mas isso certamente não se encontra na lista de obrigações desta semana. Muitos aplicativos estão clamando por uma infra-estrutura de chaves públicas ("PKI"), e o DNS é um candidato em potencial para fornecê-las. Entretanto, temos um pouco daquele problema do ovo e da galinha. Precisamos ter certeza de que o DNS é seguro antes de podermos confiar a ele nossas chaves para outras transações na Internet. Porém, precisamos de uma infra-estrutura de chaves públicas para assegurar o DNS.

Alterações recentes nas especificações DNSSEC o tornaram muito mais perto de ser implantável do que era no passado. A especificação DNSSEC original mantinha cópias das chaves assinadas para as zonas filhas tanto nos domínios pai como nos domínios filhos, portanto uma comunicação ininterrupta era necessária se o filho ou o pai quisesse alterar sua chave. Como há múltiplas cópias de cada chave, não ficava claro em qual cópia você deveria acreditar quando as duas estavam fora de sincronia.

A RFC4034 introduziu um novo tipo de registro de recursos (DS, o assinante de delegação) e separou as chaves entre chave de assinatura de chaves e chaves de assinatura de zonas e alterou os nomes de alguns registros de recursos:

- KEY tornou-se DNSKEY, para chaves criptográficas
- SIG tornou-se RRSIG, a assinatura para um conjunto de registros de recursos
- NXT tornou-se NSEC para identificar a próxima entrada segura na delegação

Os registros KEY e SIG ainda são utilizados com a segurança SIG(0) e TSIG. DNSKEY, RRSIG e NSEC são utilizados com DNSSEC e NXT está obsoleto. As alterações de SIG para RRSIG e de NXT para NSEC são menos importantes; os nomes mudaram para não confundir software antigo com o novo esquema.

As chaves incluídas em um registro de recursos DNSKEY podem ser chaves de assinatura de chave (*key-signing keys* – KSKs) ou chaves de assinatura de zonas (*zone-signing keys* – ZSKs). Um novo flag, chamado SEP (*secure entry point*) para

"ponto de entrada segura", faz uma distinção entre elas. O bit 15 do campo de flags é configurado como 1 para KSKs e como 0 para ZSKs. Essa convenção torna ímpares as KSKs e as ZSKs mesmo quando tratadas como números decimais.

O registro DS aparece apenas na zona pai e indica que uma subzona é segura (assinada). Ele também identifica a chave utilizada pelo filho para que ele auto-assine seu próprio RRset de chave. O registro DS inclui um identificador de chave (um número de 5 dígitos), algoritmo criptográfico, tipo de resenha e uma resenha do registro da chave pública permitida (ou utilizada) para assinar o registro do recurso da chave do filho.

Alterar as chaves existentes nas zonas pai e nas zonas filhas era uma questão complicada que parecia destinada a exigir cooperação e comunicação entre o pai e o filho. A criação do registro DS, o uso de chaves separadas para assinatura de chaves e para assinatura de zonas e o uso de múltiplos pares de chaves ajudaram a resolver esse problema.

Múltiplas chaves podem ser geradas e assinadas de modo que seja possível uma transição suave entre uma chave e a seguinte. O filho pode alterar suas chaves de assinatura de zona sem notificar o pai; ele só precisa coordenar com o pai se ele alterar sua chave de assinatura de chave. À medida que as chaves são distribuídas, há um período durante o qual tanto a chave antiga como a nova chave são válidas; depois que os valores armazenados em cache na Internet expiram, a chave antiga pode ser aposentada.

As versões atuais do BIND removeram o OpenSSL da distribuição, assim, se você quiser utilizar o DNSSEC, terá de obter um pacote que inclua suporte DNSSEC ou obter as bibliotecas SSL em www.openssl.org e então recompilar o BIND com o suporte criptográfico ativado (utilize a opção **--with-openssl** para **./configure**). Se não fizer isso, **dnssec-keygen** irá reclamar. Entretanto, isso ainda funcionará para gerar chaves TSIG, visto que elas não requerem o OpenSSL.

Você realiza vários passos para criar e utilizar zonas assinadas. Primeiro, você gera um ou mais pares de chaves para a zona. Por exemplo,

```
# dnssec-keygen -a DSA -b 768 -n ZONE mydomain.com
```

gera um par de chaves de 768 bits utilizando o algoritmo DSA a ser empregado para assinar uma zona chamada mydomain.com. Há vários algoritmos de criptografia disponíveis com uma série de comprimentos de chave: RSAMD5 (512–4096), RSASHA1 (512–4096), DH (128–4096) e DSA (512–1024 e divisível por 64). Você pode incluir o flag **-f KSK** (não utilizado aqui) para identificar a chave gerada como uma chave de assinatura de chave configurando seu bit SEP.

dnssec-keygen imprime o padrão a seguir:

```
Kmydomain.com.+003+50302
```

onde Kmydomain.com é o nome da chave, 003 é o identificador do algoritmo DSA e 50302 é o identificador da chave. **dnssec-keygen** cria arquivos contendo as chaves públicas e privadas:

```
Kmydomain.com.+003+50302.key      # Public key
Kmydomain.com.+003+50302.private # Private key
```

A chave privada é utilizada para assinar os registros de dados de uma zona e a chave pública é utilizada para verificar assinaturas. A chave pública é colocada tipicamente no arquivo de zonas logo após o registro SOA.

Idealmente, a parte da chave privada de qualquer par de chaves seria mantida off-line ou pelo menos em uma máquina que não está na Internet. Essa precaução é impossível para zonas atualizadas dinamicamente e complicada para chaves de assinatura de zonas, mas é perfeitamente razoável para chaves de assinatura de chave, que presumivelmente têm um longo tempo de vida.

É difícil compreender esses arquivos gerados por **dnssec-keygen** sem inspecionar seu conteúdo, portanto vamos examiná-lo rapidamente:

```
Kmydomain.com.+003+50302.key
mydomain.com. IN DNSKEY 256 3 3
    BMORyx8sRz6EJ6ETfRjOPh4uraB1tLZTYI1WU6D7O7/GiBXwxAsvpgH6
    sNXE3uwZVaQFxvDHfa6amy3JSSilcRNfiiOs3LfoyZzUWOceVo6zRBoO
    3GTYpZ6efrFUackXKr9WsadC+4W+2fGx4yL8N6B32akBTiIMLpOIFOJe
    xqLe6QrJVE21eXzRqC5BTC25R6TPMoOH6cuue5wBeNphcsOsGRfOf4hy
    lOwkb6T7etH//EQgfkLWqcwolVF9hjzskX64eOQeeENXRV8sFvTMVzTk
    qA4KJsBCclVzrDSLAsLZtYH4g6VvrMZHuQ5C/ArCIsdnOROOmpH6ZUIl
    WaSIElpAxaZ7ynD4hTlRB5br2KiyGTr27dHi7QS4vOW7oDDPI9+lwAcK
    g2A3LHpmglS59utmpxJa

Kmydomain.com.+003+50302.private

Private-key-format: v1.2
Algorithm: 3 (DSA)
```

```
Prime(p): tLZTYI1WU6D7O7/GiBXwxAsvpgH6sNXE3uwZVaQFxvDHfa6amy3JSSilc
    RNfiiOs3LfoyZzUWOceVo6zRBoO3GTYpZ6efrFUackXKr9WsadC+4W+2fGx4yL
    8N6B32akB
Subprime(q): w5HLHyxHPoQnoRN9GPQ+Hi6toHU=
Base(g): TiIMLpO1FOJexqLe6QrJVE21eXzRqC58TC25R6TPMoOH6cuue5w8eNphcs
    OsGRfOf4hylOwkb6T7etH//EQgfkLWqcwolVF9hjzskX64eOQeeENXRV8sFvTM
    VzTkqA4K
Private_value(x): GqcOz8K56CmUxgo6ERuyEWMLVME=
Public_value(y): JsBCclVzrDSLAsLZtYH4g6VvrMZHuQ5C/ArCIsdnORO0mpH6ZUIl
    WaSIElpAxaZ7ynD4hT1RB5br2KiyGTr27dHi7QS4vOW7oDDPI9+lwAcKg2A3L
    Hpmg1S59utmpxJa
```

Para facilitar a troca de chave, você deve criar vários pares de chaves KSK e fazer com que todas sejam assinadas pela sua zona pai. Depois que você e sua zona pai têm um relacionamento de confiança estabelecido (uma KSK assinada pelo pai e incluída no registro DS que delega sua zona segura), você pode utilizar esse relacionamento para inicializar o processo de troca da chave. Apenas envie essa chave ao pai para estabelecer quem você é juntamente com as KSKs adicionais que você gostaria que fossem assinadas e apontadas pelos registros DS. Depois que as novas chaves foram assinadas, você pode aposentar a original.

Uma vez que o pai assinou sua KSK e a inseriu no seu registro DS, você está pronto para assinar os dados reais da sua zona. A operação de assinatura recebe um arquivo de dados de zona normal como a entrada e adiciona os registros RRSIG e NSEC logo depois de cada conjunto de registros de recursos. Os registros RRSIG são as assinaturas reais e os registros NSEC suportam assinatura de respostas negativas.

Para assinar uma zona, utilize o comando **dnssec-signzone**. Por exemplo, o comando:

```
# dnssec-signzone -o mydomain.com db.mydomain Kmydomain.com+003+50302
```

lê o arquivo de zona **db.mydomain** e produz uma versão assinada desse arquivo de zona chamada **db.mydomain.signed**. Também cria dois arquivos de keyset:

```
keyset-mydomain.com.      # Keyset
dsset-mydomain.com.       # Secure delegation keyset
```

Esses arquivos de conjuntos de chaves são utilizados para assinar zonas e chaves, respectivamente. Eles contêm:

```
keyset-mydomain.com.

$ORIGIN .
mydomain.com      172800  IN      DNSKEY  256 3 3 (
                                  BMORyxBsRz6EJ6ETfRjOPh4uraB1tLZTYI1W
                                  U6D7O7/GiBXwxAsvpgH6sNXE3uwZVaQFxvDH
                                  fa6amy3JSSilcRNfiiOs3LfoyZzUWOceVo6z
                                  RBoO3GTYpZ6efrFUackXKr9WsadC+4W+2fGx
                                  4yL8N6B32akBTiIMLpO1FOJexqLe6QrJVE21
                                  eXzRqC58TC25R6TPMoOH6cuue5w8eNphcsOs
                                  GRfOf4hylOwkb6T7etH//EQgfkLWqcwolVF9
                                  hjzskX64eOQeeENXRV8sFvTMVzTkqA4KJsBC
                                  clVzrDSLAsLZtYH4g6VvrMZHuQ5C/ArCIsdn
                                  ORO0mpH6ZUIlWaSIElpAxaZ7ynD4hT1RB5br
                                  2KiyGTr27dHi7QS4vOW7oDDPI9+lwAcKg2A3
                                  LHpmg1S59utmpxJa
 ) ; key id = 50302

dsset-mydomain.com.

mydomain.com. IN DS 50302 3 1 1B44471AFD5B4F4463BB3A0D7B66B6ABC018DA96
```

Quaisquer arquivos de conjuntos de chaves para zonas filhas no diretório atual são também assinados e incorporados aos registros DS, que delegam as subzonas seguras aos servidores das zonas filhas e as autenticam com as chaves assinadas. Pode levar um bom tempo para assinar uma zona.

Se uma zona assinada for passada como o argumento para **dnssec-signzone**, as assinaturas de quaisquer registros que estão em vias de expirar são renovadas. "Em vias de expirar" é definido como sendo três quartos do período de validade. Por padrão, as assinaturas são válidas por 30 dias, mas um período diferente pode ser especificado na linha de comando **dnssec-signzone**. Novas assinaturas em geral resultam em alterações, mas não mudam (ainda) automaticamente o número

serial, portanto os administradores de sistemas precisarão atualizar o número serial manualmente. Os servidores escravos da zona terão de realizar uma operação de transferência de zona para se sincronizarem novamente.

O exemplo a seguir mostra os dados da zona antes e depois da assinatura.[23] Eis o antes:

```
$TTL 172800 ; 2 days
@                   IN      SOA     ns.mydomain.com. hostmaster.mydomain.com. (
                                            2006081200      ; serial
                                            7200            ; refresh (2 hours)
                                            3600            ; retry (1 hour)
                                            1728000         ; expire (2 weeks 6 days)
                                            172800          ; minimum (2 days)
                                            )

                    IN      NS      ns.cs.colorado.edu.

mydomain.com.       IN      DNSKEY  256 3 3
    BMORyx8sRz6EJ6ETfRjOPh4uraB1tLZTYI1WU6D7O7/GiBXwxAsvpgH6
    sNXE3uwZVaQFxvDHfa6amy3JSSi1cRNfiiOs3LfoyZzUWOceVo6zRBoO
    3GTYpZ6efrFUackXKr9WsadC+4W+2fGx4yL8N6B32akBTiIMLpO1FOJe
    1Owkb6T7etH//EQgfkLWqcwolVF9hjzskX64eOQeeENXRVBsFvTMVzTk
    qA4KJsBCclVzrDSLAsLZtYH4g6VvrMZHuQ5C/ArCIsdnORO0mpH6ZUI1
    WaSIE1pAxaZ7ynD4hT1RB5br2KiyGTr27dHi7QS4vOW7oDDPI9+1wAcK
    g2A3LHpmg1S59utmpxJa

anchor              IN      A       128.138.242.1
                    IN      A       128.138.243.140
                    IN      MX      10 anchor
                    IN      MX      99 @
awesome             IN      A       128.138.236.20
...
```

E o depois:

```
; File written on Thu Nov 11 17:41:25 2006
; dnssec_signzone version 9.3.0
mydomain.com. 172800 IN SOA ns.mydomain.com. hostmaster.mydomain.com. (
                                2006081200      ; serial
                                7200            ; refresh (2 hours)
                                3600            ; retry (1 hour)
                                1728000         ; expire (2 weeks 6 days)
                                172800          ; minimum (2 days)
                                 )
      172800  RRSIG        SOA 3 2 172800 20061211164125 (
                             20061111164125 50302 mydomain.com.
                             BElneYxZ3g9JnKbXdnmPhKVWfd13JTU8aj00
                             5dQta2WeBAatNuWt8dQ= )
      172800  NS           ns.cs.colorado.edu.
      172800  RRSIG       NS 3 2 172800 20061211164125 (
                             20061111164125 50302 mydomain.com.
                             BLG6LRrXtRHRdRFtTOmlQsadOIefqHAq5Rid
                             PHZ74vOl/UkEW6wY6VA= )
      172800  NSEC        anchor.mydomain.com. NS SOA RRSIG NSEC DNSKEY
      172800  RRSIG       NSEC 3 2 172800 20061211164125 (
                             20061111164125 50302 mydomain.com.
                             BCz31GPChdQrmNrZypv4xxmXDCThZOI1kEGL
                             TSkf7Q+TmCDmAADxmBE= )
      172800  DNSKEY      256 3 3 (
                             BMORyx8sRz6EJ6ETfRjOPh4uraB1tLZTYI1W
                             U6D7O7/GiBXwxAsvpgH6sNXE3uwZVaQFxvDH
```

23. Tentamos simplificar esse exemplo utilizando apenas dois hosts e seus registros A e MX, mas dnssec-signzone requer uma zona "real" e assim os registros SOA, NS e DNSKEY tiveram de ser adicionados para um exemplo mínimo. Desculpe-nos pela confusão.

```
                                fa6amy3JSSilcRNfiiOs3LfoyZzUWOceVo6z
                                RBoO3GTYpZ6efrFUackXKr9WsadC+4W+2fGx
                                4yL8N6B32akBTiIMLpO1FOJexqLe6OrJVE21
                                eXzRqC58TC25R6TPMoOH6cuue5w8eNphcsOs
                                GRfOf4hylOwkb6T7etH//EOgfkLWqcwolVF9
                                hjzskX64eOOeeENXRV8sFvTMVzTkqA4KJsBC
                                clVzrDSLAsLZtYH4g6VvrMZHuO5C/ArCIsdn
                                OROOmpH6ZUIlWaSIE1pAxaZ7ynD4hT1RB5br
                                2KiyGTr27dHi7QS4vOW7oDDPI9+lwAcKg2A3
                                LHpmg1S59utmpxJa
                                ) ; key id = 50302
     172800  RRSIG       DNSKEY 3 2 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BAgZDfk/YCOhVfuoyG5pgfyFCmsGqg4W7uuM
                                Rm5eNP9BnOEbBnuT6XO= )
anchor.mydomain.com. 172800 IN A 128.138.242.1
     172800  IN A        128.138.243.140
     172800  RRSIG       A 3 3 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BIRtKWOUm7ItfbPqRew+jKol52WJh+4nHkmK
                                1ePNxjsQWcgaKm5jiMU= )
     172800   MX         10 anchor.mydomain.com.
     172800   MX         99 mydomain.com.
     172800  RRSIG       MX 3 3 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BGtmN2u3OylpMDzstWGgWZfXB3lDlmy5W6DP
                                t/8D3lOpyYN8jJPb8J4= )
     172800  NSEC        awesome.mydomain.com. A MX RRSIG NSEC
     172800  RRSIG       NSEC 3 3 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BArN6oES72gzFgQmBHL3NzlquMbDbLfpvj7J
                                3CSb/c8U/bciWGXsV3Q= )
awesome.mydomain.com. 172800 IN A 128.138.236.20
     172800  RRSIG       A 3 3 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BJ/qWBgLgS/2N5CoXGnI4vs91SsyIBKKfoq9
                                R+VsMpRmnVrSi1DUln8= )
     172800  NSEC        mydomain.com. A RRSIG NSEC
     172800  RRSIG       NSEC 3 3 172800 20061211164125 (
                                20061111164125 50302 mydomain.com.
                                BKoByqF5wUceb2vc8H2uealgKrejH4VZOS5m
                                O4KukWCUo2IAFX+msO4= )
```

Como você pode ver, as zonas assinadas são feias. Normalmente são quatro vezes maiores do que a zona original e aquele belo aspecto de ordenação lógica é perdido. Para todos os propósitos práticos, um arquivo de zona assinado não é mais legível por humanos e não pode ser editado manualmente devido aos registros RRSIG e NSEC. Nenhuma parte interna útil para o usuário!

Um registro RRSIG contém uma grande quantidade de informações:

- O tipo de conjunto de registros que está sendo assinado
- O algoritmo de assinatura que está sendo utilizado (em nosso caso, ele é 3, o algoritmo DSA)
- O TTL do conjunto de registros que foi assinado
- O momento em que a assinatura expira (no formato *aaaammddhhssss*)
- O momento em que o conjunto de registros foi assinado (também no formato *aaaammddhhssss*)
- O identificador de chaves (em nosso caso, 50302)
- O nome do assinante (mydomain.com.)
- E, finalmente, a assinatura digital em si

Para utilizar a zona assinada, altere o parâmetro `file` na instrução `zone` de **named.conf** para mydomain.com a fim de apontar para **db.mydomain.signed** em vez de **db.mydomain**. Ufa! Isso é tudo.

Respostas negativas

As assinaturas digitais são ótimas para respostas positivas como "Eis o endereço IP para o host anchor.cs.colorado.edu, juntamente com uma assinatura para provar que ele realmente veio de cs.colorado.edu e que os dados são válidos". Mas o que dizer das respostas negativas como "Não encontrado nenhum host com essas especificações?" ["No such host?"]. Tais respostas negativas normalmente não retornam nenhum registro assinável.

No DNSSEC, esse problema é tratado por registros NSEC que listam o próximo registro na zona em uma ordem canônica.[24] Se o próximo registro após anchor em cs.colorado.edu fosse awesome.cs.colorado.edu e uma consulta por anthill.cs.colorado.edu chegasse, a resposta seria um registro NSEC assinado como:

```
anchor.cs.colorado.edu.   IN     NSEC   awesome.cs.colorado.edu A MX NSEC
```

Esse registro diz que o nome imediatamente após anchor na zona cs.colorado.edu é awesome.cs.colorado.edu e que anchor possui pelo menos um registro A, um registro MX e um registro NSEC. Registros NSEC também são retornados caso o host exista, mas o tipo de registro consultado não existe. Por exemplo, se a consulta fosse para buscar um registro LOC para anchor, o mesmo registro NSEC de anchor seria retornado e mostraria somente registros A, MX e NSEC.

O último registro NXT em uma zona retorna para o primeiro nome da zona. Por exemplo, o registro NSEC para zamboni.cs.colorado.edu apontaria de volta para o primeiro registro, aquele do próprio cs.colorado.edu:

```
zamboni.cs.colorado.edu.  IN     NSEC   cs.colorado.edu A MX NSEC
```

Os registros NSEC fornecem uma maneira para que um consultador persistente enumere o conteúdo de uma zona um registro por vez. Isso não é bom para segurança nem para a privacidade.

Microsoft e DNS

O Windows utiliza os registros de recurso SRV para descobrir praticamente tudo: servidores de nomes, impressoras, sistemas de arquivos e assim por diante. A Microsoft seguiu as especificações IETF na sua implementação dos registros SRV, mas a maneira como os registros são inseridos no DNS por meio de uma atualização dinâmica segura é não-padrão. A Microsoft utiliza uma variação das assinaturas de transação chamada GSS-TSIG que utiliza um segredo compartilhado obtido do Kerberos KDC (Key Distribution Center). No momento, a implementação da Microsoft utiliza o campo das extensões dos fornecedores e, portanto, não é compatível com a versão de código-fonte aberto do Kerberos 5.

Se quiser executar Win2K e usar registros SRV, você terá de parar de usar o seu domínio Kerberos existente e executar um servidor Kerberos Win2K em suas redes. Para alguns sites com uma infra-estrutura Kerberos rica, esse problema é uma interrupção temporária dos serviços. A Microsoft parece estar utilizando protocolos abertos o suficiente apenas para surrupiar (atendendo furtivamente) listas de verificação de compras de empresas, mas não o suficiente para permitir que qualquer outra pessoa interopere e venda em seu mercado. Espera-se que a Microsoft documente suas extensões sem o acordo de não-revelação exigido atualmente para que o ISC possa fazer com que seu TSIG e o GSS-TSIG da Microsoft interoperem.

Uma semana depois de o Windows 2000 ter sido lançado, a carga de consultas nos servidores-raiz DNS aumentou significativamente. Um pouco de investigação revelou que computadores com Windows 2000 estavam tentando atualizar as zonas-raiz ou de primeiro nível dinamicamente. O número de consultas UDP ao servidor-raiz A atingiu mais do que o dobro como conseqüência disso. Para piorar as coisas ainda mais, quando suas requisições de atualização eram recusadas, os computadores com Win2K solicitavam um registro KEY para tentar uma atualização autenticada. Isso também falhou, de modo que eles tentaram uma última vez abrindo uma conexão TCP para fazer uma atualização dinâmica autenticada. Um servidor-raiz não tem tempo para atender as zilhões de requisições de conexões TCP resultantes.

Bugs que afetam a infra-estrutura da Internet são tanto sérios como potencialmente embaraçosos, assim supomos que a Microsoft já tenha corrigido esse problema. Pensávamos em converter esta seção em uma nota de rodapé histórica, mas, apenas para sermos completos, revisamos o tráfego de rede de um sistema Windows XP recém-instalado executando o Service Pack 2. Surpreendentemente, descobrimos que ele continua mais intransigente do que nunca.

O problema aparece mais comumente (mas não exclusivamente) nas zonas reversas para os espaços de endereços IP privados definidos na RFC1918. Informações sobre esses endereços nunca deveriam sair do ambiente local, mas as máquinas Windows tentam atualizar seus domínios de inclusão e, se isso falhar, elas prosseguem para a árvore de atribuição de nomes do DNS até que finalmente alcançam os servidores-raiz.

Por exemplo, se o endereço 192.168.1.10 fosse atribuído por um servidor DHCP, uma máquina Windows tentaria atualizar dinamicamente o domínio 1.168.192.in-addr.arpa. Se isso falhar, ela tentará atualizar 168.192.in-addr.arpa, 192.in-addr.arpa, in-addr.arpa e por fim arpa.

---

24. A ordem de certo modo é alfabética, mas com nomes que se encontram mais acima na árvore DNS vindo em primeiro lugar. Por exemplo, na zona cs.colorado.edu, cs.colorado.edu vem antes de qualquer host cs.colorado.edu. Em um nível hierárquico, a ordem é alfabética.

Os sistemas Windows que não redefiniram seus padrões de DNS são partes de um contínuo, maciço, lento, não intencional e distribuído ataque de negação de serviço contra os servidores de nomes-raiz. Para impedir que os sistemas Windows 2000 ou sistemas XP tentem atualizar as zonas-raiz e fazer com que eles executem de uma maneira mais elegante com seus servidores de nomes UNIX ou Linux, tente este procedimento:

- Clique com o botão direito do mouse em Meus Locais de Rede e selecione Propriedades, que exibe uma janela chamada Conexões de Rede
- Dê um clique com o botão direito do mouse em cada conexão alternadamente e selecione Propriedades
- Clique sobre Internet Protocol (TCP/IP), depois clique sobre o botão Propriedades
- Clique no botão Configurações Avançadas... na parte inferior da página de propriedades
- Clique sobre a guia DNS na parte superior
- Na parte inferior da página, desmarque a caixa "Registre o endereço desta conexão em DNS"
- Vá clicando em OK até chegar ao fim

Para aumentar o tempo disponível, os operadores das zonas-raiz delegavam as zonas reversas do espaço de endereçamento na RFC1 918 aos servidores chamados prisoner.iana.org, blackhole1.iana.org e blackhole2.iana.org.. Esses servidores são autorizados (prisoner o mestre, blackholes os escravos) para essas zonas e interceptam as atualizações dinâmicas geradas pelos sistemas Windows que querem dizer ao mundo que o nome "mypc" deve ser associado com o endereço 10.0.0.1. Esses servidores iana.org são anycast e têm uma instância na maioria dos grandes ISPs, portanto as mensagens de atualização de DNS fictícias que escapam do site local não têm de poluir a Internet até os servidores-raiz.

Mas espere, isso não é o fim do mau comportamento do DNS da Microsoft. Se os servidores DNS preferidos dessa empresa não responderem a uma consulta dentro de um segundo, os sistemas Windows rapidamente se expandem para emitir múltiplas consultas simultâneas a cada servidor DNS que eles conhecem.[25] Em outras palavras, no primeiro sinal de sobrecarga do servidor ou congestionamento da rede, o Windows tenta gerar o maior número de tráfego de rede e de servidor possível. Você pode desligar esse comportamento desativando o serviço DNS Client, mas esteja ciente de que fazer isso desativa também o cache DNS local de cada máquina. Eis como desativar o serviço DNS Client:

- Selecione Configurações->Painel de controle, no menu Iniciar
- No Painel de controle, dê um clique duplo em Ferramentas Administrativas
- Na página Ferramentas Administrativas, dê um clique duplo em Gerenciamento de Computador
- Do lado esquerdo, clique no sinal "+" próximo a Serviços e Aplicativos para expandi-lo
- Selecione Serviços no lado esquerdo e dê um clique duplo em Cliente DNS no lado direito
- Clique no menu suspenso ao lado de "Tipo de inicialização" e selecione Manual
- Clique no botão Stop a fim de parar a cópia já em execução do serviço
- Vá clicando em OK até chegar ao fim

Se não estiver utilizando o serviço Active Directory da Microsoft, provavelmente será melhor utilizar os servidores UNIX/Linux para DNS e evitar a implementação da Microsoft. Infelizmente, o Active Directory complica as coisas, mas você pode suportar Active Directory com o BIND.

Uma questão final: os servidores Microsoft geram um tumulto quando consultam uma zona cujos servidores estão fora de combate. Se seu site faz delegações a servidores que parecem estar inativos, é uma boa prática pelo menos instalar um arquivo de zonas vazio que substitua servidores ausentes. Essa medida impedirá que servidores Microsoft que consultam o servidor defeituoso acabem tornando-se confusos e "surrando" os servidores-raiz e os servidores gTLD.

15.16 TESTANDO E DEPURANDO

named fornece várias ferramentas de depuração já embutidas, destacando-se entre elas o sistema de registro (log) bastante configurável. Você pode especificar níveis de depuração na linha de comando ou configurá-los com **rndc**. Você também pode instruir **named** a fazer um dump das estatísticas operacionais em um arquivo. Você pode verificar pesquisas de nomes com ferramentas externas como **dig**.

25. Esse comportamento é aparentemente intencional. Consulte o *white paper* Microsoft DNS indicado no artigo 286834 do Microsoft Knowledge Base (support.microsoft.com) para informações adicionais.

Registrando em log

Consulte o Capítulo 10 para informações adicionais sobre syslog.

Os recursos de registro em log de **named** são suficientemente flexíveis para deixar você de cabelo em pé. O BIND 4 usava syslog para registrar mensagens de erro e anomalias. Versões recentes generalizam os conceitos do syslog adicionando uma outra camada de indireção e suporte a registro em log diretamente nos arquivos. Antes de mergulharmos no assunto, vamos examinar o miniglossário de termos de registro do BIND mostrado na Tabela 15.12.

Você configura o registro em log do BIND com uma instrução `logging` em **named.conf**. Você primeiro define os canais, os destinos possíveis para mensagens. A seguir informamos as várias categorias de mensagem que vão para determinados canais.

Quando uma mensagem é gerada, ela recebe uma categoria, um módulo e uma gravidade em seu ponto de origem. Ela é então distribuída para todos os canais associados a seu módulo e sua categoria. Cada canal possui um filtro de gravidade que informa que nível de gravidade uma mensagem deve ter para poder passar. Canais que levam ao syslog também são filtrados de acordo com as regras contidas em **/etc/syslog.conf**.

Tabela 15.12 Um léxico de registro em BIND

Termo	O que isso significa
Channel (canal)	Local onde mensagens podem ser enviadas: syslog, um arquivo ou **/dev/null**[a]
Category (categoria)	Uma classe de mensagens que **named** é capaz de gerar; por exemplo, mensagens sobre atualizações dinâmicas ou mensagens sobre respostas a consultas
Module (módulo)	O nome do módulo-fonte que gera uma mensagem
Facility (recurso)	Um nome de recurso syslog. O DNS não possui um recurso próprio específico, mas você pode escolher entre as várias opções padrão
Severity (gravidade)	A gravidade de uma mensagem de erro, o que o syslog chama de prioridade

a. **/dev/null** é um pseudodispositivo que descarta todas as entradas.

Eis a descrição geral de uma instrução `logging`:

```
logging {
     def_do_canal;
     def_do_canal;
     ...
     category nome_do_categoria {
         nome_do_canal;
         nome_do_canal;
         ...
     };
};
```

Canais

Uma *definição_do_canal* tem um aspecto ligeiramente diverso dependendo de o canal ser um canal de arquivos ou um canal de syslog. Você tem de escolher `file` ou `syslog` para cada canal; um canal não pode se comportar de ambas as maneiras simultaneamente.

```
channel nome_do_canal {

    file caminho [versions numvers | unlimited] [size especif-tamanho];
    syslog recurso;
    severity gravidade;

    print-category yes | no;
    print-severity yes | no;
    print-time yes | no;
};
```

Para um canal de arquivos, *numvers* informa o número de versões de backup de um arquivo que devem ser mantidas e *especif-tamanho* especifica qual o tamanho máximo permitido para o arquivo (exemplos: `2048`, `100k`, `20m`, `15g`, `unlimited`,

`default`) antes dele automaticamente sofrer uma rotação. Se você chamar um canal de arquivos de **mylog**, as versões rotacionadas serão **mylog.0**, **mylog.1** e assim por diante.

Consulte a página 151 para uma lista de nomes dos recursos syslog.

No caso de syslog, *recurso* especifica qual o nome do recurso utilizado para registrar em log a mensagem. Poderá ser qualquer recurso padrão. Na prática, somente `daemon` e `local0` a `local7` são opções razoáveis.

O restante das instruções em uma *definição_do_canal* é opcional. *gravidade* também pode ter os valores (em ordem descendente) `critical`, `error`, `warning`, `notice`, `info` ou `debug` (com um nível numérico opcional, por exemplo, `severity debug 3`). O valor `dynamic` também é reconhecido e concorda com o nível de depuração atual do servidor.

As diversas opções `print` acrescentam ou suprimem prefixos de mensagens. Syslog anexa o horário e o host relator a cada mensagem registrada em log, mas não coloca informações relativas à gravidade ou à categoria. O nome do arquivo fonte (módulo) que gerou a mensagem também está disponível como uma opção `print`. Faz sentido ativar `print-time` somente para canais de arquivos; não há necessidade de duplicar os registros de data/hora do syslog.

Os quatro canais listados abaixo são predefinidos por padrão. Tais padrões devem ser bons para a maioria das instalações.

Tabela 15.13 Canais predefinidos de registro em log no BIND

Nome do canal	O que ele faz
`default_syslog`	Envia ao syslog, daemon de recursos, informações de gravidade
`default_debug`	Registra informações no arquivo **named.run** em log, gravidade configurada como dinâmica
`default_stderr`	Envia informações de gravidade ao erro padrão do processo **named**
`null`	Descarta todas as mensagens

Categorias

As categorias são determinadas pelo programador no momento em que o código é escrito; elas organizam as mensagens de log por tópico ou funcionalidade em vez de apenas pela severidade. A Tabela 15.14 mostra a lista atual das categorias de mensagem.

Tabela 15.14 Categorias de registro em log do BIND

Categoria	O que ela inclui
`client`	Requisições de clientes
`config`	Processamento e análise sintática de arquivo de configuração
`database`	Mensagens sobre operações em bancos de dados
`default`	O padrão para categorias sem opções específicas de registro em log
`delegation-only`	Consultas forçadas para NXDOMAIN pelas zonas apenas de delegação
`dispatch`	Despachando pacotes entrantes para os módulos de servidor
`dnssec`	Mensagens DNSSEC
`general`	Que abrange tudo para mensagens não classificadas
`lame-servers`	Servidores que devem servir uma zona, mas não o fazem[a]
`network`	Operações de rede
`notify`	Mensagens sobre o protocolo de notificação de "zona modificada"
`queries`	Uma mensagem de log curta para cada consulta que o servidor recebe (!)
`resolver`	Conversão DNS, por exemplo, pesquisas recursivas para clientes
`security`	Requisições aprovadas/desaprovadas
`unmatched`	Consultas identificadas não podem classificar (classe ruim, sem visualização)
`update`	Mensagens sobre atualizações dinâmicas
`xfer-in`	Transferências de zona que o servidor está recebendo
`xfer-out`	Transferências de zona que o servidor está enviando

a. Tanto a zona pai como a zona filha poderiam estar com falhas; é impossível afirmar sem investigação.

Mensagens de log

A configuração padrão de registro em log é:

```
logging {
    category default { default_syslog; default_debug; };
};
```

Você deve observar os arquivos de log ao fazer modificações importantes no BIND, e talvez aumentar o nível de registro em log. Mais tarde, reconfigure para preservar apenas as mensagens sérias depois de verificar se **named** é estável.

Registro em log de consultas pode ser bastante educativo. Você pode verificar se suas cláusulas `allow` estão funcionando, ver quem está consultando você, identificar clientes falhos etc. É uma boa verificação a realizar depois das reconfigurações importantes, especialmente se você tiver uma boa percepção de como sua carga de consulta estava antes das alterações.

Para começar a consultar o registro em log, simplesmente direcione a categoria `queries` para um canal. Gravar no syslog é menos eficiente que gravar diretamente em um arquivo, portanto use um canal de arquivo em um disco local ao registrar em log cada consulta. Tenha reservado uma boa quantidade de espaço em disco e esteja pronto para desativar o registro em log de consultas assim que já tiver dados suficientes. (**rndc querylog** ativa e desativa o registro em log das consultas dinamicamente.)

As vistas podem ser irritantes de depurar, mas felizmente, a que correspondeu a uma consulta particular é registrada em log bem como a consulta.

Algumas mensagens de log comuns são listadas a seguir.

- *Lame server*. Se obtiver essa mensagem sobre uma de suas próprias zonas, você configurou algo incorretamente. A mensagem é relativamente inofensiva se for sobre alguma zona externa, na Internet; trata-se de problema alheio. Trata-se de uma boa mensagem para jogar fora direcionando-a para o canal nulo.
- *Bad referral*. Essa mensagem indica uma comunicação incorreta entre os servidores de nomes de uma zona.
- *Not authoritative for*. Um servidor escravo é incapaz de obter dados autorizados para uma zona. Talvez ele esteja apontando para o mestre incorreto ou talvez o mestre tenha tido problemas para carregar a zona em questão.
- *Rejected zone*. **named** rejeitou um arquivo de zonas, pois continha erros.
- *No NS RRs found*. Um arquivo de zonas não incluiu registros NS após o registro SOA. Poderia ser que os registros estivessem ausentes ou que não começassem com tabulação ou espaço em branco. No último caso, os registros não são anexados à zona do registro SOA e são, conseqüentemente, mal interpretados.
- *No default TTL set*. A maneira preferida para configurar o TTL padrão é com uma instrução `$TTL` no topo do arquivo de zonas. Essa mensagem de erro indica que a `$TTL` não está presente; ela é necessária no BIND 9.
- *No root name server for class*. Seu servidor está tendo problemas para encontrar os servidores de nomes-raiz. Verifique seu arquivo hints e a conectividade do servidor com a Internet.
- *Address already in use*. A porta na qual **named** quer executar já está sendo usada por um outro processo, provavelmente uma outra cópia de **named**. Se você não encontrar nenhum outro **named**, ele pode ter travado e deixado um socket de controle **rndc** aberto, o qual você terá de rastrear e remover.[26] Uma boa maneira de corrigir isto é parar e reiniciar o processo **named**, por exemplo:

  ```
  # /etc/init.d/named stop (or /sbin/service named stop in Red Hat)
  # /etc/init.d/named start (or /sbin/service named start in Red Hat)
  ```

- *Dropping source port zero packet from …* Versões recentes do BIND permitem que você configure o número da porta de origem de consultas, e os administradores de sistemas utilizam esse recurso para acrescentar regras a seus firewalls que podem reconhecer seus pacotes DNS pela porta de origem. Entretanto, 0 é um valor ilegal para um número de porta TCP/UDP. Se a mensagem de erro estiver relacionada a um de seus hosts, você deve alterar a diretiva `query-source` no arquivo **named.conf** do host para corrigir esse erro.
- *Denied update from […] for …* Uma atualização dinâmica para uma zona foi tentada e negada devido à cláusula `allow-update` ou `update-policy` em **named.conf** para essa zona.

26. Em um servidor gTLD, essa mensagem provavelmente significa que com ainda está carregando. :-)

Exemplo de configuração de registro em log

O trecho a seguir do arquivo ISC **named.conf** para um servidor de nomes TLD ocupado ilustra um abrangente processo de registro em log.

```
logging {

      channel default_log {
          file "log/named.log" versions 3 size 10m;
          print-time yes;
          print-category yes;
          print-severity  yes;
          severity info;
      };

      category default { default_log; default_debug; };

      channel xfer-log {
         file "log/xfer.log" versions 3 size 10m;
         print-category yes;
         print-severity yes;
         print-time yes;
         severity info;
      };
      channel db-log {
         file "log/db.log" versions 3 size 1M;
         severity debug 1;
         print-severity yes;
         print-time yes;
     };

         category database { db-log; };
         category dnssec { xfer-log; };
         category xfer-in { xfer-log; };
         category xfer-out { xfer-log; };
         category notify { xfer-log; };
};
```

Níveis de depuração

Os níveis de depuração **named** são indicados pelos inteiros de 0 a 100. Quanto maior o número, mais longa a saída. O nível 0 desativa a depuração. Os níveis 1 e 2 são bons para depurar sua configuração e banco de dados. Níveis acima próximos de 4 são apropriados para aqueles que fazem a manutenção do código.

Você invoca a depuração na linha de comando **named** com o flag **-d**. Por exemplo,

```
# named -d2
```

iniciaria **named** no nível de depuração 2. Por padrão, informações de depuração são gravadas no arquivo **named.run** no diretório de trabalho atual a partir do qual **named** é iniciado. O arquivo **named.run** cresce muito rapidamente; portanto, não saia para tomar uma cerveja enquanto estiver no processo de depuração, ou terá problemas maiores ao retornar.

Você também pode ativar a depuração enquanto **named** está executando com **rndc trace**, o que incrementa o nível de depuração por 1 ou com **rndc trace** *nível*, que configura o nível de depuração como o valor especificado. **rndc notrace** desativa a depuração completamente. Você também pode ativar a depuração definindo um canal de registro em log que inclua uma especificação de gravidade como

```
severity debug 3;
```

que enviará todas as mensagens de depuração até o nível 3 para determinado canal. Outras linhas na definição do canal especificam o destino dessas mensagens de depuração. Quanto maior o nível de gravidade, mais informações são registradas em log.

Notar as saídas dos registros em log ou a saída de depuração ilustra a freqüência com que os dados DNS são configurados erroneamente no mundo real. Aquele maldito pontinho no final dos nomes (ou melhor, a falta dele) é responsável por um nível assustador de tráfego DNS. Teoricamente, o ponto é necessário no final de cada nome de domínio totalmente qualificado.

Depurando com rndc

A Tabela 15.15 mostra algumas opções aceitas por **rndc**. Digitar **rndc** sem argumentos fornece uma lista dos comandos disponíveis. Comandos que produzem arquivos os colocam no diretório especificado como diretório inicial de **named** em **named.conf**.

rndc reload faz com que **named** releia seu arquivo de configuração e recarregue arquivos de zonas. O comando **reload** *zona* é útil apenas quando uma única zona foi modificada e você não deseja recarregar todas as zonas, especialmente em um servidor ocupado. Você também pode especificar uma classe e vista para recarregar apenas a vista selecionada dos dados da zona. **rndc reconfig** facilita a adição de novas zonas sem realizar uma recarga completa; relê o arquivo de configuração e carrega todas as novas zonas sem alterar as zonas existentes.

rndc freeze *zona* pára as atualizações dinâmicas e reconcilia o arquivo journal das atualizações pendentes com os arquivos de dados. Depois de congelar a zona, você pode editar os dados sobre a zona manualmente. Desde que a zona esteja congelada, as atualizações dinâmicas serão recusadas. Depois de terminar de editar, utilize **rndc thaw** *zona* para começar a aceitar atualizações dinâmicas novamente.

rndc dumpdb faz com que **named** faça o dump do seu banco de dados em **named_dump.db**. O arquivo de dump é grande e inclui não apenas dados locais, mas também quaisquer dados armazenados em cache que o servidor de nomes acumulou. Um dump recente do cache do banco de dados em nosso servidor primário ocupava mais de 16 MB, mas os dados de zona carregados ocupavam menos do que 200 K. Muito cache nesse caso.

Se estiver executando **named** em um ambiente em que foi aplicado **chroot**, você terá de dar a **rndc** o caminho para o socket **rndc**, pois ele não estará em **/var/run** que é o local em que **rndc** esperaria que ele estivesse.

Use algo como:

```
$ sudo rndc -l /var/named/var/run/rndc
```

No BIND 9, é importante ter certeza de que a versão de **named** e a de **rndc** sejam coincidentes, para que você não receba uma mensagem de erro sobre uma discordância entre versão de protocolos. Apenas instale esses dois arquivos ao atualizar.

Tabela 15.15 Comandos rndc[a]

Comando	Função
dumpdb	Faz um dump do banco de dados DNS para **named_dump.db**
flush [*vista*]	Esvazia todos caches ou aqueles para uma *vista* especificada
flushname *nome* [*vista*]	Esvazia o *nome* especificado do cache do servidor
freeze *zona* [*classe* [*vista*]]	Suspende atualizações em uma *zona* dinâmica
thaw *zona* [*classe* [*vista*]]	Retoma atualizações para uma *zona* dinâmica
halt	Pára **named** sem gravar as atualizações pendentes
querylog	Ativa/desativa rastreamento de consultas que chegam
notrace	Desativa a depuração
reconfig	Recarrega o arquivo de configuração e carrega quaisquer novas zonas
refresh *zona* [*classe* [*vista*]]	Agenda a manutenção de agendas para uma *zona*
reload	Recarrega **named.conf** e arquivos de zonas
reload *zona* [*classe* [*vista*]]	Recarrega somente a *zona* ou a *vista* especificada
restart[b]	Reinicia o servidor
retransfer *zona* [*classe* [*vista*]]	Copia novamente os dados para a *zona* do servidor mestre
stats	Faz um dump das estatísticas para **named.stats**
status	Exibe o estado atual do processo **named** em execução
stop	Salva atualizações pendentes e então pára **named**
trace	Incrementa o nível de depuração por 1
trace *nível*	Altera o nível de depuração

a. O argumento *class* aqui é o mesmo que para os registros de recursos, em geral IN para a Internet.
b. Ainda não implementada no BIND 9 (9.3.0), mas prometida.

Estatísticas do BIND

named mantém algumas informações de resumo cuja cópia de dump pode ser feita no arquivo **named.stats** no seu diretório funcional ao receber uma sinalização de **rndc**:

```
$ sudo rndc stats
```

Eis um exemplo da saída de um servidor ocupado em isc.org:

```
+++ Statistics Dump +++ (1088897026)
success 19917007
referral 981446459
nxrrset 9958824
nxdomain 199644113
recursion 0
failure 322556
--- Statistics Dump --- (1088897026)
```

As estatísticas mostram o número de sucesso *versus* falha nas pesquisas e categorizam os diversos tipos de erros. Esse servidor respondeu a 19,9 milhões de consultas com sucesso, disponibilizou 980 milhões de referrals, respondeu "no such domain" (`nxdomain`) aproximadamente 200 milhões de vezes, respondeu "no such resource record set" (`nxrrset`) mais ou menos 10 milhões de vezes e não pôde responder 320 mil vezes.

Os 200 milhões de erros `nxdomain` (quase 20% de um total de 1 bilhão de consultas) seriam um número muito alto se suas fontes primárias fossem erros de digitação e erros de ortografia do usuário. Nesse caso, a maioria dos erros é proveniente das configurações erradas do Windows e de um bug no conversor da Microsoft[27] que envia muitas consultas aos registros de endereços com o campo de dados configurado como um endereço IP em vez de um nome de host como requerido pela especificação do DNS. Essas consultas bizarras parecem ter um domínio de primeiro nível algo como 56 (ou seja lá qual for o último byte da string de endereço IP), de modo que elas fazem com que o servidor DNS local incomode os servidores-raiz sobre esse domínio inexistente, levando, em última instância, a um erro `nxdomain`.

A entrada `failure` conta as falhas que não são `nxdomain` ou `nxrrset`.

Depurando com dig

Três ferramentas podem ser utilizadas por meio do shell para consultar o banco de dados DNS: **nslookup**, **dig**, e **host**. Todas são distribuídas com o BIND. **nslookup** é a mais antiga dessas ferramentas e sempre fez parte da distribuição do BIND. **dig**, o buscador de informações de domínio, foi originalmente escrito por Steve Hotz; essa ferramenta foi reescrita para o BIND 9 por Michael Sawyer e agora também é distribuída com o BIND 9. **host**, de Eric Wassenaar, é uma outra ferramenta de código-fonte aberto; ela apresenta uma saída amigável ao usuário e funções que verificam a sintaxe dos seus arquivos de zona.

Recomendamos e preferimos **dig** ao **nslookup**; **host** também é uma boa ferramenta. Aqui, discutimos apenas o **dig** em detalhes. Às vezes, talvez você obtenha diferentes resultados a partir dessas ferramentas devido às diferentes bibliotecas de conversão que elas utilizam: **dig** e **host** utilizam a conversão do BIND e **nslookup** tem sua conversão própria.

Por padrão, **dig** utiliza o servidor de nomes local e retorna informações sobre os endereços. O argumento *@nameserver* faz com que ele consulte um servidor de nomes específico.[28] A capacidade de consultar um servidor particular permite confirmar se todas as alterações feitas em uma zona foram propagadas para os servidores secundários e para o mundo externo. Esse recurso é especialmente útil se você utilizar vistas e precisar verificar se elas foram configuradas corretamente.

Se especificar um tipo de registro, **dig** só consultará esse tipo. O pseudotipo **ANY** retorna todos os dados associados com o nome especificado. Um outro flag útil é **-x**, que inverte os bytes de um endereço IP para você e faz uma consulta reversa. O flag **+trace** mostra os passos iterativos no processo de conversão das raízes.

Eis como é a saída do **dig**:

```
$ dig yahoo.com

; <<>> DiG 9.3.0rc2 <<>> yahoo.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16507
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 5, ADDITIONAL: 5

;; QUESTION SECTION:
;yahoo.com.                IN      A
```

27. Corrigido no Windows 2000 Service Pack 2 e nas versões superiores do Windows

28. Se o servidor não existir, as versões mais antigas do **dig** irão silenciosamente recorrer ao servidor local, portanto leia sua saída atentamente para verificar a fonte. A partir do BIND 9.2, **dig** retorna um erro e nenhuma resposta nessa situação.

```
;; ANSWER SECTION:
yahoo.com. 300          IN      A       66.94.234.13
yahoo.com. 300          IN      A       216.109.127.28
yahoo.com. 300          IN      A       216.109.127.29

;; AUTHORITY SECTION:
yahoo.com. 74804        IN      NS      ns3.yahoo.com.
yahoo.com. 74804        IN      NS      ns4.yahoo.com.
yahoo.com. 74804        IN      NS      ns1.yahoo.com.
yahoo.com. 74804        IN      NS      ns2.yahoo.com.

;; ADDITIONAL SECTION:
ns1.yahoo.com.    72593  IN      A       66.218.71.63
ns2.yahoo.com.    72596  IN      A       66.163.169.170
ns3.yahoo.com.    72594  IN      A       217.12.4.104
ns4.yahoo.com.    72594  IN      A       63.250.206.138

;; Query time: 6 msec
;; SERVER: 204.152.184.109#53(204.152.184.109)
;; WHEN: Sun Jul  9 16:03:43 2006
;; MSG SIZE  rcvd: 245
```

dig inclui a notação `aa` na sua lista de `flags` se a resposta for autorizada. Aqui estamos perguntando ao servidor de nomes local sobre yahoo.com, portanto a resposta não é autorizada.

A saída de **dig** inclui não só as mesmas informações de domínio, como também o número de consultas enviadas e o tempo de ida e volta das respostas. A saída é formatada corretamente para ser usada em um arquivo de zonas, que é particularmente prático quando buscamos informações de servidores-raiz para nossos arquivos hint. O ponto-e-vírgula é o caractere de comentário em um arquivo de zona é utilizado pelo **dig** para embutir comentários na sua saída.

Ao testar uma nova configuração, certifique-se de pesquisar dados tanto para hosts locais como para hosts remotos. Se você puder acessar um host pelo endereço IP, mas não pelo nome, o DNS é provavelmente o culpado.

Delegações defeituosas

Ao registrar um nome de domínio, você está solicitando que uma parte da árvore de nomes DNS seja delegada ao seu servidor de nomes primário e ao seu administrador DNS. Se você nunca usa o domínio ou altera os servidores de nomes sem atualizar os registros de adesão do domínio pai, isso resultará em uma "delegação defeituosa".

Os efeitos de uma delegação defeituosa podem ser muito maléficos. Se um usuário tentar consultar um host em seu domínio incorreto, o seu servidor de nomes recusará a consulta. O DNS tentará novamente a consulta centenas de vezes, massacrando tanto o seu servidor mestre como os servidores-raiz. O BIND usa uma "área de penalidade" de servidor defeituoso para ajudar na carga criada pela delegação defeituosa, mas os servidores Microsoft não implementam esse recurso.

Há duas maneiras de localizar delegações defeituosas: revisando os arquivos de log ou utilizando uma ferramenta chamada **doc** (domain obscenity control). Examinaremos alguns exemplos de **doc** na próxima seção, mas primeiro vamos revisar algumas entradas de log.

Muitos sites apontam o canal de registro em log `lame-servers` para **/dev/null** sem se preocuparem com as delegações defeituosas de outras pessoas. Isso é bom contanto que o próprio domínio esteja perfeitamente limpo e ele mesmo não seja uma fonte ou vítima das delegações defeituosas.

Em um arquivo de log com 3,5 MB depois de aproximadamente uma semana (no nível "info"), mais de um terço das entradas de log eram para delegações defeituosas. Destas, 16% envolviam consultas aos servidores-raiz, supostamente para domínios inexistentes. Um usuário persistente consultou os servidores-raiz procurando tokyotopless.net centenas de vezes. Lastimável. Eis o exemplo de uma mensagem do log de delegação defeituosa:

```
Jan 29 05:34:52 ipn.caida.org named[223]: Lame server on 'www.games.net' (in 'GAMES.net'?):
    [207.82.198.150].53 'NS2.EXODUS.net'
```

Eis como monitoramos o problema com **dig** (truncamos parte da saída prolixa de **dig**):

```
$ dig www.games.net.
;; —
;; QUESTIONS:
;;      www.games.net, type = A, class = IN
;; ANSWERS:
```

```
www.games.net.     3600    A       209.1.23.92
:: AUTHORITY RECORDS:
games.net.         3600    NS      ns.exodus.net.
games.net.         3600    NS      ns2.exodus.net.
games.net.         3600    NS      ns.pcworld.com.
:: ADDITIONAL RECORDS: …
```

A primeira consulta no servidor local retorna o registro de endereço para www.games.net e uma lista de servidores autorizados. O servidor em ns.exodus.net funcionava bem quando o consultávamos (não mostrado), mas ns2.exodus.net é uma outra história:

```
$ dig @ns2.exodus.net www.games.net.
:: QUESTIONS:
::      www.games.net, type = A, class = IN
:: AUTHORITY RECORDS:
net.               244362  NS      F.GTLD-SERVERS.net.
net.               244362  NS      J.GTLD-SERVERS.net.
net.               244362  NS      K.GTLD-SERVERS.net.
net.               244362  NS      A.GTLD-SERVERS.net.
:: …
```

ns2 é listado como sendo um servidor autorizado para www.games.net, mas ele não retorna registros e nos refere aos servidores para o domínio de primeiro nível. Portanto, podemos concluir que ns2.exodus.net foi configurado incorretamente.

Às vezes, quando você utiliza **dig** em um servidor autorizado como uma tentativa de localizar defeitos, o **dig** não retorna nenhuma informação. Tente a consulta novamente com o flag **+norecurse** para que possa ver exatamente o que o servidor em questão conhece.

doc: domain obscenity control

doc não é uma parte do BIND. Atualmente ele é mantido por Brad Knowles, de cujo site Web podemos fazer seu download:

www.shub-internet.org/brad/dns/ *(Nota: "shub", não "shrub")*

doc é um script de shell C. Se planejar utilizá-lo ou executá-lo a partir do **cron**, você precisará editar o script e configurar a variável `auxd` para que ela aponte para o diretório de instalação.

doc verifica as delegações fazendo chamadas repetidas a **dig**. Ele informa inconsistências, erros e outros problemas relacionados ao nome de um domínio particular. A saída de tela resume as questões que ele encontra; ele também cria um arquivo de log prolixo no diretório atual com os detalhes.

Vejamos alguns exemplos completos. Primeiro, atrust.com, em que tudo parece estar OK:

```
$ doc atrust.com
Doc-2.2.3: doc atrust.com
Doc-2.2.3: Starting test of atrust.com.   parent is com.
Doc-2.2.3: Test date - Thu Jul  6 08:54:38 MDT 2006
Summary:
   No errors or warnings issued for atrust.com.
Done testing atrust.com.  Thu Jul  6 08:54:46 MDT 2006
```

O arquivo de log (**log.atrust.com.**) menciona que tanto com como atrust.com concordam sobre as identidades dos servidores de nomes para atrust.com e que esses dois servidores têm o mesmo número serial nos seus registros SOA. **doc** também verifica os registros PTR para cada um dos servidores de nomes. As consultas **dig** reais que foram realizadas pelo **doc** são listadas no final do arquivo de log, junto com as respostas que elas provocaram.

Nosso próximo exemplo é para o domínio cs.colorado.edu:

```
$ doc cs.colorado.edu
Doc-2.2.3: doc cs.colorado.edu
Doc-2.2.3: Starting test of cs.colorado.edu.   parent is colorado.edu.
Doc-2.2.3: Test date - Thu Jul  6 08:55:15 MDT 2006
dig: Couldn't find server 'rs0.netsol.com.': No address associated with hostname
DIGERR (NOT_ZONE): dig @rs0.netsol.com. for SOA of parent (colorado.edu.) failed
DIGERR (NOT_AUTHORIZED): dig @pacifier.com. for SOA of cs.colorado.edu. failed
dig: Couldn't find server 'xor.com.': No address associated with hostname
DIGERR (FORMAT_ERROR): dig @xor.com. for SOA of cs.colorado.edu. failed
Summary:
   ERRORS found for cs.colorado.edu. (count: 2)
```

```
   Incomplete test for cs.colorado.edu. (2)
Done testing cs.colorado.edu.  Thu Jul  6 08:55:26 MDT 2006
```

Aqui podemos ver vários problemas no domínio pai, colorado.edu. Os dados da zona do pai listam incorretamente rs0.netsol.com como um servidor para rs0.colorado.edu e xor.com e pacifier.com como os servidores para cs.colorado.edu. Para ver toda a história, teríamos de executar **doc** contra o domínio pai, colorado.edu, e revisar o arquivo de log detalhado.

Por fim, vamos examinar nsa.cctldmd.net:

```
$ doc nsa.cctldmd.net
Doc-2.2.3: doc nsa.cctldmd.net
Doc-2.2.3: Starting test of nsa.cctldmd.net.   parent is cctldmd.net.
Doc-2.2.3: Test date - Thu Jul  6 08:56:20 MDT 2006
SYSerr: No servers for nsa.cctldmd.net. returned SOAs ...
Summary:
   YIKES: doc aborted while testing nsa.cctldmd.net.  parent cctldmd.net.
   WARNINGS issued for nsa.cctldmd.net. (count: 2)
   Incomplete test for nsa.cctldmd.net. (1)
Done testing nsa.cctldmd.net.  Thu Jul  6 08:56:22 MDT 2006
```

O domínio cctldmd.net tem uma delegação defeituosa para nsa.cctldmd.net. A entrada abaixo do arquivo **log.nsa.cctldmd.net.** esclarece o problema:

```
WARNING: register.dns.md. claims to be authoritative for nsa.cctldmd.net.
   == but no NS record at parent zone
```

Se gerenciar um domínio que inclui subdomínios (ou não confiar nos gerenciadores do seu domínio pai), considere executar **doc** a partir do **cron** uma vez por semana para verificar se todas as delegações relacionadas ao seu domínio estão corretas.

Outras ferramentas de verificação de sanidade do DNS

Há várias outras ferramentas que verificam vários aspectos do seu ambiente de DNS. **named-checkconf** e **named-checkzone** são distribuídas com o BIND 9; essas ferramentas verificam a sintaxe básica (não a semântica) do arquivo **named.conf** e dos seus arquivos de zona. A ferramenta original para verificar o DNS é **nslint**, escrita por Leres Craig quando trabalhava nos Lawrence Berkeley Labs. A ferramenta **lamers** (do mesmo site Web da **doc**) faz uma varredura pelos arquivos de log e envia um e-mail aos administradores de DNS dos sites ofensores informando que eles têm uma delegação defeituosa e descrevendo como corrigir o problema. **DDT**, de Jorge Frazão e Artur Romão, depura os dados armazenados em cache.

dnswalk atravessa sua árvore de delegação e identifica inconsistências entre o pai e o filho ou entre os registros reversos e de encaminhamento. Ele também localiza pontos ausentes, registros de adesão desnecessários etc. É uma importuna higiene geral no DNS. **dnswalk** precisa ser capaz de fazer transferências de zona a fim de que sua mágica funcione.

Questões de desempenho

Atualmente, o BIND 9 tem algumas questões de desempenho que estão sendo resolvidas pelo ISC. O desempenho do BIND 9 está se aproximando daquele do BIND 8, mas sua eficiência nas arquiteturas de multiprocessadores até agora tem decepcionado. As máquinas f.root-servers.net utilizam o BIND 9 e tratam sem problemas cerca de 5.000 consultas por segundo; o desempenho provavelmente seja razoável para a maioria dos sites.

Já afirmamos isso várias vezes, mas vale repetir: configure seu TTL com valores razoáveis (semanas ou dias, não horas ou minutos). O uso de TTL curtos pune tanto você (porque é necessário reservar constantemente os mesmos registros) como seus clientes Web (porque eles precisam buscá-los constantemente).

A paginação degrada o desempenho do servidor de maneira não linear, assim não seja um sovina com a memória nos servidores que executam **named**. Você precisará esperar mais ou menos uma semana para que o footprint de memória do **named** estabilize; consulte a página 293.

Utilize encaminhadores. Consulte a página 297 para uma discussão sobre a arquitetura de encaminhamento.

15.17 ESPECIFICIDADES DAS DISTRIBUIÇÕES

Esta seção descreve os detalhes do software **named** de cada distribuição e suas configurações padrão. As distribuições do Linux são muito boas para a atualização quando novas versões do software BIND são distribuídas pelo ISC.

Cada uma das nossas de distribuições de referência utiliza o BIND como o servidor de nomes padrão, mas somente um subconjunto das distribuições o inclui na instalação padrão. As outras distribuições incluem os utilitários BIND (**dig**, **nslookup**, **host** etc.), mas não o próprio software do servidor de nomes. Você tem de especificar que quer instalar o BIND ao gerar a máquina ou instalar o pacote BIND separadamente.

Um cliente Linux BIND especifica o domínio padrão, os outros domínios para pesquisar quando os nomes não são completamente qualificados e os endereços IP dos servidores de nomes locais no arquivo **/etc/resolv.conf**. Um host Linux que atua como um servidor BIND também utiliza os arquivos **named** listados na Tabela 15.16.

Tabela 15.16 Arquivos BIND no Linux.

Arquivo	Diretório[a]	Descrição
resolv.conf	**/etc**	Arquivo de configuração de biblioteca de conversão
named, lwres	**/usr/sbin**	Daemon de servidor de nomes
named.conf	**/etc** **/etc/bind**	Arquivo de configuração **named** (RHEL, Fedora e SUSE) Arquivo de configuração **named** (Debian, Ubuntu)
named.pid	**/var/run/named** **/var/run/bind/run** *home*/**var/run/named**	PID do **named** em execução (RHEL/Fedora) PID do **named** em execução (Debian, Ubuntu) PID do **named** em execução (SUSE)
named.run	*home*	Saída gerada no modo de depuração
named.stats	*home* *home*/**var/log**	Saída de informações estatísticas Saída das estatísticas (SUSE)
named_dump.db	*home* *home*/**var/log**	Dump do banco de dados inteiro Faz um dump do banco de dados inteiro (SUSE)

a. Por *home*, temos em mente o diretório especificado em **named.conf** como o diretório inicial para arquivos BIND.

O Linux inclui um arquivo de switch, **/etc/nsswitch.conf**, que especifica como os mapeamentos de nome de host para endereços IP devem ser feitos e se o DNS deve ser experimentado em primeiro lugar, em último ou não deve ser utilizado. Se não houver nenhum arquivo de switch presente, o comportamento padrão é:

```
hosts: dns [!UNAVAIL=return] files
```

A cláusula `!UNAVAIL` significa que se o DNS estiver disponível, mas um nome não for encontrado lá, a tentativa de pesquisa deve falhar em vez de tentar continuar para a próxima entrada (neste caso, o arquivo **/etc/hosts**). Se nenhum servidor de nomes estivesse ativo (como poderia ser o caso durante o processo de inicialização), o processo de pesquisa *consultaria* o arquivo **hosts**.

Todas as nossas distribuições de exemplo fornecem este arquivo **nsswitch.conf** padrão:

```
hosts: files dns
```

Não há uma maneira "melhor" de configurar as pesquisas — isso depende de como seu site é gerenciado. Em geral, preferimos manter o maior número possível de informações sobre host no DNS em vez de no NIS ou em arquivos simples, mas também tentamos preservar a capacidade de recorrer ao arquivo **hosts** estático durante o processo de inicialização se necessário.

Instalar o RHEL ou o pacote Fedora BIND (atualmente **bind-9.2.3-13.i386.rpm**) coloca o binário em **/usr/sbin**, coloca as páginas man em **/usr/share/man**, adiciona um usuário e grupo chamado "named" e cria diretórios para os arquivos de zona. O usuário "named" tem acesso aos arquivos de dados por meio das permissões de grupo.

A menos que você altere isso no **/etc/sysconfig/named**, o arquivo de configuração **named.conf** é colocado em **/etc** (como Paul Vixie e Deus pensavam) e os arquivos de zona são colocados em **/var/named**. Não são fornecidos exemplos de arquivos, mas o pacote **bindconf** deve tê-los.

O Debian e o Ubuntu fornecem uma escolha entre o BIND 8 e o BIND 9 dependendo do pacote que você instala (**bind** *versus* **bind9**). Os detalhes a seguir supõem o BIND 9.

Os programas e arquivos têm um proprietário root e um proprietário de grupo "bind" com as permissões configuradas para permitir acesso se **named** for invocado como usuário "bind" em vez de root.

Alguns arquivos úteis de exemplo permanecem ocultos em **/etc/bind**. São incluídos um arquivo **named.conf** e arquivos hint de zonas sobre root, host local, os endereços de broadcast e espaço de endereçamento privado. O **named.conf** fornecido inclui os arquivos **named.conf.options** e **named.conf.local**. Ele configura o diretório padrão do BIND como **/var/cache/bind**; assim como vem de fábrica, o diretório existe, mas é vazio.

A lógica por trás das informações de configuração está em **/etc** e as informações sobre a zona em **/var** são se você for um servidor secundário para outros sites você não controla o tamanho dos arquivos de zona que **named** gravará e assim você provavelmente irá querer manter os arquivos em **/var**. As zonas para as quais você é o servidor primário podem residir com os arquivos de configuração e utilizar caminhos absolutos no arquivo **named.conf** ou também podem residir em **/var/cache/bind**.

O modelo de arquivo **named.conf** do Debian não precisa ser modificado caso você queira usar um servidor somente para armazenamento em cache. Você deve adicionar todas as zonas para as quais você é autorizado, preferível ao arquivo **named.conf.local** fornecido.

Os arquivos de exemplo fornecidos pelo Debian utilizam alguns novos recursos do BIND para ajudar seus servidores a ser bons cidadãos DNS na rede. Por exemplo, eles configuram as zonas com e net como zonas somente de delegação a fim de impedir que os erros de digitação dos seus usuários gerem renda de anúncios publicitários para a VeriSign por meio da ferramenta Site Finder. Se você não utilizar o espaço de endereçamento privado (RFC1918) internamente, os arquivos de zona vazios da RFC1918 impedem que esses endereços escapem da rede local. Prefira o Debian!

O diretório **/usr/share/doc/bind9** contém várias referências úteis. Confira o arquivo **README.Debian** (até mesmo no Ubuntu) para entender a estratégia do Debian para configuração do BIND.

Instalar o BIND a partir de um pacote **rpm** no SUSE era frustrante por causa das dependências ausentes, mas depois que essas dependências foram solucionadas a instalação prosseguiu sem problemas. A instalação informa o que ela está fazendo e produz uma instalação de servidor de nomes razoável e bem documentada. Um assistente SUSE posteriormente sugeriu o uso do YaST em vez do **rpm** como a verdadeira maneira SUSE de instalar as coisas e essa é provavelmente a resposta certa.

Por padrão, **named** executa em um ambiente com **chroot** abaixo de **/var/lib/named** como um usuário e grupo "named". O instalador cria o diretório cadeia **chroot** e o preenche com todos os arquivos necessários para executar **named**, mesmo refinamentos como o soquete de domínio UNIX para o syslog. Arquivos de configuração extras (não **named.conf**) e os arquivos de zona residem em **/etc/named.d** e são copiados para a cadeia quando **named** é iniciado. Se você não quiser executar **named** em um ambiente com **chroot**, modifique a linha que informa que

```
NAMED_RUN_CHROOTED="yes"
```

está em **/etc/sysconfig/named**. Isso é tudo que você tem de alterar; os scripts de inicialização em **/etc/init.d** referem-se a essas informações e são capazes de iniciar **named** de qualquer maneira.

SUSE fornece um arquivo **/etc/named.conf** de exemplo com comentários úteis que explicam boa parte das opções. O arquivo **/etc/named.conf** do SUSE não é mundialmente legível como é comum ocorrer. O arquivo padrão importa um arquivo chamado **named.conf.include**, que então importa o arquivo **rndc-access.conf** de **/etc/named.d**, os dois são mundialmente legíveis. Não está completamente claro o que o SUSE tem em mente aqui no que diz respeito à segurança. **rndc** é pré-configurado para só aceitar comandos de controle a partir do endereço de host local.

O arquivo **named.conf** do SUSE pode ser utilizado exatamente como vem de fábrica para executar um servidor somente de cache. Se você quiser servir suas próprias zonas, coloque os arquivos de zona no diretório **/etc/named.d** e liste os nomes das zonas no arquivo **/etc/named.conf.include**.

A documentação ISC BIND reside em **/usr/share/doc/packages/bind9**.

15.18 LEITURA RECOMENDADA

O DNS e o BIND são descritos por uma série de fontes, inclusive pela documentação que acompanha a distribuição, capítulos em diversos livros sobre tópicos da Internet, um livro completo na série Nutshell, da O'Reilly, e vários recursos on-line.

Listas de discussão e grupos de discussão

As listas de discussão (*mailing lists*) a seguir estão associadas ao BIND:

- bind-announce – envie um e-mail para bind-announce-request@isc.org
- namedroppers – envie um e-mail para namedroppers-request@internic.net para se associar
- bind-users – envie um e-mail para bind-users-request@isc.org para se associar
- bind9-workers – envie um e-mail para bind9-workers-request@isc.org (para guerreiros da codificação)

Envie relatórios sobre bugs a bind9-bugs@isc.org.

Livros e documentação diversa

THE NOMINUM AND ISC BIND DEVELOPMENT TEAMS. *BINDv9 Administrator Reference Manual.* Disponível em: <www.isc.org>. Distribuidor BIND (**doc/arm**).

Esse documento delineia a administração e o gerenciamento do BIND 9. Uma nova versão da documentação BIND, *BIND Reference Manual* (provavelmente **doc/brm**), está sendo atualmente desenvolvida pelo ISC, mas ainda não está disponível para revisão. Um documento anterior, *Bind Operations Guide* (ou BOG, como foi chamado), descreveu a operação e a configuração do BIND 4. O BOG está incluído nas distribuições BIND até a versão 8.

ALBITZ, PAUL E LIU, CRICKET. *DNS and BIND*. 5. ed. Sebastopol, CA: O'Reilly Media, 2006.

Esse livro popular e bem respeitado sobre o BIND inclui cobertura tanto do BIND 8 como do BIND 9. Ele é bem completo.

Recursos on-line

O DNS Resources Directory, www.dns.net/dnsrd, é um útil conjunto de recursos e ponteiros para os recursos mantidos por András Salamon.

O Google indexou recursos DNS em

directory.google.com/Top/Computers/Internet/Protocols/DNS

A lista de discussão comp.protocols.dns.bind também contém excelentes informações.

As RFCs

As RFCs que definem o sistema DNS estão disponíveis em www.rfc-editor.org. Idéias iniciais e em evolução aparecem primeiro na série Internet-Drafts e posteriormente vão para a série RFC. Costumávamos listar aproximadamente uma página das RFCs mais importantes relacionadas com DNS, mas agora há tantas (mais de 100, com outros 50 como Internet drafts) que é melhor você pesquisar a página Web rfc-editor.org a fim de acessar todo o repositório de arquivos. Consulte os diretórios **doc/rfc** e **doc/draft** da distribuição BIND atual para ver toda a série.

Os padrões originais definitivos para o DNS, *vintage* 1987, são:

- 1034 – Domain Names: Concepts and Facilities
- 1035 – Domain Names: Implementation and Specification (Nomes de Domínio: Implementação e Especificação)

15.19 EXERCÍCIOS

E15.1 Explique a função de cada um dos registros DNS: SOA, PTR, A, MX e CNAME.

E15.2 Que são registros de adesão e por que são necessários? Use **dig** para descobrir os registros de adesão que conectam sua zona local ao seu pai.

E15.3 Quais são as implicações do cache negativo? Por que ele é importante?

E15.4 Crie pseudo-registros SPF para seu site para ajudar a controlar spam.

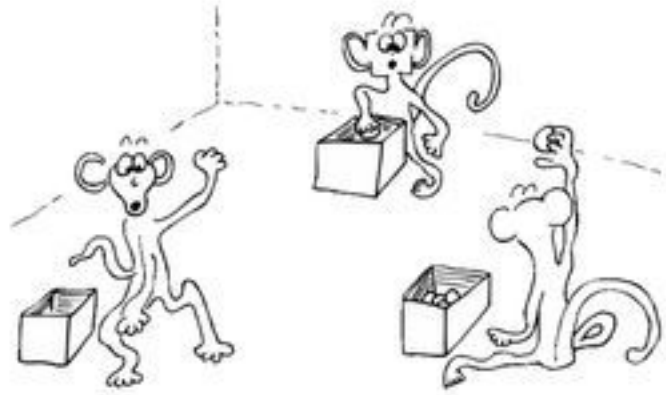
★E15.5 Quais são os passos necessários para configurar um novo domínio de segundo nível? Inclua tanto fatores técnicos quanto de procedimento.

★E15.6 Qual é a diferença entre uma resposta autorizada e uma não-autorizada a uma consulta DNS? Como você poderia garantir que uma resposta fosse autorizada?

★E15.7 Que máquina é o seu servidor de nomes local? Que medidas ela deve tomar para descobrir o nome www.admin.com, supondo-se que nenhuma informação sobre esse domínio seja armazenada em cache em algum ponto no DNS?

★E15.8 Explique a relevância para DNS do limite de 512 bytes dos pacotes UDP. Quais soluções intermediárias resolvem potenciais problemas?

16 O sistema de arquivos de rede



O Network File System, comumente conhecido como NFS, permite compartilhar sistemas de arquivos entre computadores. O NFS é quase transparente para os usuários e é "sem estado", o que significa que nenhuma informação é perdida quando um servidor NFS trava. Os clientes podem simplesmente aguardar o servidor retornar e depois continuar como se nada tivesse acontecido.

O NFS foi introduzido em 1985 pela Sun Microsystems. Ele foi originalmente implementado como um sistema de arquivos substituto para máquinas clientes sem disco, mas o protocolo demonstrou ser bem projetado e muito útil como uma solução genérica para compartilhamento de arquivos. De fato, é difícil lembrar de como era a vida antes do NFS. Todos os fornecedores UNIX fornecem uma versão do NFS; muitos usam código licenciado pela Sun. Todas as distribuições do Linux completas suportam o NFS.

16.1 Informações gerais sobre o NFS

Hoje, o NFS só é utilizado para compartilhar arquivos entre máquinas Linux e UNIX. Os clientes Windows devem utilizar o CIFS/Samba para o serviço de arquivos. Veja a página 568 para informações adicionais sobre o Samba e o CIFS.

O NFS consiste em vários componentes, incluindo um protocolo de montagem e um servidor de montagem, daemons que coordenam o serviço de arquivos básico e vários utilitários de diagnóstico. Parte do software no lado servidor, bem como no lado cliente, reside no kernel. Essas partes do NFS, porém, não precisam de nenhuma configuração e são amplamente transparentes sob o ponto de vista do administrador.

Versões do protocolo NFS

O protocolo NFS tem sido admiravelmente estável ao longo do tempo. Seu lançamento público original foi com a versão 2. No início dos anos 90, um conjunto de mudanças foi integrado ao protocolo para produzir a versão 3, o que aumentou o seu desempenho e forneceu melhor suporte para arquivos grandes. As primeiras implementações da versão 4 estão agora disponíveis. A versão 4 inclui muitos novos aprimoramentos, que descrevemos a seguir.

Como os clientes NFS versão 2 não podem supor que uma operação de gravação esteja completa até receberem uma confirmação do servidor, servidores versão 2 têm de gravar cada bloco modificado em disco antes de responder, para evitar discrepâncias no caso de uma pane no sistema. Essa restrição introduz um atraso significativo nas gravações NFS porque blocos modificados normalmente seriam gravados apenas no cache de buffer em memória.

A versão 3 do NFS elimina esse gargalo com um esquema de coerência que permite gravações assíncronas. Ele também atualiza vários outros aspectos do protocolo que causaram problemas de desempenho. O resultado de rede é que a versão 3 do NFS é bem mais rápida do que a versão 2. O software da versão 3 sempre consegue interoperar com a versão 2, embora ele simplesmente recorra ao uso do protocolo anterior.

A versão 4 do NFS está se tornando mais estável e é distribuída em algumas versões do Linux. Ela requer um kernel 2.6.1 ou posterior e precisa ser manualmente ativada no kernel. Os principais aprimoramentos são:

- Compatibilidade e cooperação com firewalls e dispositivos NAT
- Integração do bloqueio e os protocolos de montagem com o protocolo NFS básico
- Operação com informações de estado
- Forte segurança integrada
- Suporte à replicação e migração
- Suporte tanto a clientes UNIX como a clientes Windows
- Listas de controle de acesso (*access control list* – ACLs)
- Suporte a nomes de arquivos Unicode
- Bom desempenho mesmo em conexões de baixa largura de banda

Acesse nfs.sourceforge.net para informações atuais sobre o estado do NFSv4 no Linux e referências para as distribuições de software mais recentes.

Como o NFSv4 continua em desenvolvimento, não iremos discuti-lo em detalhes neste capítulo. Mas fique de olho nele; muitos recursos planejados resolvem deficiências de longa data do NFS. Esperamos que a versão 4 cumpra com suas especificações bastante promissoras.

Escolha de transporte

O NFS executa sobre o protocolo RPC (Remote Procedure Call) da Sun, que define uma maneira independente do sistema para os processos se comunicarem através de uma rede. Um efeito colateral vantajoso dessa arquitetura é que ela permite o uso do UDP ou do TCP como o protocolo subjacente de transporte.

O NFS utilizava originalmente o UDP, pois esse era o que apresentava melhor desempenho nas LANs e computadores dos anos 80. Embora o NFS faça a sua própria remontagem de seqüência de pacotes e verificação de erros, faltam no UDP e no NFS os algoritmos de controle de congestionamento que são essenciais para um bom desempenho em uma grande rede IP.

Para reparar esses potenciais problemas, todos os sistemas modernos agora permitem utilizar o TCP em vez do UDP como o transporte para o NFS. Essa opção foi explorada pela primeira vez como uma maneira de ajudar o NFS a trabalhar através de roteadores e da Internet. Entretanto, o consenso atual parece ser o de que o TCP normalmente é a melhor opção para tráfego NFS local também. Ao longo do tempo, a maioria das razões originais para preferir UDP ao TCP foi se evaporando sob a luz quente das CPUs rápidas, das memórias baratas e das placas de rede mais inteligentes. O Linux suporta o serviço NFS para o TCP desde o kernel 2.4.

Grande parte dos servidores que suporta o TCP geralmente aceitará conexões em ambos os protocolos de transporte, de modo que a decisão entre TCP e UDP é tomada pelo cliente. O cliente especifica sua preferência como uma opção para o comando **mount** (para montagens manuais) ou em um arquivo de configuração como **/etc/fstab**.

Bloqueio de arquivos

O bloqueio de arquivo (como implementado pelas chamadas de sistema **flock**, **lockf** e/ou **fcntl**) tem sido uma questão difícil nos sistemas UNIX há muito tempo. Em sistemas de arquivos locais, é fato conhecido que ele não funciona de maneira perfeita. No contexto do NFS, o terreno é ainda mais instável. Por projeto, os servidores NFS são sem estado: eles não têm a mínima idéia de quais máquinas estão utilizando um dado arquivo. Entretanto, essas informações são necessárias para implementar o bloqueio de arquivos. O que fazer?

A resposta tradicional tem sido implementar o bloqueio de arquivos separadamente do NFS. A maioria dos sistemas fornece dois daemons, **lockd** e **statd**, que tentam ser bem-sucedidos. Infelizmente, a tarefa é difícil por uma série de razões sutis, e o bloqueio de arquivos NFS geralmente tende a ser frágil.

Cotas de disco

O acesso a informações de cotas em discos remotos pode ser fornecido por um servidor "fora da banda" (*out-of-band server*), **rquotad**. Os servidores NFS impõem cotas de disco se elas estiverem ativadas no sistema de arquivos subjacente, mas os usuários não conseguem visualizar as informações sobre as cotas a menos que **rquotad** esteja em execução no servidor remoto.

Consideramos as cotas de disco como amplamente obsoletas; mas algumas empresas ainda dependem delas para evitar que os usuários poluam todo o espaço em disco disponível. Se fornecer suporte a uma dessas empresas, você poderá consultar a documentação sobre cotas para as distribuições Linux apropriadas. Não discutimos **rquotad** em detalhes.

Cookies e montagem sem estado

Um cliente tem de montar explicitamente um sistema de arquivos NFS antes de usá-lo, da mesma maneira que um cliente tem de montar um sistema de arquivos em um disco local. Entretanto, devido ao fato do NFS ser sem estado, o servidor não controla quais clientes montaram cada sistema de arquivos. Em vez disso, o servidor apenas expõe um "cookie" secreto no final de uma negociação de montagem bem-sucedida. O cookie identifica o diretório montado para o servidor NFS, e assim fornece uma maneira para o cliente acessar seu conteúdo.

Desmontar e remontar um sistema de arquivos no servidor normalmente altera seu cookie. Como caso especial, os cookies persistem após uma reinicialização para que um servidor que trave possa retornar ao seu estado anterior. Mas não tente inicializar em modo monousuário, mexer nos sistemas de arquivos, e depois inicializar novamente; esse procedimento invalida os cookies e incapacita os clientes de acessar os sistemas de arquivos que eles montaram até reinicializarem ou remontarem.

Assim que um cliente tiver um cookie mágico, ele utiliza o protocolo RPC para solicitar operações de sistemas de arquivos, como, por exemplo, criar um arquivo ou ler um bloco de dados. Como o NFS é um protocolo sem informações de estado, o cliente é responsável por assegurar que os servidores reconheçam requisições de gravações antes de excluir sua própria cópia dos dados a serem gravados.

Convenções de atribuição de nome para sistemas de arquivos

É mais fácil gerenciar o NFS se seu esquema de atribuição de nomes para montagens incluir o nome de cada servidor remoto (por exemplo, **/anchor/tools** para um sistema de arquivos que reside em anchor). Esses nomes são úteis porque deixam que os usuários convertam anúncios como "anchor estará fora do ar sábado para atualização" em informações úteis como "sábado não poderei utilizar **/anchor/tools/TeX** para terminar minha tese, portanto, vou à praia".

Infelizmente, esse esquema requer que o diretório **/anchor** exista no diretório-raiz de todas as máquinas clientes. Se um cliente obtiver sistemas de arquivos de vários outros hosts, a raiz pode se tornar poluída. Considere a possibilidade de fornecer uma hierarquia mais profunda (por exemplo, **/home/anchor**, **/home/rastadon** etc.). Recomendamos a implementação desse esquema com um daemon automounter como descrito a partir da página 346.

Segurança e NFS

O NFS é uma maneira conveniente para acessar arquivos em uma rede e, portanto, ele possui grande potencial para provocar problemas de segurança. Sob diversos aspectos, o NFS versão 3 é um informante de tudo o que há ou houve de errado com a segurança do UNIX e Linux. Na essência, o protocolo foi originalmente projetado sem nenhuma preocupação com relação à segurança; e a conveniência tem seu preço. Felizmente, o Linux suporta uma série de recursos que reduzem e isolam os problemas de segurança dos quais o NFS tradicionalmente sofria.

*Consulte a página 341 para informações adicionais sobre o arquivo **exports**.*

O acesso aos volumes NFS é concedido por um arquivo chamado **/etc/exports** que enumera os nomes de hosts (ou endereços IP) de sistemas que devem ter acesso aos sistemas de arquivos de um servidor. Infelizmente, essa é uma forma frágil de implementação de segurança, pois o servidor confia nos clientes para informá-lo sobre quem eles são. É fácil fazer os clientes mentirem sobre suas identidades, de modo que esse mecanismo não é totalmente confiável. Não obstante, você deve exportar sistemas de arquivos somente para clientes nos quais confia e deve sempre verificar se você não exportou acidentalmente sistemas de arquivos para o mundo inteiro.

O acesso a portas NFS sempre deve ser rigidamente controlado. Felizmente, todas as versões do Linux incluem um firewall que pode tratar essa tarefa.

Como acontece nos sistemas de arquivos locais, o controle de acesso em nível de arquivo em sistemas de arquivos NFS é controlado de acordo com o UID, o GID e as permissões do arquivo. Mas, mais uma vez, o servidor NFS confia no cliente para informá-lo sobre quem está acessando arquivos. Se mary e bob compartilham o mesmo UID em dois clientes distintos, eles terão acesso aos arquivos NFS um do outro. Além disso, os usuários que possuem acesso de root em um sistema podem mudar para o UID que bem entenderem; o servidor inocentemente dará acesso aos arquivos correspondentes. Por essas razões, recomendamos o emprego de UIDs globalmente exclusivos e a opção `root_squash` descrita na próxima seção.

Uma instituição educacional bastante grande que conhecemos cometeu o equívoco de não utilizar `root_squash`. Como resultado, cinco grandes servidores e 60 desktops foram comprometidos. Levou um longo feriado prolongado para conter o incidente e reconstruir as máquinas.

Consulte a página 484 para informações adicionais sobre firewalls de rede.

Caso seu site já tenha instalado um firewall de rede, uma boa idéia é bloquear acesso às portas 2049 TCP e UDP, que são utilizadas pelo NFS. Você também deve bloquear acesso ao daemon **portmap**, que normalmente ouve nas portas TCP e UDP 111. Fica implícito nessas precauções, mas vale a pena mencionar explicitamente que sistemas de arquivos NFS não devem ser exportados para máquinas não-locais (com exceção do WebNFS) ou exportados para a Internet aberta.

Acesso de root e a conta nobody

Embora os usuários geralmente devam receber privilégios idênticos seja para onde forem, é tradicional impedir que root seja executado irrestritamente em sistemas de arquivos montados com NFS. Por padrão, o servidor Linux NFS intercepta requisições de entrada feitas em nome de UID 0 e as altera para que pareçam provir de algum outro usuário. Essa modificação é chamada de "squashing root" (encurralando root). A conta root não é fechada inteiramente, contudo ela é limitada às habilidades de um usuário comum.

Uma conta substituta chamada "nobody" é definida especificamente para ser o pseudo-usuário como aquele com que um root remoto se mascara em um servidor NFS. O UID tradicional de nobody é 65534 (o equivalente em complemento de dois do UID -2).[1] Você pode alterar as associações UID e GID padrão para root por meio das opções de exportação `anonuid` e `anongid`. Pode também usar a opção `all_squash` para associar os UIDs de todos os clientes ao mesmo UID no servidor. Essa configuração elimina todas as distinções entre usuários e cria uma espécie de sistema de arquivos de acesso público.

---

1. Embora o servidor NFS do Red Hat assuma UID -2 por padrão, a conta nobody no arquivo **passwd** utiliza UID 99. Você pode deixar as coisas como estão, acrescentar uma entrada **passwd** para UID -2 ou então alterar `anonuid` e `anongid` para 99 se quiser. As demais distribuições utilizam UID -2 para as contas nobody e nogroup no arquivo **passwd**, como esperado.

Na outra ponta do espectro, a opção `no_root_squash` desativa a associação UID para root. Essa opção algumas vezes é necessária para suportar as máquinas clientes sem disco ou para software que exija acesso de root ao sistema de arquivos. Geralmente é uma má idéia ativar esse recurso porque ele permite que usuários com privilégios de root em um cliente modifiquem os arquivos que normalmente estão protegidos. Contudo, essa opção está disponível.

A intenção por trás dessas precauções é boa, mas seu valor final não é tão bom quanto possa parecer. Lembre-se de que root em um cliente NFS pode executar **su** para qualquer UID que ele quiser; portanto, os arquivos de usuário jamais estarão realmente protegidos. Logins de sistema como "bin" e "sys" não são mapeados por UID, quaisquer arquivos que eles possuam, como ocasionais binários de sistema ou aplicativos de terceiros, podem ser vulneráveis a ataques. O único efeito real da associação UID é o de impedir acesso a arquivos cujo proprietário seja root e que não possam ser lidos ou gravados por todo o mundo.

16.2 NFS NO LADO SERVIDOR

Diz-se que um servidor "exporta" um diretório quando ele torna esse diretório disponível para uso de outras máquinas.

No NFS versão 3, o processo utilizado por clientes para montar um sistema de arquivos (isto é, para tomar conhecimento do seu cookie secreto) é completamente separado do processo utilizado para acessar arquivos. As operações empregam protocolos distintos e as requisições são atendidas por diferentes daemons: **mountd** para requisições de montagem e **nfsd** para serviço de arquivos em si. Em alguns sistemas, esses daemons são chamados **rpc.nfsd** e **rpc.mountd** como um lembrete de que eles contam com o RPC como um protocolo subjacente (e, portanto, exigem que o **portmap** esteja em execução; consulte a página 613). Omitimos o prefixo **rpc** para fins de legibilidade.

Em um servidor NFS, tanto **mountd** como **nfsd** devem iniciar quando o sistema inicializa, e ambos devem permanecer em execução enquanto o sistema estiver ativo. Em geral, os scripts de inicialização de sistema executam os daemons automaticamente se você tiver outras exportações configuradas. Os nomes dos scripts de inicialização de servidor NFS para cada distribuição são mostrados na Tabela 16.1.

Tabela 16.1 Scripts de inicialização de servidor NFS

Distribuição	Caminhos para os scripts de inicialização
Red Hat Enterprise	**/etc/rc.d/init.d/nfs**
Fedora	**/etc/rc.d/init.d/nfs**
SUSE	**/etc/init.d/nfsboot**[a]
Debian	**/etc/init.d/nfs-kernel-server** **/etc/init.d/nfs-common**
Ubuntu	**/etc/init.d/nfs-kernel-server** **/etc/init.d/nfs-common**

a. **/etc/init.d/nfs** monta os sistemas de arquivos NFS de clientes no SUSE.

mountd e **nfsd** compartilham um único banco de dados de controle de acesso que informa quais sistemas de arquivos devem ser exportados e quais clientes podem montá-los. A cópia operacional desse banco de dados normalmente é mantida em um arquivo chamado **/usr/lib/nfs/xtab** além das tabelas internas ao kernel. Como o propósito do **xtab** e do **sharetab** é não serem legíveis por humanos, utilize um comando auxiliar — **exportfs** — para adicionar e modificar as entradas. Para remover entradas da tabela de exportação, utilize **exportfs -u**.

Na maioria dos sistemas, **/etc/exports** é a lista canônica legível por humanos dos diretórios exportados. Por padrão, todos os sistemas de arquivos em **/etc/exports** são exportados no momento da inicialização. Você pode exportar manualmente todos os sistemas de arquivos listados em **/etc/exports** utilizando **exportfs -a**, que deve ser executado após o arquivo **exportfs** sofrer alterações. Você também pode exportar sistemas de arquivos especificando o cliente, caminho e as opções diretamente na linha de comando de **exportfs**.

O NFS lida com a camada lógica do sistema de arquivos. Qualquer diretório pode ser exportado; ele não precisa ser um ponto de montagem ou a raiz do sistema de arquivos físico. Entretanto, por segurança, o NFS realmente presta atenção aos limites entre os sistemas de arquivos e requer efetivamente que cada dispositivo seja exportado separadamente. Por exemplo, em uma máquina que utiliza uma partição **/users**, você poderia exportar o diretório-raiz sem exportar **/users**.

Os clientes normalmente têm permissão para montar subdiretórios de um diretório exportado se quiserem, embora o protocolo não exija esse recurso. Se, por exemplo, um servidor exporta **/chimchim/users**, um cliente poderia montar somente **/chimchim/users/joe** e ignorar o restante do diretório **users**. A maioria das versões do UNIX não permite exportar subdiretórios de um diretório exportado com opções diferentes, mas o Linux permite.

O arquivo exports

O arquivo **/etc/exports** enumera os sistemas de arquivos exportados por meio do NFS e os clientes que podem acessar cada um deles. Um espaço em branco separa o sistema de arquivos da lista de clientes e cada cliente é seguido imediatamente por uma lista entre parênteses das opções separadas por vírgulas. As linhas podem ser continuadas com uma barra invertida.

Eis o aspecto deste formato:

```
/home/boggs        inura(rw,no_root_squash) lappie(rw)
/usr/share/man     *.toadranch.com(ro)
```

Não há nenhum modo de listar vários clientes em um único conjunto de opções, embora algumas especificações de clientes realmente referenciem vários hosts. A Tabela 16.2 lista os quatro tipos de especificações do cliente que podem aparecer em um arquivo **exports** do Red Hat.

Tabela 16.2 Especificações de cliente no arquivo/etc/exports do Red Hat

Tipo	Sintaxe	Significado
Nome do host	*nomeDoHost*	Hosts individuais
Grupo de redes	*@nomeDoGrupo*	Grupos de rede NIS; consulte a página 359 para detalhes
Curingas	* e ?	FQDNs[a] com curingas; "*" não irá procurar um ponto
Redes IP	*endereçoIP/máscara*	Especificações tipo CIDR (por exemplo, 128.138.92.128/25)

a. Nomes de domínio totalmente qualificados

A Tabela 16.3 mostra as opções de exportação entendidas pelo Red Hat mais comumente utilizadas.

Tabela 16.3 Opções comuns de exportação para o Red Hat

Opção	Descrição
`ro`	Exporta somente leitura
`rw`	Exporta para leitura e gravação (o padrão)
`rw=`*lista*	Exporta na maior parte somente para leitura. *lista* enumera os hosts que têm permissão para montar para gravação; todos os demais têm de montar somente para leitura
`root_squash`	Mapeia ("encurrala") UID 0 e GID 0 para os valores especificados por `anonuid` e `anongid`; esse é o padrão.
`no_root_squash`	Permite acesso normal por parte de root — perigoso
`all_squash`	Associa todos os UIDs e GIDs a suas versões anônimas; útil para suportar PCs e hosts monousuários não-confiáveis.
`anonuid=`*xxx*	Especifica o UID em que roots remotos devem ser encurralados
`anongid=`*xxx*	Especifica o GID em que roots remotos devem ser encurralados
`secure`	Requer que o acesso remoto se origine de uma porta privilegiada
`insecure`	Permite acesso remoto de qualquer porta
`noaccess`	Impede o acesso a esse diretório e seus subdiretórios (usado com exportações aninhadas)
`wdelay`	Retarda gravações na esperança de mesclar múltiplas atualizações
`no_wdelay`	Grava dados no disco assim que possível
`async`	Faz o servidor responder a requisições de gravação antes de uma gravação no disco real
`nohide`	Revela sistemas de arquivos montados dentro das árvores de arquivo exportadas
`hide`	O oposto de `nohide`
`subtree_check`	Verifica se cada arquivo solicitado está dentro uma subárvore exportada
`no_subtree_check`	Verifica apenas se requisições de arquivo referem-se a um sistema de arquivos exportado
`secure_locks`	Requer autorização para todas as requisições de bloqueio
`insecure_locks`	Especifica um critério menos rigoroso de bloqueio (suporta clientes mais antigos)
`auth_nlm`	Sinônimo de `secure_locks`
`no_auth_nlm`	Sinônimo de `insecure_locks`

O servidor NFS do Linux tem um recurso incomum de permitir que subdiretórios dos diretórios exportados sejam exportados com diferentes opções. Utilize a opção `noaccess` para "não exportar" subdiretórios que você preferiria não compartilhar.

Por exemplo, a configuração

```
/home           *.toadranch.com(rw)
/home/boggs     (noaccess)
```

permite que hosts no domínio toadranch.com acessem todo o conteúdo de **/home** no ponto de montagem, exceto **/home/boggs**. A ausência de um nome de cliente na segunda linha significa que a opção se aplica a todos os hosts; talvez seja um pouco mais seguro proceder desta maneira.

A opção `subtree_check` (o padrão) verifica se cada arquivo acessado pelo cliente reside dentro de um subdiretório exportado. Se desativar essa opção, somente o fato do arquivo estar dentro de um sistema de arquivos exportado é verificado. A verificação de subárvore pode ocasionalmente causar problemas quando um arquivo requisitado é renomeado enquanto o cliente tem o arquivo aberto. Se antecipar várias situações como essas, pense em configurar `no_subtree_check`.

A opção `secure_locks` requer autorização e autenticação a fim de que os arquivos sejam bloqueados. Alguns clientes NFS não enviam credenciais com solicitações de bloqueio e não funcionam com `secure_locks`. Nesse caso, você só seria capaz de bloquear arquivos mundialmente legíveis. Substituir esses clientes por aqueles que suportam credenciais corretamente é a melhor solução. Entretanto, você pode especificar a opção `insecure_locks` como um substituto temporário.

Não se esqueça de executar **exportfs -a** depois de atualizar o arquivo **exports** para que suas alterações tenham efeito.

nfsd: servindo arquivos

Assim que a requisição de montagem de um cliente tiver sido validada por **mountd**, o cliente poderá solicitar várias operações de sistemas de arquivos. Essas solicitações são manipuladas no servidor por **nfsd**, o daemon de operações NFS.[2] **nfsd** não precisa ser executado em um cliente NFS a menos que o cliente exporte sistemas de arquivos próprios.

nfsd recebe um argumento numérico que especifica quantas threads de servidor ele deve gerar. Selecionar o número apropriado de **nfsd**s é importante e, infelizmente, envolve certa dose de magia. Se o número for muito baixo ou muito alto, o desempenho do NFS pode sofrer com isso.

Geralmente, 8 **nfsd** threads são adequados para um servidor pouco utilizado e o número deles é pequeno o suficiente para não causar problemas de desempenho. Em um servidor em um ambiente de produção, algo entre 12 e 20 é um bom número. Se observar que **ps** mostra os **nfsd**s no estado D a maior parte do tempo e se uma CPU ociosa estiver disponível, considere aumentar o número de threads. Se achar que a média da carga (conforme relatado por **uptime**) estiver aumentando à medida que você acrescenta **nfsd**s, você acabou indo longe demais; diminua um pouco a partir desse limiar. Você também deve executar **nfsstat** regularmente para verificar problemas de desempenho que poderiam estar associados ao número de threads **nfsd**. Consulte a página 345 para informações adicionais sobre **nfsstat**.

Em um servidor NFS com muitos clientes UDP, sockets UDP poderão causar sobrecarga nas tabelas (overflow) caso as requisições cheguem enquanto todas as threads **nfsd** já estiverem em uso. Você pode monitorar o número de overflows com **netstat -s**. Adicione mais **nfsd**s até que os overflows de socket UDP caiam para zero. Overflows indicam uma falta grave de daemons de servidor, portanto você provavelmente deve adicionar um número maior do que essa métrica indicaria.

É possível mudar o número de processos **nfsd** editando o script de inicialização apropriado em **/etc/init.d** ou especificando o número na linha de comando ao iniciar **nfsd** manualmente. Veja na Tabela 16.1, na página 348, o nome do script a editar.

16.3 NFS NO LADO CLIENTE

Os sistemas de arquivos são montados quase da mesma maneira que os sistemas de arquivos de disco locais. O comando **mount** entende a notação *nomeDoHost:diretório* como o caminho do *diretório* na máquina *nomeDoHost*. Da mesma forma como ocorre com sistemas de arquivos locais, **mount** associará o *diretório* remoto do *host* remoto a um diretório contido na árvore de arquivos local. Após a montagem, um sistema de arquivos montado com NFS é acessado da mesma maneira que um sistema de arquivos local. O comando **mount** e suas extensões NFS associadas representam as preocupações mais significativas para um administrador de sistemas de um cliente NFS.

Antes de um sistema de arquivos NFS poder ser montado, ele deve ser exportado apropriadamente (veja a página 348). Para verificar se um servidor exportou apropriadamente seus sistemas de arquivos sob a perspectiva do cliente, você pode usar o comando **showmount** do cliente:

```
$ showmount -e coyote
Export list for coyote:
/home/boggs inura.toadranch.com
```

2. Na realidade, **nfsd** normalmente é um programa muito simples que faz uma chamada de sistema sem retorno ao código do servidor NFS embutido no kernel.

Esse exemplo informa que o diretório **/home/boggs** no servidor coyote foi exportado para o sistema cliente inura.toadranch.com. A saída gerada por **showmount** deveria ser a primeira coisa a checar caso uma montagem nfs não esteja funcionando e se você já tiver verificado que os sistemas de arquivos foram exportados apropriadamente no servidor via **exportfs**. (Talvez você simplesmente tenha se esquecido de executar **exportfs -a** depois de atualizar o arquivo **exports**.)

Se o diretório tiver sido exportado apropriadamente no servidor, mas **showmount** retornar um erro ou uma lista vazia, talvez você tenha de verificar cuidadosamente que todos os processos necessários estejam sendo executados no servidor (**portmap**, **mountd**, **nfsd**, **statd** e **lockd**), que os arquivos **hosts.allow** e **hosts.deny** permitam acesso a esses daemons e que você se encontre no sistema cliente correto.

Para efetivamente montar o sistema de arquivos, você usaria um comando como indicado a seguir:

```
# mount -o rw,hard,intr,bg coyote:/home/boggs /coyote/home/boggs
```

As opções após **-o** especificam que o sistema de arquivos deve ser montado para leitura/gravação, que é possível interromper as operações e que novas tentativas devem ser feitas em segundo plano. Esses flags são padrões; outros flags comuns estão listados na Tabela 16.4.

Tabela 16.4 Flags/opções de montagem do NFS

Flag	Descrição
`rw`	Monta o sistema de arquivos para leitura-gravação (deve ser exportado assim)
`ro`	Monta o sistema de arquivos somente leitura
`bg`	Se a montagem falhar (o servidor não responder), continue tentando em segundo plano e continue com outras requisições de montagem
`hard`	Se um servidor sair do ar, as operações que tentam acessá-lo serão bloqueadas até que o servidor volte a entrar em operação
`soft`	Se um servidor sair do ar, as operações que tentarem acessá-lo falharão e retornarão um erro; esse recurso é útil para evitar que processos fiquem "pendurados" em montagens não-essenciais
`intr`	Permite que usuários interrompam operações bloqueadas (e retornem um erro)
`nointr`	Não permite que o usuário provoque uma interrupção
`retrans=n`	Especifica o número de vezes para repetição de uma requisição antes de retornar um erro em um sistema de arquivos montado com a opção soft
`timeo=n`	Configura o período de expiração (em décimos de segundo) para as requisições
`rsize=n`	Configura o tamanho do buffer de leitura para *n* bytes
`wsize=n`	Configura o buffer de gravação para *n* bytes
`nfsvers=n`	Seleciona as versões 2 ou 3 do protocolo NFS (normalmente de forma automática)
`tcp`	Seleciona o transporte via TCP; UDP é o padrão
`async`	Faz com que o servidor responda a requisições de gravação antes de uma gravação no disco real

Sistemas de arquivos montados com a opção `hard` (o padrão) podem fazer com que processos fiquem pendurados quando seus servidores ficarem inativos. Esse comportamento é particularmente incômodo quando os processos em questão são daemons padrão, de modo que não recomendamos servir binários de sistema críticos via NFS. Geralmente, o uso das opções `soft` e `intr` reduz o número de problemas relacionadas ao NFS. Entretanto, essas opções podem ter seus efeitos colaterais indesejáveis, como abortar uma simulação de 20 horas depois de ela ter executado por 18 horas simplesmente devido a um problema menor e passageiro de rede.[3] Soluções de automontagem como autofs, discutidas posteriormente neste capítulo, também fornecem algumas curas para doenças de montagem.

Os tamanhos de buffer de escrita e de gravação se aplicam tanto a montagens UDP quanto TCP, mas os valores ótimos diferem. Pelo fato de se poder confiar no TCP para transferência de dados eficiente, os valores devem ser maiores; 32 K é uma boa opção. Para o UDP, os valores indicados quando o servidor e o cliente se encontram na mesma rede é de 8 K.[4] O padrão é 1 K, muito embora o manual recomende aumentá-lo para 8 K para fins de melhora de desempenho.

---

3. Jeff Forys, um de nossos revisores técnicos, salientou: "A maioria das montagens deveria usar `hard`, `intr` e `bg`, pois essas opções preservam melhor os objetivos de projeto originais do NFS (confiabilidade e ausência de estados). `soft` é abominável, um horrível hack satânico! Se o usuário quiser interromper, tudo bem. Caso contrário, aguarde o servidor e tudo por fim estará bem novamente sem nenhuma perda de dados.".
4. Se você usar **iptables**, talvez tenha de adicionar uma regra para aceitar fragmentos, pois 8 K está acima do MTU para Ethernet. Aceitar fragmentos pode lhe tornar mais vulnerável a um ataque de recusa de serviços.

Nos kernels herdados do Linux 2.2 e 2.4, o tamanho padrão da fila de entrada é 64 KB. Com oito threads **nfsd** em execução no servidor NFS, somente uma requisição pode estar pendente em cada instância de **nfsd** antes de as requisições começarem a ser descartadas. Portanto, talvez se deva considerar aumentar o tamanho da fila de recebimento somente para o **nfsd**, retornando-a ao valor padrão após executar **nfsd** para que outros processos não sejam afetados negativamente pela modificação. Você pode alterar o tamanho da fila de entrada nos scripts de inicialização de sistema utilizando **procfs**.

Esse exemplo configura o tamanho da fila em 256 K, o que é um padrão razoável:

```
rmem_default=`cat /proc/sys/net/core/rmem_default`
rmem_max=`cat /proc/sys/net/core/rmem_max`
echo 262144 > /proc/sys/net/core/rmem_default
echo 262144 > /proc/sys/net/core/rmem_max
```

Execute ou reinicialize **rpc.nfsd**, depois retorne a configuração para seus valores originais:

```
echo $rmem_default > /proc/sys/net/core/rmem_default
echo $rmem_max > /proc/sys/net/core/rmem_max
```

Você pode testar a montagem com **df** assim como testaria um sistema de arquivos local:

```
$ df /coyote/home/boggs
Filesystem          1k-blocks      Used  Available   Use%   Mounted on
coyote:/home/boggs  17212156    1694128  14643692    11%    /coyote/home/boggs
```

Você desmonta partições NFS com o comando **umount**. Se o sistema de arquivos NFS estiver em uso ao tentar desmontá-lo, será gerado um erro como:

```
umount: /coyote/home/boggs: device is busy
```

Como outros sistemas de arquivos, um sistema de arquivos NFS não pode ser desmontado enquanto estiver em uso. Utilize **lsof** para localizar processos com arquivos abertos no sistema de arquivos; elimine-os ou, no caso de shells, mude os diretórios. Se todas essas medidas falharem ou se seu servidor estiver inativo, tente **umount -f** para forçar o sistema de arquivos a ser desmontado.

Montando sistemas de arquivos remotos em tempo de inicialização

*Consulte a página 346 para obter informações adicionais sobre **usermod**.*

Você pode usar o comando **mount** para estabelecer montagens de rede temporárias, mas deve listar montagens que fazem parte de uma configuração permanente do sistema em **/etc/fstab** para que elas sejam montadas automaticamente no momento da inicialização. Alternativamente, as montagens podem ser tratadas por um serviço de montagem automática como o autofs.

As entradas **fstab** a seguir montam os sistemas de arquivos **/home** e **/usr/local** a partir dos hosts coyote e inura:

```
# filesystem      mountpoint     fstype  flags                        dump   fsck
coyote:/home      /coyote/home   nfs     rw,bg,intr,hard,nodev,nosuid 0      0
inura:/usr/local  /usr/local     nfs     ro,bg,intr,soft,nodev,nosuid 0      0
```

*Consulte a página 89 para informações adicionais sobre o arquivo **fstab**.*

Ao adicionar entradas a **fstab**, certifique-se de criar os diretórios de ponto de montagem apropriados com **mkdir**. Você pode fazer com que suas alterações tenham efeito imediatamente (sem reinicializar) executando **mount -a -t nfs** para montar todos os sistemas de arquivos do tipo `nfs` em **fstab**.

O campo `flags` de **/etc/fstab** especifica as opções para montagens NFS; essas opções são as mesmas que você especificaria na linha de comando **mount**.

Restringindo exportações a portas inseguras

Clientes NFS são livres para usar qualquer porta de origem TCP ou UDP que eles quiserem ao se conectarem a um servidor NFS. Entretanto, os servidores Linux podem insistir que as requisições provenham de uma porta privilegiada (uma porta com numeração inferior a 1024) se o sistema de arquivos for exportado com a opção de exportação `secure`, que está ativa por padrão. No mundo dos PCs e desktops Linux, o emprego de portas privilegiadas fornece pouca segurança na verdade.

Os clientes NFS Linux adotam a abordagem tradicional (e ainda recomendada) de usar por padrão uma porta privilegiada, para impedir potencial para conflito. Para aceitar montagens de portas de origem não-privilegiadas, exporte o sistema de arquivos com a opção de exportação `insecure`.

16.4 NFSSTAT: FAZENDO DUMP DE ESTATÍSTICAS DO NFS

O comando **nfsstat** exibe várias estatísticas mantidas pelo sistema NFS. **nfsstat -s** exibe estatísticas para os processos do servidor NFS e **nfsstat -c** mostra as informações relacionadas a operações do lado cliente. Por exemplo:

```
$ nfsstat -s
Server rpc stats:
calls             badcalls       badauth        badclnt        xdrcall
24314112          311            9              302            0
Server nfs v2:
getattr           null           setattr        root           lookup         readlink
8470054    34%    58     0%      55199   0%     0      0%      1182897 4%     917    0%
read              wrcache        link           create         remove         rename
6602409    27%    0      0%      7452    0%     61544  0%      46712   0%     11537  0%
write             symlink        mkdir          rmdir          readdir        fsstat
7785789    32%    744    0%      3446    0%     2539   0%      13614   0%     69201  0%
```

Esse exemplo é de um servidor NFS relativamente sadio. Se mais de 3% das chamadas forem ruins, o problema possivelmente reside no seu servidor NFS ou na rede. Verifique a saída gerada por **netstat -s** para estatísticas de rede genéricas. Ela poderá revelar problemas com pacotes perdidos, remontagem de fragmentos ou overruns na fila de rede que afetarão o desempenho do NFS. Consulte a página 449 para informações adicionais sobre como depurar uma rede com **netstat**.

Executar **nfsstat** e **netstat** ocasionalmente e familiarizar-se com sua saída ajudará você a descobrir problemas relacionados ao NFS antes dos seus usuários.

16.5 SERVIDORES DE ARQUIVOS NFS DEDICADOS

Serviço de arquivos rápido e confiável é um dos elementos de maior importância de qualquer ambiente computacional de produção. Embora você possa certamente rodar o seu próprio servidor de arquivos em uma estação de trabalho Linux e um punhado de discos rígidos comuns, fazer isso normalmente não é a melhor solução em termos de desempenho e facilidade de administração (embora normalmente seja a mais barata).

Servidores de arquivos NFS dedicados já estão no mercado há mais de uma década. Eles oferecem uma infinidade de vantagens em potencial em relação à abordagem de desenvolvimento interno:

- São otimizados para serviços de arquivos e fornecem tipicamente o melhor desempenho NFS possível
- À medida que as exigências de armazenamento aumentam, eles podem ser ampliados de maneira gradual para suportar terabytes de armazenamento e centenas de usuários
- São mais confiáveis do que os computadores Linux graças a seus softwares simplificados, hardware redundante e uso de espelhamentos de discos
- Normalmente fornecem serviços de arquivos tanto para clientes Linux como para clientes Windows; alguns até mesmo contêm servidores FTP e Web completos
- Eles normalmente são mais fáceis de serem administrados do que os servidores de arquivos Linux
- Eles costumam incluir recursos para backup e verificação que são superiores àqueles encontrados em sistemas Linux comuns

Alguns dos nossos servidores NFS dedicados favoritos são fabricados pela Network Appliance, Inc. (www.netapp.com). Seus servidores abrangem uma gama que vai desde os muito pequenos até os muito grandes e seus preços são razoáveis. A EMC é uma outra empresa no mercado de servidores topo de linha. Eles oferecem bons produtos, mas esteja preparado para o choque do preço e prepare sua tolerância para chavões de marketing.[5] Servidores de arquivo (*storage area network* – SAN) agora também são muito populares. Eles diferem dos servidores de arquivos NFS dedicados pelo fato de que eles não conhecem nada sobre os sistemas de arquivos; eles simplesmente servem blocos de disco. Um servidor SAN, portanto, não tem o tempo extra gasto por um sistema operacional e fornece acesso de leitura/gravação muito rápido. Dito isso, na prática descobrimos que esses servidores ainda não estão prontos para uso em produção. A integração deles em um ambiente existente pode ser bem complexa e aparentemente eles exigem quantidades significativas de tempo de suporte.

5. Falando em chavões, um dos principais que você ouvirá neste contexto é "network attached storage" ("armazenamento anexável à rede"), também conhecido como NAS. É só uma maneira extravagante de dizer "servidor de arquivos".

16.6 MONTAGEM AUTOMÁTICA

Montar sistemas de arquivos um a um listando-os em **/etc/fstab** ou **/etc/vfstab** cria uma série de problemas em redes grandes.

Primeiramente, manter **/etc/fstab** em uma centena de máquinas pode ser tedioso. Cada uma pode ser ligeiramente diferente e, portanto, requerer atenção individual.

Em segundo lugar, se os sistemas de arquivos forem montados a partir de vários hosts diferentes, o caos se estabelecerá quando um desses servidores travar. Cada comando que acessar os pontos de montagem ficará pendurado.

Em terceiro lugar, quando um servidor trava, ele pode incapacitar usuários tornando indisponíveis partições importantes como **/usr/share/man**. Nessa situação, é melhor se uma cópia da partição puder ser montada temporariamente em um servidor de backup.

Um daemon de automontagem monta sistemas de arquivos quando são referidos e os desmonta quando eles não são mais necessários. Esse procedimento minimiza o número de pontos de montagem ativos e é em grande parte transparente para o usuário. A maioria dos automontadores também funciona com uma lista que você fornece dos sistemas de arquivos "replicados" (idênticos) de modo que a rede possa continuar a funcionar quando um servidor primário não está disponível.

Para implementar essa montagem e desmontagem nos bastidores, o automontador monta um driver de sistemas de arquivos virtual nos diretórios que você designou como as localizações para a ocorrência de montagem automática. No passado, o automontador fazia isso se apresentando como um servidor NFS, mas esse esquema padece de várias limitações significativas e raramente é encontrado nos sistemas contemporâneos. Na atualidade, é utilizado um driver de sistema de arquivos residente em kernel chamado **autofs**.

Em vez de espelhar um sistema de arquivos de verdade, um automontador "falsifica" uma hierarquia de sistema de arquivo de acordo com as especificações que você lista em seu arquivo de configuração. Quando um usuário fizer referência a um diretório dentro do sistema de arquivos virtual do automontador, o automontador interceptará a referência e montará o sistema de arquivos que o usuário está tentando alcançar.

automount: montando sistemas de arquivos sob demanda

A idéia de um automontador é originalmente da Sun. O automontador do Linux, chamado **automount**, foi desenvolvido para simular o automontador da Sun, embora seja uma implementação independente do conceito e seja diferente em uma série de aspectos.

automount é um processo em segundo plano que configura um único ponto de montagem para autofs, a porção kernel do automontador do Linux. O script de inicialização **/etc/init.d/autofs** analisa sintaticamente um arquivo mestre (normalmente **/etc/auto.master**) e executa **automount** para cada um dos pontos de montagem listados. É típico ver uma instância em execução de **automount** para cada ponto de montagem automático que foi configurado.

Será raro você precisar executar o comando **automount** diretamente, já que quase toda a administração do automontador é realizada por meio do script **/etc/init.d/autofs** (ou no raro caso do Red Hat e Fedora, **/etc/rc.d/init.d/autofs**).[6] Como a maioria dos scripts de inicialização, o script **autofs** aceita um único parâmetro na linha de comando que pode ser **start**, **stop**, **reload**, **restart** ou **status**. Sempre que forem feitas modificações na configuração do automontador, você terá de executar **autofs reload** para que as alterações entrem em efeito. **autofs status** lhe dá o estado das automontagens existentes.

O arquivo **auto.master** associa um ponto de montagem a um "mapa". Um mapa converte o nome de diretório acessado — conhecido como a "chave" — em uma linha de comando que **mount** pode usar para realizar a montagem real. Um mapa pode ser um arquivo de texto, um programa executável ou um banco de dados NIS ou LDAP.

Quando um usuário faz referência a um diretório que foi montado com o módulo de sistema de arquivos de kernel autofs, ele notifica o processo **automount** sobre o acesso. O processo de **automount** descobre qual sistema de arquivos ele deve montar consultando o arquivo ou programa de mapa relevante. Ele então realiza a montagem antes de retornar o controle ao usuário que desencadeou a pesquisa.

É possível ver os sistemas de arquivos autofs e os processos **automount** aos quais eles estão anexados executando-se os comandos **mount** e **ps**:

```
$ mount
/dev/hda3 on / type ext2 (rw)
proc on /proc type proc (rw)
/dev/hda1 on /boot type ext2 (rw)
automount(pid8359) on /misc type autofs  // automontador sistemaDeArquivos
    (rw,fd=5,pgrp=8359,minproto=2,maxproto=4)
```

6. Não confunda o script **autofs** com o sistema de arquivos autofs. A relação entre eles é que o script informa ao kernel como configurar o sistema de arquivos.

```
automount(pid8372) on /net type autofs   // automontador sistemaDeArquivos
    (rw,fd=5,pgrp=8372,minproto=2,maxproto=4)
$ ps auxw | grep automount
root      8359  0.0  1.0  1360  652 ?        S    Dec27   0:00 /usr/sbin/automount /misc file
    /etc/auto.misc
root      8372  0.0  1.0  1360  652 ?        S    Dec27   0:00 /usr/sbin/automount /net program
    /etc/auto.net
```

Aqui podemos observar dois sistemas de arquivos autofs montados em **/misc** e **/net**. Esses sistemas de arquivos virtuais são anexados aos processos **automount** com PIDs 8359 e 8372, respectivamente. Os comandos **automount** executados pelo script **/etc/init.d/autofs** podem ser vistos na saída gerada por **ps**. **auto.misc** é um arquivo de mapas comum e **auto.net** é um programa executável. Esses mapas são descritos com mais detalhes a seguir.

O arquivo mestre

O arquivo **etc/automo.master** lista os diretórios que deveriam ter sistemas de arquivos autofs montados neles e associa um mapa a cada diretório. Além de fornecer o diretório-raiz para o mapa e o nome do mapa, você também pode especificar opções adicionais no formato "**-o**" utilizado pelo comando **mount**. Essas opções se aplicam a cada entrada do mapa. As convenções do Linux diferem daquelas da Sun em que as opções do arquivo mestre são cumulativas com aquelas do mapa; ambos os conjuntos de opções são transmitidos para o **mount**.

Um arquivo mestre simples que faz uso do arquivo de mapas mostrado na próxima seção ficaria parecido com este:

```
# Dir      Mapa              Opções
/chimchim  /etc/auto.chim    -secure,hard,bg,intr
```

O arquivo mestre pode ser substituído ou aumentado por uma versão compartilhada via NIS. A fonte de informações sobre a automontagem do sistema é especificada pelo campo `automount` em **/etc/nsswitch.conf**. Consulte *Priorizando fontes de informações administrativas* na página 358 para informações adicionais sobre o arquivo **nsswitch.conf**.

Arquivos de mapas

Os arquivos de mapas (conhecidos como "mapas indiretos" em outros sistemas) montam automaticamente diversos sistemas de arquivos sob um diretório comum. O caminho do diretório é especificado no arquivo mestre, não no próprio mapa. Por exemplo, um mapa indireto para sistemas de arquivos montados sob **/chimchim** (correspondente ao exemplo acima) poderia parecer isto:

```
users      chimchim:/chimchim/users
devel      -soft,protonfsproto=udp 3 chimchim:/chimchim/devel
info       -ro chimchim:/chimchim/info
```

Nessa configuração, os caminhos no host chimchim e no host local são idênticos. Entretanto, essa correspondência não é necessária.

Mapas executáveis

Se um arquivo de mapas indireto é executável, presume-se que é um script ou programa que gera dinamicamente informações de automontagem. Em vez de ler o mapa como um arquivo de texto, o automontador o executará com um argumento (a "chave") que indica o subdiretório que um usuário tentou acessar. O script é responsável por imprimir uma entrada de mapa apropriada; se a chave especificada não for válida, o script pode simplesmente sair sem imprimir nada.

Esse poderoso recurso compensa as muitas deficiências na estranha configuração de sistema do **automounter**. De fato, ele permite que você defina facilmente um arquivo de configuração de automontagem válido para um site em um formato de sua própria preferência. Você pode escrever um script Perl simples para decodificar a configuração global em cada máquina. Alguns sistemas vêm acompanhados com um prático mapa executável **/etc/auto.net** que utiliza um nome de host como uma chave e monta todos os sistemas de arquivos exportados neste host.

O automontador tem uma característica confusa que merece ser mencionada aqui: ao listar o conteúdo de um diretório pai do sistema de arquivos automontado, o diretório parece vazio independentemente de quantos sistemas de arquivos foram automontados aí. Você não pode pesquisar as automontagens em um navegador para sistema de arquivos com interface gráfica. Um exemplo:

```
$ ls /portal
$ ls /portal/photos
art_class_2004     florissant_1003              rmnp03
blizzard2003       frozen_dead_guy_Oct2004      rmnp_030806
boston021130       greenville.021129            steamboat2002
```

O sistema de arquivos **photos** está vivo e bem e foi automontado sob **/portal**. Ele é acessível por meio do seu nome de caminho completo. Entretanto, uma revisão do diretório **/portal** não revela sua existência. Se tivesse montado esse sistema de arquivos por meio do arquivo **/etc/fstab** ou com um comando **mount** manual, ele iria se comportar como qualquer outro diretório e seria visível como um membro do diretório pai.

Uma maneira de contornar esse problema de navegação é criar um diretório sombra que contém links simbólicos para os pontos de automontagem. Por exemplo, se **/automounts/photos** for um link simbólico para **/portal/photos**, você pode usar **ls /automounts** para descobrir que **photos** é um diretório automontado. As referências a **/automounts/photos** ainda são roteadas pelo automontador e funcionam de modo correto. Infelizmente, esses links simbólicos exigem manutenção e podem sair de sincronia com as automontagens reais, a menos que eles sejam periodicamente reconstruídos por um script.

16.7 Leitura recomendada

CALLAGHAN, Brent. *NFS Illustrated*. Reading, MA: Addison-Wesley, 1999.

STERN, Hal e EISLER, Mike e LABIAGA, Ricardo. *Managing NFS and NIS. 2 ed.* Sebastopol, CA: O'Reilly Media, 2001.

A Tabela 16.5 lista as diversas RFCs para o protocolo NFS.

Tabela 16.5 RFCs relacionadas com o NFS

RFC	Título	Autor	Data
1094	Network File System Protocol Specification	Sun Microsystems	Março de 1989
1813	NFS Version 3 Protocol Specification	B. Callaghan et al.	Junho de 1995
2623	NFS Version 2 and Version 3 Security Issues	M. Eisler	Junho de 1999
2624	NFS Version 4 Design Considerations	S. Shepler	Junho de 1999
3530	NFS Version 4 Protocol	S. Shepler et al.	Abril de 2003

16.8 Exercícios

★ E16.1 Explore sua configuração NFS local. O NFS é utilizado ou há uma outra solução em seu lugar? A automontagem é utilizada? Que decisões foram tomadas?

★ E16.2 Qual é a relação entre **mountd**, **nfsd** e **portmap**? Qual o significado da dependência do NFS para com **portmap** em termos de segurança?

★★ E16.3 Quais são algumas das implicações de projeto do NFS ser um protocolo sem estados? Discuta, em particular, qualquer efeito que essa condição de ausência de estados tem sobre o bloqueio de arquivos, permissões de acesso e segurança. Em que um sistema de arquivos de rede com estados diferiria do NFS?

★★ E16.4 Seu empregador precisa exportar **/usr** e **/usr/local** via NFS. Foram fornecidas as informações e solicitações a seguir:

a) Devido à política da empresa, você quer que somente o seu departamento (sub-rede local 192.168.123.0/24) seja capaz de usar esses sistemas de arquivos exportados. Que linhas teriam de ser adicionadas a quais arquivos para implementar essa configuração? Preste atenção às opções de exportação apropriadas.

b) Liste as etapas necessárias para fazer com que **mountd** e **nfsd** reconheçam esses novos sistemas de arquivos compartilhados. Como você poderia verificar se os diretórios estavam sendo compartilhados sem ter de montá-los?

c) Descreva uma estratégia que faria com que todas as máquinas de sua sub-rede local montassem automaticamente os diretórios exportados nos pontos de montagem **/mnt/usr** e **/mnt/usr/local**.

17 Compartilhando arquivos de sistema

Um sistema funcionando de maneira adequada depende de dezenas, talvez centenas, de arquivos de configuração, todos contendo os trechos corretos de informação. Ao multiplicarmos o número de arquivos de configuração em um host pelo número de hosts em uma rede, o resultado pode se transformar em milhares de arquivos — um número muito grande para ser gerenciado manualmente.

No mundo real, as máquinas costumam ser semelhantes do ponto de vista administrativo. Em vez de editar arquivos de texto em cada máquina, você pode, por uma questão de eficiência, combinar máquinas em grupos que compartilham informações de configuração. Você pode combinar máquinas de várias formas diferentes.

A maneira mais simples é preservar uma cópia mestre de cada arquivo de configuração em um local e distribuí-la aos membros do grupo sempre que o arquivo for alterado. Essa solução tem as vantagens de ser simples e de funcionar em qualquer sistema Linux (e UNIX).

Uma outra forma seria eliminar todos os arquivos de texto e fazer com que cada máquina obtivesse suas informações de configuração de um servidor central. Essa solução é mais complicada do que cópia de arquivo, mas resolve alguns problemas adicionais. Por exemplo, os clientes não podem perder atualizações, mesmo se eles estiverem fora do ar quando acontecer uma modificação. Talvez também seja mais rápido obter informações de um servidor do que de um arquivo, dependendo da velocidade do disco local e do nível de caching executado pelo servidor. Por outro lado, toda a rede pode parar quando o servidor central tornar-se inativo.

Para aumentar o desafio, a maioria das empresas atuais enfrenta o suporte a uma mistura de plataformas — algumas UNIX, algumas Linux e outras Windows — e os usuários estão cada vez mais irritados quando precisam lidar com inconveniências como a de se lembrar de uma senha diferente (e alterá-la) em cada plataforma. Sincronizar informações de configuração de usuário entre sistemas completamente diferentes (como Windows e Linux) era antigamente apenas um sonho. Hoje, isso é corriqueiro.

A história das tentativas de desenvolver bancos de dados administrativos distribuídos para grandes redes remonta a várias décadas e produziu muitos sistemas interessantes. Mas, nenhum dos sistemas atualmente em uso geral parece exatamente correto em sua abordagem. Alguns são simples, mas inseguros e não podem ser expandidos. Outros são funcionais, mas difíceis de serem manejados. Todos os sistemas parecem ter limitações que podem impedir você de configurar a rede da maneira desejada e nenhum deles gerenciará todas as informações que talvez você queira compartilhar entre suas máquinas.

Neste capítulo, discutiremos primeiramente algumas técnicas básicas para manter os arquivos sincronizados dentro de uma rede. Então descrevemos o NIS, um sistema de banco de dados historicamente popular originalmente introduzido para UNIX. Por fim, abordamos o LDAP, um sistema independente de plataforma mais sofisticado que está se tornando o padrão de fato. Atualmente, a maioria das instalações está migrando para o LDAP, guiada principalmente pela adoção por parte da Microsoft (da maior parte) do padrão LDAP em seu produto Active Directory e o desejo de melhor integrar ambientes Linux e Windows.

17.1 O QUE COMPARTILHAR

Dos diversos arquivos de configuração em um sistema Linux, somente um subconjunto deles pode ser compartilhado de forma útil entre as máquinas. Os arquivos mais comumente compartilhados são apresentados na Tabela 17.1.

Muitos outros arquivos de configuração poderiam ser compartilhados entre os sistemas, dependendo do nível de similaridade que você deseja ter em sua instalação. Em grande parte, esses outros arquivos de configuração estão associados a aplicativos específicos (por exemplo, **/etc/sendmail.cf** para **sendmail**) e não são suportados por sistemas de bancos de dados administrativos como o NIS e o LDAP; você terá de compartilhar os arquivos por meio de suas cópias.

Tabela 17.1 Os arquivos de sistema que são comumente compartilhados

Nome de arquivo	Função
/etc/passwd	Banco de dados de informações de contas de usuário
/etc/shadow[a]	Armazena senhas de sombra
/etc/group	Define grupos UNIX
/etc/hosts	Associações entre nomes de host e endereços IP
/etc/networks[b]	Associa nomes de texto a números de rede IP
/etc/services	Lista números de portas para serviços de rede bem conhecidos
/etc/protocols	Mapeia nomes de texto para números de protocolo
/etc/ethers[b]	Mapeia entre nomes de host e endereços Ethernet
/etc/mail/aliases	Armazena aliases de e-mails
/etc/rpc	Lista números de ID para serviços RPC
/etc/netgroup[b]	Define conjuntos de hosts, usuários e redes
/etc/cups/printcap	Banco de dados de informações sobre impressoras
/etc/printcap.cups	Banco de dados de informações sobre impressoras (caminho alternativo)
/etc/termcap	Banco de dados de informações sobre tipo de terminal

a. Não necessariamente compartilhável com outras versões do UNIX já que a criptografia pode variar; consulte a página 68.

b. Não utilizado em todos os sistemas.

Consulte a página 471 para informações adicionais sobre PAM.

Historicamente, muitos dos arquivos apresentados na Tabela 17.1 têm sido acessados por meio de rotinas na biblioteca C padrão. Por exemplo, o arquivo **/etc/passwd** pode ser pesquisado com as rotinas **getpwuid**, **getpwnam** e **getpwent**. Essas rotinas cuidam da abertura, leitura e análise sintática do arquivo **passwd** para que os programas em nível de usuário não tenham de fazer isso por conta própria. As modernas distribuições Linux também utilizam módulos de autenticação plugáveis (*pluggable authentication modules* – PAM), que oferecem uma interface de programação padrão para realizar pesquisas relacionadas com segurança. O PAM permite que sistemas como Kerberos e LDAP sejam facilmente integrados no Linux.

Sistemas de bancos de dados administrativos complicam as coisas fornecendo fontes alternativas para grande parte destas informações. As rotinas de biblioteca C tradicionais (**getpwent** etc.) estão cientes dos sistemas de bancos de dados comuns e podem acessá-los além dos (ou em vez dos) arquivos de texto simples padrão. O complemento exato de fontes de dados que são consultadas é configurado pelo administrador do sistema; consulte a seção *Priorizando fontes de informações administrativas* na página 358 para detalhes adicionais.

17.2 NSCD: ARMAZENANDO EM CACHE OS RESULTADOS DAS PESQUISAS

Em algumas distribuições, outro dedo na torta de arquivos de sistema pertence ao **nscd** (*name service cache daemon*), o daemon de cache do serviço de nomes, cujo nome pode ser um pouco enganoso. Ele funciona em conjunto com a biblioteca C para armazenar em cache os resultados de chamadas de biblioteca como **getpwent**. **nscd** é simplesmente um empacotador para essas rotinas de biblioteca; ele não sabe nada sobre as origens de dados reais sendo consultadas. Em teoria, ele deve aprimorar o desempenho de pesquisas, mas qualquer melhora é amplamente imperceptível na visualização subjetiva do usuário.

Consulte o Capítulo 15 para informações adicionais sobre DNS.

Dissemos que "daemon de cache do serviço de nomes" é enganoso, pois o termo "serviço de nomes" geralmente se refere ao DNS, o sistema de banco de dados distribuído que faz o mapeamento entre nomes de hosts e endereços Internet. O **nscd** efetivamente coloca em cache os resultados das pesquisas DNS (pois envolve **gethostbyname** etc.), mas também envolve as rotinas de bibliotecas que acessam informações dos arquivos passwd e **group** e seus equivalentes de bancos de dados de rede. (Por segurança, as pesquisas em **/etc/shadow** não são armazenadas em cache.)

Teoricamente, o **nscd** não deve apresentar nenhum efeito na operação do sistema a não ser acelerar pesquisas repetidas. Na prática, ele pode provocar comportamentos inesperados, pois mantém sua própria cópia de resultados da pesquisa. As pesquisas são armazenadas no cache por um período fixo de tempo (definido no arquivo de configuração do **nscd**, **/etc/nscd.conf**) e sempre há a possibilidade de que alterações recentes não se reflitam no cache do **nscd** até que os dados anteriores tenham expirado. O **nscd** é inteligente o bastante para monitorar fontes de dados locais (como **/etc/passwd**) em busca de alterações, de modo que atualizações locais devem ser propagadas dentro de 15 segundos. Para entradas remotas, como aquelas recuperadas por meio de NIS, talvez seja necessário aguardar o período de expiração completo antes de as alterações entrarem em efeito.

Entre os exemplos de distribuições, apenas o SUSE executa o **nscd** por padrão. O Fedora e o RHEL instalam o **nscd**, mas não o iniciam em tempo de inicialização por padrão; para ativar o uso do **nscd**, basta executar **chkconfig nscd on**. O Debian e o Ubuntu são compatíveis com **nscd**, mas não incluem **nscd** na instalação padrão; execute **apt-get install nscd** para fazer seu download.

O **nscd** inicia em tempo de inicialização e executa continuamente. O padrão **/etc/nscd.conf** especifica um tempo-limite de 10 minutos para dados **passwd** e de uma hora para **hosts** e **group**, com um tempo-limite negativo de 20 segundos (o período de tempo antes de uma pesquisa malsucedida ser novamente executada). Na prática, esses valores raramente precisam ser alterados. Se uma alteração que você fez recentemente não é exibida, é provável que o motivo seja o **nscd**.

17.3 COPIANDO ARQUIVOS

A cópia de arquivo pelo método da força bruta não é uma solução elegante, mas funciona em qualquer tipo de máquina e é fácil de ser configurada e mantida. Também é um sistema confiável, pois minimiza as dependências entre as máquinas (embora também torne mais fácil as máquinas saírem de sincronismo). A cópia de arquivos oferece ainda a máxima flexibilidade em termos do que pode ser distribuído e como. De fato, ela também é usada freqüentemente para manter arquivos de dados e aplicativos atualizados, bem como arquivos de sistema.

Bem poucos arquivos de configuração não são suportados por qualquer um dos serviços de bancos de dados comuns. Alguns exemplos são **/etc/ntp.conf**, o que determina como os hosts mantêm seus relógios sincronizados e **/etc/sendmail.cf**, que instruem o **sendmail** sobre como entregar o e-mail. Para manter esses arquivos em sincronia (o que é em geral sensato), você realmente não tem escolha exceto utilizar algum tipo de sistema de cópia de arquivo, mesmo que você distribuia outros tipos de informações de configuração por meio do NIS ou LDAP.

Os sistemas de cópia de arquivos podem utilizar tanto um modelo "push" como um modelo "pull". Com o modelo "push", o servidor mestre distribui periodicamente os arquivos mais recentes a cada cliente, independentemente do cliente querê-los ou não. Os arquivos podem ser enviados explicitamente sempre que uma modificação for feita ou poderão ser simplesmente distribuídos segundo um cronograma regular (talvez com alguns arquivos sendo transferidos com maior freqüência do que outros).

O modelo "push" possui a vantagem de manter o sistema de distribuição centralizado em uma máquina. Arquivos, listas de clientes, scripts de atualização e cronogramas são mantidos em um único local, facilitando o controle. Uma desvantagem é que cada cliente tem de permitir que o mestre modifique seus arquivos de sistema, criando assim um risco de segurança.

Em um sistema pull, cada cliente é responsável pela própria atualização a partir do servidor. Essa é uma forma menos centralizada de distribuição de arquivos, mas também é mais adaptável e segura. Quando os dados são compartilhados entre limites administrativos, um sistema de extração é especialmente atraente porque as máquinas mestre e cliente não precisam estar em execução pelo mesmo grupo administrativo ou facção política.

rdist: distribuindo arquivos

O comando **rdist** é a maneira mais fácil de distribuir arquivos a partir de um servidor central. Ele tem certa semelhança com **make**: você utiliza um editor de texto para criar uma especificação dos arquivos a serem distribuídos e depois usa **rdist** para alinhar a realidade com a especificação feita. **rdist** copia arquivos somente quando estão desatualizados, assim você pode criar uma especificação como se todos os arquivos devessem ser copiados e deixar que **rdist** otimize o trabalho desnecessário.

rdist preserva informações relativas ao proprietário, grupo, modo e horário de modificação dos arquivos. Quando **rdist** atualiza um arquivo existente, ele primeiramente elimina a versão antiga antes de instalar a nova. Esse recurso faz com que **rdist** seja adequado para transferência de executáveis que poderiam estar em uso durante a atualização.[1]

Historicamente, **rdist** era executado sobre o **rsh** e utilizava autenticação do tipo **rsh** para ganhar acesso a sistemas remotos. Infelizmente, esse sistema não é seguro e está desativado por padrão em sistemas operacionais modernos. Ainda que a documentação **rdist** continue a falar sobre **rsh**, não se engane pensando que **rsh** é uma escolha razoável.

Versões atuais de **rdist** são melhores pelo fato de permitirem que qualquer comando que compreenda a mesma sintaxe possa ser um substituto do **rsh**. Na prática, o substituto é o **ssh**, que utiliza criptografia para verificar a identidade dos hosts e para impedir escutas na rede de obterem cópias de seus dados. A desvantagem é que você deve executar servidores **ssh** remotos de um modo que não exija uma senha (mas autentique o cliente com um par de chaves criptográficas). Isso é menos seguro do que normalmente recomendaríamos, mas ainda é um imenso progresso em relação ao **rsh**. Consulte a página 481 para informações adicionais sobre o **sshd** e seus modos de autenticação.

---

1. Embora a versão antiga desapareça do espaço de nomes do sistema de arquivos, ela continua a existir até que todas as referências tenham sido liberadas. Você também tem de estar ciente desse efeito ao gerenciar arquivos de log. Consulte a página 144 para obter informações adicionais.

Agora que já esgotamos os perigos de **rdist**, vejamos como ele realmente funciona. Como **make**, **rdist** procura um arquivo de controle (**Distfile** ou **distfile**) no diretório atual. **rdist -f** *distfile* especifica explicitamente o nome do caminho do arquivo de controle. Dentro de **Distfile**, tabulações, espaços e novas linhas são utilizados intercambiavelmente como separadores. Os comentários são introduzidos pelo símbolo de cerquilha (#).

O núcleo de um **Distfile** consiste em instruções na forma:

```
rótulo: nomesDeCaminhos -> destinos comandos
```

O campo *rótulo* associa um nome à instrução. A partir do shell, você pode executar **rdist** *rótulo* para distribuir somente os arquivos descritos em uma dada instrução.

nomesDeCaminho e *destinos* são, respectivamente, listas de arquivos a serem copiados e hosts para os quais copiar. Se uma lista contiver mais de uma entrada, a lista deve estar entre parênteses e os elementos devem estar separados por espaços em branco. Os *nomesDeCaminho* podem incluir caracteres curinga no estilo shell (por exemplo, **/usr/man/man**[**123**] ou **/usr/lib/***). A notação ~*usuário* também é aceitável, mas é avaliada separadamente nas máquinas de origem e de destino.

Por padrão, **rdist** copia os arquivos e diretórios listados em *nomesDeCaminho* para os caminhos equivalentes em cada máquina de destino. Você pode modificar esse comportamento fornecendo uma seqüência de comandos e terminando cada um deles com um ponto-e-vírgula.

Os seguintes comandos são compreendidos:

```
install opções [diretórioDeDestino];
notify listaDeNomes;
except listaDeCaminhos;
except_pat listaDePadrões;
special [listaDeCaminhos] string;
cmdspecial [listaDeCaminhos] string;
```

O comando `install` configura opções que afetam a maneira como **rdist** copia arquivos. As opções controlam tipicamente o tratamento de links simbólicos, a correção do algoritmo de verificação de diferenças de **rdist** e a maneira como as exclusões são manipuladas. As opções são especificadas de maneira diferente em sistemas diferentes e não são explicadas aqui em detalhes; verifique as páginas man para obter informações adicionais. As opções, que devem ser precedidas por `-o`, consistem em uma lista de nomes de opções separadas por vírgulas. Por exemplo, a linha:

```
install -oremove,follow ;
```

faz com que **rdist** siga links simbólicos (em vez de simplesmente copiá-los como links) e remove arquivos existentes na máquina de destino que não tenham nenhum equivalente na máquina de origem. Consulte a página man de **rdist** para uma lista completa das opções. Os padrões são quase sempre aquilo que você quer.

O nome "install" é um tanto impreciso, já que os arquivos são copiados independentemente de um comando `install` estar presente ou não. As opções são especificadas como seriam na linha de comando de **rdist**, mas, quando incluídas em **Distfile**, elas se aplicam somente ao conjunto de arquivos manipulados por esse comando `install`.

O *diretórioDeDestino* opcional especifica um diretório de instalação nos hosts de destino. Por padrão, **rdist** utiliza os nomes de arquivos originais.

O comando `notify` usa uma lista de endereços de e-mail como argumento. **rdist** envia e-mails a esses endereços sempre que um arquivo é atualizado. Qualquer endereço que não contiver o símbolo "at" (@) receberá como sufixo o nome do host de destino. Por exemplo, **rdist** expandiria "pete" para "pete@anchor" ao relatar uma lista de arquivos atualizada no computador anchor.

Os comandos `except` e `except_pat` eliminam nomes de caminhos da lista de arquivos a serem copiados. Argumentos para `except` devem ser literalmente coincidentes, e aqueles para `except_pat` são interpretados como expressões regulares. Esses comandos de exceção são úteis, pois **rdist**, assim como **make**, permite a definição de macros no início de seu arquivo de controle. Talvez você queira utilizar uma lista semelhante de arquivos para várias instruções, especificando somente as adições e exclusões para cada host.

O comando `special` executa um comando shell (o argumento *string*, entre aspas) em cada computador remoto. Se uma *listaDeCaminhos* estiver presente, **rdist** executa o comando uma vez depois de copiar cada um dos arquivos especificados. Sem *listaDeCaminhos*, **rdist** executa o comando depois de cada arquivo. `cmdspecial` é semelhante, mas executa o comando de shell assim que a cópia estiver completa. (O conteúdo da *listaDeCaminhos* é passado para o shell como uma variável de ambiente.)

Eis um exemplo simples de um **Distfile**:

```
SYS_FILES  = (/etc/passwd /etc/group /etc/mail/aliases)
GET_ALL    = (chimchim lollipop barkadon)
GET_SOME   = (whammo spiff)
all: ${SYS_FILES} -> ${GET_ALL}
```

```
        notify barb;
        special /etc/mail/aliases "/usr/bin/newaliases";
    some: ${SYS_FILES} -> ${GET_SOME}
        except /etc/mail/aliases;
        notify eddie@spiff;
```

Consulte a página 382 para informações adicionais sobre o newaliases.

Essa configuração replica os três arquivos de sistema listados em chimchim, lollipop e barkadon e envia um e-mail para barb@*destino* descrevendo quaisquer atualizações ou erros que venham a ocorrer. Depois de **/etc/mail/aliases** ser copiado, **rdist** executa **newaliases** em cada um dos destinos. Somente dois arquivos são copiados para whammo e spiff. **newaliases** não é executado, e um relatório é enviado via e-mail para eddie@spiff.

Para que **rdist** funcione entre as máquinas, você também tem de instruir o daemon **sshd** nos hosts receptores a confiar no host do qual você está distribuindo os arquivos. Para tanto, você gera uma chave em puro texto para o host mestre e armazena uma cópia da parte pública no arquivo **~root/.ssh/authorized_keys** em cada receptor. Provavelmente também é interessante restringir o que essa chave é capaz de fazer e a partir de onde ela pode fazer login. Consulte a descrição do "método B" na página 481 para informações adicionais.

rsync: transferindo arquivos de maneira mais segura

rsync está disponível a partir de rsync.samba.org.

rsync, criado por Andrew Tridgell e Paul Mackerras, é semelhante em espírito a **rdist**, mas com um foco um pouco diferente. Ele não utiliza um arquivo de controle de cópia de arquivos da maneira como faz **rdist** (embora no servidor realmente haja um arquivo de configuração). **rsync** é uma espécie de versão envenenada de **rcp** que é escrupulosa em relação a preservar links, horários de modificação e permissões. É mais eficiente em termos de rede do que **rdist**, pois procura dentro de arquivos individuais e tenta transmitir somente as diferenças entre versões. A maioria das distribuições Linux fornece uma versão pré-empacotada de **rsync**, embora ele possa estar instalado por padrão.

Sob nossa perspectiva, a principal vantagem de **rsync** é o fato de máquinas receptoras poderem executar o lado remoto como um processo de servidor proveniente de **inetd** ou **xinetd**. O servidor (na verdade apenas um modo diferente de **rsync**, que obrigatoriamente deve ser instalado tanto no mestre como nos clientes) é bastante configurável: ele pode restringir o acesso remoto a um determinado conjunto de diretórios e pode exigir que o mestre prove sua identidade por meio de uma senha. Como não é necessário nenhum acesso **ssh**, podemos configurar **rsync** para distribuir arquivos de sistema sem muito comprometimento da segurança. (Entretanto, caso prefira usar **ssh** em vez de um processo de servidor baseado em **inetd**, **rsync** permite que você faça isso também.) Além disso, **rsync** também pode ser executado no modo pull (extraindo arquivos do servidor **rsync**, em vez de receber dele), o que é até mais seguro (veja a seção *Extraindo arquivos*, na página 354).

Infelizmente, **rsync** não é tão flexível quanto **rdist** e sua configuração é menos sofisticada do que o **distfile** de **rdist**. Não podemos executar comandos arbitrários nos clientes e não podemos aplicar **rsync** a vários computadores de uma vez.

Como exemplo, o comando:

```
# rsync -gopt --password-file=/etc/rsync.pwd /etc/passwd lollipop::sysfiles
```

transfere o arquivo **/etc/passwd** para a máquina lollipop. As opções **-gopt** preservam as permissões, propriedades e os horários de modificação do arquivo. Os dois-pontos duplicados em **lollipop::sysfiles** fazem com que **rsync** contate o **rsync** remoto diretamente na porta 873 em vez de utilizar **ssh**. A senha armazenada em **/etc/rsync.pwd** autentica a conexão.[2]

Esse exemplo transfere somente um arquivo, mas **rsync** é capaz de manipular vários arquivos de uma só vez. Além disso, os flags **--include** e **--exclude** permitem que especifiquemos uma lista de expressões regulares para identificar nomes de arquivos, de modo que podemos configurar um conjunto de critérios de transferência relativamente sofisticado. Se a linha de comando se tornar muito extensa, você também poderá ler os padrões de arquivos separados por meio das opções **--include-file** e **--exclude-file**.

Assim que o pacote **rsync** é instalado, o Red Hat, o Fedora e o SUSE fornecem configurações **xinetd** para **rsync**. Entretanto, você deve editar **/etc/xinetd.d/rsync** e mudar `disable = yes` para `disable = no` para realmente ativar o servidor.

O pacote **rsync** em Debian e Ubuntu requer uma entrada em **/etc/inetd.conf** para ativar a funcionalidade de servidor. Supondo que você utilize empacotadores TCP, a entrada deve ser semelhante a esta:

```
rsync  stream tcp nowait root /usr/sbin/tcpd /usr/bin/rsyncd --daemon
```

2. Embora a senha não seja enviada em texto puro pela rede, os arquivos transferidos não são criptografados. Se você usar **ssh** como transporte (**rsync -gopt -e ssh /etc/passwd /etc/shadow lollipop:/etc** — observe que apenas um caractere de dois-pontos), a conexão será criptografada, embora **sshd** tenha de ser configurado para não exigir uma senha. Escolha seu veneno!

É uma boa idéia configurar o daemon **tcpd** para bloquear acesso de todos os hosts exceto daquele que distribuirá os arquivos de sistema. A rejeição de hosts também pode ser especificada em **rsyncd.conf**, mas não custa nada erguer várias barreiras.

Assim que tiver habilitado **rsync**, você precisará configurar uma série de arquivos de configuração para instruir o servidor **rsync** sobre como se comportar. O principal arquivo é **/etc/rsyncd.conf**, que contém tanto parâmetros de configuração globais como um conjunto de "módulos", cada um dos quais é uma árvore de diretórios para exportação ou importação. Uma configuração razoável para um módulo que funciona no modo push (isto é, que aceitará transferências de arquivos enviados pelo cliente com que se estabelece a conexão) será parecida com o seguinte:

```
# o sysfiles é somente um título arbitrário para o módulo particular.
[sysfiles]
# Esse é o caminho para o qual você permite que os arquivos sejam distribuídos. Poderia ser só /.
path = /etc
# Esse é o arquivo que especifica o par usuário/senha para autenticar o módulo
secrets file = /etc/rsyncd.secrets
# Somente pode ser lido se você estiver distribuindo arquivos
read only = false
# UID e GID sob os quais a transferência será feita
uid = root
gid = root
# Lista de hosts que têm permissão para se conectar
hosts allow = nomeDeHostDaDistribuiçãoMaster
```

Muitas outras opções podem ser configuradas, mas os padrões são razoáveis. Essa configuração limita as operações ao diretório **/etc** e permite o acesso somente pelo host listado. Sob a perspectiva do usuário ou do cliente, podemos enviar arquivos via **rsync** para o servidor com o destino *nomeDoHost*::**sysfiles**, que mapeia para o módulo anterior. Se você quiser configurar **rsync** no modo pull (extraindo arquivos de um servidor **rsync** central), a configuração acima ainda funcionaria, embora talvez você queira restringir um pouco mais as coisas (por exemplo, configurando o modo de transferência para somente leitura).

A última coisa que você precisa fazer é configurar um arquivo **rsyncd.secrets**. Ele é geralmente mantido em **/etc** (embora possa ser colocado em outro lugar) e contém as senhas que clientes podem utilizar para se autenticar. Por exemplo:

```
root:senha
```

Como regra geral, as senhas **rsync** devem ser diferentes das senhas de sistema. Pelo fato de serem mostradas no formato texto puro, **rsyncd.secrets** deverá ser legível somente pelo root.

Extraindo arquivos

Você pode implementar um sistema de extração (*pulling system*) de várias maneiras. A maneira mais simples e direta é disponibilizar os arquivos em um servidor FTP ou Web[3] central e deixar os clientes fazerem o download automaticamente quando necessário. Antigamente, os administradores criariam seus próprios utilitários para fazer isso (muitas vezes criando um script **ftp** com um sistema como o **expect**), mas agora os utilitários padrão podem fazer isso para você.

Um tipo de utilitário assim que vem com a maioria das distribuições Linux é o popular **wget**. Trata-se de um programa pequeno e simples que pesquisa o conteúdo de uma URL (FTP ou HTTP). Por exemplo, para transferir um arquivo via FTP com **wget**, basta executar:

```
wget ftp://usuário:senha@nomeDoHost/caminho/para/arquivo
```

O *arquivo* especificado é depositado no diretório atual.

Outra alternativa somente para FTP é **ncftp**, que também acompanha a maioria das distribuições. Na realidade, trata-se apenas de um cliente FTP otimizado que permite a criação fácil de scripts.

Alguns sites distribuem arquivos publicando-os em um sistema de arquivos em rede como o NFS. Essa talvez seja a técnica mais simples sob o ponto de vista de automação — tudo o que é necessário é **cp** (copiar), pelo menos teoricamente. Na prática, provavelmente você vai querer ser um pouco mais sofisticado e verificar sintomas de problemas de segurança e conteúdo corrompido antes de copiar cegamente os arquivos de sistema. Divulgar arquivos de sistema críticos via NFS tem muitas desvantagens sob o ponto de vista de segurança, mas é uma maneira simples e eficiente de transferir informações. Você pode sempre encriptar os dados para reduzir a chance de interceptação por um invasor.

---

3. Tenha em mente que tanto HTTP como FTP transportam dados em texto simples. É recomendável considerar HTTPS ou SFTP, respectivamente, se a segurança for uma preocupação.

Uma opção final é utilizar **rsync** como descrito na seção anterior. Se você executar um servidor **rsync** em um computador de distribuição central, os clientes podem simplesmente baixar os arquivos via **rsync**. Utilizar esse método é talvez ligeiramente mais complexo do que utilizar FTP, mas dá acesso a todos os recursos de **rsync**.

Independentemente do sistema utilizado, tome cuidado para não sobrecarregar o seu servidor de dados. Se um monte de máquinas da rede tentar acessar o servidor ao mesmo tempo (por exemplo, se todo o mundo tentar atualizar **cron** simultaneamente), você poderá provocar inadvertidamente um ataque de negação de serviços. Sites grandes devem ter esse problema em mente e fornecer escalonamento de tempo ou randomização. Uma maneira simples de fazer isto é empacotar tarefas **cron** em um script Perl como o indicado a seguir:

```
#!/usr/bin/perl
sleep rand() * 600; # sleep between 0 and 600 seconds (i.e., 10 minutes)
system(copy_files_down);
```

17.4 NIS: O NETWORK INFORMATION SERVICE

O NIS, lançado pela Sun nos anos 80, foi o primeiro banco de dados administrativo de primeira linha. Ele se chamava originalmente Sun Yellow Pages, mas teve de receber um outro nome devido a questões judiciais. Os comandos do NIS ainda começam com as letras **yp**, portanto, é difícil se esquecer do nome original. O NIS foi amplamente adotado entre os fornecedores de UNIX e é suportado por todas as distribuições do Linux.

A unidade de compartilhamento no NIS é o registro, não o arquivo. Um registro normalmente corresponde a uma linha em um arquivo de configuração. Um servidor mestre mantém as cópias autorizadas dos arquivos de sistema, que são mantidas em suas localizações e formatos originais e são editadas com um editor de texto exatamente como antes. Um processo de servidor disponibiliza o conteúdo dos arquivos através da rede. Um servidor e seus clientes formam um "domínio" NIS.[4]

Os arquivos de dados são pré-processados nos arquivos de bancos de dados pela biblioteca de hashing Berkeley DB para melhorar a eficiência das pesquisas. Depois de editar arquivos no servidor mestre, você utiliza **make** para dizer ao NIS para convertê-los a seu formato em hash, utilizando **make** ou um script chamado **ypmake**, dependendo do sistema.

Somente uma chave pode ser associada a cada entrada, de modo que um arquivo de sistema talvez tenha de ser convertido em vários "mapas" NIS. Por exemplo, o arquivo **/etc/passwd** é convertido em dois mapas diferentes chamados **passwd.byname** e **passwd.byuid**. Um mapa é utilizado para pesquisar entradas por nome de usuário, e o outro, para pesquisar entradas por UID. Qualquer um deles pode ser utilizado para enumerar as entradas no arquivo **passwd**. Entretanto, pelo fato de as bibliotecas de hashing não preservarem a ordem dos registros, não há nenhuma maneira de reconstruir uma cópia exata do arquivo original (a menos que ele tenha sido classificado).

O NIS permite que repliquemos os mapas de rede em um conjunto de servidores escravos. Fornecer mais de um servidor ajuda a aliviar a carga sobre o mestre e a manter os clientes funcionando mesmo quando alguns servidores tornarem-se indisponíveis. Toda vez que um arquivo é alterado no servidor mestre, o mapa NIS correspondente tem de ser enviado para os escravos para que todos os servidores forneçam os mesmos dados. Os clientes não fazem distinção entre servidores mestre e escravo.

Na implementação original do NIS, tínhamos de colocar pelo menos um servidor NIS em cada rede física. Os clientes utilizam broadcast de IP para localizar servidores, e pacotes de broadcast não são encaminhados por roteadores e gateways. O comando **ypset** pode apontar um cliente em um determinado servidor; entretanto, a primeira dica de problema, o cliente tenta localizar um novo servidor por broadcast. A menos que um servidor na rede do cliente responda, essa seqüência de eventos pode fazer com que o cliente fique pendurado.

Esse sistema provoca uma série de problemas, e um dos não menos importantes é o fato de ser extremamente inseguro. Um invasor pode configurar um servidor NIS falso que responda a broadcasts e forneça dados falsos ou então emita um ataque de negação de serviços permitindo conexões e depois bloqueando requisições verdadeiras. Nos dias de hoje, a técnica de gerenciamento preferida é fornecer a cada cliente uma lista explícita de seus servidores NIS legítimos. Esse sistema também possui a vantagem de que os servidores não precisam estar em uma sub-rede local.

No Linux, os servidores são listados em **/etc/yp.conf**. Eis um exemplo para o atrustnis de domínio NIS:

```
domain  atrustnis  server  10.2.2.3
domain  atrustnis  server  10.2.2.4
```

4. Não confunda domínios NIS com domínios DNS. Eles são completamente diferentes e não têm nada a ver um com o outro.

Há uma linha para cada servidor; se um deles falhar, o NIS passará para outro. Note que os servidores são dados na forma de endereços IP. **yp.conf** aceita nomes de hosts, mas esses nomes de hosts têm de ser determinados no momento da inicialização (isto é, enumerados no arquivo **/etc/hosts** ou convertidos pelo DNS).

Se você for *obrigado* a utilizar o modo de broadcast, a sintaxe é:

```
domain  atrustnis  broadcast
```

Entendendo o funcionamento do NIS

Os arquivos de dados do NIS são armazenados no diretório **/var/yp**.Cada mapa NIS é armazenado em um formato hashed em um subdiretório do diretório NIS nomeado conforme o domínio NIS. Há um mapa (de arquivo) para cada chave pela qual um arquivo pode ser pesquisado. Por exemplo, no domínio cssuns, os arquivos DB para os mapas **/etc/passwd** poderiam ser:

```
/var/yp/cssuns/passwd.byname
/var/yp/cssuns/passwd.byuid
```

O arquivo **passwd** é pesquisável tanto pelo nome como pelo UID; portanto, dois mapas são derivados dele.

O comando **makedbm** gera mapas NIS a partir de arquivos de texto. Entretanto, você não precisa invocar esse comando diretamente; um **Makefile** em **/var/yp** gera todos os mapas NIS comuns. Depois de modificar um arquivo de sistema, mude para o diretório **/var/**e execute **make**. O comando **make** compara a data/hora de modificação de cada arquivo com a dos mapas derivados dele e executa **makedbm** para cada mapa que precisa ser reconstruído.

Os mapas são copiados do servidor mestre para os servidores escravos por meio do comando **ypxfr**. **ypxfr** é um comando de extração (*pull*); ele tem de ser executado em cada servidor escravo para fazer com que esse servidor importe o mapa. Os escravos normalmente executam **ypxfr** freqüentemente para verificar se realmente estão de posse dos mapas mais recentes; podemos utilizar **cron** para controlar a freqüência com que isso deve ser feito.

A implementação padrão da cópia de mapas é um tanto ineficiente. O Linux equipa um daemon chamado **rpc.ypxfrd** que pode ser executado no servidor mestre para aumentar a velocidade de respostas para solicitações **ypxfr**. **rpc.ypxfrd** desvia o protocolo NIS normal e simplesmente distribui cópias dos arquivos de mapa.

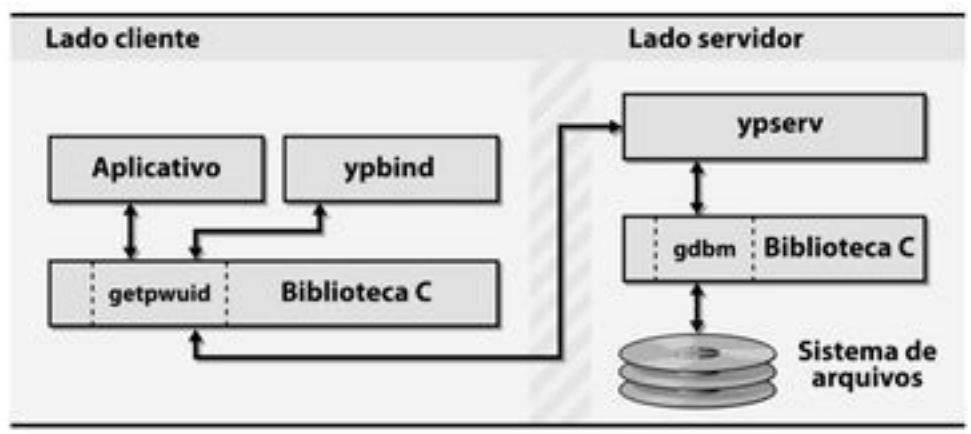
yppush é um comando "push" utilizado no servidor mestre. Ele, na verdade, não transfere qualquer dado, mas sim instrui cada escravo para executar um comando **ypxfr**. O comando **yppush** é utilizado por **Makefile** no diretório NIS para garantir que mapas recém-atualizados sejam propagados a escravos.

O mapa especial chamado **ypservers** não corresponde a nenhum arquivo de texto simples. Esse mapa contém uma lista de todos os servidores do domínio. Ele é construído automaticamente quando o domínio é configurado com **ypinit** (consulte a seção *Configurando servidores NIS*, na página 359). Seu conteúdo é examinado quando o servidor mestre precisar distribuir mapas para os escravos.

Após a configuração inicial, os únicos componentes ativos do sistema NIS são os daemons **ypserv** e **ypbind**. **ypserv** é executado somente nos servidores (tanto mestre como escravo); ele aceita consultas de clientes e as responde pesquisando informações nos arquivos de mapas de hash.

ypbind é executado em cada máquina do domínio NIS, inclusive servidores. A biblioteca C contata o daemon **ypbind** local quando precisa responder a uma consulta administrativa (desde que **/etc/nsswitch.conf** solicite para proceder desta maneira). **ypbind** localiza um **ypserv** no domínio apropriado e retorna sua identidade à biblioteca C, que a seguir contata o servidor diretamente. O mecanismo de consulta é ilustrado na Ilustração A.

Ilustração A Procedimento de consulta no NIS



As versões atuais de **ypbind** no Linux verificam periodicamente se estão de fato lidando com o servidor mais responsivo de um domínio NIS. Essa é uma melhora em relação à implementação tradicional, que se fixa em um determinado servidor. Outro recurso único do Linux é que os clientes podem se vincular a diferentes domínios NIS para diferentes mapas.

Uma vez que **ypbind** localiza um servidor, ele continua a depender desse servidor para todas as consultas até que o servidor saia do ar ou que ocorra algum outro problema de comunicação. Um **ypbind** em um servidor não dá a si próprio tratamento preferencial, de modo que os servidores não necessariamente se vinculam a si mesmos.

O NIS inclui uma série de comandos secundários que examinam mapas, descobrem que versão de um mapa que cada servidor está utilizando e controlam a vinculação entre clientes e servidores. Uma lista completa dos daemons e comandos do NIS é dada na Tabela 17.2. (As linhas são sombreadas para melhorar a legibilidade.)

Tabela 17.2 Comandos e daemons no NIS

Programa	Descrição
ypserv	É o daemon de servidor NIS, iniciado em tempo de inicialização
ypbind	É o daemon de cliente NIS, iniciado em tempo de inicialização
domainname	Configura o domínio NIS em que a máquina se encontra (executado em tempo de inicialização)
ypxfr	Faz o download da versão atual de um mapa por meio do servidor mestre
ypxfrd	Atende a solicitações de **ypxfr** (executado no servidor mestre)
yppush	Faz com que os servidores escravos atualizem suas versões de um mapa
makedbm	Constrói um mapa de hash por meio de um arquivo de texto simples
ypmake	Reconstrói os mapas de hash por meio de arquivos de texto simples que foram modificados
ypinit	Configura um host como servidor mestre ou escravo
ypset	Faz **ypbind** conectar-se a um servidor particular[a]
ypwhich	Descobre o servidor que o host atual está utilizando
yppoll	Descobre a versão de um mapa que um servidor está utilizando
ypcat	Imprime os valores contidos em um mapa NIS
ypmatch	Imprime as entradas de mapa para uma chave especificada
yppasswd	Altera uma senha no servidor mestre NIS
ypchfn	Altera as informações GECOS no servidor mestre NIS
ypchsh	Altera um shell de login no servidor mestre NIS
yppasswdd	É o servidor de **yppasswd**, **ypchsh** e **ypchfn**

a. Deve ser especificamente ativado com **ypbind -ypsetme** ou **ypbind -ypset** (perigoso).

Pesando as vantagens e desvantagens do NIS

Um excelente recurso do NIS é que ele pode ser compreendido por meros mortais. O NIS é análogo a copiar arquivos; na maioria dos casos, não é necessário que os administradores estejam cientes dos formatos de dados internos do NIS. A administração é feita com os mesmos velhos arquivos de texto simples e somente um ou dois novos procedimentos precisam ser aprendidos.

Como os domínios NIS não podem ser vinculados, o NIS não é adequado para gerenciar uma grande rede de máquinas a menos que uma configuração única seja aplicada a cada máquina. Você pode dividir uma grande rede em vários domínios NIS, mas cada domínio tem de ser administrado separadamente. Mesmo se uma grande rede efetivamente utilizar uma única configuração, limitações na expansão de servidores escravos significam que na prática esses sites normalmente terão algum outro tipo de mecanismo para manter seus servidores NIS em sincronia. Eles normalmente acabam empregando seus próprios bancos de dados auxiliares e fazendo com que seus servidores NIS atualizem seus dados com base nessa fonte central.

Consulte o Capítulo 8 para informações adicionais sobre **cron**.

Se um servidor escravo estiver fora do ar ou inacessível quando um mapa for alterado, a cópia do escravo não será atualizada. Os escravos têm de sondar o mestre periodicamente para se certificarem de que têm a versão mais recente de cada mapa. Embora sejam fornecidas com o NIS ferramentas básicas para sondagem, você tem de implementar o esquema desejado utilizando **cron**. Mesmo assim, é possível que duas versões diferentes de um mapa fossem simultaneamente atendidas por um tempo, com os clientes tendo uma visão aleatória um ou do outro.

Ele apresenta segurança mínima. O modo de broadcast é particularmente ruim; qualquer host em uma rede pode se dizer servidor de um determinado domínio e alimentar clientes NIS com dados administrativos adulterados. Você pode evitar esse problema enumerando explicitamente os servidores NIS permissíveis para cada cliente.

Podemos restringir os hosts capazes de ler os mapas de um servidor listando-os explicitamente em **/etc/ypserv.conf**, mas isso não é 100% seguro. Você também pode aumentar a segurança de um sistema distribuindo o arquivo de senhas sombras com alguma outra técnica (como **rdist** ou **rsync**); mas não recomendamos o emprego de NIS para servir senhas sombras.

Versões mais antigas do NIS do Linux contêm brechas de segurança conhecidas. Caso esteja rodando um sistema mais antigo, certifique-se de obter as atualizações mais recentes antes de iniciar o NIS.

Priorizando fontes de informações administrativas

As informações de configuração podem ser distribuídas de diversas formas. Cada sistema entende arquivos de texto simples e sabe como utilizar o DNS para pesquisar nomes de hosts e endereços Internet. A maioria também entende NIS. Como uma determinada parte das informações poderia vir de várias fontes potenciais, o Linux fornece uma maneira de especificar as fontes que são verificadas e a ordem em que essas verificações são feitas.

Na implementação original do NIS (anterior ao Linux), alguns arquivos de configuração (particularmente os arquivos **/etc/passwd** e **/etc/group**) tinham de ser configurados para "convidar" os conteúdos dos mapas NIS correspondentes. O convite era estendido por inclusão de comandos especiais dentro dos próprios arquivos. Um sinal "+" isolado no início de uma linha incluiria todo o mapa NIS, "*+@grupo_de_rede*" incluiria somente as entradas relevantes a um dado grupo de rede e "*+nome*" incluiria uma única entrada.

Essa abordagem jamais foi benquista e foi suplantada por um arquivo de configuração, **/etc/nsswitch.conf**, que permite a especificação de um caminho de busca explícito para cada tipo de informação administrativa. O comportamento original pode ser emulado pelo uso de um modo de compatibilidade, mas você provavelmente não iria querer utilizar esse recurso em uma rede configurada recentemente. (Infelizmente, a emulação é o padrão na maioria das distribuições.)

Um arquivo **nsswitch.conf** típico seria parecido com o seguinte:

```
passwd:    files nis
hosts:     files dns
group:     files
...
```

Cada linha configura um tipo de informação (normalmente, um equivalente ao arquivo de texto simples). Os códigos-fonte comuns são `nis`, `nisplus`, `files`, `dns` e `compat`; eles se referem aos arquivos simples básicos NIS+,[5], NIS+, (ignorando tokens como "+"), DNS e arquivos de texto simples NISificados (honrando "+"), respectivamente. O DNS é uma fonte de dados válida somente para informações de host e de rede.

Suporte para cada tipo de código-fonte vem de uma biblioteca compartilhada (**/lib/libnss***), então as distribuições variam ligeiramente nas fontes que suportam. Algumas distribuições fornecem suporte imediato para LDAP (consulte a página 361) e/ou Hesiod, um serviço de diretório baseado em DNS. Outro código-fonte mais suportado em Linux (e, infelizmente, não muito bem documentado) é `db`, que lê uma versão em hash do mapa a partir de **/var/db** (como **/var/db/passwd.db**). Se os arquivos de texto simples forem grandes, o emprego de versões de hash pode aumentar substancialmente a velocidade de pesquisa.

As fontes são tentadas da esquerda para a direita até que uma delas produza uma resposta para a consulta. No exemplo anterior, a rotina **gethostbyname** verificaria primeiramente o arquivo **/etc/hosts** e, se o host não estivesse listado lá, ela iria então verificar o DNS. Por outro lado, consultas sobre grupos UNIX verificariam somente o arquivo **/etc/group**.

Se necessário, podemos definir a "falha" de uma fonte mais especificamente colocando expressões entre colchetes após a falha. Por exemplo, a linha:

```
hosts:   dns [NOTFOUND=return] nis
```

faz com que seja utilizado DNS exclusivamente se ele estiver disponível; uma resposta negativa do servidor de nomes faz com que as consultas retornem imediatamente (com um código de falha) sem verificar o NIS. Entretanto, o NIS é utilizado se nenhum servidor de nomes estiver disponível. Os vários tipos de falhas são mostrados na Tabela 17.3; cada uma delas pode ser configurada para `return` ou `continue`, indicando se a consulta deve ser abortada ou encaminhada para a próxima fonte.

Tabela 17.3 Modos de falha reconhecidos em /etc/nsswitch.conf

Condição	Significado
`UNAVAIL`	A fonte não existe ou está fora do ar
`NOTFOUND`	A fonte existe, mas não pode responder à consulta
`TRYAGAIN`	A fonte existe, mas está ocupada
`SUCCESS`	A fonte foi capaz de responder à consulta

5. Um sucessor infeliz do NIS original agora descontinuado pela Sun, mas ainda suportado por alguns sistemas por razões históricas.

Por padrão, todas as distribuições Linux vêm com arquivos **nsswitch.conf** que são razoáveis para uma máquina dedicada sem NIS. Todas as entradas vão para arquivos de texto simples, exceto as pesquisas de host, que primeiro consultam arquivos de texto simples e depois o DNS. A maioria das distribuições assume por padrão o modo `compat` para **passwd** e **group**, o que talvez valha a pena mudar. Se você realmente usa NIS, basta colocá-lo explicitamente no arquivo **nsswitch.conf**.

O Debian e seu primo Ubuntu são distribuídos com `protocols`, `services`, `ethers` e `rpc` indo para `db` e, então, `files`. Isto é um pouco estranho, já que o Debian e o Ubuntu não incluem, de fato, **/var/db** ou qualquer outro mecanismo para mantê-lo. Provavelmente seria um pouco mais eficiente ir diretamente para `files`; se quiser, você pode modificar os ajustes de configuração para fazer isso.

Utilizando netgroups

O NIS introduziu uma abstração popular conhecida como grupos de rede **netgroups**. Os grupos de rede nomeiam conjuntos de usuários, máquinas e redes para fácil referência em outros arquivos de sistema. Eles são definidos em **/etc/netgroup** e também são compartilhados na forma de um mapa NIS.

O formato de uma entrada **netgroup** é:

```
nomeDoGrupo listaDeMembros
```

Os membros são separados por um espaço em branco. Um membro pode ser um nome de grupo de rede ou uma trinca na forma

```
(nomeDeHost, nomeDoUsuário, nomeDeDominioNIS)
```

Qualquer campo vazio em uma trinca é um curinga; portanto, a entrada `(boulder,,)` se refere a todos os usuários em todos os domínios do host boulder (ou o próprio host boulder, dependendo do contexto no qual o grupo de rede é utilizado). Um traço em um campo indica negação, ou seja, a entrada `(boulder,-,)` se refere à máquina boulder e nenhum usuário. As definições de grupos de rede podem ser aninhadas.

Veja um exemplo simples de um arquivo **/etc/netgroup**:

```
bobcats          (snake,,) (headrest,,)
servers          (anchor,,) (moet,,) (piper,,) (kirk,,)
anchorclients    (xx,,) (watneys,,) (molson,,)
beers            (anchor,,) (anchor-gateway,,) anchorclients
allhosts         beers bobcats servers
```

Esses grupos de rede são definidos em termos de hosts; isto é típico para uso no mundo real.

Consulte o Capítulo 16 para informações adicionais sobre NFS.

Os grupos de rede são uma boa idéia. Eles simplificam os arquivos de sistema, tornando-os mais compreensíveis. Eles também acrescentam uma camada dissimulada que permite que o estado de um usuário ou máquina seja modificado em um arquivo em vez de em 15.

Configurando um domínio NIS

Você deve inicializar o NIS no servidor mestre, nos servidores escravos e em cada cliente. Isto é feito em duas etapas. Em primeiro lugar, execute **ypinit** em cada servidor. Em seguida, em cada máquina do domínio, configure o nome de domínio a partir de **/etc/domainname** ou um dos arquivos de inicialização de sistema e configure **/etc/nsswitch.conf** para importar dados NIS.

Configurando servidores NIS

O lado servidor do NIS deve, em geral, ser instalado como um pacote opcional separado chamado **ypserv**. O Debian e o Ubuntu fazem as coisas de modo ligeiramente diferente; seu pacote **nis** inclui tanto o lado cliente como o lado servidor.

ypinit inicializa tanto o servidor mestre como os servidores escravos para um domínio. No mestre, são utilizados os seguintes comandos:

```
# cd /var/yp                /* O diretório NIS, onde quer que ele esteja */
# domainname foo            /* Nome do novo domínio. */
# /usr/lib/yp/ypinit -m     /* Inicializa como servidor mestre. */
# ypserv                    /* Inicia o servidor NIS. */
```

O flag **-m** informa **ypinit** que ele está configurando um servidor mestre; ele solicita que você insira uma lista de servidores escravos. Assim que o mestre estiver funcionando, prepare cada servidor escravo executando **ypinit** com o flag **-s** (slave):

```
# cd /var/yp
# /usr/lib/yp/ypinit -s mestre   /* O argumento é o nome de host do mestre. */
# ypserv
```

ypinit -s faz uma cópia local dos dados atuais do mestre; a presença dos arquivos de dados do domínio é suficiente para que **ypserv** saiba que ele deve atender o domínio.

Consulte o Capítulo 8 para informações adicionais sobre cron.

Em cada escravo, você deve configurar entradas crontab para extrair cópias recentes de todos os mapas do mestre. O comando **ypxfr** *mapa*, onde *mapa* é um nome como **passwd.byuid**, transfere o mapa especificado do servidor mestre. Você tem de executar o comando uma vez para cada mapa. Os mapas tendem a mudar em velocidades diferentes, desse modo, você pode querer transferir alguns mapas mais freqüentemente que outros. Na maioria das circunstâncias, transferir todos os mapas uma a duas vezes ao dia (talvez tarde da noite) já seja suficiente. O script a seguir transfere todos os mapas:

```
#!/bin/sh
mydomain = `/usr/bin/domainname`
cd /var/yp/$mydomain   # o diretório NIS
for map in `/bin/ls`; do
     /usr/lib/yp/ypxfr $map
done
```

Além disso, os scripts em **/usr/lib/yp** transferem mapas NIS em várias freqüências (**ypxfr_1perday**, **ypxfr_2perday** e **ypxfr_1perhour**).

Se quiser que os usuários sejam capazes de modificar suas senhas com **yppasswd**, você terá de executar o daemon **yppasswdd** no servidor NIS mestre. A versão Linux desse servidor ficou conhecida por travar freqüentemente; portanto, certifique-se de verificar que ela ainda esteja executando se o comando **yppasswd** parecer não estar funcionando.

Configurando opções de controle de acesso em /etc/ypserv.conf

Você pode configurar opções para a versão Linux do daemon **ypserv** em **/etc/ypserv.conf**; mas apenas algumas opções são definidas e a maioria das instalações não precisará que seus valores padrão sejam alterados.

Mais importante que isso é que o **ypserv** procura o arquivo **ypserv.conf** para obter instruções sobre como controlar acesso aos dados NIS. Em vez de simplesmente emitir a resposta para cada consulta que entra, como a implementação tradicional faz, o **ypserv** do Linux verifica requisições de entrada em uma lista de acesso. Cada linha de controle tem a forma:

```
host:dominioNIS:mapa:segurança
```

host, *dominioNIS* e *mapa* identificam um determinado subconjunto de requisições e o parâmetro *segurança* diz como tratá-lo: `deny` para rejeitar a requisição; `port` para permitir a requisição contanto, que ela origine em uma porta de rede privilegiada (< 1024), e `none` para sempre permitir a requisição. Eis um exemplo de configuração:

```
128.138.24.0/255.255.252.0:atrustnis:*:none
*:*:passwd.byuid:deny
*:*:passwd.byname:deny
128.138.:atrustnis:*:port
*:*:*:deny
```

Você pode utilizar um asterisco nos campos *host*, *nisdomain* e *mapa* para corresponder a qualquer valor, mas correspondências parciais não são permitidas. (Não é possível, por exemplo, utilizar `passwd.*` para corresponder todos os mapas derivados do arquivo **/etc/passwd**.) As linhas de controle são verificadas em ordem até uma linha correspondente ser localizada. Se nenhuma linha corresponder, o padrão é responder à solicitação.

Consulte a página 201 para uma discussão sobre máscaras de rede CIDR.

O parâmetro *host* pode incluir uma máscara de rede, como na primeira linha, mas **ypserv** não entende a notação CIDR mais comum. Como mostrado na quarta linha, você também pode omitir componentes finais de um endereço de IP para fazer **ypserv** preenchê-lo com zeros e fornecer uma máscara de rede análoga.

As regras acima permitem acesso a partir de qualquer host em uma das redes 128.138.24/22. Os hosts dentro de 128.138 podem acessar todos os mapas em atrustnis, exceto os derivados do arquivo **/etc/passwd**, contanto que a solicitação se origine de uma porta privilegiada. Todos os outros acessos são negados.

Nunca se esqueça de que esse tipo de controle de acesso é, no melhor dos casos, uma medida improvisada. Você pode dissuadir a navegação casual de não-funcionários da empresa, mas isso não será um impedimento muito eficiente para um invasor determinado.

Um mecanismo de segurança mais antigo, o arquivo **/var/yp/securenets**, também é suportado por razões históricas. Novas configurações devem utilizar **ypserv.conf**.

Configurando clientes NIS

Após configurar servidores, informe a cada máquina cliente que ela é um membro do novo domínio. Os servidores de um domínio geralmente também são clientes.

Consulte o Capítulo 2 para mais informações sobre os scripts de inicialização de sistema.

O comando **domainname** configura o domínio NIS de uma máquina. Normalmente ele é executado no momento da inicialização por meio de um dos scripts de inicialização. Os mecanismos exatos necessários para configurar isso variam conforme as distribuições de sistema; os detalhes são fornecidos a seguir.

Cada cliente deve ter pelo menos uma versão privada mínima dos arquivos **passwd**, **group** e **hosts**. **passwd** e **group** são necessários para permitir que root faça o login quando nenhum servidor NIS estiver disponível. Eles devem conter as contas e grupos de sistema padrão: root, bin, daemon etc. O arquivo **hosts** (ou DNS) tem de estar presente para responder a consultas no momento da inicialização que ocorrem antes de o NIS estar ativo e em funcionamento.

Detalhes do NIS de acordo com o sistema de distribuição

No Fedora e RHEL, você define o nome de domínio NIS em **/etc/sysconfig/network** configurando a variável `NISDOMAIN`. O lado servidor do NIS é instalado como um pacote separado chamado **ypserv**. Os daemons **ypbind**, **ypserv** e **yppasswdd** são ativados e desativados com **chkconfig**; por exemplo:

```
# chkconfig ypbind on
```

O SUSE configura o nome de domínio NIS em tempo de inicialização a partir do arquivo **/etc/domainname**. O lado servidor do NIS é instalado como um pacote separado chamado **ypserv**. Utilize **chkconfig** para forçar o sistema a iniciar automaticamente **ypserv** e/ou **ypbind** em tempo de inicialização. Você pode configurar opções de linha de comando para **ypbind** em **/etc/sysconfig/ypbind**. Você deve configurar `YPBIND_BROADCAST` como `yes` nesse arquivo ou instalar um arquivo **/etc/yp.conf**; caso contrário, os scripts de inicialização recusarão a inicialização de **ypbind**.

O Debian e o Ubuntu mantêm o nome do domínio NIS em **/etc/defaultdomain**. Os scripts de inicialização executam **ypbind** automaticamente se esse arquivo estiver presente. Para executar **ypserv**, edite o arquivo **/etc/default/nis** e configure o valor de `NISSERVER` como `slave` ou `master`.

17.5 LDAP: O LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL

Os sites UNIX e Linux precisam de uma boa maneira de distribuir seus dados de configuração administrativos; entretanto, o problema é realmente mais abrangente do que isso. Por exemplo, e os dados não-administrativos como listas de telefone e de endereços de e-mail? E as informações que você deseja compartilhar com o mundo externo? O que todo o mundo realmente precisa é de um serviço de diretórios generalizado.

Um serviço de diretórios é apenas um banco de dados, mas um que faz algumas suposições. Qualquer conjunto de dados que possui características que atendam às suposições feitas é um candidato à inclusão. As suposições básicas são as seguintes:

- Objetos de dados são relativamente pequenos
- O banco de dados será amplamente replicado e colocado em cache
- As informações se baseiam em atributos
- Os dados são lidos freqüentemente, mas gravados com pouca freqüência
- A pesquisa é uma operação comum

O sistema atual de candidato a padrão do IETF desenvolvido para preencher esse papel é o LDAP (Lightweight Directory Access Protocol). As especificações LDAP não falam do banco de dados em si, mas são apenas a maneira como ele é acessado por meio de uma rede. Mas pelo fato de especificarem como os dados são esquematizados e como as pesquisas são realizadas, eles também implicam um modelo de dados razoavelmente específico.

O LDAP foi originalmente projetado como um protocolo de gateway que permitiria clientes TCP/IP conversar com um serviço de diretório mais antigo chamado X.500, que é agora obsoleto. Ao longo do tempo, ficou claro que o X.500 estava destinado à morte e também que o UNIX realmente precisava de um diretório padrão de alguma espécie. Esses fatores levaram o LDAP a ser desenvolvido como um sistema de diretórios completo e independente (e talvez não mais merecendo levar a letra L de *lightweight*, ou seja, "leve").[6]

Nessa altura (no ano de 2006), o LDAP tornou-se um dos mais importantes protocolos, talvez estimulado em parte pela adoção do LDAP por parte da Microsoft como a base de seu serviço Active Directory. No lado UNIX e Linux, o pacote OpenLDAP (www.openldap.org) tornou-se a implementação padrão.

---

6. Devido à torturante história do LDAP, muitas fontes tendem a ser muito detalhistas em relação a conexões X.500 e OSI do LDAP. Entretanto, essa história não é relevante para a utilização contemporânea do LDAP. Ignore-a.

A estrutura de dados LDAP

Os dados LDAP assumem a forma de listas de propriedades, que são conhecidas no mundo LDAP como "entradas". Cada entrada consiste em um conjunto de atributos identificados (como "uid" ou "description") junto com os valores desses atributos. Os usuários do Windows talvez reconheçam a semelhança dessa estrutura com a do Registro do Windows. Como no Registro, um atributo individual pode ter vários valores.

Como exemplo, eis uma típica (mas simplificada) linha de **/etc/passwd** expressa como uma entrada LDAP:

```
uid: ghopper
cn: Grace Hopper
userPassword: {crypt}$1$pZaGA2RL$MPDJocOafuhHY6yk8HQFp0
loginShell: /bin/bash
uidNumber: 1202
gidNumber: 1202
homeDirectory: /home/ghopper
```

Essa notação é um exemplo simples de LDIF, o LDAP Data Interchange Format, que é utilizado pela maioria das ferramentas e implementações de servidor relacionadas com o LDAP. O fato de os dados LDAP poderem ser facilmente convertidos em texto simples e vice-versa é parte da razão do seu sucesso.

As entradas são organizadas em uma hierarquia pelo uso de "nomes distintos" (nome de atributo: dn) que forma um tipo de caminho de pesquisa. Por exemplo, o dn para o usuário acima poderia ser

```
dn: uid=ghopper,ou=People,dc=navy,dc=mil
```

Como no DNS, o "bit mais significativo" encontra-se à direita. Aqui, o nome de DNS navy.mil foi utilizado para estruturar os níveis mais altos da hierarquia LDAP. Ele foi dividido em dois componentes de domínio (do dc), "navy" e "mil", mas essa é apenas uma das várias convenções comuns.

Cada entrada tem exatamente um nome distinto. Portanto, a hierarquia de entradas é como uma árvore ramificada sem loops. Mas há provisões para links simbólicos entre entradas e para referrals a outros servidores.

As entradas LDAP são em geral esquematizadas pelo uso de um atributo objectClass. As classes de objeto especificam os atributos que uma entrada pode conter, alguns dos quais podem ser exigidos para validade. O esquema também atribui um tipo de dados a cada atributo. O aninhamento e a combinação das classes de objeto são feitos no tradicional estilo orientado a objetos. O nível superior da árvore de classes de objeto é a classe chamada "top", que meramente especifica que uma entrada deve ter um atributo objectClass.

A Tabela 17.4 mostra alguns atributos LDAP comuns cujos significados talvez não sejam imediatamente aparentes.

Tabela 17.4 Alguns nomes de atributo comuns encontrados em hierarquias LDAP

Atributo	Significa	O que ela é
`o`	Organização	Identifica freqüentemente a entrada de primeiro nível de uma instalação
`ou`	Unidade organizacional	Uma subdivisão lógica, por exemplo, "marketing"
`cn`	Nome comum	O nome mais natural para representar a entrada
`dc`	Componente de domínio	Utilizado em sites que modelam sua hierarquia LDAP com base no DNS
`objectClass`	Classe de objetos	Esquema ao qual os atributos dessa entrada obedecem

A questão do LDAP

Até você adquirir alguma experiência do LDAP, ele pode ser um conceito difícil de entender. Diferentemente do NIS, o LDAP em si não resolve nenhum problema administrativo específico. Não há uma "tarefa principal" para a qual o LDAP foi especificamente criado para tratar, e as instalações divergem amplamente em suas razões para implantar servidores LDAP. Portanto, antes de partirmos para as especificidades da instalação e configuração do OpenLDAP, talvez valha a pena revisar algumas razões pelas quais você poderia querer investigar o LDAP para empregá-lo em sua instalação. Eis as maiores:

- Você pode utilizar o LDAP como um substituto do NIS. As vantagens sobre o NIS são numerosas, mas as principais são melhor segurança, melhor suporte de ferramentas e aumento de flexibilidade.

Consulte a página 384 para informações adicionais sobre como utilizar o LDAP com ***sendmail****.*

- De modo semelhante, o LDAP pode ser utilizado para distribuir informações de configuração que residem fora do domínio tradicional do NIS. A maioria dos sistemas de correio, inclusive **sendmail** e Postfix, pode passar uma grande parte de suas informações de roteamento pelo LDAP e essa é, de fato, uma das aplicações mais populares do LDAP. Ferramentas tão variadas

quanto o servidor Web Apache e o automontador autofs também podem ser configuradas para prestar atenção no LDAP. É possível que com o tempo o suporte LDAP se torne cada vez mais comum.

- É fácil acessar dados LDAP com ferramentas de linha de comando como **ldapsearch**. Além disso, o LDAP é bem suportado por linguagens de criação de scripts comuns como Perl e Python (pelo uso de bibliotecas). Daí o LDAP ser uma maneira impressionante de distribuir informações de configuração para scripts e utilitários administrativos escritos localmente.
- Excelentes ferramentas baseadas na Web estão disponíveis para gerenciar o LDAP, por exemplo, phpLDAPadmin (phpldapadmin.sourceforge.net) e Directory Administrator (diradmin.open-it.org). Essas ferramentas são tão fáceis de utilizar que você simplesmente as abre e começa a trabalhar sem precisar ler o manual.
- O LDAP é bem suportado como um serviço de diretório público. Os clientes de e-mail mais importantes, incluindo aqueles executados em PCs, suportam o uso de LDAP para acessar diretórios de usuário. Pesquisas LDAP simples também são suportadas por muitos navegadores Web pelo uso de um tipo LDAP URL.
- O Active Directory da Microsoft é baseado em LDAP, e a versão atual do Windows 2003 Server R2 inclui extensões (originalmente chamadas de "Services for UNIX", e agora de "Windows Security and Directory Services for UNIX") que facilitam o mapeamento de usuários e grupos Linux.

Documentação e especificações do LDAP

Não conhecemos nenhuma fonte de informações introdutórias realmente excelente sobre a arquitetura geral do LDAP. O FAQ onetime não mais é mantido. Infelizmente, a fonte que recomendamos para LDAP versão 2, *Understanding LDAP*, de Johner et al, não foi atualizada para versão 3 e as diferenças no protocolo são tão significativas que provavelmente merecem ser procuradas em outra parte.

Os melhores lugares atuais para começar são a documentação do pacote OpenLDAP e o Linux *LDAP HOWTO*. Citações de dois livros razoáveis sobre LDAP também são dadas no final deste capítulo.

As RFCs relacionadas com o LDAP são numerosas e variadas. Como um grupo, eles tendem a dar uma impressão de grande complexidade, o que é algo não representativo do uso médio. Alguns pontos altos são listados na Tabela 17.5, a seguir.

Tabela 17.5 RFCs importantes relacionadas com LDAP

RFC	Título
4519	LDAP: Schema for User Applications (esquema para aplicações do usuário)
4517	LDAP: Syntaxes and Matching Rules (regras de sintaxe e pesquisa)
4516	LDAP: Uniform Resource Locator
4515	LDAP: String Representation of Search Filters (representação de strings em filtros de pesquisa)
4514	LDAP: String Representation of Distinguished Names (representação de strings de nomes)
4513	LDAP: Authentication Methods and Security Mechanisms (métodos de autenticação e mecanismos de segurança)
4512	LDAP: Directory Information Models (modelos de informação de diretório)
4511	LDAP: The Protocol (o protocolo)
3672	Subentries in the Lightweight Directory Access Protocol (LDAP) (entradas secundárias do LDAP)
3112	LDAP Authentication Password Schema (esquema de autenticação usando senhas)
2849	LDAP Data Interchange Format (LDIF) — Technical Specification (o formato LDIF — especificação técnica)
2820	Access Control Requirements for LDAP (requerimentos de controle de acesso para o LDAP)
2307	An Approach for Using LDAP as a Network Information Service (uma abordagem usando o LDAP como um NIS)

OpenLDAP: o LDAP para Linux

O OpenLDAP é uma extensão do trabalho originalmente feito na Universidade de Michigan e agora continuado como um projeto de código-fonte aberto. Ele vem com a maioria das distribuições, embora não seja necessariamente incluído na instalação padrão. A documentação talvez seja mais bem descrita como "rápida".

Na distribuição de OpenLDAP, **slapd** é o daemon de servidor padrão e **slurpd** trata da replicação. Uma seleção de ferramentas de linha de comando permite a consulta e a modificação de dados LDAP.

A configuração é simples e direta. Primeiro, crie um arquivo **/etc/openldap/slapd.conf** copiando o exemplo instalado com o servidor OpenLDAP. Essas são as linhas nas quais você precisa prestar atenção:

```
database bdb
suffix "dc=meuDominio, dc=com"
```

```
rootdn "cn=administração, dc=meuDomínio, dc=com"
rootpw {crypt}abJnggxhB/yWI
directory /var/lib/ldap
```

O formato do banco de dados assume por padrão Berkeley DB, o que é bom para dados que residem no sistema OpenLDAP. Você pode utilizar uma variedade de outros back ends, inclusive métodos *ad hoc* como scripts que criam os dados durante a execução.

O `sufix` é o seu "nome de base LDAP". Ele é a raiz de sua parte do espaço de nomes LDAP, semelhante conceitualmente a seu nome de domínio DNS. Esse exemplo ilustra uma prática comum: utilizando seu nome de domínio de DNS como seu nome de base LDAP.

`rootdn` é o seu nome de administrador e `rootpw` é a senha formato UNIX (DES) do administrador. Note que os componentes de domínio que levam ao nome do administrador também devem ser especificados. Você pode copiar e colar a senha de **/etc/shadow** (caso não use senhas MD5) ou então gerá-la com um simples script Perl de uma linha:

```
perl -e "print crypt('senha','sal');"
```

onde *senha* é a senha desejada e *sal* é uma string arbitrária de dois caracteres. Devido à presença dessa senha, certifique-se de que as permissões em seu arquivo **slapd.conf** sejam 600 e que o arquivo pertença a root.

Você deve editar **/etc/openldap/ldap.conf** para configurar o servidor padrão e nome de base para as requisições de cliente LDAP. É muito simples — basta configurar o argumento da entrada `host` como o servidor e configurar `base` com o mesmo valor do `suffix` em **slapd.conf**. (Certifique-se de remover os caracteres de comentário em ambas as linhas.)

Nesta altura, você deve estar apto a iniciar **slapd** simplesmente executando-o sem nenhum argumento.

Substituindo o NIS por LDAP

Consulte a página 471 para informações adicionais sobre módulos de autenticação plugáveis.

Na realidade, você não precisa utilizar NIS para "migrar" para LDAP, mas como o NIS define um conjunto padrão de arquivos de sistema compartilhados, ele serve como um ponto de referência útil.

Os arquivos distribuíveis por NIS já são "virtualizados" pela comutação de serviço de nomes (o arquivo **/etc/nsswitch.conf**), então, é relativamente fácil adicionar o suporte LDAP do lado cliente. Algumas distribuições instalam o pacote **nss_ldap** necessário por padrão, mas, se não o fizer, o pacote é normalmente fornecido como uma opção. Esse pacote inclui um módulo PAM que permite utilizar LDAP com módulos de autenticação plugáveis além da comutação de serviço de nomes.

Os padrões LDAP do lado cliente para esse pacote são configurados em **/etc/ldap.conf**, que compartilha seu formato com o arquivo **/etc/openldap/ldap.conf** descrito na seção anterior, mas que tem opções adicionais específicas ao serviço de nomes e contextos PAM. Você também deve editar o arquivo **/etc/nsswitch.conf** em cada cliente para adicionar `ldap` como uma fonte para cada tipo de dados ao qual você quer aplicar LDAP. (As alterações de **nsswitch.conf** fazem a biblioteca C passar requisições para a biblioteca **libnss_ldap**, que então utiliza as informações de **/etc/ldap.conf** para descobrir como realizar as consultas de LDAP.)

A RFC2307 define o mapeamento padrão a partir de conjuntos de dados UNIX tradicionais, como os arquivos **passwd** e **group**, no espaço de nomes LDAP. É um documento de referência útil para administradores de sistemas utilizando LDAP como um substituto do NIS, pelo menos em teoria. Na prática, as especificações são lidas com muito mais facilidade por computadores do que por humanos; é melhor examinar os exemplos.

A Padl Software oferece um conjunto gratuito de scripts Perl que migram arquivos de texto simples existentes ou mapas NIS para LDAP. Esse conjunto está disponível em www.padl.com/tools.html, e a execução dos scripts é simples e direta. Eles podem ser utilizados como filtros para gerar LDIF, ou executados em um servidor ativo para carregar os dados diretamente. Por exemplo, o script **migrate_group** converte essa linha de **/etc/group**:

```
csstaff:x:2033:evi,matthew,trent
```

na seguinte LDIF:

```
dn: cn=csstaff,ou=Group,dc=domainname,dc=com
cn: csstaff
objectClass: posixGroup
objectClass: top
userPassword: {crypt}x
gidNumber: 2033
memberuid: evi
memberuid: matthew
memberuid: trent
```

(Note a classe de objetos e as especificações de nome distinto, que foram omitidas no exemplo **passwd** na página 362.)

Assim que um banco de dados tiver sido importado, você pode verificar se a transferência funcionou corretamente executando o utilitário **slapcat**, que exibe todo o banco de dados.

LDAP e segurança

Tradicionalmente, o LDAP era utilizado mais como uma lista telefônica do que qualquer outra coisa e, para esse propósito, enviar dados sem encriptá-los era normalmente aceitável. Como resultado, a implementação LDAP "padrão" concede acesso não-criptografado pela porta TCP 389. Entretanto, não aconselhamos nunca o uso de LDAP não-criptografado para a transmissão de informações de autenticação, mesmo se senhas forem codificadas ou encriptadas individual e manualmente.

Como alternativa, o LDAP sobre SSL (conhecido como LDAPS, que normalmente executa na porta TCP 686) está disponível na maioria das situações (inclusive no mundo Microsoft) tanto no cliente como no servidor. Esse método de acesso é preferível porque protege as informações contidas tanto na consulta como na resposta. Utilize LDAP quando possível.

Um sistema com a complexidade e com tantas partes móveis como o LDAP inevitavelmente apresenta o potencial para ser mal configurado de tal modo que enfraquece a segurança. Naturalmente, é provável que ele também contenha algumas brechas de segurança simples e ultrapassadas.

17.6 LEITURA RECOMENDADA

MALÈRE, LUIZ ERNESTO PINHEIRO. *LDAP Linux HOWTO*. Disponível em: <www.tldp.org/HOWTO/LDAP-HOWTO/>

VOGLMAIER, REINHARD. *The ABCs of LDAP:* How to Install, Run, and Administer LDAP Services. Boca Raton, FL: Auerbach Publications, 2004.

Carter, GERALD. *LDAP System Administration*. Sebastopol, CA: O'Reilly Media, 2003.

17.7 EXERCÍCIOS

E17.1 Por que um método "pull" de atualização dos arquivos de uma máquina local é mais seguro do que um método "push"?

E17.2 Explique o seguinte trecho de um arquivo de controle **rdist**:

```
LINUX_PASSWD = ( redhatbox debianbox susebox )

passwd:
( /etc/passwd ) -> ( ${LINUX_PASSWD} )
install /etc/passwd.rdist;
cmdspecial /etc/passwd.rdist "/usr/local/sbin/mkpasswd";
```

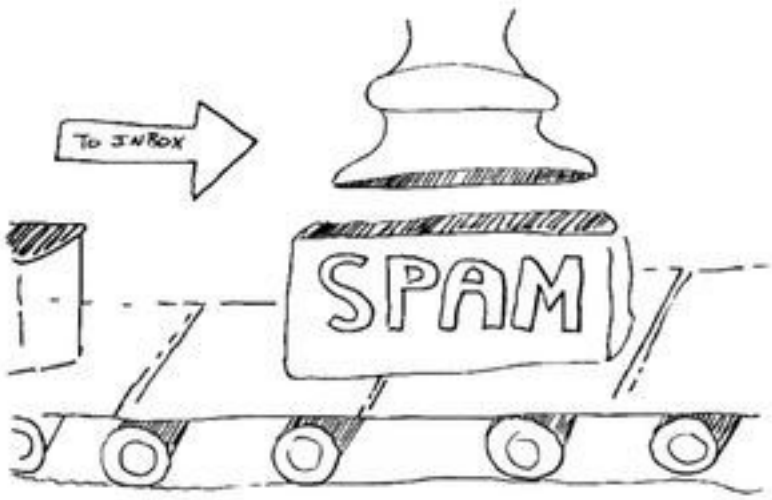
★ E17.3 Explique as diferenças entre **rdist** e **rsync**. Em que situações seria melhor utilizar um em relação ao outro?

★ E17.4 Compare NIS e LDAP. Quando você deveria utilizar um e não o outro? Alguma vez seria interessante utilizar ambos?

★ E17.5 Que método sua instalação emprega para compartilhar arquivos de sistema? Quais são os problemas de segurança relacionados com tal método? Sugira uma maneira alternativa de compartilhar arquivos de sistema em sua instalação e detalhe quais preocupações ela deve resolver. Quais são as desvantagens, se houver?

★★★★★ E17.6 Projete um esquema LDAP que armazene informações de usuário como login, senha, shell, máquinas autorizadas etc. Crie uma ferramenta que insira novos usuários no banco de dados interativamente ou a partir de um arquivo contendo uma lista de usuários. Crie uma ferramenta que gere os arquivos **passwd**, **group** e **shadow** a partir do banco de dados LDAP para as máquinas de seu laboratório. Permita que os usuários tenham senhas diferentes em cada máquina se eles quiserem. (Nem todos os usuários têm necessariamente autorização para utilizar todos os computadores.) Seu sistema **adduser** deverá ser capaz de imprimir listas de nomes de login de usuários e imprimir pares login/senha para novos usuários.

18 *E-mail*



É difícil se lembrar de como era o planeta sem e-mails. Todo mundo, dos filhos adolescentes, avós até as corporações mais prosaicas agora utilizam e-mail rotineiramente para comunicação com a família, colaboradores, parceiros, clientes e mesmo com o governo. Vivemos num mundo completamente dependente do e-mail.[1]

O e-mail é popular porque o público consegue entender facilmente o conceito do envio de uma mensagem cujo modelo é semelhante ao de uma carta escrita tradicional. E o e-mail "simplesmente funciona"; se você sabe o endereço de e-mail de uma pessoa, é só digitar uma mensagem endereçada a ela e pressionar Enviar. Voilà! Segundos mais tarde, a mensagem está na caixa postal eletrônica dela, esteja ela no apartamento ao lado ou no outro lado do planeta. Da perspectiva do usuário, nada poderia ser mais fácil.

Infelizmente, nem sempre foi assim tão fácil e, mesmo hoje, a infra-estrutura subjacente que torna possível o uso do e-mail em uma escala tão ampla é bastante oneroso. Há vários pacotes de software que você pode executar em um sistema Linux para transportar e gerenciar e-mails (três dos quais serão discutidos mais adiante neste capítulo), mas todos requerem um pouco de configuração e gerenciamento. Além disso, é importante entender os conceitos e os protocolos subjacentes associados com e-mail de modo que você não acabe com a ilusão dos seus usuários de que e-mail interorganizacional para diversas plataformas é um presente dos deuses que funciona de maneira mágica todo o tempo.

Hoje, há alternativas à maneira como entendemos e administramos a infra-estrutura de e-mail. Alguns provedores de serviço agora fornecem serviço de e-mail "gerenciado", em que, na verdade, o e-mail fica hospedado em servidores em um distante centro de dados e é cobrada uma taxa mensal ou anual pelo acesso (possivelmente por usuário). Da mesma forma, alguns serviços "gratuitamente" hospedados, como o Yahoo! Mail, MSN Hotmail e Google Gmail tornaram-se populares. Se você precisa de uma conta pessoal de e-mail ou de uma conta para uma empresa de pequeno porte (bem pequeno), esses podem ser opções viáveis para você. Junto com esses serviços vêm algumas tarefas, incluindo armazenamento, gerenciamento de servidor, atualizações de software, configuração, filtragem de spam e vigilância da segurança, para citar alguns. Em troca do serviço gratuito, talvez você veja alguns anúncios publicitários. Parece um bom negócio em muitos casos; se essa opção funciona para você, pelo menos você terá o benefício de não precisar ler o restante deste capítulo.

Entretanto, e-mail hospedado não é a solução para todo mundo. Negócios e outras grandes empresas cujas operações dependem do serviço de e-mail freqüentemente não podem correr o risco de hospedar e-mail fora do site. Essas empresas talvez tenham várias razões para hospedar seus próprios sistemas de e-mail, inclusive segurança, desempenho e disponibilidade. Este capítulo é para elas.

O grande volume deste capítulo — quase 100 páginas — atesta a complexidade dos sistemas de e-mail. O capítulo contém tanto informações de formação como detalhes de configuração de software, grosseiramente nesta ordem.

Tentamos dividir este capítulo em cinco seções menores (sistemas de e-mail, configuração de **sendmail**, spam, Exim e Postfix), mas isso seria confuso, cheio de problemas do tipo "quem nasceu primeiro: o ovo ou a galinha?" e, acreditamos, menos útil. Em vez disso, oferecemos um guia exibido na Tabela 18.1.

Essa organização torna o fluxo um pouco mais tranqüilo quando o capítulo é lido de uma só vez, mas ela algumas vezes separa os itens relevantes a uma determinada tarefa relacionada a e-mail em particular. O postmaster de uma empresa de médio porte precisaria ler o capítulo inteiro, mas um administrador de sistemas configurando o suporte para e-mail em PCs para um cliente típico certamente não teria de fazê-lo.

---

1. Mesmo quando Evi navega por locais remotos e obscuros, ela quase sempre consegue acessar seu e-mail pelo BlackBerry devido à onipresença da rede GPRS. Essa rede funciona em 113 países, mas não em Vermont!

Tabela 18.1 Um guia para este capítulo.

	Seção	Página	Conteúdo
Informações	1	530	Sistemas de e-mail e seus diversos componentes
	2	534	Endereçamento, sintaxe de endereço, cabeçalhos de e-mail
	3	539	Filosofia, projeto cliente/servidor, diretórios de armazenamento de e-mail
	4	544	Aliases, roteamento de correio, LDAP
	5	551	Software de lista de discussão
Configuração de **sendmail**	6	557	sendmail: instalação, inicialização, fila de e-mails
	7	565	Introdução à configuração de **sendmail**, macros **m4**
	8	570	Elementos básicos da configuração de sendmail
	9	574	Elementos mais sofisticados de configuração de sendmail
	10	588	Spam, banco de dados de acesso **sendmail**
	11	599	Estudo de caso de configuração
	12	603	Segurança
	13	611	Desempenho
	14	615	Coletando estatísticas, testando e depurando
Outros	15	621	Exim, uma alternativa ao **sendmail**
	16	623	Postfix, uma outra alternativa ao **sendmail**
	17	639	Fontes adicionais de informações

A Tabela 18.2 apresenta um guia de navegação para várias tarefas comuns de um administrador de sistemas.

Tabela 18.2 Seções deste capítulo relevantes a várias tarefas.

Tarefa	Seções
Atualizando **sendmail**	6, 7
Configurando **sendmail** pela primeira vez	3, 6, 7, 8, 9, 10, 11, 12
Modificando o arquivo de configuração	7
Projetando um sistema de e-mails para um site	3, 4, 5, 6, 7, 8, 9, 10, 11
Combatendo o spam	10
Auditando a segurança	12
Configurando um PC para receber e-mail	1, 3
Configurando uma lista de discussão	5
Ajustando o desempenho	3, 9, 13
Hospedagem virtual	9
Utilizando o Exim em vez de **sendmail**	15
Utilizando o Postfix em vez do **sendmail**	16

A maior parte deste capítulo lida com a configuração do **sendmail**, o programa padrão que analisa sintaticamente e roteia e-mail. **sendmail** foi originalmente escrito por Eric Allman da Universidade da Califórnia, Berkeley. Há três versões principais: a versão 5, IDA, e a versão 8. Uma versão completamente reprojetada, Sendmail X, foi recém-distribuída como uma versão beta, mas ela ainda não está pronta para uso na produção. (De acordo com o pessoal interno, ela talvez nunca substitua a versão 8.) A versão 5 e o IDA não são mais comumente utilizadas; a versão 8 as substituiu. Neste capítulo discutiremos a versão 8 (mais precisamente, 8.13).

sendmail está sendo desenvolvido comercialmente pela Sendmail, Inc., que também mantém uma versão de código-fonte aberto gratuita. As versões comerciais dispõem de uma ferramenta gráfica para configuração e geração de relatórios e monitoramento centralizados, recursos particularmente úteis em sites de e-mail de grande volume.

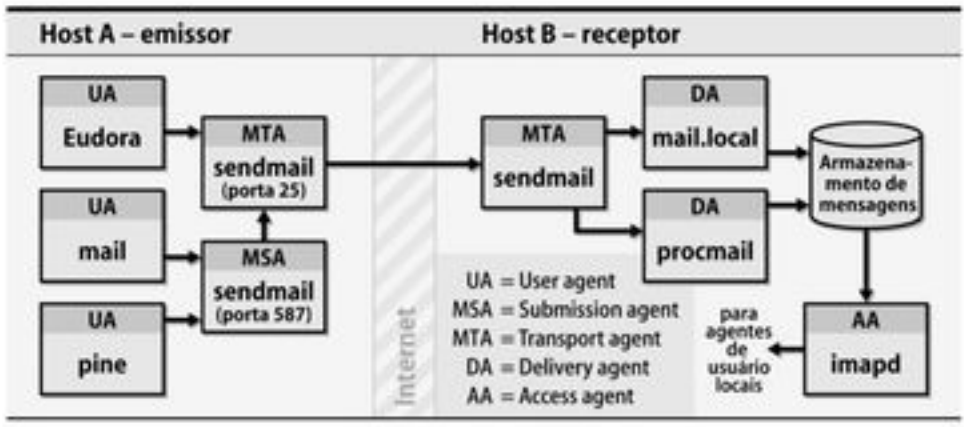
18.1 SISTEMAS DE E-MAIL

Teoricamente, um sistema de e-mail é formado por quatro componentes distintos:

- Um "agente de usuário de correio" (*mail user agent* – MUA) que permite aos usuários ler e compor mensagens
- Um "agente de transporte de correio" (*mail transport agent* – MTA) que roteia mensagens entre as máquinas
- Um "agente de entrega" que coloca mensagens em um armazenamento local de mensagens;[2] ele é às vezes chamado agente local de entrega (*local delivery agent* – LDA)
- Um "agente de acesso" opcional (*access agent* – AA) que conecta o agente de usuário ao armazenamento de mensagens (por exemplo, pelo protocolo IMAP ou POP)

Alguns sites também utilizam um agente de envio de correio que entende o SMTP (Mail Transport Protocol) e faz parte do trabalho do agente de transporte. A Ilustração A mostra o relacionamento desses componentes.

Ilustração A Componentes de um sistema de e-mail



User agent = Agente do usuário; *Submission agent* = Agente de envio; *Transport agent* = Agente de transporte; *Delivery agent* = Agente de entrega; *Access agent* = Agente de acesso.

Agentes de usuário

Os usuários de e-mail empregam um agente de usuário para ler e compor mensagens. As mensagens de e-mail inicialmente eram formadas somente por texto, mas um padrão conhecido como MIME (Multipurpose Internet Mail Extensions) é agora utilizado para codificar formatos de texto e anexos (inclusive muitos vírus) no e-mail. Ele é suportado pela maioria dos agentes de usuário. Como não afeta o endereçamento ou o transporte de e-mail, não trataremos dele neste capítulo.

Uma tarefa dos agentes de usuário é certificar-se de que qualquer texto embutido no conteúdo de uma mensagem de e-mail que possa vir a ser mal interpretada pelo sistema de e-mails se torne protegido. Um exemplo é a string "From" que serve como um separador de registros entre mensagens.

/bin/mail era o agente de usuário original e continua sendo "o bom e velho agente confiável" para ler mensagens de e-mail de texto em um prompt de shell. Para melhor ou pior, e-mail na Internet foi além da era de texto, portanto agentes de usuário baseados em texto não são mais práticos para a maioria dos usuários. As interfaces gráficas com o usuário permitem acesso do tipo apontar e clicar para mensagens e lidam apropriadamente com anexos como imagens, documentos do Microsoft Word e planilhas.

Um dos recursos elegantes ilustrados na Ilustração A é que um agente de usuário não precisa estar necessariamente em execução no mesmo sistema, nem mesmo na mesma plataforma, como o restante do seu sistema de correio. Os usuários poderiam executar um dos muitos agentes de usuários distribuídos com Linux ao se conectarem a uma máquina Linux, mas eles também podem acessar o e-mail por meio dos protocolos de agente de acesso (AA) como o IMAP ou o POP a partir de seus laptops Windows. Essa é, de longe, a configuração mais comum hoje. Quem diz que o Windows e o Linux não podem conviver alegremente?

Um exemplo de agentes de usuário comum é listado a seguir, junto com suas fontes originais:

- **/bin/mail** no Red Hat e Fedora é a versão BSD do comando mail original do UNIX; no SUSE, Debian e Ubuntu, esse agente de usuário encontra-se em **/usr/bin/mail**.[3] Esse agente de usuário é de somente texto e requer um armazenamento de correio local

2. As caixas postais dos usuários receptores ou, algumas vezes, um banco de dados.

- Thunderbird, da Mozilla, para Linux, Windows e Mac OS
- A evolução (conhecida como Novell Evolution, anteriormente Ximian Evolution) para Linux, Windows e Mac OS
- **pine**, da Universidade de Washington, www.washington.edu/pine
- Eudora, da Qualcomm, para Macs ou PCs executando Windows
- Outlook, da Microsoft, para Windows

Agentes de transporte

Um agente de transporte deve aceitar e-mail de um agente de usuário, compreender os endereços dos receptores e de alguma maneira mandar o e-mail para os hosts corretos para entrega. A maioria dos agentes de transporte também atua como agente de envio de mensagens para a introdução inicial de novas mensagens no sistema de e-mail. Os agentes de transporte falam protocolo de transporte de e-mail simples (Simple Mail Transport Protocol – SMTP) definido na RFC2821 (originalmente definido na RFC821) ou o protocolo ESMTP (Extended SMTP) definido nas RFCs 1869, 1870, 1891 e 1985.

Vários agentes de transporte estão disponíveis para os sistemas UNIX e Linux (PMDF, Postfix, **smail**, **qmail**, Exim e **zmailer**, entre outros), porém **sendmail** é o mais abrangente, o mais flexível e o mais amplamente utilizado. Uma pesquisa[4] de 2001 de sistemas de e-mail relatou que o **sendmail** era utilizado por cerca de 60% dos domínios; Exim, por 8%; Microsoft Exchange Server, por 4%; e o Postfix, por 2%. Os demais (cerca de 50 deles) eram irrelevantes.

O Red Hat, o Fedora e o SUSE são distribuídos com o **sendmail** instalado. O Debian finge incluir o **sendmail**, mas se você prestar bastante atenção descobrirá que **sendmail** é realmente um link para o agente de transporte de correio Exim. O Exim foi cuidadosamente elaborado para compreender os flags de linha de comando de **sendmail**. Os agentes de usuário que chamam "**sendmail**" explicitamente para submeterem e-mail não perceberiam a diferença. O Ubuntu distribui o Exim por padrão.

Agentes de entrega

Um agente de entrega aceita e-mail de um agente de transporte e, na verdade, o remete aos receptores locais apropriados. O e-mail pode ser entregue para uma pessoa, para uma lista de discussão, para um arquivo ou até mesmo para um programa.

Cada tipo de destinatário talvez requeira um agente diferente. **/bin/mail** é o agente de entrega para usuários locais. **/bin/sh** é o agente de entrega original para correio sendo transmitido a um programa; transmissão a um arquivo é tratada internamente. Versões recentes do **sendmail** são distribuídas com agentes locais de entrega mais seguros chamados **mail.local** e **smrsh** (pronuncia-se "smursh"). **procmail** em www.procmail.org também pode ser utilizado como um agente de entrega local; consulte a página 405. Da mesma forma, se você executar o Cyrus **imapd** como seu AA, ele incluirá um agente de entrega local próprio.

Armazenamentos de mensagens

O armazenador de mensagens é o ponto na máquina local em que as mensagens de e-mail são armazenadas. Ele costumava ser o diretório **/var/spool/mail** ou **/var/mail**, com os e-mails sendo armazenados em arquivos cujos nomes se baseavam nos nomes de login dos usuários. Esse ainda é o tipo de armazenador de mensagens mais comum, mas os ISPs com milhares ou milhões de clientes de e-mail estão procurando outras tecnologias para o armazenamento de mensagens (bancos de dados, normalmente).

Em sistemas que utilizam o armazenador **/var/spool/mail** ou **/var/mail**, o diretório para e-mail é criado durante a instalação do sistema operacional. Ele deve ter permissões configuradas para o modo 775, com o dono do grupo mail,[5] a menos que você utilize **mail.local** como seu programa de entrega local, em cujo caso o modo pode ser 755. Em plataformas Linux isso varia um pouco:

```
Red Hat:   drwxrwxr-x   2   root   mail   1024 Dec  5 11:16   /var/spool/mail
Fedora:    drwxrwxr-x   2   root   mail   4096 Mar 17 08:42   /var/spool/mail
SUSE:      drwxrwxrwt   2   root   root   4096 Aug  2 23:25   /var/spool/mail
Debian:    drwxrwsr-x   2   root   mail   4096 Aug  3 16:17   /var/mail
Ubuntu:    drwxrwsr-x   2   root   mail   4096 Jan 8 03:22    /var/mail
```

3. Esse agente de usuário algumas vezes é chamado de **Mail** ou **mailx** em outros sistemas. Muito apropriadamente, o Red Hat fornece um link chamado **Mail**, e o SUSE e o Debian fornecem links chamados **Mail** e **mailx**.
4. Estudo privado realizado pela Matrix.net para a Sendmail, Inc.
5. Sistemas que entregam e-mails criando arquivos com um **chown** não-root também devem ter permissão de gravação de grupo no diretório. Em geral, **chown** não-root é uma má idéia.

Consulte a página 58 para informações adicionais sobre o sticky bit.

As permissões do SUSE são um tanto generosas, contudo os arquivos dentro do diretório de spools de correio são modo 660 com grupo root. Diretórios com o bit sticky ativo (o `t` nas permissões) não permitem que usuários eliminem arquivos de outros usuários muito embora eles tenham permissão de gravação no diretório. Entretanto, um usuário mal-intencionado poderia preencher o spool de e-mail, usá-lo como uma partição de rascunho ou criar a caixa postal de outro usuário.

Agentes de acesso

Programas como **imapd** e **spop** são agentes de acesso para usuários de PC, Mac ou Linux cujo correio é entregue a um servidor Linux e, então, o download é feito com o Internet Message Access Protocol (IMAP) ou o Post Office Protocol (POP), respectivamente. O IMAP e o POP são vistos inicialmente na página 375.

Agentes de envio de e-mails

Um novo participante da arena de sistemas de e-mails que era requerido em sites com grande volume de e-mails é o agente de envio. O agente de transporte em um centralizador de e-mails congestionado perde muito tempo pré-processando mensagens de e-mail: garantindo que todos os nomes de hosts estejam totalmente qualificados, modificando cabeçalhos herdados de agentes de usuários de e-mail defeituosos, registrando em log mensagens de erro, reescrevendo cabeçalhos e assim por diante. A RFC2476 introduziu a idéia de separar o agente de envio de e-mail (Mail Submission Agent – MSA) do agente de transporte de e-mail (Mail Transport Agent – MTA) para distribuir a carga de trabalho e maximizar o desempenho.

A idéia é utilizar o MSA, que executa em uma porta diferente, como uma espécie de "recepcionista" para novas mensagens introduzidas no sistema de e-mails por agentes de usuário local. O MSA realiza todo o trabalho preparatório e de verificação de erros que deve ser feito antes de uma mensagem poder ser enviada pelo agente de transporte. É como se estivéssemos inserindo um verificador de sanidade entre o MUA e o MTA.

Em particular, o MSA garante que todos os nomes de hosts estejam totalmente qualificados; ele verifica se os nomes de hosts são legítimos antes de acrescentar a porção de domínio local. O MSA também corrige cabeçalhos de mensagens se estiverem ausentes ou não atenderem aos padrões. Normalmente, o MSA adiciona um cabeçalho From ou Date ou ajusta o cabeçalho Message-Id. Uma tarefa final que um MSA pode fazer é reescrever o endereço do remetente com base em um nome de login para uma forma externa preferencial como *Primeiro_Último*.

Para que esse esquema funcione, os agentes de usuário têm de ser configurados para se conectarem ao MSA da porta 587 em vez da porta 25, que é a porta tradicional para e-mails. Se seus agentes de usuário não forem capazes de utilizar a porta 587, você ainda pode executar um MSA na porta 25, mas em um outro sistema que não seja aquele que executa o seu MTA. O agente de transporte também deve ser configurado para que ele não duplique o trabalho feito pelo MSA. Duplicar processamento não afetará a correção da manipulação de e-mails, mas representa de fato um trabalho extra inútil.

Por padrão, **sendmail** atua como um MSA e também como um MTA. Iniciando com o **sendmail** 8.10, uma única instância do programa escuta tanto na porta 25 como na porta 587. Os agentes de usuário normalmente chamam **sendmail** diretamente com flags que pedem a ele para aceitar uma mensagem de e-mail (**-bs** ou **-bm**) ou sem utilizar nenhum flag, em cujo caso o comportamento de **sendmail** adotará como padrão **-bm**. O processo de **sendmail** monitora a maneira como ele foi chamado e se torna um MSA se chamado com flags **-bs** ou **-bm** ou um MTA se chamado com -bd.

Os agentes de usuário que abrem diretamente uma conexão SMTP têm de ser modificados para utilizar a porta 587 para tirar proveito de um MSA.

18.2 A ANATOMIA DE UMA MENSAGEM DE E-MAIL

Uma mensagem de e-mail apresenta três partes distintas que precisamos entender antes de nos envolvermos com a configuração de **sendmail**.

- O envelope
- Os cabeçalhos
- O corpo da mensagem

O envelope determina onde a mensagem será entregue ou, se a mensagem não puder ser entregue, para quem ela deve ser retornada. Os endereços do envelope geralmente concordam com as linhas From e To do cabeçalho para um único receptor, mas não concordam se a mensagem for enviada para uma lista de discussão. Os endereços são fornecidos separadamente para o MSA. O envelope permanece invisível aos usuários e não faz parte da mensagem em si; ele é utilizado internamente pelo **sendmail** para descobrir aonde enviar a mensagem.

Os cabeçalhos são um conjunto de pares propriedade/valor formatados de acordo com a RFC2822. Eles registram todos os tipos de informações sobre a mensagem, como a data e o horário em que ela foi enviada e os agentes de transporte pelos quais ela passou em sua jornada. Os cabeçalhos são uma parte autêntica da mensagem de e-mail, embora agentes de usuário normalmente ocultem alguns dos menos interessantes ao exibir mensagens para o usuário.

O corpo da mensagem é o conteúdo efetivo a ser enviado. Ele deve ser formado por texto puro, embora esse texto normalmente represente uma codificação segura para e-mails de vários conteúdos binários.

Quando ingressarmos na seção sobre configuração, algumas vezes falaremos do remetente e dos receptores do envelope e outras vezes do remetente e dos receptores do cabeçalho. Tentamos especificar a quais endereços estamos nos referindo se isso não ficar claro pelo contexto.

Endereços de e-mail

O endereçamento local é simples, pois o nome de login de um usuário é um identificador exclusivo. Um endereço Internet também é simples: *usuário@host.domínio* ou *usuário@domínio*. Nos primórdios do e-mail e da Internet, endereços como os mostrados na Tabela 18.3 eram comuns.

Tabela 18.3 Exemplos de tipos de endereços obsoletos

Tipo de endereço	Exemplo de endereço	Forma moderna
UUCP	mcvax!uunet!ucbvax!hao!boulder!lair!evi	evi@lair
Baseado em rotas	<@site1,@site2,...,@siteN:user@final-site>	user@final.site
"Notação com o símbolo de porcentagem"	user%host1%host2@host3	user@host1

Grande parte da complexidade da configuração de **sendmail** provém da antiga exigência de manipular tais endereços. Cada uma dessas formas de endereçamento depende de retransmissão e, graças aos spammers, os sites estão lentamente desativando a retransmissão. A notação com o símbolo de porcentagem (última linha da Tabela 18.3) é uma ferramenta favorita dos spammers que tentam ocultar suas identidades ou retransmitir e-mails por meio de suas máquinas. Se você precisar lidar com qualquer uma dessas formas de endereço, consulte a documentação do **sendmail** ou o livro da O'Reilly sobre **sendmail** para uma ajuda.

Interpretação do cabeçalho de correio

Toda mensagem começa com várias linhas chamadas de cabeçalhos que contêm informações sobre a mensagem. Cada cabeçalho começa com uma palavra-chave como To, From ou Subject, seguida por dois-pontos e o conteúdo do cabeçalho. O formato dos cabeçalhos padrão é definido na RFC2822; mas também são permitidos cabeçalhos personalizados. Qualquer cabeçalho começando com "X-" é ignorado pelo sistema de e-mails, mas é propagado juntamente com a mensagem. Portanto, você pode adicionar um cabeçalho como X-Pegadinha-do-Dia a suas mensagens de e-mail sem interferir na capacidade do sistema de e-mails de encaminhá-las.[6]

Alguns cabeçalhos são adicionados pelo agente de usuário e outros pelo agente de transporte. Vários cabeçalhos rastreiam o caminho de uma mensagem por meio do sistema de e-mails. Muitos dos agentes de usuário ocultam esses cabeçalhos "desinteressantes", mas usualmente há uma opção para fazer com que o agente os revele. Ler cabeçalhos está se tornando uma habilidade importante à medida que somos bombardeados com spam e temos algumas vezes de tentar rastrear uma mensagem em seu caminho inverso até chegarmos à sua origem. Eis o bloco de cabeçalho de uma mensagem simples:

```
From trent Fri, 30 Jun 2006 20:44:49 -0600
Received: from bull.atrust.com (bull.atrust.com [127.0.0.1]) by bull.atrust.com (8.13.1/8.13.1)
   with ESMTP id k612inkG001576 for <ned@bull.atrust.com>; Fri, 30 Jun 2006 20:44:49 -0600
Date: Fri, 30 Jun 2006 20:44:48 -0600
From: trent@atrust.com
Message-Id: <200607010244.k612im9h001575@bull.atrust.com>
To: ned@bull.atrust.com
Cc: steve@bull.atrust.com
Subject: Yonder Mountain

------ body of the message was here ---
```

Essa mensagem permaneceu completamente na máquina local; o remetente era trent e o destinatário era ned. A primeira linha From foi adicionada por **mail.local** que, nesse caso, era o agente de entrega local. As linhas de cabeçalho Subject e Cc foram adicionadas pelo agente de usuário de e-mail de trent, que provavelmente também adicionou os cabeçalhos To, From e Date. **sendmail**, o agente de transporte de correio, adiciona cabeçalhos To, From e Date se eles não são fornecidos pelo MUA. Cada máquina (ou, mais precisamente, cada MTA) que tem contato com uma mensagem adiciona um cabeçalho Received.

6. Tecnicamente, você pode adicionar o cabeçalho que desejar, pois o roteamento de e-mails utiliza apenas o envelope e ignora os cabeçalhos.

Os cabeçalhos em uma mensagem de e-mail fornecem uma grande quantidade de informações relativas aos locais por onde a mensagem esteve, por quanto tempo ela permaneceu nesses locais e quando ela foi finalmente entregue ao seu destinatário. A seguir apresentamos uma análise mais completa de uma mensagem de e-mail enviada pela Internet. Ela é entremeada com comentários que descrevem o propósito dos diversos cabeçalhos e identificam os programas que os adicionaram. Os números de linhas à esquerda são para referência na discussão que segue e não fazem parte da mensagem. Algumas linhas foram reduzidas para permitir que o exemplo coubesse na página.

```
1: From eric@knecht.sendmail.org
```

A linha 1 foi adicionada por **/bin/mail** ou **mail.local** durante a entrega final para separar essa mensagem das demais na caixa postal do usuário receptor. Alguns programas para leitura de e-mails reconhecem os limites da mensagem procurando uma linha em branco seguida pelos caracteres "From"; observe o espaço no final. Essa linha não existe até que a mensagem seja entregue e ela é distinta da linha de cabeçalho "From:". Muitos programas de leitura de e-mails não exibem essa linha, de modo que você poderá não vê-la.

```
2: Return-Path: eric@knecht.Neophilic.COM
```

A linha 2 especifica um caminho de retorno, que pode ser um endereço diferente daquele exibido na linha From: posteriormente no cabeçalho do e-mail. Mensagens de erro devem ser enviadas à linha de cabeçalho Return-Path; ela contém o endereço do remetente do envelope.

```
3: Delivery-Date: Mon, 06 Aug 2001 14:31:07 -0600
```

A linha 3 mostra a data em que o e-mail foi entregue para a caixa postal local de evi. Ela inclui a diferença de fuso horário entre o horário UTC e o horário local (mountain daylight time – MDT).

```
4: Received: from anchor.cs.colorado.edu (root@anchor.cs.colorado.edu [128.138.242.1]) by
    rupertsberg.cs.colorado.edu (8.10.1/8.10.1) with ESMTP id f76KV7J25997 for
    <evi@rupertsberg.cs.colorado.edu>; Mon, 6 Aug 2001 14:31:07 -0600 (MDT)
5: Received: from mroe.cs.colorado.edu (IDENT:root@mroe.cs.colorado.edu [128.138.243.151]) by
    anchor.cs.colorado.edu (8.10.1/8.10.1) with ESMTP id f76KV6418006 for
    <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 14:31:06 -0600 (MDT)
6: Received: from knecht.Neophilic.COM (knecht.sendmail.org [209.31.233.176]) by
    mroe.cs.colorado.edu (8.10.1/8.10.1) with ESMTP id f76KV5Q17625 for
    <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 14:31:05 -0600 (MDT)
7: Received: from knecht.Neophilic.COM (localhost.Neophilic.COM [127.0.0.1]) by
    knecht.Neophilic.COM (8.12.0.Beta16/8.12.0.Beta17) with ESMTP id f76KUufp084340 for
    <evi@anchor.cs.colorado.edu>; Mon, 6 Aug 2001 13:30:56 -0700 (PDT)
```

As linhas 4 a 7 documentam a passagem da mensagem por meio dos diversos sistemas no percurso para a caixa postal do usuário. Cada máquina que manipula uma mensagem de e-mail acrescenta uma linha Received ao cabeçalho da mensagem. Novas linhas são adicionadas na parte superior para que, ao lê-las, você esteja rastreando a mensagem do receptor pelo caminho inverso ao remetente. Se a mensagem que você estiver examinando for um spam, a única linha Received em que você realmente poderá acreditar é aquela gerada pela sua máquina local.

Cada linha Received (Recebida) inclui o nome da máquina remetente, o nome da máquina receptora, a versão de **sendmail** (ou seja lá qual for o agente de transporte utilizado) na máquina receptora, o identificador exclusivo da mensagem enquanto na máquina receptora, o receptor (se houver somente um), a data e o horário e, finalmente, a diferença de fuso horário entre o horário UTC (Universal Coordinated Time, anteriormente chamado de GMT, de Greenwich Mean Time) da zona de horário local. Esses dados são coletados das variáveis de macro internas de **sendmail**. Nos próximos parágrafos, rastreamos a mensagem do remetente ao receptor (percorrendo o caminho inverso, sob o ponto de vista das linhas de cabeçalho).

Consulte a página 276 para informações adicionais sobre registros MX.

A linha 7 mostra que a mensagem foi da interface localhost de knecht (cujo agente de usuário de e-mail particular de Eric escolheu para sua conexão inicial) para a interface externa de knecht por meio do pseudodispositivo de loopback de kernel. A linha 6 documenta o fato de que knecht envia depois a mensagem para mroe.cs.colorado.edu, muito embora a mensagem tivesse sido endereçada a evi@anchor.cs.colorado.edu (veja o cabeçalho da linha 9). Uma rápida verificação com **dig** ou **nslookup** mostra que o host anchor tem um registro MX que aponta para mroe, fazendo com que a entrega seja desviada. A máquina knecht estava usando o **sendmail** versão 8.12.0Beta16.

A máquina mroe estava usando **sendmail** versão 8.10.1 e ela identificou a mensagem com a fila de ID f76KV5Q17625 enquanto se encontrava lá. A seguir mroe encaminhou a mensagem para anchor.cs.colorado.edu conforme endereçado (linha 5), o que pode parecer estranho dado que a transmissão original de knecht se desviou de anchor para mroe devido aos registros MX. A razão para essa aparente inconsistência é que o domínio cs.colorado.edu utiliza uma configuração "DNS dividida". O registro MX para anchor que é visível ao mundo externo aponta para a máquina mestra para e-mails que chegam (mroe). Entretanto, um registro diferente é visto dentro do próprio domínio cs.colorado.edu. A versão interna do registro aponta primeiramente para o próprio anchor e depois para mroe como um backup.

Assim que o e-mail chegou em anchor, ele foi imediatamente encaminhado novamente, dessa vez para rupertsberg. A causa desse salto foi o emprego de aliases (nomes alternativos), um recurso para manipulação de e-mails que é descrito em detalhes a partir da página 377.

Os aliases desempenham um papel importante no fluxo de e-mails. Um alias associa um nome de usuário a uma outra coisa; por exemplo, para o mesmo usuário em uma máquina diferente, para um grupo de usuários, ou até mesmo para uma forma de escrita alternativa do nome de usuário. Você não pode determinar por que a mensagem foi desviada examinando apenas os cabeçalhos de exemplo. Com registros MX, você deve buscar fontes externas de informações.

As linhas Received 5 e 6 incluem a frase "for <evi@anchor.cs.colorado.edu>", que identifica como o e-mail foi endereçado quando ele chegou ao site local. Essas informações auxiliam ao tentar retirar a inscrição de uma lista de discussão que exige que você envie a mensagem de retirada de inscrição (*unsubscribe*) a partir do mesmo host que você fez a assinatura (algumas vezes anos atrás) ou que conheça esse endereço e use-o como um parâmetro na mensagem de retirada de inscrição.

A linha Received final (linha 4) mostra "for <evi@rupertsberg.cs.colorado.edu>"; o valor da macro de endereço de destino de **sendmail** foi alterado pela pesquisa de alias na máquina anchor. O agente de entrega de e-mails local em rupertsberg colocou o e-mail na caixa postal de evi.

```
8: Message-Id: <200108062030.f76KUufp084340@knecht.Neophilic.COM>
```

A linha 8 contém o ID da mensagem, que é diferente de um ID de fila e é exclusivo dentro do sistema de e-mails mundial. Ele é adicionado à mensagem quando ela é enviada inicialmente ao sistema de e-mails.

```
9: To: evi@anchor.cs.colorado.edu
10: From: Eric Allman <eric@Sendmail.ORG>
11: X-URL: http://WWW.Sendmail.ORG/~eric
12: Subject: example message for Evi
13: Date: Mon, 06 Aug 2001 13:30:56 -0700
14: Sender: eric@knecht.Neophilic.COM
```

As linhas 9, 10, 12, 13 e 14 são padrão. Embora um cabeçalho Subject (Assunto) não seja exigido, a maioria dos agentes de usuário o inclui. A linha To (Para) contém o endereço do receptor ou receptores primários. A linha From (De) lista o remetente como eric@sendmail.org; entretanto, as linhas Received listam a máquina remetente como estando no domínio neophilic.com — a máquina knecht de Eric tem vários domínios virtuais presos a ela além de sendmail.org.

A linha Date (Data e hora) mostra a data e o horário em que a mensagem foi enviada. Nesse caso o horário de envio quase coincide com as datas nas linhas Received (Recebida em), muito embora cada uma delas fosse medida com um relógio diferente.

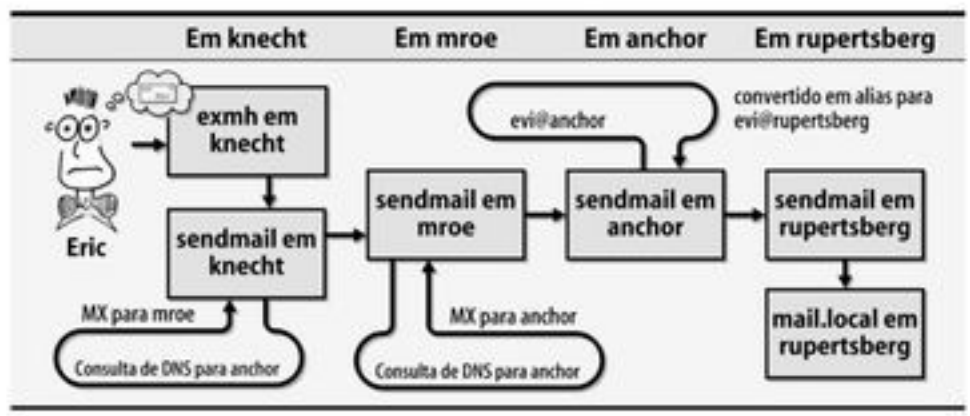
A linha 11 identifica o URL da homepage de Eric. Note que ela começa com um X, tornando-a um cabeçalho não-oficial. Quando o e-mail foi especificado pela primeira vez, não havia coisas como a Web ou os URLs.

As linhas Received normalmente são adicionadas pelo agente de transporte (a menos que elas sejam forjadas) e os demais cabeçalhos são acrescentados pelo agente de usuário. Alguns agentes de usuário são incorretos e não adicionam os cabeçalhos apropriados; nesse caso, **sendmail** entra em ação para adicionar os cabeçalhos ausentes.

A primeira linha Received adicionada (normalmente na máquina remetente, quando o e-mail é transferido para a interface de saída), algumas vezes inclui uma cláusula "ident" que dá o nome de login do remetente. Ele deve ser o mesmo do nome encontrado na linha From, mas não será se a linha From for forjada. Em nosso exemplo, a máquina knecht de Eric não estava executando o daemon que implementa esse recurso (**identd**), de modo que não há nenhuma cláusula que liste o nome de login do remetente.

A Ilustração B demonstra a jornada percorrida por essa mensagem no sistema de e-mails. Ela mostra quais medidas foram tomadas, onde elas aconteceram e quais programas as realizaram.

Ilustração B Uma mensagem de Eric



Como pode ser observado, **sendmail** é o burro de carga nesse processo. Ele trata a mensagem desde o momento em que ela deixa **exmh** em Berkeley até chegar a rupertsberg para entrega.

18.3 FILOSOFIA DO E-MAIL

A filosofia de e-mail que descrevemos neste capítulo é quase obrigatória para manter gerenciável a administração de sites de médio e grande porte. Entretanto, ela também se aplica a sites pequenos. Os principais conceitos que levam a uma administração mais tranqüila são os seguintes:

- Servidores para o recebimento e envio de e-mails; ou, para sites realmente grandes, uma hierarquia
- Uma central de e-mails ("mail home") para cada usuário em um site físico
- IMAP ou POP[7] para integrar clientes remotos, PCs e Macs

Consulte a página 276 para informações adicionais sobre registros MX.

Discutiremos cada uma dessas questões fundamentais a seguir e depois daremos alguns exemplos. Outros subsistemas também devem cooperar com o projeto de seu sistema de e-mails: os registros MX do DNS devem ser configurados corretamente, firewalls de Internet devem permitir a entrada e a saída de e-mails, a(s) máquina(s) para armazenamento de mensagens deve(m) ser identificada(s) e assim por diante.

Os servidores de e-mails têm quatro funções:

- Aceitar e-mails que saem de agentes de usuários e introduzi-los no sistema de e-mails
- Receber e-mails provenientes do mundo externo
- Entregar e-mails nas caixas postais dos usuários finais
- Permitir que os usuários acessem suas caixas postais com IMAP ou POP

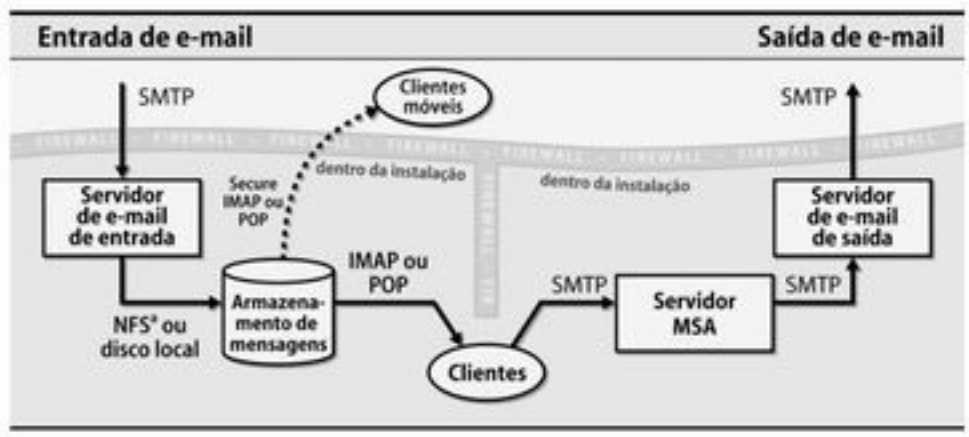
Em um site pequeno, os servidores que implementam essas funções poderiam estar todos na mesma máquina desempenhando funções diferentes. Em sites maiores, eles devem ser máquinas distintas. É muito mais fácil configurar as regras de firewall de uma rede se os e-mails recebidos chegarem somente em uma máquina e os que são enviados parecerem se originar apenas de uma máquina.

Alguns sites usam um proxy para receber e-mails do mundo externo. O proxy realmente não processa e-mails; ele simplesmente os aceita e os joga em spools. Um processo separado encaminha então o e-mail em spool para **sendmail** para transporte e processamento. **smtpd** e **smtpfwdd** de www.obtuse.com são exemplos de tais proxies para **sendmail**; **smtpd** também pode filtrar e-mails recebidos com listas de acesso. Ambos são produtos de código-fonte aberto. Nenhuma das distribuições Linux utilizadas como exemplo neste livro os inclui no pacote de instalação padrão.

Utilizando servidores de e-mail

Escolha máquinas estáveis e confiáveis para seus servidores de e-mails. Aqui, descrevemos o projeto de um sistema de e-mails que parece ser expansível, e relativamente fácil de gerenciar e proteger. Ele centraliza a manipulação tanto de e-mails recebidos como enviados em servidores dedicados para esses propósitos. A Ilustração C demonstra uma forma desse sistema.

Ilustração C Arquitetura do sistema de correio



7. Hoje o IMAP é preferido em relação ao POP. Se fornecer suporte a acesso externo de e-mail na sua empresa, certifique-se de utilizar as versões SSL criptografadas desses protocolos (IMAPS e POPS, respectivamente). Consulte a página 375 para mais detalhes.

a. Não recomendamos utilizar NFS para armazenamento de mensagens devido aos potenciais problemas de bloqueio.

O sistema de e-mails representado na Ilustração C possui um único ponto de exposição para o mundo externo: o servidor de e-mails de entrada que recebe mensagens da Internet. O servidor de e-mails de saída também está conectado diretamente com a Internet, mas ele é menos exposto, pois inicia conexões em vez de aceitar conexões de sites externos. O servidor de e-mails que chegam deve ser monitorado cuidadosamente, deve ser atualizado com correções para problemas de segurança e deve usar a última versão de **sendmail** com filtros contra spam para os e-mails que chegam.

O servidor que manipula e-mails enviados também deve ter uma boa manutenção. Ele pode incluir filtros de spam próprios para verificar se nenhum usuário local está contribuindo para o problema de spam. Se seu site tiver preocupações quanto a vazamento de informações proprietárias, estabelecer um único servidor por meio do qual todos os e-mails enviados devem passar facilita a implementação ou reforça as políticas de conteúdo. Se seu site controla grandes listas de discussão, o servidor para e-mails enviados pode ser configurado para tirar proveito de alguns recursos orientados ao desempenho de **sendmail**; consulte a página 422 para mais detalhes.

Tanto os servidores de e-mails de entrada como os de e-mails de saída podem ser replicados se a carga de e-mails exigir isso. Por exemplo, vários servidores para e-mails de entrada podem ocultar na retaguarda um computador para balanceamento de carga ou utilizar registros MX do DNS para balancear grosseiramente a carga. Entretanto, máquinas clientes diferentes são capazes de direcionar e-mails por meio de servidores de saída. Entretanto, não passe qualquer e-mail diretamente entre os servidores de entrada e de saída; eles devem ser separados uns dos outros por um firewall interno.

Em sites realmente grandes, os servidores de entrada e de saída devem ser replicados. Uma camada extra de roteamento poderia ser adicionada para pesquisar as caixas postais dos usuários (talvez via LDAP) e direcionar os e-mails para o armazenador de mensagens apropriado. A camada de roteamento também poderia executar filtragem de spam e de vírus antes de entregar mensagens às caixas postais dos usuários.

Os ISPs que estão projetando um sistema de e-mails para clientes devem adicionar um outro servidor que atue como o destino dos registros MX de backup dos clientes e trate das listas de discussão. Essa máquina tem de aceitar e-mails e retransmiti-los, mas tem de ser altamente filtrada para garantir que retransmita somente os e-mails de clientes verdadeiros. Ela também deve ser separada dos servidores de e-mails de entrada e de saída por um firewall.

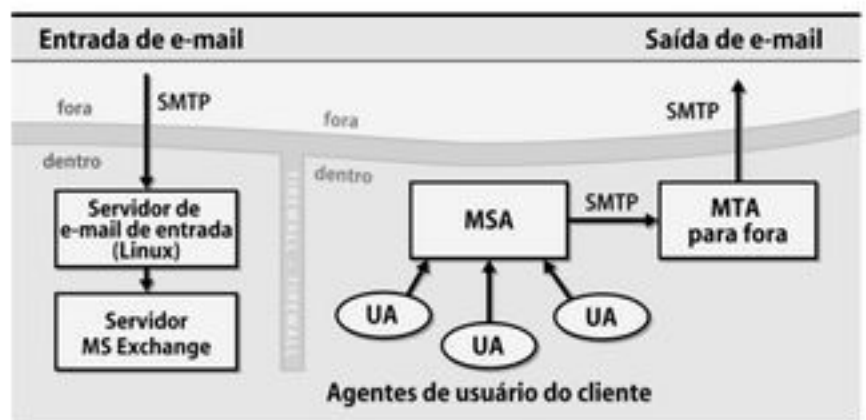
Consulte a página 351 para uma discussão sobre as questões de distribuição de arquivos.

Os hosts Linux comuns podem receber uma configuração mínima para **sendmail** que encaminhe, para processamento, e-mails para o servidor. Eles não precisam aceitar e-mails da Internet. Alguns sites podem querer relaxar um pouco esse modelo de afunilamento e permitir que hosts arbitrários enviem e-mails diretamente para a Internet. Em qualquer um dos casos, as máquinas não servidoras podem compartilhar a mesma configuração para **sendmail**. Talvez você queira distribuir a configuração com uma ferramenta como **rdist** ou **rsync**.

Sites que utilizam software como o Microsoft Exchange e Lotus Notes, mas que não se sentem à vontade para expor diretamente esses aplicativos à Internet, poderão utilizar um projeto que se baseie no modelo apresentado na Ilustração D.

Independentemente do projeto escolhido, certifique-se de que sua configuração **sendmail**, seus registros MX do DNS e suas regras de firewall estão implementando a mesma política em relação ao e-mail.

Ilustração D Arquitetura do sistema de correio — 2ª ilustração



Utilizando centrais de e-mails

É conveniente para os usuários receberem e preservarem suas mensagens de e-mail em uma única máquina, mesmo se eles quiserem acessar essas mensagens a partir de vários sistemas diferentes. Você pode implementar centrais de e-mails utilizando o arquivo **aliases** ou um banco de dados LDAP (consulte a página 384). Você pode fornecer acesso remoto à caixa postal de cada usuário via IMAP ou POP.

O esquema de aliases que utilizamos permite que os arquivos alias sejam os mesmos em todas as máquinas dentro de um domínio administrativo. Essa uniformidade é uma grande conquista do ponto de vista do administrador. (Partimos do pressuposto de que os nomes de login e os UIDs são exclusivos ao longo de todas as máquinas; uma política que recomendamos.)

Alguns sites centralizam e-mails exportando **/var/spool/mail** via NFS. Problemas de bloqueio com arquivos NFS podem fazer com que os usuários percam e-mails ou tenham seus arquivos de spool corrompidos. Culpar o NFS, o **sendmail** ou os agentes de entrega em relação a bloqueio apropriado não ajuda o pobre usuário cuja caixa postal foi corrompida (mas **sendmail** é sempre inocente, já que ele realmente nunca entrega e-mails).

Algumas implementações do NFS (tais como as em servidores NFS dedicados) incluem um gerenciador de bloqueio que funciona perfeitamente. A maioria das implementações não suporta bloqueio ou o faz de forma inadequada. Alguns sites simplesmente ignoram o problema de bloqueio e torcem pelo melhor e outros exigem que os usuários leiam e-mails no servidor de arquivos.

Nosso conselho é não utilizar um **/var/spool/mail** compartilhado por NFS.

Utilizando IMAP ou POP

IMAP e POP são protocolos que copiam ou movem os e-mails para a máquina desktop de um usuário quando ela estabelece conexão com a rede. É a maneira ideal de gerenciar e-mails, especialmente para hosts que nem sempre estão conectados, seja devido a estarem desligados quando não estão em uso ou por estarem em um ambiente doméstico compartilhando a linha telefônica com os adolescentes da casa. Uma senha é exigida; não deixe de utilizar uma versão (IMAPS e POPS) que incorpore a criptografia SSL e, conseqüentemente, não transmita a senha em texto pela Internet.

Gostamos do IMAP, o Internet Message Access Protocol, mais do que do POP. Ele entrega sua correspondência uma mensagem por vez, em lugar de todas de uma só vez, o que é muito mais gentil com a rede (especialmente em um link lento) e melhor para alguém que está sempre se deslocando de um lugar para outro. Ele é especialmente interessante para lidar com os anexos gigantes que alguns colegas adoram enviar: você pode percorrer os cabeçalhos de suas mensagens de e-mail e não fazer a cópia dos anexos até estar preparado para fazê-lo.

O IMAP gerencia pastas de e-mails entre vários sites, por exemplo, entre o seu servidor de e-mails e o seu PC. E-mails que permanecem no servidor Linux podem fazer parte do cronograma normal de backup. www.imap.org contém muitas informações sobre o IMAP e uma lista de implementações disponíveis.

O POP, o Post Office Protocol, é similar, mas pressupõe um modelo no qual é feita a cópia de todos os e-mails do servidor para o PC. Ele pode ser eliminado do servidor (caso em que ele poderá ficar sem backup) ou então salvo no servidor (caso em que seu arquivo spool para e-mail vai crescendo cada vez mais). O paradigma "toda a caixa postal de uma só vez" é difícil na rede e menos flexível para o usuário. Pode ser realmente lento em linhas discadas se você for um destes colecionadores de pequenos itens inúteis e tiver um arquivo spool de e-mail grande. Os e-mails acabam se espalhando com o POP.

Ambos os protocolos podem se tornar devoradores de recursos se os usuários nunca eliminarem mensagens. No caso do IMAP, leva uma eternidade carregar os cabeçalhos de todas as mensagens de e-mail; o POP transmite toda a caixa postal. Certifique-se de que seus usuários compreendem a importância de eliminar mensagens ou de armazená-las em pastas locais.

Uma implementação razoável do POP3, a versão atual desse protocolo, está disponível a partir da Qualcomm em www.eudora.com/products/unsupported/qpopper. O servidor **qpopper** inclui autenticação TLS/SSL entre o servidor e o cliente e criptografa as mensagens.

Você pode encontrar na Internet vários outros servidores POP3 para o Linux; escolha um que seja ativamente mantido.

O software de servidor IMAP se encontra disponível em www.washington.edu/imap. Não é necessária nenhuma configuração, exceto colocar as entradas IMAP adequadas nos arquivos **/etc/services** e **/etc/inetd.conf** e certificar-se de que seu firewall (se houver um) não impede o IMAP de funcionar. O IMAP foi acusado, no passado, de problemas relativos à segurança; consulte os alertas CERT e certifique-se de obter a versão mais recente do IMAP, especialmente se houver boletins de segurança com datas posteriores à da sua distribuição Linux.

A Carnegie Mellon University desenvolveu um servidor IMAP chamado Cyrus IMAP que também suporta o protocolo POP. Gostamos mais dele do que da implementação UW IMAP pelo seu rico conjunto de recursos e desempenho para usuários sofisticados.

O Dovecot é um pacote mais recente que implementa tanto o serviço IMAP como o serviço POP. Ele foi escrito de acordo com diretrizes estritas e explícitas de codificação que melhoram sua segurança, pelo menos teoricamente. O Dovecot também tem alguns recursos interessantes, como a capacidade de armazenar e-mail em um banco de dados SQL em vez de no sistema de arquivos. O Dovecot ainda não tem o mesmo número de adeptos ou base instalada do Cyrus, mas definitivamente é um projeto a ser acompanhado e avaliado.

Todas as distribuições Linux usadas como exemplo neste livro incluem um servidor IMAP chamado **imapd** e um cliente, **fetchmail**, que fala tanto IMAP quanto POP. O **imapd** do Red Hat é o servidor Cyrus IMAP da CMU; o SUSE, Debian e Ubuntu usam o IMAP da Universidade de Washington. O Red Hat também inclui o **pop3d**, um servidor POP. Para não ser ultrapassado, o SUSE inclui três: **qpopper** (que o SUSE rebatizou de **popper**), **pop2d** e **pop3d**. O Debian tem várias ferramentas para gerenciar caixas postais com o IMAP; **man -k imap** informa os nomes dessas ferramentas.

18.4 ALIASES DE E-MAIL

Os aliases permitem que e-mails sejam redirecionados pelo administrador de sistemas ou por usuários individuais.[8] Eles podem definir listas de discussão, encaminhar e-mails entre as máquinas ou permitir que os usuários sejam referidos por mais de um nome. O processamento de aliases é recursivo, de modo que é permitido para um alias apontar para outros destinos que são em si aliases.

sendmail suporta vários mecanismos de nomes alternativos:

- Uma variedade de mapas de arquivos simples que usuários e administradores de sistema podem configurar facilmente (como **/etc/aliases**)
- Sistemas de distribuição de arquivos de legado promovidos pelos fornecedores como NIS e NIS+ da Sun[9] e NetInfo da Apple
- Vários bancos de dados de roteamento de correio
- LDAP (o Lightweight Directory Access Protocol)

Consulte a página 384 para informações adicionais sobre LDAP.

Arquivos simples, como o arquivo **/etc/aliases** (discutido mais adiante nesta seção), são de longe os mais simples e os mais fáceis de configurar para sites de pequeno e médio porte. Se você quiser utilizar o conceito de centrais de e-mail (*mail homes*) e tiver um site grande e complexo, recomendamos que você implemente centrais de e-mail armazenando seus aliases em um servidor LDAP.

Abordamos o LDAP de forma mais detalhada em três locais deste capítulo e também o mencionamos no Capítulo 17, *Compartilhando arquivos de sistema*. Introduzimos e explicamos o LDAP na página 384. Então descrevemos a interação do LDAP com **sendmail** e com o arquivo de configuração do **sendmail** na página 401. Mas antes de nos aprofundarmos no LDAP, primeiro descreveremos os mecanismos tradicionais do sistema de aliases para arquivo simples.

Os aliases podem ser definidos nos três locais a seguir (infelizmente, com três sintaxes diferentes):

- No arquivo de configuração de um agente de usuário (pelo usuário remetente)
- O arquivo **/etc/mail/aliases** no nível do sistema (pelo administrador de sistemas)
- No arquivo de encaminhamento de um usuário, **~/.forward** (pelo usuário receptor)[10]

O agente de usuário procura aliases nos arquivos de configuração do usuário e os expande antes de introduzir a mensagem no sistema de e-mails. O agente de transporte, **sendmail**, procura aliases no arquivo global **aliases** e depois nos arquivos de encaminhamento do receptor. Os aliases são aplicados somente a mensagens que **sendmail** considera como locais.[11]

Eis alguns exemplos de aliases no formato de arquivo **aliases**:

```
nemeth: evi
evi: evi@mailhub
authors: evi,garth,trent
```

A primeira linha diz que o e-mail enviado a "nemeth" deve ser entregue ao usuário evi na máquina local. A segunda linha diz que todos os e-mails para evi devem ser entregues na máquina mailhub e a terceira linha que o e-mail endereçado aos autores ("authors") deve ser entregue aos usuários evi, garth e trent. A recursão é suportada, de modo que o e-mail enviado a nemeth na verdade acaba indo para evi@mailhub.

Consulte o Capítulo 17 para informações adicionais sobre NIS.

O caminho para o arquivo **aliases** global é especificado no arquivo de configuração do **sendmail**. Ele é **/etc/aliases** no Red Hat, SUSE e Ubuntu. **/etc/mail/aliases** é na verdade a localização "padrão". Os sites podem ter vários arquivos **aliases** e eles também podem utilizar maneiras alternativas para armazenar mapeamentos de aliases como NIS ou arquivos de bancos de dados.

O formato de uma entrada no arquivo **aliases** é

```
nome-local: receptor1,receptor2,…
```

onde *nome-local* é o endereço original a ser comparado com as mensagens de entrada e a lista de receptores contém endereços de receptores ou outros aliases. As linhas recuadas são consideradas continuações das linhas anteriores.

Sob o ponto de vista do e-mail, o arquivo **aliases** prevalece sobre **/etc/passwd**, de modo que a entrada

---

8. Tecnicamente, os aliases são configurados somente por administradores de sistemas. O controle do usuário sobre o direcionamento de e-mails por meio de um arquivo **.forward** não é realmente um alias (nome alternativo), mas nós os agrupamos aqui.
9. Suporte a NIS+ foi oficialmente descontinuado pela Sun em 2005. O LDAP é a substituição designada.
10. **~/.forward** é o local padrão que **sendmail** procura. Mas, é possível sobrescrever esse caminho configurando a opção `ForwardPath` de **sendmail**.
11. De fato, isso não é totalmente verdadeiro. Se adicionarmos o flag F=A ao mailer SMTP, você também pode implementar aliases remotos.

```
david: david@somewhere-else.edu
```

impediria que o usuário local david recebesse qualquer e-mail. Portanto, os administradores e as ferramentas **adduser** devem verificar tanto o arquivo **passwd** como o arquivo **aliases** ao selecionarem novos nomes de usuários.

O arquivo **/etc/mail/aliases** sempre deve conter um alias chamado "postmaster" que encaminha e-mails para quem quer que faça a manutenção do sistema. Da mesma forma, um alias para "abuso" é adequado caso alguém de fora da sua empresa precise contatá-lo no que diz respeito a spam ou a um comportamento de rede suspeito que se origina no seu site. Um alias para mensagens automáticas a partir de **sendmail** também deve estar presente; esse normalmente é chamado de Mailer-Daemon e freqüentemente lhe é aplicado o alias postmaster.

Você deve redirecionar os e-mails do root para os administradores de seu site ou para alguém que entre no sistema todos os dias. As contas bin, sys, daemon, nobody e hostmaster (e qualquer outra conta de pseudo-usuário que você configurar) também devem ter aliases que encaminhem e-mails para um ser humano. O arquivo **sendmail/aliases** da distribuição é um bom modelo para os aliases no nível do sistema que devem ser inclusos. Ele também inclui sugestões relativas à segurança e um exemplo de como algumas solicitações de usuário comuns são roteadas na Berkeley.

sendmail detecta loops que poderiam fazer com que o e-mail ficasse indo e voltando para sempre contando o número de linhas Received no cabeçalho de uma mensagem e retornando-a ao remetente quando a contagem atingisse um limite preestabelecido (normalmente 25).[12] Cada visita a uma nova máquina é chamada de "salto" no jargão do **sendmail**; retornar uma mensagem ao remetente é conhecido como "bouncing" ("ricochete"). A sentença anterior, devidamente transformada em jargão, ficaria: "A mensagem de e-mail ricocheteia após 25 saltos".[13]

Além de uma lista de usuários, os aliases podem se referir a:

- Um arquivo contendo uma lista de endereços
- Um arquivo ao qual as mensagens devem ser anexadas
- Um comando para o qual as mensagens devem ser dadas como entrada

Como o remetente de uma mensagem determina totalmente seu conteúdo, esses destinos de entrega eram freqüentemente aproveitados por hackers. **sendmail** se tornou muito exigente em relação à propriedade e às permissões de tais arquivos e comandos. Para superar a paranóia do **sendmail**, você deve configurar uma das opções `DontBlameSendmail`, assim chamadas para desencorajá-lo a fazer isso. Infelizmente, as mensagens de erro que **sendmail** produz ao encontrar permissões ou propriedades inseguras nem sempre são claras.

Obtendo listas de discussão a partir de arquivos

A diretiva `:include:` é um excelente meio para permitir que os usuários gerenciem suas próprias listas de discussão. Ela permite que os membros de um alias sejam extraídos de um arquivo externo em vez de listados diretamente no arquivo **aliases**. O arquivo também pode ser modificado localmente sem a intervenção do administrador de sistemas responsável pelo arquivo global **aliases**.

Ao configurar a lista, o administrador tem de introduzir o alias no arquivo global **aliases**, criar o arquivo incluído e aplicar **chown** neste arquivo incluído para o usuário que mantém a lista de discussão. Por exemplo, o arquivo **aliases** poderia conter:

```
sabook: :include:/usr/local/mail/lah.readers
```

O arquivo **lah.readers** deve estar em um sistema de arquivos local, não em um sistema de arquivos montado com NFS,[14] e ser gravável somente pelo próprio dono. Para ser realmente completo, também devemos incluir aliases para o proprietário da lista de discussão para que erros (ricochetes) sejam enviados ao proprietário da lista, e não ao remetente de uma mensagem endereçada à lista:

```
owner-sabook: evi
```

Consulte a página 382 para obter mais detalhes sobre listas de discussão e a interação destas com o arquivo **aliases**.

---

12. O limite padrão de saltos é de 25, contudo você pode modificá-lo no arquivo de configuração.
13. Não fomos consistentes na terminologia deste capítulo, algumas vezes chamando uma mensagem retornada de "ricochete" e algumas vezes chamando-a de "erro". O que realmente quisemos dizer foi que a notificação de estado de entrega (Delivery Status Notification – DSN) foi gerada. Tal notificação normalmente significa que uma mensagem não pôde ser entregue e que, portanto, ela retornará ao remetente.
14. Se o sistema de arquivos NFS for montado com a opção "hard" e o NFS falhar, **sendmail** irá bloquear com vários manipuladores de arquivos abertos e vários processos em estado de espera. Talvez você acabe sem IDs de processos ou manipuladores de arquivos e tenha de reinicializar a máquina para corrigir as coisas.

Enviando e-mails para arquivos

Se o destino de um alias for um nome de caminho absoluto (entre aspas duplas se ele incluir caracteres especiais), as mensagens serão anexadas ao arquivo especificado. O arquivo já tem de existir. Por exemplo:

```
complaints: /dev/null
```

É útil ser capaz de enviar e-mails a arquivos e programas, mas esse recurso introduz preocupações relativas à segurança e, portanto, é restringido. Essa sintaxe só é válida no arquivo **aliases** e no arquivo **.forward** de um usuário (ou, em um arquivo que é interpolado em um destes arquivos via `:include:`). Um nome de arquivo não é entendido como um endereço normal; portanto, e-mails endereçados a /etc/passwd@host.domínio voltarão.

Alguns agentes de usuário permitem que você salve mensagens de e-mail em um arquivo local (como se fosse uma pasta de caixa de saída). Entretanto, essa cópia da mensagem é salva pelo agente de usuário e nunca é realmente processada pelo sistema de e-mails.

Se o arquivo de destino for referido no arquivo **aliases**, ele deverá permitir a gravação para todo o mundo (o que não é recomendado), terá de ser setuid, mas não executável, ou o seu proprietário terá de ser o usuário padrão de **sendmail**. A identidade do usuário padrão é configurada com a opção `DefaultUser`. Normalmente ela é mailnull, sendmail, daemon ou UID 1, GID 1.

Se o arquivo for referido em um arquivo **.forward**, ele deve pertencer ao receptor da mensagem original e ser gravável por ele, que deve ser um usuário válido com uma entrada no arquivo **/etc/passwd** e um shell válido que é listado em **/etc/shells**. Para arquivos pertencentes a root, use o modo 4644 ou 4600, setuid, mas não executável.

Enviando e-mails para programas

Um alias também pode direcionar e-mails para a entrada padrão de um programa. Esse comportamento é especificado com uma linha como

```
autoftp: "|/usr/local/bin/ftpserver"
```

É ainda mais fácil criar brechas de segurança com esse recurso do que com o envio de um e-mail para um arquivo, portanto, mais uma vez, ele só é permitido nos arquivos **aliases**, **.forward** ou `:include:`, e a configuração padrão do **sendmail** agora requer o uso do shell restringido **smrsh**.[15] No arquivo **aliases**, o programa executa como usuário padrão do **sendmail**; caso contrário, o programa executa como o proprietário do arquivo **.forward** ou `:include:`. Esse usuário deve estar listado no arquivo **/etc/passwd** com um shell válido (um que esteja listado em **/etc/shells**).

O programa entregador local de e-mails (mailer) muda seu diretório de trabalho para o diretório inicial do usuário (ou, se esse diretório for inacessível, para o diretório-raiz) antes de executar o comando que deve receber o e-mail. O padrão costumava ser o diretório de filas do **sendmail**, mas alguns shells baseados em **csh** objetaram.

Criando aliases seguindo um exemplo

Eis alguns aliases típicos que um administrador de sistemas poderia usar.

```
# Aliases requeridos[16]
postmaster: trouble, evi
postmistress: postmaster
MAILER-DAEMON: postmaster
hostmaster: trent
abuse: postmaster
webmaster: trouble, trent
root: trouble, trent

# include para o alias trouble local
trouble: :include:/usr/local/mail/trouble.alias
troubletrap: "/usr/local/mail/logs/troublemail"
tmr: troubletrap,:include:/usr/local/mail/tmr.alias

# conveniências para sysadmin
```

15. Enviar e-mails para programas é uma grande brecha de segurança potencial. Consulte a seção *Segurança e sendmail* na página 417 para informações adicionais sobre **smrsh**.

16. Uma pequena e inofensiva mentira. Somente postmaster e MAILER-DAEMON são realmente requeridos (pelas RFCs), mas é convencional incluir também hostmaster, abuse e webmaster.

```
diary: "/usr/local/admin/diary"
info: "|/usr/local/bin/sendinfo"

# aliases de classe que mudam todo semestre
sa-class: real-sa-class@nag
real-sa-class: :include:/usr/local/adm/sa-class.list
```

Nesse exemplo, gostaríamos que os usuários de todo o campus fossem capazes de enviar e-mails para um único alias "trouble" (problema) sempre que ocorresse um problema. Relatórios de problemas sempre devem ser roteados para um grupo adequado de administradores de redes locais. Particularmente, gostaríamos de configurar aliases de e-mails para que

- E-mails referentes a problemas sempre fossem destinados a um grupo apropriado
- Uma única versão do arquivo de aliases fosse utilizada em todos os hosts
- Grupos de administradores individuais controlassem suas próprias listas de distribuição
- Uma cópia de todos os e-mails referentes a problemas fosse para um arquivo de log local de cada grupo

A configuração anterior satisfaz esses objetivos extraindo a definição do alias para problemas de um arquivo contido em cada máquina. E-mails enviados para os endereços trouble@anchor e trouble@boulder acabariam indo para lugares diferentes muito embora anchor e boulder usem o mesmo arquivo **/etc/mail/aliases**.

E-mails referentes a problemas são normalmente manipulados em uma determinada máquina em cada local. Por exemplo, o arquivo trouble.alias em uma máquina escrava poderia conter o endereço:

```
trouble@mestre
```

para que e-mails referentes a problemas fossem para a máquina mestra apropriada.

Quando uma mensagem referente a problemas é determinada, ela é enviada para o alias tmr, que significa "trouble mail readers" ("leitores de mensagens referentes a problemas"). O alias tmr arquiva a mensagem para o alias de captura de problemas e também a envia a uma lista de usuários extraídos de um arquivo na máquina mestra. Acrescentar administradores novatos à lista tmr é uma excelente maneira de fazê-los observar as questões de suporte que surgem, as respostas dos administradores e o tom bajulador apropriado que deve ser utilizado com os usuários (isto é, com os clientes).

O alias sa-class possui dois níveis, de modo que o arquivo de dados contendo a lista de estudantes só precisa ser mantido em uma única máquina, nag. O exemplo de alias sabook, apresentado antes, deveria realmente ter esse mesmo tipo de indireção para que o arquivo não precisasse ser replicado.

O alias diary é conveniente e funciona bem como uma técnica de extração de documentação para administradores de sistemas novatos que ficam indignados em documentar o que eles fazem. Os administradores de sistemas podem criar memoriais de eventos importantes na vida da máquina (atualizações de sistemas operacionais, mudanças de hardware, quedas etc.) enviando e-mail para o arquivo diary. Não coloque o arquivo em um sistema de arquivos que contenha seus arquivos de logs; isso permitiria que hackers preenchessem o sistema de arquivos e impediria que syslog gravasse suas entradas de log (encobrindo assim seus rastros).

Encaminhamento de correio

aliases é um arquivo de configuração no nível do sistema que deve ser mantido por um administrador. Se os usuários quiserem redirecionar seus próprios e-mails (e o seu site não utilizar POP ou IMAP para acessar e-mail), eles podem fazê-lo criando arquivos **.forward** em seus diretórios iniciais . **sendmail** sempre pesquisa o diretório inicial de um usuário em busca de um arquivo **.forward** a menos que a variável `ForwardPath` esteja configurada e sobrescreva a localização padrão. É conveniente utilizar um arquivo **.forward** quando um usuário quer receber e-mails em um determinado host ou quando alguém deixa o seu site e quer que o e-mail seja encaminhado para uma nova localização.

Um arquivo **.forward** é formado por uma lista de endereços separados por vírgulas em uma única linha ou várias entradas em linhas distintas. Por exemplo,

```
evi@ipn.caida.org
evi@atrust.com
```

ou

```
\mcbryan, "/home/mcbryan/archive", mcbryan@flsupil.gmd.de
```

No primeiro exemplo, os e-mails para evi não são entregues na máquina local, mas encaminhados à máquina ipn no CAIDA em San Diego e para atrust.com. A segunda entrada é de um usuário que não confia nos sistemas de e-mails e quer que seus e-mails sejam replicados em três locais: o spool de e-mails regular na máquina, um arquivo permanente de todos os e-mails que chegam e um endereço temporário na Alemanha por onde ele está viajando no momento. A barra invertida

antes de seu nome de usuário diz para entregar e-mails localmente, independentemente do que os arquivos de aliases ou de encaminhamento possam instruir.

Para mudanças temporárias no roteamento de e-mails, o emprego de um arquivo **.forward** é preferível ao uso do arquivo global **aliases**. O custo adicional (tempo de processamento e tempo das pessoas) necessário para alterar os aliases no nível do sistema é muito elevado.

O arquivo **.forward** de um usuário tem de pertencer ao usuário e não pode ser gravável pelo grupo ou por todo o mundo. Se **sendmail** acredita que o caminho de diretórios para o arquivo **.forward** é seguro (isto é, as permissões a partir da raiz para baixo são boas), ele pode ser um link; caso contrário, ele não poderá ser um link. **sendmail** ignora arquivos de encaminhamento nos quais as permissões pareçam suspeitas; as permissões no diretório pai também têm de ser seguras (graváveis somente pelo usuário ao qual os arquivos pertencem).

Naturalmente, **sendmail** tem de ser capaz de acessar o diretório de origem de um usuário na máquina em que os e-mails são entregues para determinar se ele contém um arquivo **.forward**. Mudanças permanentes de endereço devem ser colocadas no arquivo **/etc/mail/aliases**, pois os arquivos e o diretório de origem de um usuário podem acabar sendo removidos.

sendmail tem um recurso muito interessante, `FEATURE(`redirect')`, que ajuda no gerenciamento de mudanças de endereços de e-mail permanentes. Se um alias apontar para usuário@novosite.REDIRECT, os e-mails serão retornados ao remetente com uma notificação do novo endereço. A mensagem não é encaminhada para o novo endereço, de modo que o remetente terá de atualizar a sua lista de endereços e reenviar a mensagem.

Você poderá configurar **sendmail** para suportar um diretório central para arquivos **.forward**, mas os usuários não esperam essa configuração. A localização dos arquivos **.forward** é controlada pela opção `ForwardPath`, que normalmente aponta para esse diretório central e depois para o diretório de origem do usuário. O arquivo de domínio **generic.m4** ilustrado na página 396 contém um exemplo de local centralizado para arquivos **.forward**.

Uma entrada no arquivo **aliases** global tem precedência sobre uma entrada em um arquivo **.forward**. Como esses arquivos são mantidos por pessoas diferentes, os usuários têm de ser cuidadosos para não criarem inadvertidamente loops de e-mails. Se um usuário de uma rede tiver uma central de e-mail (e, conseqüentemente, uma entrada no arquivo global **aliases**), esse usuário não poderá utilizar um arquivo **.forward** para redirecionar e-mails para uma outra máquina que compartilhe os mesmos aliases. Por exemplo, na Universidade do Colorado, onde usamos um arquivo **aliases** que abrange todo o site, uma entrada como

```
evi: evi@boulder
```

e um arquivo **.forward** na máquina boulder contendo

```
evi@anchor.cs
```

criaria um loop. E-mails endereçados a evi seriam encaminhados a boulder, onde o arquivo **.forward** faria com que eles fossem enviados para anchor no subdomínio cs. O arquivo **aliases** em anchor faria com que eles fossem encaminhados de volta para boulder e assim por diante. Após 25 saltos, os e-mails seriam retornados ao remetente.

Notificar um usuário sobre loop de e-mails é desafiador se seu modo primário de comunicação for e-mail. E-mails para *\usuário*[17] entrega a mensagem na máquina local, independentemente do que o arquivo **aliases** no nível do sistema ou o arquivo **.forward** do usuário possa instruir. Se a máquina local é onde o usuário espera ler os e-mails, tudo bem; caso contrário, envie um e-mail ao postmaster para relatar o loop ou então pegue o telefone!

O banco de dados de hash de aliases

Como as entradas no arquivo **aliases** não se encontram em nenhuma ordem em particular, seria ineficiente para **sendmail** procurar esse arquivo diretamente. Em vez disso, uma versão com hash é construída com o sistema de banco de dados Berkeley DB. Esse hashing aumenta significativamente a velocidade de pesquisas de aliases, especialmente quando o arquivo se torna grande.

O arquivo derivado de **/etc/mail/aliases** é chamado **aliases.db**. Cada vez que você mudar o arquivo **aliases**, terá de reconstruir o banco de dados hashed com o comando **newaliases**. **newaliases** é realmente apenas **sendmail** em forma dissimulada com flags de linha de comando (**-bi**) que o informam para reconstruir o banco de dados. Salve a saída de erro gerada se você executar **newaliases** automaticamente — talvez você tenha introduzido erros de formatação.

---

17. Talvez você tenha de usar duas ou mais barras invertidas para que uma delas passe o shell e vá para **sendmail**.

18.5 LISTAS DE DISCUSSÃO E SOFTWARE RELACIONADO

Uma lista de discussão é um alias gigante que envia uma cópia de cada mensagem postada nela a cada pessoa que se afiliou à lista. É como um Usenet newsgroup de antigamente que é entregue por e-mail. Algumas listas de discussão têm milhares de receptores.

Listas de discussão são normalmente especificadas no arquivo **aliases**, porém mantidas em um arquivo externo. Algumas convenções de nomes padrão são compreendidas pelo **sendmail** e pela maioria dos programas de listas de discussão. Usuários experientes vieram a depender delas também. As mais comuns são o sufixo "-request" e o prefixo "owner-", utilizados para atingir os mantenedores da lista. As convenções são ilustradas pelos seguintes aliases:

```
mylist: :include:/etc/mail/include/mylist
owner-mylist: mylist-request
mylist-request: evi
owner-owner: postmaster
```

Nesse exemplo, mylist é o nome da lista de discussão (*mailing list*). Os membros, então, são lidos no arquivo **/etc/mail/include/mylist**. Ricochetes do mailing para a lista são enviados ao seu proprietário, evi, assim como o são as solicitações para se associar à lista. A indireção de "owner" a "request" para evi é útil, pois o endereço do proprietário (neste caso, mylist-request) se torna o endereço Return-Path em cada mensagem enviada à lista. mylist-request é um pouco melhor do que o real mantenedor para esse campo. Erros em mensagens para o alias owner-mylist (na verdade, evi) seriam enviados para owner-owner.

O caso em que uma mensagem não pode ser entregue é chamado de ricochete. O caso no qual a mensagem de erro enviada sobre o ricochete não puder ser entregue é chamado de ricochete duplo. Portanto, em nosso exemplo, ricochetes duplos são enviados para owner-owner ou postmaster.

Se você usa um arquivo de aliases válido para todo o site, poderá precisar adicionar um nível extra de indireção apontando mylist para myreallist@master para que o arquivo de dados contendo a lista de membros precise existir apenas em um único local.

Pacotes de software para manutenção de uma lista de discussão

Vários pacotes de software automatizam a manutenção de listas de discussão. Eles normalmente permitem que os usuários se acrescentem e se eliminem da lista, obtenham informações sobre a lista e obtenham arquivos via e-mail. Alguns dos gerenciadores de listas de discussão populares (e os locais para sua obtenção) são

- Majordomo, em www.greatcircle.com (incluso no SUSE)
- Mailman, o processador GNU de listas de discussão, em www.list.org
- ListProc, em www.cren.net
- SmartList, derivado do **procmail**
- **listmanager**, em www.listmanager.org
- LISTSERV Lite, em www.lsoft.com (versão gratuita do produto comercial LISTSERV)

Geralmente, SmartList é pequeno e simples, ListProc é grande e complexo e os demais são intermediários. Eles diferem quanto a suas filosofias da manutenção de listas, apresentando alguns uma propensão aos administradores de sistemas como administradores (ListProc), e outros, uma propensão a usuários como mantenedores (Majordomo, Mailman, SmartList, LISTSERV Lite). Majordomo, Mailman, **listmanager** e LISTSERV Lite suportam administração remota; o mantenedor da lista não precisa nem mesmo ter um login na máquina em que a lista se localiza, pois todas as transações ocorrem via e-mail. A maioria dos pacotes para listas de discussão permite que informações postadas à lista sejam montadas em resenhas, algumas de forma automática (ListProc, Mailman, **listmanager** e LISTSERV Lite) e outras via configuração manual (SmartList e Majordomo).

Mailman é o nosso gerenciador de listas favorito. É uma alegria de administrar e permite que mantenedores de listas ajustem todos os recursos de suas próprias listas. ListProc e LISTSERV Lite são proprietários: o primeiro é muito caro, o outro somente binário e enfraquecido. Não experimentamos SmartList, mas gostamos de **procmail**, do qual ele depende.

Descrevemos brevemente cada um desses pacotes a seguir. Para mais detalhes, consulte a documentação de cada pacote ou o livro da O'Reilly *Managing Mailing Lists* de Alan Schwartz e Paula Ferguson.

Majordomo

Majordomo é um pacote Perl/C disponível em www.greatcircle.com. Ele foi originalmente concebido por Brent Chapman. O desenvolvimento do Majordomo parou; Majordomo 2 é uma total reformulação que parece ter fracassado em 2004, assim descrevemos apenas a versão original, que continua em uso comum. Entre as distribuições utilizadas como exemplo neste livro, apenas o SUSE vem com o Majordomo. Apesar do que diz a página man (**/usr/lib/mail/majordomo**), ele se encontra oculto no diretório **/usr/lib/majordomo**.

Consulte a página 417 para informações adicionais sobre usuários confiáveis.

O Majordomo é executado como um usuário não privilegiado, tipicamente com o nome de usuário majordom ou mdom e grupo padrão daemon. Como o Linux suporta nomes de usuários longos (mais de 8 caracteres), você também pode utilizar majordomo como nome de login. O usuário tem de ser um que **sendmail** reconheça como "confiável" e, portanto, tem de ser mencionado em sua configuração **sendmail**, normalmente em uma declaração `confTRUSTED_USERS`.[18]

Majordomo é configurado por meio do arquivo **majordomo.cf**, que é formado por comandos Perl válidos que inicializam variáveis, definem os diretórios nos quais as coisas se encontram (ou onde elas deveriam ser colocadas), especificam as listas a serem suportadas e configuram a manipulação de e-mail ricocheteado. Um programa auxiliar, **config-test**, testará o seu arquivo de configuração em busca de variáveis ausentes ou de sintaxe incorreta. O SUSE coloca o arquivo de configuração em **/etc** e deixa **config-test** com a distribuição Majordomo em **/usr/lib/majordomo**.

Majordomo requer que sejam instalados aliases especiais no arquivo **aliases** do **sendmail**. A maneira mais elegante de integrar esses aliases é criar um arquivo de aliases diferente utilizado apenas para Majordomo (versões recentes de **sendmail** suportam vários arquivos de aliases). O arquivo contém um conjunto de aliases para o próprio Majordomo e um conjunto para cada lista de discussão que ele gerencia. A distribuição contém um modelo de arquivo de aliases, **majordomo.aliases**.

A pergunta mais comum feita pelos usuários sobre as listas de discussão é a seguinte: "Como faço para cancelar minha assinatura?". Para listas gerenciadas pelo Majordomo, a resposta para nome-da-lista@host é enviar um e-mail para o endereço majordomo@host com as palavras "unsubscribe nome-da-lista" ou "unsubscribe nome-da-lista endereço-de-email" no corpo da mensagem (e não na linha de assunto).

Com a primeira forma, você precisa enviar a mensagem de cancelamento de assinatura para o mesmo host que você usou ao assinar a lista; pela segunda forma, esse host faz parte do endereço de e-mail. Consulte a página 408 para dicas sobre como coletar essas informações nos cabeçalhos de e-mail para poder cancelar a assinatura de maneira apropriada, mesmo que você tenha se esquecido de qual máquina usou ao se inscrever na lista. Algumas listas de discussão também aceitam e-mail para nome-da-lista-request@host apenas com a palavra "unsubscribe" no corpo.

Nunca, mas nunca mesmo, envie uma mensagem de cancelamento de assinatura para a própria lista. Se fizer isso, sua mensagem estará patenteando a todos os receptores da lista a sua incompetência.

Mailman

Mailman, um acréscimo relativamente recente à frota de programas para listas de discussão (versão 2.1.8 lançada em abril de 2006), se encontra disponível em www.list.org ou nos arquivos GNU. Ele foi criado originalmente por John Viega e atualmente está sendo desenvolvido em colaboração com Ken Manheimer e Barry Warsaw. Assim como o Majordomo, Mailman é basicamente escrito em uma linguagem de script com empacotadores em C, mas, neste caso, a linguagem é a Python (disponível em www.python.org).

Mailman foi inspirado pelo emprego do Majordomo por parte dos autores e pela frustração proveniente de erros ricochetes, configuração complicada dos recursos avançados como resenhas e listas moderadas e dificuldades de desempenho com grandes volumes de e-mails. Um script do Mailman importa listas do Majordomo. O Mailman também pode detectar e controlar spam até certo ponto.

A grande alegação de Mailman para a fama é sua interface com a Web, que facilita o gerenciamento de uma lista por parte do moderador ou postmaster e também facilita a assinatura, cancelamento de assinatura e configuração de suas opções por parte dos usuários.

ListProc

ListProc é um velho conhecido dos programas para gerenciamento de listas de discussão. Ele foi escrito em 1991 por Anastasios Kotsikonas e mantido até por volta de 1994. Depois permaneceu parado por alguns anos, mas ressuscitou com uma nova versão em 1998. Ele costumava estar disponível gratuitamente no departamento de Ciências da Computação da Boston University, mas com algumas regras de licenciamento um tanto estranhas. Agora pode ser encontrado em www.cren.net por uma taxa de licenciamento pesada (US$ 2.000 por cópia, até mesmo para universidades). Esqueça-se do ListProc e adote um dos pacotes gratuitos e de código-fonte aberto.

SmartList

SmartList foi originalmente escrito por Stephen R. van den Berg, que também é o autor original do **procmail**. Ele está disponível em www.procmail.org. O SmartList usa o **procmail**, portanto você precisará fazer a cópia tanto do **procmail.tar.gz** como do **SmartList.tar.gz**. É provável que seja mais fácil simplesmente pegar o pacote Linux apropriado ao seu sistema.

---

18. Um usuário "confiável" tem permissão para alterar a linha de cabeçalho From de uma mensagem e reconstruir o arquivo **aliases**.

SmartList é compacto e simples. É uma combinação de código C, regras de **procmail** e scripts de shell. Ricochetes, a dor de cabeça das listas de discussão, são tratados de forma automática pelo software. Os usuários são eliminados automaticamente de uma lista após certo número de ricochetes em relação a seus endereços. O SmartList exige uma entrada de login no arquivo **passwd** ("smart" ou talvez "list"), que é um usuário confiável no arquivo de configuração do **sendmail**.

A instalação inclui **led**, um empacotador de bloqueio para edição que tenta proteger SmartList contra ser deixado com um arquivo de configuração incoerente e parcialmente editado.

listmanager

listmanager de Murray Kucherawy é escrito em C e, portanto, é mais rápido do que os pacotes construídos sobre uma linguagem de scripting como o Perl ou o Python. **listmanager** também melhora a velocidade utilizando o pacote de banco de dados DB de sleepycat.com em vez de arquivos simples e do sistema de arquivos. Sua lista de recursos ocupa aproximadamente uma página e inclui uma interface Web, resenhas e várias melhorias de segurança.

listmanager realmente se parece com um gerenciador de listas arrasador — o único ponto negativo é que o código-fonte ainda não está sendo distribuído. De acordo com a página Web em www.listmanager.org, o código está sendo retido até o autor finalizar um trabalho de limpeza para evitar possíveis embaraços. Há um binário Linux disponível.

LISTSERV Lite

LISTSERV Lite de Eric Thomas é uma versão deficiente de LISTSERV, o produto comercial da L-Soft International, Inc. Alguns dos recursos da versão comercial estão ausentes nesta edição gratuita e o software está limitado a gerenciar 10 listas de discussão de até 500 pessoas cada. LISTSERV Lite precisa ser executado como o pseudo-usuário listserv, que deve ser o proprietário dos seus arquivos. Ele também gosta de ter um grupo listserv. LISTSERV Lite dispõe de uma interface Web tanto para assinar uma lista como para sua manutenção.

A distribuição se encontra disponível em www.lsoft.com. O código-fonte não é distribuído, mas os binários pré-compilados e os stubs para diversas versões do UNIX e do Linux são fornecidos. Se você estiver familiarizado com LISTSERV e tiver listas que o usem, talvez seja justificável o emprego de um gerenciador de listas enfraquecido e apenas com binários. Se estiver começando do zero, opte por uma das alternativas de código-fonte aberto e irrestritas já mencionadas.

LDAP: o Lightweight Directory Access Protocol

O LDAP é um protocolo que fornece acesso a um serviço de diretório administrativo genérico. Ele está no mercado há alguns anos, mas só recentemente começou a se tornar popular.

Os administradores estão descobrindo que o LDAP é uma boa alternativa para várias coisas:

- Configuração de **sendmail**: aliases, domínios virtuais, centrais de e-mails, bancos de dados de acesso e tabelas
- Gerenciamento de usuários: nomes de login, senhas, hosts (por exemplo, Stanford University)
- Arquivos de configuração administrativos (por exemplo, SUSE)
- Uma substituição ao NIS
- Um servidor de calendário
- Utilize com módulos de autenticação plugáveis (*pluggable authentication module* – PAM)

É previsto que o LDAP acabe se tornando um sistema de diretórios global utilizado para diversos tipos de propósitos. Infelizmente, ainda faltam ferramentas para automação de tarefas típicas desempenhadas pelos administradores de sistemas com LDAP.

O LDAP nasceu dos protocolos ISO e do serviço de diretórios X.500. Essa ascendência imediatamente sugere os termos complexo, inflado, prolixo, ruim etc., mas o L na sigla LDAP supostamente deve cuidar de tudo isso. As versões 1 e 2 deste protocolo foram padronizadas. A versão 3 está próxima disso. Felizmente, todas as versões são compatíveis com as anteriores. A versão 3 do protocolo LDAP suporta servidores hierárquicos; consultando um servidor para um bloco de dados pode resultar em uma referência a um outro servidor. A versão 2 suportava *dados* hierárquicos, mas os *servidores* hierárquicos são suportados somente na versão 3.

Aliases de e-mail são e, em particular, um caso adequado para o LDAP, especialmente agora que **sendmail** suporta LDAP internamente. **sendmail** pode consultar o servidor LDAP para obter pesquisas de aliases em vez de fazê-las diretamente. O LDAP também é capaz de gerenciar o roteamento de e-mails e domínios virtuais. O suporte LDAP deve ser compilado no binário **sendmail**.

Se estiver procurando uma implementação LDAP, recomendamos o servidor produzido pelo grupo OpenLDAP em www.openldap.org. Esse grupo assumiu e aperfeiçoou o código de um servidor anterior que foi desenvolvido na Universidade de Michigan. Para mais informações sobre software relacionado com LDAP, consulte a página 361.

As entradas de bancos de dados LDAP relembram uma entrada termcap com nomes de variáveis mais longos. Os atributos (nomes de variáveis) no banco de dados LDAP ainda não estão totalmente padronizados e esse fato pode resultar em incompatibilidades entre as diversas implementações.

Os atributos na primeira linha de uma entrada de banco de dados são definidos pelo arquivo de configuração do LDAP. Os exemplos desta seção pressupõem que o daemon de servidor LDAP (**slapd**, no caso do OpenLDAP) foi configurado com um nome root distinto (`rootdn`) de:

```
"cn=root, dc=synack, dc=net"
```

O atributo `dc` aparece duas vezes, pois os valores de componentes de domínios não podem conter um ponto; para expressar o domínio synack.net são necessárias duas entradas. Os demais atributos, ou nomes de variáveis, podem ser o que você quiser. Eles não fazem distinção entre letras maiúsculas e minúsculas. **sendmail** (cujo código procura nomes de atributos específicos e atribui a eles interpretações predeterminadas), o servidor LDAP e o construtor do banco de dados LDAP têm de cooperar e utilizar as mesmas convenções de atribuição de nomes.

Alguns atributos possíveis que podem aparecer na primeira linha de uma entrada de banco de dados (as chaves de banco de dados) são `dn` para um nome de domínio, `dc` para um componente de domínio, `o` para o nome de uma organização, `c` para um nome de país e `uid` para um ID exclusivo (por exemplo, um nome de login).

sendmail utiliza um servidor LDAP de maneira muito parecida com a qual ele usa o servidor de nomes DNS. Ele chama o servidor DNS para converter nomes em endereços IP para que as mensagens possam ser enviadas. Ele usa o servidor LDAP para pesquisar aliases de modo que as mensagens possam ser roteadas para o local correto. Em ambos os casos, as pesquisas passaram de arquivos simples (**hosts** e **aliases**) para bancos de dados, com os servidores gerenciando as consultas.

Como padrão, **sendmail** reconhece os seguintes tags de dados LDAP:

```
mailLocalAddress
mailRoutingAddress
mailHost
```

A versão 8.12 expandiu extensivamente esse conjunto padrão (veja **cf/sendmail.schema**) e também permite que você defina o seu próprio esquema para utilizar a tag LDAP que você quiser.

Eis um exemplo do arquivo **slapd ldap.conf** da implementação OpenLDAP:

```
# LDAP Defaults, ldap.conf file, should be world-readable.
#
BASE       dc=synack, dc=net
HOST       gw.synack.net
PORT       389
```

que suporta entradas de banco de dados da forma

```
dn: uid=jon, dc=synack, dc=net
objectClass: inetLocalMailRecipient
mailLocalAddress: jon@synack.net
mailRoutingAddress: stabilej@cs.colorado.edu
uid:jon
```

Quando `FEATURE(ldap_routing)` tiver sido especificado no arquivo de configuração **sendmail**, o receptor da mensagem é comparado com o campo `mailLocalAddress`. Se for coincidente, o e-mail será redirecionado para `mailRoutingAddress`. A linha `objectClass` deve estar presente — ele vem do rascunho da RFC que define a interação LDAP e sistemas de correio.

No host gw.synack.net, essa entrada de banco de dados corresponde ao alias:

```
jon: stabilej@cs.colorado.edu
```

Entediante, não é mesmo? As entradas de bancos de dados poderiam substituir as entradas típicas no arquivo **aliases** para definição de uma central de e-mails para cada usuário.

Algumas grandes empresas já armazenam informações sobre usuários, como a caixa postal preferida de cada usuário, em um outro banco de dados. Se esse for o caso no seu site, é recomendável escrever alguns scripts curtos para extrair essas informações e adicioná-las ao LDAP. Se sua empresa for de pequeno porte, é melhor permanecer com o arquivo **/etc/mail/aliases** tradicional ou escrever um script para gerar as entradas LDAP corretas a partir da linha de comando.

O arquivo **aliases** também ainda é a melhor maneira para definir listas de discussão (com a diretiva `:include:`). Os programas para lista de discussão normalmente direcionam a mensagem para um script empacotador e a reenviam. Uma consulta LDAP pode retornar aliases que apontam para os arquivos `:include:` ou para programas como **majordomo**, mas ela não pode chamar um programa diretamente.

Na versão 8.12, o LDAP também pode ser usado para armazenar o conteúdo de alguns dos outros dados de **sendmail** (por exemplo, tabelas e classes). Consulte a página 401 para informações adicionais sobre como configurar o **sendmail** para utilizar o LDAP.

18.6 SENDMAIL: O DIRETOR DE ESPETÁCULOS DO CIRCO DO E-MAIL

Todas as três plataformas de referência Linux adotadas neste livro vêm com um agente para transporte de e-mails. O Red Hat, o Fedora e o SUSE são distribuídos com o **sendmail**; o Debian e o Ubuntu incluem o Exim por padrão. Descrevemos rapidamente o Exim a partir da página 429. Muitas das opções de configuração do Exim possuem correspondentes no mundo **sendmail**. Como este capítulo é muito longo, veremos somente **sendmail** de forma detalhada. Descrevemos o Exim em termos dos recursos equivalentes no **sendmail**.

sendmail pode adaptar-se aos caprichos daqueles que criam os padrões graças em parte à flexibilidade de seu arquivo de configuração, que permite que **sendmail** atenda às necessidades de uma comunidade de usuários diversa. O restante deste capítulo é dedicado basicamente ao entendimento e à construção de seu arquivo de configuração, o infame **sendmail.cf**.

sendmail é um agente de transporte, um programa que faz a interface entre os agentes de usuário e os agentes de entrega. Ele fala o protocolo SMTP e entrega mensagens para peers (pares) nas máquinas remotas pela Internet. A lista de tarefas do **sendmail** inclui:

- Controlar mensagens à medida que elas deixam o teclado do usuário
- Compreender os endereços dos receptores
- Escolher um agente de transporte ou de entrega apropriado
- Reescrever endereços para um formato que possa ser compreendido pelo agente de entrega
- Reformatar cabeçalhos conforme a necessidade
- Passar a mensagem transformada para o agente de entrega

sendmail também gera mensagens de erro e retorna mensagens não-entregáveis para o remetente.

Versões do sendmail

Quando escrevíamos este livro, as versões das distribuições dos fornecedores para nossos exemplos do **sendmail** eram derivadas do V8. Entretanto, elas estão em geral uma versão ou duas atrás da principal versão da Sendmail, Inc. Os fornecedores normalmente personalizam uma determinada versão de **sendmail** e, então, ficam relutantes em atualizar seus sistemas base para incluir revisões atuais.

Baseamos nossa discussão de **sendmail** na V8.12 e ignoramos totalmente tanto as versões V5 como a IDA, que são obsoletas. A V8 usa o processador de macros **m4** para permitir uma fácil configuração dos casos padrão. Essa "configuração leve" é tudo do que a maioria dos sites precisa.

Infelizmente, se a sua configuração apresentar algum problema, talvez você tenha de basear a sua depuração em uma compreensão do arquivo de configuração, que temos ouvido ser descrito como inacessível, assustador, enjoado, criptográfico, oneroso, infame, enfadonho, sádico, confuso, tedioso, ridículo, obscuro e enrolado. Falamos um pouco do arquivo de configuração em versões mais antigas deste livro, mas já que sua relevância para administradores modernos se esvaiu, agora indicamos o livro da O'Reilly sobre **sendmail** de Bryan Costales e Eric Allman (intitulado *sendmail*) ou o *Sendmail Installation and Operations Guide*, que é incluído na distribuição **sendmail**.

Novas versões de **sendmail** são lançadas algumas vezes para corrigir problemas relacionados com segurança; sugerimos que você verifique as notas de sua versão ("release notes") em www.sendmail.org e atualize o seu sistema caso tenha perdido qualquer correção referente a questões de segurança. Normalmente, você pode obter pacotes **sendmail** atualizados diretamente do seu distribuidor Linux, mas certifique-se de que o binário recebe os pacotes correspondes à versão atual do sendmail. Se precisar compilar e instalar a distribuição de código-fonte diretamente de www.sendmail.org, você precisará de um compilador C e do pré-processador de macro **m4** (os dois costumam vir com as distribuições Linux).

Algumas vezes é difícil determinar exatamente a verdadeira versão base de **sendmail**, mas se o seu fornecedor não remexeu muito, você será capaz de executar

```
$ /usr/sbin/sendmail -d0.1 -bt < /dev/null
```

fazer com que **sendmail** revele o número de sua versão, as opções que nela foram compiladas e quem ela acredita ser após ler o arquivo de configuração. O flag **-d** configura um nível de depuração (consulte a página 426 para mais informações sobre níveis de depuração em **sendmail**); o flag **-bt** coloca **sendmail** no modo de teste de endereços; e o redirecionamento a

partir de **/dev/null** não dá a ele nenhum endereço para teste. Eis um exemplo de saída gerada (ligeiramente truncada) de um sistema Red Hat:

```
Version 8.11.6
 Compiled with: LDAPMAP MAP_REGEX LOG MATCHGECOS MIME7TO8 MIME8TO7 NAMED_BIND NETINET NETINET6
    NETUNIX NEWDB NIS QUEUE SASL SCANF SMTP TCPWRAPPERS USERDB
============ SYSTEM IDENTITY (after readcf) ============
 (short domain name) $w      =    coyote
 (canonical domain name) $j =    coyote.toadranch.com
 (subdomain name) $m         =    toadranch.com
 (node name) $k              =    coyote.toadranch.com
=========================================================
```

sendmail sempre deve utilizar registros MX (mail exchanger) do DNS e faz desta maneira se compilado com a opção `NAMED_BIND` (como no exemplo anterior). As variáveis de uma letra como `$w` são do arquivo de configuração ou determinadas em tempo de execução.

Instalação de sendmail em sendmail.org

Na versão 8.12, o ambiente de instalação para o **sendmail** mudou um pouco. Ele não mais executa setuid para root, mas, em vez disso, executa setgid para o grupo sendmail, smmsp. Antes de instalar **sendmail**, você tem de criar o usuário smmsp e o grupo smmsp (o nome misterioso significa **sendmail mail submission program**). Utilizamos UID e GID 25 para concordar com o bem conhecido número de porta do protocolo de e-mail SMTP. O usuário smmsp deve ter smmsp como seu grupo de login padrão, que é configurado tipicamente no arquivo **/etc/passwd**. A adição de um usuário e grupo sendmail dedicados permite que **sendmail** seja executado com privilégios reduzidos e aumenta a segurança.

A segunda maior alteração sob o ponto de vista do administrador é que **sendmail** agora usa dois arquivos de configuração: **submit.cf** e **sendmail.cf**. O arquivo **submit.cf** controla a manipulação de e-mail que se origina na máquina local (e está sendo inicialmente enviado para o sistema de e-mails) e **sendmail.cf** controla e-mails que chegam e e-mails colocados em fila durante o processo de envio. **submit.cf** é fornecido com a distribuição e é o mesmo para todos os sites; normalmente não há nenhuma necessidade de personalizá-lo.

Esta seção descreve brevemente o processo de instalação; consulte as notas de instalação na distribuição **sendmail** para detalhes e para questões relacionadas com arquiteturas ou sistemas operacionais em particular. A próxima seção descreve a instalação de **sendmail** em um sistema Debian via **apt-get**. Se estiver substituindo a versão original de **sendmail** do seu sistema, algumas das tarefas de configuração (como instalação dos arquivos de ajuda) talvez já tenham sido feitas para você.

Os envolvidos:

- O binário **sendmail**, normalmente instalado em **/usr/sbin** ou **/usr/lib**.

```
-r-xr-sr-x root   smmsp  /usr/sbin/sendmail
```

- Os arquivos de configuração **/etc/mail/sendmail.cf** e (nas versões 8.12 e posteriores) **/etc/mail/submit.cf**, instalados pelo administrador do sistema:

```
-rw-r--r-- root   bin    /etc/mail/sendmail.cf
-rw-r--r-- root   bin    /etc/mail/submit.cf
```

- Os diretórios de filas de e-mail, **/var/spool/mqueue** e (nas versões 8.12 e superiores) **/var/spool/clientmqueue**, criados pelo administrador do sistema ou pelo processo de instalação:

```
drwxrwx--- smmsp  smmsp  /var/spool/clientmqueue
drwx------ root   wheel  /var/spool/mqueue
```

- Vários links para o binário de **sendmail** (**newaliases**, **mailq**, **hoststat** etc.)[19]
- Agentes de entrega locais mais seguros do **sendmail**, **smrsh** e **mail.local** (normalmente instalados em **/usr/libexec**)

Você pode fazer a cópia da versão mais recente de **sendmail** no endereço www.sendmail.org. Para compilar e instalar o pacote, siga as instruções contidas no arquivo **INSTALL** no diretório de base. Comece adicionando o usuário e o grupo smmsp; não dê um shell verdadeiro a esse usuário. Eis uma entrada **/etc/passwd** típica:

---

19. Tome cuidado aqui. Alguns fornecedores utilizam links físicos ("hard"), cuja conseqüência é que, ao fazer uma atualização, talvez você acabe terminando com uma versão de **sendmail** não correspondente à versão de **newaliases**, criando problemas sutis e difíceis de serem encontrados.

```
smmsp:x:25:25:Sendmail Mail Submission Prog:/nonexistent:/bin/false
```

E eis uma entrada **/etc/group** típica:

```
smmsp:*:25:smmsp
```

Para compilar o software, mude para o diretório de distribuição de **sendmail**, execute o script **Build** e a seguir execute **Build install**. O arquivo **devtools/OS/Linux** contém os procedimentos utilizados para gerar o **sendmail** em sistemas Linux. As distribuições Linux não padronizaram onde as coisas devem se encontrar, de forma que **devtools/OS/Linux** é um local provável, mas que talvez não seja exatamente correto para o caso da sua distribuição.

Entretanto, antes de começar a compilar, você deve decidir por um formato de banco de dados e uma estratégia para interfacear **sendmail** com bancos de dados administrativos como o NIS. Para bancos de dados instalados em disco, recomendamos o pacote Berkeley DB especificado no **Makefile** como `NEWDB` (o padrão).

Para personalizar o Makefile, não o edite; crie o seu próprio arquivo **site.config.m4** e coloque-o no diretório **devtools/Site** para ajustá-lo ao seu sistema operacional e ambiente local. Por exemplo, se você pretende utilizar o LDAP e a nova biblioteca para filtragem de spam, crie nesse diretório um arquivo **site.config.m4** contendo as linhas

```
APPENDDEF(`confMAPDEF', `-DLDAPMAP')
APPENDDEF(`confLIBS', `-lldap -llber')
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
```

Um `define` substitui a definição atual de um atributo; a macro `APPENDDEF` anexa à definição atual.

Compilar **sendmail** com

```
$ sh ./Build
```

inclui automaticamente as entradas específicas do site. Para instalar **sendmail** no local apropriado, execute

```
$ sudo sh ./Build install
```

sendmail normalmente *não* deve ser configurado para ser controlado por **inetd/xinetd**, de modo que ele precisa ser iniciado explicitamente nos arquivos **rc** em tempo de inicialização. Uma seqüência típica ficaria parecida com a seguinte:

```
if [-f /usr/sbin/sendmail -a -f /etc/mail/sendmail.cf]:
then
    (cd /var/spool/clientmqueue; rm -f [tTx]f*)
    (cd /var/spool/mqueue; rm -f [tTx]f*)
    /usr/sbin/sendmail -bd -q30m ### queue runner for regular queue
    /usr/sbin/sendmail -Ac -q30m &       ### queue runner for client queue (8.12)
    echo -n ' sendmail' > /dev/console
fi
```

Essas linhas verificam a existência do binário **sendmail** e seu arquivo de configuração e depois iniciam o programa no modo daemon. O pacote **sendmail** para sua distribuição Linux deve fornecer um script de inicialização adequado que resida em **/etc/init.d**.

Vários agentes de usuário executam **sendmail** explicitamente (algumas vezes com os flags **-bm** ou **-bs**) quando eles enviarem a mensagem de um usuário ao sistema de e-mails em vez de falarem o protocolo SMTP diretamente. Nessa situação, o **sendmail** utiliza o arquivo de configuração **submit.cf** e coloca as mensagens na fila **/var/spool/clientqueue**. Chamar **sendmail** com os flags **-Am** ou **-Ac** força as mensagens para **mqueue** ou **clientqueue**, respectivamente.

O Red Hat e o Fedora têm um script de inicialização de **sendmail** (**/etc/init.d/sendmail**) que não limpa o lixo dos diretórios **mqueue** (como demonstrado no exemplo anterior). Entretanto, ele realmente reconstrói o banco de dados. Com o **sendmail** 8.12 e versões posteriores, o LDAP pode ser utilizado para mapas de bancos de dados, e reconstruir os mapas após alterações não é necessário. Parâmetros definidos em **/etc/sysconfig/sendmail** determinam se **sendmail** deve ser iniciado no modo daemon e com que freqüência ele deve executar a fila, assim, é lá que você deve configurar o comportamento de inicialização de **sendmail** para o seu site. O Red Hat, conforme distribuído, inicia **sendmail** no modo daemon e executa a fila a cada 30 minutos.

O script de inicialização do SUSE (**/etc/init.d/sendmail**) só verifica o binário e o arquivo de configuração e, então, inicia **sendmail** com os argumentos definidos na variável de ambiente SENDMAIL_ARGS; estas são configuradas no arquivo **/etc/rc.config.d/sendmail.rc.config**. O SUSE também assume o padrão do modo daemon e executa a fila a cada 30 minutos.

Adicione o trecho **sh** anterior (ou um como ele) a **/etc/init.d/sendmail** se sua distribuição Linux não incluir um script de inicialização **sendmail**. Um script mais exagerado no guia de instalação tenta limpar as execuções de filas previamente interrompidas. Misture e componha à vontade.

Historicamente, os arquivos de suporte de **sendmail** ficavam perambulando em torno do sistema de arquivos em destinos glamorosos como **/usr/lib**, **/etc**, **/usr/ucblib** e **/usr/share**. Espera-se que, com a versão 8.10 de **sendmail**, todos os arquivos (exceto os diretórios de filas) sejam mantidos sob o mesmo diretório **/etc/mail.**[20] Vamos esperar que os fornecedores aceitem a dica e os deixem em um único local consistente.

Instalação do sendmail em sistemas Debian e Ubuntu

Você pode utilizar o programa **apt-get** para instalar o **sendmail**. O pacote instala **sendmail** e **m4** e também desinstala o Exim. Depois que o **apt-get** fez o download e instalou o pacote **sendmail**, ele o abre para configuração. Responder yes chamará um script que lhe fará algumas perguntas (cerca de 20) sobre a configuração desejada para **sendmail**. Os padrões são razoáveis na maioria das vezes. A única pergunta que você realmente deve responder de forma diferente do valor padrão é "mail name"; a resposta padrão é o nome de host não completamente qualificado (por exemplo, lappie), mas ele precisa ser o nome completamente qualificado (por exemplo, lappie.toadranch.com).

Responder às perguntas feitas pelo script fará mais sentido se você já estiver familiarizado com as opções e os recursos de **sendmail**, que são discutidos de forma mais detalhada mais adiante neste capítulo. O script de configuração não leva em consideração o conteúdo do arquivo de domínio genérico que ele inclui como padrão. Conseqüentemente, você pode não aceitar um recurso, mas acabar com ele ativado de qualquer maneira (por exemplo, o recurso `redirect`).

O arquivo de comutação

O switch de serviço é abrangido em mais detalhes no Capítulo 17.

Muitas distribuições Linux têm um arquivo de configuração de "comutação de serviço", **/etc/nsswitch.conf**, que enumera os métodos que podem satisfazer várias consultas padrão como pesquisas de usuário e de host. Se mais de um método de resolução for listado para um determinado tipo de consulta, o comutador de serviços também determinará a ordem na qual os métodos serão consultados.

O emprego da comutação de serviços normalmente é transparente para o software. Entretanto, **sendmail** gosta de exercer um controle mais refinado sobre as pesquisas, de modo que ele atualmente ignora o arquivo de comutação do sistema e usa seu próprio arquivo interno (**/etc/mail/service.switch**) em seu lugar.

Dois campos no arquivo de comutação causam impacto no sistema de e-mails: `aliases` e `hosts`. Os possíveis valores para o serviço de hosts são `dns`, `nis`, `nisplus` e `files`. Para aliases, os valores possíveis são `files`, `nis`, `nisplus` e `ldap`. Devem ser compilados no suporte **ao sendmail** para todos os mecanismos utilizados (exceto `files`) antes de o serviço poder ser utilizado.

O arquivo de comutação de serviços interno de **sendmail** contém o seguinte:

```
aliases    files nisplus nis     # if compiled with nis/nis+
hosts      dns nisplus nis files
```

Modos de operação

Você pode executar o **sendmail** de diferentes maneiras, selecionado com o flag **-b**. **-b** significa "be" ou "become" e sempre é utilizado com um outro flag que determina o papel que o **sendmail** desempenhará. A Tabela 18.4 lista os valores válidos.

Tabela 18.4 Flags em linha de comandos para os principais modos do sendmail

Flag	Significado
-bd	Executar no modo daemon, escutar conexões na porta 25
-bD	Executar no modo daemon, mas em primeiro plano em vez de no segundo plano[a]
-bh	Ver as informações de conexão recentes (o mesmo que **hoststat**)
-bH	Eliminar a cópia de disco de informações de conexões desatualizadas (o mesmo que **purgestat**)
-bi	Inicializar aliases (o mesmo que **newaliases**)
-bm	Executar como entregador de e-mail; entregar e-mails da maneira usual (padrão)
-bp	Imprimir a fila de e-mails (o mesmo que **mailq**)
-bP	Imprimir o número de entradas nas filas via memória compartilhada (versões 8.12 e posteriores)
-bs	Entrar no modo servidor SMTP (na entrada padrão, e não na porta 25)
-bt	Entrar no modo de teste de endereços
-bv	Verificar somente endereços de e-mail; não enviar e-mails

a. Esse modo é utilizado para depuração, para você poder ver mensagens de erro e de depuração.

20. Bem, isso não é totalmente verdadeiro apesar de todos os arquivos serem mantidos sob **/etc/mail**. O arquivo **sendmail.pid** e algumas vezes o arquivo de estatísticas ainda são mantidos em algum outro local.

Se você espera que os e-mails provenham da Internet, execute **sendmail** no modo daemon (**-bd**). Nesse modo, **sendmail** escuta a porta de rede 25 e aguarda por trabalho.[21] Em geral, você também especificará o flag **-q** — ele configura o intervalo no qual **sendmail** processa a fila de e-mails. Por exemplo, **-q30m** executa a fila a cada trinta minutos e **-q1h** a executa de hora em hora.

sendmail em geral tenta entregar as mensagens imediatamente, salvando-as na fila apenas por um tempo para garantir confiabilidade. Mas se o seu host for muito ativo ou a máquina de destino estiver inacessível, **sendmail** colocará a mensagem em fila e tentará enviá-la mais tarde. **sendmail** costumava utilizar um processo filho cada vez que processava a fila, mas agora ele suporta executores de filas persistentes que normalmente são inicializados em tempo de inicialização. A RFC1123 recomenda pelo menos 30 minutos entre as execuções. **sendmail** realiza bloqueio, de modo que várias execuções de filas simultâneas são seguras.

O **sendmail** 8.12 adicionou um novo recurso para ajudar a lidar com grandes filas e listas de discussão: grupos de filas com divisão de envelope. Isso é visto de forma mais detalhada a partir da página 423.

sendmail lê o seu arquivo de configuração, **sendmail.cf**, apenas ao inicializar. Portanto, você tem de abortar e reiniciar **sendmail** ou, então, enviar-lhe um sinal HUP ao alterar o arquivo de configuração. **sendmail** cria um arquivo **sendmail.pid** que contém o ID de processo e o comando que o iniciou. Você deve iniciar **sendmail** com um caminho absoluto, pois ele se auto-executa novamente ao receber o sinal HUP. O arquivo **sendmail.pid** permite que o processo receba o sinal HUP via:

```
# kill -HUP `head -1 sendmail.pid`
```

A localização do arquivo PID era inicialmente um parâmetro definido em tempo de compilação, mas agora ele pode ser configurado no arquivo de configuração **.mc** pela opção `confPID_FILE`.

```
define(confPID_FILE, `/var/run/sendmail.pid')
```

O valor padrão é dependente do sistema operacional, mas normalmente é **/var/run/sendmail.pid** ou **/etc/mail/sendmail.pid**. As distribuições Red Hat, Fedora, Ubuntu e Debian utilizam **/var/run/sendmail.pid** e o SUSE o mantém no diretório **/var/run/sendmail**.

A fila de e-mails

As mensagens de e-mail são ordenadas no diretório de filas quando a máquina está muito ocupada para poder entregá-las imediatamente ou quando a máquina de destino se encontra indisponível. **sendmail** serve como um agente de envio de e-mails escutando na porta 587, bem como cumprindo seu papel tradicional, ou seja, como um daemon escutando na porta 25 e utilizando a fila **/var/spool/mqueue**. Alguns agentes de usuário (**/bin/mail**, por exemplo) usam a porta 587, a porta de envio de correio, mas outros (Eudora, Outlook etc.) utilizam o SMTP para falar diretamente com o **sendmail** em execução na porta 25. Começando com a versão 8.12, programas de envio de correio injetam novas mensagens no sistema de correio utilizando o diretório de fila **/var/spool/clientmqueue** e o arquivo de configuração **submit.cf**. Todas as mensagens entram na fila logo depois que elas chegam.

sendmail permite mais de uma fila de correio e você pode identificar subconjuntos das filas que pertencem a um grupo de filas. Por exemplo, se o diretório **mqueue** contivesse os subdiretórios **q1**, **q2** e **q3** e você especificasse o diretório de fila como **/var/spool/mqueue/q***, todas as três filas seriam então utilizadas. A capacidade do **sendmail** de tratar múltiplas filas melhora o desempenho sob cargas altas.[22] Se um site estiver executando uma grande lista de discussão, **sendmail** divide a lista de destinatário de envelope em várias listas menores e as atribui a diferentes grupos de fila. Esse truque pode melhorar muito o desempenho, pois as listas de receptores menores podem ser processadas em paralelo.

Os grupos de filas são novos na versão 8.12 e dão um controle mais refinado sobre tipos de mensagens individuais. Qualquer um dos parâmetros associados a filas também pode ser configurado em um determinado grupo de filas, incluindo a prioridade de execução (com a chamada de sistema **nice**). Os e-mails são submetidos a um grupo de filas de acordo com o endereço do primeiro receptor da mensagem. O grupo de filas padrão chama-se `mqueue` e é definido automaticamente e disponível para uso sem configuração adicional. Grupos de filas são abordados em detalhes na página 423.

Quando uma mensagem é colocada em fila, ela é salva em partes nos vários arquivos diferentes. Cada nome de arquivo possui um prefixo de duas letras que identifica a parte, seguido de um ID aleatório formado do ID de processo de **sendmail**. Esse ID não é fixo, pois **sendmail** está criando processos constantemente e cada cópia obtém um novo ID de processo. A Tabela 18.5 ilustra as seis partes possíveis.

Se os subdiretórios **qf**, **df** ou **xf** existirem em um diretório de filas, esses trechos da mensagem são colocados no subdiretório apropriado. A fila **qf** contém não apenas o cabeçalho da mensagem, como também os endereços dos envelopes, a data em que a mensagem deve ser retornada como não-entregável, a prioridade da mensagem na fila e a razão pela qual a mensagem está na fila. Cada linha começa com um código de uma só letra que identifica o restante da linha.

---

21. As portas que **sendmail** escuta são determinadas por `DAEMON_OPTIONS`; a porta 25 é o padrão.

22. Diretórios são um mecanismo de armazenamento eficiente se não contiverem muitos arquivos. Se você tiver um servidor de e-mails com muitas listas de discussão que se tornam defasadas, o diretório de filas pode facilmente se tornar tão grande a ponto de não mais ser possível tratá-lo com eficiência.

Tabela 18.5 Prefixos para arquivos na fila de e-mails

Prefixo	Conteúdo dos arquivos
qf	O cabeçalho da mensagem e o arquivo de controle
df	O corpo da mensagem
tf	Uma versão temporária do arquivo **qf** enquanto este estiver sendo atualizado
Tf	Significa que ocorreram 32 ou mais tentativas falhas de bloqueio
Qf	Significa que a mensagem ricocheteou e não pôde ser retornada
xf	Arquivo de transcrição temporário das mensagens de erros dos entregadores de e-mail

Cada mensagem colocada na fila deve ter um arquivo **qf** e **df**. Todos os demais prefixos são usados por **sendmail** durante uma tentativa de entrega. Quando uma máquina trava e reinicializa, a seqüência de inicialização para **sendmail** deve eliminar os arquivos **tf**, **xf** e **Tf** de cada diretório de filas. O administrador de sistemas responsável pelos e-mails deve verificar ocasionalmente a existência de arquivos **Qf** no caso de a configuração local estar provocando os ricochetes.

A fila de e-mails abre diversas oportunidades para as coisas darem errado. Por exemplo, o sistema de arquivos pode ficar lotado (evite colocar **/var/spool/mqueue** e **/var/spool/news** na mesma partição), a fila pode se tornar obstruída, e mensagens de e-mails órfãs podem ficar emperradas na fila.

sendmail tem uma opção de configuração (`confMIN_FREE_BLOCKS`) para auxiliar a gerenciar o espaço em disco. Quando o sistema de arquivos contendo a fila de e-mails ficar muito cheio, os e-mails serão rejeitados com uma mensagem de erro "try again later" (ou seja, tente mais tarde) até que haja mais espaço disponível. Essa opção deixa um pouco de espaço sobrando de modo que os e-mails começam a ser rejeitados antes de o sistema de arquivos ficar totalmente cheio.

Consulte a página 276 para informações adicionais sobre registros MX do DNS.

Se um centralizador de e-mails importante cair, seus sites backup MX poderão ficar sobrecarregados com milhares de mensagens. **sendmail** poderá criar muitas cópias de si mesmo e derrubar uma máquina. Várias opções ajudam no desempenho em máquinas muito ativas; reunimos essas na seção sobre desempenho a partir da página 422. Para manipular uma fila temporariamente obstruída antes da versão 8.12, você tinha que mover a fila obstruída, continuar processando novos e-mails como de praxe e executar uma cópia distinta de **sendmail** na fila obstruída depois que as coisas se acalmassem. Por exemplo, o procedimento para tratar um único diretório de filas seria:

```
# kill `head -1 sendmail.pid`
# mv mqueue cloggedqueue          /* Para outra FS se necessário */
# mkdir mqueue                    /* Configura proprietário/permissões também */
# chown root mqueue
# chmod 700 mqueue
# /usr/sbin/sendmail -bd -qlh &
```

Quando as coisas se acalmarem, execute **sendmail** com os seguintes flags:

```
# /usr/sbin/sendmail -oQ/var/spool/cloggedqueue -q
```

Esses flags apontam **sendmail** para o diretório da fila obstruída e especificam que **sendmail** deve processá-lo imediatamente. Repita esse comando até que a fila se esvazie. A partir da versão 8.12, **sendmail** usa hard links em modos que irão quebrar caso você desloque uma fila. Uma melhor maneira de lidar com filas obstruídas é utilizar uma máquina de reserva (*fallback*) e registros MX; consulte a seção sobre desempenho para mais detalhes.

O ponto no qual a fila se torna obstruída depende do site e do hardware no qual **sendmail** está sendo executado. O seu sistema e o centralizador de e-mail para aol.com, que processa milhões de mensagens por dia, terão visões diferentes do que é uma fila. Consulte a página 425 para informações sobre como medir níveis de tráfego.

18.7 CONFIGURAÇÃO DE SENDMAIL

Antes da versão 8.12, as ações do **sendmail** eram controladas por um único arquivo de configuração, **/etc/mail/sendmail.cf** (ele era encontrado anteriormente em **/etc** ou **/usr/lib**). A versão 8.12 introduziu uma segunda instância do arquivo de configuração chamada **submit.cf** (também no diretório **/etc/mail**). Os flags com os quais o **sendmail** é iniciado determinam o arquivo de configuração que ele utiliza: **-bm**, **-bs** e **-bt** utilizam **submit.cf** se ele existir e todos os outros modos usam **sendmail.cf**. Naturalmente, alguns flags de linha de comando e opções de arquivo de configuração mudam os nomes dos arquivos de configuração, mas é melhor deixar os nomes como estão. O arquivo de configuração determina o seguinte para **sendmail**:

- Escolha de agentes de entrega
- Regras para reescrita de endereços

- Formatos de cabeçalhos de e-mails
- Opções
- Precauções quanto à segurança
- Resistência a spams

O formato do arquivo de configuração foi concebido para ser fácil de analisar sintaticamente. Esse foco o deixou um tanto carente em termos de recursos amigáveis. A manutenção do arquivo de configuração é a tarefa administrativa mais importante relacionada com os e-mails e aterroriza até mesmo os administradores de sistemas mais experientes.

Toda versão de **sendmail** utiliza um arquivo de configuração, porém as versões modernas facilitam o processo de configuração pelo emprego de macros **m4**, que ocultam grande parte da complexidade subjacente. Pode-se dizer que o arquivo de configuração bruto se encontra em nível de linguagem assembly, ao passo que a configuração via **m4** está mais para o nível da programação Perl.[23]

Quando as macros **m4** foram introduzidas pela primeira vez, esperava-se que elas fossem capazes de manipular de 80% a 90% dos casos. De fato, a taxa de cobertura acabou sendo até maior, provavelmente perto dos 98%. Neste livro, abordamos somente o "arquivo de configuração leve" baseado em macros **m4**. Você precisará se aprofundar no arquivo de configuração de baixo nível somente se estiver depurando um problema espinhoso, ampliando o seu site de e-mail de modo bizarro ou executando um centralizador de e-mail de altíssimo volume.

Três documentos importantíssimos são o livro da O'Reilly, *sendmail*, de Bryan Costales e Eric Allman, o artigo *Sendmail Installation and Operations Guide*, de Eric Allman (incluso no diretório **doc/op** da distribuição), e o arquivo **README** (no diretório **cf**). Normalmente nos referimos a *sendmail* como uma fonte para informações adicionais e nos referimos a ele como "o livro do **sendmail**". Da mesma forma, nos referimos ao artigo sobre instalação como "o guia de instalação" e ao arquivo README como **cf/README**.

Utilizando o pré-processador m4

Primeiramente descrevemos alguns recursos **m4**, mostramos como gerar um arquivo de configuração por meio de um arquivo mestre **m4** e, finalmente, descrevemos algumas importantes macros **m4** pré-empacotadas que acompanham a distribuição de **sendmail**.

m4 destinava-se originalmente como um front end para linguagens de programação que deixava o usuário gravar programas mais legíveis (ou quem sabe mais criptográficos). **m4** é suficientemente poderoso para ser útil em diversas situações de transformação de entradas e ele funciona perfeitamente para arquivos de configuração do **sendmail**.

As macros **m4** assumem a seguinte forma:

```
name(arg1, arg2, ..., argn)
```

Não deve haver nenhum espaço entre o nome e o parêntese de abertura. Aspas simples esquerdas e direitas designam strings como argumentos. As convenções para uso de aspas de **m4** são diferentes das de outras linguagens que talvez você já tenha utilizado, visto que as aspas esquerda e direita são caracteres diferentes.[24] Aspas aninhadas, também. Com as ferramentas de geração dos compiladores atuais, alguém pode ficar imaginando como **m4** sobreviveu com uma sintaxe tão rígida e exótica.

m4 tem algumas macros predefinidas, e os usuários também podem definir suas próprias macros. A Tabela 18.6 lista as macros predefinidas mais comuns utilizadas na configuração do **sendmail**.

Tabela 18.6 Macros m4 comumente utilizadas com o sendmail

Macro	Função
`define`	Define uma macro chamada *arg1* com o valor *arg2*
`undefine`	Descarta uma definição de macro anterior chamada *arg1*
`include`	Inclui (interpola) o arquivo chamado *arg1*
`dnl`	Descarta caracteres até e inclui a próxima nova linha
`divert`	Gerencia fluxos de saída

23. A linguagem de configuração de **sendmail** é "completa em termos da máquina de Turing", o que significa que ela pode ser utilizada para escrever qualquer programa de computação possível. Os leitores que se depararam com o arquivo de configuração bruto entenderão quão assustador é esse conceito...

24. Você pode modificar os caracteres de citação com a macro `changequote`, mas agir desta forma romperia totalmente o emprego do **m4** no **sendmail** porque as várias macros fazem suposições sobre os caracteres de citação.

Alguns sites adicionam uma macro `dnl` no final de cada linha para manter mais organizado o arquivo **.cf** convertido; sem `dnl`, **m4** adiciona linhas em branco extras ao arquivo de configuração. Essas linhas em branco não afetam o comportamento de **sendmail**, mas tornam o arquivo de configuração mais difícil de ser lido. Omitimos `dnl` de nossos exemplos. Outros sites utilizam `dnl` no início de suas linhas destinadas a comentários.

m4 realmente não honra os comentários em arquivos. Um comentário como:

```
# And then define the …
```

não faria o que você esperaria, pois `define` é uma palavra-chave **m4** e seria expandida. Em vez dela, utilize a palavra-chave `dnl` de m4 (que significa "delete to newline" — "delete até a nova linha"). Por exemplo,

```
dnl # And then define the …
```

funcionaria. Você teria de colocar um espaço após `dnl` ou então um sinal de pontuação para ele ser reconhecido como um comando **m4**.

As peças da configuração de sendmail

A distribuição **sendmail** inclui um subdiretório **cf** contendo todas as peças necessárias para a configuração de **m4**: um arquivo README e vários subdiretórios, listados na Tabela 18.7 a seguir.

Tabela 18.7 Subdiretórios de configuração

Diretório	Conteúdo
cf	Modelos de arquivos **.mc** (master configuration)
domain	Modelos de arquivos **m4** para vários domínios na Berkeley
feature	Fragmentos que implementam diversos recursos
hack	Recursos especiais de valor ou implementação dúbia
m4	O arquivo de configuração básico e outros arquivos importantes
ostype	Localizações e peculiaridades de arquivos dependentes do sistema operacional
mailer	Arquivos **m4** que descrevem mailers comuns (agentes de entrega)
sh	Scripts de shell utilizados por **m4**

O diretório **cf/cf** contém exemplos de arquivos **.mc**. De fato, ele contém tantos exemplos que os seus se perderiam no meio desse entulho. Recomendamos que você mantenha seus arquivos **.mc** separados do diretório **cf** distribuído. Crie um diretório nomeado para o seu site (**cf/***nome-do-site*) ou mude o diretório **cf** para **cf.examples** e crie um diretório **cf**. Se você fizer isso, copie o **Makefile** e o script **Build** para o seu novo diretório para que as instruções no arquivo **README** ainda funcionem. É melhor também copiar todos os arquivos de configuração **.mc** para um local centralizado em vez de deixá-los dentro da distribuição **sendmail**. O script **Build** usa nomes de caminhos relativos que terão de ser modificados caso você tente criar um arquivo **.cf** por meio de um arquivo **.mc** e não esteja na hierarquia de distribuição.

Criando um arquivo de configuração com base em um modelo de arquivo .mc

Antes de mergulharmos nas páginas e páginas de detalhes sobre as várias macros de configuração, recursos e opções, colocaremos o carro antes dos bois e criaremos uma configuração "sem ornamentações" para ilustrar o processo. Nosso exemplo é para um nó tipo folha, foo.com; o arquivo de configuração mestre é chamado **foo.mc**.

Colocaremos **foo.mc** em nosso novo e brilhante diretório **cf**. O arquivo de configuração convertido (pelo **m4**) será **foo.cf** no mesmo diretório e, em última instância, o instalamos como **sendmail.cf** em **/etc** ou em **/etc/mail**. **/etc/mail** é o padrão **sendmail** para a localização do arquivo de configuração, mas muitas distribuições utilizam **/etc**.

Algumas frases padrão devem ser colocadas em cada arquivo **.mc** novo:

```
divert(-1)
#### basic .mc file for foo.com
divert(0)
VERSIONID(`$Id$')
```

Se quiser adicionar comentários ao início do seu arquivo, utilize uma instrução `divert` na primeira linha para fazer com que o **m4** descarte o material desnecessário e reduza a necessidade de `dnl` em comentários no estilo #. Os comentários vêm após, seguido de uma outra instrução `divert`.

Uma linha `VERSIONID` (aqui, no formato RCS) completa as frases padrão. Ela é descrita detalhadamente na próxima seção.

Em muitos casos, especificar um `OSTYPE` (consulte a página 395) para acionar caminhos ou parâmetro dependentes de sistema operacional e também um conjunto de `MAILER`s (consulte a página 397) completará a configuração:

```
OSTYPE(`linux')
define(`confCOPY_ERRORS_TO', `postmaster')
MAILER(`local')
MAILER(`smtp')
```

Nesse caso, também configuramos uma opção (`confCOPY_ERRORS_TO`) que envia uma cópia de cabeçalhos de qualquer e-mail ricocheteado para o postmaster local. Essa notificação permite que o postmaster intervenha quando o problema se encontrar no site local.

Para criar o arquivo real de configuração, basta executar o comando **Build** que você copiou para o novo diretório **cf**:

```
# ./Build foo.cf
```

ou

```
# make foo.cf
```

Por fim, instale **foo.cf** no local correto — em geral **/etc/mail/sendmail.cf**, mas tanto o Red Hat como o SUSE o ocultam em **/etc/sendmail.cf**.

Um site maior pode criar um arquivo **m4** separado para armazenar padrões válidos para todo o site no diretório **cf/domain**; hosts individuais então incluem o conteúdo deste arquivo. Nem todo host precisa de um arquivo de configuração separado, mas cada grupo de hosts similares (mesma arquitetura e mesmo papel: servidor, cliente etc.) provavelmente precisará de uma configuração própria.

Alterando a configuração de sendmail

Muitas vezes você achará que a sua configuração de **sendmail** está quase perfeita, mas você gostaria apenas de experimentar um novo recurso, adicionar uma nova regra de spam ou fazer uma alteração simples. Para tanto:

- Edite o arquivo **.mc** e introduza as alterações desejadas
- Recrie o arquivo de configuração com o script **Build** no diretório de configuração
- Instale o arquivo **cf** resultante como **sendmail.cf** no diretório correto
- Envie um sinal HUP para **sendmail** para fazer com que ele releia seu arquivo de configuração[25]

Mesmo com as facilidades do novo sistema de configuração do **sendmail**, você ainda tem de tomar uma série de decisões referentes ao seu site. À medida que for lendo sobre os recursos descritos a seguir, imagine como eles se encaixariam na organização de seu site. Um site pequeno provavelmente terá apenas um nó centralizador e nós de extremidade, portanto, só precisará de duas versões do arquivo de configuração. Um site maior poderia precisar de centralizadores separados para e-mails de entrada e de saída e, talvez, de um servidor POP/IMAP separado.

Seja lá qual for a complexidade de seu site e seja lá qual for o seu aspecto (exposto, por trás de um firewall ou em uma rede virtual privada, por exemplo), é provável que o diretório **cf** contenha alguns trechos de configuração já prontos e adequados apenas aguardando que sejam personalizados e colocados em funcionamento.

18.8 ELEMENTOS BÁSICOS DA CONFIGURAÇÃO DE SENDMAIL

Os comandos de configuração para **sendmail** distinguem o emprego de letras maiúsculas/minúsculas. Por convenção, os nomes de macros predefinidas são todos em letras maiúsculas (por exemplo, `OSTYPE`), comandos **m4** são em letras minúsculas (por exemplo, `define`) e nomes de variáveis configuráveis começam com um `conf` em letras minúsculas e terminam com um nome de variável com todas as letras maiúsculas (por exemplo, `confCOPY_ERRORS_TO`). Em geral, macros se referem a um arquivo **m4** chamado **../*NomeDaMacro/arg1*.m4**. Por exemplo, a macro `OSTYPE(`linux')` faz com que **../ostype/linux.m4** seja incluído.

Nesta seção veremos os comandos de configuração básicos e deixaremos os recursos mais sofisticados para mais tarde.

---

25. Utilize o comando **kill** para fazer isso. O arquivo **sendmail.pid** facilita o processo de localização do ID de processo do **sendmail**; infelizmente, sua localização não é consistente entre as distribuições (tente **/var/run/sendmail.pid**). Consulte a página 391 para obter um exemplo de como utilizá-lo.

A macro VERSIONID

Você deve manter seus arquivos de configuração com CVS ou RCS, não somente para poder retornar para uma configuração anterior caso necessário, como, também, para poder identificar as versões dos arquivos **m4** que formarão o arquivo de configuração. Utilize a macro VERSIONID para embutir automaticamente informações relativas à versão. A sintaxe é:

```
VERSIONID(`$Id$')
```

As informações de versão em si serão preenchidas pelo RCS à medida que você registra o arquivo. Elas aparecerão no arquivo **sendmail.cf** final na forma de um comentário. Essas informações também podem ser úteis se você se esquecer de onde colocou sua distribuição **sendmail**; em geral, a localização dos arquivos é ditada pelo espaço em disco disponível, e não pela lógica de projeto do sistema de arquivos.

A macro OSTYPE

Arquivos no diretório **ostype** são nomeados para o sistema operacional cujos valores padrão eles contêm. Um arquivo OSTYPE empacota uma série de informações específicas do fabricante, como os locais esperados para arquivos relacionadas com e-mail, caminhos para comandos de que **sendmail** precisa, flags para programas de entrega local etc.

Por convenção, as informações específicas a sistema operacional são interpoladas no arquivo de configuração com a macro OSTYPE.[26] Cada arquivo de configuração tem de incluir uma macro OSTYPE perto do início, normalmente após VERSIONID.

Os arquivos OSTYPE fazem seu trabalho basicamente definindo outras variáveis **m4**. Por exemplo,

```
define(`ALIAS_FILE', `/etc/aliases')
```

especifica a localização do arquivo de aliases válidos para todo o sistema. Se quiser, você pode posteriormente fazer com que outros valores para o seu sistema operacional sobrescrevam valores padrão no arquivo **.mc**, mas não altere o arquivo OSTYPE da distribuição original a menos que ele realmente esteja errado, caso em que você também deve abrir um relatório de bug e enviá-lo para sendmail-bugs@sendmail.org. Alguns sites querem uma localização consistente para o arquivo **aliases** entre plataformas e, portanto, redefinem sua localização em seus arquivos DOMAIN.

O arquivo **README** no diretório **cf** lista todas as variáveis que podem ser definidas em um arquivo OSTYPE. Algumas das variáveis importantes são mostradas na Tabela 18.8, juntamente com várias que você talvez queira configurar para combater spams (mas que, por padrão, não se encontram definidas). Os valores padrão são o que você obtém se o arquivo OSTYPE não especificar mais nada.

Tabela 18.8 Valores padrão de algumas variáveis configuradas nos arquivos OSTYPE

Variável	Valor padrão
ALIAS_FILE	/etc/mail/aliases
HELP_FILE	/etc/mail/helpfile
STATUS_FILE	/etc/mail/statistics
QUEUE_DIR	/var/spool/mqueue
MSP_QUEUE_DIR	/var/spool/clientmqueue
LOCAL_MAILER_PATH	/bin/mail
LOCAL_SHELL_PATH	/bin/sh
LOCAL_MAILER_MAX	*indefinido*
LOCAL_MAILER_MAXMSGS	*indefinido*
SMTP_MAILER_MAX	*indefinido*
SMTP_MAILER_MAXMSGS	*indefinido*

Se você instalar **sendmail** em uma versão de sistema operacional nova ou em uma nova arquitetura, não deixe de criar um arquivo OSTYPE correspondente e dá-lo para sendmail.org para que possa ser incluído na próxima versão. Simplesmente modele o novo arquivo baseado naqueles que já existem e compare-o com a tabela de padrões no arquivo **cf/README**. Se o valor de uma variável no novo sistema for o mesmo que o valor padrão, você não precisará incluir uma entrada para ele (mas não custa nada se proteger no caso do padrão mudar).

26. Então, onde afinal a macro OSTYPE em si é definida? Em um arquivo no diretório **cf/m4** que é magicamente anexado ao seu arquivo de configuração quando você executar o script **Build**.

A Tabela 18.9 ilustra os arquivos OSTYPE para as plataformas de referência adotadas neste livro.

Tabela 18.9 Arquivos OSTYPE para sistemas Linux

Sistema	Arquivo	Diretório	Emprego
Red Hat	linux.m4	/usr/share/sendmail-cf	OSTYPE(`linux')
Fedora	linux.m4	/usr/share/sendmail-cf	OSTYPE(`linux')
SUSE	suse-linux.m4	/usr/share/sendmail	OSTYPE(`suse-linux')
Debian	debian.m4	/usr/share/sendmail/sendmail.cf	OSTYPE(`debian')
Ubuntu	linux.m4	/usr/share/sendmail	OSTYPE(`linux')

O SUSE coloca a distribuição de **sendmail** em **/usr/share/sendmail**. O arquivo OSTYPE **suse-linux.m4** se encontra no diretório **ostype** e não faz parte da distribuição **sendmail** de sendmail.org. Esse arquivo é muito longo (mais de 80 linhas) e contém numerosos recursos (FEATUREs) e outras macros que normalmente são encontradas no arquivo de configuração mestre de um site (o arquivo **.mc**), e não no arquivo OSTYPE . Isso oculta a real configuração do administrador de sistemas — uma bênção conflitante, quem sabe, mas *não* uma prática que nós recomendamos.

O Debian oculta os arquivos de configuração sob **/usr/share/sendmail/sendmail.cf/**. O diretório **sendmail.cf** (escolha confusa de nomes por parte do pessoal do Debian) corresponde ao diretório **cf** na distribuição **sendmail** e contém todas as peças de configuração de que você precisa, inclusive um arquivo OSTYPE específico de Debian, **ostype/debian.m4**. O arquivo OSTYPE tem 50 linhas de comprimento e é formado em sua maior parte por nomes de caminhos e comentários, como deveria. Muitos são idênticos aos padrões atuais de sendmail.org e, portanto, realmente não precisam ser reiniciados de modo explícito. Entretanto, reafirmá-los protege o Debian contra alterações nos padrões que de outra maneira poderiam introduzir inconsistências ou erros. O único pecado do Debian no que diz respeito ao OSTYPE de **sendmail** é a inclusão do arquivo DOMAIN genérico que vem junto com a distribuição **sendmail**. Uma instrução DOMAIN deve aparecer logo no início do arquivo **.mc** real, e não ficar oculta no arquivo OSTYPE.

A macro DOMAIN

A diretiva DOMAIN permite especificar informações gerais no nível do site em um único lugar (**cf/domain/***nomeDeArquivo***.m4**) e, então, faz referência a esse lugar no arquivo de configuração de cada host com:

```
DOMAIN(`nomeDeArquivo')
```

Escolha um nome de arquivo que descreva seu site. Por exemplo, nosso arquivo para o departamento de ciências da computação (*science department*) se chama **cs.m4** e aparece em nossos arquivos **.mc** como:

```
DOMAIN(`cs')
```

Assim como OSTYPE, DOMAIN é realmente apenas uma maneira interessante de fazer uma inclusão. Mas torna mais clara a estrutura do arquivo de configuração e fornece um gancho para futuros ajustes. Ele é útil principalmente quando centralizamos e geramos os arquivos **.cf** de todos os nossos sites por meio de arquivos **.mc** mantidos em uma única localização.

Sites pequenos normalmente não precisam de um arquivo de domínio, mas sites maiores em geral os utilizam para referências a máquinas retransmissoras, opções de privacidade ou de mascaramento válidas para todo o site e referências a tabelas para entregadores locais, domínios virtuais e bancos de dados de spam.

O arquivo DOMAIN genérico incluso com a distribuição mostra os tipos de entradas que normalmente são colocados em arquivos de domínios válidos para todo o site. Seu conteúdo (sem comentários e dnls) é mostrado a seguir:

```
VERSIONID(`$Id: generic.m4,v 8.15 1999/04/04 00:51:09 ca Exp $')
define(`confFORWARD_PATH', `$z/.forward.$w+$h:$z/.forward+$h:$z/.forward.$w:$z/.forward')
define(`confMAX_HEADERS_LENGTH', `32768')
FEATURE(`redirect')
FEATURE(`use_cw_file')
EXPOSED_USER(`root')
```

Os arquivos configuram o caminho para as localizações dos arquivos de encaminhamento dos usuários, limita comprimentos de cabeçalho,[27] inclui o recurso redirect para usuários que deixaram a empresa e ativam o recurso use_cw_file para a manipulação dos nomes de máquinas equivalentes. Se o seu arquivo **.mc** incluir mascaramento, o usuário root não será mascarado. Cada uma dessas construções será descrita de modo mais detalhado posteriormente, ainda neste capítulo.

27. Os hackers usaram cabeçalhos extremamente longos como uma forma de provocar uma recusa de serviços em versões mais antigas de **sendmail**. Essa linha está aí no caso de você ainda usar uma dessas versões vulneráveis (anteriores à versão 8.9.3).

A macro MAILER

Você tem de incluir uma macro MAILER para cada agente de entrega que quiser ativar. Você encontrará uma lista completa dos agentes de entrega suportados no diretório **cf/mailers** na distribuição **sendmail**. Atualmente, as opções são local, smtp, fax, usenet, procmail, qpage, cyrus, pop, phquery e uucp. Tipicamente, você precisará pelo menos de:

```
MAILER(`local')
MAILER(`smtp')
```

A primeira linha inclui os agentes de entrega local e prog. A segunda linha inclui smtp, esmtp, dsmtp, smtpB e relay. Foi acrescentado suporte a endereços de e-mail *usuário+detalhes@site.domínio* ao agente de entrega local a partir da versão 8.7.[28] *usuário* define a caixa postal para a qual as mensagens devem ser entregues e *detalhes* fornece um parâmetro extra que um programa de e-mail local como **procmail** poderá utilizar para ordenar e-mails que chegam.

Se planeja ajustar qualquer macro relacionada a agentes de entrega (como USENET_MAILER_ARGS ou FAX_MAILER_PATH), certifique-se de que as linhas que configuram esses parâmetros *precedam* a linha que invoca o agente de entrega em si; caso contrário, os valores antigos serão utilizados. Por essa razão, as declarações MAILER normalmente vêm mais para o final do arquivo de configuração.

O agente de entrega pop faz a interface para o programa **spop** que faz parte do pacote manipulador de e-mails **mh** e implementa o Post Office Protocol definido na RFC1460. Ele é utilizado por PCs e Macs que precisam acessar e-mails em um host UNIX. O agente de entrega cyrus é para uso com o servidor IMAP da CMU e vem em duas variantes: cyrus para entregar e-mails para caixas postais de usuários e cyrusbb para entregar e-mails para um sistema eletrônico de quadros de mensagens centralizado. O agente de entrega cyrus também compreende a sintaxe usuário+detalhes; sua especificação MAILER tem que vir depois daquela do agente de entrega local.

HylaFAX está disponível em www.hylafax.org.

O agente de entrega fax integra o pacote HylaFAX de Sam Leffler no sistema de e-mails. O SUSE o inclui como /usr/bin/faxmail; o Red Hat, o Fedora, o Debian e o Ubuntu não incluem HylaFAX por padrão. Enviar e-mails para *usuário@destino*.fax envia o corpo da mensagem como um documento de fax. *destino* é, tipicamente, um número de telefone. Para possibilitar que nomes simbólicos (em vez de apenas números de telefones) sejam utilizados como destinos, use um arquivo de banco de dados com chaves.

ghostscript está disponível em www.gnu.org.

Você tem de juntar HylaFAX a **sendmail** instalando um script da distribuição HylaFAX em **/usr/local/bin**. Talvez você também precise alterar o valor da macro FAX_MAILER_PATH. Intervenção humana ainda é necessária para transferir faxes que chegam da área de spool para a caixa postal de um usuário. Você pode converter documentos fax em PostScript (com HylaFAX) e visualizá-los com o pacote GNU **ghostscript**.

O agente de entrega qpage faz a interface para o software QuickPage para entregar e-mails para o seu pager. Consulte o site www.qpage.org para mais informações sobre o QuickPage.

As macros VERSIONID, OSTYPE e MAILER são tudo do que você precisa para construir um arquivo *nomeDoHost*.**mc** básico.

18.9 ELEMENTOS MAIS SOFISTICADOS DE CONFIGURAÇÃO DE SENDMAIL

Nas próximas seções, descrevemos algumas outras macros e alguns dos recursos (FEATUREs) mais comuns utilizados para modificar o comportamento padrão de **sendmail**. Também discutimos algumas questões de política no contexto da configuração de **sendmail**: segurança, privacidade, spam e a técnica de ocultar informações pelo emprego de mascaramento e de domínios virtuais.

A macro FEATURE

Por meio da macro FEATURE podemos ativar várias opções comuns incluindo arquivos **m4** do diretório **feature**. Na discussão a seguir, entremeamos nossa apresentação de FEATUREs e algumas outras macros de **sendmail**, já que ocasionalmente elas se entrelaçam. Quando a configuração **m4** foi adicionada pela primeira vez ao **sendmail**, descrever a macro FEATURE se tornou uma grande seção de nosso capítulo sobre e-mail. Agora, foram adicionados tantos recursos que a macro FEATURE quase precisa de um capítulo próprio.

A sintaxe é:

```
FEATURE(palavraChave, argumento, argumento …)
```

onde *palavraChave* corresponde a um arquivo *palavraChave*.**m4** no diretório **cf/feature**, e os argumentos (*args*) são passados para ele. Consulte o próprio diretório ou o arquivo **cf/README** para uma lista conclusiva dos recursos. Alguns poucos mais comumente utilizados são descritos a seguir.

---

28. A sintaxe usuário+detalhes originou-se na Carnegie Mellon University, onde ele é usado com ferramentas locais para direcionar e ordenar e-mails.

O recurso `use_cw_file`

A classe `w` interna de **sendmail** (daí o nome **cw**) contém os nomes de todos os hosts locais que esse host aceita e entrega e-mails. Uma máquina cliente poderia incluir o seu nome de host, seus apelidos e host local nessa classe. Se o host que está sendo configurado for o seu centralizador de e-mails, então a classe `w` também deve incluir quaisquer hosts locais e domínios virtuais para os quais você aceita e-mails.

O recurso `use_cw_file` define a classe `w` por meio do arquivo **/etc/mail/local-host-names** (que costumava ser chamado de **sendmail.cw**). O nome de arquivo exato é configurável pela opção `confCW_FILE`, discutida posteriormente. Sem esse recurso, **sendmail** entrega e-mails localmente apenas se forem endereçados à máquina na qual **sendmail** está sendo executado. Um servidor para e-mails que chegam deve listar no arquivo **local-host-names** todos os nomes de domínios e máquinas para os quais ele irá manipular e-mails. Se você alterar o arquivo, terá de enviar um sinal HUP a **sendmail** para que suas alterações entrem em vigor, pois **sendmail** lê esse arquivo apenas ao iniciar.

```
FEATURE(`use_cw_file')
```

chama o recurso e usa o arquivo **local-host-names** como fonte de dados; eis um exemplo de arquivo **local-host-names**:

```
# local-host-names - include all aliases for your machine here.
toadranch.com
coyote.toadranch.com
big-tr.com
yoherb.com
herbmorreale.com
appliedtrust.com
applied-trust.com
atrust.com
```

Nesse exemplo, as entradas são domínios virtuais hospedados localmente.

O recurso `redirect`

Quando funcionários deixam uma empresa, normalmente encaminhamos seus e-mails ou deixamos que eles ricochateiem de volta para o remetente com um erro. O recurso `redirect` fornece suporte para uma maneira mais elegante de ricochetear e-mails. Se Joe Smith se formou e passou de oldsite.edu para newsite.com, então ativar o `redirect` com

```
FEATURE(`redirect')
```

e acrescentar a linha

```
smithj: joe@newsite.com.REDIRECT
```

ao arquivo **aliases** faz com que e-mails para smithj sejam retornados para o remetente com uma mensagem de erro que sugere que o remetente tente o endereço joe@newsite.com em seu lugar. A mensagem em si não é enviada automaticamente.

O recurso `always_add_domain`

Esse recurso faz com que **sendmail** adicione o nome de host local a endereços de destino locais que não são totalmente qualificados. Suponha, por exemplo, que lynda@cs.colorado.edu envie uma mensagem para os usuários locais barb e evi. Sem `always_add_domain`, os cabeçalhos de e-mails exibiriam os endereços do remetente e do receptor como nomes de login simples. Com o recurso `always_add_domain` ativado, todos os endereços se tornarão totalmente qualificados antes de a mensagem deixar a máquina de lynda.

Use `always_add_domain` ao compartilhar diretórios de spool entre máquinas que não compartilham um arquivo de alias ou que não têm o mesmo arquivo **passwd** (incidentemente, talvez você não deva fazer tal compartilhamento). Os e-mails para um alias ou usuário que não é conhecido em lugar nenhum seriam totalmente qualificados na máquina da qual eles se originam e, portanto, poderiam ser respondidos.

Um outro ponto a favor para esse recurso é que nomes não qualificados normalmente são rejeitados como spam. Recomendamos que você sempre o utilize. (A menos que esteja enviando spam!)

Se estiver utilizando `MASQUERADE_AS` (consulte a página 402), `always_add_domain` adiciona o nome do host que você está se fazendo passar, não o nome de host local. Essa convenção pode provocar problemas se o arquivo **aliases** ou o arquivo **passwd** no host local não for um subconjunto do arquivo equivalente no host `MASQUERADE_AS`.

O recurso `nocanonify`

sendmail tipicamente verifica se a parte do nome de domínio de um endereço é totalmente qualificada e não um CNAME no DNS. Se isso não for verdade, **sendmail** reescreve o endereço. Esse processo se chama canonificação e, normalmente, é feito executando-se uma pesquisa DNS no nome de host. O recurso `nocanonify` diz não a essa reescrita, e a pes-

quisa DNS que é necessária para entregar uma mensagem é postergada. Por exemplo, em um site com um centralizador de e-mail mestre e máquinas clientes que encaminham todos os seus e-mails por meio do mestre, os clientes poderiam utilizar

```
FEATURE(`nocanonify')
```

para impedir a realização de pesquisas DNS localmente. **sendmail** não controla que pesquisas DNS foram feitas à medida que uma mensagem se desloca de máquina em máquina dentro de um site local — ele não é capaz de fazê-lo. O recurso `nocanonify` permite que você controle o período dessas pesquisas. Consulte o estudo de caso de configuração (página 414) para um exemplo.

`nocanonify` também pode ser utilizado em um esquema MSA/MTA como o que poderia ser utilizado em um grande site de e-mails. Nessa situação, o MSA realiza todas as pesquisas DNS, e a máquina mestra que executa o MTA especifica `nocanonify`.

Algumas vezes queremos impedir pesquisas DNS que são potencialmente caras, mas queremos realizar as pesquisas para o domínio local. Você pode dispensar domínios específicos da especificação `nocanonify` incluindo as macros `CANONIFY_DOMAIN` ou `CANONIFY_DOMAIN_FILE`, que usam, respectivamente, uma lista de domínios ou um nome de arquivo como argumento. Por exemplo, as linhas a seguir

```
FEATURE(`nocanonify')
CANONIFY_DOMAIN(`cs.colorado.edu cs')
```

postergariam pesquisas DNS, exceto para endereços no formato *usuário*@cs.colorado.edu ou *usuário*@cs. Essas macros de exceção foram introduzidas pela primeira vez na versão 8.12.

Tabelas e bancos de dados

sendmail possui vários recursos (`FEATURE`s) que usam uma construção chamada "tabela" para descobrir para onde o e-mail deve ser roteado. Uma tabela normalmente é um arquivo de texto de roteamento, aliases, política ou outras informações, convertidas externamente para um formato de banco de dados com o comando **makemap** e depois utilizadas como banco de dados interno para diversas pesquisas de **sendmail**. Embora os dados normalmente comecem como um arquivo de texto, isso não é obrigatório; os dados para as tabelas **sendmail** podem vir do DNS, NIS, LDAP ou de outras fontes. O emprego de um servidor IMAP ou POP centralizado libera **sendmail** da tarefa de caçar usuários e torna obsoletas algumas das tabelas discutidas a seguir. A Tabela 18.10 na página 404 inclui um resumo das tabelas disponíveis.

Duas bibliotecas de banco de dados são suportadas: a biblioteca **dbm/ndbm** que é o padrão na maioria das versões do Linux; e Berkeley DB, uma biblioteca mais extensível que suporta vários esquemas de armazenamento. Sua escolha em termos de bibliotecas de bancos de dados tem de ser especificada em tempo de compilação. Recomendo DB caso você possa instalá-la; ela é mais rápida do que **dbm** e cria arquivos menores. DB está disponível em sleepycat.com.

Há três tipos de mapas de bancos de dados disponíveis:

- `dbm` – utiliza um algoritmo de hash extensível (**dbm/ndbm**)
- `hash` – utiliza um esquema de hash padrão (DB)
- `btree` – utiliza uma estrutura de dados árvore-B (DB)

Para a maioria das aplicações de tabelas no **sendmail**, o tipo de banco de dados `hash` — o padrão — é o melhor. Utilize o comando **makemap** para construir o arquivo de banco de dados por meio de um arquivo de texto; você especifica o tipo de banco de dados e o nome base do arquivo de saída. A versão em forma de texto do banco de dados deve aparecer na entrada padrão de **makemap**, por exemplo:

```
# makemap hash /etc/mail/access < /etc/mail/access
```

À primeira vista esse comando se parece com um erro que faria com que o arquivo de entrada fosse sobrescrito por um arquivo de saída vazio. Entretanto, **makemap** agrega um sufixo apropriado, de modo que o arquivo de saída real é **/etc/mail/access.db** e, na realidade, não há nenhum conflito. Cada vez que o arquivo de texto for modificado, o arquivo de banco de dados terá de ser recriado com **makemap** (mas não é necessário enviar o sinal HUP para **sendmail**).

Na maioria das circunstâncias é usada a mais longa concordância possível para chaves de bancos de dados. Como acontece com qualquer estrutura de dados hashed, a ordem das entradas no arquivo de texto de entrada não é significativa. `FEATURE`s que esperam um arquivo de banco de dados como parâmetro usam `hash` como padrão para o tipo de banco de dados e **/etc/mail/***nomeDaTabela***.db** como o nome de arquivo para o banco de dados. Para sobrescrever esse comportamento, especifique o tipo de banco de dados desejado tanto para o comando **makemap** quanto para o `FEATURE` ou reinicialize o padrão definindo um valor diferente para a variável `DATABASE_MAP_TYPE`. Por exemplo:

```
define(`DATABASE_MAP_TYPE', `dbm')
```

Para utilizar o seu novo banco de dados **access.db**, você adicionará a seguinte linha ao seu arquivo **.mc**:

```
FEATURE(`access_db', `hash /etc/mail/access')
```

Como essa linha usa o tipo e o esquema de atribuição de nomes padrão, você poderia simplesmente escrever

```
FEATURE(`access_db')
```

Você pode especificar o nome de arquivo de banco de dados seja com ou sem o sufixo (.db); sem é preferível.

Não se esqueça de recriar o arquivo de banco de dados com **makemap** toda vez que alterar o arquivo de texto, ou suas alterações não terão efeito.

Abordamos os FEATUREs mailertable, genericstable e virtusertable em algumas seções a seguir. access_db é abrangido mais adiante na seção sobre spam. user_db não é discutido porque tornou-se obsoleto e foi removido.

A partir da versão 8.12, todos os mapas e classes podem especificar LDAP como fonte de seus dados, de modo que você pode solicitar que **sendmail** contate o servidor LDAP para determinar o direcionamento do e-mail e o novo cabeçalho. Apenas especifique LDAP como segundo parâmetro:

```
FEATURE(`access_db', `LDAP')
```

Essa linha fará com que access_db utilize o esquema LDAP padrão que é definido no arquivo **cf/sendmail.schema** na distribuição **sendmail**. Você também pode definir o seu próprio esquema de banco de dados com argumentos adicionais à diretiva FEATURE; veja o arquivo **cf/README** para mais detalhes.

O recurso mailertable

mailertable redireciona e-mails endereçados a um determinado host ou domínio para um destino alternativo por meio de um agente de entrega em particular. Ele é aplicado quando o e-mail sai de um site. O recurso mailertable pesquisa apenas a porção host do endereço, e não a porção usuário. O endereço de cabeçalho não é regravado, portanto o correio continua a ser endereçado ao mesmo usuário, mas é enviado a um host diferente por meio de um mailer diferente. mailertable foi originalmente projetado para lidar com outros sistemas de correio como o UUCP, DECnet e BITNET, mas hoje em dia ele costuma ser utilizado para redirecionar correio de uma máquina de gateway para um servidor interno ou para um servidor em um site remoto que não tem acesso direto à Internet.

Para utilizar um mailertable, inclua a seguinte linha em seu arquivo .mc:

```
FEATURE(`mailertable')
```

Uma entrada em mailertable tem o seguinte formato:

```
domínio_antigo    agente de entrega:destino
```

Um ponto em frente da chave no lado esquerdo é um caractere curinga que significa qualquer host neste domínio. Somente nomes de domínio e host são permitidos como chaves mailertable; nomes de usuários não são permitidos. O valor *destino* do lado direito pode ser um domínio, uma estrutura usuário@domínio ou, até mesmo, nulo, caso em que o envelope não é alterado. O valor *agenteDeEntrega* tem de ser o nome de um agente de entrega definido em uma estrutura MAILER; consulte a página 397.

Suponha, por exemplo, que você usasse MS Exchange como o seu servidor de e-mails interno principal, mas fosse relutante em permitir que ele mantivesse contato com a Internet. Você colocaria um computador Linux na Internet como o seu gateway de e-mails e em seguida encaminharia todos os e-mails para o servidor Exchange depois de ter sido feita uma varredura de vírus ou qualquer outro pré-processamento desejado. Eis a entrada mailertable que faria isso, supondo-se que o servidor Exchange tivesse o endereço IP interno exibido:

```
my-domain  esmtp:[192.168.1.245]
```

Entretanto, essa é uma forma de retransmissão que, como será visto na página 407, precisa ser controlada. Para completar esse exemplo, você precisaria colocar a linha

```
To: my-domain      RELAY
```

em seu banco de dados de acesso para permitir retransmissão para todos os e-mails direcionados a qualquer usuário em my-domain.

O recurso genericstable

O recurso genericstable ("generics table" e não "generic stable") é como alias para os e-mails que chegam. Ele pode, por exemplo, mapear trent@xor.com a trent.hein@xor.com nos e-mails que saem. São os cabeçalhos que são escritos novamente, não o envelope. A entrega de e-mails não é afetada, apenas as respostas.

Vários mecanismos podem mapear nomes de host, mas genericstable é o único que inclui o nome de usuário e o nome de host como parte da chave de mapeamento. Os recursos masquerade_envelope e allmasquerade discutidos posteriormente nesta seção também podem ser aplicados a endereços no recurso genericstable.

Para utilizar `genericstable`, certifique-se de que o seu domínio se encontra na classe `generics`. Para colocar um domínio na classe `generics`, você pode listá-lo na macro `GENERICS_DOMAIN` ou colocá-lo no arquivo especificado pela macro `GENERICS_DOMAIN_FILE`.

Por exemplo, para utilizar `genericstable` com os padrões para o banco de dados, acrescente

```
GENERICS_DOMAIN_FILE(`/etc/mail/local-host-names')
FEATURE(`genericstable')
```

ao seu arquivo de configuração **.mc**. Nesse exemplo, qualquer host no qual você aceita e-mails será incluído. Ativar o recurso `genericstable` diminui um pouco o desempenho de **sendmail**, pois todos os endereços de remetente têm de ser pesquisados.

O recurso `virtusertable`

A tabela de usuários virtual suporta aliases de domínios para e-mails que chegam. Esse recurso permite que vários domínios virtuais sejam hospedados em uma máquina e é comum em sites de hospedagem Web.

O campo key da tabela contém um endereço de e-mail (*usuário@host.domínio*) ou uma especificação de domínio (*@domínio*). O campo value é um endereço de e-mail local ou um endereço de e-mail externo. Se key for um domínio, o valor pode passar o campo *usuário* juntamente com a variável `%1` ou rotear o e-mail para um usuário diferente. Se a especificação de usuário tiver a forma *usuário+detalhes*, então a variável `%2` conterá *detalhes* e a variável `%3` conterá *+detalhes*; use a forma que você quiser.

Vejamos alguns exemplos (adicionamos os comentários):

```
info@foo.com      foo-info              # rota para um usuário local
info@bar.com      bar-info              # outro usuário local
joe@bar.com       error:No such user    # para retornar um erro
@baz.org          jane@elsewhere.com    # todos os e-mails para jane
@zokni.org        %1@elsewhere.com      # para o mesmo usuário, domínio diferente
```

Todas as chaves de host do lado esquerdo dos mapeamentos de dados têm de ser listadas no arquivo **cw**, **/etc/mail/local-host-names** (ou a classe `VirtHost`); caso contrário, **sendmail** tenta localizar o host na Internet e entregar o e-mail lá. Se DNS apontar **sendmail** de volta para esse servidor, você obterá uma mensagem "local configuration error" em ricochetes. Infelizmente, **sendmail** não é capaz de dizer que a mensagem de erro para essa instância deverá realmente ser "virtusertable key not in cw file".

Várias peças estão realmente envolvidas aqui:

- Registros MX do DNS devem existir de modo que, antes de tudo, o e-mail seja roteado para o host correto; então
- entradas **cw** devem estar presentes ou `VIRTUSER_DOMAIN` especificada (ou, de modo equivalente, `VIRTUSER_DOMAIN_FILE`) para permitir que a máquina local aceite o e-mails; e, finalmente
- a tabela de usuário virtual tem de informar a **sendmail** o que fazer com o e-mail.

O recurso é invocado com:

```
FEATURE(`virtusertable')
```

Os exemplos da página 414 utilizam `virtusertable` para implementar hospedagem virtual.

O recurso `ldap_routing`

E como último participante neste antro de aliases, reescrito e falsificado, temos o LDAP (Lightweight Directory Access Protocol). LDAP (ver a página 384 para informações gerais) pode ser um substituto a `virtusertable` no que diz respeito ao roteamento de e-mail e aceitação de correio por domínios virtuais. Ele também pode gerenciar aliases, mapas e classes. E, a partir da versão 8.12, ele pode realizar um trabalho decente com listas de discussão.

Para utilizar LDAP dessa maneira, você tem de incluir várias instruções em seu arquivo de configuração e tem de construir **sendmail** para incluir suporte ao LDAP. Em seu arquivo **.mc** você precisa das linhas

```
define(`confLDAP_DEFAULT_SPEC', `-h servidor -b baseDePesquisa')
FEATURE(`ldap_routing')
LDAPROUTE_DOMAIN(`meu_domínio')
```

para informar **sendmail** de que você quer utilizar um banco de dados LDAP para rotear e-mails que chegam endereçados ao domínio especificado. A opção `LDAP_DEFAULT_SPEC` identifica o servidor LDAP e o nome base de pesquisa de banco de dados.

No exemplo a seguir, a base de pesquisa é `o=sendmail.com, c=US`. Se você estiver utilizando LDAP em uma porta personalizada (não a 389), adicione `-p` *nºDaPortaLDAP* a `LDAP_DEFAULT_SPEC`.

sendmail utiliza os valores de duas tags no banco de dados LDAP:

- `mailLocalAddress` para o destinatário de e-mails que chegam
- `mailRoutingAddress` para o alias ao qual enviar

sendmail também suporta o tag `mailHost`, que, se presente, roteia e-mails para os registros MX para o host especificado, com o recurso `mailRoutingAddress` como receptor.

Por exemplo, a entrada LDAP (para um servidor configurado com um nome root distinto `cn=root, o=sendmail.com, c=US`)

```
dn: uid=eric, o=sendmail.com, c=US
objectClass: inetLocalMailRecipient
mailLocalAddress: eric@sendmail.org
mailRoutingAddress: eric@eng.sendmail.com
```

faria com que e-mails endereçados para eric@sendmail.org (que os registros MX do DNS fizeram com que fossem entregues para sendmail.com) fossem enviados para eric@eng.sendmail.com. Se a entrada também contivesse a linha

```
mailHost: mailserver.sendmail.com
```

então, e-mails para eric@sendmail.org seriam endereçados para eric@eng.sendmail.com e enviados para o host mailserver.sendmail.com após pesquisas MX.

As entradas de banco de dados LDAP suportam uma entrada de caractere curinga, *@domínio*, que roteia novamente os e-mails endereçados a qualquer um no domínio especificado (como foi feito no recurso `virtusertable`).

Nas versões 8.12 e posteriores, acrescentou-se um pouco mais de flexibilidade na forma de uma primitiva de configuração, `LDAPROUTE_EQUIVALENT` (ou `LDAPROUTE_EQUIVALENT_FILE`), que permite definir versões equivalentes do nome de domínio que estamos reroteando com LDAP. Por exemplo, e-mails chegando para usuário@host1.meuDomínio normalmente seriam consultados literalmente no banco de dados LDAP e depois consultados como @host1.meuDomínio. Incluindo-se a linha:

```
LDAPROUTE_EQUIVALENT('host1.mydomain')
```

também seriam tentadas as chaves usuário@meuDomínio e @meuDomínio. Esse recurso permite que um único banco de dados roteie correio em um site complexo.

Argumentos adicionais ao recurso `ldap_routing` agora permitem que especifiquemos mais detalhes sobre o esquema LDAP para utilizar e especificar a manipulação de nomes de usuário que possuem uma parte +detalhe. Na versão 8.13 do sendmail, um novo argumento `ldap_routing` (`sendertoo`) pode rejeitar correio SMTP de um remetente que não existe no LDAP. Como sempre, veja o arquivo **cf/README** para obter informações adicionais.

Mascaramento e a macro `MASQUERADE_AS`

Com a macro `MASQUERADE_AS`, podemos especificar uma única identidade atrás da qual outras máquinas se ocultam. Todos os e-mails parecem emanar da máquina ou domínio designado. O endereço do remetente é reescrito para ser *usuário@nomeDeMascaramento* em vez de *usuário@host-original.domínio*. Obviamente, esses endereços mascarados têm de ser válidos para que as pessoas possam responder aos e-mails.

Essa configuração permite que todos os usuários de um site usem um endereço de e-mail genérico. Por exemplo, se todos os hosts em atrust.com se disfarçarem por trás do domínio atrust.com, então e-mails de *usuário@host*.atrust.com serão estampados como provenientes de *usuário*@atrust.com, sem nenhuma menção do nome de host verdadeiro do qual o usuário enviou o e-mail. A máquina que representa atrust.com tem de saber como entregar os e-mails de todos os usuários, mesmo e-mails para usuários que não tenham um login no servidor de e-mails de entrada. Naturalmente, nomes de login devem ser exclusivos por todo o domínio.

Alguns usuários e endereços (como root, postmaster, hostmaster, trouble, operations, Mailer-Daemon etc.) devem ser excluídos desse comportamento. Eles podem ser excluídos explicitamente com a macro `EXPOSED_USER`.

Por exemplo, a seqüência:

```
MASQUERADE_AS(`atrust.com')
EXPOSED_USER(`root')
EXPOSED_USER(`Mailer-Daemon')
```

estamparia e-mails como provenientes de usuário@atrust.com a menos que tenham sido enviados por root ou pelo sistema de e-mails; nestes casos, os e-mails transportariam o nome do host de origem.

Um recurso introduzido na versão 8.12 permite que você exclua os e-mails para o domínio local (ou e-mails para os hosts específicos listados como exceções) do mascaramento. Por exemplo, esse recurso poderia ser útil para um site que usa um nome de domínio privado não registrado localmente e desejasse mascaramento somente em mensagens ligadas à Internet.

A sintaxe é:

```
FEATURE(`local_no_masquerade')
MASQUERADE_EXCEPTION(`host.domain')
MASQUERADE_EXCEPTION_FILE(`filename')
```

A macro MASQUERADE_AS básica tem várias extensões, tanto por meio de outras macros como por meio de FEATUREs:

- A macro MASQUERADE_DOMAIN
- A macro MASQUERADE_DOMAIN_FILE
- A macro MASQUERADE_EXCEPTION
- A macro MASQUERADE_EXCEPTION_FILE
- O recurso limited_masquerade
- O recurso allmasquerade
- O recurso masquerade_envelope
- O recurso masquerade_entire_domain

Recomendamos utilizar a macro MASQUERADE_AS juntamente com os recursos allmasquerade e masquerade_envelope. O recurso limited_masquerade modifica o comportamento de MASQUERADE_DOMAIN e é útil para ambientes de hospedagem virtual. MASQUERADE_DOMAIN permite que você liste domínios que deseja mascarar; a lista é pré-carregada por meio da classe w que é tipicamente definida com o recurso use_cw_file e lista os hosts em seu domínio. limited_masquerade não pré-inicializa a lista com a classe w. Todos esses domínios são ocultos pelo domínio com o qual você se mascara.

O recurso allmasquerade estende o mascaramento para os receptores da mensagem (ao contrário de apenas o remetente), e o recurso masquerade_envelope o estende ao envelope, bem como os endereços de cabeçalho.[29] Com essas duas extensões, todos os endereços são ocultos de uma forma consistente. O recurso masquerade_entire_domain estende o mascaramento a todos os hosts em uma lista especificada de outros domínios.

Se quiser utilizar outras técnicas de mascaramento, poderá ler sobre seus comportamentos no arquivo **cf/README** ou no livro sobre **sendmail**. Leia com atenção; algumas das primitivas de mascaramento podem ocultar muita coisa.

As macros MAIL_HUB e SMART_HOST

O mascaramento faz com que todos os e-mails pareçam provenientes de um único host ou domínio, reescrevendo os cabeçalhos e, opcionalmente, o envelope. Alguns sites podem querer que todos os e-mails realmente provenham (ou sejam enviados para) de uma única máquina. Você pode obter essa configuração com as macros MAIL_HUB para e-mails que chegam e SMART_HOST para e-mails que saem.

Se você quiser rotear todos os e-mails que chegam a um servidor central para entrega, configure MAIL_HUB para o valor *agenteDeEntrega:host*, onde *agenteDeEntrega* é o agente a ser utilizado para atingir o *host* designado. Se você não especificar um agente de entrega, então relay será utilizado. Por exemplo:

```
define(`MAIL_HUB', `smtp:mailhub.cs.colorado.edu')
```

A designação SMART_HOST faz com que um host entregue e-mail local, mas mande e-mails externos para SMART_HOST. Esse recurso é útil para máquinas protegidas por um firewall e, portanto, não pode utilizar o DNS diretamente. Essa sintaxe se compara àquela de MAIL_HUB; o agente de entrega padrão é novamente relay. Por exemplo:

```
define(`SMART_HOST', `smtp:mailhub.cs.colorado.edu')
```

Nesses exemplos, a mesma máquina atua como servidor tanto para e-mails que chegam como para aqueles que saem. Um site maior poderia dividir isso em máquinas separadas. O SMART_HOST tem de permitir a retransmissão para que as máquinas clientes possam enviar e-mails por meio dele. Entradas mailertable sobrescrevem a designação SMART_HOST.

Mascaramento e roteamento

Com todos esses recursos e macros prontos e esperando para manipular seus endereços de e-mail, imaginamos que seria interessante tentar comparar os diversos mecanismos em relação à sua alteração nos cabeçalhos, no envelope ou na entrega de uma mensagem, ou à sua aplicação em mensagens que chegam e que saem, endereços de remetente ou de receptor etc. Se a página tivesse uma largura dupla ou tripla, talvez pudéssemos realmente ter sido bem-sucedidos em ilustrar as diferenças entre os diversos comandos.

---

29. Os endereços de cabeçalho são os endereços To, From, Cc e Bcc que aparecem no cabeçalho de uma mensagem. Os endereços de envelope são os endereços para o qual o e-mail é efetivamente entregue. Os endereços de envelope são originalmente construídos a partir dos endereços de cabeçalho pelo agente de usuário, mas são processados separadamente pelo **sendmail**. Muitos dos recursos de mascaramento e de redirecionamento de **sendmail** seriam impossíveis de implementar se a distinção entre os endereços de cabeçalho e de envelope não fosse mantida.

Em vez disso, fornecemos uma dica na Tabela 18.10, a seguir; você terá de pesquisar os detalhes na documentação do **sendmail** para obter as nuances das diferentes variações.

As entradas da Tabela 18.10 que aparecem letras maiúsculas são macros **m4**. As entradas com letras minúsculas são os nomes dos recursos que foram chamados com a macro `FEATURE`. Itens recuados dependem dos itens acima; por exemplo, um recurso que modifica o comportamento `MASQUERADE_AS` não faz nada a menos que `MASQUERADE_AS` tenha sido ativado. Na tabela, o recurso está recuado para indicar essa dependência. O mascaramento afeta os endereços de cabeçalho em e-mails que saem e se uma mensagem pode ser respondida; o roteamento afeta a entrega em si dos e-mails.

Tabela 18.10 Comparação entre os recursos de mascaramento e de roteamento

	Construção	Direção	Afeta[a]	Que trecho
Mascaramento	`MASQUERADE_AS`	fora	SH	host.domínio
	`allmasquerade`	fora	RH[b]	host.domínio
	`MASQUERADE_DOMAIN[_FILE]`	fora	SH	host.domínio
	`masquerade_entire_domain`	fora	SH	host.sub.domínio
	`limited_masquerade`	fora	SH	host.domínio
	`masquerade_envelope`	fora	SE[c]	host.domínio
	`genericstable`	fora	SH	user@host.domain
Roteamento	`mailertable`	fora	MAD	host.domínio
	`virtusertable`	dentro	RD	user@host.domain
	`ldap`	dentro	RD	user@host.domain
	`mailhub`	dentro	RD	e-mail local
	`smarthost`	fora	RD	e-mail remoto

a. S = sender (remetente), R = recipient (receptor), D = delivery (entrega), H = header (cabeçalho), E = envelope, M = mailer (agente de entrega), A = address (endereço).

b. Assim que a reescrita de receptor tiver sido ativada com o recurso `allmasquerade`, todas as demais construções de mascaramento reescrevem não somente o remetente, como também o receptor;

c. Assim que a reescrita de envelope tiver sido ativada com o recurso `masquerade_envelope`, todas as demais construções de mascaramento reescrevem não somente o cabeçalho, como também o envelope.

O recurso `nullclient`

`nullclient` é utilizado para um host que nunca deve receber e-mails diretamente e que envia todos os e-mails a um servidor central. O arquivo **.mc** para esse tipo de host tem apenas duas linhas.

```
OSTYPE(`tipoDeSistemaOperacional')
FEATURE(`nullclient', `servidorDeEmails')
```

O recurso `nullclient` sobrescreve vários outros recursos. Todos os e-mails, sem exceções, são entregues a *servidorDeEmails* para processamento.[30] Observe que o servidor precisa permitir que o cliente retransmita por meio deste se os usuários regularmente originam o e-mail no cliente e não usam um servidor separado para e-mails que saem. Versões recentes do **sendmail** têm a retransmissão desativada por padrão. Consulte a seção sobre spam (página 407) para detalhes sobre como controlar a retransmissão. Uma configuração `nullclient` mascara como *servidorDeEmails*, assim você talvez queira incluir uma cláusula `EXPOSED_USER` para root.

O cliente que usa o recurso `nullclient` deve ter um registro MX associado que aponte para o servidor. Ele também deve ser incluído no arquivo **cw** do servidor, que normalmente é **/etc/mail/local-host-names**. Essas configurações permitem que o servidor aceite e-mails para o cliente.

Um host com uma configuração `nullclient` não deve aceitar e-mails que chegam. Se fizer isso, ele simplesmente encaminhará o e-mail para o servidor de qualquer maneira. Iniciar **sendmail** sem o flag **-bd** de modo que ele não escute conexões SMTP na porta 25 é um modo de impedir o recebimento de e-mails. Mas, alguns agentes de usuário (MUAs) tentam a submissão inicial de uma mensagem de e-mail por meio da porta 25 e, portanto, serão frustrados em seu intento caso seu **sendmail** não esteja escutando. Uma maneira melhor para desativar a chegada de e-mails é executar **sendmail** com o flag **-bd**, mas usar

30. Se você configura um cliente dessa maneira e, então, testa a configuração com **sendmail –bt**, o cliente parece entregar o e-mail local localmente. A razão é que uma diretiva `nullclient` é processada posteriormente, na ruleset 5 do arquivo de configuração.

DAEMON_OPTIONS para escutar apenas na interface de loopback. De qualquer maneira, deixe o flag **-q30m** na linha de comando; se o *servidorDeEmails* cair, o cliente poderá colocar em fila o seu e-mail e tentar enviá-lo mais tarde para *servidorDeEmails*.

nullclient é apropriado para nós de extremidade em sites que possuem uma máquina central para e-mails. Em sites maiores, considere a carga de e-mails na máquina centralizadora. Talvez você queira separar os servidores de entrada e de saída ou adotar uma abordagem hierárquica.

O SUSE é distribuído com um arquivo nullclient **mc** de exemplo em **/etc/mail/linux.nullclient.mc**. Simplesmente preencha o nome do seu *servidorDeEmails*, execute **m4** nele para criar o arquivo **sendmail.cf** e isso é tudo.

Os recursos local_lmtp e smrsh

Como padrão, o agente de entrega local utiliza **/bin/mail** como agente de entrega local para usuários e arquivos, e **/bin/sh** como agente de entrega para programas. **sendmail** agora dispõe de alternativas melhores, especialmente para a entrega para programas. Ambas as opções estão disponíveis por meio da macro FEATURE.

Se o recurso local_lmtp for especificado, então seu argumento é um agente de entrega local capaz de falar LMTP (Local Mail Transport Protocol) (consulte a RFC2033). O padrão de entrega para usuários é o programa **mail.local** da distribuição **sendmail**. Da mesma forma, o recurso smrsh especifica o caminho que o programa deve utilizar para a entrega de correio aos programas. Consulte a página 419 para uma discussão mais detalhada de **mail.local** e **smrsh**.

O recurso local_procmail

Você pode usr **procmail** de Stephen van den Berg como seu agente de entrega local habilitando o recurso local_procmail. Ele utiliza até três argumentos: o caminho para o binário **procmail**, o vetor de argumento para chamá-lo e flags para o agente de entrega. Os valores padrão são razoáveis, porém o caminho padrão (**/usr/local/bin/procmail**) conflita com a localização usual (**/usr/bin**) de **procmail** da maioria das distribuições.

procmail pode realizar coisas mais sofisticadas para o usuário do que o **/bin/mail** ou **mail.local** simples pode. Além de entregar correio para as caixas de correio dos usuários, ele pode classificar as mensagens em pastas, salvá-las em arquivos, executar programas e filtrar spam. O emprego do recurso local_procmail anula em grande parte as melhorias de segurança fornecidas por **smrsh**. Entretanto, caso não precise restringir os programas que seus usuários usam (isto é, você confia em *todos* os seus usuários), **procmail** pode ser muito prático. **procmail** não é distribuído com o **sendmail**; obtenha-o em www.procmail.org caso ele não seja instalado pelo seu fornecedor.

Você pode utilizar outros programas para processamento de e-mails em conjunto com esse recurso simplesmente mentindo para **sendmail** e dizendo que você está apenas mostrando a ele a cópia local de **procmail**:

```
FEATURE(`local_procmail', `/usr/local/bin/mymailer')
```

Se utilizar **procmail**, experimente **/usr/bin/mailstat** para algumas estatísticas práticas sobre **procmail** (não se confunda com **/usr/sbin/mailstats**, que apresenta estatísticas de **sendmail**). Ele é instalado em todas as distribuições utilizadas como exemplo neste livro e pode ser utilizado para sintetizar arquivos de log do **procmail**.

As macros LOCAL_*

Se realmente precisar sujar as mãos e escrever algumas novas regras exóticas para lidar com situações locais especiais, você pode utilizar um conjunto de macros com o prefixo LOCAL_. A seção sobre spam, mais adiante neste capítulo, tem alguns exemplos desta construção de baixo nível.

Opções de configuração

As opções de arquivos de configuração e macros (os comandos O e D na linguagem de configuração bruta) podem ser configuradas por meio do comando define de **m4**. Uma lista completa de opções acessíveis como variáveis de **m4** e seus valores padrão são dados no arquivo **cf/README**. Os valores padrão são razoáveis para a maioria dos sites.

Alguns exemplos:

```
define(`confTO_QUEUERETURN', `7d')
define(`confTO_QUEUEWARN', `4h')
define(`confPRIVACY_FLAGS', `noexpn')
```

A opção "queue return" determina por quanto tempo uma mensagem permanecerá na fila de e-mails caso ela não consiga ser entregue. A opção "queue warn" determina por quanto tempo uma mensagem permanecerá antes de o remetente ser notificado de que pode haver problemas com a entrega. As duas primeiras linhas configuram essas opções para, respectivamente, 7 dias e 4 horas.

Consulte a página 419 para informações adicionais sobre as opções de privacidade.

A linha seguinte configura os flags de privacidade para desautorizar o comando SMTP EXPN (expandir endereço). A opção confPRIVACY_FLAGS usa uma lista de valores separados por vírgulas. Algumas versões de **m4** exigem dois conjuntos de aspas para proteger as vírgulas num campo com várias entradas, mas o GNU **m4** que vem com o Linux é mais inteligente e não requer tais aspas extras:

```
define('confPRIVACY_FLAGS', ``noexpn, novrfy'')
```

Os valores padrão para a maioria das opções são bem razoáveis para um site típico que não seja muito paranóico em relação a questões de segurança ou que não seja muito preocupado com o desempenho. Em particular, os padrões o protegem contra spams desativando a retransmissão, exigindo que endereços sejam totalmente qualificados e exigindo que os endereços tenham endereços IP. Se a sua máquina centralizadora de e-mail for muito ocupada e tiver que atender a um grande número de listas de discussão, talvez seja preciso ajustar alguns dos valores relativos ao desempenho.

A Tabela 18.11 lista algumas opções que talvez você precise para ajustar (em torno de 15% de cerca de 175 opções de configuração), juntamente com seus valores padrão. Para poupar espaço, os nomes das opções são mostrados sem seus prefixos `conf`; por exemplo, a opção `FALLBACK_MX` na verdade se chama `confFALLBACK_MX`. Dividimos a tabela em subseções que identificam o tipo de problema que a variável atende: genérico, recursos, desempenho, segurança e combate a spams e diversos. Algumas opções se encaixam claramente em mais de uma categoria, mas as listamos apenas uma vez.

Tabela 18.11 Opções básicas de configuração

	Nome da opção	Descrição e (valor padrão)
Genérico	COPY_ERRORS_TO	Endereços para Cc em mensagens de erro (nenhum)
	DOUBLE_BOUNCE_ADDRESS	Captura uma grande quantidade de spams, alguns sites usam **/dev/null**, mas isso pode ocultar problemas sérios (postmaster)
Recursos	MIN_FREE_BLOCKS	Espaço de sistema de arquivos mínimo para aceitar e-mails (100)
	MAX_MESSAGE_SIZE	Tamanho máximo em bytes de uma única mensagem (infinito)
	TO_ *muitasCoisas*	Tempos-limite para todos os tipos de coisas (vários)
	TO_IDENT	Tempo-limite para consultas ident para verificação da identidade do remetente; se for 0, as verificações de identidade não serão feitas (5s)
	MAX_DAEMON_CHILDREN	Número máximo de processos filho[a] (nenhum limite)
Desempenho	MCI_CACHE_SIZE	Número de conexões TCP para fora abertas e colocadas em cache (2)
	MCI_CACHE_TIMEOUT	Tempo para manter abertas conexões em cache (5m)
	HOST_STATUS_DIRECTORY	Consulte a página 424 para uma descrição (sem valor padrão)
	FALLBACK_MX	Consulte a página 424 para uma descrição (sem valor padrão)
	FAST_SPLIT	Suprime pesquisas MX enquanto os receptores são ordenados e distribuídos entre filas; consulte a página 423 (1 = verdadeiro)
	QUEUE_LA	Carga média na qual os e-mails devem ser colocados em fila em vez de serem enviados imediatamente (8 * Num.CPUs)
	REFUSE_LA	Carga média na qual se deve recusar e-mails (12 * Num.CPUs)
	DELAY_LA	Carga média para reduzir a velocidade de entregas (0 = nenhum limite)
	MIN_QUEUE_AGE	Tempo mínimo que as tarefas têm de permanecer em fila; faz com que uma máquina muito ativa manipule melhor a fila (0)
Segurança/spam	TRUSTED_USERS	Para proprietários de software de listas de discussão; permite forjar a linha From e recriar o banco de dados de aliases (root, daemon, uucp)
	PRIVACY_FLAGS	Limita as informações dadas por SMTP (avisos automáticos)
	INPUT_MAIL_FILTERS	Lista filtros para correio recebido (vazio)
	MAX_MIME_HEADER_LENGTH	Tamanho máximo dos cabeçalhos MIME (nenhum limite)[b]
	CONNECTION_RATE_THROTTLE	Diminui os ataques DoS limitando a velocidade na qual conexões de e-mails são aceitas (nenhum limite)
	MAX_RCPTS_PER_MESSAGE	Diminui a entrega de spams; protela receptores extras e envia uma mensagem de erro temporária (infinito)
	DONT_BLAME_SENDMAIL	Sobrescreve a verificação de arquivos e segurança de **sendmail**; não brinque com isso! (segurança)
	AUTH_MECHANISMS SMTP	Mecanismos de autorização SMTP para o Cyrus SASL[c]
Misc	LDAP_DEFAULT_SPEC	Especificação de mapas para bancos de dados LDAP, inclusive o host e a porta que o servidor está executando (indefinido)

a. Mais especificamente, o número máximo de processos filhos que podem ser executados de uma vez. Quando o limite é atingido, o sendmail rejeita conexões. Essa opção pode evitar (ou criar) ataques de negação de serviço (*denial of service* – DoS).

b. Essa opção pode impedir que o agente de usuário sofra um estouro em seus buffers. "256/128" é um bom valor a utilizar — significa 256 bytes por cabeçalho e 128 bytes por parâmetro para esse cabeçalho.

c. O valor padrão é "EXTERNAL GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5"; não adicione PLAIN LOGIN, a menos que você queira reduzir a segurança.

18.10 RECURSOS RELACIONADOS COM SPAM NO SENDMAIL

Spam é o jargão para correspondência indesejada, também conhecida como e-mail comercial não-solicitado. Ele tornou-se um problema sério, principalmente porque os remetentes em geral não pagam pelos bytes, mas, em vez disso, pagam uma taxa fixa pela conectividade. Ou, se pagam por byte, eles enviam uma única mensagem com milhares de receptores, retransmitindo-a por meio de uma outra máquina. A outra máquina paga o grande custo por byte e o spammer paga somente por uma cópia. Em vários países, usuários finais pagam por bytes recebidos e ficam muito irritados por terem de pagar para receber spam.

Sob o ponto de vista do pessoal de marketing, o spam funciona bem. As taxas de respostas são altas, os custos são baixos e a entrega é instantânea. Uma lista de 30 milhões de endereços de e-mail custa cerca de US$ 40.

Muitos spammers tentam parecer gentis sugerindo que você responda à mensagem deles com uma mensagem dizendo "remove" caso você queira ser removido da lista de e-mail deles. Embora até possam removê-lo da lista, você apenas acabou confirmando que eles estão de posse de um endereço de e-mail atualizado e válido; essas informações podem levá-lo a ser incluído em outras listas. Os spammers também adoram bagunçar os cabeçalhos de seus e-mails em uma tentativa de despistar de quem veio o e-mail e em que máquina ele se originou.

O pessoal que vende endereço de e-mail para spammers começou recentemente a adotar uma forma de ataque-dicionário para descobrir endereços desconhecidos. Começando com uma lista de sobrenomes comuns, o software de varredura adiciona iniciais diferentes na esperança de atingir um endereço de e-mail válido. Para verificar os endereços, o software se conecta a servidores de e-mails em, digamos, 50 grandes ISPs e faz um VRFY ou RCPT em cada um dos zilhões de endereços.

Essa sondagem tem um impacto enorme sobre o servidor de e-mails e sobre sua capacidade de entregar a correspondência legítima para seus clientes. **sendmail** pode lidar com essa situação por meio da opção `PrivacyOption goaway` que é vista a partir da página 419. Mas os programas para spam mais inteligentes são muito robustos; se VRFY for bloqueado, eles tentam EXPN, e, se ambos falharem, tentam RCPT. Eles podem experimentar milhões de endereço dessa forma e nunca enviar uma única mensagem — mas, certamente, mantêm o seu servidor de e-mails bem ocupado.

sendmail tem uma opção, `BAD_RCPT_THROTTLE`, para anular esse comportamento. Se o número de endereços rejeitados no envelope de uma mensagem exceder o valor desta opção, **sendmail** entra em estado de hibernação por um segundo depois de cada comando RCPT rejeitado.

sendmail agregou alguns recursos muito interessantes para ajudar no controle de spam e também para ajudar a combater vírus de computadores ocasionais originados em mensagens de e-mails. Infelizmente, a maioria dos ISPs tem de repassar todas as mensagens; ou seja, tais recursos podem vir a ser muito draconianos para a política de clientes (ou, mais uma vez, talvez não sejam). Entretanto, os recursos podem ser utilizados com grande efeito no site do usuário final.

Há quatro versões de recursos de controle de spam:

- Regras que controlam a retransmissão promíscua ou de terceiros, que é o emprego de seu servidor de e-mails por um usuário fora do site para enviar e-mails para um outro usuário fora do site. Spammers costumam utilizar a retransmissão para mascarar a verdadeira fonte dos seus e-mails e assim evitar a detecção pelos ISPs. Ela também permite que eles usem os *seus* ciclos e economizem os deles próprios. Isto é o principal.
- O banco de dados de acesso, por meio do qual o correio é filtrado por endereço individual, é quase como um firewall para e-mail.
- Listas negras contendo retransmissores abertos e sites conhecidos e amigos dos spams os quais **sendmail** pode consultar.
- Verificação de cabeçalhos e filtragem de e-mail de entrada por meio de uma interface de filtragem de e-mails genérica chamada **libmilter**. Ela permite varredura arbitrária dos cabeçalhos das mensagens *e conteúdo* e permite que você rejeite mensagens que coincidam com um determinado perfil.

Descreveremos esses novos recursos aqui e depois veremos alguns exemplos de spam que recebemos recentemente para ver como poderíamos ter ajustado nosso sistema de e-mails para reconhecê-los e rejeitá-los automaticamente.

Retransmissão

sendmail e outros agentes de transporte de e-mail aceitam e-mails que chegam, examinam os endereços do envelope, decidem para onde as mensagens devem ser encaminhadas e depois as repassam para um destino apropriado. Esse destino pode ser local ou um outro agente de transporte mais distante na cadeia de entrega. Quando uma mensagem que chega não tem nenhum receptor local, diz-se que o agente de transporte que a trata está atuando como retransmissor.

Antes da versão 8.9 do **sendmail**, a retransmissão promíscua (também chamada de retransmissão aberta) era ativa por padrão. **sendmail** aceitaria qualquer mensagem apresentada a ele na porta 25 e faria o máximo para realizar a entrega. Era a política de vizinhança da Internet.

Infelizmente, os spammers começaram a abusar da retransmissão; eles a exploraram para disfarçar suas identidades e, mais importante, utilizar sua largura de banda e ciclos em vez de suas próprias. Atualmente é considerado muito ruim con-

figurar o seu servidor de e-mails como um retransmissor aberto. Contudo, vários servidores ainda são configurados como retransmissões abertas.

Somente hosts marcados com `RELAY` no banco de dados de acesso (consulte a página 409) ou listados em **/etc/mail/relay-domains** têm a permissão para enviar e-mails para retransmissão. Em alguns anos, a proporção de retransmissões abertas deve cair como resultado dessa alteração no comportamento padrão, aumentando a consciência pública e verificando preventivamente com base nas várias listas negras.

Portanto, a retransmissão promíscua é negativa. Ao mesmo tempo, alguns tipos de retransmissão são úteis e legítimos. Como você poderia dizer quais mensagens devem ser retransmitidas e quais devem ser rejeitadas? A retransmissão só é realmente necessária em duas situações:

- *Quando o agente de transporte atua como gateway para hosts que não são atingíveis de outra maneira*; por exemplo, hosts que nem sempre estão ativos (hosts discados, Windows PCs) e hosts virtuais. Nesta situação, todos os receptores para os quais você quer retransmitir estão dentro do mesmo domínio.
- *Quando o agente de transporte é o servidor de e-mails de saída para outros hosts não tão inteligentes.* Neste caso, todos os nomes de hosts dos remetentes ou endereço IP serão locais (ou pelo menos enumeráveis).

Qualquer outra situação que pareça requerer retransmissão provavelmente é apenas uma indicação de projeto ruim (com a possível exceção do suporte para usuários móveis). Você pode prevenir o primeiro emprego de retransmissão (acima) designando um servidor centralizado para receber e-mails (com POP ou IMAP utilizado para acesso a clientes). O segundo caso deve sempre ser permitido, mas apenas para os seus próprios hosts. Você pode verificar endereços IP ou nomes de host; nomes de host são mais fáceis de adulterar, mas o **sendmail** verifica se eles não são forjados.

Embora **sendmail** venha com a opção de retransmissão desativada por padrão, vários recursos foram adicionados para reativá-la, de uma maneira total ou limitada e controlada. Esses recursos são listados a seguir por uma questão de completude, porém, nossa recomendação é que você seja cuidadoso em relação a abrir demais as coisas. A maioria dos sites não precisa de nenhum dos recursos realmente perigosos da segunda lista apresentada a seguir. O recurso `access_db`, visto na próxima seção, é a maneira mais segura de permitir retransmissão limitada.

- `FEATURE('relay_entire_domain')` – permite retransmissão apenas para o seu domínio
- `RELAY_DOMAIN('`*domínio*`, …')` – adiciona mais domínios a serem retransmitidos
- `RELAY_DOMAIN_FILE('`*nomeDeArquivo*`')` – o mesmo, mas seleciona a lista de domínios de um arquivo
- `FEATURE('relay_hosts_only')` – afeta `RELAY_DOMAIN`, `accessdb`

Você precisará abrir uma exceção se você utilizar as designações `SMART_HOST` ou `MAIL_HUB` para rotear e-mail por meio de uma determinada máquina servidora para e-mails. Esse servidor terá de ser configurado para retransmitir e-mails de hosts locais. Configure-o com

```
FEATURE('relay_entire_domain')
```

Sites que realizam hospedagem virtual talvez também precisem de `RELAY_DOMAIN` para permitir a retransmissão para seus nomes virtuais, embora

```
FEATURE('use_cw_file')
```

efetivamente abra retransmissão para esses domínios ou hosts.

As poucas outras possibilidades estão repletas de problemas:

- `FEATURE('promiscuous_relay')` – permite todas as retransmissões; não o utilize
- `FEATURE('relay_based_on_MX')` – retransmissões para qualquer pessoa que lhe envia um MX
- `FEATURE('loose_relay_check')` – permite endereçamento de "porcentagem de hack"
- `FEATURE('relay_local_from')` – baseia retransmissões no endereço From

O recurso `promiscuous_relay` permite retransmitir de qualquer site para qualquer outro site. Utilizá-lo é um bilhete só de ida para as listas negras. *Não* use esse recurso em uma máquina que possa ser atingida via Internet pública.

O recurso `relay_based_on_MX` é ruim, pois você não controla que sites têm permissão para apontar seus registros MX para você. Tipicamente, os únicos hosts que possuem um registro MX apontando para o seu servidor para e-mail são os seus próprios hosts, mas nada impede que outros sites alterem seus registros MX para apontar para você. Spammers normalmente não podem alterar registros MX, mas sites de reputação duvidosa certamente poderiam.

O recurso `loose_relay_check` permite a forma "% hack" (notação com o símbolo de porcentagem) de endereçamento (consulte a página 535) que os spammers adoram usar.

O recurso relay_local_from confia no endereço de remetente contido no envelope da mensagem e retransmite mensagens que pareçam ser provenientes de um endereço local. Obviamente, tanto o envelope como os cabeçalhos das mensagens de e-mail são fáceis de serem forjados, e os spammers são especializados em falsificações.

Caso considere ativar a retransmissão de alguma forma, consulte a documentação de **sendmail** em **cf/README** para garantir que você não se torne inadvertidamente um amigo dos spammers. Quando tiver terminado, faça com que um dos sites de verificação de retransmissão verifique se você não criou inadvertidamente um retransmissor aberto — experimente spam.abuse.net.

Há configurações incorretas na qual seu host poderia ser convencido a retransmitir endereços estranhos que utilizam a sintaxe de endereçamento UUCP de maneira abusiva. Apenas para se certificar, caso você não tenha conectividade UUCP, poderá usar

```
FEATURE(`nouucp', `reject')
```

para impedir tal possibilidade. O **sendmail** atual não tem como padrão suporte a qualquer uma das tecnologias de rede antigas como UUCP, BITNET ou DECnet.

Uma outra retransmissão comum é LUSER_RELAY para usuários locais que não existem. Ela é definida por padrão como

```
define(`LUSER_RELAY', `error:No such user')
```

Um site com **sendmail** configurado incorretamente algumas vezes deixa vazar nomes de usuários locais não qualificados para a Internet (em geral na linha Cc). Alguém que responda ao e-mail responde a um usuário aparentemente local que não existe. Essa retransmissão normalmente é chamada de "retransmissão de perdedor" e é direcionada para o agente de entrega chamado error.

Você não precisa alterar essa configuração, a menos que queira retornar uma mensagem diferente ou implementar algum tipo de tratamento especial. Alguns sites redirecionam e-mails de "perdedor" para uma pessoa ou programa que realize concordância flexível, no caso de o remetente cometer um erro de digitação ou apenas ter o nome de login ligeiramente errado.

O banco de dados de acesso

sendmail inclui suporte a um banco de dados de acesso que você poderá utilizar para construir um firewall específico para e-mails para o seu site. O banco de dados de acesso verifica e-mails provenientes do mundo externo e os rejeita se vierem de usuários ou domínios específicos. Você também pode utilizar o banco de dados de acesso para especificar quais domínios uma máquina estará disposta a retransmitir.

O banco de dados de acesso é ativado com a seguinte linha

```
FEATURE(`access_db', `tipo nomeDeArquivo')
```

Se *tipo* e *nomeDeArquivo* não forem especificados, o banco de dados usa como padrão o tipo hash (se forem utilizados bancos de dados DB — depende da configuração de DATABASE_MAP_TYPE) criado a partir do arquivo **/etc/mail/access**. Os bancos de dados DBM não usam o campo *tipo*. Como sempre, crie o banco de dados utilizando **makemap**:

```
# makemap hash /etc/mail/access < /etc/mail/access
```

O campo key do arquivo de acesso pode conter endereços de e-mail, nomes de usuário, nomes de domínios ou números de rede.

Por exemplo:

```
cyberspammer.com  550 Spam not accepted
okguy@cyberspammer.com    OK
badguy@aol.com    REJECT
sendmail.org      RELAY
128.32     RELAY
170.201.180.16    REJECT
hotlivesex@550 Spam not accepted
friend@    550 You are not my friend!
```

A parte de valor deve conter um dos itens mostrados na Tabela 18.12. O valor RELAY é o mais permissivo; ele simplesmente aceita a mensagem e a encaminha para o seu destino final. OK aceita a mensagem, mas não permitirá retransmissão. REJECT não aceitará a mensagem de forma alguma. SKIP permite que você abra exceções. Se você quiser, por exemplo, retransmitir e-mails para todos os hosts exceto dois em um determinado domínio, poderá listar os dois hosts com a ação SKIP e, a seguir, listar o domínio com a ação RELAY. A ordem não importa.

Tabela 18.12 Coisas que podem aparecer no campo de valor do banco de dados de acesso

Valor	O que ele faz
OK	Aceita e-mail e o entrega normalmente
RELAY	Aceita o e-mail como endereçado e o retransmite ao seu destino; ativa a retransmissão por host
SKIP	Permite exceções a regras mais genéricas
REJECT	Rejeita o e-mail com uma mensagem de erro genérica
DISCARD	Descarta a mensagem silenciosamente
FRIEND	Para spams, utilizado pelo recurso `delay-checks`; se for coincidente, pula outras verificações de cabeçalho
HATER	Para spams, utilizado pelo recurso `delay-checks`; se for coincidente, aplica outras verificações de cabeçalho
`xxx message`	Retorna um erro; *xxx* tem de ser um código numérico da RFC821[a]
`ERROR:xxx message`	Mesmo que o anterior, mas claramente marcada com uma mensagem de erro
`ERROR:x.x.x message`	*x.x.x* tem de ser uma notificação de estado de entrega concordante com a RFC1893 (uma generalização do código de erro 550)

a. Por exemplo, 550 é um código de erro único.

O arquivo de banco de dados anterior permitiria mensagens de okguy@cyberspammer.com, mas rejeitaria todos os outros e-mails de cyberspammer.com com a mensagem de erro indicada. E-mails de sendmail.org ou então de 128.32.0.0/16 (rede da UC Berkeley) seriam retransmitidos. O correio de badguy@aol.com e de hotlivesex, ou um do gênero, em qualquer domínio também seria rejeitado.

Endereços IPv6 separados por dois-pontos podem ser utilizados também no lado esquerdo, porém teriam antes o prefixo "IPv6:". O símbolo @ depois dos nomes de usuários hotlivesex e friend é necessário; ele diferencia nomes de usuário de nomes de domínios.

550 é um código de erro RFC821. Os códigos de erro RFC1893 (ou "mensagens de notificação de estado de entrega", como são chamados) são mais abrangentes. O primeiro dígito 4 indica um erro temporário; 5 significa um erro permanente. Listamos alguns na Tabela 18.13.

Tabela 18.13 Códigos de estado de entrega da RFC1893

Temporário	Permanente	Significado
4.2.1	5.2.1	A caixa postal está desativada
4.2.2	5.2.2	A caixa postal está cheia
4.2.3	5.2.3	A mensagem é muito longa
4.2.4	5.2.4	Problema de expansão de lista
4.3.1	5.3.1	O sistema de e-mails está cheio
4.4.4	5.4.4	Incapaz de direcionar
4.4.5	5.4.5	Congestionamento de e-mails
4.7.*	5.7.*	Violação da política do site

Para controle ainda melhor, o campo key (lado esquerdo) pode conter as tags `Connect`, `To`, `From` e, nas versões 8.12 em diante, `Spam` para controlar a maneira pela qual o filtro é aplicado. `Connect` refere-se a informações de conexão tais como o endereço IP do cliente ou o nome de host do cliente. `To` e `From` referem-se aos endereços de envelope, não a cabeçalhos. A tag `Spam` permite exceções às regras globais por meio dos testes "spam friend" e "spam hater". Ela é ativada com o recurso `delay_checks`:

```
FEATURE(`delay_checks', `friend')
FEATURE(`delay_checks', `hater')
```

O primeiro recurso pula outros conjuntos de regras que poderiam rejeitar a mensagem caso houvesse uma entrada concordante no `access_db` com `FRIEND` como o lado direito do mapeamento. O segundo aplica os demais conjuntos de regras caso `access_db` seja `HATER`. Essas quatro tags fornecem melhor controle sobre a retransmissão e rejeição de correio; elas também sobrescrevem outras restrições. Usuários individuais que reclamam de sua política contra spams válida para todo o site podem ser acomodados com as tags de spam `FRIEND` ou `HATER`.

Se uma dessas tags for usada, a pesquisa será tentada primeiro com a tag `info` e depois sem ela, para manter compatibilidade com versões de bancos de dados de acesso mais antigas.

Eis alguns exemplos:

```
From:spammer@some.domain  REJECT
To:good.domain            RELAY
Connect:good.domain       OK
Spam:abuse@               FRIEND
```

E-mails de spammer@algum.domínio seriam bloqueados, porém você ainda poderia enviar e-mails para esse endereço, mesmo que ele se encontrasse na lista negra. E-mails serão retransmitidos para bom.domínio, porém não dele (supondo-se que a retransmissão tivesse sido desativada em algum ponto). As conexões em bom.domínio seriam permitidas mesmo que ele se encontrasse em uma das listas de rejeição segundo os DNS. E-mails para abuse@dominiolocal passariam, mesmo de spammer@algum.domínio cujo e-mail teria sido rejeitado pela primeira linha do banco de dados de acesso.

Muitos sites usam um banco de dados de acesso para controlar spams ou políticas. A máquina mestre de correio entrante no departamento de ciência de computação da Universidade de Colorado utiliza o recurso `access_db` para rejeitar correio a partir de aproximadamente 500 spammers conhecidos identificados pelos endereços, domínios ou redes IP.

Lista negra de usuários ou sites

Se você tem hosts ou usuários locais para os quais deseja bloquear certos tipos de e-mails, utilize

```
FEATURE(`blacklist_recipients')
```

que suporta os seguintes tipos de entrada em seu arquivo de acesso:

```
To:nobody@                    550 Mailbox disabled for this user
To:printer.mydomain.edu       550 This host does not accept mail
To:user@host.mydomain.edu     550 Mailbox disabled for this user
```

Essas linhas bloqueiam e-mails para o usuário nobody em qualquer host, para a impressora de host e para o endereço de um determinado usuário em uma dada máquina. O emprego da tag `To:` permite que esses usuários enviem mensagens, mas não os deixa recebê-las; algumas impressoras têm essa capacidade.

Infelizmente, hoje em dia é quase impossível manter manualmente uma lista negra como essa. Felizmente, várias listas negras mantidas pela comunidade, estão acessíveis por meio do DNS, e algumas dessas são fornecidas sem nenhum custo.

Para incluir uma lista negra no estilo DNS, utilize o recurso `dnsbl`:

```
FEATURE(`dnsbl', `sbl-xbl.spamhaus.org')
```

Esse recurso faz com que o **sendmail** rejeite e-mails de qualquer site cujo endereço IP esteja na Spamhaus Block List de spammers conhecidos mantida em sbl-xbl.spamhaus.org. Outras listas catalogam sites que executam retransmissões abertas e blocos conhecidos de endereços discados que provavelmente são um esconderijo para spammers.

Essas listas negras são distribuídas por meio de um ajuste fino mais inteligente do sistema DNS; daí, o nome `dnsbl`. Por exemplo, um registro de recurso DNS especial da forma

```
endereço-IP.sbl-xbl.spamhaus.org  IN  A  127.0.0.2
```

colocado no banco de dados DNS do domínio sbl-xbl.spamhaus.org bloquearia e-mails a partir desse host se o recurso `dnsbl` estivesse ativado (pois **sendmail** verificaria explicitamente se um registro deste existia). *endereço-IP* neste exemplo é um endereço de host em sua forma de separação por quatro pontos com a ordem dos octetos invertida.

Você pode incluir o recurso `dnsbl` várias vezes para verificar diferentes listas de corruptores: acrescente apenas um segundo argumento para especificar o servidor de nomes de lista negra e um terceiro com a mensagem de erro que você gostaria que fosse retornada. Se o terceiro argumento fosse omitido, uma mensagem de erro fixa do banco de dados DNS contendo os registros seria retornada.

Verificação de cabeçalhos

Os spammers normalmente tentam ocultar suas identidades. A partir do **sendmail** 8.9, se o endereço From de envelope não estiver no formato *usuário@domínio.valido*, o e-mail será rejeitado. Você pode abrir mão desse comportamento com os recursos a seguir:

```
FEATURE(`accept_unresolvable_domains')
FEATURE(`accept_unqualified_senders')
```

Com o primeiro recurso, o **sendmail** aceita correio de domínios que não existem ou que não estão resolvidos na árvore de atribuição de nomes do DNS. Com o segundo, o **sendmail** aceita endereços From que só contêm um nome de usuário sem nenhuma parte de host ou de domínio. Não use nenhum desses recursos a menos que esteja protegido por um firewall e tenha somente dados DNS locais disponíveis aí. Caso esteja querendo ativar esses recursos, você deverá provavelmente pensar em reprojetar os seus ambientes DNS e **sendmail**. Exigir um endereço de remetente de envelope válido reduz significativamente os spams.

A verificação detalhada de cabeçalhos é um mecanismo anti-spam poderoso que faz uso da sintaxe de arquivo de configuração **sendmail** de baixo nível, que não é tratado aqui. Utilizando a verificação de cabeçalhos, **sendmail** pode procurar padrões especificados em cabeçalhos (por exemplo, "To :friend@public.com") e rejeitar mensagens antes de serem entregues às caixas postais de seus usuários.

A verificação de cabeçalho também pode reconhecer vírus enviados por e-mail se eles tiverem uma linha de cabeçalho distinta. Por exemplo, o vírus Melissa de 1999 continha a linha de assunto: "Important Message From ...". Poucas horas do vírus Melissa ter sido lançado e reconhecido, sendmail.com postou um conjunto de regras local para identificá-lo e descartá-lo.

Quando a impressão digital de um vírus for distinta e fácil de expressar nas regras do **sendmail**, sendmail.com rapidamente postará uma correção para ele tanto em sendmail.com como em sendmail.org.

Para uma amostra representativa das regras de filtragem contra spam e vírus, veja a configuração de **sendmail** para a máquina doméstica de Eric Allman, chamada knecht. Essa configuração está incluída na distribuição **sendmail** como **cf/cf/knecht.mc**. Roube as regras de filtragem de spam e adicione-as ao final do seu arquivo **.mc**.

Ao analisarmos vários exemplos, encontramos regras de verificação de cabeçalhos para:

- E-mails endereçados a qualquer usuário no domínio public.com
- E-mails endereçados a "friend" ou "you"
- E-mails com o cabeçalho X-Spanska, indicativo do worm Happy99
- E-mails com a linha de assunto "Important Message From ..." (o vírus Melissa)
- E-mails com a linha de assunto "all.net and Fred Cohen ..." (o vírus Papa)
- E-mails com a linha de assunto "ILOVEYOU" (o vírus iloveyou e variantes)
- Zilhões de mensagens de spam de marketing
- E-mails com um cabeçalho Outlook Express defeituoso (o worm SirCam)

Todas as regras de verificação de cabeçalho entram nas instruções `LOCAL_CONFIG` e `LOCAL_RULESETS` no final do arquivo de configuração **.mc**. Com o auxílio do comando `divert` do **m4**, **sendmail** simplesmente sabe onde colocá-las no arquivo de configuração bruto.

Até certo ponto, qualquer combate a spams que você implementa bloqueia alguns spammers, mas abre um pouco mais a porta para os remanescentes. Utilize o agente de entrega `error` com uma mensagem de erro "user unknown" em vez do agente de entrega `discard`, pois muitos spammers limpam suas listas. Listas limpas são mais valiosas, de modo que você pode ser removido de algumas caso consiga interceptar o spam, filtrá-lo e responder com uma mensagem de erro.

Taxas e limites de conexão

O **sendmail** 8.13 adicionou um recurso `ratecontrol` que impõe limites por host individual ou por rede individual sobre a taxa em que conexões entrantes são aceitas. Essa restrição pode ser particularmente útil para diminuir a velocidade de spam de fontes que não podem ser completamente bloqueadas sem causar problemas, como grandes ISPs com uma alta porcentagem de usuários com conexões discadas. Para ativar os controles de taxa, adicione uma linha como esta ao arquivo **.mc**:

```
FEATURE(`ratecontrol', `nodelay',`terminate')
```

Além disso, você também deve listar os hosts ou redes a controlar e seus limiares de restrição no arquivo **/etc/mail/access**. Por exemplo, as linhas a seguir

```
ClientRate:192.168.6.17   2
ClientRate:170.65.3.4     10
```

limitam os hosts 192.168.6.17 e 170.65.3.4 a duas novas conexões por minuto e dez novas conexões por minuto, respectivamente.

O recurso `conncontrol` impõe restrições semelhantes sobre o número de conexões simultâneas. Você ativa esse recurso no arquivo.mc com a seguinte linha:

```
FEATURE(`conncontrol', `no delay',`terminate')
```

Como ocorre com `ratecontrol`, você especifica quais hosts e redes limitar em `/etc/mail/access`:

```
ClientConn:192.168.2.8    2
ClientConn:175.14.4.1     7
ClientConn:10
```

Essa configuração resulta em limites de duas conexões simultâneas a 192.168.2.8, sete conexões simultâneas a 175.14.4.1 e dez conexões simultâneas a todos os outros hosts.

Slamming

Um outro recurso inteligente introduzido na versão 8.13 é `greet_pause`. Quando um MTA remoto conecta-se ao seu servidor **sendmail**, o protocolo SMTP obriga que ele espere a saudação de boas-vindas do seu servidor antes de falar. Entretanto, é comum que agentes de entrega de spam (e worms/vírus) anunciem um comando EHLO/HELO imediatamente. Esse comportamento é parcialmente explicável como uma implementação ruim do protocolo SMTP nas ferramentas de envio de spam, mas também pode ser um recurso que tem por objetivo economizar tempo em favor do spammer. Qualquer que seja a causa verdadeira, esse comportamento é suspeito e é conhecido como "slamming".

O recurso `greet_pause` faz com que **sendmail** espere por um período de tempo especificado no começo da conexão antes de saudar seu amigo recém-encontrado. Se o MTA remoto não esperar ser saudado adequadamente e prosseguir com um comando EHLO ou HELO durante o inconveniente momento planejado, o **sendmail** registrará em log um erro e recusará comandos subseqüentes do MTA remoto.

Você pode ativar a pausa de saudação com esta entrada no arquivo **.mc**:

```
FEATURE(`greet_pause', `700')
```

Essa linha causa um retardo de 700 milissegundos no começo de cada nova conexão. É possível configurar retardos por host individual ou por rede individual de maneira semelhante a `conncontrol` e `ratecontrol`, mas a maioria dos sites utiliza um *blanket value* geral para esse recurso.

Miltering: filtragem de e-mails

O **sendmail** versão 8.12 introduziu uma generalização da filtragem de cabeçalho que poderia se transformar em uma ferramenta de combate a spams eficiente. Trata-se de uma API (*application programming interface*) de filtragem de e-mails que as pessoas podem utilizar para desenvolver seus próprios programas de filtragem de e-mails. Esses programas de filtragem ficam entre **sendmail** e mensagens que chegam e podem ser utilizados para reconhecer o perfil de uma mensagem com spam ou vírus e descartá-la ou registrá-la em log (ou tomar qualquer outra providência que você julgue apropriada). Tanto conteúdo de mensagens como de metadados podem ser alvos da filtragem.

Miltering é, potencialmente, uma ferramenta poderosa tanto para combate a spams como para violar a privacidade dos usuários. Administradores que querem saber exatamente quais informações estão saindo da empresa via e-mail podem ser os primeiros a adotá-la. Miltering para e-mail que sai não é disponível na versão 8.12, mas se encontra na lista de prioridades a serem concretizadas.

A biblioteca miltering se chama **libmilter**. **sendmail** invoca a filtragem de entrada com as diretivas de configuração `INPUT_MAIL_FILTER` ou `MAIL_FILTER` e controla a ação de miltering com opções chamadas `MILTER_MACROS_*` que permitem um controle mais refinado sobre os filtros aplicados a cada estágio da conversação SMTP.

Por exemplo, a linha

```
INPUT_MAIL_FILTER(`filtername', `S=mailer:/var/run/filtername.socket')
```

passa cada mensagem que chega ao programa **/etc/mail/nomeDoFiltro** pelo socket especificado no segundo argumento.

Para informações adicionais, veja **libmilter/README** ou a documentação HTML no diretório **libmilter/docs** da distribuição **sendmail**. O arquivo **README** dá uma visão geral e um exemplo simples de um filtro que registra mensagens em um arquivo. Os arquivos no diretório **docs** descrevem a interface da biblioteca e informa como utilizar as diversas chamadas para construir os seus próprios programas de filtragem.

Tratamento de spam

O combate ao spam pode ser uma tarefa difícil e frustrante. Passado determinado ponto, isso também é muito fútil. Não se deixe seduzir pela idéia de capturar spammers individuais, muito embora vários passarão pelos seus escudos antispam. O tempo gasto analisando cabeçalhos de spam e se preocupando com spammers é tempo perdido. Isso mesmo, está em uma luta justa, mas o tempo gasto nestas questões provavelmente não vai reduzir a quantidade de spam que chega ao site.

Você pode agarrar spammers estacionários bem rapidamente classificando-os pelos ISPs deles, mas spammers tipo "bate e corre" que usam uma conta ISP uma única vez e depois a abandonam são muito mais difíceis de ser controlados. Se eles anunciam um site Web, então o site Web é o responsável; se for um número de telefone ou um endereço postal, é mais difícil de identificar o perpetrador, mas não impossível. Spammers móveis parecem basicamente isentos de punição.

As diversas listas negras têm sido um tanto eficientes no bloqueio a spam e reduziram drasticamente o número de retransmissões abertas. Estar em uma lista negra pode afetar seriamente os negócios, de modo que alguns ISPs e empresas são cuidadosos no policiamento de seus usuários.

Nossa principal recomendação em relação a spam é que você utilize as medidas preventivas e listas negras mantidas publicamente que se encontram à disposição. Uma outra possibilidade é redirecionar seus e-mails que chegam para uma empresa terceirizada que combate spam como a Postini (www.postini.com). Entretanto, essa opção talvez resulte em algum comprometimento do desempenho, privacidade ou confiabilidade.

Aconselhe seus usuários a simplesmente eliminar o spam que recebem. Muitas mensagens de spam contêm instruções sobre como aqueles que recebem a mensagem podem ser retirados da lista de endereços. Se você seguir essas instruções, os spammers poderão retirá-lo da lista atual, mas o adicionarão imediatamente a várias outras listas com a nota "chegou a uma pessoa que leu a mensagem". O seu endereço de e-mail passa então a valer ainda mais.

Se quiser pegar carona na luta contra o spam, alguns sites podem ajudar. Um bom site é www.abuso.net. Dois outros que valem ser notados são spamcop.net e cauce.org. O SpamCop tem ferramentas que ajudam a analisar sintaticamente cabeçalhos de correio e determinar o remetente real. O site cauce.org tem boas informações sobre leis de spam. Nos Estados Unidos é possível obter ajuda da Federal Trade Commission. Visite-a em www.ftc.gov/spam.

SpamAssassin

O SpamAssassin é um filtro (que pode ser invocado por meio de um milter do **sendmail**) que é muito eficaz na identificação de spam. Ele utiliza um sistema de pontos para avaliar os pecados de uma mensagem. Ele captura praticamente qualquer spam real e raramente tem resultados falso-positivos. Se uma mensagem acumular muitos pontos (configurável tanto globalmente em um site como por usuário individual), o SpamAssassin marcará essa mensagem. Você pode então regravar as mensagens suspeitas em uma pasta de spam, executando um filtro no lado servidor, como o **sieve** do Cyrus, ou configurando o seu agente de usuário. Você pode até mesmo ensinar mensagens boas e ruins ao SpamAssassin. Não deixe de examinar minuciosamente todo o volume de spam à medida que configura o SpamAssassin e ajusta seus parâmetros. Informações adicionais em spamassassin.apache.org.

SPF e Sender ID

A melhor maneira de combater spam é pará-lo na origem. Isso parece simples e fácil, mas na realidade é um desafio quase impossível. A estrutura da Internet torna difícil monitorar a origem real de uma mensagem e verificar sua autenticidade. A comunidade precisa de uma maneira completamente segura de verificar se a entidade que envia um e-mail é realmente quem ou o que ela afirma ser.

Muitas propostas resolveram esse problema, mas o SPF e o Sender ID tiverem maior aceitação. O SPF (Sender Policy Framework) foi padronizado pelo IETF (RFC4408). Ele define um conjunto de registros TXT do DNS (ver a página 280) pelo qual uma empresa pode identificar suas retransmissões de e-mail externas "oficiais". MTAs podem então recusar qualquer e-mail do domínio dessa empresa se o e-mail não se originar dessas fontes oficiais. Naturalmente, isso só funciona bem se a maior parte da empresas publicar registros SPF. Vários milters disponíveis para download implementam essa funcionalidade no **sendmail**.

O Sender ID e o SPF são praticamente idênticos em relação à forma e função. Entretanto, partes fundamentais do Sender ID foram patenteadas pela Microsoft e por isso tem sido motivo de muita controvérsia. Quando escrevíamos este livro, a Microsoft ainda tentava forçar a indústria a adotar seus padrões proprietários .

18.11 ESTUDO DE CASO DE ARQUIVO DE CONFIGURAÇÃO

Como um estudo de caso sobre a maneira como o **sendmail** é configurado na prática, esta seção revisa os arquivos de configuração da pequena, mas inteligente empresa do **sendmail**, Sendmail, Inc. O projeto de correio deles inclui uma máquina mestre centralizadora de correio tanto para e-mails recebidos como para e-mails enviados. Todos os e-mails são aceitos e roteados imediatamente a um conjunto de servidores IMAP internos que verificam a existência de vírus em cada mensagem antes de liberá-la para a caixa postal de um usuário. A máquina centralizadora de e-mail também verifica a eventual existência de vírus para cada mensagem a ser enviada, de modo que a Sendmail, Inc. nunca é responsável por disseminar vírus por e-mail. Analisamos primeiramente a configuração dos clientes, depois inspecionamos as máquinas mestras mais complexas.

Em todos os exemplos, modificamos ligeiramente os originais, eliminando os textos referentes a direito autoral, eventualmente acrescentando alguns comentários e excluindo a diretiva `dnl` do **m4** nos finais de linhas. Se você utilizar qualquer um desses exemplos como modelo para o seu arquivo **.mc**, não deixe de eliminar os comentários no final das linhas.

Máquinas clientes em sendmail.com

O arquivo **smi-client.mc** para máquinas clientes é bem simples. Ele utiliza a máquina mestra smtp.sendmail.com que é, na verdade, apenas um outro nome para foon.sendmail.com. Utilizar um registro MX (ou um CNAM[31]) para apontar para o servidor de e-mails é uma boa idéia; é fácil de alterar quando você quiser mudar sua máquina mestra para e-mails.

Observe que a data neste arquivo é outubro de 1998. **sendmail** foi atualizado várias vezes desde então, mas o arquivo de configuração não precisou ser modificado.

```
divert(-1)
#####  This file contains definitions for a Sendmail,
#####  Inc. client machine's .mc file.
divert(0)
VERSIONID(`@(#)smi-client.mc 1.0 (Sendmail) 10/14/98')
OSTYPE(`bsd4.4')
FEATURE(`nocanonify')
undefine(`ALIAS_FILE')
define(`MAIL_HUB', `smtp.sendmail.com')
define(`SMART_HOST', `smtp.sendmail.com')
define(`confFORWARD_PATH', `')
MAILER(`local')
MAILER(`smtp')
```

As linhas `MAIL_HUB` e `SMART_HOST` direcionam e-mails que chegam e que saem para o host smtp.sendmail.com. Registros MX em DNS devem cooperar e listar aquele host de maior prioridade (o número mais baixo no registro MX) em vez das máquinas clientes individuais. O caminho para os arquivos **.forward** são configurados como null e o arquivo de alias também é configurado como null; toda a expansão de aliases ocorre na máquina mestra. O recurso `nocanonify` é especificado aqui para poupar tempo, já que as pesquisas DNS são feitas de qualquer maneira no mestre.

Máquina mestra em sendmail.com

A máquina mestra em sendmail.com pode ser uma das instalações de **sendmail** existentes mais atacadas. Ela tem de ser segura contra todos os tipos de ataques de entregas de e-mail que as pessoas possam engendrar e tem de proteger as máquinas por trás dela. Eis seu arquivo de configuração:

```
divert(-1)
# Created with Sendmail Switch, sendmail.com's commercial product.
divert(0)
ifdef(`COMMERCIAL_CONFIG', `INPUT_MAIL_FILTER(`mime-filter', `S=local:/var/run/mime-filter/mime-
    filter.sock')')
LDAPROUTE_DOMAIN(`sendmail.com sendmail.net sendmail.org')
MASQUERADE_AS(`sendmail.com')
MASQUERADE_DOMAIN(`sendmail.com')
RELAY_DOMAIN(`sendmail.com sendmail.net sendmail.org')
define(`MAIL_HUB', `internal-hub.sendmail.com')
define(`QUEUE_DIR', `/var/spool/mqueue/q*')
define(`SMART_HOST', `virus-scan.sendmail.com')
ifdef(`COMMERCIAL_CONFIG', `define(`confCACERT', `/local/certs/cacert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confCACERT_PATH', `/local/certs/trustedcerts')')
define(`confCHECK_ALIASES', `True')
ifdef(`COMMERCIAL_CONFIG', `define(`confCLIENT_CERT', `/local/certs/cert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confCLIENT_KEY', `/local/certs/key.pem')')
define(`confEIGHT_BIT_HANDLING', `mimify')
define(`confLDAP_DEFAULT_SPEC', ` -h "ldap.sendmail.com ldap2.sendmail.com" -b "dc=sendmail,dc=com"
    -p 1389')
define(`confREFUSE_LA', `99')
define(`confRUN_AS_USER', `mailnull')
ifdef(`COMMERCIAL_CONFIG', `define(`confSERVER_CERT', `/local/certs/cert.pem')')
ifdef(`COMMERCIAL_CONFIG', `define(`confSERVER_KEY', `/local/certs/key.pem')')
define(`confTO_IDENT', `0s')
define(`confTO_QUEUEWARN', `2d')
ifdef(`confPOP_TO', `', `define(`confPOP_TO', `900')')
```

31. Um registro MX é, na verdade, mais eficiente que um CNAME; os CNAMEs exigem uma segunda pesquisa no nome verdadeiro para obter o endereço IP.

```
FEATURE(`accept_unqualified_senders')
FEATURE(`accept_unresolvable_domains')
FEATURE(`allmasquerade')
FEATURE(`always_add_domain')
FEATURE(`domaintable')
FEATURE(`ldap_routing', `ldap -1 -v mailHost -k ldap -1 -v mailhost -k
    (&(objectclass=mailRecipient)(|(mail=%0)(|(mailAlternateAddress=%0))))', `ldap -1 -v mail -k
    (&(objectclass=mailRecipient)(|(mailalternateaddress=%0)))', `passthru')
FEATURE(`mailertable')
FEATURE(`masquerade_entire_domain')
FEATURE(`masquerade_envelope')
FEATURE(`relay_entire_domain')
FEATURE(`use_cw_file')
MAILER(`local')
MAILER(`smtp')

LOCAL_RULESETS
SLocal_check_rcpt
R$* $: $&{verify}
ROK $# OK
```

A máquina mestra roteia o e-mail recebido para o servidor interno correto e serve como o host de retransmissão inteligente para o e-mail enviado. Devido às duas linhas a seguir,

```
FEATURE(`accept_unqualified_senders')
FEATURE(`accept_unresolvable_domains')
```

todos os e-mails que chegam são aceitos, mesmo e-mails de remetentes não qualificados e de domínios não conversíveis. Dessa forma, clientes em potencial que possuem **sendmail** ou DNS mal configurado ainda podem passar. Essas regras desfazem os padrões que capturam uma grande quantidade de spams com cabeçalhos forjados. Ident é desativada (tempo-limite configurado em 0) para acelerar a entrega de e-mail que chega.

Essa máquina mestra de e-mails verifica primeiramente as mensagens que chegam para determinados tipos de anexos MIME freqüentemente usados por vírus (instrução `INPUT_MAIL_FILTER`). O **mime-filter** chamado lá contém linhas como:

```
*:anniv.doc:*     error:Your email was not accepted by Sendmail, it appears to be infected with the
    Melissa-X virus.
*:.vbs:*   error:For security and virus protection reasons, Sendmail does not accept messages with
    VBS files attached.  Please retransmit your message without the VBS file.
```

Anexos MIME do tipo **.vba**, **.dot**, **.exe**, **.com**, **.reg** e assim por diante são rejeitados, mas uma varredura completa contra vírus não é feita aqui, pois ela diminuiria a velocidade de processamento de e-mails que chegam. O mestre usa LDAP (com um esquema específico de site) para pesquisar o receptor de cada mensagem e roteá-la para o servidor IMAP/POP interno correto. Se o receptor não for encontrado no banco de dados LDAP, o e-mail será enviado para uma máquina mestra interna (a instrução `MAIL_HUB`) para processamento adicional. Tanto os servidores IMAP/POP como a máquina mestra interna realizam varredura contra vírus antes de encaminharem uma mensagem para a caixa postal de um usuário.

E-mails que saem também são roteados por meio desta máquina mestra por instruções `SMART_HOST` nas máquinas clientes. Para enviar uma mensagem via servidores de e-mail sendmail.com, os hosts fora do domínio sendmail.com têm de apresentar um certificado assinado pela autoridade de certificação de sendmail.com. Funcionários visitando o site de um cliente podem retransmitir e-mails para um terceiro por meio de sendmail.com com esse mecanismo, mas outros não poderão fazê-lo. Essa convenção autentica cada usuário e impede que e-mails forjados transitem por sendmail.com.

Após aceitar e-mails destinados à Internet, a máquina mestra os passa para `SMART_HOST` para verificação da existência de vírus. A máquina mestra de e-mails não é tão ocupada para não poder ela própria fazer a varredura contra vírus, porém, se a varredura fosse feita aí, usuários enviando e-mails teriam de esperar a varredura ser completada antes de suas mensagens realmente serem enviadas. Jogá-las em fila na máquina de varredura de vírus deixa os usuários satisfeitos — suas mensagens parecem partir instantaneamente.

As regras `LOCAL_CONFIG` no final do arquivo de configuração são o local onde a verificação de cabeçalhos contra vários vírus e spammers conhecidos são normalmente colocados. Podem ser encontrados bons exemplos no arquivo **knecht.mc** da distribuição de **sendmail**. Incluímos um exemplo a seguir.

Consulte página 413 para mais detalhes sobre libmilter.

Durante o verão de 2001, o arrasador worm SirCam estava circulando ferozmente. O trecho a seguir extraído do arquivo **knecht.mc** na distribuição **sendmail** o captura. O SirCam é uma das primeiras mensagens sórdidas a ter cabeçalhos aleatórios. As ferramentas usuais para capturá-lo teriam falhado, exceto pelo fato de seus autores terem cometido um erro que diferenciava uma mensagem SirCam de uma verdadeira

mensagem Outlook Express. O conteúdo da mensagem é bem comum (ele solicita sua atenção ao anexo contido na mensagem) e seria, portanto, um candidato para os recursos de filtragem do novo **libmilter** da versão 8.12. Sem garantias de responsabilidade do produto no mundo do software, parece que a única solução para todos esses vírus e worms Microsoft é jogar fora o Windows e instalar o Linux em tudo quanto é canto!

```
LOCAL_RULESETS

KSirCamWormMarker regex -f -aSUSPECT multipart/mixed;boundary=----
    .+_Outlook_Express_message_boundary
HContent-Type:$>CheckContentType

SCheckContentType
R$+             $: $(SirCamWormMarker $1 $)
RSUSPECT        $#error $: "553 Possible virus, see
    http://www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html"

HContent-Disposition:$>CheckContentDisposition

SCheckContentDisposition
R$-             $@ OK
R$- ; $+        $@ OK
R$* $#error     $: "553 Illegal Content-Disposition"
```

Os clientes em sendmail.com não têm nenhum controle de spam nos seus arquivos de configuração. A razão é que todo e-mail que chega ao site passa por um ponto central de correio externo e por um ponto central interno e o spam é separado aí. Alguns dos recursos e outras construções nesse exemplo não são vistos em nossa seção sobre configuração, mas você poderá localizar a documentação sobre eles no arquivo **cf/README**.

18.12 Segurança e sendmail

Com o crescimento explosivo da Internet, programas como o **sendmail** que aceitam entrada arbitrária fornecida pelo usuário e a entrega para usuários locais, arquivos ou shells freqüentemente fornecem uma via de acesso de ataque para hackers. **sendmail**, juntamente com o DNS e mesmo o IP, flerta com a autenticação e a criptografia como uma solução predefinida para algumas dessas questões fundamentais de segurança.

A recente suavização das leis de exportação dos Estados Unidos em relação à criptografia liberou **sendmail** de sair da fábrica com ganchos predefinidos para criptografia. As versões 8.11 e superiores suportam tanto criptografia quanto autenticação SMTP com TLS (Transport Layer Security, segurança de camada de transporte) (previamente conhecida como SSL, Secure Socket Layer). **sendmail** usa o termo TLS nesse contexto e o implementou como uma extensão, STARTTLS, ao protocolo SMTP. TLS trouxe com ele seis novas opções de configuração para arquivos de certificado e arquivos de chaves. Novas ações para concordâncias de banco de dados de acesso podem exigir que a autenticação tenha de ser bem-sucedida.

Nesta seção, descrevemos a evolução do modelo de permissões, propriedades e proteção de privacidade do **sendmail**. A seguir, discutimos brevemente o TLS e o SASL (Simple Authentication and Security Layer) e seus usos com o **sendmail**.

sendmail tem gradualmente enrijecido sua segurança ao longo do tempo e agora está muito chato em relação a permissões de arquivos antes de acreditar no conteúdo de, digamos, um arquivo **.forward** ou **aliases**. Embora esse enrijecimento da segurança geralmente seja bem-visto, algumas vezes é necessário relaxar as rígidas novas políticas. Para esse fim, **sendmail** introduziu a opção `DontBlameSendmail`, assim chamada na esperança de que o nome sugerirá aos administradores de sistemas que aquilo que eles estão fazendo é considerado perigoso.

Essa opção possui vários valores possíveis — 55 na última contagem. O padrão é `safe`. Para uma lista completa desses valores, veja **doc/op/op.ps** na distribuição **sendmail**. Os valores não são listados na segunda edição do livro da O'Reilly, sendmail, mas certamente estarão na terceira. Ou simplesmente deixe a opção configurada em `safe`.

Propriedades

Três contas de usuário são importantes no universo do **sendmail**: o `DefaultUser`, `TrustedUser` e o `RunAsUser`.

Como padrão, todos os agentes de entrega de **sendmail** são executados como `DefaultUser` a menos que os flags do agente de entrega especifiquem algo em contrário. Se um usuário mailnull, sendmail ou daemon existir no arquivo **/etc/passwd**, `DefaultUser` será esse. Caso contrário, o seu padrão será UID 1 e GID 1. Recomendamos o uso da conta mailnull e de um grupo mailnull. Adicione-o a **/etc/passwd** com um asterisco como senha, nenhum shell válido, nenhum diretório de origem e um grupo padrão mailnull. Você também terá de adicionar a entrada mailnull ao arquivo **/etc/group**. A conta mailnull não deve possuir qualquer arquivo. Se o **sendmail** não estiver em execução como root, os agentes de entrega precisam ser setuid.

Se RunAsUser estiver configurado, **sendmail** ignorará o valor de DefaultUser e fará tudo como RunAsUser. Se você estiver executando **sendmail** como setgid (para smmsp), então o **sendmail** de envio passará mensagens para o verdadeiro **sendmail** por meio de SMTP. O verdadeiro **sendmail** não tem o seu bit setuid ativo, mas é executado como root nos arquivos de inicialização.

TrustedUser de **sendmail** pode possuir mapas e arquivos de aliases. TrustedUser tem permissão para iniciar o daemon ou recriar o arquivo **aliases**. Esse recurso existe principalmente para oferecer suporte a interfaces gráficas para **sendmail** que precisam fornecer controle administrativo limitado a certos usuários. Se você ativar TrustedUser, não deixe de proteger a conta para a qual ele aponta, pois essa conta poderá ser facilmente explorada para ganhar acesso root. TrustedUser é diferente da classe TRUSTED_USERS, que determina quem pode reescrever a linha From de mensagens.[32]

RunAsUser é o UID que **sendmail** executa após abrir sua conexão de socket com a porta 25. Portas de numeração inferior a 1.024 podem ser abertas somente pelo superusuário; portanto, **sendmail** tem de ser executado inicialmente como root. Entretanto, após realizar essa operação, **sendmail** pode mudar para um UID diferente. Tal mudança reduz o risco de dano ou acesso se **sendmail** for enganado e levado a fazer algo ruim. Não utilize o recurso RunAsUser em máquinas que suportem contas de usuário ou outros serviços; ele é concebido para uso em firewalls ou hosts protegidos somente.

Como padrão, **sendmail** não muda identidades e continua a ser executado como root. Se você alterar RunAsUser para algo diferente de root, terá de alterar várias outras coisas também. RunAsUser deve ser dono da fila de e-mails, ser capaz de ler todos os mapas e arquivos de inclusão, ser capaz de executar programas etc. Espere gastar algumas horas para descobrir todas as propriedades de arquivo e de diretórios que devem ser alteradas.

Permissões

As permissões sobre arquivos e diretórios são importantes para a segurança de **sendmail**. Utilize as configurações listadas na Tabela 18.14 para estar seguro.

Tabela 18.14 Proprietário e permissões para diretórios relacionados com sendmail

Caminho	Proprietário	Modo	O que ele contém
/var/spool/clientmqueue	smmsp	770	Fila de e-mail para envios iniciais[a]
/var/spool/mqueue	RunAsUser	700	Diretório de fila de e-mails
/, /var, /var/spool	root	755	Caminho para **mqueue**
/etc/mail/*	TrustedUser	644	Mapas, o arquivo de configuração, aliases
/etc/mail	TrustedUser	755	Diretório pai para mapas
/etc	root	755	Caminho para o diretório **mail**

a. Versão 8.12 e superior

sendmail não irá ler arquivos que têm permissões frouxas (por exemplo, arquivos que podem ser gravados pelo grupo ou por todos ou que residam em diretórios que podem ser gravados pelo grupo ou por todos). Parte do rigor de **sendmail** em relação a propriedades e permissões foi motivada pelos sistemas operacionais que permitiam aos usuários transferirem a posse de seus arquivos com **chown** (esses derivados, em grande parte, do System V).

Os sistemas Linux, como padrão, possuem uma versão razoável de **chown** e não permitem a transferência de posse de arquivos. Entretanto, um #ifdef no código (CAP_CHOWN) pode ser utilizado para dar semântica de System V a **chown**. Você teria então que regerar o kernel. Mas esse comportamento é prejudicial; não force seu coerente **chown** do Linux a se comportar da maneira falha adotada pelo System V.

Em particular, **sendmail** é *muito* exigente em relação ao caminho completo a qualquer arquivo de aliases ou arquivo de encaminhamento. Essa exigência algumas vezes colide com a maneira pela qual sites gostam de gerenciar aliases de listas de discussão Majordomo. Se, por exemplo, a lista Majordomo estiver em **/usr/local**, o caminho todo terá de ser admitido como confiável; nenhum componente pode ter permissão de gravação de grupo. Essa restrição torna mais difícil para o proprietário da lista gerenciar o arquivo de aliases. Para ver onde você se encontra em relação às idéias de **sendmail** sobre permissões, execute

```
# sendmail -v -bi
```

O flag **-bi** inicializa o banco de dados de aliases e o alerta sobre permissões inapropriadas.

32. O recurso TRUSTED_USERS normalmente é utilizado para suportar softwares de listas de discussão Por exemplo, se utilizar o Majordomo, você precisará adicionar o usuário "majordomo" à classe TRUSTED_USERS. Os usuários root e daemon são os membros padrão dessa classe.

sendmail não lê mais um arquivo **.forward** que tem uma contagem de links maior do que 1 se o caminho de diretório para ele for inseguro (tiver permissões frouxas). Essa regra pegou a Evi recentemente quando seu arquivo **.forward**, que tipicamente era um link físico (hard link) tanto para **.forward.to.boulder** como para **.forward.to.sandiego**, falhou silenciosamente no encaminhamento de seu e-mail de um pequeno site no qual ela não recebia tantas mensagens. Levou meses antes de ela perceber que "Nunca recebi seu e-mail" era sua própria falha e não uma desculpa válida.

Você pode desativar várias das políticas restritivas de acesso a arquivos mencionadas anteriormente utilizando a opção `DontBlameSendmail`. Mas não faça isso.

E-mails mais seguros para arquivos e programas

Recomendamos que você use **smrsh** em vez de **/bin/sh** como shell do seu programa de entrega de e-mail e que utilize **mail.local** em vez de **/bin/mail** como seu agente de entrega local. Ambos os programas são inclusos na distribuição **sendmail**. Para incorporá-los em sua configuração, adicione as linhas

```
FEATURE(`smrsh',          `caminho-para-smrsh')
FEATURE(`local_lmtp',     `caminho-para-mail.local')
```

ao seu arquivo **.mc**. Se você omitir os caminhos explícitos, os comandos serão assumidos como residentes em **/usr/libexec**. Você pode utilizar a opção `confEBINDIR` de **sendmail** para alterar a localização padrão dos binários para aquilo que desejar. A instalação padrão do Red Hat não inclui **mail.local**. SUSE coloca-o em **/usr/lib/sendmail.d/bin**, e o Debian e o Ubuntu o colocam em **/usr/lib/sm.bin**.

smrsh é um shell restrito que só executa os programas contidos em um diretório (**/usr/adm/sm.bin**, por padrão). O Red Hat e o Fedora instalam o binário do **smrsh** em **/usr/sbin**, o SUSE o coloca em **/usr/lib/sendmail.d/bin**, e o Debian e o Ubuntu o colocam em **/usr/lib/sm.bin**. **smrsh** ignora caminhos especificados pelo usuário e tenta localizar quaisquer comandos solicitados no seu próprio diretório reconhecidamente seguro. **smrsh** também bloqueia o uso de certos metacaracteres de shell como "<", o símbolo de redirecionamento de entrada. São permitidos links simbólicos em **sm.bin**; portanto, você não precisa fazer cópias dos programas que você permite.[33]

Eis alguns exemplos de comandos de shell e suas possíveis interpretações **smrsh**:

```
vacation eric                  # executa /usr/adm/sm.bin/vacation eric
cat /etc/passwd                # rejeitado, cat não está em sm.bin
vacation eric < /etc/passwd    # rejeitado, < não é permitido
```

A opção `SafeFileEnvironment` de **sendmail** controla onde os arquivos podem ser gravados quando os e-mails são redirecionados a um arquivo por meio de um arquivo **aliases** ou **.forward**. Ela faz com que **sendmail** execute uma chamada de sistema **chroot**, fazendo com que a raiz do sistema de arquivos não seja mais **/**, mas sim **/safe**, ou seja lá qual for o caminho especificado na opção `SafeFileEnvironment`. Um alias que direcionou e-mails para o arquivo **/etc/passwd**, por exemplo, realmente seria gravado em **/safe/etc/passwd**.

A opção `SafeFileEnvironment` também protege arquivos de dispositivos, diretórios e outros arquivos especiais permitindo gravações somente em arquivos regulares. Além de melhorar a segurança, essa opção aperfeiçoa os efeitos dos equívocos de usuários. Alguns sites configuram a opção para **/home** para permitir acesso a diretórios de usuários embora mantendo arquivos de sistema fora dos limites.

Os agentes de entrega também podem executar em um diretório ao qual foi aplicado **chroot**. Atualmente, essa opção deve ser especificada na definição do agente de entrega (mailer), mas ela deverá em breve ser configurável com **m4**.

Opções relacionadas com privacidade

sendmail também tem opções de privacidade que controlam

- O que o pessoal de fora pode determinar em relação ao seu site a partir do SMTP
- O que você exige do host na outra ponta de uma conexão SMTP
- Se seus usuários podem ver ou executar a fila de e-mails

A Tabela 18.15 lista os possíveis valores para as opções de privacidade na época em que este livro era escrito; consulte o arquivo **doc/op/op.ps** na distribuição para informações atuais.

---

33. Não coloque programas como o **procmail** que pode gerar um shell em **sm.bin**. E não utilize o **procmail** como o agente de entrega local, porque os usuários podem executar qualquer programa que eles querem a partir do arquivo **~/.procmailrc**. Isso não é seguro.

Tabela 18.15 Valores da variável PrivacyOption

Valor	Significado
`public`	Não realiza nenhuma verificação de privacidade/segurança
`needmailhelo`	Requer HELO do SMTP (identifica host remoto)
`noexpn`	Desativa o comando SMTP EXPN
`novrfy`	Desativa o comando SMTP VRFY
`needexpnhelo`	Não expande endereços (EXPN) sem um HELO
`needvrfyhelo`	Não verifica endereços (VRFY) sem um HELO
`noverb`[a]	Desativa modo prolixo para EXPN
`restrictmailq`	Permite somente que o grupo do diretório **mqueue** visualize a fila
`restrictqrun`	Permite somente que o proprietário do diretório **mqueue** execute a fila
`restrictexpand`	Restringe informações exibidas pelos flags **-bv** e **-v**[b]
`noetrn`[c]	Desativa execuções de fila assíncrona
`authwarnings`	Adiciona o cabeçalho Authentication-Warning (isso ocorre por padrão)
`noreceipts`	Desativa a notificação de status de entrega para recibos retornados com sucesso
`nobodyreturn`	Não retorna um corpo de mensagem em um DSN
`goaway`	Desativa todas as consultas de status SMTP (EXPN, VRFY etc.)

a. O modo prolixo segue arquivos .forward quando um comando EXPN é dado e informa mais informações sobre a localização do e-mail de um usuário. Utilize noverb ou, melhor ainda, noexpn em qualquer máquina exposto ao mundo exterior.
b. A menos que executado pelo root ou pelo `TrustedUser`.
c. ETRN é um comando ESMTP para ser utilizado por um host discado. Ele exige que a fila seja executada apenas no caso de mensagens para esse host.

Recomendamos conservadorismo; utilize:

```
define(`confPRIVACY_OPTIONS', ``goaway, authwarnings, restrictmailq, restrictqrun'')
```

no seu arquivo **.mc**. O valor padrão do **sendmail** para as opções de privacidade é `authwarnings`; a linha acima redefiniria esse valor. Note os conjuntos de aspas duplas; algumas versões de **m4** as requerem para proteger as vírgulas na lista de valores de opção de privacidade. O Red Hat usa como padrão `authwarnings` e o SUSE `authwarnings`, `needmailhelo`, `novrfy`, `noexpn` e `noverb`.

Executando um sendmail com chroot (para o verdadeiro paranóico)

Se estiver preocupado em relação ao acesso que **sendmail** tem sobre o seu sistema de arquivos, você poderá iniciá-lo em uma cela onde se aplicou **chroot** Crie um sistema de arquivos mínimo em sua cela, incluindo coisas como **/dev/null**, arquivos **/etc** essenciais (**passwd**, **group**, **resolv.conf**, **sendmail.cf**, quaisquer arquivos de mapeamento, **mail/***), as bibliotecas compartilhadas de que **sendmail** precisa, o binário **sendmail**, o diretório de filas de e-mails e quaisquer arquivos de log. Provavelmente você terá de mexer na lista para fazer com que ela fique perfeita. Utilize o comando **chroot** para iniciar um **sendmail** deste tipo. Por exemplo:

```
# chroot /jail /usr/sbin/sendmail -bd -q30m
```

Ataques de negação a serviços

Os ataques de negação a serviços são difíceis de serem evitados, pois não há nenhuma maneira *a priori* que determine se uma mensagem é um ataque em vez de um e-mail válido. Os atacantes podem tentar várias coisas sórdidas, entre as quais inundar a porta SMTP com conexões inválidas, preencher partições de discos com mensagens gigantescas, entupir as conexões de saída e bombardeio de e-mail. **sendmail** possui alguns parâmetros de configuração que poderão ajudá-lo a reduzir ou limitar o impacto de um ataque de negação a serviços, mas esses parâmetros poderão também interferir nas mensagens de e-mail legítimas. A biblioteca de filtragem de e-mails (milter) pode ajudar os administradores de sistemas a impedir um ataque prolongado de negação de serviço.

A opção `MaxDaemonChildren` limita o número de processos de **sendmail**. Ela impede que o sistema fique sobrecarregado com trabalho de **sendmail**, mas também permite que um atacante facilmente derrube um serviço SMTP. A opção `MaxMessageSize` pode ajudar a impedir a superlotação do diretório de filas de e-mails, mas se você configurá-la com um

valor muito baixo, e-mails legítimos irão voltar. (Você pode comunicar o seu limite aos usuários para que eles não fiquem surpresos quando seus e-mails voltarem. De qualquer maneira recomendamos um limite relativamente alto, já que alguns e-mails legítimos são imensos.) A opção `ConnectionRateThrottle`, que limita o número de conexões permitidas por segundo, pode desacelerar as coisas um pouco. E, finalmente, configurar `MaxRcptsPerMessage`, que controla o número máximo de receptores permitidos em uma única mensagem, poderá vir a ser útil.

sendmail sempre foi capaz de recusar conexões (opção `REFUSE_LA`) ou colocar e-mails em fila (`QUEUE_LA`) de acordo com a carga média de um sistema. Uma variação, `DELAY_LA`, introduzida na versão 8.12 mantém o correio fluindo, mas a uma velocidade reduzida. Consulte a página 424 na seção sobre desempenho para mais detalhes.

Apesar de todos esses "botões" para ajuste a fim de ativar a proteção de seu sistema de e-mails, alguém que o esteja bombardeando com e-mails ainda interferirá em e-mails legítimos. O bombardeamento de e-mails pode ser bem sórdido.

A Universidade de Colorado fornece a cada aluno (~ 25.000) uma conta de e-mail com **pine** como o leitor padrão de correio. Alguns anos atrás, um estudante recém-contratado em uma loja de informática da região foi convencido a dar uma cópia do arquivo de senhas a seu empregador. A empresa enviou então uma propaganda para todo mundo que aparecia no arquivo de senhas, em lotes de aproximadamente 1.000 receptores por vez (o que gerou uma linha To: muito longa).

O **pine** tinha sido compilado com o modo de resposta padrão configurado para responder a todos os receptores, bem como ao remetente. Muitos estudantes responderam a perguntas como: "Por que você me enviou esse lixo?", e, obviamente, isso também foi enviado a cada uma das pessoas listadas na linha To:. O resultado foi uma negação de serviço total no servidor — para e-mails ou qualquer outro emprego. **sendmail** assume todos os ciclos de CPU, a fila de e-mails ficou enorme e todo o trabalho útil ficou em um estado de travamento. A única solução que parecia colocar a máquina off-line era entrar nas filas de correio e na caixa postal de cada usuário e remover as mensagens prejudiciais. (Uma verificação de cabeçalho na linha Subject também poderia ser utilizada.)

Falsificações

No passado a falsificação de e-mails era trivial. Antigamente, qualquer usuário poderia forjar um e-mail para fazer com que ele parecesse proveniente do domínio de uma outra pessoa. Iniciando com o **sendmail** 8.10, foi instituída a autenticação SMTP para verificar a identidade da máquina de envio. A verificação de autenticação tem de ser ativada com a opção `AuthMechanisms`. Infelizmente, a autenticação de **sendmail** não é de ponta a ponta, mas apenas entre servidores adjacentes. Se uma mensagem for manipulada por diversos servidores, a autenticação ajudará, mas não garantirá que a mensagem não tenha sido forjada.

Da mesma forma, qualquer usuário pode personificar mensagens de correio. Tome cuidado se mensagens forem o veículo de autorização de sua organização para coisas como chaves, cartões de acesso e dinheiro. Você deve alertar os usuários administrativos sobre esse fato e sugerir que caso notem algum e-mail suspeito que pareça provir de uma pessoa com autoridade, eles devem verificar a validade da mensagem. Isto é ainda mais verdadeiro se a mensagem solicitar que privilégios injustificáveis sejam dados a uma pessoa não usual. E-mail autorizando uma grande chave mestra para um estudante pode ser suspeito!

A opção de privacidade `authwarnings` sinaliza tentativas locais de falsificação adicionando um cabeçalho Authentication-Warning a e-mails de saída que parecem ser forjados. Contudo, muitos agentes de usuário ocultam esse cabeçalho por padrão.

Se e-mail forjado estiver vindo de uma máquina que você controla, é possível, na verdade, fazer muito para frustrá-la. Você poderá utilizar o daemon **identd** para verificar o verdadeiro nome de login de um remetente. **sendmail** faz uma chamada de volta ao host remetente para solicitar ao **identd** que está sendo executado lá o nome de login do usuário que está enviando o e-mail. Se **identd** não estiver sendo executado no host remoto, **sendmail** não descobrirá nada. Se a máquina remota for uma estação de trabalho de um único usuário, seu proprietário poderá configurar **identd** para retornar uma resposta falsa. Mas se o host remoto for uma máquina multiusuário como aquela encontrada nos centros de computação de várias universidades, **identd** retornará o verdadeiro nome de login do usuário para **sendmail** colocar no cabeçalho da mensagem.

Muitos sites não usam **identd**; ele normalmente é bloqueado por firewalls. **identd** só é realmente útil dentro de um site, já que as máquinas que você não controla podem mentir. Em um site grande, com usuários um tanto irresponsáveis (por exemplo, uma universidade), esse daemon é fantástico — mas ele também implicará perda de desempenho para **sendmail**.

Há vários anos, quando estávamos experimentando **identd** pela primeira vez, um estudante em nosso site ficou frustrado com os membros de sua equipe de projetos sênior. Ele tentou enviar e-mails para os seus colegas como se fosse o instrutor deles, informando-lhes que sabia que não estavam contribuindo o suficiente e que deveriam trabalhar de forma mais árdua. Infelizmente, ele cometeu um erro de sintaxe e a mensagem rebateu para o instrutor. O uso pelo **sendmail** do protocolo IDENT informou quem ele era. O **sendmail** incluiu as linhas a seguir na mensagem rebatida:

```
The original message was received at Wed, 9 Mar 1994 14:51 -0700 from aluno@benji.Colorado.EDU
    [128.138.126.10]
```

Mas os cabeçalhos da mensagem em si contavam uma outra história:

```
From: instrutor@cs.Colorado.EDU
```

Moral da história: evite erros de sintaxe ao aprontar por aí. Nossa política sobre falsificação de e-mails fez com que o login desse estudante fosse desativado pelo resto do semestre, fazendo, na verdade, exatamente aquilo que o estudante queria, pois, assim, ele não podia trabalhar no projeto e seus companheiros tiveram de trabalhar mais para suprir essa deficiência.

Privacidade de mensagens

Consulte a página 480 para informações adicionais sobre PGP.

A privacidade de mensagens basicamente não existe a menos que você use um pacote de criptografia externo como o PGP (Pretty Good Privacy) ou S/MIME. Por padrão, todas as mensagens são enviadas sem criptografia. A criptografia de ponta a ponta requer suporte de agentes de usuário de e-mails.

Tanto o S/MIME como o PGP são documentados em RFCs, com o S/MIME já no caminho para se tornar um padrão. Entretanto, preferimos o PGP; ele é mais fácil de ser encontrado e foi concebido por um excelente especialista em criptografia, Phil Zimmermann, em quem confiamos. Esses padrões emergentes oferecem uma base para confidencialidade de e-mails, autenticação, garantia da integridade da mensagem e não-repúdio por parte da origem. A análise de tráfego ainda é possível, já que os cabeçalhos e o envelope são enviados em texto simples.

Informe a seus usuários que eles têm de gerar a sua própria criptografia se eles quiserem que seus e-mails sejam privados.

SASL: o Simple Authentication and Security Layer

O **sendmail** 8.10 e versões posteriores oferecem suporte à autenticação SMTP definida na RFC2554. Ele se baseia no SASL (Simple Authentication and Security Layer) (RFC2222). O SASL é um sistema de segredo compartilhado, é geralmente de host para host; você tem de fazer arranjos explícitos para cada par de servidores que devem se autenticar mutuamente.

SASL é um mecanismo de autenticação genérico que pode ser integrado em uma série de protocolos. Até então, o **sendmail**, o **imapd** da Cyrus, o Netscape, o Outlook, o Mulberry e algumas versões do Eudora o utilizam. A estrutura SASL (ele é uma biblioteca) possui dois conceitos fundamentais: um identificador de autorização e um identificador de autenticação. Ele pode mapear estes para permissões em arquivos, senhas de conta, bilhetes Kerberos etc. SASL contém tanto uma parte de autenticação como um componente de criptografia. Para utilizar SASL com o **sendmail**, obtenha o Cyrus SASL em asg.web.cmu.edu/sasl.

TLS, um outro sistema de criptografia/autenticação, é especificado na RFC2487. Ele é implementado no **sendmail** como uma extensão do SMTP chamada STARTTLS. Você pode até mesmo utilizar SASL e TLS juntos.

TLS é um pouco mais difícil de ser configurado e requer uma autoridade de certificação. Você pode pagar uma nota para a VeriSign para emitir seus certificados (chaves públicas assinadas identificando uma entidade) ou configurar sua própria autoridade de certificação. É usada autenticação rigorosa no lugar de um nome de host ou endereço IP como o identificador de autorização para retransmissão de e-mails ou para aceitar uma conexão de um host em primeiro lugar. Uma entrada como

```
TLS_Srv:secure.example.com        ENCR:112
TLS_Clt:laptop.example.com        PERM+VERIFY:112
```

em `access_db` indica que STARTTLS se encontra em uso e que e-mails para o domínio secure.example.com devem ser criptografados com chaves criptográficas de pelo menos 112 bits. E-mails de um host no domínio laptop.example.com devem ser aceitos somente se o cliente tiver se autenticado.

Gregory Shapiro, também da Sendmail, Inc., criou alguns tutoriais excelentes sobre segurança e **sendmail**, que podem ser encontrados em www.sendmail.org/~gshapiro.

18.13 O DESEMPENHO DO SENDMAIL

sendmail tem várias opções de configuração que aprimoram o desempenho. Embora as tenhamos espalhado ao longo do capítulo, tentamos detalhar as mais importantes nesta seção. Há opções e recursos que você deve considerar se cuidar de um sistema de e-mails de grande volume (em ambas as direções). Na verdade, se você realmente precisa enviar 1.000.000 de mensagens de e-mail por hora e não for um spammer, sua melhor opção é utilizar o lado comercial de **sendmail**, da Sendmail, Inc.

Modos de entrega

sendmail tem quatro modos básicos de entrega: segundo plano, interativo, fila e protelação. Cada um deles representa uma relação de compromisso entre latência e throughput. O modo em segundo plano entrega e-mails imediatamente, mas requer que **sendmail** crie um processo para fazê-lo. O modo interativo também entrega imediatamente, mas a entrega é feita pelo mesmo processo e faz com que o lado remoto aguarde resultados. O modo fila enfileira e-mails que chegam para entrega por um controlador de filas em algum momento no futuro. O modo de protelação é similar ao modo fila, mas também protela todas as pesquisas de mapa, de DNS, de aliases e de encaminhamento. O modo interativo é raramente utilizado. O modo em segundo plano favorece baixa latência e os modos de protelação ou de filas aumentam o throughput. O modo de entrega é configurado com a opção `confDELIVERY_MODE` e assume o padrão de segundo plano.

Grupos de filas e divisão de envelopes

Os grupos de filas são um recurso novo do **sendmail** 8.12 que permite que você crie várias filas para mensagens que saem e controle individualmente os atributos de cada grupo de fila. Os grupos de fila podem conter um único diretório de filas ou vários diretórios. Se, por exemplo, o seu computador com Linux for o centralizador de e-mails para um ISP, você poderá definir um grupo de filas para os seus usuários discados e depois dar permissão aos usuários discados para iniciar uma execução de filas (usando o comando ETRN do SMTP) quando eles se conectarem para fazer o download dos e-mails. Os grupos de fila são utilizados com um recurso de divisão envelope com o qual um envelope com muitos destinatários pode ser dividido por todos os grupos de fila. Esse recurso e o emprego de vários diretórios de filas por grupo de filas ajudam a amenizar os problemas de desempenho causados por termos um número excessivo de arquivos em um único diretório de sistema de arquivos.[34]

Quando uma mensagem adentra o sistema de e-mails, ela é atribuída a um ou mais grupos de filas. O grupo de filas para cada receptor é determinado de forma independente. Os envelopes são reescritos para corresponderem a atribuições do grupo de filas. Se forem usados vários diretórios de filas, as mensagens são atribuídas aleatoriamente às filas no grupo de filas correto.

Se um grupo de filas tiver um limite em relação ao número máximo de receptores por envelope, **sendmail** subdividirá o envelope da mensagem em vários envelopes menores que atendam aos parâmetros do grupo de filas.

Os grupos de filas são declarados com diretivas no arquivo **.mc**, mas são realmente configurados na linguagem do arquivo de configuração pelo `LOCAL_RULESETS`, que não descrevemos neste livro. O exemplo a seguir o introduzirá no assunto se você quiser utilizar grupos de filas para aumentar o desempenho ou para oferecer diferentes qualidades de serviço a diferentes destinos.

A Tabela 18.16 lista os atributos que podem ser especificados para um grupo de filas. Somente a primeira letra do nome do atributo precisa ser especificada quando o grupo de filas é definido.

Tabela 18.16 Atributos de grupos de filas

Atributo	Significado
Flags	Principalmente para ajustes futuros; tem de ativar o flag **f** para ter vários executores de filas
Nice	Prioridade para esse grupo de filas; diminui a prioridade de modo semelhante à chamada de sistema **nice**
Interval	Tempo a aguardar entre execuções de filas
Path	Caminho para o diretório de filas associado ao grupo de filas (obrigatório)
Runners	Número de processos **sendmail** a ser executados concorrentemente no grupo de filas
recipients	Número máximo de receptores por envelope

Eis um exemplo que possui grupos de filas para e-mails locais, e-mails destinados a aol.com, e-mails destinados a outros sites remotos e uma fila padrão para todos os demais e-mails. As linhas a seguir entram na parte comum do arquivo **.mc**:

```
dnl ##### -- queues
QUEUE_GROUP(`local', `P=/var/spool/mqueue/local')
QUEUE_GROUP(`aol', `P=/var/spool/mqueue/aol, F=f, r=100')
QUEUE_GROUP(`remote', `P=/var/spool/mqueue/remote, F=f')
```

E depois no final do arquivo **.mc**:

```
LOCAL_RULESETS
Squeuegroup
R<$+>              $1
R$*@aol.com        $# aol
R$*@mydomain.com   $# local
R$*@$*             $# remote
R$*                $# mqueue
```

Nesse exemplo, especificamos um limite de 100 receptores por mensagem ao definirmos o grupo de filas AOL. Se uma mensagem de saída tivesse 10.000 receptores, dos quais 1.234 estivessem em AOL, a subdivisão de envelope colocaria 13 mensagens no grupo de filas `aol`, 12 de 100 receptores cada e um com os 34 restantes.

Para acelerar as coisas ainda mais, tente a divisão rápida, que posterga pesquisas MX durante o processo de classificação:

```
define(`confFAST_SPLIT', `1')
```

34. Se você usa um sistema de arquivos ext3 com um kernel 2.6 ou posterior, os índices de diretório também podem ajudar a reduzir o impacto sobre o desempenho dos grandes diretórios. Você pode modificar um sistema de arquivos existente para utilizar esse recurso com o comando **tune2fs -O dir_index**.

Executores de filas

sendmail cria cópias de si mesmo para executar o transporte real das mensagens. Você pode controlar o número de cópias de **sendmail** que está sendo executado em um dado momento e até mesmo quantas são alocadas a cada grupo de filas. Utilizando esse recurso, você pode equilibrar as atividades do **sendmail** e do sistema operacional na suas ocupadas máquinas centralizadoras de e-mail.

Três opções de **sendmail** controlam o número de daemons de executores de filas processando cada fila:

- A opção `MAX_DAEMON_CHILDREN` especifica o número total de cópias do daemon **sendmail** que são permitidas para serem executadas de uma só vez, incluindo aquelas filas em execução e as que aceitam mensagens que chegam
- A opção `MAX_QUEUE_CHILDREN` configura o número máximo de executores de filas por vez
- A opção `MAX_RUNNERS_PER_QUEUE` configura o limite de executores padrão por fila se não for especificado nenhum valor explícito no parâmetro `Runners=` (ou `R=`) na definição do grupo de filas

Se você configurar valores que podem conflitar (por exemplo, um máximo de 50 executores de filas no total, porém 10 para a fila local, 30 para a fila mydomain e 50 para a fila AOL), **sendmail** colocará em lote as filas nos grupos de trabalho e fará um rodízio (*round robin*) entre os grupos de trabalhos. Nesse exemplo, as filas local e mydomain seriam um grupo de filas e as filas AOL seriam um segundo grupo de trabalho. Se você optar por limites que certamente entrarão em conflito (por exemplo, max=50, porém AOL=100), **sendmail** usará `MAX_QUEUE_CHILDREN` como seu limite absoluto no número de executores de filas.

Controles de carga média

sendmail sempre foi capaz de recusar conexões ou colocar mensagens em filas em vez de entregá-las quando a carga média do sistema fica muito alta. Infelizmente, a carga média possui uma granularidade de apenas um minuto, portanto não é uma ferramenta muito efetiva para o controle dos recursos consumidos por **sendmail**. A nova primitiva `DELAY_LA` permite configurar um valor de carga média a partir do qual **sendmail** deve desacelerar; ela entrará em hibernação por um segundo entre comandos SMTP para as conexões atuais e antes de aceitar novas conexões. O valor padrão é 0, que desliga o mecanismo.

Mensagens não-entregáveis na fila

Mensagens que se encontram em uma fila de e-mails, mas que não podem ser entregues, realmente podem acabar com o desempenho de servidor de e-mails. **sendmail** possui vários recursos que o ajudam na questão de mensagens que não podem ser entregues. A mais eficiente é a opção `FALLBACK_MX`, que despacha uma mensagem para uma outra máquina se ela não puder ser entregue na primeira tentativa. Sua máquina primária então processa rapidamente as mensagens para bons endereços e põe de lado filhos problemáticos em uma máquina de reserva secundária. Uma outra ajuda é o diretório de status de host, que armazena o status dos hosts remotos por todas as execuções de fila.

A opção `FALLBACK_MX` é um grande trunfo para melhoria de desempenho para um site com grandes listas de discussão que invariavelmente contêm endereços que não podem, temporária ou permanentemente, ser atingidos. Para utilizá-lo, você deve especificar um host para tratar o correio adiado. Por exemplo,

```
define(`confFALLBACK_MX', `mailbackup.atrust.com')
```

encaminharia todas as mensagens que falhassem em sua primeira tentativa de entrega para o servidor central mailbackup.atrust.com para processamento posterior. Na versão 8.12, pode haver múltiplas máquinas de reserva se os hosts designados tiverem múltiplos registros MX no DNS.

Nas máquinas de reserva você pode utilizar a opção `HOST_STATUS_DIRECTORY` para ajudar em falhas múltiplas. Essa opção direciona **sendmail** a manter um arquivo de estados para cada host para o qual as mensagens são enviadas e a utilizar essas informações de estados para priorizar os hosts cada vez que a fila for executada. Essas informações de estados implementam caching negativo e permitem que informações sejam compartilhadas entre as execuções de filas. É um ganho de desempenho em servidores que manipulam listas de discussão com um grande número de endereços ruins, mas isso pode ser caro em termos de I/O de arquivos.

Eis um exemplo que utiliza o diretório **/var/spool/mqueue/.hoststat** (crie o diretório em primeiro lugar):

```
define(`confHOST_STATUS_DIRECTORY', `/var/spool/mqueue/.hoststat')
```

Se o diretório **.hoststat** for especificado com um caminho relativo, ele será armazenado abaixo do diretório de filas. **sendmail** cria a sua própria hierarquia interna de subdiretórios tomando como base o nome de host de destino.

Por exemplo, se e-mails para evi@anchor.cs.colorado.edu fossem falhar, as informações de status iriam para o diretório **/var/spool/mqueue/.hoststat/edu./colorado./cs./** em um arquivo chamado **anchor**, pois o host anchor possui um registro MX que aponta para ele mesmo como de maior prioridade. Se os registros MX do DNS tivessem direcionado mensagens de e-mail do anchor para o host foo, então o nome de arquivo teria sido **foo** e não anchor.

Uma terceira melhoria em termos de desempenho para máquinas muito ativas envolve configurar um período de fila mínimo de modo que qualquer mensagem que não possa ser entregue na tentativa inicial possa ser colocada em fila e permanecer lá por um período mínimo entre tentativas. Essa técnica normalmente é acoplada a flags em linhas de comando que executam a fila mais freqüentemente (por exemplo, **-q5m**). Se um executor de filas ficar travado em uma mensagem incorreta, um outro se iniciará em 5 minutos, aumentando o desempenho para as mensagens que podem ser entregues. A fila inteira é executada em lotes determinados pelos quais mensagens ficaram por lá pelo tempo mínimo exigido. Executar **sendmail** com os flags **-bd -q5m** e incluir a opção

```
define(`confMIN-QUEUE_AGE', `27m')
```

no arquivo de configuração poderia resultar em um sistema mais responsivo.

Ajustando o kernel

Caso pretenda utilizar um computador com Linux como servidor de e-mails de alto volume, você deve modificar vários parâmetros de configuração de rede do kernel e, talvez até mesmo gerar um kernel personalizado (dependendo de sua configuração de hardware e carga esperada). Remova quaisquer drivers desnecessários para iniciar com um kernel mais enxuto que é suficiente para a sua configuração de hardware.

O kernel personalizado poderia incluir suporte a vários processadores se a máquina host tivesse mais do que um processador (SMP). (Percebemos que, para os verdadeiros *geeks* do Linux, esse comentário é análogo a um lembrete para não se esquecer de respirar. Mas como as capacidades para geração de um kernel variam entre os usuários do Linux, ignoramos os comentários dos revisores e deixamos esse lembrete aqui.)

Para reinicializar os parâmetros dos programas de rede, utilize o comando **echo** do shell redirecionado para a variável apropriada no sistema de arquivos **/proc**. O Capítulo 12, *Redes TCP/IP*, contém uma descrição genérica desse procedimento a partir da página 219. A Tabela 18.17 mostra os parâmetros para alterar em um servidor de e-mails de alto volume juntamente com seus valores sugeridos e padrões. Essas alterações provavelmente deveriam ser colocadas em um script de shell que é executado em tempo de inicialização e realiza os **echos** correspondentes.

Tabela 18.17 Parâmetros de kernel a serem modificados em servidores de e-mails de alto volume

Variável (relativo a /proc/sys)	Padrão	Sugerido
net/ipv4/tcp_fin_timeout	180	30
net/ipv4/tcp_keepalive_time	7200	1800
net/core/netdev_max_backlog	300	1024
fs/file_max	4096	16384
fs/inode_max	16384	65536

Você poderia, por exemplo, utilizar o comando

```
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
```

para alterar o valor de tempo-limite FIN do TCP.

18.14 ESTATÍSTICAS, TESTES E DEPURAÇÃO DE SENDMAIL

sendmail pode coletar informações estatísticas sobre o número e tamanho de mensagens que ele manipulou. Exibe esses dados com o comando **mailstats**, que organiza os dados por agente de entrega local. A opção `STATUS_FILE` do **sendmail** (no arquivo `OSTYPE`) especifica o nome do arquivo em que as estatísticas devem ser mantidas. A existência do arquivo especificado ativa a função de contabilidade.

O local padrão para o arquivo de informações estatísticas de **sendmail** é **/etc/mail/statistics**, mas alguns fornecedores chamam o arquivo **sendmail.st** e o colocam em **var/log** ou **/usr/lib**. Os totais exibidos pelos **mailstats** são cumulativos desde a criação do arquivo de estatísticas. Se desejar estatísticas periódicas, você pode fazer um rodízio e reinicializar o arquivo por meio de **cron**.

Eis um exemplo:

```
$ mailstats
Statistics from Tue Aug  1 02:13:30 2006
 M  msgsfr  bytes_from     msgsto  bytes_to      msgsrej        msgsdis      Mailer
 4      12        25K         63      455K            0              0      esmtp
 7       0         0K         18       25K            0              0      relay
 8      54       472K          0        0K            0              0      local
=============================================================
 T      66       497K         81      480K            0              0
 C      66                    81                      0
```

Se o arquivo de estatísticas de e-mails tiver permissão de leitura para todo o mundo, você não precisa ser root para executar **mailstats**.

Seis valores são mostrados: mensagens e kilobytes recebidos (`msgsfr`, `bytes_from`), mensagens e kilobytes enviados (`msgsto` , `bytes_to`), mensagens rejeitadas (`msgsrej`) e mensagens descartadas (`msgsdis`). A primeira coluna é um número identificando o agente de entrega e a última coluna lista o nome do agente de entrega. A linha `T` indica o total de mensagens em bytes e a linha `C` indica as conexões. Esses valores incluem tanto e-mails locais como retransmitidos.

Testando e depurando

As configurações baseadas em **m4** são, de certo modo, testadas de antemão. Provavelmente não será necessário fazer depuração de baixo nível se você utilizá-las. Uma coisa que os flags de depuração não são capazes de testar é o seu projeto. Enquanto pesquisávamos este capítulo, encontramos erros em vários dos projetos e arquivos de configuração que examinamos. Os erros variavam desde chamar um recurso sem a macro pré-requisito (por exemplo, usar `masquerade_envelope` sem ter ativado o mascaramento com `MASQUERADE_AS`) até conflito total entre o projeto da configuração de **sendmail** e o firewall que controlava se e sob quais condições as mensagens tinham permissão para entrar.

Você não pode projetar um sistema de e-mails no vácuo. Você tem de estar sincronizado com (ou pelo menos não estar em conflito) com seus registros MX do DNS e sua política de firewall.

sendmail fornece um dos conjuntos mais ricos do mundo em termos de ferramentas para depuração, com os flags de depuração que não são simples valores booleanos (verdadeiro ou falso) nem mesmo inteiros, mas quantidades bidimensionais *x.y*, onde *x* seleciona o tópico e *y* seleciona a quantidade de informações a ser exibida. Um valor igual a 0 indica nenhuma depuração e 127 desperdiça muitas árvores se você imprimir a saída. Os tópicos variam de 0 a 99; atualmente são definidos cerca de 80.

O arquivo **sendmail/TRACEFLAGS** na distribuição lista os valores em uso e os arquivos e funções nas quais eles são utilizados. Todo o suporte de depuração se encontra no nível do arquivo de configuração. Em muitos casos, é útil examinar a fonte de **sendmail** juntamente com a saída de depuração.

Se **sendmail** for chamado com um flag **-d***x.y*, a saída da depuração vai para a tela (erro padrão). A Tabela 18.18 ilustra valores importantes de *x* e alguns valores sugeridos para *y*. Seja cauteloso caso ative a depuração para um **sendmail** executado como daemon (**bd**), pois a saída de depuração pode acabar misturada no diálogo SMTP e provocar falhas estranhas quando **sendmail** dialogar com hosts remotos.

Tabela 18.18 Tópicos de depuração

Tópico	Significado e sugestões
0	Mostra flags de compilação e identidade de sistema (experimente y = 1 ou 10)
8	Mostra a resolução de nomes DNS (experimente y = 78)
11	Rastreia a entrega (mostra chamadas ao agente de entrega)
12	Mostra a conversão de nomes de local para remoto
17	Lista hosts MX
21	Rastreia regras de reescrita (use y = 2 ou y = 12 para mais detalhes)
27	Mostra aliases e encaminhamento (experimente y = 4)
44	Mostra tentativas de abertura de arquivos no caso de falhas (y = 4)
60	Mostra pesquisas de mapas de bancos de dados

checksendmail está disponível em www.harker.com.

Gene Kim e Rob Kolstad escreveram um script Perl chamado **checksendmail** que chama **sendmail** em modo de teste de endereço com um arquivo de endereços de teste que você fornece. Ele compara os resultados àqueles esperados. Esse script permite testar novas versões do arquivo de configuração em relação a um conjunto de testes dos endereços típicos de seu site para, certificar-se de que você inadvertidamente não quebrou qualquer coisa que costumava funcionar.

Entrega prolixa

Muitos agentes de usuário que chamam **sendmail** na linha de comandos aceitam um flag **-v**, que é passado para **sendmail** e faz com que ele exiba as etapas tomadas para entregar a mensagem. O exemplo a seguir usa o comando **mail**. Os termos em negrito foram digitados como entrada para o agente de usuário, e o restante é saída prolixa de **sendmail**.

```
$ mail -v trent@toadranch.com
Subject: just testing, please ignore
hi
```

```
.
Cc:
trent@toadranch.com... Connecting to coyote.toadranch.com. via esmtp...
220 coyote.toadranch.com ESMTP Sendmail 8.11.0/8.11.0; Tue, 7 Aug 2001 20:08:51 -0600
>>> EHLO anchor.cs.colorado.edu
250-coyote.toadranch.com Hello anchor.cs.colorado.edu [128.138.242.1], pleased to meet you
250-ENHANCEDSTATUSCODES
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250-AUTH DIGEST-MD5 CRAM-MD5
250 HELP
>>> MAIL From:<evi@anchor.cs.colorado.edu> SIZE=65
250 2.1.0 <evi@anchor.cs.colorado.edu>... Sender ok
>>> RCPT To:<trent@toadranch.com>
250 2.1.5 <trent@toadranch.com>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 2.0.0 f7828pi03229 Message accepted for delivery
trent@toadranch.com... Sent (f7828pi03229 Message accepted for delivery)
Closing connection to coyote.toadranch.com.
>>> QUIT
221 2.0.0 coyote.toadranch.com closing connection
```

O **sendmail** de **anchor** se conectou ao **sendmail** em toadranch.com. Cada máquina usou o protocolo ESMTP para negociar o intercâmbio da mensagem.

Falando em SMTP

Você pode fazer uso direto do SMTP ao depurar o sistema de e-mails. Para iniciar uma sessão SMTP, use **sendmail -bs** ou **telnet** para porta TCP 25. Por padrão, essa é a porta em que **sendmail** ouve quando executado no modo de daemon (**-bd**); o **sendmail** usa a porta 587 ao ser executado como o agente de envio de correio. A Tabela 18.19 mostra os comandos SMTP mais importantes.

Tabela 18.19 Comandos SMTP

Comando	Função
HELO *nomeDoHost*	Identifica o host de conexão se estiver falando SMTP
EHLO *nomeDoHost*	Identifica o host de conexão se estiver falando ESMTP
MAIL From: *revpath*	Inicia uma transação de e-mail (remetente de envelope)
RCPT To: *fwdpath*[a]	Identifica receptor(es) de envelopes
VRFY *endereço*	Verifica se o *endereço* é válido (entregável)
EXPN *endereço*	Mostra a expansão de mapeamentos de aliases e **.forward**
DATA	Inicia o corpo da mensagem[b]
QUIT	Finaliza o intercâmbio e fecha a conexão
RSET	Reinicializa o estado da conexão
HELP	Imprime uma síntese dos comandos SMTP

a. Pode haver múltiplos comandos RCPT para uma mensagem.

b. Você termina o corpo inserindo um ponto em uma linha própria.

A linguagem completa possui somente 14 comandos, de modo que é bastante fácil aprendê-la e usá-la. Ela não distingue o uso de letras maiúsculas ou minúsculas. A especificação para o SMTP pode ser localizada na RFC2821.

A maioria dos agentes de transporte, inclusive **sendmail**, fala tanto SMTP como ESMTP; **smap** é a exceção isolada nos dias de hoje. Infelizmente, muitos computadores para firewalls que fornecem filtragem ativa não falam ESMTP.

Aqueles que falam ESMTP começam a conversa com o comando EHLO, e não HELO. Se o processo na outra ponta entende e responde com um OK, os participantes então negociam extensões suportadas e chegam a um mínimo denominador comum para a troca. Se for retornado um erro, então aquele que fala ESMTP retorna ao uso do SMTP.

Monitoramento de fila

Você pode utilizar o comando **mailq** (que é equivalente ao **sendmail -bp**) para visualizar o status das mensagens enfileiradas. As mensagens são "enfileiradas" enquanto são entregues ou quando a entrega foi tentada, mas falhou.

mailq imprime um resumo legível dos arquivos em **/var/spool/mqueue** a qualquer dado momento. Essa saída é útil para determinar por que uma mensagem talvez tenha sido protelada. Se parecer que está ocorrendo um acúmulo de correio, você pode monitorar o status das tentativas do **sendmail** de limpar o congestionamento.

Observe que na versão 8.12 do **sendmail** e superiores, há duas filas padrão: uma para mensagens recebidas na porta 25 e outra para mensagens recebidas na porta 587 (a fila de envio de cliente). Você pode invocar **mailq -Ac** para ver a fila do cliente.

Eis uma saída típica de **mailq**. Esse caso mostra três mensagens que esperam ser entregues:

```
$ sudo mailq
/var/spool/mqueue (3 requests)
-----Q-ID-----     --Size--       -----Q-Time-----       ------------Sender/Recipient-----------
k623gYYk008732     23217          Sat Jul 1 21:42        MAILER-DAEMON
      8BITMIME   (Deferred: Connection refused by agribusinessonline.com.)
                                                         <Nimtz@agribusinessonline.com>
k5ULkAHB032374     279            Fri Jun 30 15:46       <randy@atrust.com>
                 (Deferred: Name server: k2wireless.com.: host name lookup fa)
                                                         <relder@k2wireless.com>
k5UJDm72023576     2485    Fri Jun 30 13:13       MAILER-DAEMON
                 (reply: read error from mx4.level3.com.)
                                                         <lfinist@bbnplanet.com>
```

Se você acha que entende a situação melhor que o **sendmail** ou apenas quer que o **sendmail** tente outra vez entregar imediatamente as mensagens enfileiradas, você pode forçar uma execução de fila com **sendmail -q**. Se utilizar **sendmail -q -v**, o **sendmail** mostra os resultados detalhados de cada tentativa de entrega, o que freqüentemente é útil para depuração. Deixado para seus próprios dispositivos, o **sendmail** tenta novamente entregar cada intervalo de execução de fila (em geral 30 minutos).

Registrando em log

Consulte o Capítulo 10 para informações adicionais sobre syslog.

sendmail usa syslog para registrar em log erros e mensagens de estados com o recurso syslog "mail" e níveis "debug" por meio de "crit"; as mensagens são marcadas com a string "sendmail". Você pode sobrescrever a string de registro em log "sendmail" com a opção de linha de comando **-L**, que é útil se você estiver depurando uma cópia de **sendmail** enquanto outras cópias realizam tarefas de e-mail regulares.

A opção `confLOG_LEVEL`, especificada na linha de comandos ou no arquivo de configuração, determina o nível de gravidade que **sendmail** usa como um limiar para registro em log. Valores altos do nível de log implicam níveis de gravidade baixos e fazem com que mais informações sejam registradas em log.

Lembre-se de que uma mensagem registrada em syslog em um determinado nível é relatada a esse nível e a todos aqueles acima dele. O arquivo **/etc/syslog.conf** determina o destino conseqüente de cada mensagem.

Nos sistemas Red Hat e Fedora, os logs do **sendmail** são colocados em **/var/log/maillog** por padrão. Para o SUSE, é **/var/log/mail**; para o Debian e o Ubuntu, é **/var/log/mail.log**. Não seria interessante se os esforços de padronização pudessem pôr em ordem algumas dessas diferenças aleatórias aparentemente sem sentido para que nossos scripts fossem mais portáveis?

A Tabela 18.20 dá uma idéia aproximada do mapeamento entre níveis de log de **sendmail** e os níveis de gravidade de syslog.

Tabela 18.20 Níveis de log de sendmail *versus* níveis de syslog

L	Níveis	L	Níveis
0	Nenhum registro em log	4	notice
1	alerta ou crítico	5–11	info
2	crit	≥ 12	debug
3	erro ou alerta		

Um programa muito interessante chamado **mreport**, de Jason Armstrong, se encontra disponível em

ftp://ftp.riverdrums.com/pub/mreport

Ele resume arquivos de log gravados pelo **sendmail**. Ele pode ser compilado sem problemas simplesmente usando-se **make** e **make install**. Eis um exemplo da saída gerada por **mreport** em um sistema Red Hat:

```
# mreport -f -i /var/log/maillog -o mreport.out
[redhat.toadranch.com] [/var/log/maillog]

*  [      7]     592601  herb@yoherb.com               trent@toadranch.com
*  [      8]     505797  SNYDERGA@simon.rochester.edu  trent@toadranch.com
 [  1]     179386  steph@toadranch.com                 bennettr@ci.boulder.co.us
 [  1]     65086   herb@yoherb.com                     ned@xor.co
 [  7]     19029   evi@anchor.cs.colorado.edu          trent@toadranch.com
 [  11]    17677   lunch-request@moose.org             trent@toadranch.com
 [  2]     16178   trent@toadranch.com                 ned@camelspit.org
 [  3]     15229   reminders@yahoo-inc.com             herb@toadranch.com
 [  2]     4653    trent@toadranch.com                 garth@cs.colorado.edu
 [  2]     1816    UNKNOWN                             trent@toadranch.com

. . . muitas linhas excluídas . . .

====================
Total Bytes        : 7876372
Number of Records  : 192
--------------------
User Unknown       : 125
--------------------
Host Name          : redhat.toadranch.com
Input File         : maillog
Output File        : mreport.out
First Record       : Aug  5 04:47:31
Last Record        : Aug  7 18:16:25
--------------------
Time Taken         :  24317 µs
====================
```

Você tem de utilizar **sudo** ou ser root para executar **mreport** se os seus arquivos de log de e-mail forem lidos apenas por root (como deveriam ser). Os flags e as opções estão documentados na página man do **mreport**. O flag **-f** nesse caso diz para agregar e ordenar por remetente; um flag análogo **-t** permite ordenar por receptor. O argumento **-i** é o arquivo de entrada e **-o** o arquivo de saída.

Caso pretenda executar **mreport** regularmente, você deve informar os seus usuários. Eles podem achar uma invasão de privacidade os administradores de sistemas ficarem xeretando seus logs de e-mails por meio de uma ferramenta tão legal como essa.

18.15 O SISTEMA DE E-MAIL EXIM

O Debian e o Ubuntu são distribuídos com o agente de transporte de correio Exim em vez do **sendmail**. Trata-se de um sistema mais simples com menos frescuras, mas também mais fácil de ser configurado para sites que não apresentam uma carga pesada de e-mails e não possuem um projeto de sistema de e-mails bizarro. A funcionalidade do Exim corresponde aos recursos mais comumente utilizados de **sendmail**.

Histórico

O Exim foi escrito em 1995 por Philip Hazel da Universidade de Cambridge e é distribuído sob a GNU Public License. A versão 4 do Exim (conhecida como exim4) foi distribuída em 2002 e atualmente é a versão distribuída com o Debian e o Ubuntu. Philip Hazel também escreveu um livro chamado *The Exim Smtp Mail Server: Official Guide for Release 4* que foi publicado em 2003. Submetemos os detalhes sobre a configuração do Exim a esse livro e à documentação do Exim em www.exim.org e só forneceremos uma breve descrição aqui.

Exim no Linux

Como no conjunto **sendmail**, alguns comandos separados no Exim desempenham funções de correio específicas. Essas são realmente implementadas por uma chamada a **exim** com certos flags de linhas de comando. A Tabela 18.21 ilustra os comportamentos e seus flags equivalentes.

Tabela 18.21 Utilitários do Exim (com flags equivalentes)

Comando	Equivalente	Função
mailq	exim -bp	Mostra a fila de e-mails
rsmtp	exim -bS	Conexão SMTP em lote
rmail	exim -i	Para compatibilidade com smail [a]
runq	exim -q	Executa a fila de e-mail

a. Aceita uma mensagem terminada com um ponto sozinho em uma linha.

Configuração do Exim

O Exim contém três partes lógicas: diretores, roteadores e transportadores. Os diretores tratam dos endereços locais, isto é, endereços dentro do domínio de origem. Os roteadores manipulam endereços remotos, e os transportadores fazem a entrega em si.

O Exim é configurado de maneira muito parecida com o **smail3** ou Postfix, com a linguagem de configuração assumindo a forma *palavra-chave = valor*. No Exim 4, o sistema é configurado por meio de uma coleção elaborada de arquivos de configuração que residem em **/etc/exim4/conf.d**. Depois de adicionar ou modificar um desses arquivos, você deve executar o script **update-exim4.conf**, que então cria **/var/lib/exim4/config.autogenerated**. O arquivo **config.autogenerated** é utilizado pelo Exim em tempo de execução.

O arquivo **exim.conf** é muito bem documentado, com comentários precedendo cada variável que descrevem o que a variável faz, qual é o valor usual e que coisas (ruins) poderiam acontecer se você não conseguisse defini-la. Os arquivos de configuração padrão têm cerca de 100 variáveis definidas, mas isto é em grande parte linguagem padrão que imita muito do comportamento de **sendmail** documentado nas primeiras seções.

Em nosso sistema de testes tivemos de configurar somente quatro variáveis para fazer com que o sistema básico para e-mails funcionasse. Na verdade, duas destas variáveis se relacionavam com questões de privacidade e não eram absolutamente exigidas. Eis as quatro variáveis, com nossos comentários ao lado:

```
qualify_domain = nomeDoDominio   ### por padrão configurado como nome de host não-qualificado
local_domains = localhost:nomeDoHost
smtp_verify = false              ### padrão é on, off desativa SMTP VRFY
modemask = 002                   ### padrão 022 assume um grupo para cada usuário
```

Na próxima seção, descrevemos alguns recursos do Exim em termos do **sendmail** para que você possa comparar as funcionalidades dos dois sistemas e decidir se quer instalar o **sendmail** real em vez do Exim.

Semelhanças entre o Exim e o sendmail

Algumas partes do Exim executam setuid para root quase como o **sendmail** fazia antes da versão 8.12. Portanto, é particularmente importante permanecer atualizado em relação aos patches de segurança. O conceito do Exim de usuários confiáveis coincide com aquele do **sendmail** e ajuda principalmente a facilitar o gerenciamento de listas de discussão, para as quais linhas From são rotineiramente reescritas. O Exim também lhe permite definir usuários administrativos que têm permissão para iniciar execuções de filas. O comando verify do SMTP (VRFY) é permitido por padrão.

exim tem de receber o sinal SIGHUP com o comando **kill** quando seu arquivo de configuração muda. Ele armazena seu ID de processo em **/var/spool/exim/exim-daemon.pid**. Em geral, ele se conecta aos arquivos no diretório **/var/log/exim**, mas também pode utilizar o syslog. Como padrão, o **exim** registra em log nos seus próprios arquivos e ignorando syslog, embora **syslog.conf** seja configurado com vários arquivos de log de e-mail que são vazios e sofrem rotação e são compactados todos os dias. (Arquivos vazios são maiores após a compactação do que antes dela!)

O Exim permite encaminhamento de correio enviado a um host inteligente e filtragem de correio interno tanto no nível de host como de usuário. Ele suporta domínios virtuais e possui um banco de dados para nova tentativa, similar em funcionalidade ao diretório de estados de host de **sendmail**, para cuidar das entregas difíceis. Um mecanismo de filtragem que abranja todo o sistema (não por usuário) pode verificar a existência de anexos Microsoft, worms, vírus etc.

O Exim inclui um excelente recurso que não se encontra disponível em **sendmail**, ou seja, reconhecimento de loops em aliases e em encaminhamento e sua manipulação inteligente.

Por fim, **man -k exim** nos leva a diversas ferramentas úteis para ajudar a manter o sistema de e-mails organizado.

18.16 POSTFIX

O Postfix ainda é uma outra alternativa ao **sendmail**. O projeto Postfix começou quando Wietse Venema teve uma licença remunerada no T. J. Watson Research Center da IBM. Os objetivos do projeto Postfix incluíam uma diretiva de distribuição de código-fonte aberto, desempenho rápido, robustez, flexibilidade e segurança. É um concorrente direto do **qmail** de Dan Bernstein. Todas as principais distribuições do Linux incluem o Postfix e, desde a versão 10.3, o Mac OS X também o distribui em vez do **sendmail** como o sistema padrão de correio.

Os aspectos mais importantes sobre o Postfix são, primeiro, que ele funciona praticamente como ele sai da fábrica (os arquivos de configuração mais simples têm apenas uma ou duas linhas de comprimento) e, segundo, que ele tira vantagem dos mapas de expressões regulares para filtrar e-mails de modo efetivo, especialmente em conjunção com a biblioteca PCRE (Perl Compatible Regular Expression). O Postfix é compatível com **sendmail** no sentido de que os arquivos **aliases** e **.forward** do Postfix têm o mesmo formato e semântica daqueles do **sendmail**.

O Postfix fala ESMTP. Domínios virtuais e filtragem de spam são suportados. O Postfix não utiliza uma linguagem de rescrita de endereço como o **sendmail** utiliza; em vez disso, ele conta com pesquisas de tabela dos arquivos simples Berkeley DB, **dbm**, LDAP, NIS, NetInfo ou bancos de dados como o MySQL.

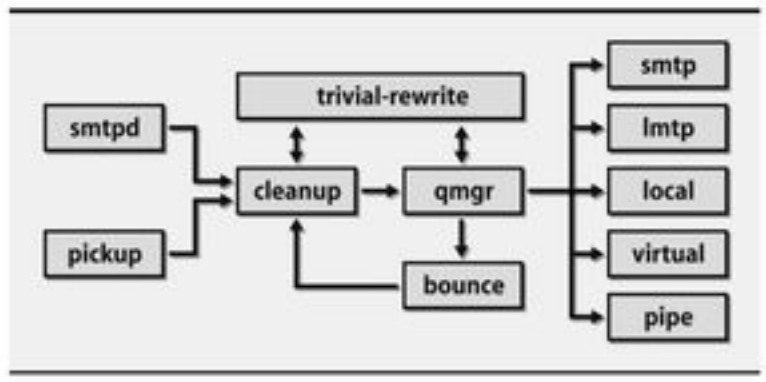
Arquitetura do Postfix

O Postfix abrange vários pequenos programas que atuam cooperativamente e que enviam mensagens pela rede, recebem mensagens, entregam e-mails localmente etc. A comunicação entre esses programas é realizada por meio de sockets de domínio UNIX ou FIFOs. Essa arquitetura é bem diferente daquela do **sendmail**, pelo fato de que um único grande programa realiza a maior parte do trabalho.

O programa **master** inicia e monitora todos os processos do Postfix. O arquivo de configuração desse programa, **master.cf**, lista os programas subsidiários juntamente com as informações sobre a maneira como eles devem ser iniciados. Os valores padrão configurados nesse arquivo estão corretos para tudo, exceto para máquinas (ou redes) muito rápidas ou muito lentas; em geral, nenhum ajuste é necessário. Uma alteração comum é remover o comentário de um programa, por exemplo, **smtpd**, quando um cliente não deve ouvir na porta SMTP.

Os programas de servidor mais importantes envolvidos na entrega de e-mails são mostrados na Ilustração E a seguir.

Ilustração E Programas de servidor Postfix



Recebendo e-mail

O e-mail é recebido na porta SMTP pelo **smtpd**, que também verifica se os clientes que se conectam têm autorização de enviar o e-mail que eles estão tentando entregar. Quando o e-mail é enviado localmente por meio do programa de compatibilidade **/usr/lib/sendmail**, um arquivo é gravado no diretório **/var/spool/postfix/maildrop**. Esse diretório é varrido periodicamente pelo programa **pickup**, que processa todos os novos arquivos que ele localiza.

Todos os e-mails que chegam passam por **cleanup**, que adiciona cabeçalhos ausentes e regrava endereços de acordo com os mapas `canonical` e `virtual`. Antes de inseri-lo na fila `incoming`, **cleanup** envia o e-mail a **trivial-rewrite**, que faz correções menores dos endereços, como acrescentar um domínio de correio a endereços que não são completamente qualificados.

O gerenciador de filas

E-mail esperando pela entrega é controlado pelo **qmgr**, o gerenciador de cinco filas:

- `incoming` – email que está chegando
- `active` – e-mail que está sendo entregue
- `deferred` – e-mail no qual a entrega falhou no passado
- `hold` – e-mail bloqueado na fila pelo administrador
- `corrupt` – e-mail que não pode ser lido nem analisado sintaticamente

O gerenciador de fila normalmente seleciona a próxima mensagem a processar com uma estratégia FIFO simples, mas ele também suporta um algoritmo de preempção complexo que prefere mensagens com poucos destinatários ao volumoso correio

Para não sobrecarregar um host receptor, especialmente depois que ele estava inativo, o Postfix usa um algoritmo de inicialização lenta para controlar a rapidez com que ele tenta entregar e-mails. Mensagens adiadas recebem um registro de data/hora do tipo "tente novamente" que atrasa exponencialmente de modo que não sejam desperdiçados recursos nas mensagens não-entregáveis. Um cache de status de destinos inacessíveis evita tentativas de entrega desnecessárias.

Enviando e-mail

qmgr decide com a ajuda de **trivial-rewrite** aonde uma mensagem deve ser enviada. A decisão de roteamento feita por **trivial-rewrite** pode ser sobrescrita pelo mapa `transport`.

A entrega para hosts remotos por meio do protocolo SMTP é realizada pelo programa **smtp**. **lmtp** entrega correio utilizando o LMTP (Local Mail Transfer Protocol), definido na RFC2033. O LMTP é baseado no SMTP, mas o protocolo foi modificado para que o servidor de correio não precisasse gerenciar uma fila de e-mails. Esse agente de entrega local é particularmente útil para entregar e-mails para servidores de caixa postal como o conjunto Cyrus IMAP.

O trabalho do **local** é entregar e-mails localmente. Ele converte endereços na tabela **aliases** e segue as instruções localizadas nos arquivos **.forward** dos destinatários. As mensagens são encaminhadas a um outro endereço, passadas para um programa externo para processamento ou armazenadas nas pastas de correio dos usuários.

O programa **virtual** entrega e-mail para "caixas postais virtuais"; isto é, caixas postais que não estão relacionadas com uma conta Linux local, mas que ainda representam destinos válidos de e-mail. Por fim, **pipe** implementa a entrega por meio de programas externos.

Segurança

O Postfix implementa a segurança em vários níveis. A maioria dos programas de servidor Postfix pode executar em um ambiente **chroot**. Eles são programas separados sem relacionamentos pai/filhos. Nenhum deles são setuid. O diretório mail drop é gravável pelo grupo postdrop, para o qual o programa **postdrop** é setgid.

Impressionantemente, nenhuma exploração além de ataques de negação de serviço foi identificada em qualquer versão do Postfix.

Comandos e documentação Postfix

Vários utilitários de linha de comando permitem que os usuários interajam com o sistema de correio:

- **sendmail**, **mailq**, **newaliases** – são substituições compatíveis com o **sendmail**
- **postfix** – inicia e pára o sistema de e-mails (deve ser executado como root)
- **postalias** – cria, modifica e consulta tabelas de alias
- **postcat** – imprime o conteúdo dos arquivos de fila
- **postconf** – exibe e edita o arquivo de configuração principal, **main.cf**
- **postmap** – cria, modifica ou consulta tabelas de pesquisa
- **postsuper** – gerencia as filas de e-mails

A distribuição do Postfix inclui um conjunto de páginas man que descreve todos os programas e suas opções. Adicionalmente, documentos on-line em www.postfix.org explicam como configurar e gerenciar vários aspectos do Postfix. Esses mesmos documentos estão incluídos na distribuição do Postfix, portanto, você deve encontrá-los instalados no seu sistema, normalmente em um diretório chamado **README_FILES**.

Configurando o Postfix

O arquivo **main.cf** é o principal arquivo de configuração do Postfix. O arquivo **master.cf** configura os programas de servidor. Ele também define várias tabelas de pesquisa que são referenciadas a partir de **main.cf** e que fornecem diferentes tipos de mapeamentos de serviço.

A página man do **postconf**(5) descreve cada parâmetro que pode ser configurado no arquivo **main.cf**. Se simplesmente digitar **man postconf**, você obterá a página man do programa **postconf**. Utilize **man -s 5 postconf** para obter a versão que descreve as opções **main.cf**.

A linguagem de configuração do Postfix é um pouco semelhante a uma série de comentários e instruções de atribuição do shell Bourne. As variáveis podem ser referenciadas na definição de outras variáveis sendo prefixadas com um $. As definições das variáveis são armazenadas à medida que aparecem no arquivo de configuração; elas não são expandidas até serem utilizadas e quaisquer substituições ocorrem nesse momento.

Você pode criar novas variáveis atribuindo-lhe valores. Tenha cuidado para escolher nomes que não entram em conflito com as variáveis existentes de configuração.

Todos os arquivos de configuração do Postfix, inclusive as tabelas de pesquisa, consideram as linhas que iniciam com espaço em branco como linhas de continuação. Essa convenção resulta em arquivos de configuração muito legíveis, mas você deve iniciar novas linhas na coluna um.

O que colocar em main.cf

Mais de 300 parâmetros podem ser especificados no arquivo **main.cf** Entretanto, alguns deles precisam ser configurados em um site médio, uma vez que os padrões, na maior parte, são bons. O autor do Postfix recomenda veementemente que apenas parâmetros com valores não-padrão sejam incluídos na sua configuração. Dessa maneira, se o valor padrão de um parâmetro mudar no futuro, sua configuração adotará automaticamente o novo valor.

O arquivo **main.cf** de exemplo que vem com a distribuição inclui muitos parâmetros de exemplo comentados, juntamente com alguma breve documentação. É melhor deixar a versão original apenas como referência. Inicie com um arquivo vazio para sua própria configuração de modo que suas especificações não fiquem perdidas em um mar de comentários.

Configurações básicas

Vamos começar com a configuração mais simples possível: um arquivo vazio. Surpreendentemente, essa é uma configuração Postfix perfeitamente razoável. Ela resulta em um servidor de correio que entrega e-mail localmente dentro do mesmo domínio que o do nome de host local e que envia todas as mensagens direcionadas para endereços não-locais diretamente aos servidores remotos apropriados.

Uma outra configuração simples é um "cliente nulo"; isto é, um sistema que não entrega nenhum e-mail localmente, mas, em vez disso, encaminha e-mail externo a um servidor central designado. Para essa configuração, definimos vários parâmetros, iniciando com `mydomain`, que define a parte do domínio do nome de host e, `myorigin`, que é o domínio de correio acrescentado a endereços de e-mails não-qualificados. Se os parâmetros `mydomain` e `myorigin` forem idênticos, podemos escrever algo como isto:

```
mydomain = cs.colorado.edu
myorigin = $mydomain
```

Outro parâmetro que devemos configurar é `mydestination`, que especifica os domínios de correio que são locais. (Estes também são conhecidos como domínios "canônicos".) Se o endereço do destinatário de uma mensagem tiver `mydestination` como seu domínio de correio, a mensagem é entregue pelo programa **local** para o usuário correspondente (supondo que nenhum alias nem o arquivo **.forward** relevante foi localizado). Se mais de um domínio de correio for incluído em `mydestination`, todos esses domínios são considerados aliases para o mesmo domínio.

Não queremos nenhuma entrega local para nosso cliente nulo, portanto esse parâmetro deve estar vazio:

```
mydestination =
```

Por fim, o parâmetro `relayhost` instrui o Postfix a enviar todas as mensagens não-locais a um host especificado em vez de enviá-las diretamente aos seus destinos aparentes:

```
relayhost = [mail.cs.colorado.edu]
```

Os colchetes instruem o Postfix a tratar a string especificada como um nome de host (registro DNS A) em vez de um nome de domínio de correio (registro MX do DNS).

Como clientes nulos não devem receber e-mails de outros sistemas, a última coisa a fazer em uma configuração de cliente nulo é remover o comentário da linha **smtpd** no arquivo **master.cf**. Essa alteração impede que o Postfix execute **smtpd**. Com apenas essas poucas linhas, definimos um cliente nulo completamente funcional!

Para um servidor de correio "real", você precisará de algumas outras opções de configuração e também de algumas tabelas de mapeamento. Abrangemos estas em algumas seções a seguir.

Utilizando postconf

postconf é uma ferramenta útil que o ajuda a configurar o Postfix. Quando executada sem argumentos, ela imprime todos os parâmetros da maneira como estão configurados atualmente. Se nomear um parâmetro específico como um argumento, **postconf** imprimirá o valor desse parâmetro. A opção **-d** faz com que **postconf** imprima os padrões em vez dos valores configurados atualmente. Por exemplo:

```
$ postconf mydestination
mydestination =
$ postconf -d mydestination
mydestination = $myhostname, localhost.$mydomain, localhost
```

Uma outra opção útil é **-n**, que faz com que **postconf** imprima apenas os parâmetros que diferem do padrão. Se pedir ajuda na lista de discussão do Postfix, essas são as informações de configuração que você deve colocar no seu e-mail.

Tabelas de pesquisa

Muitos aspectos do comportamento do Postfix são modelados pelo uso de tabelas de pesquisa, que podem mapear chaves para valores ou implementar listas simples. Por exemplo, a configuração padrão para a tabela `alias_maps` é

```
alias_maps = dbm:/etc/mail/aliases, nis:mail.aliases
```

As origens de dados são especificadas com a notação *tipo:caminho*. Note que essa tabela em particular na verdade utiliza duas fontes distintas de informações simultaneamente: a um banco de dados **dbm** e um mapa NIS.

Múltiplos valores podem ser separados por vírgulas, espaços ou ambos. A Tabela 18.22 lista as origens de dados disponíveis; **postconf -m** também mostra essas informações.

Tabela 18.22 As fontes de informações utilizáveis como tabelas de pesquisa do Postfix

Tipo	Descrição
`dbm / sdbm`	Arquivo de banco de dados **dbm** ou **gdbm** tradicional
`cidr`	Endereços de rede na forma CIDR
`hash / btree`	Tabela de hash Berkeley DB (substituição a **dbm**) ou arquivo B-tree
`ldap`	Serviço do diretório LDAP
`mysql`	Banco de dados MySQL
`nis`	Serviço de diretório NIS
`pcre`	Expressões regulares compatíveis com Perl
`pgsql`	Banco de dados PostgreSQL
`proxy`	Acesso por meio do **proxymap**, por exemplo, para escapar de um **chroot**
`regexp`	Expressões regulares POSIX
`static`	Retorna o valor especificado como *path* independentemente da chave
`unix`	Arquivos **/etc/passwd** e **/etc/group** do Linux; utiliza a sintaxe NIS[a]

a. `unix:passwd.byname` é o arquivo **passwd** e `unix:group.byname` é o arquivo **group**.

Use os tipos `dbm` e `sdbm` somente para compatibilidade com a tabela de alias **sendmail** tradicional. Berkeley DB (`hash`) é uma implementação mais moderna; é mais segura e mais rápida. Se a compatibilidade não for um problema, utilize

```
alias_database = hash:/etc/mail/aliases
alias_maps = hash:/etc/mail/aliases
```

O `alias_database` especifica a tabela que é reconstruída por **newaliases** e deve corresponder à tabela que você especifica em `alias_maps`. A razão de ter dois parâmetros é que `alias_maps` poderia incluir fontes não-DB como `mysql` ou `nis` que não precisam ser recriadas.

Todas as tabelas de classes DB (`dbm`, `sdbm`, `hash` e `btree`) estão baseadas em um arquivo de texto que é compilado para um formato binário pesquisável eficientemente. A sintaxe para esses arquivos de texto é semelhante àquela dos arquivos de configuração com relação a comentários e linhas de continuação. As entradas são especificadas como pares chave/valor simples separados por espaço em branco, exceto para tabelas de aliases, que devem ter um dois-pontos depois da chave para reter compatibilidade **sendmail**. Por exemplo, as linhas a seguir são apropriadas para uma tabela de aliases:

```
postmaster:    david, tobias
webmaster:     evi
```

Como outro exemplo, eis uma tabela de acesso para retransmissão de e-mail a partir de qualquer cliente com um nome de host que termina em cs.colorado.edu.

```
.cs.colorado.edu   OK
```

Os arquivos de texto são compilados para seus formatos binários com o comando **postmap** para tabelas normais e com o comando **postalias** para tabelas de aliases. A especificação da tabela (incluindo o tipo) deve ser dada como o primeiro argumento. Por exemplo:

```
$ postmap hash:/etc/postfix/access
```

postmap também pode consultar valores em uma tabela de pesquisa:

```
$ postmap -q blabla hash:/etc/postfix/access
$ postmap -q .cs.colorado.edu hash:/etc/postfix/access
OK
```

Entrega local

O programa **local** é responsável pela entrega de correio para o domínio canônico. Ele também trata aliases locais. Por exemplo, se `mydestination` for configurado como cs.colorado.edu e um e-mail para evi@cs.colorado.edu chegar, **local** primeiro realiza uma pesquisa na tabelas `alias_maps` e então substitui todas as entradas correspondentes recursivamente.

Se nenhum alias corresponder, **local** procura um arquivo.**forward** no diretório inicial do usuário evi e segue as instruções nesse arquivo se ele existir. (A sintaxe é a mesma que aparece do lado direito de um mapa de aliases.) Por fim, se nenhum **.forward** for localizado, o e-mail será entregue para a caixa postal local de evi.

Por padrão, **local** grava os arquivos no formato mbox padrão sob **/var/mail**. Você pode alterar esse comportamento com os parâmetros mostrados na Tabela 18.23.

Tabela 18.23 Parâmetros para entrega local de caixa postal (configurado em main.cf)

Parâmetro	Descrição
`mail_spool_directory`	Entrega correio para um diretório central e serve todos os usuários
`home_mailbox`	Entrega correio para *~usuário* sob o caminho relativo especificado
`mailbox_command`	Entrega e-mails com um programa externo, em geral **procmail**
`mailbox_transport`	Entrega e-mails por meio de um serviço como definido em **master.cf**[a]
`recipient_delimiter`	Permite nomes de usuário estendidos (ver descrição a seguir)

a. Essa opção cria uma interface com os servidores de caixa postal como o Cyrus **imapd**.

As opções `mail_spool_directory` e `home_mailbox` normalmente geram caixas postais no formato mbox, mas também podem gerar caixas postais **Maildir** no estilo do **qmail**. Para solicitar esse comportamento, adicione uma barra ao final do nome de caminho.

Se `recipient_delimiter` for configurado como `+`, e-mails endereçados a evi+qualquer_coisa@cs.colorado.edu são aceitas para entrega à conta evi. Com esse recurso, os usuários podem criar endereços de uso especial e classificar seus e-mails por endereço de destino. O Postfix primeiro tenta pesquisar o endereço completo e, apenas se isso falhar, é que ele pesquisa os componentes estendidos e recorre ao endereço básico. O Postfix também procura um arquivo de encaminhamento correspondente, **.forward+whatever**, para mais aliases.

Domínios virtuais

Se quiser hospedar um domínio de correio no seu servidor de e-mails Postfix, você tem três escolhas:

- Listar o domínio em `mydestination`. A entrega é realizada como descrito acima: os aliases são expandidos e o e-mail é entregue aos usuários correspondentes.
- Listar o domínio no parâmetro `virtual_alias_domains`. Esta opção fornece ao domínio um espaço de nome de endereçamento próprio que é independente das contas de usuário do sistema. Todos os endereços dentro do domínio devem ser conversíveis (por meio de mapeamento) para os endereços reais fora dele.
- Listar o domínio no parâmetro `virtual_mailbox_domains`. Como ocorre com a opção `virtual_alias_domains`, o domínio tem um espaço de nome próprio. Entretanto, o e-mail pode ser entregue a toda caixa postal sob um caminho especificado, independentemente das contas de usuário.

Liste o domínio apenas em um desses três lugares. Escolha cuidadosamente, pois muitos elementos de configuração dependem dessa escolha. Já revisamos como tratar o método `mydestination`. As outras opções são discutidas a seguir.

Domínios virtuais de aliases

Se um domínio estiver listado como um valor do parâmetro `virtual_alias_domains`, e-mails para esse domínio serão aceitos pelo Postfix e deverão ser encaminhados para um destinatário real na máquina local ou em outra parte.

O encaminhamento de endereços no domínio virtual deve ser definido em uma tabela de pesquisa incluída no parâmetro `virtual_alias_maps`. As entradas nessa tabela têm o endereço no domínio virtual no lado esquerdo e o endereço real de destino, no lado direito.

Um nome não-qualificado à direita é interpretado como um nome de usuário na máquina local.

Considere o exemplo a seguir de **main.cf**:

```
myorigin = cs.colorado.edu
mydestination = cs.colorado.edu
virtual_alias_domains = admin.com
virtual_alias_maps = hash:/etc/mail/admin.com/virtual
```

Em **/etc/mail/admin.com/virtual** poderíamos então ter as linhas:

```
postmaster@admin.com      evi, david@admin.com
abuse@admin.com           evi
david@admin.com           david@schweikert.ch
evi@admin.com             evi
```

O correio para evi@admin.com seria redirecionado para evi@cs.colorado.edu (`myorigin` foi acrescentado) e, em última instância, seria entregue para a caixa postal do usuário evi porque cs.colorado.edu foi incluído em `mydestination`.

As definições podem ser recursivas: o lado direito pode conter endereços que ainda são mais definidos no lado esquerdo. Observe que o lado direito só pode ser uma lista de endereços. Se você precisar executar um programa externo ou utilizar os arquivos `:include:`, redirecione os e-mails a um alias, que pode então ser expandido de acordo com suas necessidades.

Para manter tudo em um arquivo, você pode configurar `virtual_alias_domains` como a mesma tabela de pesquisa de `virtual_alias_maps` e adicionar uma entrada especial à tabela para marcá-la como um domínio virtual de aliases. Em **main.cf**:

```
virtual_alias_domains = $virtual_alias_maps
virtual_alias_maps = hash:/etc/mail/admin.com/virtual
```

Em **/etc/mail/admin.com/virtual**:

```
admin.com                 notused
postmaster@admin.com      evi, david@admin.com
...
```

O lado direito da entrada para o domínio de correio (admin.com) nunca é utilizado; a existência de admin.com na tabela como uma entrada independente é suficiente para fazer com que o Postfix considere-o um domínio virtual de aliases.

Domínios virtuais de caixa postal

Os domínios listados sob `virtual_mailbox_domains` são semelhantes aos domínios locais (canônicos), mas a lista de usuários e suas caixas postais correspondentes devem ser gerenciados de maneira independente das contas de usuário do sistema.

O parâmetro `virtual_mailbox_maps` aponta para uma tabela que lista todos os usuários válidos no domínio. O formato do mapa é

```
usuário@domínio    /caminho/para/caixaPostal
```

Se o caminho terminar com uma barra, as caixas de correio serão armazenadas no formato **Maildir**. O valor de `virtual_mailbox_base` sempre é prefixado para os caminhos especificados.

Talvez você queira criar com freqüência alguns aliases dos endereços no `virtual_mailbox_domain`. Utilize um `virtual_alias_map` para fazer isso. Eis um exemplo completo.

Em **main.cf**:

```
virtual_mailbox_domains = admin.com
virtual_mailbox_base = /var/mail/virtual
virtual_mailbox_maps = hash:/etc/mail/admin.com/vmailboxes
virtual_alias_maps = hash:/etc/mail/admin.com/valiases
```

/etc/mail/admin.com/vmailboxes poderia conter entradas como estas:

```
evi@admin.com            nemeth/evi/
```

/etc/mail/admin.com/valiases poderia conter:

```
postmaster@admin.com     evi@admin.com
```

Você pode utilizar mapas de aliases virtuais até mesmo nos endereços que não são domínios de aliases virtuais. Mapas de aliases virtuais permitem redirecionar qualquer endereço a partir de qualquer domínio, independentemente do tipo do domínio (canônico, alias virtual ou caixa postal virtual). Como os caminhos da caixa postal só podem ser colocados no lado direito do mapa de caixa postal virtual, o uso desse mecanismo é a única maneira de configurar aliases nesse domínio.

Controle de acesso

Os servidores de e-mails devem retransmitir o correio a terceiros somente em favor de clientes confiáveis. Se um servidor de e-mails encaminhar correio a partir de clientes desconhecidos para outros servidores, isso é chamado de retransmissão aberta, que é ruim. Consulte a seção *Retransmissão,* na página 407, para mais detalhes sobre essa questão.

Felizmente, o Postfix não atua como uma retransmissão aberta por padrão. De fato, os padrões do Postfix são bem restritivos; é mais provável que você precise liberar permissões do que torná-las mais rígidas. O controle de acesso para transações SMTP é configurado no Postfix por meio de "listas de restrição de acesso". Os parâmetros mostrados na Tabela 18.24 controlam o que deve ser verificado durante as diferentes fases de uma sessão SMTP.

Tabela 18.24 Parâmetros Postfix para restrição de acesso SMTP

Parâmetro	Quando aplicado
`smtpd_client_restrictions`	Na solicitação de conexão
`smtpd_helo_restrictions`	No comando HELO/EHLO (início da sessão)
`smtpd_sender_restrictions`	No comando MAIL FROM (especificação de remetente)
`smtpd_recipient_restrictions`	No comando RCPT TO (especificação de destinatário)
`smtpd_data_restrictions`	No comando DATA (corpo do e-mail)
`smtpd_etrn_restrictions`	No comando ETRN[a]

a. Esse é um comando especial utilizado para reenviar mensagens na fila.

O parâmetro mais importante é `smtpd_recipient_restrictions`, visto que o controle de acesso é mais facilmente realizado quando o endereço do destinatário é conhecido e pode ser identificado como sendo local ou não. Todos os outros parâmetros na Tabela 18.24 permanecem vazios na configuração padrão. O valor padrão é

```
smtpd_recipient_restrictions =   permit_mynetworks, reject_unauth_destination
```

Cada uma das restrições especificadas é testada sucessivamente até uma decisão definitiva sobre o que fazer com o e-mail ser tomada. A Tabela 18.25 mostra as restrições mais comuns.

Tabela 18.25 Restrições de acesso Postfix comuns

Restrição	Função
`check_client_access`	Verifica o endereço do host de cliente utilizando uma tabela de pesquisa
`check_recipient_access`	Verifica o endereço do destinatário utilizando uma tabela de pesquisa
`permit_mynetworks`	Concede acesso aos endereços listados em `mynetworks`
`reject_unauth_destination`	Rejeita e-mail de destinatários não-locais; sem retransmissão

Tudo pode ser testado nessas restrições, não apenas informações específicas como o endereço de remetente em `smtpd_sender_restrictions`. Portanto, por simplicidade, é recomendável colocar todas as restrições sob um único parâmetro, que deve ser `smtpd_sender_restrictions`, uma vez que ele é o único que pode testar tudo (exceto a parte DATA).

`smtpd_recipient_restriction` é onde a retransmissão de correio é testada. Você deve manter a restrição `reject_unauth_destination` e cuidadosamente escolher as restrições de "permissão" antes dela.

Tabelas de acesso

Cada restrição retorna uma das ações mostradas na Tabela 18.26. As tabelas de acesso são utilizadas nas restrições como `check_client_access` e `check_recipient_access` para selecionar uma ação com base no endereço do host de cliente ou no endereço do destinatário, respectivamente.

Tabela 18.26 Ações para tabelas de acesso

Ação	Significado
4nn texto	Retorna o código temporário de erro *4nn* e a mensagem *text*
5nn texto	Retorna o código permanente de erro *5nn* e a mensagem *text*
`DEFER_IF_PERMIT`	Se as restrições resultarem em `PERMIT`, altere-as para um erro temp
`DEFER_IF_REJECT`	Se as restrições resultarem em `REJECT`, altere-as para um erro temp
`DISCARD`	Aceita a mensagem, mas a descarta silenciosamente
`DUNNO`	Finge que a chave não foi localizada; testa outras restrições
`FILTER` *transporte:dest*	Passa o e-mail pelo filtro *transport:dest*[a]
`HOLD`	Bloqueia o e-mail na fila
`OK`	Aceita o e-mail
`PREPEND` *cabeçalho*	Adiciona um cabeçalho à mensagem
`REDIRECT` *endereço*	Encaminha esse e-mail a um endereço especificado
`REJECT`	Rejeita o e-mail
`WARN` *mensagem*	Insere o aviso *message* dado nos registros em log

a. Consulte a seção sobre tratamento de spam e de vírus no Postfix a partir da página 431.

Como exemplo, suponha que você quisesse permitir retransmissões para todas as máquinas dentro do domínio `cs.colorado.edu` e também quisesse permitir que apenas clientes confiáveis postassem uma mensagem na lista de discussão interna `newsletter@cs.colorado.edu`. Você poderia implementar essas diretivas com as seguintes linhas em **main.cf**:

```
smtpd_recipient_restrictions =
            check_client_access hash:/etc/postfix/client_access
            check_recipient_access hash:/etc/postfix/recipient_access
            reject_unauth_destination
```

Observe que as vírgulas são opcionais quando a lista dos valores para um parâmetro é especificada.

Em **/etc/postfix/client_access**:

```
.cs.colorado.edu  OK
```

Em **/etc/postfix/recipient_access**:

```
newsletter@cs.colorado.edu REJECT Internal list
```

O texto depois de `REJECT` é uma string opcional enviada ao cliente junto com o código de erro. Ela informa o remetente por que o correio foi rejeitado.

Autenticação de clientes

Para usuários que enviam e-mails de uma instalação doméstica, normalmente é mais fácil rotear o e-mail enviado por meio do servidor de e-mails ISP doméstico, independentemente do endereço do remetente que aparece nesse e-mail. A maioria dos ISPs confia nos seus clientes diretos e permite a retransmissão. Se essa configuração não for possível ou se você estiver utilizando um sistema como o Sender ID ou o SPF, assegure que usuários móveis fora da sua rede tenham autorização de enviar mensagens para o seu **smtpd**.

Uma maneira de resolver esse problema é fornecer uma brecha de segurança (*piggyback*) sobre os protocolos de autenticação utilizados pelo POP ou IMAP. Usuários que precisam enviar e-mails também precisam lê-los, portanto a autenticação de um único passo pode ser utilizada para permitir os dois processos.

O sistema de "brecha de segurança" (*piggybacking*), chamando POP antes de SMTP ou IMAP antes de SMTP, funciona desta maneira: assim que um usuário foi autenticado pelo daemon POP ou IMAP, o endereço IP do usuário é colocado explicitamente em uma lista branca para o SMTP por aproximadamente 30 minutos. O banco de dados dos clientes em uma lista branca é mantido por um daemon dedicado a essa tarefa. A configuração no Postfix se parece com isto:

```
smtpd_recipient_restrictions = permit_mynetworks
          check_client_access hash:/etc/postfix/pop-before-smtp
          reject_unauth_destination
```

Uma solução mais interessante para esse problema é utilizar o mecanismo SMTP AUTH para autenticar diretamente no nível do SMTP. O Postfix deve ser compilado com suporte à biblioteca SASL para que isso funcione. Você pode então configurar o recurso assim:

```
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions = reject_non_fqdn_recipient
          permit_mynetworks
          permit_sasl_authenticated
```

Também é necessário suportar conexões criptografadas para impedir o envio de senhas em texto. Com as versões do Postfix anteriores a 2.2, corrija a fonte com o patch Postfix/TLS. Adicione linhas como estas a **main.cf**:

```
smtpd_use_tls = yes
smtpd_tls_auth_only = yes
smtpd_tls_cert_file = /etc/certs/smtp.pem
smtpd_tls_key_file = $smtpd_tls_cert_file
```

Combatendo spam e vírus

O Postfix tem muitos recursos que podem ajudar a bloquear e-mail suspeito.

Uma classe dos recursos de proteção exige uma implementação estrita do protocolo SMTP. Servidores de e-mails legítimos devem respeitar esse protocolo, mas remetentes de spam e vírus costumam ser precipitados e, assim, se expõem. Infelizmente, agentes de entrega local quebrados que tratam de correio legítimo continuam por aí, portanto, essa técnica não é à prova de falhas. Escolha restrições atentamente e monitore os arquivos de log. A Tabela 18.27 mostra alguns recursos nessa categoria.

Tabela 18.27 Parâmetros e restrições para verificação estrita do protocolo SMTP

Opção	Propósito

`reject_non_fqdn_sender`
`reject_non_fqdn_recipient`
`reject_non_fqdn_hostname`
	Rejeita mensagens sem um domínio de remetente completamente qualificado, sem um domínio de destinatário completamente qualificado ou nome de host HELO/EHLO (restrição)

`reject_unauth_pipelining`	Aborta a sessão atual se o cliente não espera para ver o status de um comando antes de prosseguir (restrição)
`reject_unknown_sender_domain`	Rejeita mensagens que têm um domínio de remetente que não pode ser resolvido (restrição)[a]
`smtpd_helo_required`	Requer HELO/EHLO no início da conversa (parâmetro, tanto `yes` como `no`)
`strict_rfc821_envelopes`	Requer sintaxe correta para endereços de e-mail nos comandos MAIL FROM e RCPT TO (parâmetro, tanto `yes` como `no`)

a. Retorna uma mensagem de erro temporário porque o problema talvez seja resultante de um problema transitório de DNS.

Para testar uma restrição antes de impô-la à produção (sempre uma boa idéia), insira a restrição `warn_if_reject` em frente dele para converter o efeito da rejeição total em mensagens de log de alerta.

Listas negras

Como ocorre com o **sendmail**, você pode instruir o Postfix a comparar os e-mails que chegam em uma lista negra baseada no DNS; ver a seção *Lista negra de usuários ou sites*, na página 411 para mais detalhes. Para ativar esse comportamento, utilize a restrição `reject_rbl_client` seguida pelo endereço do servidor DNS a ser consultado. Um recurso semelhante é `reject_rhsbl_sender`, que verifica o nome de domínio do endereço do remetente em vez do nome de host do cliente.

O exemplo a seguir representa uma configuração relativamente completa do combate a spam a partir do arquivo **main.cf**:

```
strict_rfc821_envelopes = yes
smtpd_helo_required = yes
smtpd_recipient_restrictions =   reject_unknown_sender_domain
       reject_non_fqdn_sender
       reject_non_fqdn_recipient
```

```
        permit_mynetworks
        check_client_access hash:/etc/postfix/client_access
        reject_unauth_destination
        reject_unauth_pipelining
        reject_rbl_client relays.ordb.org
        reject_rhsbl_sender dsn.rfc-ignorant.org
```

Observe que impomos algumas restrições em frente de `permit_mynetworks`. Esse ajuste permite verificar se nossos clientes estão enviando e-mail corretamente formatado. Essa é uma maneira fácil de descobrir erros de configuração.

SpamAssassin e procmail

O Postfix suporta o SpamAssassin e outros filtros desse tipo. Consulte as seções *SpamAssassin*, na página 414, e *O recurso local_procmail*, na página 405, para informações gerais sobre essas ferramentas.

procmail pode ser iniciado a partir dos arquivos **.forward** dos usuários, mas isso é complicado e propenso a erros. Uma solução melhor é colocar esta linha em **main.cf**:

```
mailbox_command = /usr/bin/procmail -a "$EXTENSION"
```

O Postfix então utiliza o **procmail** para entregar e-mail em vez de gravar as mensagens diretamente no spool de correio. Os argumentos fornecidos a **procmail** passam a extensão de endereço (a parte depois do +); ele pode então ser acessado no **procmail** como `$1`.

Daemons de diretiva

A versão 2.1 do Postfix introduziu um mecanismo para delegar controle de acesso a programas externos. Esses programas, chamados daemons de diretiva, recebem todas as informações que o Postfix tem sobre uma mensagem de e-mail e devem retornar uma das ações de transmissão listadas na Tabela 18.26 na página 438.

Talvez o recurso mais interessante que pode ser implementado com um daemon de diretiva sejam as "listas cinzas". Listas cinzas classificam cada mensagem recebida pela trinca nome de host de cliente, endereço de remetente e endereço de destinatário. Na primeira vez em que uma dada trinca é vista, o Postfix retorna uma mensagem de erro temporário ao remetente. Servidores de e-mails legítimos tentam uma nova entrega depois de aproximadamente 10 minutos, tempo em que é permitida a entrada da mensagem. Como a primeira tentativa de uma nova entrega normalmente ocorre em uma questão de minutos, o e-mail não é protelado indevidamente.

Essencialmente, uma lista cinza é semelhante aos recursos do Postfix que exigem conformidade estrita com o protocolo SMTP. No caso de uma lista cinza, a nova tentativa de entrega em si constitui uma evidência de servidor de e-mail legítimo. Listas cinzas provaram ser bem efetivas em eliminar spam porque muitas fontes de spam utilizam softwares não-sofisticados que não tentam novas entregas.

Filtragem de conteúdo

O Postfix pode utilizar expressões regulares para verificar nos cabeçalhos e nos corpos mensagens de e-mail ilegais. Ele também pode passar mensagens para outros programas como ferramentas dedicadas ao combate de spam ou a aplicativos de antivírus.

As verificações de cabeçalho e de corpo são realizadas em tempo real à medida que as mensagens são aceitas pelo SMTP. Cada expressão regular que é verificada invoca uma ação como especificado na Tabela 18.26 na página 438 se o regex corresponder. Por exemplo, a linha

```
header_checks = regexp:/etc/postfix/header_checks
```

no **main.cf** junto com esta linha em **/etc/postfix/header_checks**

```
/^Subject: reject-me/ REJECT You asked for it
```

rejeitaria todas as mensagens cujo assunto inicia com "reject-me". Embora o suporte de expressão regular seja sempre interessante, ele cria muitas armadilhas no contexto do processamento de e-mails. Em particular, esse não é um método efetivo de filtragem de spam ou de vírus.

Filtragem de vírus com capacidade industrial é normalmente implementada por meio do Amavis, um programa Perl que faz interface com um software de servidor de e-mails com um ou mais aplicativos de antivírus. Esses filtros são configurados com parâmetro `content_filter` do Postfix, que instrui o Postfix a passar cada mensagem recebida uma vez pelo serviço especificado. Além de configurar o parâmetro `content_filter`, você deve modificar algumas entradas existentes no arquivo **master.cf** e adicionar algumas novas. O Amavis vem com instruções detalhadas sobre como fazer isso. Há muitas variantes do Amavis disponíveis: recomendamos **amavisd-new** de Mark Martinec.

Depurando

Se tiver um problema com o Postfix, primeiro verifique os arquivos de log. É mais provável que as respostas a suas perguntas estejam aí; é apenas uma questão de localizá-las. Cada programa Postfix normalmente emite uma entrada de log para cada mensagem que ele processa. Por exemplo, a trilha de uma mensagem enviada talvez se pareça com isto:

```
Aug 18 22:41:33 nova postfix/pickup: 0E4A93688: uid=506 from=<dws@ee.ethz.ch>
Aug 18 22:41:33 nova postfix/cleanup: 0E4A93688: message-id= <20040818204132.GA11444@ee.ethz.ch>
Aug 18 22:41:33 nova postfix/qmgr: 0E4A93688: from=<dws@ee.ethz.ch>, size=577,nrcpt=1 (queue
    active)
Aug 18 22:41:33 nova postfix/smtp: 0E4A93688:
    to=<evi@ee.ethz.ch>,relay=tardis.ee.ethz.ch[129.132.2.217],delay=0, status=sent (250 Ok: queued
    as 154040930B)
Aug 18 22:41:33 nova postfix/qmgr: 0E4A93688: removed
```

Como você pode ver, as informações interessantes estão distribuídas por várias linhas. Observe que o identificador 0E4A93688 é comum a cada linha: o Postfix atribui um ID de fila assim que uma mensagem entra no sistema de correio e nunca a altera. Portanto, ao pesquisar nos logs o histórico de uma mensagem, primeiro se concentre em determinar o ID de fila da mensagem. Depois que souber isso, é fácil utilizar **grep** nos logs de todas as entradas relevantes.

O Postfix é bom para registrar em log mensagens úteis sobre problemas que ele percebe. Entretanto, às vezes é difícil identificar as linhas importantes entre os milhares de mensagens com status normal. Esse é um bom lugar para considerar o uso de algumas ferramentas discutidas na seção *Condensando arquivos de log em informações úteis*, a partir da página 157.

Examinando a fila

Um outro lugar para procurar problemas é a fila de correio. Como no sistema **sendmail**, um comando **mailq** imprime o conteúdo de uma fila. Você pode utilizá-lo para ver se e por que uma mensagem ficou presa.

Uma outra ferramenta útil é o script **qshape** que é distribuído com versões Postfix recentes. Ela mostra estatística de resumo sobre o conteúdo de uma fila. A saída ficará parecida com esta:

```
# qshape deferred
                T    5   10   20   40   80  160  320  640 1280 1280+
            TOTAL   78    0    0    0    7    3    3    2   12    2   49
          expn.com   34    0    0    0    0    0    0    0    9    0   25
      chinabank.ph    5    0    0    0    1    1    1    2    0    0    0
   prob-helper.biz    3    0    0    0    0    0    0    0    0    0    3
```

qshape resume a dada fila (aqui, a fila diferida) classificada pelo domínio do destinatário. As colunas informam por quantos minutos as mensagens relevantes estiveram na fila. Por exemplo, você pode ver que 49 mensagens associadas a expn.com ficaram na fila por mais de 1280 minutos. Todos os destinos neste exemplo sugerem que as mensagens foram enviadas de scripts de férias em resposta ao spam.

qshape também pode resumir por domínio de remetente com o flag **-s**.

A configuração de `soft-bounce`

Se `soft_bounce` estiver configurado como `yes`, o Postfix enviará mensagens de erro temporário em vez de enviar mensagens de erro permanente como "user unknown" ou "relaying denied". Esse é um excelente recurso de teste; ele permite monitorar a disposição das mensagens depois de uma alteração na configuração sem o risco de perder permanentemente e-mails legítimos. Qualquer coisa que você rejeita acaba retornando para uma outra tentativa. Mas não se esqueça de desativar esse recurso depois de terminar o teste. Do contrário, você terá de lidar repetidamente com cada mensagem rejeitada.

Testando controle de acesso

A maneira mais fácil de testar restrições de controle de acesso é tentar enviar uma mensagem a partir de um host externo e ver o que acontece. Esse é um bom teste básico, mas não abrange condições especiais como correio a partir de um domínio específico em que não há nenhum login.

O Postfix 2.1 introduziu uma extensão ao protocolo SMTP chamada XCLIENT que simula envios a partir de outros lugares. Esse recurso permanece desativado por padrão, mas, com esta linha de configuração em **main.cf**, você pode ativá-lo para conexões que se originam de host local:

```
smtpd_authorized_xclient_hosts = localhost
```

Uma sessão de teste poderia ser semelhante a esta:

```
$ telnet localhost 25
```

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tardis.ee.ethz.ch ESMTP Postfix
XCLIENT NAME=mail.cs.colorado.edu ADDR=192.168.1.1
250 Ok
HELO mail.cs.colorado.edu
250 tardis.ee.ethz.ch
MAIL FROM: <evi@colorado.edu>
250 Ok
RCPT TO: <david@colorado.edu>
554 <david@colorado.edu>: Relay access denied
```

18.17 LEITURA RECOMENDADA

COSTALES, BRYAN E ALLMAN, ERIC. *sendmail. 3. ed.* Sebastopol, CA: O'Reilly Media, 2002.

Esse livro é o tomo definitivo — 1.200 páginas que valem a pena. Ele inclui um tutorial, bem como uma seção para referência completa. O livro é bem legível em um estilo "abra uma página qualquer", o que consideramos um fator importante para um livro de referência. Ele também possui um excelente índice.

RICHARD, CLAYTON. "Good Practice for Combating Unsolicited Bulk Email."

RIPE/Demon Internet. Disponível em: <www.ripe.net/ripe/docs/ripe-206.html>.

Esse documento visa aos ISPs. Ele possui uma grande quantidade de informações sobre políticas e alguns bons links para assuntos técnicos.

SCHWARTZ, ALAN. *SpamAssassin*. Sebastopol, CA: O'Reilly Media, 2005.

SCHWARTZ, ALAN E FERGUSON, PAULA. *Managing Mailing Lists*. Sebastopol, CA: O'Reilly Media, 1998.

HAZEL, PHILIP. *The Exim Smtp Mail Server: Official Guide for Release 4*. Cambridge, UK: User Interface Technologies, Ltd., 2003.

Documentação e informações sobre o Exim podem ser encontradas em www.exim.org.

A página man do **sendmail** descreve seus argumentos de linha de comando. Consulte *Sendmail: An Internetwork Mail Router*, de Eric Allman, para uma visão geral.

As instruções de instalação e uma boa descrição do arquivo de configuração são abrangidas em *Sendmail Installation and Operation Guide*, que pode ser encontrado no subdiretório **doc/op** da distribuição **sendmail**. Esse documento é bem completo e, em conjunto com o arquiovo **README** no diretório **cf**, lhe dá uma boa visão sobre os componentes e aspectos básicos do **sendmail**.

www.sendmail.org, www.sendmail.org/~ca e www.sendmail.org/~gshapiro contêm documentos, HOWTOs e tutoriais relacionados com o **sendmail**.

A RFC2822, que substitui a RFC822, descreve a sintaxe das mensagens e endereços em um sistema de e-mails em rede e a RFC1123 descreve os requisitos do host. Essas são, em certo sentido, as especificações funcionais sobe as quais o **sendmail** foi construído.

A RFC2821, que substitui a RFC821, define o SMTP (Simple Mail Transport Protocol) e as RFCs 1869, 1870, 1891 e 1985 o estendem para o ESMTP.

A RFC974 descreve registros MX no DNS e seus relacionamentos com o direcionamento de e-mails. Outras RFCs relacionadas com e-mail são:

- RFC1731 – IMAP4 Authentication Mechanisms (Mecanismos de autenticação IMAP4)
- RFC1733 – Distributed Electronic Mail Models in IMAP4
- RFC2033 – Local Mail Transfer Protocol
- RFC2076 – Common Internet Message Headers
- RFC2142 – Mailbox Names for Common Services, Roles and Functions
- RFC2505 – Anti-Spam Recommendations for SMTP MTAs

- RFC2635 – DON'T SPEW: Diretrizes contra spam em massa[35]
- RFC2821 – Simple Mail Transfer Protocol
- RFC2822 – Internet Message Format
- RFC4405 – SMTP Service Extension for Indicating Message Submitters[35]
- RFC4406 – Sender ID: Authenticating E-Mail
- RFC4408 – SPF for Authorizing Use of Domains in E-Mail, Version 1
- RFC4409 – Message Submission for Mail

As RFCs 2821 (SMTP) e 2822 (Internet Message Format) organizam algumas das RFCs sobre e-mail mais comumente referidas na Internet; elas substituem as RFCs 821, 822, 974 e 1869. As RFCs 2821 e 2822 foram publicadas pela primeira vez em abril de 2001 e são padrões propostos.

18.18 Exercícios

E18.1 [Específico de **sendmail**] Liste brevemente as diferenças e semelhanças entre `genericstable` e `virtusertable`. Em quais situações você utilizaria cada uma deles?

E18.2 [Específico de **sendmail**] Compare o emprego de **/etc/mail/aliases** com o de um servidor LDAP para armazenar aliases de e-mails. Quais são as vantagens e as desvantagens de cada um deles?

E18.3 Explique brevemente a diferença entre um agente de usuário de e-mail (Mail User Agent – MUA), um agente de entrega (Delivery Agent – DA) e um agente de acesso (Access Agent – AA). A seguir, explique a diferença entre um agente de transporte de e-mails (Mail Transport Agent – MTA) e um agente de envio de e-mails (Mail Submission Agent – MSA).

E18.4 [Específico de **sendmail**] O que é **smrsh** e por que você deve usá-lo em vez de **/bin/sh**? Se **smrsh** estiver sendo utilizado em seu site, que programas têm permissão para serem executados como o programa agente de entrega? Alguns deles são perigosamente inseguros?

E18.5 [Específico de **sendmail**] Crie um pequeno arquivo **/etc/mail/aliases** que demonstre três tipos diferentes de aliases. Fale brevemente sobre o que faz cada uma das linhas e por que poderia ser útil.

★E18.6 Descreva brevemente o seguinte cabeçalho de e-mail. Que caminho o e-mail percorreu? Para quem ele estava endereçado e para quem ele foi entregue? Quanto tempo levou para o e-mail ir do remetente ao destinatário?

```
From clements@boulderlabs.com Fri Dec 28 17:06:57 2001
Return-Path: <clements@mail.boulderlabs.com>
Received: from boulder.Colorado.EDU (boulder.Colorado.EDU
    [128.138.240.1]) by ucsub.colorado.edu (8.11.6/8.11.2/ITS-5.0/student)
    with ESMTP idfBT06vF10618 for <hallcp@ucsub.Colorado.EDU>; Fri,
    28 Dec 2001 17:06:57-0700 (MST)
Received: from mail.boulderlabs.com (mail.boulderlabs.com
    [206.168.112.48]) by boulder.Colorado.EDU
    (8.10.1/8.10.1/UnixOps+Hesiod (Boulder)) with ESMTP id
    fBT06uL13184; Fri, 28 Dec 2001 17:06:56 -0700 (MST)
Received: from ath.boulderlabs.com (cpe-24-221-212-162.co.sprintbbd.net
    [24.221.212.162]) by mail.boulderlabs.com (8.11.6/8.11.6) with ESMTP
    id fBT06oQ29214 for <booklist@boulderlabs.com>; Fri, 28 Dec 2001
    17:06:50 -0700 (MST) (envelope-from
    clements@mail.boulderlabs.com)
From: David Clements <clements@boulderlabs.com>
Received: (from clements@localhost) by ath.boulderlabs.com
    (8.11.6/8.11.4) id fBT06ma01470 for booklist@boulderlabs.com; Fri,
    28 Dec 2001 17:06:48 -0700 (MST) (envelope-from clements)
Date: Fri, 28 Dec 2001 17:06:48 -0700 (MST)
```

35. Título parafraseado.

```
Message-Id: <200112290006.fBT06ma01470@ath.boulderlabs.com>
To: boolist@boulderlabs.com
Subject: Book Questions
```

★ E18.7 [Específico de **sendmail**] Liste os prefixos para os arquivos no diretório de fila de correio e explique o que cada um significa. Por que é importante eliminar alguns deles, mas muito incorreto eliminar outros? Como alguns dos prefixos podem ser utilizados para depurar erros na configuração de **sendmail**?

E18.8 Observe **mailq** em seu servidor de e-mails. Há algum problema no diretório? Há alguma mensagem sem arquivos de controle ou arquivos de controle sem mensagens? Qual é a mensagem mais antiga da fila? (Exige acesso root.)

★ E18.9 [Específico de **sendmail**] Explique o propósito de cada uma das macros **m4** apresentadas a seguir. Se a macro inclui um arquivo, forneça uma breve descrição de qual deveria ser o conteúdo do arquivo.

a) `VERSIONID`

b) `OSTYPE`

c) `DOMAIN`

d) `MAILER`

e) `FEATURE`

★ E18.10 Explique o que é um registro MX. Por que os registros MX são importantes para entrega de e-mails? Dê um exemplo no qual um registro MX malconfigurado poderia impossibilitar a entrega do e-mail.

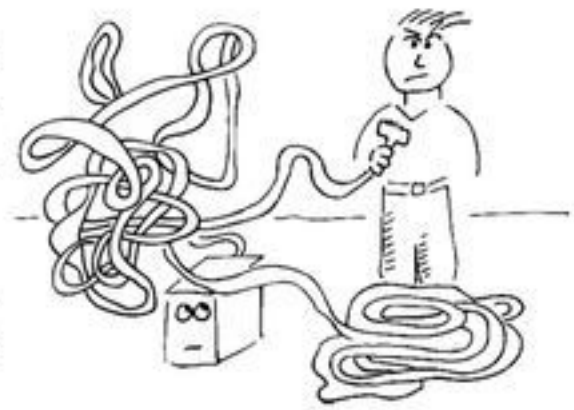
★ E18.11 Quais são as implicações de ser colocado em uma lista negra em sbl-xbl.spamhaus.org ou em uma lista negra de spam semelhante? Descreva algumas técnicas utilizadas para permanecer fora dessas listas.

★ E18.12 Se seu site permitir **procmail** e se você tiver permissão a partir do seu grupo local de administradores de sistemas, configure o arquivo de configuração **procmail** dos seus funcionários a fim de demonstrar como o **procmail** pode comprometer a segurança.

★ ★ E18.13 Explore a configuração MTA atual em seu site. Quais são alguns dos recursos especiais do MTA em utilização? Você é capaz de encontrar algum problema na configuração? De que modos a configuração poderia ser melhorada?

★ ★ E18.14 Encontre um exemplo de spam em sua caixa postal e inspecione os cabeçalhos. Relate qualquer sinal de que a mensagem foi forjada. A seguir execute algumas das ferramentas mencionadas neste capítulo, como SpamCop ou SpamAssassin, e relate suas descobertas. Como você faz para reconhecer cabeçalhos falsificados? Envie o spam e suas conclusões sobre o remetente, a validade dos hosts listados e qualquer outra coisa que pareça inadequada.

19 Gerenciamento e depuração de redes



Como as redes aumentam o número de dependências entre as máquinas, elas tendem a aumentar os problemas. Como diria o ditado, "Rede é quando você não consegue ter nenhum trabalho terminado devido à falha de uma máquina da qual você jamais ouviu falar".

O gerenciamento de redes é a arte e a ciência de manter uma rede saudável. Geralmente abrange as seguintes tarefas:

- Detecção de falhas de redes, gateways e servidores críticos
- Esquemas para notificar a um administrador a existência de problemas
- Monitoramento geral, para balanceamento de carga e planejamento de expansões
- Documentação e visualização da rede
- Administração de dispositivos de rede a partir de uma instalação central

Em um único segmento de rede, em geral não vale a pena estabelecer procedimentos formais para gerenciamento da rede. Apenas teste completamente a rede após a instalação e verifique-a de vez em quando para ter certeza de que a carga não é excessiva. Quando a rede apresentar algum problema, corrija-o.

À medida que a rede for crescendo, os procedimentos de gerenciamento devem ser automatizados. Em uma rede formada por várias sub-redes diferentes com switches (comutadores) ou roteadores, talvez você deva começar automatizando tarefas de gerenciamento com scripts de shell e programas simples. Se você tiver uma rede remota ou uma rede local complexa, considere a instalação de uma estação de gerenciamento de rede dedicada.

Em alguns casos, as necessidades de confiabilidade da organização determinarão a sofisticação do sistema de gerenciamento de redes. Um problema na rede pode fazer com que todo o trabalho seja paralisado. Caso sua instalação não possa tolerar "interrupções", vale a pena obter e instalar um sistema de gerenciamento de redes corporativas de alta capacidade.

Infelizmente, até mesmo o melhor sistema para gerenciamento de redes não é capaz de impedir todas as falhas. É crítico ter uma rede bem documentada e um pessoal altamente qualificado disponível para lidar com os colapsos inevitáveis.

19.1 Solucionando problemas de rede

Há várias ferramentas de qualidade para depurar uma rede na camada TCP/IP. A maioria delas gera informações de baixo nível, de modo que é necessário compreender as principais idéias do TCP/IP e roteamento para poder utilizar essas ferramentas de depuração.

Por outro lado, problemas de rede também podem surgir de questões referentes a protocolos de nível mais alto como DNS, NFS e HTTP. Talvez você queira ler os capítulos 12, *Redes TCP/IP*, e 13, *Roteamento*, antes de começar este.

Nesta seção, começamos com alguma estratégia genérica de diagnóstico. A seguir vemos várias ferramentas essenciais, inclusive **ping**, **traceroute**, **netstat**, **tcpdump** e Wireshark. Não discutiremos o comando **arp** neste capítulo, embora ele também seja uma ferramenta útil para depuração — consulte a página 207 para informações adicionais.

Antes de tentar resolver o problema, considere os seguintes princípios:

- Faça uma alteração de cada vez e teste cada uma delas para ter certeza de que elas têm o efeito desejado. Desfaça quaisquer alterações que produzam um efeito indesejado.
- Documente a situação em que ela se encontrava antes de você se envolver e documente cada alteração que você faça no meio do caminho.
- Inicie em uma "ponta" do sistema ou rede e vá trabalhando nos componentes críticos até atingir o problema. Por exemplo, você poderia começar analisando a configuração de rede em um cliente, percorrer as conexões físicas, investigar o hardware da rede e, por fim, verificar as conexões físicas do servidor e configurações de software.
- Comunique-se regularmente. A maioria dos problemas em uma rede envolve ou afeta um grande número de pessoas diferentes: usuários, ISPs, administradores de sistemas, engenheiros de telecomunicações, administradores de redes etc. Uma comunicação clara e consistente impede que você acabe obstruindo os esforços de outros em solucionar o problema.

- Trabalhe em equipe. Anos de experiência demonstram que as pessoas cometem menos erros estúpidos se elas tiverem um companheiro ajudando.
- Utilize as camadas de rede para negociar o problema. Comece pelo "alto" ou por "baixo" e vá percorrendo a pilha de protocolos.

Esse último ponto merece ser discutido um pouco mais. Conforme descrito na página 193, a arquitetura do TCP/IP define várias camadas de abstração nas quais os componentes da rede podem funcionar. Por exemplo, o HTTP depende do TCP, o TCP depende do IP, o IP depende do protocolo Ethernet e o protocolo Ethernet depende da integridade do cabo de rede. Você pode reduzir significativamente a quantidade de tempo gasto depurando um problema se descobrir primeiro qual camada está se comportando mal.

Faça as seguintes perguntas a si próprio à medida que você sobe (ou desce) pela pilha:

- Você tem conexão física e uma indicação de que o link funciona (como a luz do led)?
- Sua interface está configurada de maneira apropriada?
- Suas tabelas ARP mostram outros hosts?
- É possível fazer um ping para o endereço de localhost (127.0.0.1)?
- É possível fazer um ping para outros hosts locais pelo endereço IP?
- O DNS está funcionando adequadamente?[1]
- É possível fazer um ping para outros hosts locais pelo nome de host?
- É possível fazer um ping para outros hosts em uma outra rede?
- Serviços de alto nível, como servidores Web e servidores SSH, funcionam?

Assim que tiver identificado onde se encontra o problema, retorne um passo e considere o efeito que testes subseqüentes e possíveis correções terão sobre outros serviços e hosts.

19.2 PING: VERIFICANDO SE UM HOST ESTÁ FUNCIONANDO

O comando **ping** é embaraçosamente simples, mas em muitas situações é tudo de que você precisa. Ele envia um pacote ICMP ECHO_REQUEST a um host de destino e espera para ver se o host responde de volta. Apesar de sua simplicidade, o comando **ping** é um dos burros de carga na depuração de redes.

Podemos utilizar **ping** para verificar o estado de hosts individuais e testar segmentos da rede. As tabelas de rede, redes físicas e gateways estão envolvidos no processamento de um ping, de modo que a rede tem de estar mais ou menos funcionando para um comando **ping** ser bem-sucedido. Se **ping** não funcionar, pode estar quase certo de que nada mais sofisticado funcionará. Entretanto, essa regra não se aplica a redes que bloqueiam solicitações de eco ICMP por meio de um firewall. Certifique-se de que um firewall não esteja interferindo em sua depuração antes de chegar à conclusão de que o host de destino está ignorando um **ping**. Talvez você possa considerar a possibilidade de desabilitar o firewall por um curto espaço de tempo para facilitar a depuração.

ping executa em um loop infinito a menos que você forneça um argumento de contagem de pacotes. Assim que tiver satisfeito com o resultado de ping, digite o caractere de interrupção (normalmente <Control-C>) para sair.

Eis um exemplo:

```
$ ping beast
PING beast (10.1.1.46): 56 bytes of data.
64 bytes from beast (10.1.1.46): icmp_seq=0 ttl=54 time=48.3ms
64 bytes from beast (10.1.1.46): icmp_seq=1 ttl=54 time=46.4ms
64 bytes from beast (10.1.1.46): icmp_seq=2 ttl=54 time=88.7ms
^C
--- beast ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 46.490/61.202/88.731/19.481 ms
```

A saída para beast mostra o endereço IP do host, o número de seqüência ICMP de cada pacote de resposta e o tempo de viagem de ida e volta. A coisa mais óbvia que essa saída nos diz é que o servidor beast está no ar e conectado à rede.

1. Se sua máquina travar no momento da inicialização, inicializar muito lentamente ou ficar travada em conexões **telnet** de entrada, o DNS deve ser seu principal suspeito.

Em uma rede sem problemas, **ping** pode permitir que você determine se um host está inativo ou não. Por outro lado, quando um host remoto sabidamente está ativo e funcionando, **ping** pode lhe dar informações úteis sobre a saúde da rede. Os pacotes **ping** são direcionados por mecanismos IP usuais e uma viagem de ida e volta bem-sucedida significa que todas as redes e gateways que se encontram entre a fonte e o destino estão funcionando corretamente, pelo menos em uma primeira abordagem.

O número de seqüência ICMP é um fragmento particularmente valioso de informações. Descontinuidades na seqüência indicam pacotes descartados; normalmente elas são acompanhadas por uma mensagem para cada pacote ausente. Apesar do fato de o IP não garantir a entrega dos pacotes, uma rede saudável deve perder muito poucos deles. É importante monitorar problemas de pacotes perdidos porque eles tendem a ser mascarados por protocolos de nível mais alto. A rede pode parecer funcionar corretamente, mas será mais lenta do que deveria ser, não somente devido aos pacotes retransmitidos, como também pelo custo adicional do protocolo necessário para detectá-los e gerenciá-los.

Para rastrear a causa do desaparecimento de pacotes, execute primeiramente **traceroute** (veja a próxima seção) para conhecer a rota que os pacotes estão tomando em direção ao host desejado. A seguir, efetue um comando **ping** nos gateways intermediários em seqüência para descobrir que link está perdendo pacotes. Para identificar o problema, você precisa enviar um número estatisticamente significativo de pacotes. A falha na rede geralmente reside no link entre o último gateway sobre o qual você consegue aplicar um **ping** sem perdas de pacotes significativas e o gateway acima dele.

O tempo da viagem de ida e volta informado por **ping** lhe dá uma visão do desempenho global de uma rota por meio de uma rede. Pequenas variações no tempo de viagem de ida e volta normalmente não indicam problemas. Os pacotes podem ocasionalmente atrasar em cerca de décimos ou centésimos de milissegundos sem motivo aparente; essa é simplesmente a forma como o IP funciona. Você deve esperar observar um tempo de viagem de ida e volta razoavelmente consistente para a maioria dos pacotes, com lapsos ocasionais. Muitos dos roteadores atuais implementam respostas de velocidade limitada ou de baixa prioridade a pacotes ICMP, o que significa que um roteador pode atrasar para responder ao seu **ping** se já estiver lidando com muito tráfego ICMP.

O programa **ping** pode enviar pacotes de solicitação de eco de qualquer tamanho, portanto o uso de um pacote maior que o MTU da rede (1.500 bytes para Ethernet) pode forçar uma fragmentação. Essa prática ajuda a identificar erros de mídia ou outros problemas de baixo nível como aqueles de uma rede ou VPN congestionada. Para especificar em bytes o tamanho desejado de pacote, utilize o flag -s.

```
$ ping -s 1500 cuinfo.cornell.edu
```

Utilize o comando **ping** tendo em mente as seguintes limitações.

Primeiramente, é difícil distinguir uma falha de rede de uma falha de um servidor somente com o comando **ping**. Em um ambiente em que testes de ping normalmente funcionam, um ping falho só informa que *algo* está errado. (Firewalls de rede às vezes bloqueiam intencionalmente pacotes ICMP.)

Em segundo lugar, um **ping** bem-sucedido não garante muito em relação ao estado da máquina de destino. Os pacotes de solicitação de eco são manipulados dentro do protocolo IP e não requerem que um processo de servidor esteja executando no host que está sendo investigado. Uma resposta garante somente que uma máquina está ligada e não sofreu um pânico do kernel (isto é, não está travada). Você precisará de métodos de nível mais alto para verificar a disponibilidade de serviços como o HTTP e o DNS.

19.3 TRACEROUTE: RASTREANDO PACOTES IP

traceroute, originalmente escrito por Van Jacobson, revela a seqüência de gateways que um pacote IP percorre para alcançar seu destino. Todos os sistemas operacionais modernos vêm com alguma versão do **traceroute**. A sintaxe é simplesmente:

```
traceroute nomeDoHost
```

Há uma série de opções, e a maioria delas não é importante no dia-a-dia. Como de costume, o *nomeDoHost* pode ser especificado com um nome DNS ou com um endereço IP. A saída é simplesmente uma lista de hosts, iniciando com o primeiro gateway e terminando no destino.

Por exemplo, um **traceroute** proveniente do host jaguar para o host nubark gera a seguinte saída:

```
$ traceroute nubark
traceroute to nubark (192.168.2.10), 30 hops max, 38 byte packets
 1  lab-gw (172.16.8.254)  0.840 ms  0.693 ms  0.671 ms
 2  dmz-gw (192.168.1.254)  4.642 ms  4.582 ms  4.674 ms
 3  nubark (192.168.2.10)  7.959 ms  5.949 ms  5.908 ms
```

Com base nessa saída podemos dizer que jaguar se encontra exatamente a três saltos de distância de nubark e podemos ver que gateways estão envolvidos na conexão. O tempo de viagem de ida e volta para cada gateway também é mostrado — são medidas e exibidas três amostras para cada salto. Um **traceroute** típico entre hosts de Internet freqüentemente inclui mais de 15 hops.

traceroute funciona configurando-se o campo TTL (Time To Live, ou tempo de sobrevivência) de um pacote de saída com um número artificialmente baixo. À medida que pacotes chegam a um gateway, seu TTL é diminuído. Quando um gateway diminui o valor TTL até 0, ele descarta o pacote e envia uma mensagem ICMP "time exceeded" (tempo excedido) para o host de origem do comando.

Consulte a página 275 para informações adicionais sobre pesquisas de DNS inversas.

Os primeiros três pacotes **traceroute** têm seu TTL configurado como 1. O primeiro gateway a ver um pacote desses (lab-gw neste caso) determina que o TTL foi excedido e notifica jaguar sobre o pacote descartado enviando uma mensagem ICMP de volta. O endereço IP no cabeçalho do pacote de erro identifica o gateway; **traceroute** pesquisa esse endereço no DNS para descobrir o nome de host do gateway.

Para identificar o gateway do segundo salto, **traceroute** envia uma segunda rodada de pacotes com campos TTL configurados como 2. O primeiro gateway direciona os pacotes e diminui o TTL por uma unidade. No segundo gateway, os pacotes são então descartados e as mensagens de erro de ICMP são geradas como antes. Esse processo continua até que o TTL seja igual ao número de saltos para o host de destino e os pacotes atinjam seus destinos.

A maioria dos roteadores envia suas mensagens ICMP por meio da interface "mais próxima" ao seu host. Se você executar **traceroute** percorrendo o caminho contrário a partir do host de destino, provavelmente verá endereços IP diferentes sendo utilizados para identificar o mesmo conjunto de roteadores. Talvez você também veja caminhos completamente diferentes; essa configuração é conhecida como "roteamento assimétrico".

Como **traceroute** envia três pacotes para cada valor do campo TTL, eventualmente você pode observar uma documentação interessante. Se um gateway interventor multiplexar o tráfego entre várias rotas, os pacotes poderão ser retornados por hosts diferentes; nesse caso, **traceroute** simplesmente imprimirá todos eles.

Vejamos um exemplo mais interessante de um host em colorado.edu para xor.com:

```
rupertsberg$ traceroute xor.com
traceroute to xor.com (192.225.33.1), 30 hops max, 38 byte packets
 1  cs-gw3-faculty.cs.colorado.edu (128.138.236.3)  1.362 ms  2.144 ms  2.76 ms
 2  cs-gw-dmz.cs.colorado.edu (128.138.243.193)  2.720 ms  4.378 ms  5.052 ms
 3  engr-cs.Colorado.EDU (128.138.80.141)  5.587 ms  2.454 ms  2.773 ms
 4  hut-engr.Colorado.EDU (128.138.80.201)  2.743 ms  5.643 ms  2.772 ms
 5  cuatm-gw.Colorado.EDU (128.138.80.2)  5.587 ms  2.784 ms  2.777 ms
 6  204.131.62.6 (204.131.62.6)  5.585 ms  3.464 ms  2.761 ms
 7  border-from-BRAN.coop.net (199.45.134.81)  5.593 ms  6.433 ms  5.521 ms
 8  core-gw-eth-2-5.coop.net (199.45.137.14)  53.806 ms  *  19.202 ms
 9  xor.com (192.225.33.1)  16.838 ms  15.972 ms  11.204 ms
```

Essa saída mostra que os pacotes têm de passar por cinco gateways internos antes de deixarem a rede colorado.edu (de cs-gw3-faculty a cuatm-gw). O gateway do próximo salto na rede BRAN (204.131.62.6) não tem um nome no DNS. Após dois saltos em coop.net, chegamos a xor.com.

No salto 8, vemos um asterisco no lugar de um dos tempos de viagem de ida e volta. Essa notação significa que nenhuma resposta (pacote de erro) foi recebida em resposta à sondagem. Nesse caso, a causa provavelmente é congestionamento, mas essa não é a única possibilidade. **traceroute** depende de pacotes ICMP de baixa prioridade, que muitos roteadores são inteligentes o bastante para descartar em preferência a tráfego "verdadeiro". Alguns poucos asteriscos não devem fazer você entrar em pânico.

Se você vir asteriscos em todos os campos de tempo de viagem de ida e volta para um dado gateway, nenhuma mensagem "time exceeded" (tempo excedido) está chegando daquela máquina. Talvez o gateway esteja simplesmente fora do ar. Às vezes, um gateway ou um firewall é configurado de modo a descartar silenciosamente pacotes com TTLs expirados. Nesse caso, você ainda pode ver o host silencioso por trás dos gateways. Outra possibilidade é que os pacotes de erros do gateway sejam lentos para retornarem e que **traceroute** tenha deixado de esperar por eles antes de eles chegarem.

Alguns firewalls bloqueiam totalmente mensagens de ICMP de "tempo esgotado". Se um desses firewalls estiver ao longo do caminho, você não receberá informações sobre nenhum dos gateways por trás dele. Mas você ainda pode determinar o número total de saltos até o destino, pois os pacotes de sondagem acabarão percorrendo todo o caminho. Da mesma forma, alguns firewalls podem bloquear os datagramas UDP vindos de fora que **traceroute** envia para disparar respostas ICMP. Esse problema faz com que **traceroute** não gere nenhuma informação útil.

Um link lento não indica necessariamente um mal funcionamento. Algumas redes físicas têm uma latência naturalmente alta; redes sem fio 802.11 são um bom exemplo. Lentidão também pode ser um sinal de congestionamento na rede receptora, especialmente se a rede utiliza uma tecnologia CSMA/CD que faz repetidas tentativas de transmitir um pacote (Ethernet é

um exemplo). Tempos de viagem de viagem de ida e volta inconsistentes suportariam tal hipótese, uma vez que as colisões aumentam a aleatoriedade do comportamento da rede.

Algumas vezes você poderá ver a notação `!N` em vez de um asterisco ou um tempo de viagem de ida e volta. Isso indica que o gateway atual retornou um erro "network unreachable" (rede inatingível), significando que ele não sabe como direcionar o seu pacote. Outras possibilidades incluem `!H` para "host unreachable" (host inatingível) e `!P` para "protocol unreachable" (protocolo inatingível). Um gateway que retorna uma dessas mensagens de erro normalmente é o último hop que você pode obter. Esse host normalmente apresenta um problema de roteamento (possivelmente provocado por um link interrompido): suas rotas estáticas estão erradas ou os protocolos dinâmicos falharam em propagar uma rota útil ao destino.

Se **traceroute** parecer não funcionar (ou operar de maneira notavelmente lenta), talvez ele esteja expirando seu tempo-limite ao tentar determinar os nomes dos gateways utilizando DNS. Se o DNS não estiver funcionando no host a partir do qual você está fazendo o rastreamento, utilize **traceroute -n** para solicitar saída numérica. Essa opção impede o uso de DNS; talvez seja a única maneira de fazer com que **traceroute** funcione em uma rede problemática.

19.4 NETSTAT: OBTENDO ESTATÍSTICAS DE REDE

netstat coleta uma grande quantidade de informações sobre o estado do software de rede de seu computador, inclusive estatísticas de interface, informações de roteamento e tabelas de conexão. De fato, não há um tema unificador para os diferentes conjuntos de saída, exceto que todos eles estão relacionados com a rede. Pense no **netstat** como o "tudo em um" das ferramentas de rede — ele expõe uma variedade de informações sobre uma rede as quais não se ajustam em outro lugar. Aqui, discutimos os cinco empregos mais comuns de **netstat**:

- Inspecionar as informações de configuração de interface
- Monitorar o estado das conexões de rede
- Identificar serviços de rede ouvintes
- Examinar a tabela de roteamento
- Visualizar estatísticas operacionais para vários protocolos de rede

Inspecionando as informações de configuração de interface

netstat -i exibe informações sobre a configuração e o estado de cada uma das interfaces de rede do host. Você pode executar o **netstat -i** como uma boa maneira de conhecer a configuração de rede de uma nova máquina. Adicione a opção **-e** para detalhes adicionais.

Por exemplo:

```
$ netstat -i -e
Kernel Interface table
eth0       Link encap:Ethernet  HWaddr 00:02:B3:19:C8:82
           inet addr:192.168.2.1 Bcast:192.168.2.255  Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:1121527 errors:0 dropped:0 overruns:0 frame:0
           TX packets:1138477 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:100
           Interrupt:7 Base address:0xef00

eth1       Link encap:Ethernet  HWaddr 00:02:B3:19:C6:86
           inet addr:192.168.1.13  Bcast:192.168.1.255  Mask:255.255.255.0
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:67543 errors:0 dropped:0 overruns:0 frame:0
           TX packets:69652 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:100
           Interrupt:5 Base address:0xed00

lo         Link encap:Local Loopback
           inet addr:127.0.0.1  Mask:255.0.0.0
           UP LOOPBACK RUNNING  MTU:3924  Metric:1
           RX packets:310572 errors:0 dropped:0 overruns:0 frame:0
           TX packets:310572 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:0
```

Esse host tem duas interfaces de rede: uma para tráfego normal e uma segunda conexão para gerenciamento de sistema chamado eth1. `RX packets` e `TX packets` informam o número de pacotes recebido e transmitido em cada interface a partir do momento em que a máquina foi inicializada. São contabilizados diversos tipos diferentes de erros e é normal aparecerem alguns.

Os erros devem ser inferiores a 1% dos pacotes associados. Se sua taxa de erros for alta, compare as taxas de várias máquinas vizinhas. Um grande número de erros em uma única máquina sugere problemas com a conexão ou com a interface dessa máquina. Uma alta taxa de erros em todos os lugares provavelmente indica um problema de mídia ou de rede. Uma das causas mais comuns de uma taxa de erros alta é uma não-correspondência da velocidade ou da opção duplex Ethernet causada por uma falha de autodetecção ou autonegociação.

Colisões sugerem uma rede carregada; erros normalmente indicam problemas de cabeamento. Embora uma colisão seja um tipo de erro, ela é contabilizada separadamente por **netstat**. O campo chamado `Collisions` informa o número de colisões ocorridas no envio dos pacotes.[2] Utilize esse número para calcular a porcentagem de pacotes de saída (`TX packets`) que resultam em colisões. Em uma rede funcionando direito, a taxa de colisões deve ser inferior a 3% do total de pacotes de saída, e algo um pouco acima de 10% indica sérios problemas de congestionamento. As colisões nunca devem ocorrer em um link full-duplex que está operando adequadamente.

Monitorando o estado das conexões de rede

Sem utilizar nenhum argumento, **netstat** exibe o estado das portas TCP e UDP ativas. Servidores inativos ("ouvintes") esperando conexões normalmente não são mostrados; podem ser vistos com **netstat -a**.[3] A saída se parece com isto:

```
$ netstat -a
Active Internet connections (servers and established)
Proto      Recv-Q Send-Q Local Address  ForeignAddress    State
tcp        0      0      *:ldap         *:*               LISTEN
tcp        0      0      *:mysql        *:*               LISTEN
tcp        0      0      *:imaps        *:*               LISTEN
tcp        0      0      bull:ssh       dhcp-32hw:4208    ESTABLISHED
tcp        0      0      bull:imaps     nubark:54195      ESTABLISHED
tcp        0      0      bull:http      dhcp-30hw:2563    ESTABLISHED
tcp        0      0      bull:imaps     dhcp-18hw:2851    ESTABLISHED
tcp        0      0      *:http         *:*               LISTEN
tcp        0      0      bull:37203     baikal:mysql      ESTABLISHED
tcp        0      0      *:ssh          *:*               LISTEN…
...
```

Esse exemplo é do host otter, e cortamos bastante a saída; por exemplo, as conexões de socket UDP e UNIX não são exibidas. A saída anterior mostra uma conexão SSH interna, duas conexões IMAP internas, uma conexão HTTP interna, uma conexão MySQL externa e algumas portas que ouvem outras conexões.

Os endereços são mostrados no formato *nomeDoHost.serviço*, onde *serviço* é o número de uma porta. Para serviços bem conhecidos, **netstat** mostra a porta simbolicamente, utilizando o mapeamento definido em **/etc/services**. Você pode obter endereços numéricos com a opção **-n**. Como ocorre com a maioria das ferramentas de depuração de rede, se seu DNS não estiver funcionando, é complicado utilizar o **netstat** sem o flag **-n**.

`Send-Q` e `Recv-Q` ilustra os tamanhos das filas de envio e de recepção para a conexão no host local; os tamanhos de fila na outra extremidade de uma conexão TCP poderiam ser diferentes. Eles devem tender a 0 e, pelo menos, não ser consistentemente não-zero. Obviamente, se você estiver executando **netstat** por meio de um terminal de rede, a fila de envio para a sua conexão talvez nunca seja 0.

O estado da conexão tem significado somente para o TCP; o UDP é um protocolo sem conexões. Os estados mais comuns que você verá são `ESTABLISHED` para conexões ativas no momento, `LISTEN` para servidores aguardando conexões (normalmente não exibido sem a opção **-a**) e `TIME_WAIT` para conexões em processo de fechamento.

Essa tela é basicamente útil para depurar problemas de nível mais alto depois que você determinou que os recursos básicos da rede estão funcionando corretamente. Ela permite que você verifique se os servidores estão configurados corretamente e facilita o diagnóstico de certos tipos de comunicações defeituosas, particularmente com TCP. Por exemplo, uma conexão que permanece no estado `SYN_SENT` identifica um processo que está tentando contatar um servidor inexistente ou servidor de redes inacessível.

---

2. Esse campo possui significado somente em redes baseadas em CSMA/CD como Ethernet.
3. Conexões para "sockets de domínio UNIX" também são mostradas, mas como elas não estão relacionadas com redes, não vamos discuti-las aqui.

Consulte o Capítulo 28 para informações adicionais sobre ajuste do kernel.

Se **netstat** mostrar uma grande quantidade de conexões na condição `SYN_WAIT`, seu host provavelmente não poderá tratar o número de conexões sendo solicitado. Essa falta de adequação pode se dever às limitações de ajuste do kernel ou até mesmo a uma inundação maliciosa.

Identificando serviços de rede ouvintes

Uma pergunta comum nessa era preocupada com a segurança é: "Quais processos nessa máquina estão ouvindo na rede conexões entrantes?". **netstat -a** mostra todas as portas que estão ouvindo ativamente (qualquer porta TCP no estado `LISTEN` e potencialmente qualquer porta UDP); mas, em uma máquina ocupada, essas linhas podem se perder no ruído das conexões TCP estabelecidas. Use **netstat -l** para ver apenas as portas ouvintes. O formato da saída é o mesmo que o do **netstat -a**.

Você pode adicionar o flag **-p** para fazer com que o **netstat** identifique o específico processo associado com cada porta ouvinte. A saída de exemplo abaixo mostra três serviços comuns (**sshd, sendmail** e **named**), seguida por um incomum:

```
$ netstat -lp
...
tcp 0      0      0.0.0.0:22     0.0.0.0:*      LISTEN 23858/sshd
tcp 0      0      0.0.0.0:25     0.0.0.0:*      LISTEN 10342/sendmail
udp 0      0      0.0.0.0:53     0.0.0.0:*             30016/named
udp 0      0      0.0.0.0:962    0.0.0.0:*             38221/mudd
...
```

Aqui, **mudd** com PID 38221 está ouvindo na porta UDP 962. Dependendo da diretiva do seu site para software instalado pelo usuário, é recomendável seguir esse.

Examinando a tabela de roteamento

netstat -r exibe a tabela de roteamento do kernel. O exemplo a seguir é de uma máquina Red Hat com duas interfaces de rede. (A saída varia levemente entre distribuições Linux.)

```
$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS Window    irtt   Iface
192.168.1.0     0.0.0.0         255.255.255.0   U      0      0       0      eth0
10.2.5.0        0.0.0.0         255.255.255.0   U      0      0       0      eth1
127.0.0.0       0.0.0.0         255.0.0.0       U      0      0       0      lo
0.0.0.0         192.168.1.254   0.0.0.0         UG     0      0       40     eth0
...
```

Destinos e gateways podem ser exibidos tanto como nomes de host como endereços IP; o flag **-n** exige saída numérica.

Consulte a página 206 para mais informações sobre tabelas de roteamento.

Os `Flags` caracterizam a rota: `U` significa up (ativo), `G` é um gateway e `H` é uma rota de host. `U`, `G` e `H` juntos indicam uma rota de host que passa por um gateway intermediário. O flag `D` (não mostrado) indica uma rota resultante de um redirecionamento ICMP. Os campos remanescentes fornecem estatísticas sobre a rota: o número atual de conexões TCP usando a rota, o número de pacotes enviados e a interface utilizada.

Utilize essa forma de **netstat** para verificar a saúde da tabela de roteamento do sistema. É particularmente importante verificar se o sistema tem uma rota padrão e que essa rota seja correta. A rota padrão é representada por um endereço de destino todo preenchido com 0s (0.0.0.0). É possível que não haja uma entrada de rota padrão, mas essa configuração seria bastante atípica.

Visualizando estatísticas operacionais para protocolos de rede

netstat -s faz um dump do conteúdo dos contadores que estão espalhados pelo código que suporta a rede. A saída tem seções separadas para IP, ICMP, TCP e UDP. A seguir apresentamos trechos da saída gerada por **netstat -s** de um típico servidor; eles foram editados para mostrar somente os trechos de informação mais interessantes.

```
Ip:
    671349985 total packets received
    0 forwarded
    345 incoming packets discarded
    667912993 incoming packets delivered
    589623972 requests sent out
    60 dropped because of missing route
    203 fragments dropped after timeout
```

Certifique-se de que os pacotes não estão sendo excluídos ou descartados. É aceitável que alguns pacotes entrantes sejam descartados, mas um rápido aumento nessa métrica normalmente indica falta de memória ou algum outro problema de recursos.

```
Icmp:
    242023 ICMP messages received
    912 input ICMP message failed.
    ICMP input histogram:
            destination unreachable: 72120
            timeout in transit: 573
            echo requests: 17135
            echo replies: 152195
    66049 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
            destination unreachable: 48914
            echo replies: 17135
```

Neste exemplo, o número das requisições eco na seção de entrada corresponde com o número de respostas eco na seção de saída. Note que mensagens "destination unreachable" (destino inatingível) ainda podem ser geradas mesmo quando todos os pacotes são aparentemente encaminháveis. Pacotes inválidos podem eventualmente atingir um gateway que os rejeita, e mensagens de erros são então enviadas de volta ao longo da cadeia de gateways.

```
Tcp:
    4442780 active connections openings
    1023086 passive connection openings
    50399 failed connection attempts
    0 connection resets received
    44 connections established
    666674854 segments received
    585111784 segments send out
    107368 segments retransmited
    86 bad segments received.
    3047240 resets sent
Udp:
    4395827 packets received
    31586 packets to unknown port received.
    0 packet receive errors
    4289260 packets sent
```

É uma boa idéia desenvolver um entendimento sobre os intervalos normais dessas estatísticas para que você possa reconhecer estados patológicos.

19.5 SAR: INSPECIONANDO AS ATIVIDADES DE INTERFACE "AO VIVO"

Uma boa maneira de identificar problemas de rede é ver o que está acontecendo no momento. Quantos pacotes foram enviados nos últimos cinco minutos em uma dada interface? Quantos bytes? Estão ocorrendo colisões ou outros erros? Você pode responder a todas essas perguntas observando as atividades ao vivo de uma interface.

Nos sistemas UNIX tradicionais, **netstat -i** é a ferramenta preferida para esse papel. Infelizmente, a capacidade do **netstat** de informar atividades ao vivo de uma interface não funciona sob o Linux. Recomendamos uma ferramenta completamente diferente: **sar**. (Discutimos **sar** da perspectiva do monitoramento de sistemas geral na página 560.) A maioria das distribuições não instala **sar** por padrão, mas ela sempre está disponível como um pacote opcional.

Para fazer com que **sar** informe a atividade de uma interface a cada dois segundos por um período de um minuto (isto é, 30 relatórios), utilize a sintaxe **sar -n DEV 2 30**. O argumento **DEV** é uma palavra-chave literal, não um marcador de lugar para o nome de um dispositivo ou de uma interface.

A saída inclui leituras instantâneas e médias do uso da interface de rede em termos de bytes e pacotes. O exemplo a seguir é de uma máquina Red Hat com duas interfaces físicas. A segunda interface física (eth1) claramente não está em uso.

```
17:50:43   IFACE   rxpck/s  txpck/s  rxbyt/s   txbyt/s   rxcmp/s   txcmp/s rxmcst/s
17:50:45     lo      3.61     3.61   263.40    263.40      0.00      0.00     0.00
17:50:45   eth0     18.56    11.86  1364.43   1494.33      0.00      0.00     0.52
17:50:45   eth1      0.00     0.00     0.00      0.00      0.00      0.00     0.00
```

As duas primeiras colunas mostram a data/hora em que foi tirada uma amostra dos dados e os nomes das interfaces de rede. As duas colunas seguintes mostram o número de pacotes recebidos e transmitidos, respectivamente. As colunas `rxbyt/s` e `txbyt/s` são provavelmente as mais úteis uma vez que mostram a largura de banda real em uso. As três colunas finais fornecem estatísticas sobre pacotes compactados (`rxcmp/s`, `txcmp/s`) e multicast (`rxmcst/s`).

sar -n DEV é especialmente útil para rastrear a fonte dos erros. **ifconfig** pode alertá-lo sobre existência de problemas, mas ele não consegue lhe dizer se os erros provêm de um problema contínuo de baixo nível ou de um evento breve, mas catastrófico. Note a rede ao longo do tempo sob uma variedade de condições de carga para solidificar sua impressão do que está acontecendo. Tente executar **ping** com um grande payload (tamanho) de pacote enquanto observa a saída de **sar -n DEV**.

19.6 ANALISADORES DE PACOTES

tcpdump e Wireshark pertencem à mesma classe de ferramentas conhecidas como analisadores de pacotes (*packet sniffers*). Elas ouvem o tráfego em uma rede e registram ou imprimem pacotes que atendam a certos critérios especificados pelo usuário. Por exemplo, todos os pacotes enviados de ou para um determinado host ou pacotes TCP relacionados a uma determinada conexão de rede poderiam ser inspecionados.

Analisadores de pacotes são úteis tanto na solução de problemas conhecidos como na descoberta de problemas inteiramente novos. É uma boa idéia fazer uma análise ocasional de sua rede para certificar-se de que o tráfego está em ordem.

Como os analisadores de pacotes precisam estar aptos a interceptar tráfego que a máquina local não receberia normalmente (ou a que, pelo menos, prestaria atenção), o hardware da rede subjacente tem de permitir acesso a todos os pacotes. Tecnologias de transmissão como a Ethernet funcionam bem e também a maioria das outras redes locais modernas.

Consulte a página 248 para informações adicionais sobre switches de rede.

Como os analisadores de pacotes precisam ver a maior quantidade possível de tráfego de rede bruto, eles podem ser impedidos por comutadores de rede que, por projeto, tentam limitar a propagação de pacotes "desnecessários". Entretanto, ainda pode ser informativo experimentar um analisador em uma rede comutada. Talvez você descubra problemas relacionados com pacotes de broadcast ou multicast. Dependendo do fabricante de seu comutador, talvez você possa ser surpreendido com o nível de tráfego que poderá ver.

O hardware de interface, além de ter potencial acesso a todos os pacotes de rede, deve transportar esses pacotes até a camada de software. Os endereços de pacotes são normalmente verificados em hardware e somente pacotes de broadcast/multicast e aqueles endereçados ao host local são retransmitidos ao kernel. No "modo promíscuo", uma interface deixa que o kernel leia todos os pacotes da rede, mesmo aqueles destinados a outros hosts.

Os analisadores de pacotes entendem vários dos formatos de pacotes utilizados por serviços de rede padrão e normalmente são capazes de colocar esses pacotes em um formato legível para o usuário. Essa capacidade torna mais fácil rastrear o fluxo de uma conversação entre dois programas. Alguns analisadores imprimem o conteúdo ASCII de um pacote além do cabeçalho do pacote, que pode ser útil para investigar protocolos de camadas mais altas. Como alguns desses protocolos enviam informações (e até mesmo senhas) por meio da rede como texto puro, você tem de tomar cuidado para evitar invadir a privacidade de seus usuários.

Todos os exemplos de distribuição deste livro vêm com um analisador de pacotes. Um analisador tem de ler dados brutos de um dispositivo de rede; portanto, ele tem de ser executado como root. Embora diminua a chance de usuários comuns ouvirem o tráfego da rede, essa limitação não é realmente uma grande barreira. Algumas instalações optam por eliminar programas analisadores da maioria dos hosts para reduzir a chance de abuso. No mínimo, você deve verificar as interfaces de seu sistema para ter certeza de que elas não estão sendo executadas em modo promíscuo sem seu conhecimento ou consentimento. Em sistemas Linux, uma interface em modo promíscuo mostra o flag `PROMISC` na saída do status de **ifconfig**. Você também pode utilizar ferramentas como PromiScan (disponível em www.securityfriday.com) para verificar na rede interfaces que executam no modo promíscuo.

tcpdump: o rei dos analisadores

tcpdump, uma outra surpreendente ferramenta de rede de Van Jacobson, é incluída na maioria das distribuições Linux. **tcpdump** há muito tempo é um analisador padrão da indústria; a maioria das ferramentas de análise de rede lê e grava arquivos de rastreamento no formato "**tcpdump**".

Por padrão, **tcpdump** ouve a primeira interface de rede que ele encontra. Se escolher a interface errada, você poderá forçar uma interface com o flag **-i**. Se o DNS não estiver funcionando ou você simplesmente não quiser que **tcpdump** faça pesquisas de nomes, utilize a opção **-n**. Essa opção é importante porque um serviço de DNS lento pode fazer com que o filtro comece a descartar pacotes antes que eles possam ser tratados via **tcpdump**. O flag **-v** aumenta o nível de informações que podem ser vistas sobre os pacotes; e **-vv** fornece ainda mais dados. Por fim, **tcpdump** pode armazenar pacotes em um arquivo com o flag **-w** e pode lê-los de volta com o flag **-r**.

Por exemplo, a saída truncada a seguir vem da máquina chamada nubark. A especificação do filtro **host bull** limita a exibição de pacotes àqueles que envolvem diretamente a máquina bull, como uma origem ou como um destino.

```
# sudo tcpdump host bull
12:35:23.519339 bull.41537 > nubark.domain:  A? atrust.com. (28) (DF)
12:35:23.519961 nubark.domain > bull.41537:  A 66.77.122.161 (112) (DF)
```

O primeiro pacote mostra o host bull enviando uma solicitação de pesquisa de DNS sobre atrust.com a nubark. A resposta é o endereço IP da máquina associada com esse nome, que é 66.77.122.161. Observe o registro de data/hora à esquerda e o entendimento do **tcpdump** do protocolo da camada de aplicativo (nesse caso, DNS). O número da porta em bull é arbitrário e é mostrado numericamente (41537), mas como o número da porta de servidor (53) é bem conhecido, em vez disso o **tcpdump** mostra seu nome simbólico ("domínio").

Analisadores de pacotes podem produzir uma quantidade exagerada de informações — exagerada não só para você, mas também para o sistema operacional subjacente. Para evitar esse problema nas redes ocupadas, o **tcpdump** permite especificar filtros relativamente complexos. Por exemplo, o filtro a seguir só coleta o tráfego Web entrante de uma sub-rede específica:

```
# sudo tcpdump src net 192.168.1.0/24 and dst port 80
```

A página man do **tcpdump** contém vários bons exemplos de filtragem avançada junto com uma listagem completa dos primitivos.[4]

Wireshark: analisador visual

Se você for mais propenso a utilizar um programa baseado em interface gráfica para análise de pacotes, então o Wireshark talvez seja o indicado para você. Disponível sob a GNU General Public License em www.wireshark.org, o Wireshark é um analisador de pacotes com interface gráfica com o usuário baseada no GTK+(GIMP tool kit) que tem mais funcionalidades que a maioria dos produtos comerciais para análise de pacotes. Você executa o Wireshark em seu desktop Linux ou, se o seu laptop ainda estiver sofrendo terrivelmente nas sombrias eras do Windows, você também poderá fazer o download de seus binários.

Além de analisar pacotes, o Wireshark tem dois recursos que o tornam muito prático. Um excelente recurso é que o Wireshark pode ler e gravar um grande número de outros formatos de arquivos de rastreamento de pacotes, incluindo estes:

- TCPDUMP
- Sniffer da NAI
- Sniffer Pro
- NetXray
- Snoop
- Shomiti Surveyor
- Network Monitor da Microsoft
- LANalyzer da Novell
- Secure IDS iplog da Cisco

O segundo recurso superútil é você poder clicar sobre um pacote em um stream (fluxo) de TCP e pedir para o Wireshark para "remontar" (juntar) os dados de todos os pacotes contidos no stream. Esse recurso é útil se você quiser examinar rapidamente os dados transferidos durante uma conversação TCP completa, como uma conexão transportando uma mensagem de e-mail através da rede.[5]

O Wireshark tem filtros de captura que funcionam identicamente aos do **tcpdump**. Mas, cuidado — uma armadilha importante com o Wireshark é o recurso extra dos "filtros de exibição", que afeta o que você vê em vez daquilo que realmente é capturado pelo analisador. Estranhamente, filtros de exibição usam uma sintaxe completamente diferente da sintaxe dos filtros de captura.

O Wireshark é uma ferramenta de análise incrivelmente poderosa e é incluída em quase todos os kits de ferramenta de rede de especialistas. Além disso, ele também é uma ajuda inestimável à aprendizagem para aqueles começando a explorar pacotes na rede. O menu Help do Wireshark fornece vários excelentes exemplos de como começar a utilizá-lo. Não tenha medo de experimentar!

---

4. Se suas necessidades de filtragem excederem às capacidades do **tcpdump**, considere o **ngrep**, que pode filtrar pacotes de acordo com seu conteúdo.
5. Você pode usar o utilitário **tcpflow** para realizar uma operação semelhante na linha de comando a partir do rastreamento de **tcpdump**.

19.7 PROTOCOLOS DE GERENCIAMENTO DE REDES

As redes cresceram rapidamente, tanto em termos de tamanho como em valor, ao longo da última década e, juntamente com esse crescimento, veio a necessidade de uma maneira eficiente de gerenciá-las. Fabricantes e organizações de padrões abordaram esse desafio de várias maneiras diferentes. Os desenvolvimentos mais significativos foram a introdução de vários protocolos de gerenciamento de dispositivos padrão e uma fartura de produtos de alto nível que exploram esses protocolos.

Os protocolos de gerenciamento de rede padronizam o método de investigação de um dispositivo para descobrir sua configuração, saúde e conexões de rede. Além disso, eles permitem que algumas dessas informações possam ser modificadas para que o gerenciamento de rede possa ser padronizado por meio de diversos tipos de máquinas e realizado em uma localização centralizada.

O protocolo para gerenciamento mais comumente utilizado com o TCP/IP é o SNMP (Simple Network Management Protocol, protocolo simples de administração de rede). Apesar de seu nome, o SNMP é, na verdade, bem complexo. Ele define um espaço de nomes hierárquico de dados de administração e uma maneira de ler e gravar os dados em cada um dos nós. Eles também definem uma maneira como servidores e dispositivos gerenciados ("agentes") enviam mensagens de notificação de eventos ("traps") às estações de gerenciamento.

O protocolo SNMP em si é simples; a maior parte da complexidade do SNMP reside acima da camada de protocolos nas convenções para construção do espaço de nomes e no vocabulário desnecessariamente barroco que cerca o SNMP como uma concha protetora. Contanto que você não pense muito sobre seu funcionamento interno, o SNMP é fácil de utilizar.

Há vários outros padrões por aí. Muitos deles se originaram da Distributed Management Task Force (DMTF), que é responsável por conceitos como o WBEM (Web-Based Enterprise Management), DMI (Desktop Management Interface) e o CIM (Conceptual Interface Model). Alguns desses conceitos, particularmente o DMI, foram adotados por vários fornecedores importantes e podem se tornar um complemento útil (ou até mesmo um substitutivo) para o SNMP. Há muitos protocolos de gerenciamento proprietários disponíveis. Mas, por enquanto, a grande maioria das instalações utiliza SNMP para o gerenciamento de rede e do Linux.

Como o SNMP é somente um protocolo abstrato, você precisa tanto de um programa servidor ("agente") como de um cliente ("gerente") para fazer uso dele. (Talvez de forma contrária à intuição, o lado servidor do SNMP representa aquilo que está sendo gerenciado, e o lado cliente é o gerente.) Clientes variam desde utilitários de linha de comando simples a estações de gerenciamento dedicadas que exibem graficamente redes e falhas em equipamentos importantes que saltam aos olhos.

As estações dedicadas ao gerenciamento de redes são a principal razão da existência dos protocolos de gerenciamento. A maioria dos produtos permite que você crie um modelo topográfico da rede, bem como um modelo lógico; os dois são apresentados juntos na tela, juntamente com uma indicação contínua do estado de cada componente.

Da mesma maneira que um gráfico é capaz de revelar o significado oculto de uma página cheia de números, uma estação para gerenciamento de redes é capaz de sintetizar o estado de uma grande rede de um modo facilmente aceito pelo cérebro humano. Esse tipo de sumário executivo é quase impossível de se obter de uma outra maneira.

Uma grande vantagem do gerenciamento por protocolo é o fato de ele colocar todos os tipos de hardware de rede em um campo de nível de atuação. Os sistemas Linux são, basicamente, todos similares, mas roteadores, switches e outros componentes de baixo nível não são. Com SNMP, todos falam a mesma língua e podem ser sondados, reinicializados e configurados de um ponto central. É bom ter uma interface consistente com todo o hardware da rede.

19.8 SNMP: O SIMPLE NETWORK MANAGEMENT PROTOCOL

Quando o SNMP começou a ser utilizado amplamente no início dos anos 90, ele começou uma minicorrida do ouro. Centenas de empresas lançaram pacotes de gerenciamento SNMP. Da mesma forma, muitos fabricantes de hardware e de software começam a colocar um agente SNMP como parte de seus produtos.

Antes de mergulharmos nos detalhes do SNMP, devemos notar que a terminologia associada a ele é um dos jargões mais deploráveis encontrados no campo das redes computacionais. Os nomes-padrão para conceitos e objetos do SNMP complicam o entendimento do que está acontecendo. Os responsáveis por esse estado de coisas deviam ter seus teclados destruídos.

Organização do SNMP

Os dados SNMP são organizados segundo uma hierarquia padronizada. Essa organização forçada permite que o espaço de dados permaneça universal e extensível ao mesmo tempo, pelo menos teoricamente. Grandes porções são reservadas para expansões futuras e adições específicas de fornecedor são localizadas para evitar conflitos. A hierarquia de atribuição de nomes é formada por bases de informação de gerenciamento (Management Information Bases – MIBs), arquivos de texto estruturados que descrevem os dados que podem ser acessados via SNMP. Os MIBs contêm descrições de variáveis de dados específicas, referidas por meio de nomes conhecidos como OIDs (Object Identifiers).

Traduzindo para o português, isso significa que o SNMP define um espaço de nomes hierárquico de variáveis cujos valores estão associados a parâmetros "interessantes" do sistema. Um OID é apenas uma maneira sofisticada de nomear um específico trecho gerenciado das informações.

A hierarquia SNMP é muito parecida com a de um sistema de arquivos. Mas usa-se um ponto como caractere separador, e cada nó recebe um número em vez de um nome. Por convenção, os nós também recebem nomes textuais para facilidade de referência, mas essa atribuição de nomes é, na realidade, apenas uma conveniência de alto nível, e não um recurso da hierarquia (ela é similar, em princípio, à associação de nomes de hosts a endereços IP).

Por exemplo, o OID que se refere ao tempo de funcionamento do sistema é 1.3.6.1.2.1.1.3. Esse OID também é conhecido pelo nome legível por humano

iso.org.dod.internet.mgmt.mib-2.system.sysUpTime

Os níveis altos da hierarquia SNMP são artefatos políticos e geralmente não contêm dados úteis. De fato, dados úteis podem ser encontrados atualmente apenas abaixo do OID iso.org.dod.internet.mgmt (numericamente, 1.3.6.1.2).

O MIB SNMP básico para TCP/IP (MIB-I) define o acesso a dados de gerenciamento comuns: informações sobre o sistema, suas interfaces, conversão de endereços e operações de protocolos (IP, ICMP, TCP, UDP, e outros). Um trabalho posterior e mais completo desse MIB (denominado MIB-II) é definido na RFC1213. A maioria dos fabricantes que fornecem um servidor SNMP suporta o MIB-II. A Tabela 19.1 a seguir apresenta uma amostragem dos nós do espaço de nomes MIB-II.

Tabela 19.1 OIDs selecionados do MIB-II

OID [a]	Tipo	Conteúdo
system.sysDescr	string	Informações de sistema: fabricante, modelo, tipo de sistema operacional etc.
system.sysLocation	string	Localização física da máquina
system.sysContact	string	Informações para contato para o proprietário da máquina
system.sysName	string	Nome do sistema, normalmente o nome DNS completo
interfaces.ifNumber	int	Número de interfaces de rede presentes
interfaces.ifTable	table	Tabela de informações sobre cada interface
ip.ipForwarding	int	1 se o sistema for um gateway; caso contrário, 2
ip.ipAddrTable	table	Tabela de dados de endereçamento IP (máscaras etc.)
ip.ipRouteTable	table	A tabela de roteamento do sistema
icmp.icmpInRedirects	int	Número de redirecionamentos ICMP recebidos
icmp.icmpInEchos	int	Número de pings recebidos
tcp.tcpConnTable	table	Tabela de conexões TCP atuais
udp.udpTable	table	Tabela de sockets UDP com servidores escutando

a. Relativo a iso.org.dod.internet.mgmt.mib-2.

Além da MIB básico, há MIBs para vários tipos de interfaces e protocolos de hardware, MIBs para fornecedores individuais e MIBs para produtos específicos de hardware. Um MIB para você, um MIB para mim, e muitos outros.

Um MIB é somente uma convenção em relação à atribuição de nomes de dados para gerenciamento. Para ser útil, um MIB tem de ser emparelhado com o código executado no agente que faz a associação entre o espaço de nomes SNMP e o real estado do dispositivo. O agente Linux padrão já vem com código para o MIB básico (agora MIB-II). Alguns agentes são extensíveis para incluir MIBs suplementares e outros não.

Operações de protocolos SNMP

Há somente quatro operações SNMP básicas: get, get-next, set e trap.

Get e set são as operações básicas para leitura e gravação de dados em um nó identificado por um OID específico. Get-next investiga uma hierarquia MIB e também pode ler o conteúdo das tabelas.

Trap é uma notificação assíncrona e não solicitada do servidor (agente) ao cliente (gerente) que relata a ocorrência de um evento ou condição interessante. Vários traps padrão são definidos, incluindo notificações do tipo "I've just come up", relatórios de falhas ou de recuperação de um link de rede e avisos sobre os vários problemas de roteamento e autenticação. Muitos outros traps não tão padronizados são utilizados comumente, inclusive alguns que apenas observam os valores de

outras variáveis SNMP e disparam uma mensagem quando um intervalo especificado é excedido. O mecanismo pelo qual os destinos de mensagens trap são especificados depende da implementação do agente.

Como mensagens SNMP têm um grande potencial para modificar informações de configuração, é necessário algum mecanismo de segurança. A versão mais simples de segurança SNMP se baseia no conceito de "string comunitária" SNMP que, na verdade, é apenas uma maneira horrivelmente obscura de dizer "senha". Normalmente há um nome comunitário para acesso somente leitura e um outro que permite gravação.

Embora muitas empresas ainda utilizem a autenticação original baseada em string comunitária, a versão 3 do padrão SNMP introduziu métodos de controle de acesso com melhor segurança. Embora configurar essa segurança mais avançada exija um pouco de trabalho extra, a redução dos riscos compensa bastante esse esforço. Se por alguma razão você não puder utilizar a versão 3 da segurança SNMP, pelo menos se certifique de selecionar uma string comunitária difícil de adivinhar.

RMON: MIB para monitoramento remoto

O MIB RMON permite a coleta de dados genéricos de desempenho da rede (isto é, dados não associados a nenhum dispositivo em particular). Os analisadores de rede ou "sondas" podem ser empregados pela rede para coletar informações sobre uso e desempenho. Assim que uma quantidade de dados razoável tiver sido coletada, estatísticas e informações interessantes sobre os dados poderão ser enviadas a uma estação de gerenciamento central para análise e apresentação. Muitas sondas possuem um buffer de captura de pacotes e são capazes de fornecer uma espécie de recurso **tcpdump** remoto.

O RMON é definido na RFC1757, que se tornou um padrão preliminar em 1995. O MIB se divide em nove "grupos RMON". Cada grupo contém um conjunto diferente de estatísticas de rede. Se tiver uma rede grande com muitas conexões remotas, você deve considerar a possibilidade de adquirir sondas para reduzir o tráfego SNMP por meio de seus links remotos. Assim que tiver acesso a sumários estatísticos das sondas RMON, normalmente não haverá mais necessidade de coletar dados brutos remotamente. Muitos comutadores e roteadores suportam o RMON e armazenarão pelo menos algumas estatísticas da rede.

19.9 O AGENTE NET-SMNP

Quando o agente SNMP foi padronizado pela primeira vez, a Carnegie Mellon University e o MIT produziram suas implementações. A implementação da CMU era mais completa e rapidamente se tornou um padrão de fato. Quando o desenvolvimento ativo na CMU caiu bastante, os pesquisadores da UC Davis assumiram o software. Após estabilizarem o código, eles transferiram a manutenção em andamento para o repositório SourceForge. O pacote é agora conhecido como NET-SNMP.

A distribuição NET-SNMP agora é a implementação SNMP gratuita oficial para o Linux. De fato, vários fornecedores de dispositivos de rede integraram o NET-SNMP aos seus produtos. O NET-SNMP inclui um agente SNMP, algumas ferramentas de linha de comando, um servidor para receber traps e mesmo uma biblioteca para desenvolver aplicativos compatíveis com SNMP. Discutimos o agente com certo nível de detalhes aqui, e na página 458, veremos as ferramentas de linha de comando. A versão mais recente está disponível em net-snmp.sourceforge.net.

Como ocorre em outras implementações, o agente coleta informações sobre o host local e o serve aos gerentes SNMP por meio da rede. A instalação padrão inclui MIBs para estatísticas de interfaces de rede, memória, disco, processos e CPU. O agente pode ser facilmente ampliado já que ele é capaz de executar um comando Linux arbitrário e retornar a saída gerada pelo comando na forma de uma resposta SNMP. Você pode utilizar esse recurso para monitorar praticamente qualquer coisa em seu sistema com SNMP.

Por padrão, o agente é instalado como **/usr/sbin/snmpd**. Normalmente ele é iniciado em tempo de inicialização e extrai informações relativas à configuração de arquivos contidos no diretório **/etc/snmp**. O mais importante desses arquivos é **snmpd.conf**, que contém a maior parte das informações de configuração e vem com uma série de modelos para métodos de coleta de dados já ativados. Embora a intenção dos autores do NET-SNMP parece ter sido a dos usuários poderem editar somente o arquivo **snmpd.local.conf**, é necessário editar o arquivo **snmpd.conf** pelo menos uma vez para desativar qualquer método de coleta de dados padrão que você não pretenda utilizar.

O script **configure** do NET-SNMP permite especificar um arquivo de log padrão e uma série de outras configurações locais. Podemos usr **snmpd -l** para especificar um arquivo de log alternativo ou **-s** para direcionar mensagens de log para syslog. A Tabela 19.2 lista os flags do **snmpd** mais importantes. Recomendamos que você sempre utilize o flag **-a**. Para depuração, você deve utilizar os flags **-V**, **-d** ou **-D**, cada um dos quais fornece progressivamente mais informações.

Tabela 19.2 Flags úteis para o snmpd do NET-SNMP

Flag	Função
-l *arquivoDeLog*	Registra informações no *arquivoDeLog*
-a	Registra em log os endereços de todas as conexões SNMP
-d	Registra em log o conteúdo de cada pacote SNMP
-V	Ativa o registro em log prolixo
-D	Registra em log informações de depuração (um grande número delas)
-h	Exibe todos os argumentos para **snmpd**
-H	Exibe todas as diretivas de arquivos de configuração
-A	Acrescenta informações ao arquivo de log em vez de sobrescrevê-la
-s	Grava o log em syslog (utiliza o recurso daemon)

Vale a pena mencionar que há disponíveis muitos módulos Perl úteis relacionados com SNMP. Vá ao CPAN[6] para informações mais recentes caso esteja interessado em escrever os seus próprios scripts para gerenciamento de redes.

19.10 APLICATIVOS PARA GERENCIAMENTO DE REDES

Começamos essa seção explorando as ferramentas de gerenciamento SNMP mais simples: os comandos fornecidos com o pacote NET-SNMP. Esses comandos podem ajudá-lo a se familiarizar com o SNMP e também são excelentes para verificações *one-off* dos OIDs específicos. A seguir, veremos Cacti, um programa que gera gráficos com históricos de valores SNMP, e Nagios, um sistema de monitoramento baseado em eventos. Concluímos com algumas recomendações sobre o que analisar ao comprar um sistema comercial.

As ferramentas NET-SNMP

Mesmo que seu sistema venha com um servidor SNMP próprio, você ainda pode querer compilar e instalar as ferramentas do lado cliente do pacote NET-SNMP. A Tabela 19.3 lista as ferramentas mais comumente utilizadas.

Tabela 19.3 Ferramentas de linha de comando no pacote NET-SNMP

Comando	Função
snmpdelta	Monitora, ao longo do tempo, alterações em variáveis SNMP
snmpdf	Monitora espaço em disco em um host remoto via SNMP
snmpget	Obtém o valor de uma variável SNMP por meio de um agente
snmpgetnext	Obtém a próxima variável na seqüência
snmpset	Configura uma variável SNMP num agente
snmptable	Obtém uma tabela das variáveis SNMP
snmptranslate	Pesquisa e descreve OIDs na hierarquia MIB
snmptrap	Gera um alerta de captura
snmpwalk	Percorre um MIB iniciando em um dado OID

Além dos valores na linha de comandos, esses programas são tremendamente práticos em scripts simples. Normalmente é útil fazer **snmpget** salvar valores de dados interessantes em um arquivo de texto a cada minuto. (Utilize **cron** para implementar o cronograma; consulte o Capítulo 8, *Processos periódicos*.)

snmpwalk é mais uma ferramenta útil. Iniciando em um OID especificado (ou no início do MIB, por padrão), esse comando realiza repetidamente chamadas "get next" a um agente. Esse comportamento resulta em uma lista completa de OIDs disponíveis e seus valores associados. **snmpwalk** é particularmente útil quando você está tentando identificar novos OIDs a fim de monitorar a partir da sua sofisticada ferramenta de gerenciamento corporativo.

6. CPAN (Comprehensive Perl Archive Network) é um excelente conjunto de módulos Perl úteis. Veja em www.cpan.org.

Eis um exemplo truncado de **snmpwalk** do host tuva. A string comunitária é "secret813community" e **-v1** especifica autenticação simples.

```
$ snmpwalk -c secret813community -v1 tuva
SNMPv2-MIB::sysDescr.0 = STRING: Linux tuva.atrust.com 2.6.9-11.ELsmp #1
SNMPv2-MIB::sysUpTime.0 = Timeticks: (1442) 0:00:14.42
SNMPv2-MIB::sysName.0 = STRING: tuva.atrust.com
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifType.3 = INTEGER: ethernetCsmacd(6)
IF-MIB::ifPhysAddress.1 = STRING:
IF-MIB::ifPhysAddress.2 = STRING: 0:11:43:d9:1e:f5
IF-MIB::ifPhysAddress.3 = STRING: 0:11:43:d9:1e:f6
IF-MIB::ifInOctets.1 = Counter32: 2605613514
IF-MIB::ifInOctets.2 = Counter32: 1543105654
IF-MIB::ifInOctets.3 = Counter32: 46312345
IF-MIB::ifInUcastPkts.1 = Counter32: 389536156
IF-MIB::ifInUcastPkts.2 = Counter32: 892959265
IF-MIB::ifInUcastPkts.3 = Counter32: 7712325
...
```

Nesse exemplo, vemos algumas informações genéricas sobre o sistema, seguidas por estatísticas sobre as interfaces de rede do host, lo0, eth0 e eth1. Dependendo dos MIBs suportados pelo agente que você está gerenciando, um dump completo pode chegar a centenas de linhas.

Coleta e representação gráfica dos dados SNMP

Dados relacionados à rede são mais bem apreciados nos contextos visuais e históricos. É importante ter alguma maneira de monitorar e representar graficamente a métrica do desempenho, mas sua escolha exata do software para fazer isso não é essencial.

Um dos pacotes de consulta seqüencial (*polling*) e representação gráfica mais populares do SNMP inicial era o MRTG, escrito por Tobi Oetiker da ETH, em Zurique. O MRTG foi escrito principalmente em Perl, executa regularmente a partir do **cron** e pode coletar dados a partir de qualquer fonte SNMP. Cada vez que o programa é executado, novos dados são armazenados e imagens, criadas.

Recentemente, Tobi concentrou sua energia no RRDTool, um kit de ferramentas para armazenar e representar graficamente a métrica de desempenho. Todas as principais soluções de código-fonte aberto para monitoração estão baseadas no RRDTool, incluindo nossa favorita, Cacti.

O Cacti, disponível em www.cacti.net, oferece vários recursos atraentes. Primeiramente, ele tem um banco de dados dimensionado estaticamente e de manutenção zero; o software armazena somente dados suficientes para criar os gráficos necessários. Por exemplo, Cacti poderia armazenar uma amostra a cada minuto durante um dia, uma amostra a cada hora durante uma semana e uma amostra por semana durante um ano. Esse esquema de consolidação permite que você preserve informações históricas importantes sem ter de armazenar detalhes insignificantes ou perder tempo com a administração do banco de dados.

Em segundo lugar, o Cacti pode registrar e representar graficamente qualquer variável SNMP, bem como muitas outras métricas de desempenho. Você fica livre para coletar quaisquer dados que quiser. Quando combinado com o agente NET-SNMP, o Cacti pode fornecer uma perspectiva histórica de praticamente qualquer sistema ou recurso de rede.

A Ilustração A mostra alguns exemplos dos gráficos gerados por Cacti. Esses gráficos mostram a média de carga em um servidor em relação a um período de múltiplas semanas juntamente com o tráfego de um dia em uma interface de rede.

O Cacti apresenta uma configuração fácil baseada na Web e também todos os outros benefícios predefinidos do RRDTool, como manutenção baixa e uma linda representação gráfica. Acesse a home page RRDTool de Tobi Oetiker em www.rrdtool.org para links para as versões atuais do RRDtool e do Cacti e também dezenas de outras ferramentas de monitoração.

Nagios: SNMP baseado em eventos e monitoramento de serviços

O Nagios especializa-se em gerar relatórios em tempo real das condições de erro. Ele pode consultar qualquer métrica SNMP (bem como centenas de outros serviços de rede) e alertar sobre condições definidas de erro. Embora não ajude a determinar qual foi o aumento da utilização de largura de banda no último mês, ele emitirá uma mensagem via pager quando o seu servidor Web cair.

A distribuição do Nagios inclui plug-ins que supervisionam uma variedade de pontos comuns de falha. Você pode ativar novos monitoramentos em Perl ou até em C se estiver se sentindo seguro. Para os métodos de notificação, a distribuição pode enviar e-mail, gerar relatórios Web e utilizar um modem discado para enviar uma mensagem ao seu pager. Como ocorre com os plug-ins, é fácil criar seu próprio esquema de monitoração.

Além de enviar notificações em tempo real sobre falhas de serviços, o Nagios mantém um repositório de arquivos do histórico desses dados. Ele fornece várias interfaces poderosas para gerar relatórios que monitoram as tendências da disponibilidade e do desempenho. Muitas empresas utilizam o Nagios para medir a conformidade com os acordos no nível dos serviços; a Ilustração B, a seguir, mostra a disponibilidade de um servidor DNS.

Ilustração A Exemplos dos gráficos Cacti

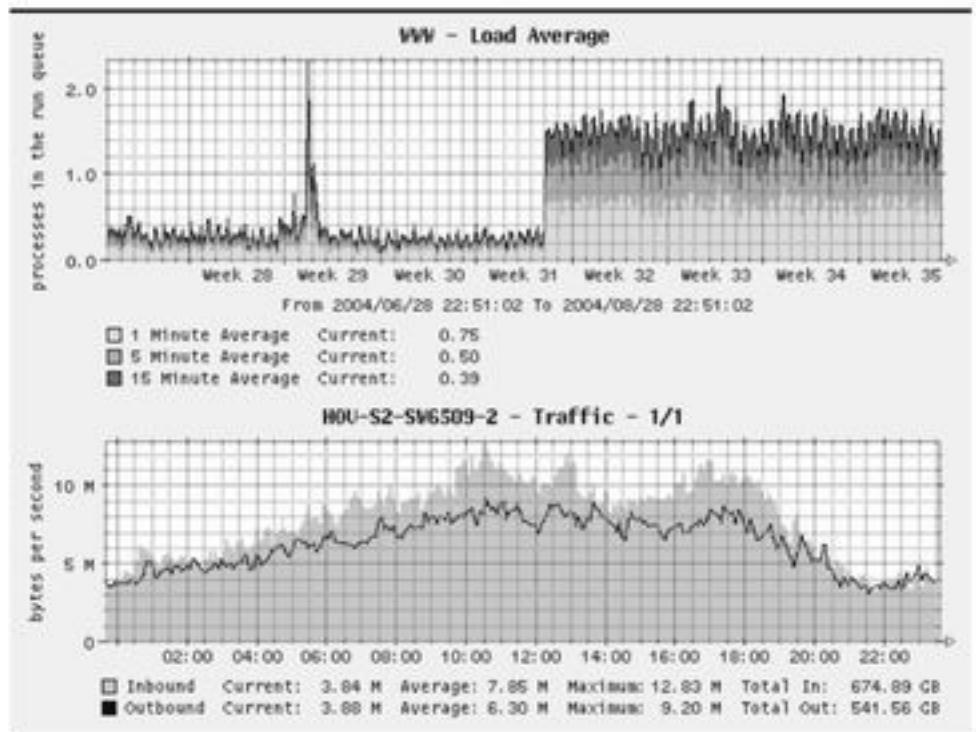
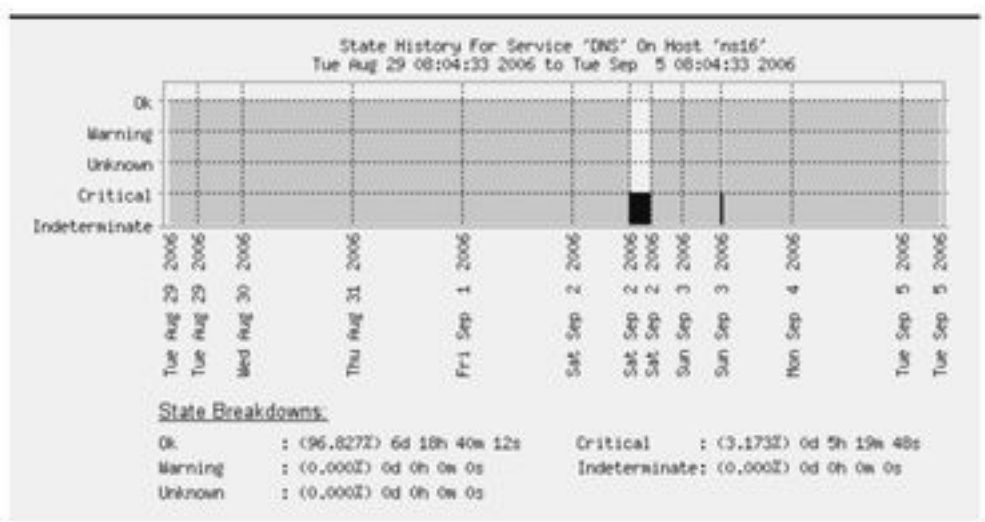


Ilustração B A disponibilidade do servidor como mostrada pelo Nagios



O Nagios funciona muito bem para redes com menos de cem mil hosts e dispositivos. Ele é fácil de personalizar e de estender e inclui recursos poderosos como redundância, monitoração remota e escalonamento das notificações. Se você não puder comprar uma ferramenta comercial de gerenciamento de rede, recomendamos muito que você considere o Nagios. Leia mais em www.nagios.org.

Plataformas comerciais para gerenciamento

Centenas de empresas vendem software para gerenciamento de redes, e novos competidores entram no mercado toda semana. Em vez de recomendar os produtos mais "quentes" do momento (que talvez possam já não mais existir quando este livro for impresso), identificaremos os recursos que você deve procurar em um sistema de gerenciamento de redes.

Flexibilidade de coleta de dados: As ferramentas de gerenciamento precisam ser capazes de coletar dados de outras fontes além do SNMP. Muitos pacotes incluem maneiras de coletar dados a partir de quase todos os serviços de rede. Por exemplo, alguns pacotes podem realizar consultas a bancos de dados SQL, verificar registros DNS e conectar-se a servidores Web.

Qualidade da interface com o usuário: Sistemas caros normalmente oferecem uma interface gráfica do usuário personalizada ou uma interface Web. A maioria dos pacotes bem comercializados hoje promove a capacidade de entender gabaritos XML para apresentação de dados. Uma interface gráfica não é apenas mais uma propaganda de marketing — você precisa de uma interface que retransmita as informações de uma maneira clara, simples e compreensível.

Valor: Alguns pacotes para gerenciamento têm um preço salgado. O OpenView do HP é um dos sistemas de gerenciamento de rede mais caros e um dos mais amplamente adotados. Muitas empresas descobrem a importância de poder dizer que seus sites são gerenciados por um sistema comercial sofisticado. Se isso não for importante para a sua empresa, você deve olhar na outra ponta do espectro para ferramentas gratuitas como Cacti e Nagios.

Descoberta automatizada: Muitos sistemas oferecem a capacidade de "descobrir" sua rede. Por meio de uma combinação de pings de broadcast, solicitações de SNMP, pesquisas de tabelas ARP e consultas DNS, eles são capazes de identificar todos os seus hosts e dispositivos locais. Todas as implementações de descoberta que vimos funcionam muito bem, mas nenhuma é muito precisa em uma rede complexa (ou fortemente protegida por firewalls).

Recursos para geração de relatórios: Muitos produtos podem enviar e-mails de alerta, ativar pagers e gerar automaticamente tíquetes para sistemas de rastreamento de problemas populares. Certifique-se de que a plataforma escolhida permita geração flexível de relatórios; ninguém sabe quais serão os tipos de dispositivos eletrônicos com os quais estaremos lidando daqui a poucos anos.

Gerenciamento de configurações: Alguns fabricantes vão muito além de monitorar e alertar. Eles permitem gerenciar as configurações reais de hosts e dispositivos. Por exemplo, uma interface CiscoWorks permite alterar a configuração de um roteador e também monitorar seu estado com o SNMP. Como as informações sobre a configuração de dispositivos aprofundam a análise dos problemas de rede, prevemos que vários pacotes serão desenvolvidos ao longo dessas linhas no futuro.

19.11 LEITURA RECOMENDADA

CISCO ONLINE. *SNMP Overview*. 1996. www.cisco.com/warp/public/535/3.html

MAURO, DOUGLAS R E SCHMIDT, KEVIN J. *Essential SNMP. 2. ed*. Sebastopol, CA: O'Reilly Media, 2005.

SIMPLEWEB. *SNMP and Internet Management Site*. Disponível em: <www.simpleweb.org>.

STALLINGS, WILLIAM. *Snmp, Snmpv2, Snmpv3, and Rmon 1 and 2. 3. ed*. MA: Addison-Wesley, 1999.

Talvez você também possa achar úteis as seguintes RFCs. Substituímos os títulos reais das RFCs por uma descrição do conteúdo da RFC, pois alguns títulos reais são uma mistura inútil de termos da moda e jargões SNMP.

- RFC1155 – Características do espaço de dados SNMP (tipos de dados etc.)
- RFC1156 – Definições MIB-I (descrição dos verdadeiros OIDs)
- RFC1157 – Simple Network Management Protocol
- RFC1213 – Definições MIB-II (OIDs)
- RFC3414 – Modelo de segurança baseado em usuário para SNMPv3
- RFC3415 – Modelo de controle de acesso baseado em visualização para SNMPv3
- RFC3512 – Configurando dispositivos com SNMP (melhor visão geral)
- RFC3584 – Coexistência prática entre diferentes versões SNMP

19.12 EXERCÍCIOS

E19.1 Você está diagnosticando um problema de rede, e **netstat -rn** gera a seguinte saída. Qual é o problema e que comando você usaria para corrigi-lo?

```
Destination       Gateway       Genmask       Flags  MSS    Window  irtt   Iface
128.138.202.0     0.0.0.0       255.255.255.0 U       40    0          0   eth0
127.0.0.0         0.0.0.0       255.0.0.0     U       40    0          0   lo
```

★ E19.2 Crie um script que monitore um dado conjunto de máquinas e notifique um administrador por e-mail caso uma máquina não responda a pings durante um período preestabelecido. Não programe diretamente no código a lista de máquinas, o endereço de e-mail para notificação ou a quantidade de tempo para determinar o comportamento não-responsivo.

★ E19.3 Experimente alterar a máscara de rede em uma máquina em sua rede local. Ela ainda funciona? Você consegue atingir tudo em sua instalação? Outras máquinas conseguem se comunicar com você? As broadcasts funcionam (por exemplo, solicitações ARP ou pacotes de descoberta DHCP)? Explique suas descobertas. (Exige acesso de root.)

★ E19.4 Utilize o comando **traceroute** para descobrir caminhos de roteamento em sua rede.

a) Quantos saltos ele leva para deixar sua instalação?

b) Há algum roteador entre as máquinas em que você tem contas?

c) Você é capaz de encontrar algum gargalo?

d) O seu site tem várias rotas de acesso?

★ ★ E19.5 Projete um MIB que inclua todas as variáveis que talvez você, como administrador de sistemas Linux, queira consultar ou configurar. Forneça maneiras de como o MIB pode ser estendido para incluir aquela nova e importante variável de administração do sistema que você esqueceu.

★ ★ E19.6 Utilize o comando **tcpdump** para capturar tráfego que ilustre os seguintes protocolos. Para sessões TCP, inclua e indique os pacotes inicial e final. Apresente a saída do **tcpdump** organizada e bem formatada. (Exige acesso de root.)

a) ARP

b) Solicitação de eco e resposta ICMP

c) SMTP

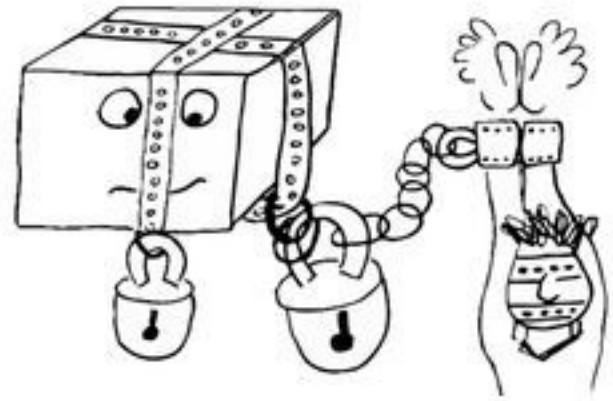
d) FTP e FTP-DATA

e) DNS (domínio chamado)

f) NFS

★ ★ E19.7 Configure gráficos do Cacti que ilustrem os pacotes transmitidos de e para um roteador local. Esse projeto exige que um pacote SNMP consulte o roteador; você tem de conhecer a string comunitária somente para leitura do roteador.

20 Segurança



Jogos de Guerra, um filme de 1983 estrelado por Matthew Broderick, é um fascinante olhar sobre a subcultura do período inicial de crack de computador. No filme, David Lightman emprega várias técnicas bem conhecidas de ataque, incluindo guerra de discagem (*war dialing*), engenharia social, hacking de telefone (conhecido como *phreaking*) e adivinhação de senha para começar e então parar um ataque nuclear. *Jogos de Guerra* colocou a segurança de computador em evidência e, com ela, veio o inapropriado nome "hacker" como uma maneira de rotular alguns tipos de adolescentes impertinentes. *Jogos de Guerra* foi só o começo do que agora passou a ser um problema global.

Cinco anos depois, o mundo da tecnologia dos computadores e redes entrou em uma nova era quando o "Internet Worm", de Robert Morris Jr., desabou sobre a humanidade. Antes desse evento, a Internet vivia a idade da inocência. Segurança era um tópico que os administradores pensavam em grande parte no sentido "o que aconteceria se". Um grande incidente de segurança normalmente consistia de algo como um usuário ganhar acesso administrativo para ler a correspondência de um outro usuário, muitas vez só para provar que ele era capaz.

O worm Morris fez com que milhares de administradores desperdiçassem milhares de horas, mas aumentou significativamente a consciência quanto à segurança na Internet. Mais uma vez, fomos penosamente relembrados de que boas cercas fazem bons vizinhos. Como resultado, surgiu uma série de ferramentas excelentes para serem utilizadas por administradores de sistemas (bem como uma organização formal para tratar de incidentes desta natureza).

De acordo com o *Computer Crime and Security Survey* do CSI/FBI (instituto para segurança eletrônica do FBI) de 2006,[1] a maioria das empresas pesquisadas informou que investiram menos de 5% do orçamento de TI em segurança. Como resultado, um total assustador de US$ 54.494.290 (US$ 174.103 por empresa) foi perdido devido a brechas de segurança nos computadores. Esses números apresentam um cenário assustador da consciência por parte das empresas em relação à segurança. Naturalmente, os dólares gastos não significam necessariamente um sistema seguro; um administrador atento pode parar qualquer invasor seguindo as pegadas dele.

Em geral, a segurança não é algo que você pode adquirir em um produto ou em um serviço oferecido por terceiros. Os serviços e produtos comerciais podem fazer parte de uma solução para o seu site, mas eles não são uma panacéia.

Alcançar um nível aceitável de segurança exige muita paciência, vigilância, conhecimento e persistência — não só de você e outros administradores, mas também de todos os seus usuários e de toda a comunidade de gerenciamento. Como administrador de sistemas, você deve assumir pessoalmente a responsabilidade de garantir que seus sistemas são seguros e que seus usuários estão adequadamente treinados. Você precisa conhecer a atual tecnologia de segurança, pesquisar ativamente listas de discussão sobre segurança e contratar especialistas em segurança para ajudá-lo nos problemas que estão além do seu conhecimento.

20.1 O LINUX É SEGURO?

Não, o Linux não é seguro. Assim como nenhum outro sistema operacional que se comunica via rede. Se precisar de segurança total, absoluta, e inquebrável você então precisará de uma "air gap"[2] bastante grande entre seu computador e um outro dispositivo qualquer. Algumas pessoas argumentam que você também precisa colocar seu computador em um local especial que impeça radiação eletromagnética. (Pesquise no Google "jaula de Faraday" ou "gaiola de Faraday".) Que graça há nisso?

Você pode trabalhar para fazer seu sistema ficar um pouco mais resistente a ataques. Mesmo assim, várias falhas fundamentais no modelo do Linux asseguram que você nunca alcançará o nirvana da segurança:

- Como o UNIX, o Linux é otimizado por conveniência e não torna a segurança fácil ou natural. A filosofia Linux enfatiza a manipulação fácil dos dados em um ambiente multiusuário em rede.

---

1. Essa pesquisa é conduzida anualmente e pode ser acessada em www.gocsi.com.
2. É claro que a tecnologia de rede sem fio introduz uma nova série de problemas. "Air gap" nesse contexto significa "nenhuma rede, qualquer que seja".

- A menos que cuidadosamente implementada, a segurança do Linux é efetivamente binária: ou você é um usuário impotente ou então é root. Recursos Linux como a execução setuid tendem a conferir total poder de uma só vez. Graças ao Security-Enhanced Linux, um projeto da National Security Agency, algum progresso foi feito para a implementação de controles de acesso mais granulares. Mas, na maioria dos casos, pequenos lapsos na segurança ainda podem comprometer sistemas inteiros.
- As distribuições Linux são desenvolvidas por uma grande comunidade de programadores. Elas variam em nível de experiência, atenção a detalhes e conhecimento sobre o sistema Linux e suas interdependências. Como resultado, mesmo os novos recursos mais bem-intencionados podem introduzir grandes brechas de segurança.

Por outro lado, como o código-fonte do Linux se encontra disponível a qualquer um, milhares de pessoas podem examinar minuciosamente (e examinam) cada linha de código em busca de uma possível ameaça em termos de segurança. Acredita-se amplamente que esse tipo de arranjo resulte em melhor segurança do que os sistemas operacionais fechados, em que um número limitado de pessoas tem a oportunidade de examinar o código em busca de brechas de segurança.

Muitos sites estão uma ou duas versões atrás, seja porque a localização é muito problemática ou porque eles não se inscreveram em um serviço de manutenção de software de um distribuidor. Em qualquer caso, quando brechas de segurança são corrigidas, o leque de oportunidades para hackers freqüentemente não desaparece da noite para o dia.

Pode parecer que a segurança no Linux devesse aumentar gradualmente ao longo do tempo enquanto problemas relacionados com segurança fossem descobertos e corrigidos, mas, infelizmente, parece que esse não é o caso. O software de sistema está crescendo e se tornando cada vez mais complicado, os hackers estão ficando melhores e mais bem organizados e os computadores estão se conectando cada vez mais intimamente na Internet. A segurança é uma batalha contínua que nunca pode ser realmente vencida. (Naturalmente, o mundo Windows é ainda pior…)

Lembre-se também de que:

$$\text{Segurança} = \frac{1}{(1.072)\ (\text{Conveniência})}$$

Quanto mais seguro for o seu sistema, maiores serão as suas restrições e a dos seus usuários. Só implemente as medidas de segurança sugeridas neste capítulo depois de considerar cuidadosamente todas as implicações para os seus usuários.

20.2 Como a segurança é comprometida

Este capítulo discute alguns problemas comuns de segurança do Linux e suas contramedidas padrão. Mas, antes de partirmos para os detalhes, devemos ter uma visão mais genérica de como problemas reais de segurança tendem a ocorrer. A maioria dos lapsos de segurança entra nesta taxinomia.

Engenharia social

Usuários humanos (e administradores também) de um sistema computacional costumam ser os elos mais fracos na cadeia de segurança. Mesmo no mundo atual em que há maior consciência quanto à segurança, usuários não suspeitos com boas intenções são facilmente convencidos a passar informações sigilosas.

Esse problema se manifesta de muitas formas. Invasores utilizam chamadas telefônicas para atacarem suas vítimas e se passam por usuários legitimamente confusos em uma tentativa de obter ajuda para acessar o sistema. Administradores postam, sem intenções, informações sigilosas em fóruns públicos ao solucionar problemas. Comprometimentos físicos ocorrem quando o pessoal de manutenção aparentemente legítimo reinstala o gabinete de cabos telefônicos.

O termo "phishing" descreve tentativas de coletar informações dos usuários por meio de e-mail enganoso, mensagens instantâneas ou mesmo mensagens SMS de celulares. A defesa contra phishing pode ser especialmente difícil porque as comunicações com freqüência incluem informações específicas das vítimas que lhes dão a aparência de autenticidade.

A engenharia social continua a ser uma técnica poderosa de hacking e é uma das ameaças mais difíceis de neutralizar. A diretiva de segurança do seu site deve incluir treinamento de novos funcionários. Comunicações normais por toda a empresa são uma maneira eficaz de fornecer informações sobre os prós e contras dos telefones, segurança física, phishing de e-mail e seleção de senhas.

Para estimar a resistência da sua empresa à engenharia social, poderia ser informativo você mesmo tentar alguns ataques de engenharia social. Mas certifique-se de que você tem permissão explícita dos seus gerentes. Essas explorações parecem muitas suspeitas se realizadas sem um mandato claro. Elas também são uma forma de espionagem interna, portanto podem gerar aborrecimentos se elas não forem tratadas de uma maneira clara e direta.

Vulnerabilidades de software

Ao longo dos anos, um número incontável de bugs referentes à segurança foram descobertos em softwares de computadores (inclusive software de terceiros, sejam eles produtos comerciais ou gratuitos). Explorando erros de programação sutis ou dependências de contexto, os hackers foram capazes de manipular o Linux a fazer o que bem entendessem.

Overflows de buffer são um erro de programação comum e um com implicações complexas. Os desenvolvedores costumam alocar uma quantidade predeterminada de espaço temporário para a memória, chamado buffer, para armazenar determinados trechos de informações. Se o código não for cuidadoso ao verificar o tamanho dos dados comparando-o com o tamanho do contêiner que supostamente deve contê-los, a memória adjacente ao espaço alocado correrá o risco de ser sobrescrita. Hackers astuciosos podem inserir dados cuidadosamente compostos que travam o programa ou, no pior caso, executam um código arbitrário.

Felizmente, nos últimos anos, o número total de explorações via overflows de buffer aumentou a consciência da comunidade de programação sobre essa questão. Embora problemas de overflow de buffer continuem a ocorrer, eles costumam ser rapidamente descobertos e corrigidos, especialmente nos aplicativos de código-fonte aberto. Os sistemas de programação mais recentes como o Java e o .NET incluem mecanismos que verificam automaticamente o tamanho dos dados e evitam overflows de buffer.

Overflows de buffer são uma subcategoria de uma classe maior de bugs de segurança de software conhecida como vulnerabilidades de validação de entrada. Quase todos os programas aceitam algum tipo de entrada dos usuários (por exemplo, argumentos de linha de comando ou formulários HTML). Se o código processar esses dados sem verificá-los rigorosamente quanto ao formato e conteúdo apropriados, coisas ruins podem acontecer. Considere o exemplo simples a seguir:

```
#!/usr/bin/perl
# Erro de validação de entrada de usuário

open (HTMLFILE, "/var/www/html/$ARGV[0]") or die "trying\n";
while (<HTMLFILE>) { print; }
close HTMLFILE;
```

É provável que a intenção desse código seja imprimir o conteúdo de algum arquivo HTML sob **/var/www/html**, que é a raiz de documentos padrão para servidor `httpd` do Apache nos servidores Red Hat. Esse código aceita um nome de arquivo do usuário e o passa como o argumento para a função `open`. Entretanto, se um usuário malicioso inserisse **../../../etc/shadow** como o argumento, o conteúdo de **/etc/shadow** seria ecoado!

O que você como um administrador pode fazer para evitar esse tipo de ataque? Muito pouco, pelo menos enquanto um bug não tiver sido identificado e tratado em uma correção. Manter-se atualizado em relação às correções e aos boletins de segurança é uma parte importante das tarefas da maioria dos administradores. A maioria das distribuições inclui utilitários automatizados de correção, como **yum** no Fedora e **apt-get** no Debian e Ubuntu. Tire proveito desses utilitários para manter seu site seguro a vulnerabilidades de software.

Erros de configuração

Muitos pacotes e software podem ser configurados de maneira segura ou não tão segura. Infelizmente, como os softwares são desenvolvidos por conveniência, o padrão costuma não ser tão seguro. Os hackers freqüentemente ganham acesso explorando recursos de software que seriam considerados úteis e convenientes em circunstâncias menos insidiosas: contas sem senhas, discos compartilhados com o mundo e bancos de dados desprotegidos, para citar alguns.

Um exemplo típico de uma vulnerabilidade de configuração de host é a prática padrão de permitir que sistemas Linux inicializem sem exigir uma senha de rotina de partida. Tanto LILO como GRUB podem ser configurados em tempo de instalação para exigir uma senha, mas os administradores quase sempre não utilizam essa opção. Essa omissão deixa o sistema aberto a ataques físicos. Entretanto, ela também é um exemplo perfeito da necessidade de balancear a segurança contra a usabilidade. Exigir uma senha significa que se o sistema for reinicializado não intencionalmente (por exemplo, depois de uma falha de energia), um administrador teria de estar fisicamente presente para colocar a máquina em execução novamente.

Uma das partes mais importantes para assegurar um sistema é simplesmente ter certeza de que você não deixou inadvertidamente um tapete de boas-vindas para hackers. Problemas que entram na última dessas categorias são os mais fáceis de localizar e corrigir, embora haja potencialmente uma grande quantidade deles e nem sempre é óbvio o que verificar. As ferramentas de varredura de porta e de vulnerabilidade, discutidas mais adiante neste capítulo, podem ajudar um administrador motivado a localizar e corrigir problemas antes de eles serem explorados.

20.3 CERTIFICAÇÕES E PADRÕES

Se você achar que o assunto deste capítulo é desencorajador, não desanime. A segurança de computador é um tópico complicado e vasto, como incontáveis livros, sites Web e revistas podem atestar. Felizmente, muito tem sido feito para ajudar a quantificar e organizar as informações disponíveis. Há dezenas de padrões e certificações e os administradores Linux atentos devem levar em conta a orientação desses padrões e certificações.

Um dos princípios filosóficos mais básicos na segurança de informações é informalmente conhecido como a "tríade CIA". O acrônimo significa:

- Confidencialidade
- Integridade
- Disponibilidade

A confidencialidade dos dados diz respeito à privacidade dos dados. Essencialmente, o acesso às informações deve estar limitado àqueles que têm autorização. A autenticação, controle de acesso e criptografia são alguns dos subcomponentes da confidencialidade. Se um hacker invadir um sistema Linux e roubar um banco de dados contendo as informações de contato dos clientes, ocorreu um comprometimento da confidencialidade.

A integridade está relacionada à autenticidade das informações. A tecnologia da integridade dos dados assegura que as informações são válidas e não foram alteradas de maneira não-autorizada. Ela também aborda a confiabilidade das fontes de informações. Quando um site seguro apresenta um certificado SSL assinado, ele está provando para o usuário não apenas que as informações que ele envia estão encriptadas, mas também que uma autoridade certificada confiável (como VeriSign ou Equifax) verificou a identidade da fonte. Tecnologias como PGP e Kerberos também garantem a integridade dos dados.

As informações devem estar disponíveis para os usuários autorizados quando eles precisam delas ou não há nenhum propósito em tê-las. Falhas não causadas por invasores, como aquelas provocadas por erros administrativos ou queda de energia, também entram na categoria dos problemas de disponibilidade. Infelizmente, a disponibilidade costuma ser ignorada até algo dar errado.

Considere os princípios CIA sempre que você projeta, implementa ou mantém sistemas Linux. Como o velho provérbio de segurança diz: "segurança é um processo".

Certificações

Esse curso rápido e intensivo sobre o CIA é apenas uma breve introdução a um campo maior da segurança das informações. Grandes corporações costumam contratar muitos funcionários em tempo integral cujo trabalho é vigiar informações. Para ganhar credibilidade nesse campo e manter o conhecimento atualizado, esses profissionais participam de treinamentos e obtêm certificações. Prepare-se para uma "sopa de letras" à medida que analisamos algumas das certificações mais populares.

Uma das certificações de segurança mais amplamente reconhecidas é a CISSP (Certified Information Systems Security Professional). Ela é administrada pelo $(ISC)^2$, o International Information Systems Security Certification Consortium (diga *isso* rapidamente dez vezes!). Um dos principais pontos da CISSP é a noção do $(ISC)^2$ de um "órgão comum de conhecimento" (*common body of knowledge* – CBK), essencialmente um guia das melhores práticas utilizado por toda a indústria para a segurança de informações. O CBK abrange legislação, criptografia, autenticação, segurança física e muito mais. Ele é uma referência incrível para profissionais de segurança.

Uma crítica à CISSP foi o fato de ela ter se concentrado na amplitude e conseqüente falta de profundidade. Tantos tópicos no CBK e tão pouco tempo! Para resolver isso, o $(ISC)^2$ emitiu programas CISSP que focalizam arquitetura, engenharia e gerenciamento. Essas certificações especializadas adicionam profundidade à certificação CISSP mais geral.

O System Administration, Networking, and Security (SANS) Institute criou o conjunto de certificações Global Information Assurance Certification (GIAC) em 1999. Três dezenas separadas de exames abrangem o reino da segurança de informações com testes divididos em cinco categorias. A complexidade dessas certificações varia dos exames GISF moderados de duas partes ao GSE de nível especialista de 23 horas. O GSE tem uma reputação notória de ser uma das certificações mais difíceis na indústria. Muitos dos exames focalizam as especificidades técnicas e exigem bastante experiência.

Por fim, a credencial Certified Information Systems Auditor (CISA) é uma certificação de auditoria e processo. Ele focaliza a continuidade, procedimentos, monitoramento e outro conteúdo do gerenciamento de negócios. Alguns consideram a CISA uma certificação intermediária apropriada ao papel do gerente de segurança de uma empresa. Um dos seus aspectos mais atraentes é o fato de que envolve apenas um único exame.

Embora as certificações sejam um esforço pessoal, suas aplicações aos negócios são inegáveis. Um número cada vez maior de empresas agora reconhece as certificações como a marca de um especialista. Muitas empresas oferecem salários mais altos e promoções aos funcionários certificados. Se decidir obter uma certificação, discuta com a gerência da sua empresa uma forma de ela o ajudar a financiar os custos associados.

Padrões

Devido à confiança cada vez maior nos sistemas de dados, foram criadas leis e regulamentos para administrar o gerenciamento das informações críticas e sigilosas das empresas. Os principais projetos de leis norte-americanos como HIPAA, FISMA e Sarbanes-Oxley Act incluem seções sobre a segurança TI. Embora os requisitos às vezes sejam caros de implementar, eles ajudaram a dar um nível adequado de foco a um aspecto da tecnologia anteriormente ignorado.

Infelizmente, os regulamentos estão cheios de termos jurídicos e podem ser difíceis de interpretar. A maioria deles não contém as especificidades de como alcançar os requisitos. Como resultado, os padrões foram desenvolvidos para ajudar os administradores a entender os rígidos requisitos legislativos. Esses padrões não são específicos aos regulamentos, mas segui-los normalmente assegura a conformidade. Pode ser intimidador confrontar os requisitos de todos os vários padrões de uma vez, mas esses planos são uma diretriz útil.

É provável que o padrão ISO/IEC 17799 seja o mais amplamente aceito no mundo. Primeiramente introduzido em 1995 como um padrão britânico, ele contém 34 páginas e está dividido em 11 seções que vão desde diretivas até segurança física para controle de acesso. Os objetivos dentro de cada seção definem requisitos específicos e os controles sob cada objetivo descrevem as melhores soluções práticas sugeridas.

Os requisitos são não técnicos e podem ser cumpridos por qualquer empresa de uma maneira que melhor atenda às necessidades dessa empresa. A desvantagem é que o texto geral do padrão deixa o leitor com uma sensação de ampla flexibilidade. Os críticos reclamam que a falta de especificidades deixam as empresas vulneráveis a ataques.

Contudo, esse padrão é um dos documentos mais valiosos disponíveis para a indústria da segurança de informações. Ele preenche uma lacuna freqüentemente tangível entre o gerenciamento e a engenharia e ajuda a se concentrar em ambos para minimizar riscos corporativos.

O Payment Card Industry Data Security Standard (PCI DSS) é algo completamente diferente. Ele surgiu da necessidade real de aprimorar a segurança na indústria de processamento de cartões de crédito depois de uma série de exposições drásticas. Por exemplo, em junho de 2005, a CardSystems Services International tornou pública a "perda" de 40 milhões de números de cartões de crédito.

O U.S. Department of Homeland Security estimou que US$ 52.6 bilhões foram perdidos no roubo de identidades apenas em 2004. Naturalmente, nem tudo isso está relacionado diretamente à exposição de números de cartões de crédito, mas melhor vigilância por parte dos fornecedores certamente teria tido um impacto positivo. O FBI até mesmo liga fraudes de cartões de crédito com o financiamento do terrorismo. Incidentes específicos incluem os bombardeios em Bali e no sistema de metrô de Madri.

O padrão PCI DSS é o resultado de um esforço conjunto entre as empresas Visa e Mastercard, embora ele seja atualmente mantido pela Visa. Diferentemente do padrão ISO 17799, a cópia é gratuita. Ele se concentra inteiramente na proteção dos sistemas de dados dos portadores de cartão de crédito e tem 12 seções que definem os requisitos da proteção.

Como o principal objetivo do PCI DSS é o processamento de cartão, ele não é adequado às empresas que não trabalham com dados de cartões de crédito. Entretanto, para aquelas que trabalham, uma rigorosa conformidade é necessária para evitar multas pesadas e possivelmente ações criminais. Você pode encontrar esse documento na seção sobre segurança de transações no site da Visa.

Muitos sites criam softwares personalizados para atender a uma necessidade única, mas a pressa freqüentemente tem um custo para a segurança. A suscetibilidade a overflows de buffer, injeções de SQL e ataques *cross-site scripting* são alguns exemplos das falhas comuns dos softwares personalizados. Felizmente, o governo norte-americano distribuiu um documento procedural para ajudar a validar a segurança desses aplicativos. A *Application Security Checklist* do Departamento de Defesa Norte-americano é um guia completo para testar as fraquezas de segurança de um aplicativo. A versão atual está disponível para cópia em

iase.disa.mil/stigs/checklist

Diferentemente dos outros padrões discutidos nesta seção, a Application Security Checklist discute mais detalhadamente as especificidades técnicas, às vezes abordando comandos individuais que devem ser executados para conduzir um teste particular. Esse padrão não é tão legível e formal quanto os outros padrões aqui, mas é imensamente útil para sites com necessidades de desenvolvimento internas.

20.4 DICAS E FILOSOFIA DE SEGURANÇA

Este capítulo discute uma ampla gama de questões relativas à segurança. Idealmente, você deve executar todas elas dentro de seu ambiente. Entretanto, se não tiver muito tempo ou paciência, eis as seis questões mais importantes a serem consideradas, mais algumas regras extras de acordo com as quais se guiar. (A maioria dos administradores deve realmente digerir o conteúdo deste capítulo todo, provavelmente mais de uma vez.)

Filtragem de pacotes

Se estiver conectando um sistema Linux a uma rede com acesso Internet, você deve *obrigatoriamente* ter um roteador para filtragem de pacotes ou firewall entre o sistema Linux e o mundo externo. Como alternativa, você pode configurar a filtragem de pacotes usando **iptables** no próprio sistema Linux (discutido na página 486). Qualquer que seja a implementação, a filtragem de pacotes só deve permitir tráfego para os serviços que você quer especificamente fornecer ou utilizar a partir do sistema Linux.

Serviços desnecessários

Distribuições Linux diferem muito no que diz respeito aos serviços de rede que são ativados como parte da instalação padrão. A maioria deles inclui vários que não são necessários. Fica a seu critério examinar os serviços que estão ativados em seu sistema e desativar qualquer um deles que não seja absolutamente necessário. Os comandos **netstat** e **fuser** são uma excelente maneira de começar. Para realmente entender quais serviços um invasor pode ver, faça uma varredura dos seus sistemas com um scanner de porta a partir de um host remoto. **nmap**, descrito a partir da página 475, é a realização de um sonho dos profissionais em segurança de obter a varredura de porta.

Correções de software

Todos os principais distribuidores Linux distribuem um fluxo constante de correções relacionadas à segurança de software, normalmente por mês. Você *deve* estar atento aos patches de segurança relevantes para os softwares do seu sistema (e a quaisquer pacotes de softwares que você está executando) e instalá-los imediatamente. Tenha em mente que, depois de um patch estar disponível, os "sujeitos maliciosos" talvez já conheçam a brecha de segurança há semanas.

Backups

Ponha aquele RAID fora de ação e bem longe do CPD, meu caro. Você *deve* realizar backups regularmente de todos os seus sistemas para poder recuperá-los de modo efetivo depois de um incidente de segurança, se ocorrer algum. Nenhum nível de tecnologia de espelhamento, RAID ou "hot standby" elimina a necessidade dos backups. Informações sobre como realizar backups são fornecidas no Capítulo 9, *Backups*.

Senhas

Somos pessoas simples com regras simples. Eis uma delas: toda conta deve ter uma senha e deve ser algo que não pode ser adivinhado facilmente. Nunca é uma boa idéia enviar senhas em texto simples reutilizáveis pela Internet. Se permitir logins remotos ao seu sistema, você deverá utilizar o SSH ou algum outro sistema seguro de acesso remoto (discutido a partir da página 481).

Vigilância

Para garantir a segurança de seu sistema, você deve monitorar regularmente (em geral, todos os dias) sua saúde, conexões de rede, tabela de processos e estado geral. Realize auto-avaliações regulares utilizando as poderosas ferramentas discutidas mais adiante neste capítulo. Os problemas de segurança costumam começar pequenos e crescem rapidamente, de maneira que, quanto antes você identificar uma anomalia, melhor você se sairá.

Filosofia geral

Uma efetiva segurança de sistemas tem suas raízes no senso comum. Algumas regras gerais:

- Não coloque arquivos em seus sistemas que possam ser interessantes para hackers ou funcionários abelhudos. Segredos de negócios, arquivos pessoais, folhas de pagamento, resultados de eleições etc. devem ser tratados com cuidado se estiverem on-line. Proteger essas informações com criptografia dá um grau de segurança bem maior do que simplesmente tentar impedir que usuários desautorizados acessem os arquivos que contêm as guloseimas.
- A política de segurança de sua instalação deve especificar com que nível de sigilo as informações são tratadas. Consulte o Capítulo 30, *Gerenciamento, diretivas e políticas,* e a seção sobre os padrões de segurança neste capítulo (página 467) para algumas sugestões.
- Não forneça lugares para que os hackers residam no seu ambiente. Os hackers normalmente invadem um sistema e depois o utilizam como uma base de operações para entrar em outros. Às vezes, os hackers podem utilizar sua rede para esconder as pegadas enquanto eles atacam um alvo real. Serviços publicamente expostos com vulnerabilidades, diretórios FTP mundialmente graváveis por qualquer pessoa, contas compartilhadas e sistemas negligenciados encorajam atividades ocultas.

- Configure armadilhas para ajudar a detectar invasões e tentativas de invasão. Ferramentas como samhain, **xinetd** e John the Ripper (descritas a partir da página 477) o manterão a par dos potenciais problemas.
- Monitore religiosamente os relatórios gerados por essas ferramentas de segurança. Um problema secundário que você ignora em um relatório pode crescer até se transformar em uma catástrofe na hora em que o próximo relatório for gerado.
- Estude segurança de sistemas. *Know-how* tradicional, preparação dos usuários e bom senso são as partes mais importantes do plano de segurança de uma instalação. Traga especialistas de fora para preencher lacunas, mas apenas sob sua aprovação e acompanhamento de perto.
- Fique na espreita observando atividade não-usual. Investigue qualquer coisa que lhe pareça estranha, como mensagens de log estranhas ou alterações na atividade de uma conta (mais atividade, atividade em horários estranhos ou quem sabe atividade enquanto o dono da conta está de férias).

20.5 PROBLEMAS DE SEGURANÇA EM /ETC/PASSWD E /ETC/SHADOW

Consulte a página 66 para mais informações sobre o arquivo ***passwd****.*

Um controle ineficiente das senhas é um ponto fraco comum em termos de segurança do sistema. Os conteúdos dos arquivos **/etc/passwd** e **/etc/shadow** determinam quem pode efetuar o login, portanto esses arquivos são a primeira linha de defesa do sistema contra invasores. Eles devem ser escrupulosamente mantidos e isentos de erros, riscos de segurança e bagagem histórica.

Nos sistemas de legado, o segundo campo de **/etc/passwd** continha uma string que representava a senha criptografada do usuário. Como **/etc/passwd** deve ser lido por todo o mundo para comandos como **ls** funcionarem, a string de senha criptografada se disponibiliza para todos os usuários do sistema. Malfeitores podiam criptografar dicionários inteiros e comparar os resultados com as strings em **/etc/passwd**. Se as strings criptografadas forem coincidentes, foi localizada uma senha.

Qual o nível de ameaça disso? Nos anos 80, havia pelo menos uma maneira rápida de decifrar senha,[3] mas hackers medianos tinham de se contentar em usar a rotina de biblioteca **crypt**[4] para criptografar palavras de dicionário para comparação. Uma máquina "rápida" dos anos 80 era capaz de realizar algumas centenas de criptografias por segundo. Por comparação, ataques de força bruta agora quebram chaves DES de 56 bits em menos de 24 horas. Felizmente, sistemas UNIX e Linux modernos não utilizam **crypt** e são consideravelmente mais seguros.

Esses resultados são atemorizantes e sugerem que acesso de usuários a strings de senhas criptografadas deve realmente ser restringido. A maneira padrão para impor restrições é colocar senhas em um arquivo separado que possa ser lido somente por root, deixando intacto o restante do arquivo **/etc/passwd**. O arquivo que contém as informações de senha reais é então chamado pelo arquivo de senhas sombras, **/etc/shadow**. Todas as distribuições Linux modernas usam senhas sombras.

Verificação e seleção de senhas

O Linux permite que os usuários escolham suas próprias senhas e embora isso seja uma grande conveniência, também leva a diversos problemas de segurança. Ao fornecer aos usuários seus logins, você também deve orientá-los sobre como escolher uma boa senha. Diga a eles para não usarem seus nomes ou iniciais, o nome do filho ou da esposa ou qualquer outra palavra que possa ser encontrada em um dicionário. As senhas derivadas de dados pessoais como números de telefone ou endereços também podem ser descobertas facilmente.

As senhas devem ter pelo menos oito caracteres de comprimento e devem incluir números, pontuação ou mudanças no uso de letras maiúsculas e minúsculas. Palavras sem sentido, combinações de palavras simples ou as primeiras letras de uma frase memorável formam as melhores senhas. Obviamente, "memorável" é bom, porém "tradicional" é perigoso. Construa a sua própria frase. Os comentários da seção *Escolhendo uma senha para root*, na página 34, também se aplicam a senhas de usuários.

É importante verificar continuamente (de preferência diariamente) se todo login tem uma senha. As entradas no arquivo **/etc/shadow** que descrevem pseudo-usuários como "daemon" que possui arquivos, mas nunca efetua o login devem ter um asterisco (*) ou um ponto de exclamação (!) nos seus campos de senhas criptografadas. O asterisco não coincidirá com nenhuma senha e, portanto, impedirá o emprego da conta.

---

3. Evi Nemeth quebrou o intercâmbio de senhas Diffie-Hellman normalmente usado com DES em 1984, por meio de um supercomputador HEP. Embora se imagine que DES seja matematicamente seguro, os comprimentos curtos das senhas em uso comum oferecem relativamente pouca segurança.
4. Não confunda a rotina de biblioteca **crypt** com o comando **crypt**, que usa um esquema de criptografia diferente e menos seguro.

A maior parte do arquivo **/etc/shadow** não precisa de manutenção. Mas, o seguinte Perl de uma linha verifica senhas nulas:

```
$ sudo perl -F: -ane 'print if not $F[1];' /etc/shadow
```

Um script que realiza essa verificação e envia e-mails para você com os resultados pode ser executado por meio de **cron**. Para ajudar a verificar se modificações nas contas são legítimas, você pode escrever um script que cria **diffs** no arquivo **passwd** contra uma versão a partir do dia anterior e enviar por e-mail quaisquer diferenças a você.

Expiração da senha

O sistema de senhas sombras do Linux pode forçar os usuários a alterar suas senhas periodicamente por meio de um recurso conhecido como expiração de senha. Isso pode, à primeira vista, parecer uma boa idéia, mas apresenta vários problemas. Os usuários muitas vezes se tornam ressentidos em ter de trocar de senha e, já que não querem se esquecer da nova senha, optam por algo simples que seja fácil de digitar e de lembrar. Muitos usuários alternam entre duas senhas cada vez que são forçados a modificá-las, debilitando o propósito do processo de expiração de senhas. Módulos PAM podem ajudar a impor senhas fortes para evitar essa armadilha.

*Veja a página 35 para informações adicionais sobre o **sudo**.*

Não obstante, as senhas devem ser alteradas regularmente, em especial a senha root. Uma senha root deve fluir facilmente de modo que possa ser digitada rapidamente e não seja descoberta por alguém que observe o movimento de seus dedos pelo teclado. Em nossa instalação a maioria das pessoas usa **sudo** em vez da senha root real, porém selecionamos cuidadosamente a senha root da mesma maneira.

O programa **chage** controla a expiração de senhas. Utilizando **chage**, os administradores podem impor tempos mínimos e máximos entre as alterações de senhas, datas de expiração de senha, o número de dias a advertir os usuários antes da expiração das suas senhas, o número de dias de inatividade permitido antes de as contas automaticamente serem bloqueadas e mais. O comando a seguir configura o número mínimo de dias entre alterações de senha como 2, o número máximo como 90, a data de expiração como 31 de julho de 2007 e adverte o usuário por 14 dias de que a data de expiração está se aproximando:

```
$ sudo chage -m 2 -M 90 -E 2007-07-31 -W 14 ben
```

Logins de grupos e logins compartilhados

Qualquer login que seja utilizado por mais de uma pessoa é sinal de más notícias. Logins de grupo (por exemplo, "convidado" ou "demo") com certeza são um terreno fértil para hackers e são proibidos em muitos contextos por regulamentos federais como HIPAA. Não os permita em sua instalação.

Da mesma forma, não permita que usuários compartilhem logins com familiares ou amigos. Se o Joãozinho precisar de um login para poder operar em seu trabalho de ciências, dê um login a ele com esse declarado propósito. É muito mais fácil tirar o login do Joãozinho quando ele abusar dele do que se livrar do papai e de sua conta, especialmente em instalações governamentais.

Na maioria delas, "root" é um login de grupo. Perigo! Recomendamos usar o programa **sudo** para controlar o acesso a poderes de root. Consulte a página 34.

Shells de usuários

Não use um script como o shell para login irrestrito (sem senha). Na realidade, se você precisar de um login sem senha, talvez você deva considerar um par de chaves SSH que não utilize frase de senha.

Entradas de root

O único aspecto que distingue o login root é o seu UID zero. Como pode haver mais de uma entrada no arquivo **/etc/passwd** que use esse UID, pode haver mais de uma maneira de fazer o login como root.

Uma maneira comum para os hackers instalarem uma porta dos fundos assim que tenham obtido um shell root é editar novos logins root em **/etc/passwd**. Programas como **who** e **w** referem-se ao nome armazenado em **/var/run/utmp** em vez do UID que é proprietário do shell de login, de modo que eles não são capazes de expor os hackers que aparentam ser inocentes usuários mas que, na verdade, estão conectados como UID 0.

A defesa contra esse subterfúgio é um miniscript similar àquele utilizado para localizar logins sem senhas:

```
$ perl -F: -ane 'print if not $F[2];' /etc/passwd
```

Esse script imprime quaisquer linhas contidas no arquivo **passwd** que possuam UIDs nulos ou 0. Você poderia adaptá-lo facilmente para localizar entradas com grupos suspeitos ou UIDs que são os mesmos daqueles de pessoas-chave dentro de sua empresa.

Você também deveria verificar a existência de entradas **passwd** que não possuam nome de usuário ou que tenham pontuação como nome de usuário. Essas entradas podem parecer sem sentido, mas elas normalmente permitirão que um hacker entre no sistema.

PAM: o módulo de autenticação

O Pluggable Authentication Modules (conhecido com PAM) foi originalmente inventado pela Sun como uma forma flexível de autenticar usuários. Por vários anos, a autenticação no ambiente UNIX era mera questão de associar usuários a suas entradas no arquivo **/etc/passwd**. A necessidade de uma maior segurança e de suporte para uma ampla gama de mecanismos de autenticação (como cartões inteligentes — "smart cards", por exemplo) criou a necessidade de uma abordagem mais flexível. Certos módulos PAM LDAP realizam autenticação centralizada para autenticação global de diretórios.

O PAM no Linux acompanha todas as distribuições sãs do Linux e não está relacionado à implementação atual da Sun do padrão PAM. O conceito é simples: os programas que exigem autenticação só precisam saber que um módulo está disponível a fim de realizar a autenticação para eles. O PAM é configurado de tal forma que esses módulos possam ser acrescentados, eliminados e reconfigurados a qualquer momento — não é necessário que os módulos sejam linkados (ou até mesmo que existam) no momento em que um utilitário é compilado. Como resultado dessa arquitetura, o PAM se tornou uma ferramenta incrivelmente poderosa para administradores de sistemas.

Dezenas de módulos PAM estão disponíveis. Você pode fazer uma cópia de módulos especializados e respectiva documentação em www.kernel.org/pub/linux/libs/pam.

Os módulos PAM são configurados por meio de arquivos no diretório **/etc/pam.d**. Arquivos por serviço nesse diretório contêm entradas no formato

```
tipo-do-módulo    flags-de-controle caminho-para-módulo argumentos
```

O campo *tipo-do-módulo* pode assumir os valores `auth`, `account`, `session` ou `password`. Uma entrada `auth` estabelece quem é o usuário e possivelmente concede associação de grupo. O tag `account` toma decisões não baseadas em autenticação, como acesso baseado em horários. Tarefas que precisam ser realizadas antes ou depois de um usuário ter recebido um serviço são implementadas com o tag `session`. Finalmente, o tag `password` é utilizado quando informações de autenticação (como, por exemplo, uma senha) são solicitadas ao usuário.

O campo *flags-de-controle* possui quatro valores possíveis: `required`, `requisite`, `sufficient` e `optional`. `required` e `optional` são mais comumente utilizados, o que significa, respectivamente, que um módulo deve ser bem-sucedido para que a execução prossiga ou então que não importa se o módulo foi ou não bem-sucedido.

O terceiro e o quarto campos são o nome de caminho e os argumentos para o objeto de módulo carregável dinamicamente. Se o primeiro caractere do caminho for /, pressupõe-se que ele é um caminho absoluto. Caso contrário, o conteúdo do campo é anexado ao caminho padrão, **/lib/security**.

Consulte a página 477 para informações adicionais sobre John the Ripper.

PAM é uma solução às complexas dificuldades das senhas descritas anteriormente. O módulo **pam_cracklib** pode fazer com que as senhas cumpram requisitos mínimos. As especificidades variam bastante, portanto utilize **grep** para localizar o arquivo de configuração adequado. Por exemplo, para assegurar que as senhas dos usuários não possam ser determinadas pelo John the Ripper, o arquivo **/etc/pam.d/system-auth** no Fedora deve conter:

```
password required pam_cracklib.so retry=3 minlen=12 difok=4
```

Com essa linha na posição correta, o PAM verifica novas senhas propostas pelos usuários em um dicionário de cracking e de conjuntos de regras de senhas. (Essa configuração requer a presença da biblioteca de sistema **libcrack** e também de um dicionário de sistema, **/usr/lib/cracklib_dict.***) Uma mensagem de erro como "A senha é muito simples" é impressa na tela se a senha de um usuário não obedecer aos requisitos **cracklib**.

As regras de argumento **cracklib** são complicadas, mas eis a interpretação da configuração específica mostrada acima:

- O argumento `retry=3` especifica que o usuário terá três chances de introduzir uma senha forte.
- `minlen=12` especifica um comprimento mínimo de senha. As letras maiúsculas, números e pontuação recebem tratamento especial pela biblioteca e reduzem o mínimo. Com `minlen=12`, a senha mais curta que um usuário pode realmente ter são 8 caracteres, não 12, mas o usuário deve incluir todos os quatro tipos de caracteres disponíveis para configurar uma senha de 8 caracteres.
- `difok=4` especifica que pelo menos quatro caracteres da nova senha não devem estar presentes na senha antiga.

Distribuições Linux modernas incluem e utilizam o módulo **pam_cracklib** por padrão, mas as regras de complexidade de senha normalmente não são permitidas.

20.6 CAPACIDADES DO POSIX

Os sistemas Linux subdividem os poderes da conta root de acordo com a noção de "capacidades" do POSIX. Por exemplo, uma capacidade permite que um processo realize operações privilegiadas nos soquetes de rede e nas interfaces de rede, enquanto um outro permite que um processo configure opções relacionadas ao hardware. As capacidades são herdadas, mascaradas e utilizadas de uma maneira sistemática por meio das quais os programas podem realizar tarefas privilegiadas sem acessar todo o poder da conta root. Como os privilégios podem ser distribuídos mais rigorosamente do que no UNIX tradicional, a probabilidade de um comprometimento à segurança que possa levar a acesso de root irrestrito é menor — pelo menos teoricamente.

Na prática, os administradores de sistema têm interesse limitado no sistema de capacidades. Softwares cientes das capacidades não exigem nenhuma atenção administrativa especial; eles simplesmente se configuram para que operem em um modo mais restrito. Mas a maioria dos softwares UNIX e Linux continua a assumir o tradicional modelo todo-poderoso de superusuário que na verdade não pode ser desativado sem sacrifícios e problemas significativos. Algumas acomodações que o kernel do Linux faz para manter a compatibilidade com os softwares UNIX tradicionais tornam o sistema de capacidades mais "frágil" do que realmente deveria ser.[5]

O resultado é que as capacidades do POSIX são uma boa vitrine, mas têm implicações reais limitadas. No geral, elas podem ser ignoradas, com uma exceção notável: o kernel aplica um "conjunto de vinculação de capacidades globais" (acessado e configurado por meio do arquivo **/proc/sys/kernel/cap-bound**) toda vez que um novo programa é executado. Se quiser desativar uma ou mais capacidades por todo o sistema, você poderá adicioná-las a esse conjunto de vinculação global. Consulte **man capabilities** para detalhes.

20.7 PROGRAMAS SETUID

Programas que executam setuid, especialmente os que executam setuid em root, são propensos a problemas de segurança. Os comandos setuid distribuídos com o Linux são teoricamente seguros; entretanto, brechas de segurança foram descobertas no passado e indubitavelmente serão descobertas no futuro.

A maneira mais garantida de minimizar o número de *problemas* relacionados com setuid é minimizar o número de *programas* setuid. Pense duas vezes antes de instalar um pacote de software que precise executar setuid e evite usar o recurso setuid em seu software caseiro. Uma distribuição normal do Linux contém aproximadamente 35 programas setuid, embora isso varie entre distribuições.

Não há nenhuma regra que diga que os programas setuid tenham de ser executados como root. Se tudo o que você precisa fazer é restringir o acesso a um determinado arquivo ou banco de dados, você poderá adicionar um pseudo-usuário ao arquivo **passwd** cuja única razão para existência é ter como propriedade os recursos restritos. Siga as convenções de pseudo-usuários normais: utilize um UID baixo, coloque um asterisco no campo de senha e faça com que o diretório de origem do pseudo-usuário seja **/dev/null**.

Você pode desativar a execução de setuid e de setgid nos sistemas de arquivos individuais especificando a opção **-o nosuid** como **mount**. É uma boa idéia usar essa opção em sistemas de arquivos que contenham diretórios iniciais de usuários ou que sejam montados por meio de domínios administrativos menos confiáveis.

É útil fazer uma varredura periódica de seus discos em busca de novos programas setuid. Um hacker que rompeu a segurança de seu sistema cria algumas vezes um utilitário ou shell setuid privado para facilitar visitas repetidas. Algumas dessas ferramentas discutidas a partir da página 475 localizam esses arquivos, mas você pode fazer o mesmo com **find**. Por exemplo,

```
/usr/bin/find / -user root -perm -4000 -print | /bin/mail -s "Setuid root files" netadmin
```

enviará uma lista de todos os arquivos setuid root para o usuário "netadmin".

20.8 PERMISSÕES DE ARQUIVOS IMPORTANTES

Muitos arquivos em um sistema Linux precisam ter permissões específicas se os problemas de segurança devem ser evitados. Alguns distribuidores enviam seu software com permissões configuradas para seus próprios ambientes de desenvolvimento "amigáveis". Essas permissões talvez não sejam apropriadas para o seu caso.

---

5. Um dos nossos revisores técnicos comentou: "de fato, a herança de capacidades nunca funcionou corretamente no Linux; é uma complicação constante".

/etc/passwd e **/etc/group** não devem ser mundialmente graváveis. Eles devem ter root para o proprietário e modo 644. **/etc/shadow** deve ter modo 600 — sem permissões para o grupo ou para o "mundo".

```
-rw-------    1 root     root        1835 May  8 08:07 /etc/shadow
```

Os grupos de todos esses arquivos devem ser configurados para algum grupo de sistema, normalmente root. (O comando **passwd** executa setuid para root de modo que os usuários possam alterar suas senhas sem ter permissão de gravação sobre **/etc/passwd** ou **/etc/shadow**.)

Consulte a página 506 informações adicionais sobre como configurar um servidor FTP.

Historicamente, programas FTP estavam infestados de brechas de segurança e configurações padrão inseguras. O FTP também transmite credenciais em texto claro, um problema inerente ao protocolo que o torna inaceitável na Internet de hoje em dia.

Novas tecnologias como o SFTP substituíram o FTP e há muito poucos casos em que o FTP deve ser permitido. Nesses casos raros, porém, leia a documentação de software completamente para assegurar uma configuração adequada. Se precisar permitir acesso FTP anônimo, desative diretórios mundialmente graváveis ou faça uma varredura neles regularmente para encontrar arquivos ilegais ou sigilosos.

Arquivos de dispositivos para partições de disco rígido são uma outra fonte potencial para problemas. Ter permissão de leitura ou gravação em um arquivo de dispositivo de disco é basicamente o mesmo que ter permissão de leitura ou gravação em todo arquivo no sistema de arquivos que ele representa. Somente root deve ter tanto permissão de leitura como de gravação. O proprietário do grupo algumas vezes recebe permissão de leitura para facilitar os backups, mas não deve haver permissões para o "mundo".

20.9 PROBLEMAS DE SEGURANÇA DIVERSOS

As seções a seguir apresentam alguns tópicos variados relacionados à segurança. A maioria cai na categoria de recursos que são úteis a você como administrador de sistemas ou recursos impróprios que podem fornecer material para instalação de hackers se não forem verificados constantemente.

Registro em log de eventos remotos

Consulte o Capítulo 10 para informações adicionais sobre syslog.

O recurso syslog permite que informações de log de kernel, bem como de processos de usuário, possam ser encaminhadas a um arquivo, uma lista de usuários ou um outro host de sua rede. Pense em configurar um host seguro que atua como uma máquina central de login, analise sintaticamente os arquivos de log e envie por e-mail eventos interessantes. Essa precaução impedirá que hackers encubram seus rastros regravando ou apagando arquivos de log.

Terminais seguros

Alguns sistemas Linux podem ser configurados para restringir logins root a terminais "seguros" específicos. Uma boa idéia é desativar logins para root em canais como SSH, VPNs ou em outros links remotos. Freqüentemente, pseudoterminais de rede são configurados para desativar login root.

Os canais seguros são especificados como uma lista de dispositivos TTY no arquivo de configuração **/etc/securetty**. Também é possível restringir logins não-root a determinadas localizações com entradas no arquivo **/etc/security/access.conf** ou para horários particulares com entradas no arquivo **/etc/security/time.conf**.

Os administradores ainda podem efetuar o login normalmente e utilizar **sudo** para acessar os poderes de superusuário.

/etc/hosts.equiv e ~/.rhosts

Os arquivos **hosts.equiv** e **~/.rhosts** definem hosts como administrativamente "equivalentes" uns aos outros, permitindo aos usuários fazerem o login (por meio de **rlogin**) e copiar arquivos (com **rcp**) entre máquinas sem digitar suas senhas. O uso desse recurso já foi comum durante os tempos de festa do UNIX, mas todo o mundo acabava acordando com uma horrível dor de cabeça e se dava conta de que não era uma boa idéia.

Felizmente, o protocolo SSH eliminou, quase de todo, o uso de equivalentes inseguros como **telnet**, **rsh** e **rlogin**. O uso desse protocolo é abrangido mais adiante neste capítulo.

Algumas das substituições para **rlogin** (inclusive {SSH!) prestam atenção a **.rhosts** e a **/etc/hosts.equiv** se eles não estiverem configurados apropriadamente. Para maior segurança, você poderá criar o arquivo **/etc/hosts.equiv** e um arquivo **~/.rhosts** para cada usuário (inclusive root) como um arquivo de comprimento zero e não-gravável. É mais fácil verificar qual era o estado de um arquivo às 3 da manhã se ele existir e não for tocado do que verificar o estado de um arquivo inexistente. Essa distinção pode ser crucial quando você estiver rastreando intrusos e suas tentativas de comprometer seu sistema.

Segurança e NIS

Consulte o Capítulo 17 para informações adicionais sobre NIS.

Além do título desta seção, essas palavras nunca devem ser utilizadas juntas. O Network Information Service (NIS, anteriormente denominado Yellow Pages) é uma ferramenta de distribuição de banco de dados da Sun que muitas instalações utilizam para manter e distribuir arquivos como **/etc/group**, **/etc/passwd** e **/etc/hosts**. Infelizmente, sua natureza intrínseca de "fácil acesso a informações" o transforma em uma apetitosa isca para hackers.

Uma maneira mais segura e confiável de distribuir esses arquivos é criar um login de serviço como "netadmin" e colocar as cópias mais recentes destes arquivos em **~netadmin**. Você então pode utilizar **cron** para executar um script em cada máquina do cliente para realizar um **scp**, verificações de sanidade e instalar os arquivos. Consulte a página 481 para informações adicionais sobre SSH, do qual **scp** faz parte.

Segurança e NFS

O NFSv4 é uma extensão IETF para o protocolo inicial da Sun que inclui segurança forte e alguns outros benefícios em relação às primeiras implementações. Embora esse protocolo ainda não esteja completo, o desenvolvimento está bem avançado e o NFSv4 é incluído na série do kernel 2.6 do Linux.

Versões mais antigas do NFS utilizam um modelo de segurança fraco. Veja a página 339 para mais informações sobre segurança NFS. Você pode usar **showmount -e** para verificar que sistemas de arquivos estão sendo exportados e para quem. Cada sistema de arquivos exportado deve ter uma lista de acesso, e todos os nomes de hosts devem ser totalmente qualificados.

Segurança e sendmail

Consulte o Capítulo 18 para informações adicionais sobre sendmail.

sendmail é um programa enorme que executa como root, pelo menos inicialmente. Como resultado, ele freqüentemente esteve sujeito a ataques de hackers. Certifique-se de que esteja usando a versão mais atualizada de **sendmail** em todos os seus sistemas. Como problemas de segurança são uma das questões mais prováveis a incitar novas versões de software, é provável que todas as versões de **sendmail** exceto a mais atual tenham vulnerabilidades.

Detalhes específicos sobre a segurança **sendmail** são abrangidos no capítulo sobre e-mail a partir da página 417; www.sendmail.org contém informações sobre versões específicas.

Segurança e backups

Consulte o Capítulo 9 para informações adicionais sobre backups.

Backups regulares do sistema são uma parte essencial do plano de segurança de qualquer instalação. Certifique-se de que todas as partições sejam regularmente colocadas em fita e que alguns backups sejam armazenados fora da instalação. Se ocorrer um incidente significativo de segurança, você terá um ponto de verificação não contaminado para poder restaurar.

Backups também podem ser uma ameaça à segurança. Uma coleção roubada de fitas pode superar o restante da segurança do sistema. Ao armazenar fitas fora do escritório, utilize um cofre à prova de fogo a fim de evitar roubo. Considere o uso da criptografia. Se você estiver pensando em utilizar uma instalação de armazenamento comercial, solicite um *tour* físico.

Vírus e worms

O Linux tem sido em grande parte imune a vírus. Há apenas alguns (a maioria dos quais é, por natureza, acadêmica) e nenhum causou grandes danos que tenha se tornado corriqueiro no mundo Windows. Contudo, isso não impediu que certos fornecedores de antivírus previssem a morte da plataforma Linux por malware — a menos que você compre seus produtos de antivírus a um preço introdutório especial, naturalmente.

A razão exata da ausência de softwares maliciosos não é clara. Alguns dizem que o Linux simplesmente tem uma fatia de mercado menor que seus concorrentes desktop e, portanto, não é um alvo interessante para os criadores de vírus. Outros insistem que o ambiente de controle de acesso do Linux limita a disseminação de danos por worms ou vírus autopropagadores.

O último argumento é parcialmente válido. Como o Linux restringe acesso de gravação aos executáveis de sistema no nível do sistema de arquivos, as contas de usuário não privilegiados não conseguem infectar o restante do ambiente. A menos que o código do vírus seja executado pelo root, o alcance da infecção é significativamente limitado. A principal recomendação é, portanto, não utilizar a conta root para atividades rotineiras.

Talvez de maneira contrária à intuição, uma razão válida para utilizar um software antivírus nos servidores Linux seja proteger o sistema Windows da sua instalação contra vírus específicos ao Windows. Um servidor de correio pode varrer vírus em anexos de e-mail recebido e um servidor de arquivos pode varrer infecções nos arquivos compartilhados. Mas essa solução deve suplementar a proteção de antivírus da área de trabalho em vez de substituí-la.

ClamAV, de Tomasz Kojm, é um produto de antivírus popular, gratuito para o Linux. Essa ferramenta GPL amplamente utilizada é um conjunto completo de ferramentas de antivírus com assinaturas para milhares de vírus. Você pode fazer a cópia da versão mais recente em www.clamav.net.

Cavalos de Tróia

Cavalos de Tróia são programas que não são o que parecem. Um exemplo de um cavalo de Tróia foi um programa chamado **turkey** distribuído na Usenet há muito tempo. O programa dizia que desenharia a figura de um peru na sua tela, mas na verdade ele apagava arquivos de seu diretório de usuário.

Um exemplo mais controverso de um cavalo de Tróia foi o software de proteção de cópias incluído em muitos CDs de áudio da Sony em 2004 e 2005. Em uma tentativa confusa de impedir que pessoas compartilhem músicas, a Sony instalou esse software de proteção nos sistemas Windows sem o consentimento ou conhecimento dos ouvintes. O software introduziu vulnerabilidades no computador host que poderiam ser exploradas por worms e vírus.

De vez em quando, fragmentos troianos aparecem nos principais pacotes de software do Linux. **sendmail**, **tcpdump**, OpenSSH e InterBase publicam relatórios sobre softwares maliciosos nos seus produtos. Esses troianos em geral embutem o código malicioso que permite que os invasores acessem os sistemas da vítima à vontade. Felizmente, a maioria dos fornecedores corrige o software e publica um relatório no prazo de uma ou duas semanas. Não deixe de verificar nas listas de discussão sobre segurança quaisquer pacotes de software de rede que você executa nos seus hosts Linux.

Dado o número de aventuras relacionadas à segurança que a comunidade Linux vem assistindo ao longo dos últimos anos, é impressionante o pequeno número de incidentes relacionados a cavalos de Tróia. O crédito para esse estado de coisas se deve, em grande parte, à comunidade Internet. Problemas de segurança óbvios tendem a ser descobertos rapidamente e discutidos amplamente. Pacotes mal-intencionados não ficam disponíveis por muito tempo em servidores Internet bem conhecidos.

Pode estar certo de que qualquer software que tenha sido descoberto como sendo malicioso provocará um grande rebuliço na Internet. Se você quiser fazer uma rápida verificação antes de instalar algo, digite o nome do pacote de software em seu mecanismo de busca favorito.

Rootkits

Os hackers mais astutos tentam esconder suas pegadas e evitar detecção. Freqüentemente, eles esperam continuar utilizando seu sistema para distribuir software ilegalmente, investigar outras redes ou lançar ataques contra outros sistemas. Eles freqüentemente utilizam "rootkits" para ajudá-los a permanecer indetectáveis. O cavalo de Tróia da Sony empregou capacidades do tipo rootkit para se auto-ocultar dos usuários.

Rootkits são programas e patches que ocultam informações importantes do sistema como processos, discos ou atividades de rede. Eles têm diferentes versões e variam quanto à sofisticação de substituições simples de aplicativo (como versões hackeadas do **ls** e **ps**) a módulos de kernel que são quase impossíveis de detectar.

Software de detecção de invasão baseado em host como samhain (descrito a seguir) é uma maneira efetiva de monitorar nos sistemas a presença de rootkits. Embora haja programas disponíveis para ajudar os administradores a remover rootkits de um sistema comprometido, o tempo que leva para realizar uma limpeza completa seria mais bem gasto salvando dados, reformatando o disco e começando a partir do zero. Os rootkits mais avançados conhecem os programas comuns de remoção e tentam subvertê-los.

20.10 FERRAMENTAS DE SEGURANÇA PODEROSAS

Algumas das tarefas demoradas mencionadas nas seções anteriores podem ser automatizadas com ferramentas gratuitas. Eis algumas delas que você deve analisar.

Nmap: varrendo portas de rede

nmap é um scanner de portas de rede. Sua principal função é verificar um conjunto de hosts de destinos para ver quais portas TCP e UDP possuem servidores escutando nelas.[6] A maioria dos serviços de rede está associada a números de portas "bem conhecidas"; essas informações dizem muito sobre o software que uma máquina está executando.

Executar o Nmap é uma excelente maneira de descobrir como um sistema parece para alguém que está tentando invadi-lo. Por exemplo, eis um relatório de uma instalação padrão do Red Hat Enterprise Linux:

```
$ nmap -sT rhel.booklab.example.com
```

6. Conforme descrito no Capítulo 12, uma porta é um canal de comunicações numerado. Um endereço IP identifica uma máquina inteira e um endereço IP + um número de porta identifica um servidor específico ou uma conversa de rede nessa máquina.

```
Starting Nmap 4.00 ( http://www.insecure.org/nmap/ ) at 2006-05-15 23:48 EDT
Interesting ports on rhel.booklab.example.com (192.168.1.31):
(The 1668 ports scanned but not shown below are in state: closed)
PORT       STATE  SERVICE
22/tcp     open   ssh
25/tcp     open   smtp
111/tcp    open   rpcbind
631/tcp    open   ipp

Nmap run completed -- 1 IP address (1 host up) scanned in .158 second
```

Por padrão, **nmap** inclui o argumento **-sT** para tentar se conectar a cada porta TCP do host de destino segundo a maneira tradicional.[7] Assim que uma conexão tiver sido estabelecida, **nmap** se desconecta imediatamente, o que não é educado, mas não é prejudicial a um servidor de rede configurado de forma apropriada.

A partir do exemplo anterior, podemos ver que o host rhel está executando vários serviços que provavelmente não são utilizados e historicamente associados com problemas de segurança: **portmap** (`rpcbind`), CUPS (`ipp`) e provavelmente **sendmail** (`smtp`). Várias linhas potenciais de ataque estão agora aparentes.

A coluna `STATE` na saída do **nmap** mostra `open` para portas com servidores, `unfiltered` para portas em um estado desconhecido, `closed` para portas sem servidor e `filtered` para portas que não podem ser investigadas por causa de um filtro de pacotes. As portas não filtradas são o caso típico e normalmente não são mostradas a menos que **nmap** esteja executando uma varredura ACK. Por exemplo, eis um dump de um servidor mais seguro, secure.example.com:

```
$ nmap -sT secure.example.com

Starting Nmap 4.00 ( http://www.insecure.org/nmap/ ) at 2006-05-15 23:55 EDT
Interesting ports on secure.example.com (192.168.1.33):
(The 1670 ports scanned but not shown below are in state: closed)
PORT       STATE  SERVICE
22/tcp     open   ssh
1241/tcp   open   nessus

Nmap finished: 1 IP address (1 host up) scanned in 0.143 seconds
```

Nesse caso, fica claro que o host foi configurado para permitir SSH e o scanner de rede Nessus. Um firewall bloqueia o acesso a outras portas.

Além de sondagens diretas em TCP e UDP, **nmap** também possui um repertório de maneiras dissimuladas de sondar portas sem realmente iniciar uma conexão. Na maioria dos casos, **nmap** investiga pacotes que parecem provenientes do meio de uma conversa TCP (e não do início) e aguarda que sejam enviados de volta pacotes de diagnóstico. Essas sondagens indetectáveis podem ser capazes de passar por um firewall ou de evitar a detecção por um monitor de segurança de rede na busca por varreduras de portas. Se a sua instalação usa um firewall (veja a seção *Firewalls* na página 484), é uma boa idéia sondá-lo com esses modos de varredura alternativos para ver o que acontece.

nmap possui a capacidade mágica e útil de adivinhar que sistema operacional está sendo utilizado por um sistema remoto analisando os detalhes de sua implementação TCP/IP. Às vezes, ele pode até identificar o software que está em execução em uma porta aberta. As opções **-O** e **-sV**, respectivamente, ativam esse comportamento. Por exemplo:

```
$ nmap -O -sV secure.example.com

Starting Nmap 4.00 ( http://www.insecure.org/nmap/ ) at 2006-05-16 00:01 EDT
Interesting ports on secure.example.com (192.168.1.33):
(The 1670 ports scanned but not shown below are in state: closed)
PORT       STATE  SERVICE VERSION
22/tcp     open   ssh     OpenSSH 3.6.1p2 (protocol 2.0)
1241/tcp   open   ssl     Nessus security scanner
Device type: general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
OS details: Linux 2.4.0 - 2.5.20, Linux 2.5.25 - 2.6.8 or Gentoo 1.2 Linux 2.4.19 rc1-rc7, Linux
    2.6.3 - 2.6.10
Nmap finished: 1 IP address (1 host up) scanned in 8.095 seconds
```

7. Na realidade, apenas as portas privilegiadas (aquelas com números abaixo de 1.024) e as portas bem conhecidas são verificadas por padrão. Utilize a opção **-p** para especificar explicitamente o intervalo das portas a ser varrido.

Esse recurso pode ser muito útil para extrair um inventário de uma rede local. Infelizmente, ele também é muito útil para os hackers que podem basear seus ataques nas fraquezas conhecidas do SO alvo.

Tenha em mente que a maioria dos administradores não gosta que você faça uma varredura nas redes deles e indique vulnerabilidades, por mais bem-intencionado que isso possa ser. Não execute **nmap** na rede de uma outra pessoa sem a permissão de um dos administradores dessa rede.

Nessus: o scanner de redes de última geração

O Nessus, originalmente distribuído por Renaud Deraison em 1998, é um scanner poderoso e útil para vulnerabilidades de software. No momento em que escrevíamos este livro, ele utilizava mais de 10.000 plug-ins para verificar falhas de segurança remotas e locais. Embora agora seja de código-fonte fechado, um produto proprietário, ele continua livremente disponível e novos plug-ins são distribuídos regularmente. Ele é o scanner para vulnerabilidade disponível mais amplamente aceito e completo.

O Nessus se orgulha de ser o scanner de segurança que não deixa passar nada. Em vez de, por exemplo, supor que todos os servidores Web são executados na porta 80, ele faz varredura em busca de servidores Web sendo executados em qualquer porta e depois os verifica em termos de vulnerabilidades. Em vez de contar com o número de versão informado pelo serviço ao qual se conectou, o Nessus tenta explorar vulnerabilidades conhecidas para ver se o serviço é suscetível.

Embora uma quantidade substancial de tempo de configuração seja necessária para colocar o Nessus em funcionamento (ele requer vários pacotes que não estão instalados na distribuição padrão típica), vale muito bem o esforço.

O sistema Nessus inclui um cliente e um servidor. O servidor atua como um banco de dados, e o cliente trata da apresentação da interface gráfica com usuário. Você deve rodar o servidor em um sistema UNIX ou Linux, mas há clientes para controlar e exibir o Nessus em várias outras plataformas.

Uma das grandes vantagens do Nessus é o seu desenho modular, que facilita o trabalho de terceiros em adicionar novas verificações de segurança. Graças a uma comunidade ativa de usuários (e apesar das alterações recentes na licença), é provável que o Nessus seja uma ferramenta útil por vários anos.

John the Ripper: localizando senhas inseguras

Uma maneira de evitar escolhas ruins de senhas é você mesmo tentar quebrar as senhas e forçar os usuários a alterar a senha que você quebrou. John the Ripper é uma ferramenta sofisticada da Solar Designer que implementa vários algoritmos de quebra de senha em uma única ferramenta. Ele substitui a ferramenta **crack**, discutida nas edições anteriores deste livro.

Mesmo que a maioria dos sistemas utilize um arquivo de senha sombra para ocultar senhas criptografadas da visualização pública, continua sendo inteligente verificar se as senhas dos seus usuários são resistentes a quebras. O conhecimento da senha de um usuário pode ser útil, pois as pessoas tendem a usar a mesma senha repetidamente. Uma única senha pode fornecer acesso a um outro sistema, decifrar arquivos armazenados no diretório de um usuário e permitir acesso a contas financeiras na Web. (Não é preciso dizer que não é muito inteligente em termos de segurança reutilizar uma senha dessa forma. Mas ninguém quer ficar se lembrando de dez senhas.)

Considerando sua complexidade interna, John the Ripper é um programa extremamente simples de usar. Direcione **john** para o arquivo a ser quebrado, mais freqüentemente **/etc/shadow** e observe a mágica que acontece:

```
$ sudo ./john /etc/shadow
Loaded 25 password hashes with 25 different salts (FreeBSD MD5 [32/32])
password   (bad2)
badpass    (bad1)
```

Neste exemplo, 25 senhas exclusivas foram lidas a partir do arquivo de sombra. À medida que as senhas são quebradas, John as imprime na tela e as salva em um arquivo chamado **john.pot**. A saída contém a senha na coluna esquerda com o login entre parênteses na coluna direita. Para reimprimir senhas depois de **john** completar, execute o mesmo comando com o argumento **-show**.

Quando escrevíamos este livro, a versão estável mais recente do John the Ripper era a 1.7.0.2. Ela está disponível em www.openwall.com/john. Como a saída do John the Ripper contém as senhas que ele quebrou, você deve proteger cuidadosamente a saída e excluí-la assim que terminar.

hosts_access: controlando acesso de hosts não-autorizados

Firewalls de rede são uma primeira linha de defesa contra acesso por hosts não-autorizados, mas eles não devem ser a única barreira. O Linux utiliza dois arquivos, **/etc/hosts.allow** e **/etc/hosts.deny**, para restringir acesso aos serviços de acordo com a origem das requisições de rede. O arquivo **hosts.allow** lista os hosts que têm permissão de conectar a um serviço

específico e o arquivo **hosts.deny** restringe o acesso. Entretanto, esses arquivos só controlam o acesso a serviços que estão cientes de **hosts_access**, como aqueles gerenciados por **xinetd**, **sshd** e algumas configurações do **sendmail**.

Na maioria dos casos é prudente ser restritivo e permitir acesso somente aos serviços essenciais a partir dos hosts designados. Sugerimos negar acesso por padrão no arquivo **hosts.deny** com a única linha:

```
ALL:ALL
```

Você pode então permitir acesso com base em casos individuais em **hosts.allow**. A configuração a seguir permite acesso a SSH a partir dos hosts nas redes 192.168/16 e a **sendmail** a partir de qualquer lugar.

```
sshd: 192.168.0.0/255.255.0.0
sendmail: ALL
```

O formato de uma entrada em um desses arquivos é *serviço: host* ou *serviço: rede*. Tentativas de conexão falhas são anotadas no syslog. Conexões de hosts que não têm permissão de acessar o serviço são imediatamente fechadas.

A maioria das distribuições Linux inclui os arquivos **hosts.allow** e **hosts.deny** por padrão, mas, na maioria das vezes, eles permanecem vazios.

Samhain: detecção de invasão baseada em host

O Samhain é um sistema de detecção de invasão baseada em host e monitor de integridade de arquivo desenvolvido e mantido por Rainer Wichmann na Samhain Labs (la-samhna.de). Graças à sua interface modular, ele pode monitorar a integridade dos arquivos, verificar os sistemas de arquivos montados, procurar rootkits e mais. Por exemplo, samhain facilita determinar se um intruso substituiu sua cópia de **/bin/login** por uma que registra senhas em um arquivo clandestino.

O projeto centrado em console do Samhain mantém os dados de log em um host confiável, o que ajuda a manter a integridade da auditoria dos dados. Os relatórios da alteração no host podem ser revisados por meio de uma interface Web ou distribuídos via e-mail. Os logs do Samhain podem ser úteis como uma ferramenta forense para ajudar a reconstruir a seqüência de eventos em um incidente de segurança.

O Samhain verifica as características do host contra um banco de dados que contém informações de arquivos e somas de verificação conhecidas como sensatas no momento em que o banco de dados foi criado. A idéia geral é criar um banco de dados de base a partir de um estado confiável do sistema e então verificar regularmente as diferenças contra o banco de dados histórico. Os arquivos que reconhecidamente mudam sob circunstâncias normais não podem ser configurados para gerar alertas. Quando a configuração de um sistema muda ou um novo software é instalado, o banco de dados deve ser reconstruído para que problemas reais não desapareçam entre uma inundação de avisos espúrios.

O sistema completo inclui três componentes. O agente **samhain** executa em clientes e gera relatórios dos dados para o servidor de log. O daemon do servidor de log, chamado **yule**, aceita conexões dos clientes **samhain** e adiciona dados aos recursos de registro em log configurados, normalmente syslog ou um banco de dados. Um console de gerenciamento baseado na Web chamado Beltane gerencia as linhas de base do sistema, edita configurações e assina as configurações e linhas base com uma assinatura digital.

O projeto baseado em componentes desse sistema oferece vários benefícios em relação aos tradicionais scanners de integridade de arquivo. Os dados que são coletados residem em um host confiável, assim os invasores têm menos chances de comprometer a integridade do banco de dados. O gerenciamento também é mais fácil, pois as informações são coletadas em uma localização central.

Você pode configurar o Samhain para ser incrivelmente paranóico. O Samhain pode fazer coisas furtivas como se ocultar na lista de processos e acrescentar seu banco de dados a um arquivo que não quebra com excesso de dados como um arquivo JPEG, ocultando efetivamente o banco de dados dentro de uma imagem. Assinaturas criptográficas dos arquivos de configuração e dos bancos de dados podem ser criadas com a interface Web para ajudar a detectar quaisquer alterações não-autorizadas nesses arquivos.

Eis um exemplo simples de um relatório de uma entrada no syslog do Samhain:

```
# CRIT   :  [2006-06-25T19:31:48-0600] msg=<POLICY [ReadOnly] --------T->, path=</bin/login>,
    ctime_old=<[2006-06-26T01:24:34]>, ctime_new=<[2006-06-26T01:31:47]>, mtime_old=<[2006-06-
    26T01:24:34]>, mtime_new=<[2006-06-26T01:31:47]>,
    48CA06CC50B857DE77C27956ADE7245B0DF63F6A8A42F5B7
```

Ela é um pouco obscura, mas essa entrada explica que uma mensagem de log da severidade CRITICAL ocorreu e que os atributos ctime e mtime do programa **/bin/login** mudaram. Hora de investigar.

Infelizmente, o Samhain requer uma quantidade extra considerável de manutenção. Como ocorre com qualquer sistema baseado em verificações contra a linha base de um sistema, ele tende a gerar várias reclamações falso-positivas causadas

por legítimas atividades do dia a dia. O arquivo de configuração samhain ajuda a remover reclamações sobre arquivos que mudam com freqüência, mas os administradores não podem prever tudo. Tenha em mente que uma quantidade abundante de suporte será necessária, especialmente em uma instalação de larga escala.

Security-Enhanced Linux (SELinux)

Como vimos, o modelo de segurança do Linux tem falhas. Controles de acesso discricionários (sendo o conceito do acesso de arquivos permitido ao arbítrio de uma conta com permissões apropriadas) é um método conveniente, mas inseguro de controlar acesso aos objetos do sistema de arquivos. Ele é inerentemente baseado na confiança: na confiança de que usuários com acesso não são maliciosos, na confiança de que os administradores conhecem as permissões adequadas para cada arquivo em um pacote de software e na confiança de que os próprios pacotes de softwares de terceiros são instalados com controles fortes corretos. Mas, mesmo se tudo isso fosse verdadeiro, uma vulnerabilidade de software ainda poderia deixar o sistema desprotegido.

O SELinux resolve esse problema utilizando controles de acesso obrigatórios, conhecidos como MAC. Sob o MAC, os usuários não têm controle autorizado sobre o acesso a objetos. Em vez disso, um administrador define diretivas de acesso em todo o sistema. Uma diretiva MAC bem implementada conta com o princípio do menor privilégio (só permitindo acesso quando necessário), quase como um firewall adequadamente projetado que permite que apenas serviços e clientes especificamente reconhecidos tenham acesso. MAC pode impedir que um software com vulnerabilidades de execução de código (por exemplo, buffer overflows) comprometa o sistema limitando o escopo da brecha a alguns poucos recursos específicos requeridos por esse software.

O SELinux é um projeto da NSA livremente disponível desde o final de 2000. Ele foi integrado à série 2.6 do kernel do Linux. Mas a adoção do SELinux por distribuições individuais foi relativamente fraca, com as notáveis exceções do Linux Red Hat Enterprise e do Fedora.

O desenvolvimento de diretivas é um tópico complexo. Pelo menos uma empresa oferece um curso de três dias sobre a diretiva SELinux. Para proteger um novo daemon, por exemplo, uma diretiva precisa enumerar cuidadosamente todos os arquivos, diretórios e outros objetos aos quais o processo precisa de acesso. Para softwares complexos como o **sendmail** ou o **httpd** Apache, essa tarefa pode ser bem complexa.

Felizmente, há várias diretivas gerais disponíveis on-line. Essas podem ser facilmente instaladas e configuradas para seu ambiente em particular. Um editor de diretiva com todos os recursos cujo objetivo é facilitar a aplicação de diretivas pode ser encontrado em seedit.sourceforge.net.

O SELinux está presente no RHEL desde a versão 4. O Fedora incorporou o software no Core 2 e incluiu suporte adicional ao Core 3. Uma instalação padrão do Fedora ou do Red Hat Enterprise Linux inclui algumas proteções SELinux prontas para uso.

O arquivo **/etc/selinux/config** controla a configuração do SELinux. As linhas interessantes são

```
SELINUX=enforcing
SELINUXTYPE=targeted
```

A primeira linha tem três possíveis valores: `enforcing`, `permissive` ou `disabled`. A configuração `enforcing` assegura que a diretiva carregada seja aplicada e proíbe violações. `permissive` permite que ocorram violações, mas as registra em log por meio do syslog. `disabled` desativa completamente o SELinux.

`SELINUXTYPE` refere-se ao tipo de diretiva a ser aplicado. O Red Hat e o Fedora têm duas diretivas: `targeted`, que define segurança adicional para daemons que o Red Hat protegeu,[8] e `strict`, que protege o sistema inteiro. Embora a diretiva `strict` esteja disponível, ela não é suportada pelo Red Hat; as restrições são tão rígidas que o sistema é difícil de usar. A diretiva `targeted` oferece proteção para daemons de rede importantes sem afetar o uso geral do sistema, pelo menos teoricamente. Mas mesmo a diretiva `targeted` não é perfeita. Se você tiver problemas com um software recém-instalado, verifique em **/var/log/messages** erros do SELinux.

O SUSE utiliza sua própria implementação do MAC chamada AppArmor e não inclui o SELinux.

Os pacotes SELinux são mantidos para o Debian e o Ubuntu por Russel Coker, a pessoa na Red Hat que gerou as diretivas `strict` e `targeted`.

---

8. Os daemons protegidos são **httpd**, **dhcpd**, **mailman**, **named**, **portmap**, **nscd**, **ntpd**, **mysqld**, **postgres**, **squid**, **winbindd** e **ypbind**.

20.11 FERRAMENTAS DE SEGURANÇA CRIPTOGRÁFICAS

Muitos dos protocolos UNIX em uso comum nos sistemas Linux datam de uma época anterior à ampla implantação da Internet e à criptografia moderna. A segurança simplesmente não era um fator levado em consideração em vários protocolos; em outros, as preocupações com a segurança eram expressas com a transmissão de uma senha em texto simples ou com uma vaga verificação para ver se os pacotes eram provenientes de host ou porta confiável.

Esses protocolos agora se encontram operando nas águas infestadas por tubarões das grandes LANs corporativas e da Internet, onde, obrigatoriamente se deve supor que todo o tráfego está aberto à inspeção. Não somente isso, mas há pouco que se possa fazer para impedir que alguém interfira ativamente em conversas que trafegam pelas redes. Como ter certeza da identidade da pessoa com quem você está falando?

A criptografia resolve boa parte desses problemas. Por um longo tempo foi possível misturar mensagens de forma que alguém na escuta não pudesse decifrá-las, mas isto é apenas o início das maravilhas da criptografia. Desenvolvimentos como criptografia de chaves públicas e hashing seguro permitiram o desenvolvimento de sistemas de criptografia que atendem a praticamente todo tipo de necessidade concebível.[9]

Infelizmente, esses avanços matemáticos falharam grandemente em se transformarem em software seguro e utilizável que seja amplamente entendido e adotado. Os desenvolvedores de sistemas de software para criptografia tendem a estar muito interessados em provável exatidão e segurança absoluta e não tão interessados se um sistema tem ou não um sentido prático para o mundo real. A maioria dos softwares atuais tende a ser muito rebuscada em termos de engenharia e talvez não seja surpresa que os usuários saiam correndo quando têm a mínima chance de fazê-lo. Hoje, a população usando criptografia é formada em grande parte por hobistas interessados no assunto, malucos teóricos acerca de conspiração e aqueles que não têm nenhuma alternativa devido à política administrativa adotada.

Pode ser que vejamos, pode ser que não, uma abordagem mais sã ao desenvolvimento de criptografia ao longo dos próximos anos. Nesse meio tempo, algumas ofertas atuais discutidas nas próximas seções poderão ajudá-lo.

Kerberos: uma abordagem unificada da segurança de redes

O sistema Kerberos, desenvolvido no MIT, tenta abordar algumas das questões relativas à segurança das redes de uma maneira consistente e também expansível. Kerberos é um sistema de autenticação, um recurso que "garante" que usuários e serviços sejam realmente quem eles afirmam ser. Ele não fornece segurança adicional ou criptografia além dessa.

O Kerberos usa DES para construir conjuntos aninhados de credenciais chamados de "tíquetes". Os tíquetes são repassados pela rede para certificar sua identidade e para lhe dar acesso a serviços de rede. Cada instalação Kerberos deve manter pelo menos uma máquina fisicamente segura (chamada de servidor de autenticação) para executar o daemon Kerberos. Esse daemon emite bilhetes a usuários ou serviços que apresentam credenciais, como senhas, quando eles solicitam autenticação.

Em suma, o Kerberos aumenta a segurança de senhas tradicional do Linux de duas maneiras apenas: ele nunca transmite senhas decriptografadas pela rede e ele libera os usuários de digitarem senhas toda hora, tornando a proteção de senhas de serviços de rede um tanto mais aceitável.

A comunidade Kerberos se gaba de ter um dos documentos mais lúcidos e agradáveis nunca antes escritos sobre um sistema criptográfico: "Designing an Authentication System: a Dialogue in Four Scenes", de Bill Bryant. É uma leitura obrigatória para qualquer um que esteja interessado em criptografia e se encontra disponível em

web.mit.edu/kerberos/www/dialogue.html

Há também uma excelente FAQ:

www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html

O Kerberos dispõe de um modelo de segurança de rede melhor do que o modelo "ignorar completamente a segurança de uma rede". Entretanto, ele não é perfeitamente seguro nem fácil de instalar e utilizar. Ele não substitui nenhuma das medidas de segurança descritas neste capítulo. Em nossa opinião, para a grande parte das instalações é melhor não optar por ele. Uma boa higiene do sistema e uma solução criptográfica mais focalizada para logins remotos como o SSH (veja a página 481) devem fornecer um nível de segurança mais do que adequado a seus usuários.

Infelizmente (e talvez previsivelmente), o sistema Kerberos distribuído como parte do Windows utiliza extensões proprietárias não-documentadas para os protocolos. Como conseqüência, ele não opera bem com uma distribuição baseada no código do MIT.

PGP: Pretty Good Privacy

O pacote PGP, de Philip Zimmermann, fornece uma "cesta" de ferramentas com utilitários de criptografia que focalizam essencialmente a segurança de e-mail. Ele pode ser utilizado para criptografar dados, gerar assinaturas e para verificar a origem de arquivos e mensagens.

---

9. Um excelente recurso para aqueles interessados em criptografia são as "RSA Labs' Frequently Asked Questions about Today's Cryptography" em www.rsasecurity.com/rsalabs/faq. Além disso, o livro de Stephen Levy, *Cripto*, é um guia abrangente da história da criptografia.

Tentativas de regular ou impedir a distribuição do PGP deram a ele uma história bem tumultuada. A McAfee inclui o PGP no seu produto de servidor E-Business e a corporação PGP utiliza o PGP para uma variedade de produtos de criptografia. O projeto GNU fornece uma excelente implementação gratuita amplamente utilizada conhecida como GnuPG em www.gnupg.org. Uma versão do PGP a ser analisada governamentalmente está disponível para uso nos Estados Unidos e uma versão internacional com uma criptografia um pouco mais variada e mais forte está disponível em www.pgpi.org. Os sites internacionais dos repositórios de arquivos aparentemente não verificam endereços nos Estados Unidos, portanto, usuários norte-americanos devem ser bastante cuidadosos para que não acessem acidentalmente www.pgpi.org e façam a cópia da versão completa com todos os recursos do PGP.

O PGP é o software para criptografia mais popular de uso comum. Infelizmente, a versão UNIX/Linux é suficientemente complexa a ponto de exigir que você entenda uma quantidade considerável de conceitos de criptografia para poder usá-lo. Felizmente (?), o PGP vem com um tratado de 88 páginas sobre criptografia que pode ajudá-lo a preparar o terreno. Embora você possa achar o PGP útil em seu próprio trabalho, nós não recomendamos que dê suporte para os usuários, já que é sabido que ele gera muitas perguntas enigmáticas. Descobrimos que a versão para Windows do PGP é consideravelmente mais fácil de ser utilizada do que o comando **pgp** com seus 38 modos diversos de operação.

Os pacotes de software na Internet são normalmente distribuídos com um arquivo de assinatura PGP que garante a origem e a pureza do software. Infelizmente, é difícil para as pessoas que não são usuários obstinados do PGP validarem essas assinaturas — não pelo processo de validação ser complicado, mas porque a verdadeira segurança PGP pode provir somente da coleta de uma biblioteca pessoal de chaves públicas de pessoas cujas identidades você verificou diretamente. Fazer a cópia de uma única chave pública juntamente com um arquivo de assinatura e distribuição de software é quase tão seguro quanto fazer a cópia da distribuição sozinha.

SSH: o shell seguro

O sistema SSH, criado por Tatu Ylönen, é um substituto seguro para **rlogin**, **rcp** e **telnet**. Ele usa autenticação criptográfica para confirmar a identidade de um usuário e criptografa todo o fluxo de comunicações entre os dois hosts. O protocolo utilizado pelo SSH é concebido para resistir a uma série de ataques potenciais. O protocolo está documentado pelas RFCs 4250 a 4256 e é agora um padrão proposto para o IETF.

O SSH sofreu uma transformação, passando de um projeto de código-fonte aberto distribuído gratuitamente (SSH1) para se tornar um produto comercial que utiliza um protocolo ligeiramente diferente (e mais seguro), o SSH2. Felizmente, a comunidade de código-fonte aberto respondeu distribuindo o excelente pacote OpenSSH (mantido pela OpenBSD) que agora implementa os dois protocolos.

Os principais componentes do SSH são um daemon de servidor, **sshd**, e dois comandos em nível de usuário: **ssh** para logins remotos e **scp** para copiar arquivos. Outros componentes são um comando **ssh-keygen** que gera pares de chaves públicas e uma série de utilitários que auxilia no suporte a X Windows seguro.

sshd pode autenticar logins de usuários de diversas maneiras diferentes. Fica a seu cargo como administrador decidir quais destes métodos são aceitáveis:

- **Método A**: Se o nome do host remoto por meio do qual o usuário está fazendo o login estiver listado em **~/.rhosts**, **~/.shosts**, **/etc/hosts.equiv** ou **/etc/shosts.equiv**, então o usuário entra no sistema automaticamente sem necessidade de verificação de uma senha. Esse esquema espelha aquele do antigo daemon **rlogin** e, em nossa opinião, *não* é aceitável para emprego comum.
- **Método B**: Como um refinamento do método A, **sshd** também pode utilizar criptografia de chaves públicas para verificar a identidade do host remoto. Para isso acontecer, a chave pública do host remoto (gerada no momento da instalação) deve ser listada no arquivo **/etc/ssh_known_hosts** do host local ou no arquivo **~/.ssh/known_hosts** do usuário. Se o host remoto puder provar que ele conhece a chave privada correspondente (normalmente armazenada em **/etc/ssh_host_key**, um arquivo que não pode ser lido por todos), então o usuário entra no sistema sem lhe ser solicitado uma senha. O método B é mais restritivo do que o método A, porém acreditamos que ele ainda não seja tão seguro. Se a segurança do host de origem estiver comprometida, a instalação local também estará comprometida.
- **Método C**: **sshd** pode utilizar criptografia de chaves públicas para estabelecer a identidade do usuário. No momento do login, o usuário precisa ter acesso a uma cópia de seu arquivo de chaves públicas e deve fornecer uma senha para decifrá-la. Esse método é o mais seguro, mas é irritante fazer sua configuração. Também significa que usuários não podem se conectar quando estiverem viajando a menos que tragam junto uma cópia dos seus arquivos de chave privada (talvez em uma chave USB, esperamos criptografada). Se decidir utilizar pares de chaves, utilize **ssh -v** extensamente durante o processo de solução de problemas.
- **Método D**: Por fim, **sshd** pode simplesmente permitir que o usuário introduza sua senha de login normal. Isso faz com que **ssh** se comporte de forma muito semelhante a **telnet**, exceto pelo fato de a senha e sessão serem ambas criptografadas. As principais desvantagens desse método são que as senhas de login do sistema podem ser relativamente fracas se você não tiver aprimorado sua segurança e que há ferramentas prontas para uso (como John the Ripper) projetadas para quebrá-las. Entretanto, esse método provavelmente é a melhor opção para uso normal.

A política de autenticação é configurada no arquivo **/etc/sshd_config**. Você logo verá que esse arquivo foi preenchido com lixo de configuração, mas a maior parte dele pode ser seguramente ignorada. As opções relevantes à autenticação são mostradas na Tabela 20.1.

Tabela 20.1 Opções em /etc/sshd_config relacionadas com a autenticação

Opção	Método[a]	Padrão	Significado quando ativa
RhostsAuthentication	A	não	Permite login via **~/.shosts**, **/etc/shosts.equiv** etc.
RhostsRSAAuthentication	B	sim	Permite **~/.shosts**, et al., mas também requer chave de host
IgnoreRhosts	A,B	não	Ignora os arquivos **~/.rhosts** e o **hosts.equiv**[b]
IgnoreRootRhosts	A,B	não[c]	Impede autenticação rhosts/shosts para root
RSAAuthentication	C	sim	Permite autenticação com criptografia de chaves públicas por usuário
PasswordAuthentication	D	sim	Permite o emprego de senha de login normal

a. Os métodos de autenticação aos quais essa variável é relevante.
b. Mas continua a honrar **~/.shosts** e **shosts.equiv**.
c. Por padrão, assume o valor de IgnoreRhosts.

Nossa sugestão de configuração, que permite os métodos C e D, mas não os métodos A ou B, é mostrada a seguir.

```
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
```

Nunca é prudente permitir que root efetue o login remotamente. O acesso de superusuário deve ser alcançado pelo uso do **sudo**. Para encorajar esse comportamento, utilize a opção

```
PermitRootLogin no
```

Senhas utilizadas uma única vez

Ferramentas de força bruta, como John the Ripper, chamam a atenção para a insegurança das senhas estáticas. Mesmo quando sujeitos a ferramentas como PAM e à expiração de senha, os usuários têm reputação de selecionarem senhas fáceis de lembrar, mas fracas, e ainda acabam escrevendo essas senhas em notas adesivas compartilhando-as com os companheiros de trabalho. Senhas utilizadas uma única vez resolvem esse problema impondo uma senha exclusiva em cada login. Como a senha muda constantemente, uma ferramenta de força bruta não vale nada.

Hoje em dia, as aparições mais comuns de senhas utilizadas somente uma vez (one-time passwords – OTP) se dão em produtos de segurança comerciais. Uma série de fabricantes oferece sistemas com senhas para um único uso com base em pequenos dispositivos do tamanho de um "chaveiro" (ou do tamanho de um cartão de crédito) com displays LCD que geram senhas instantaneamente. No mínimo, distribuir e substituir pequenos dispositivos de hardware manterá o seu assistente administrativo ocupado por algumas horas por semana.

Stunnel

Stunnel, criado por Michal Trojnara, é um pacote de código-fonte aberto que criptografa conexões TCP arbitrárias. Ele utiliza SSL (Secure Sockets Layer) para criar "túneis" de uma extremidade a outra pelos quais ele passa os dados para e a partir de um serviço não-criptografado. Sabe-se que ele funciona bem com serviços inseguros como Telnet, IMAP e POP.

Um daemon **stunnel** executa tanto em sistemas cliente como em sistemas servidor. O **stunnel** local normalmente aceita conexões na porta tradicional do serviço (por exemplo, porta 25 para SMTP) e as roteia por meio da SSL para um **stunnel** no host remoto. O **stunnel** remoto aceita a conexão, decripta os dados que chegam e os roteia para a porta remota em que o servidor está ouvindo. Esse sistema permite que serviços não criptografados tirem proveito da confidencialidade e integridade oferecidas pela criptografia sem exigir nenhuma alteração no software. O software cliente precisa ser configurado para procurar serviços apenas no sistema local em vez de no servidor que em última instância irá fornecê-los.

Telnet é um bom exemplo porque ele consiste em um daemon simples que ouve em uma única porta. Para usar **stunnel** em um link Telnet, o primeiro passo é criar um certificado SSL. Stunnel é independente da biblioteca SSL, portanto qualquer implementação baseada em padrões será suficiente; gostamos do OpenSSL. Para gerar o certificado:

```
server# openssl req -new -x509 -days 365 -nodes -out stunnel.pem -keyout stunnel.pem
Generating a 1024 bit RSA private key
.++++++
.................................++++++
writing new private key to 'stunnel.pem'
-----
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) [GB]:US
State or Province Name (full name) [Berkshire]:Colorado
Locality Name (eg, city) [Newbury]:Boulder
Organization Name (eg, company) [My Company Ltd]:Booklab, Inc.
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:server.example.com
Email Address []:
```

Esse comando cria um certificado auto-assinado, sem frase senha. Embora não utilizar uma frase senha seja uma conveniência (um ser humano não precisa estar presente para digitar uma frase senha toda vez que o **stunnel** reinicia), também introduz riscos à segurança. Tenha cuidado ao proteger o arquivo de certificado com permissões fortes.

Em seguida, defina a configuração tanto para o **stunnel** do servidor como para o **stunnel** do cliente. O arquivo de configuração padrão é **/etc/stunnel/stunnel.conf**, mas você pode criar várias configurações se quiser executar mais de um túnel.

```
cert = /etc/stunnel/stunnel.pem
chroot = /var/run/stunnel/
pid = /stunnel.pid
setuid = nobody
setgid = nobody
debug = 7
output = /var/log/stunnel.log
client = no

[telnets]
accept = 992
connect = 23
```

Há algumas questões importantes sobre a configuração do servidor. Primeiro, observe a instrução **chroot**, que confina o processo **stunnel** no diretório **/var/run/stunnel**. Os caminhos para os arquivos acessórios talvez precisem ser expressos no espaço de nome regular do sistema ou no espaço de nomes executado por **chroot**, dependendo do ponto em que forem abertos. Aqui, o arquivo **stunnel.pid** na realidade está localizado em **/var/run/stunnel**.

A seção `[telnetd]` tem duas instruções: `accept` instrui **stunnel** a aceitar conexões na porta 992 e `connect` passa essas conexões para a porta 23, o serviço Telnet real.

A configuração do cliente é muito semelhante:

```
cert = /etc/stunnel/stunnel.pem
chroot = /var/run/stunnel/
pid = /stunnel.pid
setuid = nobody
setgid = nobody
debug = 7
output = /var/log/stunnel.log
client = yes

[telnets]
accept = 23
connect = server.example.com:992
```

Duas diretivas são invertidas em relação à configuração do servidor. A instrução `client = yes` instrui o programa a iniciar conexões **stunnel** em vez de aceitá-las. O **stunnel** local ouve conexões na porta 23 e se conecta ao servidor na porta 992. O hostname na diretiva `connect` deve corresponder com a entrada especificada quando o certificado foi criado.

Tanto o **stunnel** do cliente como o **stunnel** do servidor podem ser iniciados sem argumentos de linha de comando. Se verificar com **netstat -an**, você deverá ver o **stunnel** do servidor esperando conexões na porta 992 enquanto o **stunnel** do cliente espera na porta 23.

Para acessar o túnel, um usuário simplesmente envia **telnets** ao host local:

```
client# telnet localhost 23
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.
Red Hat Enterprise Linux WS release 4 (Nahant Update 2)
Kernel 2.6.9-5.EL on an i686
login:
```

O usuário pode agora efetuar o login de maneira segura sem temer roubo de senhas. Um administrador atento seria cuidadoso ao utilizar empacotadores TCP a fim de restringir conexões no cliente apenas à interface local — a intenção não é permitir que todo mundo use **telnet** de maneira segura ao servidor! **stunnel** é um dos vários programas com suporte predefinido para empacotar aplicativos e não requer o uso de **tcpd** para restringir acesso. Acesse www.stunnel.org para instruções.

20.12 FIREWALLS

Além de proteger máquinas individuais, você também pode implementar algumas precauções de segurança no nível de rede. A ferramenta básica de segurança de rede é o "firewall". As três principais categorias de firewalls são filtragem de pacotes, proxy de serviço e inspeção de informações de estado.

Firewalls para filtragem de pacotes

Um firewall para filtragem de pacotes limita os tipos de tráfego que podem passar pelo seu gateway Internet (ou por um gateway interno que separa domínios dentro de sua empresa) baseado em informações contidas no cabeçalho do pacote. É bem semelhante a atravessar uma fronteira internacional de carro passando por um posto alfandegário. Você especifica quais são os endereços de destino, números de portas e tipos de protocolos aceitáveis, e o gateway simplesmente descarta (e, em alguns casos, registra em log) pacotes que não atendam ao perfil preestabelecido.

A filtragem de pacotes é suportada por roteadores dedicados como aqueles fabricados pela Cisco. Eles também podem estar disponíveis em software, dependendo da máquina que você estiver usando como gateway e de sua configuração. Em geral, os firewalls para filtragem de pacotes oferecem um aumento significativo na segurança com baixo custo em termos de desempenho ou de complexidade.

O Linux inclui software para filtragem de pacotes (consulte os detalhes que começam na página 486 para informações adicionais). Também é possível comprar software comercial para desempenhar essa função. Todos esses pacotes possuem valores especiais podendo fornecer um firewall razoavelmente seguro para ambientes domésticos ou pequenos escritórios. Entretanto, você deve se referir aos comentários no início deste capítulo antes de considerar um sistema Linux como um firewall corporativo em nível de produção.[10] Esse é um caso no qual você realmente deve gastar dinheiro com um aplicativo de rede dedicado, por exemplo, o firewall PIX da Cisco.

Como os serviços são filtrados

A maioria dos serviços bem conhecidos está associada a uma porta de rede no arquivo **/etc/services** ou seu equivalente específico a um determinado fabricante. Os daemons que fornecem esses serviços se vinculam a portas apropriadas e esperam conexões a partir dos sites remotos.[11] A maioria das portas bem conhecidas de serviço é "privilegiada", significando que seus números de porta estão no intervalo de 1 a 1.023. Essas portas só podem ser utilizadas por um processo que executa como root. As portas de números acima de 1.024 são conhecidas como portas não privilegiadas.

A filtragem de serviços específicos se baseia na suposição de que o cliente (a máquina que inicia uma conversa TCP ou UDP) usa uma porta não privilegiada para contatar uma porta privilegiada no servidor. Por exemplo, se você quisesse permitir somente conexões SMTP que chegassem para uma máquina com o endereço 192.108.21.200, você instalaria um filtro que permitisse pacotes TCP destinados àquele endereço na porta 25 e que permitisse pacotes TCP que saíssem desse endereço para qualquer lugar.[12] A forma exata como tal filtro seria instalado dependeria do tipo de roteador que você estivesse usando.

---

10. Partimos do pressuposto de que você já saiba que não deve levar em consideração algo como o Windows como plataforma de firewall. O nome "Windows" evoca imagens de segurança? Bobinho, o Windows é destinado a desktops.
11. Em muitos casos **xinetd** realiza a espera real em nome desses servidores. Consulte a página 609 para informações adicionais.
12. A porta 25 é a porta SMTP conforme definido em **/etc/services**.

Veja a página 506 para informações adicionais sobre como configurar um servidor ftp.

Alguns serviços, como o FTP, acrescentam uma variável inesperada ao quebra-cabeças. O protocolo FTP usa de fato duas conexões TCP ao transferir um arquivo: uma para comandos e a outra para dados. O cliente inicia a conexão de comandos e o servidor inicia a conexão de dados. Portanto, se você quiser usar FTP para recuperar arquivos da Internet, terá de permitir acesso de entrada a todas as portas TCP não privilegiadas, já que você não tem a mínima idéia de qual porta poderia ser utilizada para formar uma conexão de dados de entrada.

Esse ajuste frustra em grande parte o objetivo da filtragem de pacotes, pois alguns serviços notoriamente inseguros (por exemplo, X11 na porta 6000) naturalmente se vinculam a portas não privilegiadas. Essa configuração também cria uma oportunidade para usuários curiosos dentro de sua empresa iniciarem seus próprios serviços (como um servidor **telnet** em uma porta não padronizada e não privilegiada) que eles ou seus amigos poderão acessar pela Internet.

Uma solução comum para o problema de FTP é utilizar o protocolo de transferência de arquivo SSH. Esse protocolo é atualmente um draft na Internet, mas é amplamente utilizado e maduro. Ele é comumente utilizado como um subcomponente SSH, que fornece sua autenticação e criptografia. Diferentemente do FTP, o SFTP só utiliza uma única porta tanto para comandos como para dados, resolvendo convenientemente o paradoxo da filtragem de pacotes. Há algumas implementações do SFTP. Tivemos muita sorte com o cliente SFTP de linha de comando fornecido pelo OpenSSH.

Se precisar utilizar FTP, uma abordagem razoável é permitir o FTP ao mundo externo somente a partir de um único host isolado. Os usuários poderão fazer login na máquina FTP quando precisarem realizar outras operações de rede que são proibidas pela rede interna. Como replicar todas as contas de usuário no "servidor" FTP frustraria o objetivo da separação administrativa, talvez seja interessante criar contas FTP somente por solicitação. Naturalmente, o host FTP deveria usar um conjunto completo de ferramentas para verificação da segurança do sistema.

A maneira mais segura de utilizar um filtro de pacotes é iniciar com uma configuração que não permita nada a não ser SMTP interno ou SSH. A seguir você pode liberar o filtro pouco a pouco à medida que for descobrindo coisas úteis que não funcionam.

Alguns sites cientes quanto à segurança utilizam uma filtragem de duas etapas. Nesse esquema, um filtro é um gateway para a Internet e um segundo filtro se localiza entre o gateway externo e o restante da rede local. A idéia é deixar o gateway externo relativamente aberto e configurar o gateway interno de maneira conservadora. Se as máquinas intermediárias forem administrativamente separadas do restante da rede, elas poderão fornecer uma variedade de serviços na Internet com risco reduzido. A rede parcialmente segura normalmente é chamada "zona desmilitarizada" ou "DMZ".

Firewalls de proxy de serviços

Proxies de serviço interceptam conexões para e a partir do mundo externo e estabelecem novas conexões no lado oposto do firewall, atuando como uma espécie de serviço de leva e traz ou "acompanhante vigilante" entre esses dois mundos. É bem parecido com ir de carro até a fronteira de seu país, atravessar a fronteira a pé e alugar um carro limpinho do outro lado da fronteira para continuar a viagem.

Devido ao seu desenho, os firewalls proxy de serviços são muito menos flexíveis (e muito mais lentos) do que filtros de pacotes puros. Seu proxy deve ter um módulo que decodifique e transporte cada protocolo que você quiser que passe pelo firewall. No início dos anos 90 isso era relativamente fácil, pois havia somente alguns protocolos usados comumente. Hoje em dia, os internautas podem usar várias dezenas de protocolos em uma hora de navegação na rede. Como resultado, os proxies de serviços são relativamente impopulares em empresas que usam a Internet como principal meio para comunicação.

Firewalls de inspeção com informações de estado

A teoria por trás dos firewalls de inspeção com informações de estados está no fato de que, se você pudesse escutar cuidadosamente e compreender todas as conversas (em todas as línguas) que estivessem ocorrendo em um aeroporto lotado, você poderia ter certeza de que ninguém estaria planejando colocar uma bomba em um avião mais tarde nesse dia. Os firewalls de inspeção com informações de estado são projetados para inspecionar o tráfego que os atravessa e comparar a real atividade de rede com aquela que teoricamente "deveria" estar acontecendo. Por exemplo, se os pacotes trocados em uma seqüência de comandos FTP citarem uma porta para ser utilizada posteriormente para uma conexão de dados, o firewall deve esperar que uma conexão de dados ocorra somente naquela porta anunciada. Tentativas do site remoto de se conectar a outras portas provavelmente são adulteradas e devem ser eliminadas.

Infelizmente, a realidade em geral mata o gato aqui mesmo. Não é mais realístico tentar rastrear o "estado" das conexões de rede de milhares de hosts utilizando centenas de protocolos do que ouvir cada conversa em cada uma das línguas em um aeroporto internacional lotado. Algum dia, quando a capacidade de processamento e de memória aumentarem, talvez isso seja factível.

Portanto, o que realmente os fabricantes estão vendendo quando eles dizem fornecer inspeção com informações de estados? Os seus produtos monitoram um número muito limitado de conexões ou protocolos, ou então tentam localizar um dado conjunto de situações "ruins". Não que haja alguma coisa errada com isso; claramente, alguns benefícios são derivados de uma tecnologia qualquer que possa detectar anomalias no tráfego. Nesse caso em particular, porém, é importante se lembrar de que as alegações são, *em grande parte*, apenas exageros de marketing.

Firewalls: qual o verdadeiro nível de segurança oferecido por eles?

Um firewall não deve ser seu principal meio de defesa contra a ação de intrusos. Ele só é apropriado como uma medida de segurança complementar. O emprego de firewalls normalmente fornece um falso sentido de segurança. Se ele o induzir a relaxar em outras medidas de segurança, terá um efeito *negativo* na segurança de sua instalação.

Cada host dentro da sua empresa deve ser mantido individualmente seguro e monitorado de maneira regular com ferramentas como **xinetd**, Nmap, Nessus e samhain. Da mesma maneira, toda a sua comunidade de usuários precisa ser instruída sobre a higiene básica em termos de segurança. Caso contrário, você estará simplesmente erigindo uma estrutura que tem uma casca dura e um miolo mole e precário.

Idealmente, os usuários locais devem ser capazes de se conectar a qualquer serviço que queiram, mas as máquinas na Internet devem ser capazes de se conectar a apenas um conjunto limitado de serviços locais. Por exemplo, talvez você queira permitir acesso FTP a um servidor de arquivos local e permitir conexões SMTP (e-mail) para o seu servidor de e-mail.

Para maximizar o valor da sua conexão com a Internet, recomendamos que você dê maior ênfase à conveniência e à acessibilidade ao decidir como configurar sua rede. No final do dia, é a vigilância do administrador de sistemas que torna uma rede segura, não um hardware de firewall sofisticado.

20.13 Recursos de firewall do Linux: tabelas de IP

Tradicionalmente não recomendamos o emprego de sistemas Linux (ou UNIX ou Windows) como firewalls devido à falta de segurança geral de executar um sistema operacional completo. Dispositivos embutidos projetados especificamente para roteamento e filtragem de pacotes (como uma máquina Cisco PIX) torna os firewalls melhores,[13] mas um sistema Linux aperfeiçoado é um excelente substituto para as empresas que não têm condições financeiras de comprar um dispositivo de firewall caro.

Se você estiver decidido a usar uma máquina Linux como firewall, pelo menos se certifique de que ela esteja atualizada em relação a correções e configurações relativas à segurança de sistemas. Uma máquina firewall é um excelente local para colocar em prática todas as recomendações deste capítulo. (A seção que inicia na página 484 discute os firewalls para filtragem de pacotes em geral. Se não estiver familiarizado com o conceito básico de firewall, seria interessante ler aquela seção antes de prosseguir.)

A versão 2.4 do kernel do Linux introduziu um mecanismo de manipulação de pacotes todo novo chamado Netfilter. A ferramenta utilizada para controlar Netfilter, **iptables**, é o irmão mais velho do antigo comando **ipchains** utilizado nos kernels 2.2 do Linux. **iptables** aplica "cadeias" ordenadas de regras a pacotes de rede. Conjuntos de cadeias formam as "tabelas" e essas são utilizadas para manipular tipos de tráfego específicos.

Por exemplo, a tabela **iptables** padrão se chama "filtro". Cadeias de regras nessa tabela são utilizadas para filtragem dos pacotes do tráfego de rede. A tabela de filtros contém três cadeias padrão. Cada pacote que é tratado pelo kernel passa exatamente por uma dessas cadeias. As regras na cadeia FORWARD são aplicadas a todos os pacotes que chegam a uma interface de rede e precisam ser encaminhados a uma outra. As regras contidas nas cadeias INPUT e OUTPUT são aplicadas, respectivamente, a tráfego destinado a ou originando-se do host local. Essas três cadeias padrão normalmente são tudo de que você precisa para aplicar um firewall entre duas interfaces de rede. Se necessário, você pode definir uma configuração personalizada para suportar situações de roteamento ou de contabilidade mais complexas.

Além da tabela de filtros, **iptables** inclui as tabelas "nat" e "mangle". A tabela nat contém cadeias de regras que controlam Network Address Translation (neste caso, "nat" é o nome da tabela **iptables** e "NAT" é o nome do esquema genérico para conversão de endereços). A seção *Endereços privados e o NAT* na página 203 discute o NAT, e um exemplo da tabela nat em ação é mostrado na página 223. Posteriormente, nesta seção, utilizamos a cadeia PREROUTING da tabela nat para filtragem de pacotes anti-espionagem.

A tabela mangle contém cadeias que modificam ou alteram o conteúdo dos pacotes de rede fora do contexto da filtragem de pacotes e NAT. Embora a tabela mangle seja útil para tratamento de pacotes especiais como reinicializar o tempo de vida (*time-to-live* – TTL) do cabeçalho IP, em geral, ela não é utilizada na maioria dos ambientes de produção. Nesta seção, discutimos somente as tabelas de filtragem e nat, deixando a tabela mangle para os aventureiros.

Cada regra que forma uma cadeia possui uma estrutura "alvo" que determina o que fazer com os pacotes que atendem às regras. Quando um pacote corresponde com uma regra, seu destino é, na maioria das vezes, selado; nenhuma regra adicional será verificada. Embora muitos destinos sejam definidos internamente às **iptables**, é possível especificar uma outra cadeia como destino de uma regra.

Os alvos/destinos disponíveis a essas regras na tabela de filtragem são ACCEPT, DROP, REJECT, LOG, MIRROR, QUEUE, REDIRECT, RETURN e ULOG. Quando uma regra resultar em um ACCEPT, os pacotes que atendem a essa regra terão permissão para prosseguirem em sua trajetória. DROP e REJECT descartam seus pacotes. DROP é silencioso e

13. Dito isso, muitos dispositivos de rede voltados ao consumidor, como os produtos de roteador da Linksys, utilizam o Linux e **iptables** em sua base.

REJECT retorna uma mensagem de erro ICMP. LOG fornece uma maneira simples de monitorar pacotes à medida que eles correspondem com as regras e o ULOG fornece capacidade de registro em log estendido.

REDIRECT desvia pacotes para um proxy em vez de deixá-los continuar em suas trajetórias iniciais. Você poderia utilizar esse recurso para forçar o tráfego web de todo o seu site a passar por um cache Web como o Squid. RETURN termina cadeias definidas por usuários e é análogo à instrução return na chamada de uma sub-rotina. MIRROR faz swap dos endereços IP de origem e de destino antes de enviar o pacote. Finalmente, QUEUE encaminha pacotes a programas de usuários locais por meio de um módulo de kernel.

Um firewall Linux é implementado normalmente como uma série de comandos **iptables** contidos em um script de inicialização **rc**. Comandos **iptables** individuais normalmente assumem uma das seguintes formas:

```
iptables -F nome-da-cadeia
iptables -P nome-da-cadeia alvo
iptables -A nome-da-cadeia -i interface -j alvo
```

A primeira forma (**-F**) limpa regras anteriores da cadeia. A segunda forma (**-P**) configura uma política padrão (também conhecida como destino) para a cadeia. Recomendamos que você utilize DROP como destino para a cadeia padrão. A terceira instância (**-A**) anexa a especificação atual à cadeia. A menos que você especifique uma tabela com o argumento **-t**, seus comandos se aplicarão a cadeias na tabela de filtragem. O parâmetro **-i** aplica a regra à interface designada e **-j** identifica o alvo. **iptables** aceita muitas outras estruturas, algumas das quais são mostradas na Tabela 20.2.

Tabela 20.2 Opções de linha de comando para filtros iptables

Cláusula	Significado ou valores possíveis
-p *proto*	Concordâncias por protocolo: **tcp**, **udp** ou **icmp**
-s *ip-de-origem*	Concorda com o endereço IP de origem de rede ou host (a notação CIDR está certa)
-d *ip-de-origem*	Concorda com o endereço de destino de rede ou host
--sport *porta#*	Concorda com a porta de origem (observe os dois traços)
--dport *porta#*	Concorda com a porta de destino (observe os dois traços)
--icmp-type *tipo*	Concorda com o código de tipo ICMP (observe os dois traços)
!	Nega uma estrutura
-t *table*	Especifica a tabela à qual um comando se aplica (o padrão é **filter**)

A seguir analisamos um exemplo completo. Partimos do pressuposto de que a interface ppp0 vai para a Internet e que a interface eth0 vai para uma rede interna. O endereço IP de ppp0 é 128.138.101.4, o endereço IP de eth0 é 10.1.1.1 e ambas as interfaces têm uma máscara de rede equivalente a 255.255.255.0. Esse exemplo utiliza filtragem de pacotes sem informações de estado para proteger o servidor Web com o endereço IP 10.1.1.2, que é o método padrão para proteger servidores Internet. Posteriormente no exemplo, mostraremos como utilizar filtragem com informações de estado para proteger usuários desktop.

Antes de poder utilizar **iptables**, é necessário ativar o encaminhamento IP e certificar-se de que os diversos módulos **iptables** tenham sido carregados no kernel. Para informações adicionais sobre como ativar o encaminhamento IP, consulte as seções *Ajustando os parâmetros do kernel do Linux*, na página 600, ou *Variáveis do kernel relacionadas com segurança*, na página 223. Os pacotes que instalam **iptables** geralmente incluem scripts de inicialização para alcançar essa ativação e carregamento.

Nosso primeiro conjunto de regras inicializa a tabela de filtros. Inicialmente, todas as cadeias são esvaziadas, depois o destino padrão das cadeias INPUT e FORWARD é configurado em DROP. Como acontece com qualquer outro firewall de rede, a estratégia mais segura é a de descartar quaisquer pacotes para os quais você não tenha dado permissão explícita.

```
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
```

Como as regras são avaliadas na ordem em que aparecem em uma cadeia, colocaremos nossas regras mais usadas na frente.[14] As três primeiras regras na cadeia FORWARD permitem conexões pelo firewall para serviços de rede em 10.1.1.2. Mais especificamente, permitimos SSH (porta 22), HTTP (porta 80) e HTTPS (porta 443) pelo nosso servidor Web. A primeira regra permite todas as conexões pelo firewall que se originam de dentro da rede confiável.

```
iptables -A FORWARD -i eth0 -p ANY -j ACCEPT
```

14. Entretanto, você deve estar ciente de que reordenar regras para fins de desempenho não altera a funcionalidade.

```
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 22 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p tcp --dport 443 -j ACCEPT
```

O único tráfego TCP que permitimos para nosso host de firewall (10.1.1.1) é SSH, que é útil para gerenciar o firewall. A segunda regra listada a seguir permite tráfego de loopback, que permanece local para nosso host de firewall. Nossos administradores ficam nervosos quando não conseguem fazer um **ping** em suas rotas padrão, de forma que a terceira regra aqui permite pacotes ICMP ECHO_REQUEST por meio de nossos endereços IP internos.

```
iptables -A INPUT -i eth0 -d 10.1.1.1 -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i lo -d 127.0.0.1 -p ANY -j ACCEPT
iptables -A INPUT -i eth0 -d 10.1.1.1 -p icmp --icmp-type 8 -j ACCEPT
```

Para qualquer host TCP/IP funcionar adequadamente na Internet, determinados tipos de pacotes ICMP têm de ter permissão para passar pelo firewall. As oito regras a seguir permitem um conjunto mínimo de pacotes ICMP no host para firewall, bem como na rede que está por trás dele.

```
iptables -A INPUT -p icmp --icmp-type 0 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 5 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 11 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 0 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 3 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 5 -j ACCEPT
iptables -A FORWARD -d 10.1.1.2 -p icmp --icmp-type 11 -j ACCEPT
```

A seguir adicionamos regras à cadeia PREROUTING na tabela nat. Embora a tabela nat não se destine à filtragem de pacotes, sua cadeia PREROUTING é particularmente útil para filtragem anti-espionagem. Se colocarmos entradas DROP na cadeia PREROUTING, elas não precisarão estar presentes nas cadeias INPUT e FORWARD, já que a cadeia PREROUTING é aplicada a todos os pacotes que entram no host firewall. É mais elegante colocar as entradas em um único lugar em vez de duplicá-las.

```
iptables -t nat -A PREROUTING -i ppp0 -s 10.0.0.0/8 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 172.16.0.0/12 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 192.168.0.0/16 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 127.0.0.0/8 -j DROP
iptables -t nat -A PREROUTING -i ppp0 -s 224.0.0.0/4 -j DROP
```

Finalmente, encerramos as cadeias INPUT e FORWARD com uma regra que proíbe todos os pacotes que não possuem permissão explícita. Embora já tenhamos forçado esse comportamento por meio dos comandos **iptables -P**, o alvo LOG nos permite ver quem da Internet está batendo à nossa porta.

```
iptables -A INPUT -i ppp0 -j LOG
iptables -A FORWARD -i ppp0 -j LOG
```

Opcionalmente, poderíamos configurar IP NAT para dissimular o espaço de endereços privados utilizado na rede interna. Consulte a seção *Linux NAT* na página 223, para informações adicionais sobre o NAT.

Um dos recursos mais poderosos que Netfilter traz para os firewalls Linux é a filtragem de pacotes com informações de estado. Em vez de permitir a entrada de serviços específicos, um firewall para clientes conectados à Internet precisa permitir respostas que chegam às requisições dos clientes. A cadeia FORWARD com informações de estado simples a seguir permite que todo o tráfego deixe nossa rede, porém autoriza só a entrada de tráfego que estiver relacionada às conexões iniciadas pelos nossos hosts.

```
iptables -A FORWARD -i eth0 -p ANY -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Certos módulos de kernel têm de ser carregados para ativar **iptables** a rastrear sessões de rede complexas como aquelas de FTP e IRC. Se esses módulos não estiverem carregados, **iptables** simplesmente não autoriza tais conexões. Embora filtros de pacotes de informações de estado possam aumentar a segurança da instalação, eles também adicionam complexidade à rede. Certifique-se de que você realmente precisa desse recurso de informações de estado antes de implementá-lo no firewall.

Talvez a melhor maneira de depurar conjuntos de regras **iptables** seja utilizar **iptables -L -v**. Essas opções informam quantas vezes cada regra nas suas cadeias correspondeu com um pacote. Normalmente adicionamos regras **iptables** tempo-

rárias com o alvo LOG quando queremos informações adicionais sobre os pacotes que coincidiram com as regras preestabelecidas. É possível resolver problemas mais complexos utilizando um analisador de pacotes como **tcpdump**.

20.14 REDES PRIVADAS VIRTUAIS (VPNS)

Um dos desenvolvimentos mais interessantes nos últimos anos foi o advento das redes privadas virtuais (Virtual Private Networks – VPNs). Essa tecnologia se tornou possível em grande parte pelo aumento do poder de processamento que agora se obtém de um único chip (e nas estações de trabalho dos usuários). Em sua forma mais simples, uma VPN é uma conexão que faz com que uma rede remota pareça estar conectada diretamente, mesmo que ela esteja fisicamente a milhares de quilômetros e a muitos saltos de roteador de distância. Para maior segurança, a conexão não só é autenticada de alguma maneira (normalmente usando "segredo compartilhado" como uma senha), como também o tráfego de ponta a ponta também é criptografado. Tal arranjo é normalmente conhecido como "túnel seguro".

Eis um bom exemplo do tipo de situação na qual uma VPN é útil: suponha que uma empresa tenha filiais em Chicago, Boulder e Miami. Se cada filial tiver uma conexão com um provedor de serviços Internet local, a empresa poderá usar VPNs para conectar os escritórios de modo transparente (e, em grande parte, de forma segura) por meio da Internet não segura. A empresa poderia atingir um resultado similar alugando linhas dedicadas para conectar três escritórios, mas essa opção poderia ser consideravelmente mais cara.

Um outro bom exemplo é uma empresa cujos funcionários trabalham a partir das suas residências. As VPNs permitiriam que esses usuários colhessem os frutos de seus serviços a cabo baratos e de alta velocidade enquanto ainda fariam com que eles parecessem diretamente conectados à rede corporativa.

Devido à conveniência e à popularidade desta funcionalidade, todo o mundo está oferecendo algum tipo de solução VPN. Você pode adquiri-la de seu fabricante de roteador, na forma de um plug-in para o seu sistema operacional ou até mesmo na forma de um dispositivo VPN dedicado para sua rede. Dependendo de seu orçamento e de suas necessidades de expansão, talvez seja interessante considerar umas das várias soluções VPN comerciais existentes no mercado.

Se estiver com o orçamento pequeno e querendo uma solução rápida, o SSH fará o tunelamento seguro para você. O SSH normalmente fornece uma conectividade uma porta por vez, mas também pode fornecer funcionalidade pseudo-VPN conforme ilustrado em um exemplo na página 229 que usa PPP por meio de um túnel SSH.

Túneis IPSec

Se for um fã dos padrões IETF (ou de economizar dinheiro) e precisar de uma verdadeira solução VPN, você deverá dar uma olhada no IPsec (Internet Protocol security). O IPsec foi originalmente desenvolvido para IPv6, mas também foi largamente implementado para IPv4. O IPsec é um sistema de criptografia e de autenticação ponta a ponta aprovado pelo IETF. Quase todos os fornecedores VPN sérios oferecem um produto que tenha ao menos um modo de compatibilidade IPsec.

O IPsec utiliza criptografia forte para fornecer serviços de autenticação, bem como de criptografia. A autenticação garante que os pacotes sejam provenientes do remetente correto e que não foram alterados em trânsito, e a criptografia impede o exame desautorizado do conteúdo dos pacotes.

Em sua forma atual, o IPsec criptografa o cabeçalho da camada de transporte, incluindo os números de portas de origem e de destino. Infelizmente, esse esquema entra diretamente em conflito com a maneira pela qual a maioria dos firewalls funciona. Uma proposta para desfazer esse recurso se encontra em andamento no IETF.

Os kernels 2.5.47 do Linux e os mais recentes incluem uma implementação IPSec nativa que é inteiramente diferente da implementação FreeS/WAN comumente utilizada com a série 2.4 do kernel. Como o IPSec faz parte do kernel, ele é incluído em todas as nossas distribuições.

Observe que há uma armadilha em torno de túneis IPSec e tamanho MTU. É importante garantir que, uma vez que um pacote tenha sido criptografado pelo IPsec, não se fragmente ao longo do caminho percorrido pelo túnel. Para realizar esse feito, pode ser necessário diminuir o MTU nos dispositivos em frente ao túnel (no mundo real, 1.400 bytes normalmente funcionam). Consulte a página 195 no capítulo sobre TCP para informações adicionais sobre o tamanho de MTU.

Tudo que preciso é de uma VPN, certo?

Infelizmente, há um lado negativo em relação às VPNs. Embora elas realmente construam um túnel (em grande parte) seguro na rede não confiável entre as duas extremidades, em geral, elas não se preocupam com a segurança destas extremidades em si. Se, por exemplo, configurarmos uma VPN entre um backbone corporativo e a residência do presidente da empresa, pode ser que estejamos criando inadvertidamente um caminho para a filha de 15 anos desse presidente, fazendo com que essa adolescente tenha acesso direto a tudo o que existe na rede corporativa. Com muita sorte, essa adolescente usará seu recém-adquirido acesso somente para marcar um encontro com um rapaz bonitinho da expedição.

Em suma: você precisa tratar as conexões de túneis VPN como conexões externas e conceder a elas privilégios adicionais apenas por absoluta necessidade e após cuidadosa consideração. Talvez seja interessante considerar o acréscimo de uma seção especial à política de segurança da instalação que abranja quais regras se aplicam a conexões VPN.

20.15 DISTRIBUIÇÕES LINUX APRIMORADAS

Consulte a página 479 para informações adicionais sobre SELinux.

Felizmente (?), fomos abençoados com uma série de iniciativas para produzir versões "aprimoradas" do Linux que fornecem uma gama mais ampla de recursos de segurança do que é encontrada nas principais versões. O aprimoramento normalmente assume a forma de controles de acesso especiais e de capacidades de auditoria. Esses recursos são particularmente úteis se você estiver planejando utilizar o Linux em algum tipo de produto projetado especificamente para redes. Entretanto, não está claro se eles fornecem vantagens substanciais aos usuários mais representativos da atualidade. Eles ainda requerem boa higiene, um bom filtro de pacotes e todas as demais coisas discutidas neste capítulo. Talvez sejam bons para uma paz de espírito extra.

A Tabela 20.3 lista alguns dos projetos de aprimoramento mais conhecidos que você poderá visitar para ver o que eles têm a oferecer.

Tabela 20.3 Distribuições Linux aprimoradas

Nome do projeto	Site Web
Bastille Linux	www.bastille-linux.org
Engarde Linux	www.engardelinux.com
Openwall GNU/*/Linux (Owl)	www.openwall.com/Owl

20.16 O QUE FAZER QUANDO SUA INSTALAÇÃO TIVER SIDO ATACADA

O segredo para tratar um ataque é simples: não se desespere. É bem provável que no momento em que você descobrir a invasão, a maior parte dos danos já terá sido feita. De fato, talvez isso já venha ocorrendo há semanas ou meses. A chance de você descobrir uma invasão que acabou de acontecer uma hora atrás é mínima ou nenhuma.

Sob essa perspectiva, a sábia coruja diz para respirar fundo e começar a desenvolver uma estratégia cuidadosamente planejada para lidar com a invasão. Você precisa evitar dar indícios ao invasor anunciando a invasão ou realizando qualquer atividade que poderia parecer anormal para alguém que está observando as operações de sua instalação há várias semanas. Dica: realizar um backup do sistema normalmente é uma boa idéia nesse ponto e (assim esperamos!) parecerá ser uma atividade normal ao invasor.[15]

Isso também serve para lembrá-lo de que alguns estudos revelam que 60% dos incidentes relacionados com segurança envolvem uma pessoa interna. Tome cuidado com quem você discute o incidente até ter certeza de todos os fatos.

Eis um plano de nove etapas que pode ajudá-lo neste momento de crise:

Passo 1: Não se desespere. Em muitos casos, um problema não é percebido a não ser horas ou dias depois que ele ocorreu. Algumas poucas horas ou dias a mais não afetarão o resultado. A diferença entre uma resposta de quem entrou em pânico e uma resposta racional, por outro lado, afetará o resultado. Muitas situações de recuperação são exacerbadas pela destruição de logs, condições e informações de rastreio importantes durante um pânico inicial.

Passo 2: Decida sobre o nível apropriado da resposta. Ninguém se beneficia de um incidente de segurança amplamente divulgado. Proceda de forma calma. Identifique o pessoal e os recursos que devem participar e deixe os demais ajudarem no *post-mortem* depois de tudo estar terminado.

Passo 3: Guarde como tesouro todas as informações disponíveis de rastreio. Verifique arquivos de contabilidade e de logs. Tente determinar onde a brecha original ocorreu. Faça o backup de todos os seus sistemas. Certifique-se de que protegeu fisicamente suas fitas de backup contra gravação se as colocar em uma unidade de fita para lê-las.

Passo 4: Avalie o seu grau de exposição. Determine que informações cruciais (se houver) "deixaram" a empresa e imagine uma estratégia de mitigação apropriada. Determine o nível de risco futuro.

Passo 5: Desconecte as máquinas. Se necessário e apropriado, desconecte da rede as máquinas comprometidas. Feche brechas conhecidas e interrompa a sangria. O CERT fornece passos para analisar uma invasão. O documento pode ser encontrado em

www.cert.org/tech_tips/win-UNIX-system_compromise.html

Passo 6: Elabore um plano de recuperação. Com um colega criativo, elabore um plano de recuperação em uma lousa próxima. Esse procedimento é mais eficiente quando realizado longe do teclado. Concentre-se em apagar o incêndio e em minimizar os danos. Evite esbravejar e ou criar alvoroço. Em seu plano, não se esqueça de abordar o abalo psicológico que

15. Se os backups de sistema não forem uma atividade "normal" em sua instalação, você terá problemas muito maiores do que a própria invasão.

sua comunidade de usuários possa experimentar. Os usuários confiam inerentemente em outros usuários e violações ofensivas à confiança tornam muitas pessoas inquietas.

Passo 7: Anuncie o plano de recuperação. Instrua os usuários e os gerentes sobre os efeitos da invasão, o potencial para futuros problemas e sua estratégia de recuperação preliminar. Seja franco e honesto. Os incidentes relacionados com segurança fazem parte da vida em um ambiente em rede. Eles não são reflexo de sua capacidade como administrador de sistemas ou qualquer outra coisa para você ficar constrangido. Admitir abertamente que você teve um problema representa 90% para ganhar uma batalha, desde que você consiga demonstrar que tem um plano para remediar a situação.

Passo 8: Implemente o plano de recuperação. Você conhece os seus sistemas e redes melhor do que ninguém. Siga seu plano e seu instinto. Fale com um colega de uma instituição similar (preferencialmente um que o conheça bem) para você se manter no caminho correto.

Passo 9: Relate o incidente às autoridades competentes. Se o incidente envolveu terceiros, você deve relatar o problema ao CERT. Eles têm uma linha direta, (412) 268-7090, e podem ser contatados por e-mail em cert@cert.org. Forneça o maior número de informações possível. O órgão brasileiro responsável por receber tais denúncias é o NBSO (www.nbso.nic.br). Nesse site há documentos indicando como proceder a reclamação, bem como a indicação de quais dados são necessários.

Um formulário padrão se encontra disponível em www.cert.org para ajudá-lo a refrescar sua memória. Eis algumas das informações úteis que você poderia fornecer:

- Os nomes, tipos de hardware e versões de sistema operacional das máquinas comprometidas
- A lista de correções que foram aplicadas no momento do incidente
- Uma lista das contas sabidamente comprometidas
- Os nomes e endereços IP e quaisquer hosts remotos que foram envolvidos
- Informações de contato (se você as tiver) para os administradores de instalações remotas
- Entradas de log relevantes ou informações de auditoria

Se você acredita que um problema de software anteriormente não documentado tenha sido envolvido, também deve informar o incidente ao distribuidor Linux.

20.17 Fontes de informação sobre segurança

Metade da batalha para manter um sistema seguro consiste em estar em compasso com os desenvolvimentos relacionados com a segurança como um todo. Se sua instalação sofrer uma invasão, isso provavelmente não ocorrerá pelo uso de uma técnica recente. É mais provável que a rachadura na sua armadura seja uma vulnerabilidade conhecida já amplamente discutida nas bases de conhecimento dos fornecedores, em newsgroups relacionados à segurança e em listas de discussão.

CERT: uma marca de serviços registrada da Carnegie Mellon University

Em resposta ao alvoroço sobre o worm Internet de 1988, a DARPA (Defense Advanced Research Projects Agency) formou uma organização chamada CERT (Computer Emergency Response Team) para atuar como uma alfândega para informações sobre segurança em sistemas computacionais. O CERT ainda é o mais conhecido ponto de contato para informações sobre segurança, embora ele pareça ter se tornado um tanto lento e, ultimamente, burocrático. O CERT agora também insiste que o nome CERT não significa nada mais do que uma simples "marca de serviços registrada da Carnegie Mellon University".

Em meados de 2003, o CERT estabeleceu uma parceria com o National Cyber Security Division, NCSD, do Department of Homeland Security. Essa parceria, para melhor ou para pior, alterou a estrutura anterior da lista de discussão. A combinação, conhecida como US-CERT, oferece quatro listas de anúncios; a mais útil é a "Technical Cyber Security Alerts". Você pode se inscrever em qualquer das quatro listas em forms.us-cert.gov/maillists.

SecurityFocus.com e a lista de discussão BugTraq

SecurityFocus.com é um site especializado em notícias e informações relacionadas com segurança. As notícias incluem artigos atuais sobre questões genéricas e sobre problemas específicos; há também uma imensa biblioteca de artigos técnicos úteis, agradavelmente organizados por tópico.

O arquivo de ferramentas de segurança do SecurityFocus inclui software para uma série de sistemas operacionais, juntamente com sinopses e avaliações de usuários. É a fonte mais abrangente e detalhada de ferramentas de que temos conhecimento.

A lista BugTraq é um fórum com moderador para a discussão de vulnerabilidades sobre segurança e suas correções. Para se inscrever, visite www.securityfocus.com/archive. Entretanto, o tráfego nessa lista pode ser relativamente pesado e a relação de sinal/ruído é relativamente ruim. Um banco de dados de relatórios de vulnerabilidade da BugTraq também está disponível no site Web.

A newsletter Crypto-Gram

O boletim mensal Crypto-Gram é uma fonte de informações valiosa e algumas vezes divertida, sobre criptografia e segurança em computadores. Ele é produzido por Bruce Schneier, autor dos respeitadíssimos livros *Applied Cryptography* e *Secrets and Lies*. Localize números atuais e anteriores neste site:

www.schneier.com/crypto-gram.html

Você também pode ler o blog sobre segurança de Schneier em

www.schneier.com/blog

SANS: o System Administration, Networking, and Security Institute

SANS é uma organização profissional que patrocina conferências e programas de treinamento relacionados à segurança, bem como publica uma série de informações relacionadas à segurança. O site da SANS, www.sans.org, é um recurso útil que ocupa uma posição intermediária entre o SecurityFocus e o CERT: não tão frenético como o primeiro nem tão enfadonho quanto o último.

A SANS oferece vários boletins semanais e mensais via e-mail para os quais você pode fazer uma assinatura no site da SANS. O semanal NewsBites é substancial, ao passo que os boletins mensais contêm muita propaganda. Nem um deles é uma excelente fonte de novidades sobre segurança.

Recursos de segurança específicos à distribuição

Pelo fato de os problemas de segurança terem potencial para gerar um bocado de má publicidade, normalmente os fornecedores são ávidos para ajudar os clientes a manterem seus sistemas seguros. A maioria dos grandes fornecedores possui uma lista de discussão oficial para a qual são postados boletins relacionados com segurança de sistemas e muitos também mantêm um site sobre questões relacionadas com segurança. É comum as correções de software relacionadas com segurança serem distribuídas gratuitamente, mesmo por parte de fornecedores que em geral cobram pelo suporte de software.

Portais de segurança na Web, como www.securityfocus.com, contêm informações específicas ao fornecedor e links para o dogma oficial mais recente dos fornecedores.

Uma lista de boletins de segurança do Red Hat pode ser encontrada em www.redhat.com/security. Na época em que este livro era escrito, não havia nenhuma lista de discussão oficial sobre segurança patrocinada pelo Red Hat. Entretanto, há uma série de recursos sobre segurança de sistemas Linux na rede, cuja grande parte se aplica diretamente ao Red Hat.

Você pode encontrar boletins de segurança sobre o SUSE em

www.novell.com/linux/security/securitysupport.html

Você pode se associar à lista de discussão oficial do SUSE sobre anúncios de segurança visitando

www.suse.com/en/private/support/online_help/mailinglists/index.html

Verifique www.debian.org para ver as notícias mais recentes sobre segurança do Debian ou assine lista de discussão em

www.debian.org/MailingLists/subscribe#debian-security-announce

O Ubuntu tem uma lista de discussão sobre segurança em

https://lists.ubuntu.com/mailman/listinfo/ubuntu-security-announce

Informações de segurança sobre os produtos da Cisco são distribuídas na forma de avisos de campo, uma lista que pode ser localizada em

www.cisco.com/public/support/tac/fn_index.html

junto com um RSS de agregação. Para se inscrever na lista de discussão sobre segurança da Cisco, envie um e-mail para majordomo@cisco.com com a linha "subscribe cust-security-announce" no corpo da mensagem.

Outras listas de discussão e sites Web

Os contatos listados anteriormente são apenas alguns dos numerosos recursos relacionados com segurança disponíveis na rede. Dado o volume de informações que agora se encontra à disposição e à rapidez com a qual os recursos surgem e desaparecem, achamos mais útil indicar alguns meta-recursos.

Um bom ponto de partida é o site X-Force (xforce.iss.net) na Internet Security Systems, que mantém uma série de FAQs úteis. Uma delas é a lista atualizada de listas de discussão relacionadas com segurança. As FAQs sobre correções de segurança e fornecedores contêm informações de contato úteis para uma série de fabricantes.

www.yahoo.com tem uma extensa lista de links de segurança; procure a seção"Security and Encryption" no Yahoo! Directory. Uma outra boa fonte de links sobre segurança de rede pode ser localizada em www.wikipedia.org sob o título "computer security".

Linux Journal (www.linuxjournal.com) contém uma excelente coluna chamada "Paranoid Penguin" que abrange todos os aspectos da segurança Linux. Essa revista ocasionalmente também inclui vários artigos sobre tópicos de segurança.

O Linux Weekly News é um excelente debate que inclui atualizações regulares sobre o kernel, segurança, distribuições e outros tópicos. A seção sobre segurança do LWN pode ser localizada em lwn.net/security.

20.18 LEITURA RECOMENDADA

BRYANT, WILLIAM. *Designing an Authentication System*: a Dialogue in Four Scenes. 1988. Disponível em: <web.mit.edu/kerberos/www/dialogue.html>.

CERT COORDINATION CENTER. *Intruder Detection Checklist*. 1999. Disponível em: <www.cert.org/tech_tips/intruder_detection_checklist.html>.

CERT COORDINATION CENTER. *UNIX Configuration Guidelines*. 1997. Disponível em: <www.cert.org/tech_tips/unix_configuration_guidelines.html>.

CHESWICK, WILLIAM R.; BELLOVIN, STEVEN M. E RUBIN, AVIEL D. *Firewalls and Internet Security:* Repelling the Wily Hacker. *2. ed.* Reading, MA: Addison-Wesley, 2000.

CURTIN, MATT; RANUM, MARCUS E ROBINSON, PAUL D. *Internet Firewalls: Frequently Asked Questions*. 2004. Disponível em: <www.interhack.net/pubs/fwfaq>.

FARMER, DAN E VENEMA, WIETSE. *Improving the Security of Your Site by Breaking Into it*. 1993. Disponível em: <www.deter.com/unix/papers/improve_by_breakin.html>.

FARROW, RIK E POWER, RICHARD. *Network Defense article series*. 1998-2004. Disponível em: <www.spirit.com/Network>.

FRASER, B. (EDITOR). *RFC2196: Site Security Handbook*. 1997. Disponível em: <www.rfc-editor.org>.

BAUER, MICHAEL D. *Linux Server Security. 2. ed.* Sebastopol, CA: O'Reilly Media, 2005.

GARFINKEL, SIMSON E SPAFFORD, GENE E SCHWARTZ, ALAN. *Practical UNIX and Internet Security. 3. ed.* Sebastopol, CA: O'Reilly Media, 2003.

BARRETT, DANIEL J; SILVERMAN, RICHARD E. E BYRNES, ROBERT G. *Linux Security Cookbook*. Sebastopol, CA: O'Reilly Media, 2003.

KERBY, FRED, ET AL. *SANS Intrusion Detection and Response FAQ*. SANS. 2003. Disponível em: <www.sans.org/resources/idfaq/>.

MANN, SCOTT E MITCHELL, ELLEN L. *Linux System Security:* The Administrator's Guide to Open Source Security Tools. *2. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2002.

MORRIS, ROBERT E THOMPSON, KEN. "Password Security: A Case History". Communications of the ACM, 22 (11): 594-597, November 1979. Reprinted in *UNIX System Manager's Manual*, 4.3 Berkeley Software Distribution. University of California, Berkeley, April 1986.

PICHNARCZYK, KARYN; WEEBER, STEVE E FEINGOLD, RICHARD. *UNIX Incident Guide: How to Detect an Intrusion*. Computer Incident Advisory Capability, U.S. Department of Energy, 1994. Disponível em: <www.ciac.org/cgi-bin/index/documents>.

RITCHIE, DENNIS M. *On the Security of UNIX*." May 1975. Reprinted in *UNIX System Manager's Manual*, 4.3 Berkeley Software Distribution. University of California, Berkeley, April 1986.

SCHNEIER, BRUCE. *Applied Cryptography:* Protocols, Algorithms, and Source Code in C. New York, NY: Wiley, 1995.

THOMPSON, KEN. *Reflections on Trusting Trust*. in *ACM Turing Award Lectures: The First Twenty Years 1966-1985*. Reading, MA: ACM Press (Addison-Wesley), 1987.

SONNENREICH, WES E YATES, TOM . *Building Linux and OpenBSD Firewalls*. New York, NY: J.W. Wiley, 2000.

Trata-se de um livro pequeno impressionante: tem uma leitura fácil, apresenta bons exemplos, tem bom senso de humor e no geral é simplesmente excelente. Nossa única restrição em relação a esse livro é o fato de ser contra o emprego de **sudo** para acesso root, alegando que ele é muito difícil de ser empregado e de não valer a pena a dor de cabeça. Nós discordamos redondamente.

20.19 Exercícios

E20.1 Discuta o poder da autenticação SSH com senhas Linux *versus* autenticação SSH com um par senha e chave. Se uma delas for claramente mais segura do que a outra, você solicita automaticamente o método de autenticação mais seguro?

E20.2 Samhain identifica arquivos que foram alterados.

a) O que é necessário para configurar e usar samhain em sua máquina?

b) Contra quais males recentes da Internet samhain seria eficiente?

c) Para que problemas recentes da Internet samhain seria inútil?

d) Dado o acesso físico a um sistema, como samhain poderia ser evitado?

e) À qual conclusão chegar se samhain diz que **/bin/login** foi alterado, mas parece ter o mesmo tamanho e data de modificação? O que acontece se o programa **sum** der os mesmos resultados para as versões antiga e nova? Que tal **md5sum**?

★ E20.3 O tunelamento SSH normalmente é a única maneira de se tunelar tráfego para uma máquina remota na qual você não tem acesso como administrador. Leia a página man **ssh** e forneça uma linha de comando que faça um túnel para o tráfego da porta host local 113 para a porta 113 de mail.remotenetwork.org. O ponto de encaminhamento de seu túnel também deve ser o host mail.remotenetwork.org.

★ E20.4 Escolha um incidente de segurança recente e pesquise-o. Localize as melhores fontes de informação sobre o incidente e encontre correções ou meios de evitar que sejam apropriados aos seus sistemas. Liste as fontes e as ações propostas para proteger o seu sistema.

★ ★ E20.5 Com permissão de seu grupo de administrador de sistemas local, instale John the Ripper, o programa que procura logins com senhas frágeis.

a) Modifique o código-fonte para que ele gere a saída apenas dos nomes de login com os quais as senhas frágeis estejam associadas, não com as senhas em si.

b) Execute o John the Ripper no arquivo de senha do seu laboratório local (você precisa acesso a **/etc/shadow**) e veja quantas senhas frágeis você consegue localizar.

c) Configure sua própria senha para uma palavra de dicionário e dê a **john** simplesmente sua própria entrada em **/etc/shadow**. Quanto tempo **john** leva para descobri-la?

d) Tente outros padrões (letras maiúsculas, número após a palavra de dicionário, senha de uma só letra etc.) para ver exatamente quão inteligente **john** é.

★ ★ E20.6 No laboratório de computação, configure duas máquinas: um alvo e um sondador.

a) Instale **nmap** e Nessus na sonda. Ataque o alvo com essas ferramentas. Como você poderia detectar o ataque no alvo?

b) Configure um firewall no alvo usando **iptables** para se defender contra as sondas. Você é capaz de detectar o ataque agora? Se for, como? Se não for, por que não?

c) Que outras defesas podem ser configuradas contra os ataques?

(Exige acesso de root.)

★ ★ E20.7 Uma equipe de segurança localizou recentemente uma grande brecha em diversos servidores **sshd** antigos e recentes. Localize uma boa fonte de informações sobre a brecha e discuta as questões e a melhor maneira de enfrentá-las.

★ ★ E20.8 Programas setuid são algumas vezes um mal necessário. Entretanto, deve-se evitar aplicar setuid a scripts de shell. Por quê?

★★E20.9 Utilizee **tcpdump** para capturar tráfego FTP tanto para sessões FTP ativas quanto passivas. Como a necessidade de suportar um servidor FTP anônimo afeta a política de firewall de um site? O que as regras de firewall precisariam permitir? (Exige acesso root.)

★★E20.10 O que as regras na seguinte saída gerada por **iptables** permitem ou não? Quais seriam algumas adições muito fáceis que ajudariam a otimizar a segurança e privacidade? (Dica: as cadeias `OUTPUT` e `FORWARD` poderiam utilizar algumas outras regras.)

```
Chain INPUT (policy ACCEPT)
target     prot   opt    source destination
block      all    --     anywhere       anywhere

Chain FORWARD (policy ACCEPT)
target     prot   opt    source destination
           all    --     anywhere       anywhere

Chain OUTPUT (policy ACCEPT)
target     prot   opt    source destination

Chain block (1 references)
target     prot   opt    source destination
ACCEPT     all    --     anywhere       anywhere       state RELATED,ESTABLISHED
ACCEPT     tcp    --     anywhere       anywhere       state NEW tcp dpt:www
ACCEPT     tcp    --     anywhere       anywhere       state NEW tcp dpt:ssh
ACCEPT     tcp    --     128.138.0.0/16 anywhere       state NEW tcp dpt:kerberos
ACCEPT     icmp   --     anywhere       anywhere
DROPall    --     anywhere       anywhere
```

★★E20.11 Inspecione os conjuntos de regras do firewall local. Discuta o que você encontrar em termos de políticas. Há algumas brechas de segurança evidentes? (Este exercício pode exigir a cooperação dos administradores responsáveis pela segurança de seu site local.)

★★★★★E20.12 Crie uma ferramenta que determine se qualquer interface de rede em sua instalação está em modo promíscuo. Execute-a regularmente em suas redes para localizar rapidamente tal intrusão. Que nível de carga a ferramenta gera? Você tem de executá-la em cada máquina ou poderá executá-la remotamente? Você é capaz de criar um pacote sorrateiro que pudesse informar se uma interface estava em modo promíscuo? (Exige acesso root.)

21 Hospedagem Web e servidores Internet

A complexidade da tecnologia Web parece estar dobrando a cada ano. Felizmente, a vasta maioria dessa tecnologia reside no domínio do designer e programador Web. A hospedagem Web em si não mudou muito nas últimas décadas.

Os principais problemas foram eliminados dos softwares de servidor Web e, como resultado, esses servidores são agora bem seguros e confiáveis — pelo menos se estiverem configurados corretamente e se sua instalação não tiver programadores Web trapaceiros. Mesmo com o advento da "Web 2.0", do AJAX (Asynchronous JavaScript And XML) e do HTML dinâmico, o software de servidor Web básico permanece o mesmo.

Hoje em dia, há uma variedade de plataformas de hospedagem Web para escolha. O Microsoft Windows foi amplamente vendido como uma plataforma de hospedagem Web. A imprensa especializada publicou inúmeros artigos que perguntavam: "Qual é a melhor plataforma para hospedagem Web?"; comumente, o Windows e o Linux se posicionavam em cantos opostos do ringue. Embora essa seja uma discussão sem-fim, o Linux tornou-se a plataforma de hospedagem mais popular por causa de seu baixo custo, velocidade, confiabilidade e flexibilidade. A chamada plataforma LAMP (Linux, Apache, MySQL e PHP/Perl/Python) é o paradigma dominante para os servidores Web atuais.

Há muitos serviços centrados na Internet que você pode querer "hospedar" em sua instalação ou em um dos muitos provedores de terceirização de espaço em servidor Web. Neste capítulo, tratamos dos dois serviços mais comuns: Web e FTP.

21.1 Fundamentos sobre hospedagem Web

Hospedar um site Web não é substancialmente diferente de fornecer qualquer outro serviço de rede. A fundação da World Wide Web é o Hypertext Transfer Protocol (HTTP), um protocolo baseado em TCP simples para transmitir documentos que contêm uma variedade de tipos de mídias, incluindo texto, figuras, som, animação e vídeo. O HTTP se comporta de forma muito semelhante aos demais protocolos cliente/servidor utilizados na Internet, por exemplo, o SMTP (para e-mail) e o FTP (para transferência de arquivos).

Um servidor Web nada mais é que um sistema configurado para responder a requisições HTTP. Para converter seu sistema Linux genérico em uma plataforma para hospedagem Web, você instala um daemon que ouve conexões na porta TCP 80 (o padrão HTTP), aceita requisições de documentos e os transmite ao usuário solicitante.

Navegadores Web como o Firefox, o Opera e o Internet Explorer contatam servidores Web remotos e fazem requisições em nome dos usuários. Os documentos assim obtidos podem conter ponteiros-hipertexto (links) para outros documentos que podem ou não residir no servidor que o usuário constatou originalmente. Como o padrão de protocolo HTTP é bem definido, clientes utilizando qualquer sistema operacional ou arquitetura poderão se conectar ao servidor HTTP. Essa independência de plataforma, juntamente com a capacidade do HTTP de passar um usuário de um servidor a outro de forma transparente, ajudou a ampliar seu sucesso.

Mas há vida além do HTTP puro e simples. Muitos protocolos aprimorados foram definidos para tratar tudo, desde criptografia até vídeo streaming. Esses serviços adicionais são normalmente controlados por daemons distintos, mesmo quando são fornecidos pelo mesmo servidor físico.

Localizadores de recursos uniformes

Um localizador de recursos uniformes (*uniform resource locator* – URL) é um ponteiro para um objeto ou serviço na Internet. Ele descreve como acessar um objeto por meio de cinco componentes básicos:

- Protocolo ou aplicativo
- Nome do host
- Porta TCP/IP (opcional)
- Diretório (opcional)
- Nome de arquivo (opcional)

A Tabela 21.1 mostra alguns protocolos que podem ser utilizados em URLs.

Tabela 21.1 Protocolos URL

Protocolo	O que ele faz	Exemplo
file	Acessa um arquivo local	file://etc/syslog.conf
ftp	Acessa um arquivo remoto via FTP	ftp://ftp.admin.com/adduser.tar.gz
http	Acessa um arquivo remoto via HTTP	http://admin.com/index.html
https	Acessa um arquivo remoto via HTTP/SSL	https://admin.com/order.shtml
ldpa	Acessa serviços de diretórios LDAP	ldap://ldap.bigfoot.com:389/cn=Herb
mailto	Envia e-mail para um endereço designado	mailto:linux@book.admin.com

Como o HTTP funciona

O HTTP é o protocolo que faz com que a World Wide Web realmente funcione e, para a alegria de muitos, ele é um protocolo cliente/servidor sem informações de estado e extremamente elementar. No paradigma HTTP, o iniciador de uma conexão sempre é um cliente (normalmente um navegador). O cliente requisita ao servidor o "conteúdo" de um determinado URL. O servidor responde com um fluxo de dados ou com algum tipo de mensagem de erro. O cliente então pode continuar requisitando outro objeto.

Como o HTTP é tão simples, você pode facilmente se transformar em um navegador Web rudimentar utilizando **telnet**. Como a porta padrão para o serviço HTTP é a porta 80, basta enviar um **telnet** diretamente a essa porta no servidor Web de sua preferência. Assim que estiver conectado, você pode emitir comandos HTTP. O comando mais comum é GET, que requisita o conteúdo de um documento. Normalmente, GET / é o que você deseja, já que ele solicita o documento-raiz (normalmente a home page) de qualquer que seja o servidor a que você se conectou. O HTTP diferencia letras maiúsculas de minúsculas; portanto, você deve se certificar de que digitou os comandos em letras maiúsculas.

```
$ telnet localhost 80
Trying 127.0.0.1…
Connected to localhost.atrust.com.
Escape character is '^]'.
GET /
<conteúdo de seu arquivo padrão aparece aqui>
Connection closed by foreign host.
```

Uma requisição HTTP mais "completa" incluiria a versão de protocolo HTTP, o host a que se destina a requisição (necessário para recuperar um arquivo de um host virtual baseado em nome) e outras informações. A resposta então incluiria cabeçalhos informacionais bem como dados de resposta. Por exemplo:

```
$ telnet localhost 80
Trying 127.0.0.1…
Connected to localhost.atrust.com.
Escape character is '^]'.
GET / HTTP/1.1
Host: www.atrust.com

HTTP/1.1 200 OK
Date: Sun, 06 Aug 2006 18:25:03 GMT
Server: Apache/1.3.33 (Unix) PHP/4.4.0
Last-Modified: Sun, 06 Aug 2006 18:24:49 GMT
Content-Length: 7044
Content-Type: text/html

<conteúdo de seu arquivo padrão aparece aqui>
Connection closed by foreign host.
```

Nesse caso, dissemos ao servidor que falaríamos o protocolo HTTP versão 1.1 e identificamos o host virtual do qual solicitávamos as informações. O servidor retornou um código de status (HTTP/1.1 200 OK), sua idéia da data e hora atual, o nome e a versão do software de servidor que estava executando, a data em que o arquivo solicitado foi modificado pela última vez, o tamanho do arquivo solicitado e o tipo de conteúdo do arquivo solicitado. As informações de cabeçalho são separadas do conteúdo por uma única linha em branco.

Geração de conteúdo no processo

Além de servir documentos estáticos, um servidor HTTP é capaz de fornecer ao usuário o conteúdo que é criado imediatamente no processo. Por exemplo, se quisesse fornecer a hora e temperatura atuais aos usuários que estivessem visitando o seu site Web, você poderia fazer o servidor HTTP executar um script para obter essas informações. Esse truque do tipo "agrade os nativos" é normalmente realizado pela Common Gateway Interface (CGI).

A CGI não é uma linguagem de programação, mas, mais exatamente, uma especificação pela qual um servidor HTTP troca informações com outros programas. Os scripts CGI são escritos mais freqüentemente em Perl, Python ou PHP. Mas, na realidade, quase todas as linguagens de programação que são capazes de realizar E/S em tempo real são aceitáveis. Pense em todos aqueles programadores COBOL sem trabalho que podem aplicar suas habilidades na Internet!

Além de suportar scripts CGI externos, muitos servidores Web definem uma arquitetura de plug-ins que permite que interpretadores de script como Perl e PHP sejam embutidos no próprio servidor Web. Essa inclusão no pacote aumenta o desempenho significativamente, uma vez que o servidor Web não mais precisa gerar um processo separado para lidar com cada requisição de script. A arquitetura é amplamente invisível aos desenvolvedores de script. Sempre que o servidor vir um arquivo que termina com uma extensão especificada (como **.pl** ou **.php**), ele envia o conteúdo do arquivo a um interpretador embutido para ser executado.

Em sua maioria, scripts CGI e plug-ins são preocupação dos desenvolvedores e programadores Web. Infelizmente, eles colidem com a tarefa do administrador de sistemas em uma importante área: segurança. Como os scripts CGI e plug-ins têm acesso a arquivos, conexões de rede e outros métodos de transferência de dados de um local a outro, sua execução pode afetar potencialmente a segurança da máquina na qual o servidor HTTP está sendo executado. Em última instância, um script CGI ou um plug-in dá a qualquer um a capacidade de executar um programa (o script) em seu servidor. Portanto, os scripts CGI e os arquivos processados por plug-ins devem ser tão seguros como qualquer outro programa acessível pela rede. Uma boa fonte de informações sobre o tratamento seguro de scripts CGI é a página www.w3.org/Security/Faq. Embora essa página não seja atualizada há algum tempo, todas as suas informações ainda são relevantes.

Balanceamento de carga

É difícil prever quantos hits (requisições de objetos, incluindo imagens) ou visualizações de página (requisições de páginas HTML) um servidor pode tratar por unidade de tempo. A capacidade de um servidor depende da arquitetura de hardware do sistema (incluindo subsistemas), o sistema operacional que ele executa, a extensão e a ênfase de qualquer ajuste de sistema que foi realizado e talvez, acima de tudo, a construção dos sites sendo servidos. (Eles contêm apenas páginas HTML estáticas ou devem fazer chamadas de banco de dados e cálculos numéricos?)

Somente comparações e medidas diretas do site real executando no hardware real são capazes de responder à pergunta "quantos acessos?". Às vezes, as pessoas que já construíram sites semelhantes em hardwares semelhantes podem fornecer informações úteis para planejamento. Em nenhum caso você deve acreditar nos números informados pelos fornecedores de sistemas. Além disso, lembre-se de que sua largura de banda é uma das princiapais considerações. Uma única máquina que atende a arquivos HTML estáticos e imagens pode facilmente servir dados suficientes para saturar um link T3 (45 Mb/s).

Dito isso, em vez de contagens de hits de único servidor, um parâmetro melhor a focalizar é a escalabilidade; normalmente um servidor Web torna-se limitado pela CPU ou pela ES (entrada e saída) antes de saturar sua interface Ethernet. Certifique-se de que você e sua equipe de design Web planejam distribuir a carga de um site com intenso tráfego por múltiplos servidores.

O balanceamento de carga adiciona tanto desempenho como redundância. Várias abordagens diferentes de balanceamento de carga estão disponíveis: DNS *round robin* ("rodízio"), hardware de balanceamento de carga e balanceadores de carga baseados em software.

Consulte a página 268 para informações adicionais sobre a configuração de DNS round robin.

O DNS round robin é a forma mais simples e mais primitiva de balanceamento de carga. Nesse sistema, vários endereços IP são atribuídos a um único hostname. Quando uma requisição do endereço IP do site Web chega ao servidor de nomes, o cliente recebe um dos endereços IP em resposta. Os endereços são distribuídos um depois de outro, em uma seqüência "round robin" repetida.

O problema com DNS round robin é que, se um servidor pára de funcionar, os dados de DNS devem ser atualizados para remover o servidor do ciclo de resposta. O cache remoto de dados de DNS pode tornar essa operação difícil e incerta. Se você tiver um servidor de backup disponível, costuma ser mais fácil reatribuir o endereço IP do servidor desativado ao servidor de backup.

O hardware de balanceamento de carga é uma alternativa relativamente fácil, mas que requer alguma reserva de dinheiro. O hardware de balanceamento de carga independente comercial inclui a Big-IP Controller da F5 Networks, produtos de switching da Nortel e os Content Services Switches da Cisco. Esses produtos distribuem o trabalho que chega de acordo com uma variedade de parâmetros configuráveis e podem levar em conta os tempos de resposta atual de servidores individuais.

Os balanceadores de carga baseados em software não exigem hardware especializado; eles podem executar em um servidor Linux. Há soluções tanto de código-fonte aberto como comerciais. A categoria código-fonte aberto inclui o Linux Virtual Server (www.linuxvirtualserver.org), o Ultra Monkey (www.ultramonkey.org) e o módulo **mod_backhand** para Apache (www.backhand.org). Um exemplo de ofertas comerciais são as vendidas por Zeus, www.zeus.com.

Você pode se perguntar como um site grande como o Google trata do balanceamento de carga. Seu sistema utiliza uma combinação de servidores de DNS de balanceamento de carga personalizados e hardware de balanceamento de carga. Consulte o artigo da Wikipedia "Google platform" para detalhes adicionais.

Tenha em mente que a maioria dos sites atuais é gerada dinamicamente. Essa arquitetura coloca uma carga pesada em servidores de banco de dados. Se necessário, consulte seu administrador de banco de dados para determinar a melhor maneira de distribuir a carga por vários servidores de banco de dados.

21.2 INSTALAÇÃO DE SERVIDORES HTTP

Instalar e manter um servidor Web é fácil. Os serviços Web estão muito abaixo de e-mail e DNS em termos de complexidade e dificuldade de administração.

Escolhendo um servidor

Vários servidores HTTP estão disponíveis, mas é mais provável que você queira começar com o servidor Apache, que é bem conhecido na indústria por sua flexibilidade e desempenho. Em setembro de 2006, 63% dos servidores Web da Internet utilizavam o Apache. A Microsoft é responsável pela maior parte dos 30% de servidores restantes. Essa divisão no mercado entre o Apache e a Microsoft tem sido relativamente estável nos últimos cinco anos. Estatísticas mais detalhadas do compartilhamento de mercado no decorrer do tempo estão disponíveis aqui:

news.netcraft.com/archives/web_server_survey.html

Você pode encontrar uma comparação de servidores HTTP útil atualmente disponível no site www.serverwatch.com/stypes/index.php (selecione "Web Servers"). Eis alguns fatores que podem ser importantes ao fazer sua seleção:

- Robustez
- Desempenho
- Tempo de espera para atualizações e correções de bugs
- Disponibilidade de código-fonte
- Nível de suporte comercial ou da comunidade
- Custo
- Controle de acesso e segurança
- Capacidade de atuar como um proxy
- Disponibilidade de tratar criptografia

O servidor HTTP Apache é "livre para ser adotado" e o código-fonte completo está disponível no site do Apache Group em www.apache.org. Os menos aventureiros talvez queiram instalar o pacote binário do Apache que faz parte das distribuições Linux. (Mas é provável que já esteja instalado; tente examinar em **/etc/apache2**.)

Instalando o Apache

Se decidir pelo download do código-fonte Apache e compilá-lo você mesmo, comece executando o script **configure** incluído com a distribuição. Esse script detecta automaticamente o tipo de sistema e configura os makefiles apropriados. Utilize a opção **--prefix** para especificar onde em sua árvore de diretórios o servidor Apache deve residir. Se você não especificar um prefixo, o servidor é instalado em **/usr/local/apache2** por padrão. Por exemplo:

```
$ ./configure --prefix=/etc/httpd/
```

Você pode utilizar **configure --help** para examinar a lista inteira de opções possíveis, a maioria das quais consiste nas opções **--enable-***módulo* e **--disable-***módulo*, que incluem ou excluem vários componentes funcionais residindo no servidor Web.

Você também pode compilar os módulos em arquivos de objetos compartilhados dinamicamente especificando a opção **--enable-***módulo***=shared** (ou utilizar **--enabled-mods-shared=all** para fazer todos módulos compartilhados). Dessa maneira, você pode decidir mais tarde que módulos incluir ou excluir; somente os módulos especificados em sua configuração **httpd** são carregados em tempo de execução. Essa é realmente a configuração padrão para o pacote somente de binários do Apache — todos os módulos são compilados em objetos compartilhados e dinamicamente carregados quando o Apache é inicializado. As únicas desvantagens do uso de bibliotecas compartilhadas são um tempo de inicialização ligeiramente mais longo e uma degradação muito leve no desempenho (em geral menor que 5%). Para a maioria dos sites, o benefício de poder adicionar novos módulos durante a execução e desativar módulos existentes sem ter de fazer a recompilação compensa a leve queda de desempenho.

Para uma lista completa dos módulos padrão, consulte httpd.apache.org/docs-2.0/mod.

Embora o conjunto padrão de módulos seja razoável, talvez você queira habilitar os módulos mostrados na Tabela 21.2.

Tabela 21.2 Módulos Apache úteis que não são ativados por padrão

Módulo	Função
auth_dbm	Utiliza um banco de dados DBM para gerenciar o acesso de usuários/grupos (recomendado se você precisar de acesso baseado em senha por usuário para áreas de seu site Web)
rewrite	Regrava URLs com expressões regulares
expires	Permite que você anexe datas de expiração a documentos
proxy	Utiliza o Apache como um servidor proxy (mais informações adiante)
ssl	Ativa o suporte a Secure Sockets Layer (SSL) (para HTTPS)

Da mesma forma, talvez você queira desativar os módulos listados na Tabela 21.3. Para fins de segurança e de desempenho, é uma boa idéia desativar módulos que você sabe que não utilizará.

Tabela 21.3 Módulos Apache que sugerimos remover

Módulo	Função
asis	Permite que tipos de arquivos designados sejam enviados sem cabeçalhos HTTP
autoindex	Exibe o conteúdo de diretórios que não têm um arquivo HTML padrão
env	Permite que você configure variáveis de ambiente especiais para scripts CGI
include	Permite inclusões no servidor, um esquema obsoleto de geração dinâmica de páginas
userdir	Permite que usuários tenham seus próprios diretórios HTML

Quando **configure** tiver finalizado a execução, execute **make** e, a seguir, **make install** para realmente compilar e instalar os arquivos apropriados.

Configurando o Apache

Assim que você instalar o servidor, configure-o para o seu ambiente. Os arquivos de configuração são mantidos no subdiretório **conf** (por exemplo, **/usr/local/apache2/conf**). Examine e personalize o arquivo **httpd.conf**, que está dividido em três seções.

A primeira seção lida com configurações globais como o pool de servidores, a porta TCP em que o servidor HTTP ouve consultas (normalmente a porta 80, embora você possa escolher outra — e sim, você pode executar vários servidores HTTP em diferentes portas em uma única máquina) e as configurações para carregamento de módulo dinâmico.

A segunda seção configura o servidor "padrão", o servidor que trata de quaisquer requisições que não são respondidas pelas definições `VirtualHost` (consulte a página 503). Os parâmetros de configuração nessa seção incluem o usuário e o grupo com o qual o servidor executará (algo diferente de root!) e a importante instrução `DocumentRoot`, que define a raiz da árvore de diretórios a partir da qual documentos são atendidos. Essa seção também aborda questões como o tratamento de URLs "especiais" como aqueles que incluem a sintaxe *~user* para acessar o diretório inicial de um usuário.

Você também gerencia interesses de segurança global na segunda seção do arquivo de configuração. As diretivas controlam acesso por arquivo (a diretiva `<File>`) ou por diretório (a diretiva `<Directory>`). Essas configurações de permissão impedem acesso a arquivos sensíveis por meio de **httpd**. Você deve especificar ao menos dois controles de acesso: um que abrange o sistema de arquivos inteiro e um que se aplica à pasta principal de documento. Os padrões que vêm com o Apache são suficientes, embora recomendemos a remoção da opção `AllowSymLinks` para impedir que **httpd** siga links simbólicos na árvore de documento. (Não gostaríamos que alguém criasse acidentalmente um link simbólico para **/etc**, não é?) Para dicas adicionais sobre a segurança do Apache, consulte

httpd.apache.org/docs-2.0/misc/security_tips.html

A terceira e última seção do arquivo de configuração define os hosts virtuais. Discutimos esse tópico em mais detalhe a partir da página 502.

Assim que você fizer suas alterações de configuração, verifique a sintaxe do arquivo de configuração executando **httpd -t**. Se o Apache relatar "Sintax OK", então você pode prosseguir. Se não, verifique se há erros de digitação no arquivo **httpd.conf**.

Executando o Apache

Você pode iniciar **httpd** manualmente ou por meio de um dos scripts **rc** do seu sistema. A última forma é preferível já que essa configuração assegura que o servidor Web reinicialize toda vez que a máquina for reinicializada. Para iniciar o servidor manualmente, digite algo como:

```
% /usr/local/apache2/bin/apachectl start
```

Consulte o Capítulo 2 para informações adicionais sobre os scripts rc.

Caso queira iniciar **httpd** automaticamente no momento da inicialização, crie um link em seu diretório **rc** que aponte para o arquivo **/etc/init.d/httpd** (que se encontra instalado como parte do pacote **httpd**). É melhor iniciar **httpd** posteriormente na seqüência de inicialização, depois de os daemons que controlam funções como roteamento e sincronização de tempo tiverem sido iniciados.

Analisando arquivos de log

Com seu site Web em produção, é provável que você queira reunir informações estatísticas sobre o uso do site, como o número de requisições por página, o número médio de requisições por dia, a porcentagem de requisições que falharam e a quantidade de dados transferidos. Certifique-se de que está utilizando o formato de log "combinado" (suas diretivas `CustomLog` têm a palavra `combined` no fim em vez da palavra `common`). O formato combinado de log inclui a referrer de cada requisição (a página a partir da qual o URL foi vinculado) e agente de usuário (o navegador e sistema operacional do cliente).

Os logs de acesso e de erro aparecem no diretório **logs** do Apache. Os arquivos são legíveis, mas contêm tantas informações que você realmente precisa de um programa de análise separado para extrair dados úteis deles. Há literalmente centenas de analisadores de log diferentes, tanto gratuitos/livres como comerciais.

Dois analisadores gratuitos/livres que merecem ser examinados são Analog (www.analog.cx) e AWStats (awstats.sourceforge.net). Esses dois analisadores fornecem informações relativamente básicas. Se quiser relatórios com um pouco mais de sofisticação, você pode precisar de pacotes comerciais. Uma lista útil pode ser encontrada em www.practicalapplications.net/kb/loganalysis.html.

Otimizando para hospedagem de alto desempenho do conteúdo estático

A comunidade de hospedagem aprendeu nos últimos poucos anos que uma das maneiras mais fáceis de criar uma plataforma de hospedagem de alto desempenho é otimizar alguns servidores para hospedar conteúdo estático. O Linux oferece funcionalidade única nessa arena por meio do servidor Web TUX.

O TUX é um servidor Web baseado em kernel que é executado em conjunto com um servidor Web tradicional como o Apache. Sempre que possível, o TUX serve páginas estáticas sem jamais deixar o espaço do kernel, de maneira muito semelhante a como o **rpc.nfsd** serve arquivos. Essa arquitetura elimina a necessidade de copiar dados entre o kernel e o espaço de usuários e minimiza o número de trocas de contexto. O TUX não é recomendado para iniciantes, mas é uma escolha excelente para sites que devem fornecer conteúdo estático a uma velocidade relâmpago.

Embora o TUX tenha sido desenvolvido pela Red Hat (e se encontre disponível em www.redhat.com), ele foi lançado sob o GPL e pode ser utilizado com outras distribuições do Linux. Mas configurar o TUX pode ser um pouco desafiador. Para detalhes, visite:

www.redhat.com/docs/manuals/tux

21.3 INTERFACES VIRTUAIS

No início da Internet, normalmente uma máquina atuava como o servidor para um único site Web (por exemplo, www.acme.com). À medida que a popularidade da Web cresceu, todo o mundo queria ter um site Web e, da noite para o dia, milhares de empresas se transformaram em provedores de hospedagem Web.

Os provedores rapidamente se deram conta de que eles poderiam ter uma grande economia de escala caso fossem capazes de hospedar mais de um site em um único servidor. Esse truque permitiria que www.acme.com, www.ajax.com, www.toadranch.com e muitos outros sites fossem servidos transparentemente pelo mesmo hardware. Em resposta a essa necessidade comercial nasceram as interfaces virtuais.

As interfaces virtuais permitem que um daemon identifique conexões baseadas não somente no número da porta de destino (por exemplo, porta 80 para HTTP), como também no endereço IP de destino da conexão. Hoje em dia, as interfaces virtuais são utilizadas amplamente e se provaram úteis para outros aplicativos além da hospedagem Web.

A idéia é simples: uma única máquina responde na rede a mais endereços IP do que o número de interfaces de rede físicas que ela tem. Cada uma das interfaces de rede "virtuais" resultantes pode ser associada a um nome de domínio correspondente que os usuários na Internet poderão querer se conectar. Portanto, uma única máquina pode, literalmente, atender a centenas de sites Web.

Utilizando hosts virtuais baseados em nome

O protocolo HTTP 1.1 define uma espécie de funcionalidade similar às interfaces virtuais (oficialmente chamada de "hosts virtuais baseados em nome") que elimina a necessidade de atribuir endereços IP exclusivos a servidores Web ou de configurar uma interface especial no nível do sistema operacional. Essa abordagem economiza endereços IP e é útil para alguns sites, especialmente aqueles em que um único servidor abriga centenas ou milhares de home pages (como, por exemplo, as universidades).

Infelizmente, o esquema não é muito prático para sites comerciais. Ele reduz a escalabilidade (você deve alterar o endereço IP do site para movê-lo para um servidor diferente) e também pode ter um impacto negativo na segurança (se você filtrar acesso a um site no seu firewall de acordo com endereços IP). Além disso, hosts virtuais baseados em nome não podem utilizar SSL. Parece que interfaces virtuais verdadeiras ainda estarão por aí durante um tempo.

Configurando interfaces virtuais

Configurar uma interface virtual envolve duas etapas. Em primeiro lugar, você tem de criar a interface virtual em nível TCP/IP. Em segundo lugar, você tem de informar o servidor Apache sobre as interfaces virtuais que instalou. Tratamos dessa segunda etapa na página 503.

As interfaces virtuais no Linux recebem nomes segundo a seguinte convenção: notação *interface:instância*. Por exemplo, se sua interface Ethernet for eth0, então as interfaces virtuais associadas a ela poderiam ser eth0:0, eth0:1 e assim por diante. Todas as interfaces são configuradas com o comando **ifconfig**. Por exemplo, o comando:

```
# ifconfig eth0:0 128.138.243.150 netmask 255.255.255.192 up
```

configura a interface eth0:0 e atribui a ela um endereço na rede 128.138.243.128/26. Para tornar permanentes as atribuições de endereços virtuais, modifique os arquivos de inicialização de sistema.

Para tornar as atribuições de endereço virtual permanentes no Red Hat e Fedora, você cria um arquivo separado para cada interface virtual em **/etc/sysconfig/network-scripts**. Por exemplo, o arquivo **ifcfg-eth0:0** correspondente ao comando **ifconfig** mostrado acima contém as seguintes linhas.

```
DEVICE=eth0:0
IPADDR=128.138.243.150
NETMASK=255.255.255.192
NETWORK=128.138.243.128
BROADCAST=128.138.243.191
ONBOOT=yes
```

As abordagens do Debian e do Ubuntu são semelhantes à do Red Hat, mas as definições de interface devem aparecer no arquivo **/etc/network/interfaces**. As entradas correspondentes à interface eth0:0 em nosso exemplo são:

```
iface eth0:0 inet static
      address 128.138.243.150
      netmask 255.255.255.192
      broadcast 128.138.243.191
```

Em sistemas SUSE você pode criar interfaces virtuais com YaST ou manualmente, usando os arquivos de interface. Sob SUSE, os endereços IP da interface são todos configurados dentro de um único arquivo. Para configurar os arquivos manualmente, examine o diretório **/etc/sysconfig/network** para arquivos cujos nomes começam com **ifcfg-***ifname*. Os nomes de arquivo para interfaces reais incluem um endereço MAC de 6 bytes de aparência complexa; esses são os que você quer.

Por exemplo, um dos arquivos de configuração talvez contenha as seguintes entradas para definir duas interfaces virtuais:

```
IPADDR_0=128.138.243.149
NETMASK_0=255.255.255.192
LABEL_0=0
IPADDR_1=128.138.243.150
NETMASK_1=255.255.255.192
LABEL_1=1
STARTMODE="onboot"
NETWORK=128.138.243.128
```

Os sufixos que seguem IPADDR e NETMASK (aqui, _0 e _1) não têm de ser numéricos, mas, por consistência, essa é uma convenção razoável.

Informando o Apache das interfaces virtuais

Além de criar as interfaces virtuais, você precisa dizer ao Apache que documentos servir quando um cliente tentar se conectar a cada interface (endereço IP). Você faz isso com uma cláusula `VirtualHost` no arquivo **httpd.conf**. Há uma cláusula `VirtualHost` para cada interface virtual que você configurou. Eis um exemplo:

```
<VirtualHost 128.138.243.150>
     ServerName www.company.com
     ServerAdmin webmaster@www.company.com
     DocumentRoot /var/www/htdocs/company
     ErrorLog logs/www.company.com-error_log
     CustomLog logs/www.company.com-access_log combined
     ScriptAlias /cgi-bin/ /var/www/cgi-bin/company
</VirtualHost>
```

Nesse exemplo, qualquer cliente que se conectar ao host virtual 128.138.243.150 é atendido com documentos do diretório **/var/www/htdocs/company**. Quase toda diretiva Apache pode entrar em uma cláusula `VirtualHost` para definir configurações específicas desse host virtual. Caminhos do diretório relativos, incluindo aqueles para as diretivas `DocumentRoot`, `ErrorLog` e `CustomLog`, são interpretados no contexto do `ServerRoot`.

Com hosts virtuais baseados em nome, vários nomes DNS apontam para o mesmo endereço IP. A configuração do Apache é semelhante, mas você especifica o endereço IP primário em que o Apache deve ouvir por requisições identificadas de host virtual que chegam e omite o endereço IP na cláusula `VirtualHost`:

```
NameVirtualHost 128.138.243.150

<VirtualHost *>
     ServerName www.company.com
     ServerAdmin webmaster@www.company.com
     DocumentRoot /var/www/htdocs/company
     ErrorLog logs/www.company.com-error_log
     CustomLog logs/www.company.com-access_log combined
     ScriptAlias /cgi-bin/ /var/www/cgi-bin/company
</VirtualHost>
```

Nessa configuração, o Apache examina os cabeçalhos HTTP para determinar o site requisitado. O servidor ouve requisições para www.company.com em seu endereço IP principal, 128.138.243.150.

21.4 A Secure Sockets Layer (SSL)

O protocolo SSL[1] armazena comunicações entre um site Web e um navegador cliente. Os URLs que iniciam com https:// utilizam essa tecnologia. O SSL utiliza criptografia para impedir espionagem, adulteração e falsificação de mensagens.

O navegador e o servidor utilizam um esquema de autenticação baseado em certificado para estabelecer comunicações, depois que eles alternam para um esquema de criptografia baseado em cifra mais rápido para proteger sua conversa real.

O SSL executa como uma camada separada embaixo do protocolo de aplicativo HTTP. O SSL simplesmente fornece a segurança para a conexão e não se envolve na transação HTTP. Por causa dessa arquitetura higiênica, o SSL pode tornar seguros não só o HTTP, mas também protocolos como SMTP, NNTP e FTP. Para detalhes adicionais, consulte a entrada da Wikipedia "Secure Sockets Layer".

Consulte a página 650 para detalhes adicionais sobre questões jurídicas relacionadas à criptografia.

Nos "primeiros dias" de utilização do SSL, a maioria das chaves de criptografia simétricas era de 40 bits relativamente fracos por causa das restrições do governo dos Estados Unidos na exportação de tecnologia criptográfica. Depois de anos de controvérsia e processos, o governo relaxou alguns aspectos das restrições de exportação, permitindo que implementações SSL utilizassem chaves de 128 bits para cifras de chaves simétricas.

Gerando um certificado de solicitação de assinatura (Certificate Signing Request – CSR)

O proprietário de um site Web que deve utilizar SSL precisa gerar um CRS (Certificate Signing Request), um arquivo digital que contém uma chave pública e um nome de empresa. O "certificado" então deve ser "assinado" por uma fonte

1. O Transport Layer Security (TLS) é o protocolo sucessor do SSL e é implementado em todos os navegadores modernos. Entretanto, a comunidade Web ainda refere-se ao protocolo/conceito em geral como SSL.

confiável conhecida como autoridade certificada (Certificate Authority – CA). O certificado assinado retornado por CA contém a chave pública do site e o nome da empresa junto com o endosso da CA.

Os navegadores Web têm listas predefinidas de CAs cujos certificados assinados eles aceitarão. Um navegador que conhece a CA de seu site pode verificar a assinatura em seu certificado e obter sua chave pública, permitindo assim que ele envie mensagens que somente seu site pode decriptar. Embora você realmente possa assinar seu próprio certificado, um certificado que não vem de CA reconhecida requisita que a maioria dos navegadores notifique o usuário de que o certificado é potencialmente duvidoso. Em uma configuração comercial, esse comportamento é obviamente um problema. Mas, se você quiser configurar sua própria autoridade certificada para utilização interna e teste, consulte

httpd.apache.org/docs/2.0/ssl/ssl_faq.html#aboutcerts.

Você pode obter uma assinatura de certificado de qualquer uma das várias autoridades certificadas. Pesquise "SSL certificate" no Google e faça sua escolha. As únicas diferenças reais entre as CAs são a quantidade de trabalho que elas têm para verificar sua identidade, as garantias que oferecem e o número de navegadores que as suportam como eles vêm de fábrica (a maioria das CAs é suportada pela vasta maioria dos navegadores).

Criar um certificado para enviar para uma CA é relativamente simples e direto. O OpenSSL deve estar instalado, o que por padrão está na maioria das distribuições. Eis o procedimento.

Primeiro, crie uma chave privada RSA de 1.024 bits para seu servidor Apache:

```
$ openssl genrsa -des3 -out server.key 1024
```

Você é solicitado a inserir e confirmar uma senha para encriptar a chave de servidor. Faça backup do arquivo **server.key** para uma localização segura (legível somente por root) e não se esqueça da senha que você inseriu. Um curioso pode ver os detalhes numéricos da chave com esse comando:

```
$ openssl rsa -noout -text -in server.key
```

Então, crie um certificado de solicitação de assinatura (CSR) que incorpore a chave de servidor que você acabou de gerar:

```
$ openssl req -new -key server.key -out server.csr
```

Insira o nome de domínio completamente qualificado do servidor quando um "nome comum" (*common name*) for solicitado. Por exemplo, se o URL do site é https://www.company.com, insira "www.company.com" como seu nome comum. Note que você precisa de um certificado separado para cada hostname — a tal ponto que "company.com" é diferente de "www.company.com". As empresas normalmente registram apenas um nome comum; elas se certificam de que quaisquer links baseados em SSL apontem para esse hostname.

Você pode ver os detalhes de um CSR gerado com o seguinte comando:

```
$ openssl req -noout -text -in server.csr
```

Você agora pode enviar o arquivo **server.csr** à CA de sua escolha para obter a assinatura. Não é necessário preservar sua cópia local. O CSR assinado retornado pela CA deve ter a extensão **.crt**. Coloque o certificado assinado no mesmo lugar seguro da sua chave privada.

Configurando o Apache para utilizar SSL

As requisições HTTP entram na porta 80, e requisições HTTPS utilizam a porta 443. Tanto o tráfego HTTPS como o HTTP podem ser servidos pelo mesmo processo Apache. Entretanto, o SSL não funciona com hosts virtuais baseados em nome; cada host virtual deve ter um endereço IP específico. (Essa limitação é uma conseqüência de design de SSL.)

Para configurar o Apache para utilização com SSL, primeiro certifique-se de que o módulo SSL está ativado dentro de **httpd.conf** localizando ou adicionando a linha

```
LoadModule ssl_module      libexec/mod_ssl.so
```

Então adicione uma diretiva `VirtualHost` à porta SSL:

```
<VirtualHost 128.138.243.150:443>
     ServerName www.company.com
     ServerAdmin webmaster@www.company.com
     DocumentRoot /var/www/htdocs/company
     ErrorLog logs/www.company.com-ssl-error_log
     CustomLog logs/www.company.com-ssl-access_log combined
     ScriptAlias /cgi-bin/ /var/www/cgi-bin/company
     SSLEngine on
     SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.crt
     SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server.key
</VirtualHost>
```

Note o :443 depois do endereço IP e as diretivas SSL que dizem ao Apache onde localizar sua chave privada e o certificado assinado.

Quando você reiniciar o Apache, você será solicitado a inserir a senha frase ao seu arquivo **server.key**. Por causa dessa interação, **httpd** não pode mais iniciar automaticamente quando a máquina é inicializada. Se quiser, você pode remover a criptografia de sua chave privada para contornar a necessidade de inserir uma senha:

```
$ cp server.key server.key.orig
$ openssl rsa -in server.key.orig -out server.key
$ chmod 400 server.key server.key.orig
```

Naturalmente, qualquer pessoa que obtém uma cópia de chave descriptografada pode então se passar por seu site.

Para informações adicionais sobre SSL, consulte os seguintes recursos:

httpd.apache.org/docs-2.0/ssl/ssl_faq.ht
httpd.apache.org/docs/2.0/mod/mod_ssl.ht

21.5 O SERVIDOR PROXY E DE CACHE SQUID

A Internet e as informações nela contida estão crescendo rapidamente. Portanto, a largura de banda e os recursos computacionais necessários para suportá-la também estão crescendo rapidamente. Como esse estado de coisas pode continuar?

A única forma de lidar com tal crescimento é usar replicação. Independentemente de estar em um nível nacional, regional ou de site, o conteúdo Internet precisa estar prontamente disponível a partir da fonte mais próxima à medida que a Internet cresce. Não faz o menor sentido transmitir a mesma página Web popular da Austrália por meio de um link muito caro para a América do Norte milhões de vezes ao dia. Deveria haver uma maneira de armazenar essa informação uma vez que ela tenha sido enviada por meio do link uma vez. Felizmente, há uma maneira.

O cache Squid e servidor proxy

Uma resposta seria o pacote gratuito Squid Internet Object Cache.[2] Esse pacote é, ao mesmo tempo, um servidor proxy e um servidor de cache que executa sob o Linux e suporta diversos protocolos, entre os quais HTTP, FTP e SSL.

Eis como ele funciona. Os navegadores Web clientes entram em contato com o servidor Squid para requisitar um objeto da Internet. O servidor Squid faz então uma requisição em nome do cliente (ou fornece o objeto de seu cache, conforme discutido no parágrafo seguinte) e retorna o resultado ao cliente. Os servidores proxy desse tipo são normalmente utilizados para aumentar a segurança e para filtrar conteúdo.

Em um sistema baseado em proxy, somente uma máquina precisa de acesso direto à Internet através do firewall da organização. Em organizações como escolas do primeiro e segundo graus, um servidor proxy também pode filtrar conteúdo para que material inapropriado não caia em mãos erradas. Diversos servidores proxy comerciais e gratuitos (alguns baseados no Squid, outros não) estão disponíveis hoje. Alguns desses sistemas são puramente baseados em software (como Squid) e outros são baseados em hardware (por exemplo, o BlueCoat; consulte www.cacheflow.com). Uma lista extensa de tecnologias de servidor proxy pode ser localizada em:

www.web-caching.com/proxy-caches.html

O serviço proxy é ótimo, mas são os recursos de armazenamento em cache do Squid que realmente nos impressionam. O Squid não só armazena em cache as informações de requisições locais de usuário como permite também a construção de uma hierarquia de servidores Squid.[3] Os grupos de servidores Squid utilizam o Internet Cache Protocol (ICP) para comunicar informações sobre o que há em seus caches.

Com esse recurso, administradores podem construir um sistema em que usuários locais contatam um servidor de cache no site para obter conteúdo da Internet. Se um outro usuário nesse site tiver requisitado o mesmo conteúdo, poderá ser retornada uma cópia em velocidade de rede local (normalmente, 100 Mb/s ou mais). Se o servidor Squid local não tiver o objeto, talvez o servidor contate o servidor de cache regional. Como no caso local, se alguém da região tiver requisitado o mesmo objeto, ele será servido imediatamente. Caso contrário, talvez o servidor de cache do país ou continente possa ser contatado e assim por diante. Os usuários percebem uma melhora de desempenho; portanto, eles ficam contentes.

Para muitos, o Squid oferece benefícios econômicos. Como os usuários tendem a compartilhar suas descobertas na Web, uma significativa duplicação de requisições Web externas pode ocorrer em um site de porte razoável. Um estudo demonstra que manter um servidor de cache pode reduzir as necessidades de largura de banda em cerca de 40%.

---

2. Por que "Squid" (lula)? De acordo com a FAQ, "todos os nomes bons já haviam sido usados".
3. Infelizmente, alguns sites marcam todas as suas páginas como sendo não-armazenáveis em cache, impedindo o Squid de operar sua mágica. De uma maneira similar, o Squid não é capaz de colocar em cache páginas geradas dinamicamente.

Para fazer utilização efetiva do Squid, é provável que você queira forçar seus usuários a usar o cache. Configure um proxy padrão pelo Active Directory (em um ambiente baseado no Windows) ou configure seu roteador para redirecionar todo o tráfego baseado na Web para o cache Squid utilizando o Web Cache Communication Protocol, WCCP.

Configurando o Squid

O Squid é fácil de ser instalado e configurado. Como o Squid precisa de espaço para armazenar o seu cache, você deve executá-lo em uma máquina dedicada que tenha bastante memória e espaço livre em disco. Uma configuração para um cache relativamente grande seria uma máquina com 2 GB de RAM e 200 GB de disco.

Você pode obter o pacote Squid em formato RPM ou **apt-get** de sua distribuição ou então fazer a cópia de uma versão recente do Squid em www.squid-cache.org. Se optar pelo caminho "faça você mesmo", execute o script **configure** no alto da árvore de origem após descompactar a distribuição. Esse script assume que você quer instalar o pacote em **/usr/local/squid**. Se preferir alguma outra localização, utilize a opção **prefix**=*dir* do **configure**. Depois de **configure** ter sido completado, execute **make all** e, a seguir, **make install**.

Depois de instalar o Squid, você deve localizar o arquivo de configuração **squid.conf**. Consulte o arquivo **QUICKSTART** no diretório de distribuição para uma lista das alterações que você precisa fazer no arquivo **squid.conf** de exemplo.

Você também tem de executar **squid -z** manualmente para criar e zerar a estrutura de diretórios na qual serão armazenadas as páginas Web em cache. Por fim, você pode iniciar o servidor manualmente com o script **RunCache**; você acabará querendo chamar esse script a partir dos arquivos **rc** de seu sistema para que eles iniciem o servidor Squid quando a máquina for inicializada.

Para testar o Squid, configure o seu navegador Web para utilizar o servidor Squid como proxy. Essa opção é normalmente encontrada no painel de preferências do navegador.

21.6 CONFIGURAÇÃO DE SERVIDOR FTP ANÔNIMO

O FTP é um dos mais antigos e básicos serviços da Internet, embora ainda continue a ser utilizado amplamente hoje em dia. Embora o FTP tenha uma variedade de usos internos, a aplicação mais comum na Internet continua a ser "FTP anônimo", pelo qual usuários que não têm contas no seu site podem fazer downloads de arquivos que você disponibilizou.

O FTP é útil para distribuir correções de bugs, software, documentos etc., mas os servidores HTTP atuais têm feito tudo menos substituir os servidores FTP. Os argumentos em favor do FTP são relativamente fracos: O FTP pode ser um pouco mais confiável e os usuários não precisam de navegadores Web para acessar um site FTP (embora obviamente precisem de cliente FTP).

Utilize o FTP básico *somente* quando acesso anônimo for requerido. Para aplicativos não anônimos, utilize a variante segura, o SFTP. O FTP transmite senhas em texto simples e tem uma história de incidentes de segurança.

O **ftpd** é gerenciado pelo **inetd** e, portanto, tem uma entrada nos arquivos **/etc/inetd.conf** e **/etc/services**. (Se sua distribuição utiliza o **xinetd** em vez do **inetd**, deve existir um arquivo em **/etc/xinetd.d** para o **ftpd**.) Quando um usuário FTP se conecta anonimamente, o **ftpd** executa uma chamada de sistema **chroot** (abreviação de "change root") para tornar arquivos fora do diretório **~ftp** invisíveis e inacessíveis. Devido à natureza pública do FTP anônimo, é importante que o **ftpd** seja configurado corretamente para que arquivos confidenciais não se tornem acidentalmente disponíveis para todo o mundo.

Para permitir **ftp** anônimo ao seu site, siga estes passos na seqüência listada:

- Adicione o usuário "ftp" aos arquivos regulares de senha e de senha sombra (o usuário ftp já deve existir em todas as distribuições exceto nas do Debian). Ninguém precisa se conectar à conta ftp, então utilize um "x" como a senha do ftp. Também é uma boa idéia especificar **/sbin/nologin** ou **/sbin/nologin** como shell de login para ftp.
- Crie o diretório inicial do ftp se ele já não existir.
- Crie os subdiretórios **bin**, **etc**, **lib** e **pub** sob **~ftp**. Como uma sessão **ftp** anônima executa com **chroot** aplicado a **~ftp**, os subdiretórios **bin** e **etc** devem fornecer uma cópia de todos os comandos e informações de configuração exigidos por **ftpd**. Após o **chroot**, **~ftp/bin** e **~ftp/etc** tornam-se mascarados como **/bin** e **/etc**.
- Copie o programa **/bin/ls** para o diretório **~ftp/bin**. Para maior segurança, torne **~ftp/bin/ls** "somente executável" configurando seu modo para 111. Esse ajuste impede que clientes copiem o binário e o estudem, em busca de pontos fracos.
- Copie ou faça um link físico (*hard link*) para as bibliotecas compartilhadas de que **ls** precisa para **~ftp/lib**. Verifique a documentação da sua distribuição para descobrir que arquivos são necessários. Note que o link físico funciona somente se os arquivos residirem na mesma partição de disco.
- Copie **/etc/passwd** e **/etc/group** para **~ftp/etc**.

- Edite os arquivos **passwd** e **group**. **ftpd** utiliza somente o comando **ls** e cópias do esqueleto de **/etc/passwd** e **/etc/group** a partir de **~ftp/etc**. Os arquivos de grupo **passwd** sob **~ftp** devem conter somente root, daemon e ftp; e os campos de senha devem conter "x".
- Execute **mkpasswd** ou **pwd_mkdb** se seu sistema exigir uma versão em hash do banco de dados de senhas.
- Configure as permissões apropriadas nos arquivos e diretórios sob **~ftp**. Recomendamos que as permissões sejam configuradas conforme mostrado na Tabela 22.4.

Tabela 21.4 Permissões recomendadas sob ~ftp.

Arquivo/Diretório	Proprietário	Modo	Arquivo/Diretório	Proprietário	Modo
~ftp	root	555	**~ftp/etc/passwd**	root	444
~ftp/bin	root	555	**~ftp/etc/group**	root	444
~ftp/bin/ls	root	111	**~ftp/pub**	root	755
~ftp/etc	root	555	**~ftp/lib**	root	555

- Edite **/etc/ftpusers** e remova as entradas de "ftp" e "anonymous" para não permitir que usuários anônimos se conectem.
- Coloque os arquivos que deseja tornar disponíveis em **~ftp/pub**.

Um dos maiores riscos de segurança do FTP anônimo resulta de permitir que usuários depositem arquivos em diretórios FTP. Diretórios que podem ser gravados por qualquer um, independentemente de quão obscuros, rapidamente se tornam "ninhos" onde hackers e crianças pretendendo negociar seus "warez" armazenarão seus arquivos, consumindo toda a sua largura de banda e colocando-o diretamente no meio de uma série de atividades que provavelmente são indesejáveis, se não completamente ilegais. Não faça parte do problema; jamais permita diretórios FTP anônimos graváveis em seu sistema.

21.7 Exercícios

★ E21.1 Configure uma interface virtual em sua estação de trabalho. Execute **ifconfig** antes e depois para ver o que mudou. Você é capaz de efetuar um ping na interface virtual por meio de uma outra máquina na mesma sub-rede? E de uma rede diferente? Por que sim ou por que não? (Exige acesso de root.)

★ E21.2 Com um analisador de pacotes (**tcpdump**) capture uma conversação HTTP bidirecional que faça o envio de informações (por exemplo, preenchendo um formulário ou campo de pesquisa). Comente a sessão para mostrar como o seu navegador transmitiu informações para o servidor Web. (Exige acesso de root.)

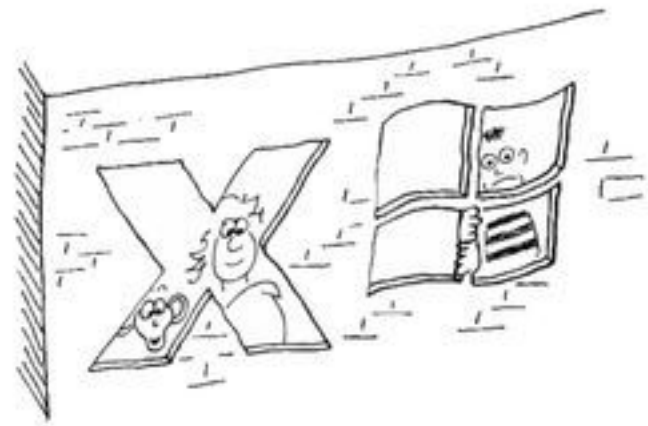
★ E21.3 Utilize um analisador de pacotes para capturar o tráfego quando você abre uma página Web com muito tráfego, como a home page da amazon.com ou cnn.com. Quantas conexões TCP distintas são abertas? Quem as inicia? Poderíamos tornar o sistema mais eficiente? (Exige acesso de root.)

★ E21.4 Localize arquivos de log de um servidor Web acessível via Internet, talvez o servidor principal de seu site. Examine os arquivos de log. O que você pode dizer sobre os padrões de acesso após um período de algumas horas? Que erros surgiram durante esse período? Quais são as preocupações em relação à segurança indicadas pelo conteúdo dos arquivos de log? (Pode exigir acesso de root.)

★★ E21.5 Instale o Apache em seu sistema e crie algumas páginas de conteúdo. A partir de outras máquinas, verifique se o seu servidor Web está operando. Localize os arquivos de log do Apache que permitem que você veja quais navegadores estão acessando o seu servidor. Configure o Apache para servir parte de suas páginas de conteúdo para a interface virtual criada no E21.1. (Exige acesso de root.)

PARTE 3
DIVERSOS

22 O X Window System



O X Window System, também chamado X11, ou simplesmente X, é a base para a maioria dos ambientes gráficos com o usuário do UNIX e do Linux. O X é o sucessor natural de um sistema de janelas chamado (acredite ou não) W, que foi desenvolvido pelo MIT, no início da década de 1980, como parte do Project Athena. A versão 10 do X Window System, distribuída em 1985, foi a primeira a alcançar implantação disseminada e a versão 11 (X11) veio logo em seguida. Graças aos termos de licenciamento relativamente liberais desse sistema, a extensão X foi rapidamente incorporada por outras plataformas surgindo assim múltiplas implementações.

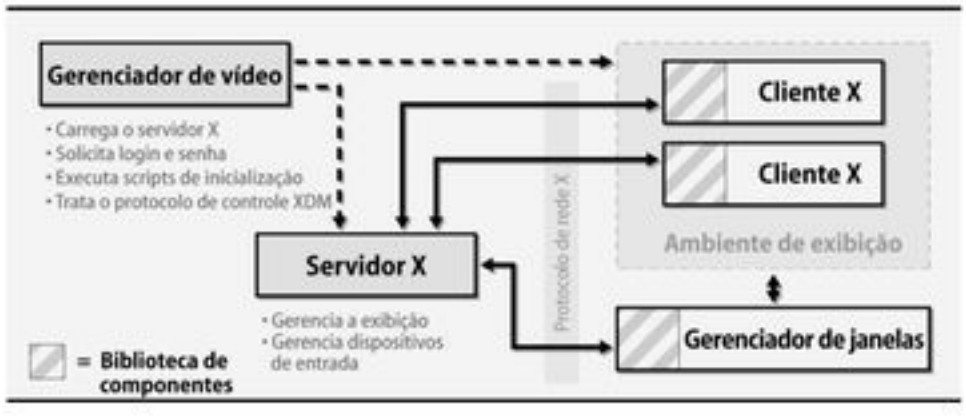
Em 1988, o MIT X Consortium foi fundado para especificar as diretrizes gerais do protocolo X. Ao longo da década seguinte, esse grupo e seus sucessores emitiram uma série constante de atualizações do protocolo. O X11R7.1 é hoje a melhor e mais recente versão, aparentemente com tendências a acrescentar novos números à designação de versão em vez de incrementar os existentes.

O XFree86 tornou-se a implementação de fato do servidor X para o Linux (e para muitas outras plataformas) até que, em 2004, uma alteração no licenciamento fez com que muitas distribuições passassem a usar uma variação do XFree86 sem os impedimentos legais da nova cláusula de licenciamento. Essa variação é mantida pela X.Org Foundation, uma fundação sem fins lucrativos, e é a implementação Linux predominante hoje em dia. Além disso, o servidor X.Org foi portado para o Windows para uso no ambiente de compatibilidade Cygwin Linux. (Vários servidores X comerciais para o Windows também estão disponíveis; veja a página 565 para informações adicionais.)

Este capítulo descreve a versão X.Org do X, que é utilizada por todos os nossos exemplos de distribuição. As implementações do X.Org e do XFree86 divergem arquitetonicamente, mas a maioria dos detalhes administrativos permanece a mesma. Freqüentemente é possível substituir "xf86" por "xorg" nos comandos e nomes de arquivos para descobrir a versão XFree86 apropriada. O XFree86 está se tornando obsoleto e não será discutido aqui.

O X Window System pode ser dividido em alguns componentes-chave. Primeiro, ele fornece um *gerenciador de vídeo* (*display manager*) cujo principal trabalho é autenticar usuários, efetuar login destes e inicializar um ambiente a partir dos scripts de inicialização. O gerenciador de vídeo também inicializa o *servidor X*, que define uma interface abstrata para as telas baseadas em mapa de bits do sistema e para dispositivos de entrada (por exemplo, teclado e mouse). Os scripts de inicialização também executam um *gerenciador de janelas (window manager)*, que permite ao usuário mover, redimensionar, minimizar e maximizar janelas, bem como gerenciar áreas de trabalho virtuais separadas. Por fim, no nível mais baixo, os aplicativos estão vinculados a uma *biblioteca de componentes* (*widget library*), que implementa mecanismos de alto nível para uma interface com o usuário como botões e menus. A Ilustração A explica o relacionamento entre o gerenciador de vídeo, o servidor X e os aplicativos clientes.

Ilustração A O modelo cliente/servidor do X



O servidor X só entende um conjunto muito básico de primitivas gráficas através de uma API de rede; ele não define uma interface de programação para entidades de alto nível como botões, caixas de texto, menus e controles deslizantes. Esse design atende a dois objetivos importantes. Primeiro, ele permite que o servidor X execute em um computador completamente separado do aplicativo cliente. Segundo, ele permite que o servidor suporte uma variedade de diferentes gerenciadores de janelas e conjuntos de componentes.

Os desenvolvedores de aplicativos têm a possibilidade de escolher entre várias bibliotecas comuns de componentes (*widgets*) e padrões de interface com o usuário. Infelizmente, essa escolha com freqüência depende mais de uma afiliação religiosa do que de quaisquer considerações sobre o projeto real. Argumenta-se que, embora a liberdade de escolha seja boa, o agnosticismo da interface do X resultou em muitos anos de interfaces ruins.

Neste capítulo, explicaremos como executar programas em uma exibição remota e como permitir autenticação. Então discutiremos como configurar o servidor X.Org e como solucionar os erros de configuração. Por fim, discutiremos brevemente alguns gerenciadores de janelas e ambientes de área de trabalho disponíveis.

22.1 O GERENCIADOR DE VÍDEO X

O gerenciador de vídeo é a primeira coisa que um usuário normalmente vê ao ligar o computador. Ele não é requerido; muitos usuários desativam o gerenciador de vídeo e iniciam o X a partir do console de texto ou do script **.login** executando **startx** (que é um empacotador do programa **xinit**, que inicia o servidor X).

Por outro lado, o gerenciador de vídeo apresenta uma atraente tela de login amigável ao usuário e adiciona algumas capacidades extras de configuração. O gerenciador de vídeo pode permitir logins remotos para outros servidores X por meio do protocolo XDMCP. Ele também pode tratar a tela de autenticação (ver *Autenticação de cliente,* na página 514). O gerenciador de vídeo original é chamado **xdm** (o gerenciador de vídeo X), mas substitutos modernos como **gdm** (o gerenciador de vídeo GNOME) e o **kdm** (o gerenciador de vídeo KDE) disponibilizam mais ou menos o mesmo conjunto de recursos e têm uma aparência muito melhor.

Em um cenário típico, o gerenciador de vídeo carrega o servidor X, autentica o usuário, efetua o login do usuário no sistema e executa os scripts de inicialização do usuário. Um conjunto de arquivos de configuração, freqüentemente localizado no diretório **/etc/X11/xdm**, especifica a maneira como **xdm** será executado. Por exemplo, talvez você queira editar o arquivo **Xservers** para alterar o número de exibição utilizado por esse servidor se múltiplos servidores forem executados em outros terminais virtuais. Ou, você poderia alterar o layout do servidor com a opção **-layout** se tiver definido diferentes layouts para múltiplos sistemas.

Consulte a página 471 para informações adicionais sobre o PAM.

Depois de carregar o servidor X, o **xdm** solicita um nome de usuário e senha. A senha do usuário é autenticada de acordo com os módulos PAM (Pluggable Authentication Modules) especificados em **/etc/pam.d/xdm** (ou **kdm/gdm** se estiver utilizando os gerenciadores de vídeo KDE ou GNOME). A tela de login também pode apresentar a opção para efetuar login a vários ambientes de área de trabalho alternativos, incluindo a importante opção *failsafe* (contra falhas) discutida a seguir.

A tarefa final do gerenciador de vídeo é executar o script de shell **Xsession**, que configura o ambiente de área de trabalho do usuário. O script **Xsession**, na maioria das vezes também localizado em **/etc/X11/xdm**, é um script de inicialização de nível de sistema. Ele configura padrões de aplicativo, instala as configurações de teclado padrão e seleciona configurações de idiomas. O script **Xsession** então executa o script de inicialização pessoal do usuário, normalmente chamado **~/.xsession**, para iniciar o gerenciador de janelas, barras de tarefa, miniaplicativos auxiliares e possivelmente outros programas. O GNOME e o KDE têm seus próprios scripts de inicialização que configuram a área de trabalho do usuário de acordo com as ferramentas de configuração do GNOME e do KDE; esse esquema é menos propenso a erros do que a edição pelos próprios usuários dos scripts de inicialização.

Quando a execução de **~/.xsession** termina, o usuário é desconectado do sistema e o gerenciador de vídeo volta a solicitar um nome de usuário e senha. Portanto, **~/.xsession** deve iniciar todos os programas em segundo plano (acrescentando um **&** ao final de cada comando), *exceto o último,* que normalmente é o gerenciador de janelas. (Se todos os comandos em **~/.xsession** forem executados em segundo plano, o script termina logo em seguida e o usuário é desconectado imediatamente depois de efetuar login.) Com o gerenciador de janela como o processo final em primeiro plano, o usuário é desconectado somente depois que o gerenciador de janelas é encerrado.

A opção de login contra falhas (*failsafe)* permite que os usuários efetuem o login e corrijam seus scripts de inicialização que não estão funcionando. Essa opção normalmente pode ser selecionada a partir da tela de login do gerenciador de vídeo. Ela abre apenas uma janela de terminal simples; depois que essa janela fecha, o sistema desconecta o usuário. Cada sistema deve permitir a opção de login contra falhas; ela ajuda os usuários a corrigir seus próprios erros evitando assim que você receba uma chamada no meio da noite.

Esquecer de deixar um processo em primeiro plano é o problema de inicialização mais comum, mas dificilmente será o único. Se a causa dos problemas não for óbvia, talvez você tenha de verificar o arquivo **~/.xsession-errors**, que contém a saída dos comandos executados a partir de **~/.xsession**. Procure erros ou outros comportamentos inesperados. Se necessá-

rio, mova o script ~/.**xsession** para outro lugar e certifique-se de que você pode efetuar login sem ele. Então, restaure uma ou duas linhas de cada vez até encontrar aquela que causa o problema.

22.2 EXECUTANDO UM APLICATIVO X

O processo necessário para executar um aplicativo X pode a princípio parecer excessivamente complexo. Entretanto, você logo descobrirá a flexibilidade fornecida pelo modelo de exibição cliente/servidor. Como as atualizações de exibição são transmitidas pela rede, um aplicativo (o cliente) pode executar em um computador completamente separado daquele que exibe a interface gráfica com o usuário (o servidor). Um servidor X pode estar conectado a vários diferentes aplicativos, todos executando em computadores separados.

Para que esse modelo funcione, precisamos informar os clientes à qual exibição eles devem se conectar e qual tela usar nessa exibição. Depois de conectado, os clientes precisam de uma autenticação no servidor X para assegurar que a pessoa que está utilizando a exibição autorizou a conexão.

Consulte a página 481 para informações adicionais sobre SSH.

Mesmo com a autenticação, a segurança intrínseca do X é relativamente fraca. Você pode gerenciar conexões de uma maneira mais segura roteando-as por meio do SSH (ver *Encaminhamento de conexões X com SSH*, na página 514). Recomendamos veementemente o uso de SSH para conexões X pela Internet. Ele também é bom para tráfego local.

A variável de ambiente DISPLAY

Aplicativos X consultam a variável de ambiente DISPLAY para descobrir onde elas são exibidas. Essa variável contém o hostname ou endereço IP do servidor, o número de exibição (identificando a instância particular de um servidor X à qual se conectar) e um número opcional de tela (para exibições com múltiplos monitores). Quando aplicativos são executados no mesmo computador que exibe suas interfaces, por simplicidade você pode omitir a maioria desses parâmetros.

O exemplo a seguir mostra o formato das informações de vídeo e a sintaxe do **bash** utilizada para configurar a variável de ambiente:

```
client$ DISPLAY=servername.domain.com:10.2; export DISPLAY
```

Essa configuração aponta aplicativos X para servername.domain.com, exibição 10, tela 2. Os aplicativos estabelecem uma conexão TCP com o servidor no número de porta 6000 mais o número da exibição (nesse exemplo, a porta 6010), em que o servidor X que trata essa exibição deve ouvir.

Tenha em mente que cada processo tem suas próprias variáveis de ambiente. Ao configurar a variável DISPLAY para um shell, seu valor é herdado apenas pelos programas que você executa a partir desse shell. Se executar os comandos acima em um **xterm** e então tentar executar seu aplicativo X favorito a partir de um outro, o aplicativo não terá acesso à sua variável DISPLAY cuidadosamente construída.

Outro ponto que vale mencionar é que, embora os aplicativos X enviem sua saída gráfica ao servidor X especificado, eles ainda têm de usar os canais stdout e stderr locais. Alguma saída de erro ainda pode aparecer na janela de terminal a partir do qual um aplicativo X foi executado.

Veja a página 291 para informações adicionais sobre a configuração do conversor (resolver) do DNS.

Se tanto o cliente como o servidor são partes da sua empresa local, normalmente é possível omitir o nome de domínio completo do servidor na variável DISPLAY, dependendo de como o conversor (*resolver*) do seu servidor de nomes foi configurado. Além disso, como a maioria dos sistemas só executa um único servidor X, a exibição normalmente é 0. O número de tela pode ser omitido, nesse caso assume-se a tela 0. Portanto, na maioria das vezes não há problemas em configurar o valor de DISPLAY como servername:0.

Se o aplicativo cliente estiver em execução na mesma máquina que a do servidor X, você pode simplificar a variável DISPLAY ainda mais omitindo o hostname. Esse recurso é mais que apenas cosmético: com um hostname nulo, as bibliotecas de cliente utilizam um socket de domínio UNIX em vez de um socket de rede para contatar o servidor X. Além de ser mais rápido e mais eficiente, esse método de conexão dribla quaisquer restrições de firewall no sistema local que está tentando bloquear conexões X externas. O valor mais simples possível para a variável de ambiente DISPLAY é, portanto, simplesmente ":0".

As mesmas bibliotecas de cliente que lêem a variável de ambiente DISPLAY normalmente também aceitam essas informações na forma de um argumento de linha de comando. Por exemplo, o comando:

```
client$ xprogram -display servername:0
```

é o equivalente a executar o programa com DISPLAY configurada como "servername:0". As opções de linha de comando sobrescrevem as configurações da variável de ambiente. Esse recurso é especialmente útil se você estiver executando na mesma máquina vários programas que são tratados por exibições diferentes.

Autenticação de cliente

Embora o ambiente X seja em geral considerado relativamente inseguro, toda precaução ajuda a evitar acesso não-autorizado. Quando a segurança ainda não era uma preocupação tão séria, era comum que servidores X aceitassem conexões de qualquer cliente em execução em um host que tivesse sido especificado como seguro com o comando **xhost**. Mas como qualquer usuário nesse host poderia então se conectar à sua exibição e provocar uma devastação (intencionalmente ou devido a confusões), o método **xhost** de conceder acesso aos clientes acabou tornando-se obsoleto. Não o discutiremos aqui.

A alternativa mais predominante à segurança baseada em host é chamada autenticação por meio de cookie mágico. Embora cookies mágicos possam suscitar flashbacks em alguns dos nossos leitores, nesse contexto eles são utilizados para autenticar conexões X. A questão básica é que o gerenciador de vídeo X gera um grande número aleatório, chamado cookie, no início do procedimento de login. O cookie para o servidor é gravado no arquivo **~/.Xauthority** no diretório inicial do usuário. Quaisquer clientes que conhecem o cookie têm permissão para se conectar. Usuários podem executar o comando **xauth** para visualizar os cookies existentes e adicionar novos a esse arquivo.

A maneira mais simples de mostrar como isso funciona é com um exemplo. Suponha que você configurou sua variável DISPLAY no sistema do cliente para exibir aplicativos X na máquina em que você está trabalhando. Mas, ao executar um programa, você recebe um erro que se parece com isto:

```
client$ xprogram -display server:0
Xlib: connection to "server:0.0" refused by server
xprogram:  unable to open display 'server:0'
```

Essa mensagem informa que o cliente não tem o cookie correto, assim o servidor remoto recusa a conexão. Para receber o cookie correto, efetue login no servidor (o que provavelmente você já fez se estiver tentando exibir nele) e liste os cookies do servidor executando **xauth list**:

```
server$ xauth list
server:0  MIT-MAGIC-COOKIE-1  f9d888df6077819ef4d788fab778dc9f
server/unix:0  MIT-MAGIC-COOKIE-1  f9d888df6077819ef4d788fab778dc9f
localhost:0  MIT-MAGIC-COOKIE-1  cb6cbf9e5c24128749feddd47f0e0779
```

Cada interface de rede no servidor tem uma entrada. Neste exemplo, temos um cookie para a Ethernet, um cookie para o socket de domínio UNIX utilizado para conexões locais e um cookie para a interface de rede de loopback do host local.

A maneira mais fácil de receber o cookie no cliente (se não utilizar SSH, que negocia o cookie para você) é com o bom e velho recortar e colar. A maioria dos emuladores de terminais (por exemplo, **xterm**) permite selecionar texto com o mouse e colá-lo em uma outra janela, normalmente pressionando o botão do meio do mouse. Convenientemente, o comando **xauth add** aceita como entrada o mesmo formato que **xauth list** exibe. Você pode adicionar o cookie ao cliente desta maneira:

```
client$ xauth add server:0  MIT-MAGIC-COOKIE-1  9d888df6077819ef4d788fab778dc9f
```

Você deve verificar se o cookie foi adicionado adequadamente executando **xauth list** no cliente. Com o conjunto de variáveis de ambiente DISPLAY e o cookie mágico correto adicionado ao cliente, os aplicativos agora devem ser exibidos corretamente no servidor.

Se tiver problemas em colocar os cookies em funcionamento, você pode voltar temporariamente à autenticação **xhost** apenas para verificar se não houve nenhum outro problema (por exemplo, restrições de firewalls ou de rede local que impedem que o cliente acesse o servidor). Sempre execute **xhost -** (isto é, **xhost** com um traço como seu único argumento) para desativar a autenticação **xhost** depois que seu teste estiver completo.

Encaminhamento de conexões X com SSH

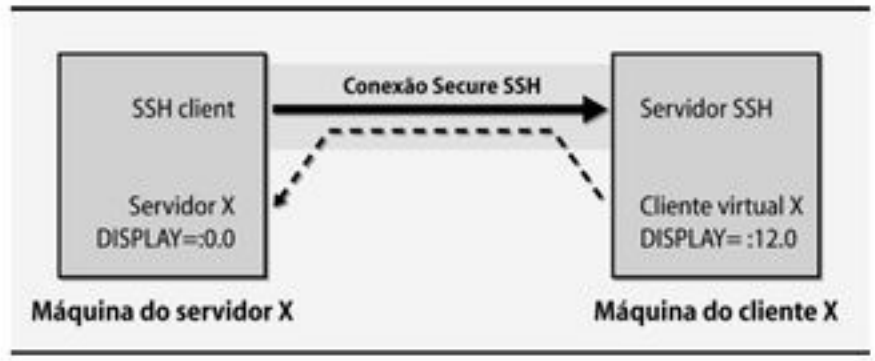
Cookies mágicos aumentam a segurança, mas dificilmente são à prova de falhas. Qualquer usuário que possa obter o cookie da sua exibição pode se conectar à exibição e executar programas que monitoram suas atividades. Mesmo sem seu cookie, o protocolo X transfere dados pela rede sem criptografia, permitindo que eles sejam espionados por praticamente qualquer pessoa.

Consulte a página 481 para informações adicionais sobre SSH.

Você pode aprimorar a segurança com SSH, o protocolo de shell seguro. O SSH fornece um serviço de terminal autenticado e encriptado. Entretanto, o SSH também pode encaminhar dados de rede arbitrários, incluindo dados do protocolo X, por um canal seguro. O encaminhamento do X é semelhante ao encaminhamento de porta SSH genérico, mas como o X é ciente do SSH, você ganha alguns recursos adicionais, incluindo uma pseudo-exibição na máquina remota e a transferência negociada dos cookies mágicos.

Em geral você inicia o **ssh** a partir da máquina em que o servidor X está execução para a máquina em que você quer executar programas X. Esse arranjo pode ser confuso de ler porque o *cliente* SSH é executado na mesma máquina do *servidor* X e ele se conecta a um *servidor* SSH que é a mesma máquina dos aplicativos *clientes* do X. Para tornar isso pior, a exibição virtual que o SSH cria para seu servidor X é local para o sistema remoto. A Ilustração B mostra como o tráfego X flui pela conexão SSH.

Ilustração B Utilizando SSH com X



Sua variável DISPLAY e as informações de autenticação são configuradas automaticamente pelo **ssh**. O número de exibição inicia em :10.0 e incrementa para cada conexão SSH que está encaminhando o tráfego X.

Um exemplo talvez ajude a mostrar esta seqüência:

```
x-server$ ssh -v -X x-client.mydomain.com
OpenSSH_3.9p1, OpenSSL 0.9.7a Feb 19 2003
debug1: Reading configuration data /home/boggs/.ssh/config
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to x-client.mydomain.com [192.168.15.9] port 22.
debug1: Connection established.
Enter passphrase for key '/home/boggs/.ssh/id_rsa':
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
debug1: Entering interactive session.
debug1: Requesting X11 forwarding with authentication spoofing.
debug1: Requesting authentication agent forwarding.
x-client$
```

Você pode ver nas duas últimas linhas que o cliente está solicitando encaminhamento para aplicativos X11. O encaminhamento do X deve ser ativado tanto no servidor SSH como no cliente SSH e o cliente precisa ainda ter o cookie correto para o servidor. Se as coisas parecem não estar funcionando da maneira correta, tente os flags **-X** e **-v** como mostrado anteriormente (para OpenSSH) para ativar explicitamente o encaminhamento do X e solicitar saída prolixa. Também verifique os arquivos de configuração SSH globais em **/etc/ssh** para certificar-se de que o encaminhamento do X11 não foi desativado administrativamente. Depois de conectado, você pode verificar sua exibição e os cookies mágicos:

```
x-client$ echo $DISPLAY
localhost:12.0
x-client$ xauth list
x-client/unix:12  MIT-MAGIC-COOKIE-1  a54b67121eb94c8a807f3ab0a67a51f2
```

Observe que a DISPLAY aponta para uma exibição virtual no servidor SSH. São atribuídas às outras conexões SSH (tanto de você como de outros usuários) diferentes números de exibição virtuais. Com a DISPLAY e o cookie adequadamente configurados, agora podemos executar o aplicativo cliente.

```
x-client$ xeyes
debug1: client_input_channel_open: ctype x11 rchan 4 win 65536 max 16384
debug1: client_request_x11: request from 127.0.0.1 35411
debug1: channel 1: new [x11]
debug1: confirm x11
debug1: channel 1: FORCE input drain
```

Com as informações de depuração ativadas com **ssh -v**, podemos ver que **ssh** recebeu a solicitação da conexão X e obedientemente encaminhou-a ao servidor X. O encaminhamento pode ser um pouco lento em um link distante, mas o aplicativo conseqüentemente deve aparecer na sua tela.

22.3 CONFIGURAÇÃO DO SERVIDOR X

O servidor **Xorg** tem uma reputação notória de ser difícil de configurar para um dado ambiente de hardware. Essa reputação é merecida. Em parte, a complexidade da configuração do **Xorg** é explicada pelo amplo array de hardware gráfico, dispositivos de entrada, modo de vídeo, resoluções e profundidades de cores que ele suporta. No início do XFree86, um novo usuário costumava ficar assustado com um arquivo de configuração obscuro contendo o que pareciam ser números aleatórios derivados de informações obscuras no fundo do inexistente manual do monitor. O arquivo de configuração do **Xorg** é significativamente mais estruturado para alguns desses aparentemente números aleatórios.

O arquivo de configuração do **Xorg** normalmente está localizado em **/etc/X11/xorg.conf**, mas o servidor X pesquisará um número bem maior de diretórios para encontrá-lo. A página man apresenta uma lista completa, mas uma questão a observar é que alguns caminhos que o **Xorg** pesquisa contêm o hostname e uma variável global, o que facilita o armazenamento de arquivos de configuração para múltiplos sistemas em um local central.

Vários programas podem ajudá-lo a configurar o X (tais como **xorgconfig**), mas uma boa idéia é entender como o arquivo de configuração é estruturado caso você precise visualizar ou editar a configuração diretamente. Você pode coletar algumas informações iniciais úteis diretamente do servidor X executando **Xorg -probeonly** e examinando na saída o chipset de vídeo e outros valores investigados. Você pode executar **Xorg -configure** para fazer o servidor X criar um arquivo de configuração inicial que se baseia nos valores investigados. É um bom lugar para começar se não houver nenhum outro.

O arquivo **xorg.conf** é organizado em várias seções, cada uma iniciando com a palavra-chave `Section` e terminando com `EndSection`. Os tipos mais comuns de seção estão listados na Tabela 22.1.

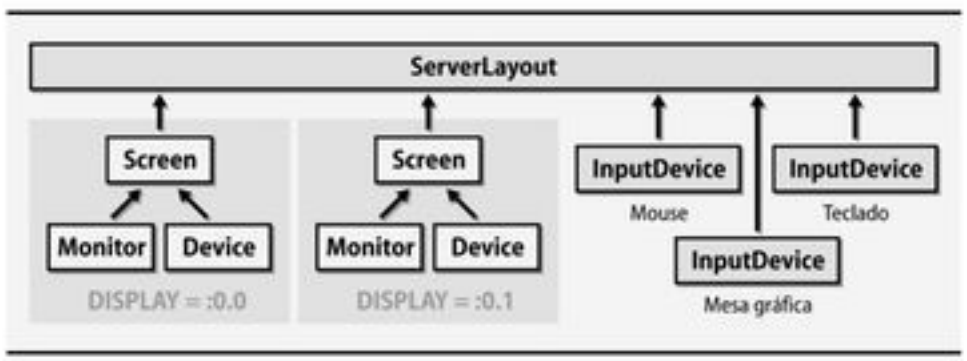
Tabela 22.1 Seções do arquivo xorg.conf

Seção	Descrição
`ServerFlags`	Lista parâmetros de configuração gerais do servidor X.
`Module`	Especifica extensões dinamicamente carregáveis para aceleração gráfica, renderização de fontes e outras extensões semelhantes.
`Device`	Configura as informações de hardware, driver e placa de vídeo.
`Monitor`	Descreve parâmetros físicos do monitor incluindo sincronização e resoluções de vídeo.
`Screen`	Associa um monitor a uma placa de vídeo (Device) e define as resoluções e profundidades de cor disponíveis nessa configuração.
`InputDevice`	Especifica os dispositivos de entrada como teclados e mouses.
`ServerLayout`	Empacota os dispositivos de entrada em um conjunto de telas e posiciona as telas uma em relação a outra.

Freqüentemente é mais simples construir um arquivo de configuração de baixo para cima definindo primeiro as seções para os dispositivos de entrada e saída e então combinando-os em vários layouts. Com essa abordagem hierárquica, um único arquivo de configuração pode ser utilizado para muitos servidores X, cada um com um hardware diferente. Também é uma abordagem razoável para um único sistema com múltiplas placas de vídeo e monitores.

A Ilustração C a seguir mostra como algumas dessas seções são agrupadas na hierarquia de configuração do X.Org. Uma exibição física `Monitor` mais uma placa de vídeo `Device` formam um `Screen`. Um conjunto de `Screen`s mais `InputDevice`s formam um `ServerLayout`. Múltiplos layouts de servidor podem ser definidos em um arquivo de configuração, embora somente um possa estar ativo para uma dada instância do processo **Xorg**.

Ilustração C Relacionamento entre as seções de configuração do xorg.conf



Algumas seções que compõem o arquivo **xorg.conf** são relativamente fixas. Na maioria das vezes os padrões podem ser utilizados diretamente de um arquivo de configuração existente ou de um arquivo de configuração de exemplo. Outros, como as seções `Device`, `Monitor`, `Screen`, `InputDevice` e `ServerLayout`, dependem da configuração de hardware do host. Discutimos a mais interessante dessas seções em mais detalhes nas subseções a seguir.

Seções `Device`

Uma seção `Device` descreve uma placa de vídeo em particular. Você deve fornecer uma string para identificar a placa e um driver apropriado para o dispositivo. O driver só é carregado se o dispositivo for referenciado por uma seção `Screen` correspondente. Uma seção típica de dispositivo poderia se parecer com isto:

```
Section "Device"
 Identifier        "STBVirge3D"
 Driver            "s3virge"
 opção             valor
 ...
EndSection
```

A página manual para o driver, nesse exemplo `s3virge`, descreve o hardware em que está baseado e as opções que o driver suporta. Se você estiver experimentando artefatos de vídeo estranhos, tente configurar as opções para desativar a aceleração de hardware (se suportada), diminuir a velocidade de acesso da memória de vídeo ou modificar os parâmetros PCI. Geralmente uma boa idéia é verificar na Web outras pessoas que talvez já tenham experimentado problemas semelhantes antes de você começar a mudar os valores aleatoriamente.

Seções `Monitor`

A seção `Monitor` descreve as exibições anexadas ao seu computador. Você pode especificar valores detalhados de sincronização. As informações de sincronização são necessárias para hardware mais antigo, mas também é possível investigar essas informações na maioria dos monitores modernos. As especificações de exibição normalmente podem ser obtidas do site do fabricante, mas não há nada melhor que o manual original que acompanha o monitor. De qualquer maneira, é recomendável conhecer pelo menos a sincronização horizontal e as freqüências de atualização vertical do seu modelo.

Uma seção `Monitor` típica se parece com isto:

```
Section "Monitor"
    Identifier     "ViewSonic"
    Option         "DPMS"
    HorizSync      30-65
    VertRefresh    50-120
 EndSection
```

Como ocorre com todas as seções, a linha `Identifier` atribui um nome com que você mais tarde identifica esse monitor. Aqui ativamos DPMS (Display Power Management Signaling) de modo que o servidor X desligue o monitor quando não estamos trabalhando.

As linhas `HorizSync` e `VertRefresh` devem ser preenchidas com os valores apropriados do seu monitor. Elas podem ser especificadas como um intervalo de freqüência (como anteriormente) ou como valores discretos separados por vírgulas. O driver pode teoricamente investigar os modos suportados, mas especificar os parâmetros impede que o driver tente utilizar freqüências não suportadas.

Há inúmeras histórias de horror dos primeiros CRTs sendo danificados por sinais que executam em freqüências impróprias, mas hoje em dia CRTs parecem ser um pouco mais resistentes. No pior dos cenários, é provável que eles emitam um chiado agudo que certamente chamará a atenção do seu cachorro. Os modernos monitores de LCD são ainda mais tolerantes a variações de sinal, mas provavelmente ainda é prudente ter cuidado ao testar freqüências de monitor. Esteja preparado para desligar o monitor se não gostar do sinal que ele está recebendo.

Seções `Screen`

Uma seção `Screen` associa um dispositivo (placa de vídeo) a um monitor em uma profundidade de cor específica e em um conjunto de resoluções de vídeo. Eis um exemplo que utiliza a placa de vídeo e o monitor especificados anteriormente.

```
Section "Screen"
    Identifier     "Screen 2"
    Device         "STBVirge3D"
    Monitor        "ViewSonic"
    DefaultDepth24
    Subsection "Display"
       Depth       8
```

```
    Modes        "640x400"
  EndSubsection
  Subsection "Display"
    Depth        16
    Modes        "640x400" "640x480" "800x600" "1024x768"
  EndSubsection
  Subsection "Display"
    Depth        24
    Modes        "1280x1024" "1024x768" "800x600" "640x400" "640x480"
  EndSubsection
EndSection
```

Como você poderia esperar, a tela é identificada com um `Identifier` e os identificadores para o dispositivo de vídeo e o monitor anteriormente definidos são mencionados. Essa é a primeira seção que introduzimos que tem subseções. Uma subseção é definida para cada profundidade de cor, com o padrão sendo especificado pelo campo `DefaultDepth`.

Uma dada instância do servidor X só pode ser executada em uma profundidade de cor. Na inicialização, o servidor determina quais resoluções são suportadas para essa profundidade de cor. As possíveis resoluções geralmente dependem da quantidade de memória na placa de vídeo. Em placas mais antigas com menos memória, é comum que a resolução esteja limitada a profundidades High Color. A seção *Combinações especiais de teclas para o X*, na página 520, descreve como alternar pelas resoluções que são definidas aqui.

Qualquer placa de vídeo moderna decente deve ser capaz de colocar seu monitor na resolução máxima com as profundidades de cor de 24 bits ou de 32 bits. Se quiser executar programas antigos que requerem um servidor em execução na profundidade de cor de 8 bits, execute um segundo servidor X em um console virtual separado. Utilize o flag **-depth 8** na a linha de comando **Xorg** para sobrescrever a opção `DefaultDepth`.

Seções `InputDevice`

Uma seção `InputDevice` descreve uma origem de eventos de entrada como um teclado ou mouse. Cada dispositivo recebe sua própria seção `InputDevice` e, como ocorre com outras seções, cada uma é nomeada com um campo `Identifier`. Se estiver compartilhando um único arquivo de configuração entre máquinas com hardware diferente, você pode definir todos os dispositivos de entrada; somente aqueles na seção `ServerLayout` referenciada são usados. Eis uma definição típica de teclado:

```
Section "InputDevice"
    Identifier     "Generic Keyboard"
    Driver "Keyboard"
    Option "AutoRepeat" "500 30"
    Option "XkbModel" "pc104"
    Option "XkbLayout" "us"
EndSection
```

Você pode configurar opções na definição do teclado de acordo com sua preferência pessoal sobre a posição adequada das teclas Control e Caps Lock, entre outras coisas. Neste exemplo, a opção `AutoRepeat` especifica quanto tempo uma tecla precisa ser mantida pressionada antes de começar a repetir e a velocidade com que ela se repete.

O mouse é configurado em uma seção `InputDevice` separada:

```
Section "InputDevice"
    Identifier     "Generic Mouse"
    Driver         "mouse"
    Option         "CorePointer"
    Option         "Device" "/dev/input/mice"
    Option         "Protocol" "IMPS/2"
    Option         "Emulate3Buttons" "off"
    Option         "ZAxisMapping" "4 5"
EndSection
```

A opção `CorePointer` designa esse mouse como o principal dispositivo indicador do sistema. O arquivo do dispositivo associado com o mouse é especificado como uma `Option`; em geral, ele é configurado como **/dev/input/mice**, que é o multiplexador de dispositivo de mouse. O protocolo depende da marca particular do mouse utilizado; você pode configurá-lo como `auto` para que o servidor tente descobri-lo para você. Se a roda do seu mouse não funciona, tente configurar o protocolo como `IMPS/2`. Se tiver mais que dois ou três botões, talvez seja necessário utilizar o protocolo `ExplorerPS/2`.

Se **/dev/input/mice** não funcionar para seu mouse, a configuração será um pouco mais complexa. O programa **gpm** implementa recursos do tipo X para ações controladas pelo mouse e de recortar e colar nos terminais virtuais de modo texto. Entretanto, somente um programa pode abrir um dispositivo tradicional de mouse em um dado momento. Para resolver esse problema, o **gpm** replica os dados do mouse em um arquivo FIFO de modo que aplicativos como o servidor X

possam vê-lo também (utilizando o FIFO como o dispositivo de mouse).[1] Esse arranjo adiciona a funcionalidade **gpm** ao sistema e ao mesmo tempo o mantém relativamente transparente.

Por exemplo, o comando a seguir faz com que **gpm** obtenha sua entrada de **/dev/mouse** com o protocolo IMPS/2 e o encaminhe ao FIFO **/dev/gpmdata** (o nome não é configurável) sem nenhuma conversão de protocolo.

```
$ gpm -m /dev/mouse -t imps2 -Rraw
```

Você então alteraria o dispositivo de mouse para o arquivo **/dev/gpmdata** no **xorg.conf**. Como o **gpm** deve ser executado antes de o servidor X ser iniciado, esse comando deve entrar em um script de inicialização de sistema como **/etc/init.d/gpm**. Consulte a página 23 para informações adicionais sobre scripts de inicialização.

A opção `Emulate3Buttons` permite que um mouse de dois botões emule um mouse de três botões definindo um clique em ambos os botões para servir como um substituto ao clique com o botão do meio. A opção `ZAxisMapping` às vezes é necessária para suportar uma roda de rolagem ou dispositivo joystick mapeando os botões apropriadamente. A maioria dos mouses atuais tem pelo menos três botões, uma roda de rolagem, um player de MP3 embutido, um massageador de pés e um isopor para cerveja.[2]

Seções `ServerLayout`

A seção `ServerLayout` é o nó de primeiro nível da hierarquia de configuração. Cada configuração de hardware em que o servidor será executado deve ter sua própria instância da seção `ServerLayout`. O layout utilizado por um servidor X particular normalmente é especificado na linha de comando do servidor.

Eis um exemplo de uma seção `ServerLayout` completa:

```
Section "ServerLayout"
 Identifier        "Simple Layout"
 Screen            "Screen 1" LeftOf "Screen 2"
 Screen            "Screen 2" RightOf "Screen 1"
 InputDevice       "Generic Mouse" "CorePointer"
 InputDevice       "Generic Keyboard" "CoreKeyboard"
 Option            "BlankTime"    "10"   # Escurece a tela em 10 minutos
 Option            "StandbyTime"  "20"   # Desliga o monitor em 20 minutos (DPMS)
 Option            "SuspendTime"  "60"   # Hibernação completa em 60 minutos (DPMS)
 Option            "OffTime"      "120"  # Desliga o monitor DPMS em 2 horas
EndSection
```

Essa seção agrupa todas as outras seções para representar uma exibição X. Ela inicia com o requisito `Identifier`, que nomeia esse layout em particular. Ela então associa um conjunto de telas com o layout.[3] Se múltiplos monitores estiverem anexados a placas de vídeo separadas, cada tela será especificada juntamente com direções opcionais a fim de indicar como eles são fisicamente organizados. Nesse exemplo, a tela um está à esquerda e a tela dois está à direita.

Algumas placas de vídeo podem controlar vários monitores ao mesmo tempo. Nesse caso, somente uma única `Screen` é especificada na seção `ServerLayout`. Para placas NVIDIA, atualmente a mais comum para essa aplicação sob o Linux, você configura uma opção na seção `Driver` para significar suporte a `TwinView`. Os detalhes dessa configuração estão além do escopo deste livro, mas podem ser facilmente localizados nos vários fóruns na Web.

Depois da lista de telas há o conjunto de dispositivos de entrada associado a esse layout. As opções `CorePointer` e `CoreKeyboard` são passadas para a seção `InputDevice` para indicar que os dispositivos devem estar ativos para a configuração. Essas opções também podem ser configuradas diretamente nas seções `InputDevice` correspondentes, mas fica mais claro configurá-las na seção `ServerLayout`.

As últimas poucas linhas configuram várias opções específicas ao layout. No exemplo anterior, todas estão relacionadas ao DPMS, que é a interface que instrui monitores compatíveis com Energy Star sobre quando se autodesligarem. Os monitores também devem ter suas opções DPMS ativadas nas seções `Monitor` correspondente.

22.4 SOLUÇÃO DE PROBLEMAS E DEPURAÇÃO

A configuração do servidor X percorreu um longo caminho na última década, mas ainda pode ser difícil fazer com que as coisas funcionem da maneira como você gostaria. Talvez você precise testar freqüências de monitor, opções de driver, drivers proprietários ou extensões para renderização em 3D. De modo irônico, é precisamente quando o vídeo não está

1. Os arquivos FIFO são criados com o comando **mknod**. Por exemplo, **mknod p /dev/gpmdata**.
2. Nem todas as opções são suportadas pelo **Xorg**. Algumas opções são vendidas separadamente.
3. Lembre-se de que telas identificam uma combinação de placa de vídeo/monitor em uma profundidade de cor particular.

funcionando corretamente é que você tem mais interesse em ver a saída de depuração na tela. Felizmente, o servidor X.Org fornece todas as informações de que você precisa (e muitas de que não precisa) para identificar o problema.

Combinações especiais de teclas para o X

Como o servidor X assume controle do seu teclado, tela, mouse e da sua vida social, você poderia pensar que a única alternativa que você tem é desligar o sistema se as coisas não estiverem funcionando. Entretanto, há algumas coisas a tentar antes disso. Se mantiver pressionadas as teclas Control e Alt e pressionar uma tecla de função (F1-F6), o servidor X o leva a um dos terminais virtuais baseado em texto. A partir daí você pode efetuar login e depurar o problema. Para voltar ao servidor X em execução no terminal virtual 7, pressione <Alt-F7>.[4] Se estiver em uma rede, você também pode tentar efetuar login a partir de um outro computador para encerrar o servidor X antes de recorrer ao botão de reinicialização.

Se o monitor não estiver em sincronia com o sinal de vídeo da placa, tente alterar a resolução de tela. As resoluções disponíveis são especificadas em uma linha `Modes` a partir da seção `Screen` do arquivo de configuração. A linha `Modes` exata que está ativa depende da profundidade de cor; veja *Seções Screen*, na página 517, para detalhes. O servidor X assume por padrão a primeira resolução mostrada na linha `Modes` ativa, mas você pode verificar diferentes resoluções mantendo pressionadas Control e Alt e pressionando a tecla de sinal de adição (+) ou de subtração (-) no teclado numérico.

Pressionar <Control-Alt-Backspace> encerra o servidor X imediatamente. Se executou o servidor a partir de um console, você retornará a esse local quando o servidor for encerrado. Se um gerenciador de vídeo iniciou o servidor, ele normalmente regenera um novo servidor e solicita novamente um login e uma senha. Você tem de encerrar o gerenciador de vídeo (**xdm**, **gdm** etc.) a partir de um console de texto a fim impedir que ele gere novos servidores X.

Quando bons servidores X falham

Depois que recuperou o controle da máquina, você pode começar a rastrear o problema. O ponto de partida mais simples é a saída do servidor X. Essa saída permanece ocasionalmente visível no terminal virtual um (<Control-Alt-F1>), que é para onde vai toda saída do programa de inicialização. Mais freqüentemente, a saída do servidor X é armazenada em um arquivo de log como **/var/log/Xorg.0.log**.

Como visto abaixo, cada linha é precedida por um símbolo que a categoriza. Você pode utilizar esses símbolos para identificar erros (`EE`) e alertas (`WW`) e também determinar como o servidor descobriu cada trecho de informações: por meio das configurações padrão (`==`), em um arquivo de configuração (`**`), detectado automaticamente (`--`) ou especificado na linha de comando do servidor X (`++`).

Vamos examinar o trecho a seguir:

```
X Window System Version 6.8.2
Release Date: 9 February 2005
X Protocol Version 11, Revision 0, Release 6.8.2
Build Operating System: Linux 2.4.21-23.ELsmp i686 [ELF]
Current Operating System: Linux chinook 2.6.12-1.1372_FC3 #1 Fri Jul 15 00:59:10 EDT 2005 i686
Markers: (--) probed, (**) from config file, (==) default setting,
        (++) from command line, (!!) notice, (II) informational,
        (WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.0.log", Time: Mon May  1 08:41:02 2006
(==) Using config file: "/etc/X11/xorg.conf"
(==) ServerLayout "Default Layout"
(**) |-->Screen "Screen0" (0)
(**) |   |-->Monitor "Monitor0"
(**) |   |-->Device "Videocard1"
(**) |-->Input Device "Mouse0"
(**) |-->Input Device "Keyboard0"
```

As primeiras linhas informam o número de versão do servidor X e a versão do protocolo X11 que ela implementa. As linhas subseqüentes informam que o servidor está utilizando valores padrão para a localização do arquivo de log, para a localização do arquivo de configuração e para o layout de servidor ativo. A exibição e os dispositivos de entrada do arquivo de configuração são ecoados de uma forma esquemática.

Um problema comum que aparece nos logs é a dificuldade com certas resoluções de tela, normalmente evidenciada pelas resoluções que não funcionam ou que fazem com que o servidor X encerre com um erro como "Unable to validate any modes; falling back to the default mode." (Incapaz de validar quaisquer modos; recorrendo ao modo padrão). Se você não

4. O servidor X exige que a tecla <Control> seja mantida pressionada junto com a combinação de teclas <Alt-Fn> para alternar entre terminais virtuais, mas o console de texto não exige isso.

especificou uma lista de freqüências para seu monitor, o servidor X investiga essas freqüências utilizando Extended Display Identification Data (EDID). Se seu monitor não suportar EDID ou se seu monitor estiver desligado quando o X é iniciado, você precisa colocar os intervalos de freqüência a ser usado pelo X na seção `Monitor` do arquivo de configuração.

Erros de arredondamento nos resultados obtidos a partir de uma investigação EDID podem tornar algumas resoluções indisponíveis mesmo que elas sejam suportadas tanto pela sua placa de vídeo como pelo monitor. Entradas de log como "No valid modes for 1280x1024; removing" são evidências disso. A solução é para instruir o servidor X a ignorar as informações EDID e utilizar as freqüências que você especifica com estas linhas para a seção `Device`:

```
Option     "IgnoreEDID" "true"
Option     "UseEdidFreqs" "false"
```

Como um outro exemplo, suponha que você se esqueceu de definir a seção mouse adequadamente. O erro apareceria como isto na saída:

```
(==) Using config file: "/etc/X11/xorg.conf"
Data incomplete in file /etc/X11/xorg.conf
        Undefined InputDevice "Mouse0" referenced by ServerLayout "Default Layout".
(EE) Problem parsing the config file
(EE) Error parsing the config file
Fatal server error:
no screens found
```

Depois que o X iniciou e está em execução, e você já efetuou o login, execute o comando **xdpyinfo** para obter informações adicionais sobre a configuração do servidor X.[5] A saída do **xdpyinfo** informa novamente o nome da exibição e as informações sobre a versão do servidor X. Ele também informa as profundidades de cor que estão disponíveis, as extensões que foram carregadas e as telas que foram definidas, juntamente com as configurações de dimensões e de cor.

A saída do **xdpyinfo** pode ser analisada sintaticamente por um script (como seu arquivo **~/.xsession**) para determinar o tamanho da tela ativa e para configurar os parâmetros de área de trabalho apropriadamente. Para depurar, **xdpyinfo** é mais útil para determinar que o servidor X está ativo para ouvir consultas de rede, que ele configurou a tela e a resolução correta, e para operar na profundidade de bits de cor desejada. Se esse passo funcionar, você está pronto para começar a executar aplicativos X.

22.5 UMA BREVE NOTA SOBRE OS AMBIENTES DE ÁREA DE TRABALHO

A flexibilidade e a simplicidade do modelo cliente/servidor do X levou, ao longo dos anos, a uma explosão de conjuntos de componentes (*widgets*), gerenciadores de janelas, navegadores de arquivo, utilitários de barra de ferramenta e programas utilitários. A partir do Project Athena no MIT, no qual o X tem suas raízes, surgiram os componentes Athena e **twm** (Tom's Window Manager, nome dado por causa do seu criador, Tom LaStrange; ele também é chamado Tab Window Manager). Essas ferramentas rudimentares formaram o padrão de fato para os primeiros aplicativos X.

OpenLook, desenvolvido pela Sun Microsystems e AT&T, era um kit de ferramentas alternativo que introduziu botões ovais e alfinetes para manter menus e caixas de diálogo visíveis. Mais ou menos na mesma época, a Open Software Foundation introduziu a plataforma Motif concorrente (mais tarde chamada CDE, ou Common Desktop Environment), que por fim também foi adotada pela Sun. Esses kits de ferramentas com uma aparência cinza tridimensional era elegante para aquela época e o predomínio de elementos de interface avançados como controles deslizantes e menus os tornaram uma escolha razoável para novos softwares. Mas esses dois kits de ferramentas eram altamente proprietários, e taxas de licenciamento para as bibliotecas de desenvolvimento e para o gerenciador de janelas tornaram-nos inacessíveis ao público geral.

Junto com os sistemas operacionais de código-fonte aberto, como o Linux, surgiram os ambientes de área de trabalho de código-fonte aberto. O FVWM (o Virtual Window Manager "F") foi popular para o Linux por causa do seu alto grau de configurabilidade e suporte a "áreas de trabalho virtuais", que expandiram a área de trabalho efetiva do usuário para além dos limites das telas de baixa resolução disponíveis na maioria dos PCs nessa época.[6] Mas não havia nenhum conjunto de componentes associado, portanto os usuários continuavam confrontados com uma variedade de programas, cada um com aparência e comportamento bastante diferentes.

À medida que os aplicativos tornavam-se cada vez mais avançados e exigiam funcionalidades da interface com o usuário mais avançadas, ficou claro que uma abordagem mais ampla precisava ser adotada para unificar a experiência do usuário e fornecer melhor suporte aos desenvolvedores de aplicações. Dessa necessidade nasceram os dois grandes fornecedores de

5. Não recomendamos efetuar o login no X como root porque essa operação pode criar vários arquivos de inicialização padrão no diretório inicial do root, que é normalmente /. Isso também é notavelmente inseguro. Em vez disso, efetue o login como um usuário regular e use **sudo**. O Debian e o Ubuntu impõem essa disciplina por padrão.

6. Na realidade, o FVWM era tão flexível que ele poderia ser configurado para parecer a um **twm** ou **mwm** (o).

ambientes de área de trabalho Linux modernos: GNOME e KDE. Embora alguns usuários tenham opiniões marcantes sobre qual desses ambientes é realmente a solução mais completa, os dois são gerenciadores de área de trabalho relativamente completos. De fato, só porque você trabalha com um deles não significa que você não possa utilizar aplicativos do outro; espere apenas um visual e comportamento diferentes e uma breve sensação de descontinuidade no universo.

O projeto freedesktop.org é dedicado a criar um ambiente que permitirá que aplicativos sejam compatíveis com qualquer ambiente de área de trabalho.

KDE

O KDE que, em inglês significa K Desktop Environment, é escrito em C++ e construído sobre biblioteca do kit de ferramentas Qt. Ele costuma ser preferido pelos usuários que gostam de embelezamentos visuais, como janelas transparentes, sombras e cursores animados. Parece interessante, mas ele pode ficar lento em qualquer máquina que não seja um PC topo de linha. Para os usuários que passam muito tempo clicando em diferentes locais na área de trabalho em vez de executar aplicativos, a relação de troca ou compensação entre o visual e o comportamento, em última instância, pode decidir se o KDE é a escolha apropriada.

O KDE freqüentemente é preferido pelas pessoas que estão migrando a partir de um ambiente Windows ou Mac por causa das bonitas imagens gráficas. Ele também e o favorito dos tecnófilos que adoram ser capazes de personalizar seus ambientes completamente. Para outros, trabalhar com o KDE é simplesmente muito complicado e o GNOME é a escolha mais simples.

Os aplicativos escritos para o KDE quase sempre contêm um K em algum lugar no nome, como Konqueror (o navegador Web/gerenciador de arquivos), Konsole (o emulador terminal) ou KWord (um processador de texto). O gerenciador padrão de janelas, KWin, suporta o padrão Window Manager Specification do freedesktop.org, "capas" ou "temas" (*skins*) configuráveis para alterar todo o visual e o comportamento e muitos outros recursos. O conjunto de aplicativos KOffice contém processadores de texto, planilhas e programas de apresentação. O KDE apresenta um conjunto abrangente de ferramentas de desenvolvimento, incluindo um ambiente de desenvolvimento integrado (*integrated development environment* – IDE). Com essas bases, o KDE fornece uma experiência poderosa e consistente de interface com o usuário.

GNOME

GNOME é um outro ambiente de área de trabalho escrito em C e baseado no conjunto de componentes GTK+. O modelo subjacente de comunicação de objetos do GNOME utiliza CORBA, a Common Object Request Broker Architecture. O nome GNOME era originalmente um acrônimo para GNU Network Object Model Environment, mas essa derivação na verdade não é mais aplicada; hoje em dia, o GNOME é apenas um nome.

O GNOME é menos "adornado" que o KDE, não é tão configurável e no geral é um pouco menos consistente. Mas é notavelmente mais limpo, mais rápido e mais simples. A maioria das distribuições Linux utiliza o GNOME como o ambiente de área de trabalho padrão.

Como ocorre com o KDE, GNOME tem um rico conjunto de aplicativos. Os aplicativos GNOME são normalmente identificáveis pela presença de um G nos seus nomes. Uma exceção é o gerenciador de janelas GNOME padrão, chamado Metacity, que fornece funções básicas de um sistema de janelas e capas ou temas para um visual e comportamento configuráveis. Seguindo o modelo GNOME, o Metacity é projetado para ser limpo e simples. Se quiser alguns recursos extras com os quais você está acostumado, uma área de trabalho virtual ou posicionamento inteligente das janelas, você precisará do suporte a aplicativos externos como **brightside** ou **devilspie**. (Essa é uma das áreas em que o KDE leva vantagem.)

Os aplicativos comerciais incluem o AbiWord para processamento de texto, o Gnumeric como uma planilha e um dos projetos mais impressionantes proveniente do GNOME, o GIMP para processamento de imagens. Um gerenciador de arquivos chamado Nautilus também é incluído, junto com Epiphany para navegação pela Web. Como com o KDE, o GNOME fornece uma infra-estrutura extensa para desenvolvedores de aplicativos. No todo, o GNOME oferece uma arquitetura poderosa para desenvolvimento de aplicativos em um ambiente de área de trabalho fácil de usar.

Qual é o melhor, GNOME ou KDE?

Faça essa pergunta em qualquer fórum público e você verá a definição de "*flame war*" (discussões acaloradas). Devido à tendência das pessoas em transformar uma preferência por uma área de trabalho em uma cruzada pessoal, os parágrafos a seguir talvez sejam alguns dos menos dogmáticos neste livro.

A melhor resposta é experimentar as duas áreas de trabalho e decidir qual melhor atende a suas necessidades. Tenha em mente que seus amigos, seus usuários e seu gerente talvez tenham preferências diferentes por um ambiente de área de trabalho e isso é normal.

Agora que freedesktop.org está criando padrões para unificar as áreas de trabalho, a animosidade desenvolvida entre as áreas do KDE e do GNOME começa a ser uma competição saudável para a criação de excelentes softwares. Lembre-se de que a escolha do seu ambiente de área de trabalho não determina quais aplicativos você pode executar. Independentemente

da área de trabalho escolhida, é possível selecionar aplicativos a partir de uma grande variedade de excelentes softwares disponibilizados por esses dois projetos (e outros) de código-fonte aberto.

22.6 LEITURA RECOMENDADA

A home page do X.org, x.org, inclui informações sobre as próximas distribuições e também links para o wiki X.Org, listas de discussão e downloads.

As páginas man do **Xserver** e do **Xorg** abrangem opções de servidor X genéricas e opções de linha de comando específicas ao **Xorg**. Elas também incluem uma visão geral da operação do servidor X. A página man do **xorg.conf** aborda o arquivo de configuração e descreve suas várias seções em detalhes. Essa página man também lista drivers de placas de vídeo na seção REFERENCES. Pesquise sua placa de vídeo aqui para descobrir o nome do driver e então leia a página man do driver para aprender as opções específicas a esse driver.

22.7 EXERCÍCIOS

E22.1 Utilize SSH para executar um programa X pela rede. Utilize **ssh -v** para verificar se o encaminhamento do X está configurado corretamente. Qual é a configuração da variável DISPLAY depois de você efetuar login? Liste os cookies executando **xauth** e verifique se autenticação por meio da cookie mágica está ativa para essa exibição.

E22.2 Escreva uma linha de comando de shell ou script de shell para analisar sintaticamente a saída de **xdpyinfo** e imprimir a resolução atual de tela no formata XxY, por exemplo, 1024 × 768.

E22.3 Examine o arquivo de log **Xorg** (**/var/log/Xorg.0.log**) e determine o maior número possível destes itens:

a) Qual tipo de placa de vídeo está presente e qual driver ela utiliza?

b) Quanta memória de vídeo a placa tem?

c) O EDID foi utilizado para investigar as configurações do monitor? Como você descobre isso?

d) Quais modos (resoluções) são suportados?

e) O DPMS está ativado?

f) Quais dimensões físicas de tela o servidor considera?

g) Qual arquivo de dispositivo é utilizado para o mouse?

E22.4 Qual flag desativa as conexões TCP não-locais para o servidor? Explique por que essa opção é útil.

23 *Impressão*

Configuração de impressora é irritante e difícil. Os usuários acham que imprimir é fácil, mas os malabarismos administrativos necessários para que páginas sejam apresentadas com perfeição em uma impressora a um metro do usuário podem ser desafiadores.

Há duas décadas, as impressoras mais comuns eram impressoras de linha ASCII. Impressoras a laser costumavam ser raras e caras. Dispositivos de saída de alta resolução exigiam drivers e programas de formatação personalizados.

Hoje, em vez de conectarem-se a um único computador por meio de uma porta paralela serial, impressoras a laser são freqüentemente conectadas a uma rede TCP/IP por uma Ethernet ou link sem fio. Impressoras a laser perderam uma grande fatia de mercado para impressoras de jato de tinta menos sofisticadas. Impressoras coloridas eram um luxo, mas, como ocorreu com a fotografia colorida e monitores coloridos, hoje, elas são comuns. Encontrar uma impressora preto-e-branco será tão difícil quanto encontrar um filme preto-e-branco.

Na área de trabalho e no mercado dos pequenos escritórios, impressoras de uso especial, scanners, copiadoras e máquinas de fax estão sendo substituídos por dispositivos multifuncionais que realizam todas essas tarefas. Às vezes, esses dispositivos conseguem até mesmo ler arquivos do cartão de memória da sua câmera digital.

Com tantas mudanças na tecnologia, poderíamos esperar que o sistema de impressão do Linux fosse flexível e, de fato, é. Mas essa flexibilidade é uma realização relativamente recente. Até há poucos anos, a maioria dos sistemas de impressão Linux estava baseada em softwares desenvolvidos para impressoras de linhas de antigamente. Esses sistemas, modificados e sobrecarregados como uma tentativa para que se mantivessem atualizados com as tecnologias em evolução, nunca estavam realmente prontos para o trabalho de suportar impressoras modernas em redes modernas. Felizmente, CUPS, o Common UNIX Printing System, entrou em cena para tratar muitas das fraquezas dos sistemas mais antigos.

Você pode encontrar o CUPS nos sistemas UNIX e Linux mais modernos, incluindo Mac OS X. Alguns sistemas de impressão mais antigos continuam em uso (como o PDQ, pdq.sourceforge.net, e o LPRng, www.lprng.com), mas os primeiros sistemas de impressão, como os do System V, Palladium, rlpr, PLP, GNUlpr e PPR V, estão todos mortos.

Neste capítulo, nosso foco será o CUPS como o padrão de fato atual. Começamos com uma discussão geral sobre impressoras e a terminologia de impressão. Então descreveremos os sistemas de impressão Linux em geral e delinearemos a arquitetura do CUPS. Em seguida, abordaremos as especificidades da configuração e administração de uma impressora e então concluiremos com um guia rápido para depuração do sistema de impressão, fazendo um tour pelos softwares opcionais relacionados à impressão e algumas dicas gerais de administração.

23.1 IMPRESSORAS SÃO COMPLICADAS

Os usuários agrupam impressoras com outros periféricos como monitores e alto-falantes, mas esse ponto de vista não revela a complexidade das impressoras. Há muito tempo, o mais poderoso computador que a Apple fabricou foi o Apple LaserWriter. Hoje, sua máquina desktop provavelmente é mais poderosa do que sua impressora, mas a impressora continua sendo um computador. Ela tem uma CPU, memória, um sistema operacional e talvez até mesmo um disco. Se for uma impressora de rede, ela tem seu próprio endereço IP e implementação TCP/IP.

Se você tiver uma impressora de rede moderna, insira o endereço de rede dela no navegador Web (por exemplo, 192.168.0.9). Provavelmente a impressora retornará algumas páginas Web que permitem administrar o hardware da impressora; a impressora está executando um servidor Web próprio.

Como os administradores de sistema se preocupam com a segurança, você poderia pensar: "Quer dizer que uma impressora poderia ser comprometida ou atingida por um ataque de negação de serviço?". Pode apostar que sim. Consulte a seção sobre segurança a partir da página 541.

Qual sistema operacional sua impressora está executando? O quê? Você não sabe? Não é uma surpresa. Provavelmente você também não consegue descobrir sem uma boa pesquisa — e talvez nem mesmo com ela. O sistema operacional varia

entre fornecedores e às vezes até mesmo entre um modelo e outro. As impressoras de médio porte mais sofisticadas podem até mesmo executar algum derivado do UNIX ou do Linux.[1]

A confusão de SO só está começando. Impressoras também tratam uma variedade de protocolos de rede e aceitam trabalhos em diferentes linguagens de descrição de páginas e de descrição de documentos específicos de impressora.

Se estiver administrando uma instalação maior, talvez você precise suportar vários modelos de impressoras de vários fabricantes diferentes. O software de impressão nos seus computadores deve estar preparado para se comunicar com vários tipos de hardware (e às vezes alguns desconhecidos) e utilizar vários protocolos.

23.2 LINGUAGENS DE IMPRESSORA

Um trabalho de impressão é na verdade um programa de computador escrito em uma linguagem de programação especializada. Essas linguagens de programação são conhecidas coletivamente como linguagens de descrição de página (*page description language* – PDLs).

Páginas codificadas em uma PDL podem ser bem menores e mais rápidas de transmitir que as imagens brutas equivalentes. Descrições PDL também podem ser independentes de dispositivo e de resolução.

As PDLs mais famosas hoje são PostScript, PCL5, PCL6 (também chamada PCL/XL ou "pxl") e PDF. Muitas impressoras podem aceitar entrada em mais de uma linguagem. Discutimos cada uma dessas linguagens brevemente nas seções a seguir.

As impressoras têm de interpretar trabalhos nessas linguagens e transformá-los em alguma forma de representação de mapa de bits que faça sentido para o hardware de geração de imagens. Portanto, as impressoras contêm interpretadores de linguagem. Assim como com C ou Java, essas linguagens existem em várias versões e as versões fazem diferença. A maioria das impressoras PostScript entende o PostScript Level 3, mas se você enviar um programa Level 3 a uma impressora que só entende o Level 2, é provável que a impressora fique confusa. Você tentaria compilar um programa FORTRAN 90 com um compilador FORTRAN 77? Certamente não.

Rasterizar (converter) a descrição PDL (ou uma outra coisa qualquer, como arquivos de imagem) em imagens de página de mapa de bits é chamado "processamento de rasterização de imagem" e um programa que faz a rasterização é chamado RIP. O RIP às vezes é utilizado informalmente como um verbo, rasterizar.

É possível rasterizar trabalhos de impressão no seu computador e visualizar as imagens na tela. Discutimos os interpretadores baseados em host que fazem isso, como o Ghostscript, na página 540. Teoricamente, é possível utilizar o computador em trabalhos de rasterização para imprimir e distribuir os mapas de bits completados (e muito maiores) para que sejam impressos por um dispositivo de impressão não muito inteligente. De fato, essa é a maneira como muitas impressoras GDI (Windows) funcionam e isso também é, até certo ponto, suportado pelo Linux.

PostScript

PostScript é a PDL mais comum encontrada nos sistemas Linux. Ele foi originalmente desenvolvido pela Adobe Systems e a maioria das impressoras PostScript utiliza um interpretador licenciado pela Adobe. Quase todos os programas de layout de página podem gerar PostScript e alguns funcionam exclusivamente com PostScript.

O PostScript é, na verdade, uma linguagem de programação madura. Você pode ler a maioria dos programas PostScript com um editor de textos ou com **less**. Os programas contêm um número enorme de parênteses, chaves e barras e, normalmente, começam com os caracteres %!PS. Embora esses caracteres iniciais não sejam exigidos pela linguagem, os interpretadores PostScript e outros softwares de impressão freqüentemente irão procurá-los ao tentar reconhecer e classificar trabalhos de impressão.

PCL

Uma alternativa ao PostScript é a PCL, Printer Control Language, da Hewlett-Packard. Ela é entendida pelas impressoras HP e também por várias outras; algumas impressoras só entendem a PCL. Diferentemente do PostScript, que é uma completa linguagem programação generalizada de Turing, a PCL apenas instrui as impressoras como imprimir páginas. Os trabalhos PCL são binários, não legíveis por humanos, e normalmente são bem mais curtos que os equivalentes em PostScript. Aplicativos Linux raramente geram PCL diretamente, mas os filtros podem converter PostScript em PCL.

Diferentemente do PostScript, cada versão da PCL é um pouco diferente. As diferenças são pequenas, mas significativas o bastante para serem irritantes. Trabalhos impressos corretamente em uma LaserJet 5si talvez não sejam impressos tão corretamente em uma LaserJet 5500, e vice-versa. Isso não ocorre apenas com esses dois modelos; toda impressora PCL tem um dialeto PCL com comandos personalizados que tiram proveito dos recursos dessa impressora.

---

1. Hackers já portaram Linux para iPods e para Xboxes; esperamos ver quem é o primeiro a portá-lo para uma HP LaserJet.

Por exemplo, se informar seu computador de que você tem uma LaserJet 4500 quando, na realidade, você tem uma LaserJet 4550, ele poderá gerar alguns comandos PCL que a 4550 ignora ou interpreta erroneamente. Além disso, se você tiver um trabalho de impressão PCL armazenado — digamos, um formulário de solicitação de compra em branco — e substituir a impressora para a qual ele foi gerado por algo mais recente, talvez seja necessário regenerar o trabalho.

Pior ainda, a HP definiu duas famílias de linguagens praticamente diferentes chamadas PCL: PCL5 (5C significa cor e 5E significa preto e branco) e PCL6 (também chamada PCL/XL). Hoje em dia, é normal que as novas impressoras HP tenham interpretadores de linguagem para essas duas famílias.

PCL4 é uma versão arcaica da PCL5. Trate uma impressora PCL4 (ou anteriores) como você faria com um interpretador Perl 4: substitua essa impressora por algo mais recente.

PDF

O Portable Document Format da Adobe é criado pelo Adobe Acrobat e por muitas outras ferramentas de editoração eletrônica. O OpenOffice, por exemplo, pode exportar documentos como PDF.

Documentos PDF são independentes de plataforma e o PDF é rotineiramente utilizado para trocar documentos eletronicamente tanto para uso on-line como off-line (impresso). O texto final deste livro foi entregue à gráfica como um arquivo PDF.

O PDF é uma linguagem de descrição de documento, não apenas uma linguagem de descrição de página. Ele não descreve apenas páginas individuais, mas também a estrutura geral de um documento: quais páginas pertencem a quais capítulos, quais colunas de texto fluem para outras colunas de texto etc. Ele também acomoda uma variedade de recursos multimídia para utilização na tela.

Algumas impressoras interpretam o PDF diretamente. Se a sua não interpreta, um conjunto de visualizadores e conversores PDF (incluindo Ghostview, **xpdf**, **kpdf**, Evince e Acrobat Reader) podem converter documentos em algo que seja (como PostScript) mais amplamente entendido. Seu sistema de impressão pode até mesmo ocultar o requisito de conversão e automaticamente converter documentos PDF antes de enviá-los à impressora.

XHTML

Na extremidade oposta do espectro, de longe no horizonte, está a XHTML-Print. Uma impressora que recebe um fluxo de dados XHTML-Print (descrevendo, por exemplo, uma página Web) produz uma representação de boa-fé do trabalho, mas impressoras diferentes podem produzir representações diferentes, assim como os navegadores podem representar a mesma página Web de diferentes maneiras.

Por que os usuários iriam querer isso? Imagine que você é uma importante pessoa de marketing navegando pela Web do seu celular e vê uma página Web relevante para um seminário que você está prestes a dar. Você vai até a impressora ativada por Bluetooth mais próxima e envia para ela o URL a partir do seu celular. A impressora faz o resto: ela faz o download da página Web, o renderiza e imprime cópias. Você pega as cópias da bandeja de saída e se dirige para o seminário.

PJL

PJL, Printer Job Language, da Hewlett-Packard, não é realmente uma PDL. Ela é uma metalinguagem que descreve trabalhos de impressora. Nós a descrevemos aqui porque você a verá mencionada nas descrições das impressoras.

PJL é uma linguagem de controle de trabalho que especifica coisas como a PDL de um trabalho, se o trabalho é dúplex ou simplex, o tamanho do papel a utilizar e assim por diante. Os comandos PJL aparecem no início do trabalho, e todas as instruções PJL iniciam com @PJL:

```
@PJL SET COPIES=3
@PJL COMMENT FOO BAR MUMBLE
@PJL SET DUPLEX=ON
@PJL SET PAGEPROTECT=OFF
@PJL ENTER LANGUAGE=PCL
```

A PJL é amplamente entendida (ou deliberadamente ignorada) por impressoras não-HP, mas, se você tiver problemas para imprimir algo que contém PJL em uma impressora não-HP, tente remover a PJL com um editor de textos e enviar o trabalho novamente.

Drivers de impressoras e como eles tratam PDLs

O software que converte um arquivo em algo que uma impressora particular entende é o "driver de impressora". Para imprimir PCL5 em uma LaserJet 5500, você precisa de um driver PCL5 da LaserJet 5500.

E se uma impressora suportar apenas um subconjunto das linguagens que você precisa processar? Se fizer a cópia de um arquivo PostScript da Web e sua impressora só entender PCL5E, o que você faria? Se sua impressora não interpretar PDF diretamente, como você imprime um arquivo PDF?

Uma opção é converter o arquivo manualmente. As máquinas Linux vêm com vários utilitários de conversão; quase sempre há alguma maneira de converter o que você tem em algo que as impressoras possam imprimir. Navegadores podem transformar páginas HTML (ou XHTML) em PostScript. O OpenOffice pode converter arquivos do MS Word em PDF. O Ghostscript pode transformar PDF em PostScript e o PostScript converte quase tudo, inclusive PCL.

Uma abordagem mais fácil é deixar que seu sistema de impressão faça o trabalho para você. Muitos sistemas têm um conhecimento predefinido sobre quais conversões precisam ser feitas e podem configurar as conversões automaticamente.

Se precisar determinar qual PDL um arquivo usa e você não souber isso pelo nome de arquivo (por exemplo, **foo.pdf**), o comando **file** poderá lhe informar qual é a PDL (a menos que o arquivo inicie com um fragmento de instruções PJL, caso em que **file** só informa "HP Printer Job Language data").

Salve alguns trabalhos de impressão em arquivos em vez de mandá-los para uma impressora e você poderá ver com que um programa em uma dessas linguagens se parece. Um ou dois minutos examinando os arquivos de cada um desses tipos no editor de textos dará uma boa idéia da diferença deles. Não os imprima diretamente do **cat** para sua tela, uma vez que apenas o PostScript é ASCII.

PostScript:

```
%!PS-Adobe-3.0
%%BoundingBox: 0 0 612 792
%%Pages: 1
% ...
% Draw a line around the polygons...
pop pop pop dup 0 setgray 0 0 moveto dup 0 lineto 0.707106781 mul dup lineto closepath stroke
```

PDF:

```
%PDF-1.3
%Ã¢Ã£ÃÃ"
 81 0 obj
<<
/Linearized 1
/O 83
/H [ 915 494 ]
/T 125075
>>
endobj
 xref
81 24
0000000016 00000 n
 Ã^<8f>
^P^@Ã¤Ã'<9e>
endstream
endobj
```

PCL5:

```
^[E^[&l1o0o1t016D^[&l1X^[*r0F^[*v0n10^[*p4300X^[%18DT~.1TR0TD1SP1FT10.50
CF3.1LB.~:^[%1A^[*c100G^[*v2T^[&a0P^[*p0X^[*p0Y^[(10U^[(s1p12vsb4148T^[&l0
E^[*p0Y^[*ct7920Y^[(10U^[(s1p12vsb4101T^[&a0P^[&l0o66f0E^[9^[&a0P^[*p0X^[*
p0Y^[*p474Y^[*p141X^[(10U^[(10U^[(s1p12vsb4101T^[*p402Y^[*p186X^[*v00^[*c9
00a4b100g2P^[*v10^[*p250Y^[*v00^[*c900a4b100g2P^[*v10^[*v00^[*c4a156b100g2
P^[*v10^[*p251Y^[*p187X^[*v00^[*c899a154b10g2P^[*v10^[*p346Y^[*p256X
```

PCL/XL:

```
Ã'X^BX^BÃ¸<89>Ã^@Ã¸<86>Ã^CÃ¸<8f>AÃ^@Ã¸<88>Ã^AÃ¸<82>HÃ^@Ã¸(Ã^@Ã¸%Ã
Ã¸cÃ^Ã^P^@TimesNewRmnBdÃ¸Å¯Ã…UUÃ©BÃ¸Å¡Ãu^BÃ¸Å*o<85>Ã"Ã>^CA^BÃ¸Lk
Ãƒ^@^@Ã¸ÅidÃ^Ã:^@
```

23.3 Arquitetura CUPS

O CUPS tem uma arquitetura cliente/servidor. Um servidor CUPS é um spooler que mantém filas de impressão para clientes.

Os clientes CUPS podem ser aplicativos com interfaces de linhas de comando como **lpr** e **lpq** ou podem ter interfaces gráficas com usuário como **kprinter**. Outros servidores CUPS também podem atuar como clientes da perspectiva de um dado servidor.

Considere a mais simples configuração possível: um servidor CUPS em uma única máquina, conectado a uma única impressora, com uma única fila de impressão. Algumas seções a seguir examinam brevemente os comandos e os processos envolvidos em algumas tarefas de impressão comuns.

Impressão de documentos

Eis como você poderia imprimir os arquivos **foo.pdf** e **/tmp/testprint.pdf**:

```
$ lpr foo.pdf /tmp/testprint.ps
```

O cliente transmite cópias dos arquivos ao servidor CUPS, que as armazena na fila de impressão. CUPS processa cada arquivo sucessivamente enquanto a impressora se prepara.

CUPS examina tanto o documento como o arquivo PostScript Printer Description (PPD) da impressora para ver o que precisa ser feito para imprimir o documento adequadamente. (Como explicaremos mais adiante, PPDs são utilizados até mesmo por impressoras não-PostScript.)

Para preparar um trabalho para impressão em uma impressora específica, o CUPS o passa por uma seqüência de filtros. Esses filtros podem desempenhar várias funções. Por exemplo, um filtro poderia reformatar o trabalho de modo que duas imagens reduzidas ao tamanho de uma página sejam impressas em cada página física ("2-up") ou transformar o trabalho de uma PDL em uma outra. Um filtro também pode realizar processamento específico de impressora, por exemplo, inicialização de impressora. Um filtro pode até mesmo realizar uma rasterização (conversão em mapa de bits) em favor de impressoras que não incluem seus próprios RIPs.

A etapa final da seqüência de impressão é um back end que envia o trabalho do host à impressora por meio de um protocolo apropriado como USB. O back end também comunica as informações de status de volta ao servidor CUPS. Para ver os back ends disponíveis, tente este comando:

```
$ locate backend | grep -i cups
```

Depois de transmitir o trabalho de impressão, o daemon CUPS volta a processar suas filas e tratar requisições dos clientes. A impressora tenta imprimir o trabalho entregue.

Visualizando e manipulando a fila de impressão

O comando **lpq** solicita informações sobre o status do trabalho a partir do servidor CUPS e as formata para exibição.

lpstat -t gera um bom resumo do status geral do servidor de impressão.

Os clientes CUPS podem pedir ao servidor para suspender e cancelar tarefas, ou priorizá-las novamente. Eles também podem mover tarefas de uma fila para outra. A maioria das alterações exige que as tarefas sejam identificadas pelo seu número, que é informado pelo **lpq**.

Por exemplo, para remover um trabalho de impressão, simplesmente execute **lprm** *IDdoTrabalho*.

Múltiplas impressoras

Se mais de uma impressora estiver conectada a uma máquina, o CUPS mantém uma fila separada para cada impressora.

Clientes de linha de comando aceitam uma opção (em geral **-P** *impressora* ou **-p** *impressora*) para especificar a fila de impressora. Você mesmo pode especificar uma impressora padrão configurando a variável de ambiente PRINTER

```
$ export PRINTER=nome-da-impressora
```

ou instruindo o CUPS a utilizar um padrão particular para sua conta.

```
$ lpoptions -dnome-da-impressora
```

lpoptions normalmente configura os seus padrões pessoais, que são armazenados em **~/.lpoptions**. Quando executado como root, ele configura padrões no nível de sistema em **/etc/cups/lpoptions**. **lpoptions -l** lista as opções atuais.

Instâncias de impressora

Se você tiver apenas uma impressora, mas quiser utilizá-la de várias maneiras — digamos, tanto para rascunhos rápidos como para a arte-final —, o CUPS permite configurar diferentes "instâncias de impressora" para esses diferentes usos.

Por exemplo, se você já tiver uma impressora chamada Phaser_6120, o comando

```
$ lpoptions -p Phaser_6120/2up -o number-up=2 -o job-sheets=standard
```

cria uma instância chamada Phaser_6120/2up que realiza uma impressão 2-up e adiciona páginas de rosto. O comando:

```
$ lpr -P Phaser_6120/2up biglisting.ps
```

então imprime o arquivo PostScript **biglisting.ps** como um trabalho 2-up com páginas de rosto.

Impressão em rede

Da perspectiva do CUPS, uma rede com muitas máquinas não é muito diferente de uma máquina isolada. Cada computador executa um daemon CUPS (**cupsd**) e todos os daemons CUPS se comunicam entre si.

Você configura um daemon CUPS para aceitar trabalhos de impressão dos sistemas remotos editando o arquivo **/etc/cups/cupsd.conf** (consulte a seção *Configuração do servidor para impressão em rede*, na página 531). Servidores CUPS configurados dessa maneira transmitem, por padrão, informações sobre as impressoras que eles servem a cada 30 segundos. Como resultado, os computadores na rede local automaticamente tomam conhecimento das impressoras disponíveis para eles.

Disponibilizar impressoras para múltiplas redes ou sub-redes é um pouco mais difícil uma vez que os pacotes de broadcast não cruzam os limites da sub-rede. A solução comum é designar um servidor escravo em cada sub-rede que consulta seqüencialmente os servidores de outras sub-redes para informações e, em seguida, transmite essas informações para máquinas em sua sub-rede local.

Por exemplo, suponha que os servidores de impressão allie (192.168.1.5) e jj (192 .168.2.14) residissem em diferentes sub-redes e que quiséssemos que os dois fossem acessíveis aos usuários em uma terceira sub-rede, 192.168.3. Para isso funcionar, simplesmente especificamos um servidor escravo (digamos, copeland, 192.168.3.10) e adicionamos estas linhas ao arquivo **cupsd.conf**:

```
BrowsePoll allie
BrowsePoll jj
BrowseRelay 127.0.0.1 192.168.3.255
```

As primeiras duas linhas instruem o **cupsd** do escravo a consultar seqüencialmente os **cupsd**s em allie e em jj para obter informações sobre as impressoras que eles servem. A terceira linha instrui copeland a retransmitir todas as informações que ele sabe para sua própria sub-rede.

Você precisa de uma configuração mais sofisticada? Múltiplas filas para uma impressora, cada uma com diferentes padrões? Um único servidor realizando balanceamento de carga por meio da distribuição de trabalhos para várias impressoras? Múltiplos servidores cada um tratando instâncias intercambiáveis do mesmo tipo de impressora? Clientes LPD ou Windows? Há tanta variação para analisar aqui, mas o CUPS trata todas essas situações e a documentação do CUPS o orienta passo a passo.

O protocolo sob o CUPS: HTTP

HTTP é o protocolo subjacente para todas as interações entre servidores CUPS e seus clientes. Os servidores CUPS ouvem conexões na porta 631. Os clientes enviam trabalhos com a operação HTTP POST. As requisições de status são implementadas pelo HTTP GET. Os arquivos de configuração CUPS também são notavelmente semelhantes aos arquivos de configuração Apache.

Um pouco de história talvez o ajude a entender como isso se concretizou.

A primeira aplicação do UNIX comercial foi a produção de documentos. Os principais programas são editores de textos, linguagens de marcação (**nroff/troff**) e software de impressão.

As impressoras eram primitivas, assim como os spoolers. Isso também era verdadeiro para sistemas não-UNIX, embora os sistemas não-UNIX fossem proprietários: os sistemas IBM sabiam operar impressoras IBM, computadores Apple sabiam operar impressoras Apple e assim por diante. Muitas vezes supunha-se (corretamente) que o computador em que você trabalha estivesse conectado diretamente à impressora. A configuração de impressora consistia em responder a perguntas como "serial ou paralela?"

Quando as impressoras de rede tornaram-se disponíveis, os problemas se multiplicaram. Os primeiros sistemas de impressão em rede eram idiossincráticos e utilizavam vários protocolos para comunicação entre a impressora e o spooler, comunicação entre o cliente e o spooler e a negociação de tráfego de rede.

À medida que o mundo tornou-se mais complexo, houve várias tentativas de criar padrões unificados, mas nenhum teve aceitação universal. Os protocolos em uso envelheceram e tornaram-se instáveis. Novos recursos de impressora, como duplexing, também estimularam vários tipos de casos especiais.

O Printer Working Group do IETF, firme e audaciosamente, criou o Internet Printing Protocol (IPP), que é construído sobre o HTTP. Essa escolha não só estruturou as interações em termos das requisições simples de GET e POST, como também permitiu que a impressão tirasse proveito das tecnologias amplamente utilizadas para autenticação, controle de acesso e criptografia padrão.

Michael Sweet e Andrew Senft, da Easy Software Products (ESP), introduziram o IPP no UNIX na forma da implementação CUPS. Hoje, o CUPS é a implementação mais completa de IPP no planeta.

Embora a ESP tenha seu próprio nicho de mercado e de produtos, o CUPS é um projeto de código-fonte aberto e é livremente redistribuível. A maioria dos sistemas Linux e UNIX atuais utiliza o CUPS como seu sistema de impressão padrão.

Um servidor CUPS *é* um servidor Web, embora um que se comunica na porta 631 em vez de na porta 80. Você pode verificar isso contatando seu servidor CUPS local por meio de um navegador Web (localhost:631). Você verá que o servidor CUPS apresenta uma interface gráfica completa com todas as suas funcionalidades. (Você também pode utilizar SSL na porta 433 para comunicação segura com as impressoras.) O CUPS fala IPP com os navegadores Web, impressoras, ferramentas gráficas e CLI (de linha de comando) e outros servidores CUPS.

Arquivos PPD

Ao invocar **kprinter** para imprimir **book.ps** na impressora colorida Pollux, **kprinter** pode voltar e perguntar em que tamanho de papel você quer imprimir. Mas espere — como o CUPS sabe como instruir seu cliente, **kprinter**, que a Pollux pode imprimir em uma folha A4? Como o CUPS sabe que Pollux pode tratar PostScript e o que o CUPS deve fazer se ela não puder? Onde o CUPS localiza as informações de que Pollux é uma impressora colorida?

Todas essas informações são mantidas em um arquivo PostScript Printer Description (PPD) que descreve os atributos e as capacidades de uma impressora PostScript. O daemon CUPS lê os PPDs para suas impressoras e passa informações sobre esses arquivos para os clientes e filtros conforme necessário.

Os PPDs foram primeiramente desenvolvidos para o mundo Mac, mas foram rapidamente adotados pelo software Windows. Cada nova impressora vem com um PPD do fornecedor. Drivers de impressoras Mac e Windows utilizam o arquivo PPD para descobrir como enviar trabalhos PostScript à impressora. Por exemplo, não faz sentido pedir a uma impressora preto-e-branco de uma única face vendida nos Estados Unidos para imprimir um documento duplex colorido em um formato de página B4 europeu.

Os sistemas de impressão UNIX e Linux mais antigos não utilizam PPDs. Os usuários tiveram de aprender a manipular o PostScript ou a viver com o que eles tinham como saída padrão. Por comparação, o CUPS foi construído a partir do zero para tirar proveito dessa rica fonte de informações. Na realidade, o CUPS depende dos PPDs.

Localizar arquivos PPD pode exigir um pouco de trabalho de detetive. Se um PPD estiver na sua máquina, ele provavelmente estará em **/etc/cups/ppd** ou **/usr/share/cups/model**. O comando **locate .ppd** ajuda a monitorar esses arquivos. Quanto a impressoras de rede, os PPDs provavelmente são armazenados remotamente; clientes CUPS recebem as informações sobre os PPDs do servidor CUPS relevante.

Arquivos PPD são simplesmente arquivos de texto. É informativo examinar um desses arquivos e ver o tipo de informações que ele contém.

Todas as impressoras PostScript têm PPDs distribuídos pelo fornecedor, que você pode obter do disco de instalação ou do site do fornecedor. Os PPDs da biblioteca distribuída com o CUPS são mantidos em **/usr/share/cups/model**; o CPUS copia PPDs que estão atualmente em uso no **/etc/cups/ppd**.

O CUPS também utiliza PPDs para descrever impressoras que não têm um interpretador PostScript. Um campo extra resolve o problema. Veja:

```
$ grep cupsFilter /usr/share/cups/model/pxlmono.ppd
*cupsFilter: "application/vnd.cups-postscript 0 pstopxl"
```

Você pode usar **diff** em alguns PPDs intimamente relacionados (tente **pxlmono.ppd** e **pxlcolor.ppd**) para ver exatamente como dois tipos de impressora diferem.

Se o fornecedor da sua impressora não fornecer um arquivo PPD — talvez porque a impressora não tem um interpretador PostScript e o fornecedor não se preocupa com nada além do Windows — acesse linuxprinting.org e procure no banco de dados Foomatic por informações adicionais. Sua impressora também pode ser suportada pelo projeto Gutenprint (gutenprint.sourceforge.net, primeiramente conhecido como Gimp-Print). Se puder escolher PPDs dessas fontes e seus usuários quiserem a melhor qualidade possível, tente cada opção e veja qual saída parece melhor.

Se um arquivo PPD não puder ser localizado, então:

- Você deveria ter consultado linuxprinting.org antes de comprar a impressora
- Também pode haver um arquivo PPD genérico que permita imprimir algo, mesmo que ele não tire proveito de todos os recursos da impressora
- Se aprimorar um arquivo PPD genérico para que ele funcione melhor com sua impressora, você deverá disponibilizar seu novo PPD no banco de dados Foomatic

Filtros

Em vez de utilizar uma ferramenta de impressão especializada para cada impressora, o CUPS utiliza uma cadeia de filtros para converter um arquivo em algo que a impressora entende.

O esquema de filtros do CUPS é elegante. Quando fornece um arquivo para o CUPS imprimir, ele descobre o tipo MIME do arquivo, os tipos MIME entendidos pela impressora e os filtros de que ele precisa para converter o primeiro tipo no último.

O CUPS utiliza regras em **/etc/cups/ mime.types** para descobrir o tipo de dados de entrada. Por exemplo, a regra

```
application/pdf   pdf string (0,%PDF)
```

significa: "se o arquivo tiver uma extensão **.pdf** ou iniciar com a string %PDF, seu tipo MIME será então aplicativo/pdf".

O CUPS descobre como converter um tipo de dados em um outro tipo pesquisando as regras no arquivo **/etc/cups/mime.convs**. Por exemplo,

```
application/pdf   application/postscript 33 pdftops
```

significa: "Para converter um arquivo aplicativo/pdf em um arquivo aplicativo/postscript, execute o filtro **pdftops**". O número 33 é o custo da conversão.

Se precisar escrever seus próprios filtros (improvável), não modifique os arquivos distribuídos. Crie um conjunto adicional de arquivos com um nome de base próprio e o coloque em **/etc/cups** em que o CUPS pode localizar os arquivos. O CUPS lê todos os arquivos com os sufixos **.types** e **.convs**, não só **mime.types** e **mime.convs**.

Os últimos componentes na seqüência CUPS são os filtros que conversam diretamente com a impressora. No PPD de uma impressora não-PostScript você pode ver linhas como:

```
*cupsFilter: "application/vnd.cups-postscript 0 foomatic-rip"
```

ou até mesmo

```
*cupsFilter: "application/vnd.cups-postscript foomatic-rip"
```

A string entre aspas tem o mesmo formato de uma linha em **mime.convs**, mas há apenas um tipo MIME em vez de dois. Essa linha informa que o filtro **foomatic-rip** converte os dados do tipo application/vnd.cups-postscript no formato de dados nativos da impressora. O custo é zero (ou omitido) porque só existe uma maneira de realizar esse passo, então por que fingir que há um custo? (Arquivos PPDs do Gutenprint para impressoras não-PostScript são ligeiramente diferentes.)

Dado um documento e uma impressora alvo, o CUPS utiliza os arquivos **types** para descobrir o tipo de documento. Ele depois consulta o PPD para descobrir que tipo de dados a impressora exige. Em seguida, ele utiliza os arquivos **.convs** para deduzir todas as cadeias de filtro que poderiam converter uma em outra e o custo de cada cadeia. Por fim, ele seleciona a cadeia com o menor custo e passa o documento por esses filtros.

O filtro final na cadeia passa o formato imprimível para um back end, que transmite esses dados à impressora utilizando o protocolo que a impressora entende seja qual for ele.

Para localizar os filtros disponíveis no seu sistema, experimente **locate pstops**. (**pstops** é um filtro popular que manipula trabalhos PostScript de várias maneiras, por exemplo, adicionar comandos PostScript para configurar o número de cópias. Os outros filtros não estarão longe.)

Você pode solicitar ao CUPS uma lista dos back ends disponíveis executando **lpinfo -v**. Se seu sistema não tiver um back-end para o protocolo de rede de que você precisa, ele pode estar disponível na Web ou do fornecedor.

23.4 ADMINISTRAÇÃO DO SERVIDOR CUPS

cupsd inicia em tempo de inicialização e executa continuamente. Todas as nossas distribuições Linux de exemplo são configuradas dessa maneira por padrão.

O arquivo de configuração do CUPS é chamado **cupsd.conf**; ele normalmente está localizado em **/etc/cups**. O formato de arquivo é semelhante àquele do arquivo de configuração Apache. Se estiver familiarizado com um desses arquivos, você estará familiarizado com o outro.

Depois de fazer as alterações no arquivo de configuração, execute **/etc/init.d/cups restart** para reiniciar o daemon a fim de que suas alterações entrem em vigor. (Distribuições Debian e Ubuntu utilizam **/etc/init.d/cupsys restart**.)

O arquivo padrão de configuração está bem comentado. Os comentários e a página man do **cupsd.conf** são suficientemente bons, assim não repetiremos as mesmas informações aqui.

Você pode editar o arquivo de configuração CUPS manualmente ou, se tiver o ambiente desktop KDE instalado, você pode configura o sistema por meio do KDE Print Manager, acessível a partir do centro de controle KDE. O *KDEPrint Handbook* documenta o processo em detalhes e é uma boa referência a variáveis CUPS, seus significados e seus valores padrão.

Você pode executar diretamente a parte do gerenciador de impressão KDE específica ao CUPS com o comando **cupsdconf**. Esse comando é incluído nos pacotes **kdelibs** da maioria dos sistemas; não é necessário instalar todo o KDE para usá-lo.

Não temos experiência de produção com a interface gráfica do KDE, mas, no nosso teste, ela não conseguiu entender certas opções localizadas nos arquivos **cupsd.conf** padrão em todos os nossos sistemas de referência. No SUSE ela se recusou a executar, aparentemente porque a linha

```
AuthType BasicDigest
```

em **cupsd.conf** fez com que ela procurasse o arquivo inexistente **/etc/passwd.md5**. (Outros sistemas utilizam `AuthType Basic` como um padrão.) O seu caso pode variar.

Configuração do servidor para impressão em rede

Para que o CUPS aceite trabalhos de impressão a partir da rede, faça duas modificações no arquivo **cupsd.conf**. Primeiro, mude

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>
```

para

```
<Location />
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
Allow From endereçoDeRede
</Location>
```

Substitua *endereçoDeRede* pelo endereço IP da rede a partir da qual aceitar trabalhos de impressão (por exemplo, 192.168.0.0). Procure então a palavra-chave `BrowseAddress` e a configure como o endereço de broadcast nessa rede e também na porta CUPS:

```
BrowseAddress 192.168.0.255:631
```

Esses dois passos instruem o servidor a aceitar requisições de qualquer máquina na rede e a transmitir o que ele sabe sobre as impressoras que ele está servindo para cada daemon CUPS na rede.

É isso aí! Depois de reiniciar o daemon CUPS, ele retorna como um servidor.

Autoconfiguração de impressora

Na verdade, você pode utilizar o CUPS sem uma impressora (por exemplo, para converter arquivos no formato PDF ou no formato de fax), mas seu papel típico é gerenciar impressoras reais. Nesta seção revisaremos as maneiras como você pode lidar com impressoras.

Em alguns casos, adicionar uma impressora é simples. O CUPS tenta autodetectar as impressoras USB quando elas estão conectadas e descobrir o que fazer com elas.

Em geral, os fabricantes de impressoras tem um software de instalação que faz a maior parte do trabalho de configuração para você no Windows e até mesmo no Mac OS X (que também utiliza o CUPS). Entretanto, você não pode supor que os fornecedores tratarão a instalação para você no Linux.

Mesmo se tiver de fazer esse trabalho manualmente, adicionar uma impressora na maioria das vezes só consiste em nada mais do que ligar o hardware, conectar a interface Web do CUPS a

localhost:631/admin

e responder a algumas perguntas. O KDE e o GNOME vêm com seus próprios componentes de configuração de impressora, o que você pode preferir à interface CUPS.

Se uma outra pessoa adicionar uma impressora e um ou mais servidores CUPS em execução na rede souberem disso, seu servidor CUPS será notificado da existência dessa impressora. Você não precisa adicionar explicitamente a impressora ao estoque local ou copiar os PPDs para sua máquina. Tudo é feito com espelhos.

Configuração de impressora de rede

Impressoras de rede precisam de alguma configuração própria para poder utilizar a rede TCP/IP. Em particular, elas precisam conhecer seus endereços IP e máscaras de rede. Essas informações normalmente são transmitidas a elas em uma de duas maneiras.

As impressoras mais modernas podem obter essas informações na rede a partir de um servidor BOOTP ou DHCP. Esse método funciona bem em ambientes com várias impressoras homogêneas. Consulte a página 217 para informações adicionais sobre DHCP.

Alternativamente, você pode configurar um endereço IP estático a partir do console da impressora, que, em geral, consiste em um conjunto de botões no painel frontal e display de uma linha da impressora. Mexa nos menus até descobrir onde configurar o endereço IP. (Se houver uma opção de menu para imprimir os menus, utilize-a e salve a versão impressa.)

Algumas impressoras fornecem acesso a um console virtual por meio de uma porta serial. É uma idéia interessante, mas talvez a quantidade total de trabalho seja semelhante ao uso dos botões no painel frontal. Os princípios são os mesmos.

Se tudo o mais falhar, muitas impressoras vêm com manuais.

Depois de configuradas, as impressoras de rede normalmente têm um "console Web" acessível a partir de um navegador. Entretanto, as impressoras precisam ter um endereço IP antes de você acessá-las dessa maneira, portanto, há uma questão de inicialização.

Depois de sua impressora estar na rede e de você enviar um ping para ela, certifique-se de protegê-la conforme descrito na seção *Proteja suas impressoras,* na página 541.

Exemplos de configuração de impressora

Como exemplos, vamos adicionar a impressora paralela groucho e a impressora de rede fezmo a partir da linha de comando.

```
# lpadmin -p groucho -E -v parallel:/dev/lp0 -m pxlcolor.ppd
# lpadmin -p fezmo -E -v socket://192.168.0.12 -m laserjet.ppd
```

Como você pode ver, groucho é anexada à porta **/dev/lp0**, e fezmo está no endereço IP 192.168.0.12. Especificamos cada dispositivo na forma de um indicador universal de recurso (*universal resource indicator* – URI) e escolhemos um PPD a partir daqueles em **/usr/share/cups/model**.

Contanto que o **cupsd** local tenha sido configurado como um servidor de rede, ele torna as novas impressoras imediatamente disponíveis para outros clientes na rede.

Em vez da interface de linha de comandos, se você preferir, use as ferramentas de configuração baseadas na Web apresentadas pelo servidor CUPS. Isso é verdadeiro para todas as tarefas administrativas nesta seção.

O CUPS aceita uma ampla variedade de URIs para impressoras. Eis mais alguns exemplos:

```
ipp://zoe.canary.com/ipp
lpd://riley.canary.com/ps
serial://dev/ttyS0?baud=9600+parity=even+bits=7
socket://gillian.canary.com:9100
usb://XEROX/Phaser%206120?serial=YGG210547
```

Alguns URIs aceitam opções (por exemplo, serial) e outros não. **lpinfo -v** lista os dispositivos que seu sistema pode ver e os tipos URIs que o CUPS entende.

Configuração da classe de impressora

Uma "classe" é um conjunto de impressoras que compartilha uma fila. Os trabalhos na fila podem ser impressos na impressora que estiver disponível primeiro. O comando a seguir cria a classe haemer e inclui nela três impressoras: riley, gilly e zoe.

```
# lpadmin -p riley -c haemer
# lpadmin -p gilly -c haemer
# lpadmin -p zoe -c haemer
```

Observe que não há nenhum passo explícito para criar a classe; a classe existe contanto que impressoras sejam atribuídas a ela. De fato, o CUPS é ainda mais inteligente do que isso: se atribuirmos o mesmo nome a múltiplas impressoras em uma rede, o CUPS irá tratá-las como uma classe implícita e dividirá a carga dos trabalhos entre elas automaticamente.

Desativação de serviço

Se quiser remover uma impressora ou uma classe, isso é facilmente feito com **lpadmin -x**.

```
# lpadmin -x fezmo
# lpadmin -x haemer
```

Mas e se você só quiser desativar temporariamente uma impressora para um serviço em vez de removê-la? Você pode bloquear a fila de impressão em uma das extremidades. Se desativar a parte final (lado da saída ou o lado da impressora) da fila, os usuários ainda poderão enviar trabalhos, mas esses trabalhos nunca serão impressos. Se desativar a parte inicial (entrada) da fila, os trabalhos que já estão na fila serão impressos, mas a fila rejeitará tentativas de enviar trabalhos adicionais.

Os comandos **disable** e **enable** controlam o lado da saída da fila e os comandos **reject** e **accept** controlam o lado do envio. Por exemplo:

```
# disable groucho
# reject corbet
```

Qual utilizar? É uma idéia ruim aceitar trabalhos de impressão que não serão impressos em um futuro previsível, portanto utilize **reject** para um tempo de inatividade estendido. Para breves interrupções que devem ser invisíveis aos usuários (por exemplo, eliminar um papel enroscado [paper jam]), utilize **disable**.

Ocasionalmente, administradores recorrem à mnemônica para ajudá-los a lembrar quais comandos controlam o final da fila. Considere: se o CUPS "rejeitar" um trabalho, isso significa que você não poderá "injetá-lo". Uma outra maneira de não se esquecer dos comandos corretos é lembrar que aceitar e rejeitar são coisas que você pode fazer para imprimir trabalhos, e desativar e ativar são coisas que você pode fazer para as impressoras. Não faz sentido "aceitar" uma impressora ou uma fila de impressão.

Uma recomendação: além de ser um comando CUPS, **enable** também é um comando **bash** predefinido. **bash** supõe que você tem em mente o próprio **enable** dele a menos que você especifique o nome de caminho completo desse comando, **/usr/bin/enable**. Como acontece, a versão **bash** do **enable** ativa e desativa os componentes predefinidos do **bash**, portanto, você pode utilizá-lo para desativá-lo:[2]

```
$ enable -n enable
```

O próprio CUPS às vezes desativa temporariamente uma impressora com a qual está tendo problemas (por exemplo, se alguém removeu um cabo). Após corrigir esse problema, lembre-se de reativar a fila. Se você se esquecer, o **lpstat** o informará. (Para uma discussão mais completa sobre essa questão e uma abordagem alternativa, acesse www.linuxprinting.org/beh.html).

Outras tarefas de configuração

As impressoras de hoje em dia são intensamente configuráveis e o CUPS permite ajustar vários recursos por meio da sua interface Web e por meio dos comandos **lpadmin** e **lpoptions**. Como regra geral, **lpadmin** é para tarefas globais por todo o sistema e **lpoptions**, para tarefas por usuário individual.

lpadmin permite restringir acesso de maneira mais refinada do que **disable** e **reject**. Por exemplo, você pode configurar cotas de impressão e especificar quais usuários podem imprimir em determinada impressora.

Tamanhos de papel

Nos Estados Unidos e no Canadá, o tamanho mais comum é chamado carta e tem 8,5 × 11 polegadas (21,59 × 27,94 cm). Algumas distribuições Linux (por exemplo, Knoppix e SUSE) são produzidas na Europa, onde as pessoas nem mesmo sabem o que são polegadas, ou na Inglaterra, onde elas sabem, porém não as utilizam para medir papel. Nesses locais, e no Japão, o tipo comum de papel é chamado A4 e todas as impressoras vêm com bandejas A4. Portanto, os utilitários de impressão de algumas distribuições produzem imagens de página A4 por padrão.

O papel A4 faz sentido porque não é racional — isto é, matematicamente. A relação entre o comprimento e a largura do papel A4 é $\sqrt{2}$. Se dividir horizontalmente um pedaço de papel A4 pela metade, você obterá duas metades do papel com a mesma relação entre comprimento e largura. Esse tamanho de papel é chamado A5. Corte o A5 pela metade e você obterá duas folhas do A6. Na outra direção, o A3 é duas vezes a área do A4, mas tem o mesmo formato, e assim por diante.

Em outras palavras, você pode produzir papel A0, que tem uma área de 1 metro quadrado, e utilizar um cortador de papel para criar as outras dimensões que você precisar. O único tamanho de papel comum nos Estados Unidos com o qual você pode fazer esse tipo de jogo é o formato ledger (11 × 17 polegadas, também conhecido como tablóide), que você pode cortar pela metade para obter duas folhas de papel carta.

Há também uma série ISO B e uma série C que preservam o formato $1:\sqrt{2}$, mas têm áreas básicas diferentes. B0 tem 1 m de altura e o papel C0 tem uma área de 2^{15} m^2. Engenheiros perceberão que as faces do papel B*n* são a média geométrica das faces A*n-1* e A*n*, enquanto as faces do papel C*n* são a média geométrica do A*n* e do B*n*.

---

2. Para pontos extras, descubra como reativar o comando **enable** predefinido do **bash** agora que você bloqueou acesso a ele. **enable enable** não funcionará!

O que tudo isso significa? B*n* tem a mesma aparência de A*n*, porém é maior, e C*n* é o intermediário entre os dois. Um relatório em uma folha A4 cabe perfeitamente em uma pasta de papel manilha C4. Dobrar uma folha A4 pela metade a torna um A5 e a faz caber em um envelope C5. Dobre-a novamente e ela caberá perfeitamente em um envelope C6.

Para confundir as coisas um pouco, o Japão tem sua própria série B que é ligeiramente diferente. Embora tenha o mesmo formato dos papéis ISO, o tamanho do papel B4 japonês é a média aritmética do A3 e do A4 que o torna um pouco maior que o papel ISO B4. Não há uma série C japonesa.

Assim como o sistema ISO facilita copiar duas páginas de um formato B5 em um único anúncio no formato B4, ele também simplifica a impressão de todos os tipos *n* e acima (imprimir várias imagens no tamanho reduzido de uma página na mesma página). Copiadoras européias freqüentemente têm botões que reduzem ou expandem por um fator de $\sqrt{2}$.

Se seu sistema tiver o comando **paperconf** instalado, você poderá utilizar esse comando para imprimir as dimensões de vários papéis identificados em polegadas, centímetros ou pontos da impressora (72 avos de uma polegada). Para os norte-americanos, a Tabela 23.1 lista algumas utilizações típicas para tamanhos comuns a fim de dar uma idéia das suas escalas.

Tabela 23.1 Utilizações comuns para tamanhos de papel ISO

Tamanhos	Usos comuns
A0, A1	Cartazes
A3, B4	Jornais
A4	"Folhas de papel" genéricas
A5	Blocos de notas/rascunhos (mais ou menos 5 x 8 polegadas)
B5, B6	Livros, cartões-postais, papel higiênico alemão
A7	Fichas de arquivo "3×5"
B7	Passaportes (até mesmo os passaportes norte-americanos são B7)
A8	Cartões de visita
B8	Cartas de baralho

Infelizmente, o papel A4 é um pouco mais fino e mais longo (8,3 x 11,7 polegadas) do que papel carta norte-americano. Imprimir um documento A4 em papel carta geralmente corta partes vitais como cabeçalhos, rodapés e números de página. Por outro lado, se você estiver na Europa ou no Japão e tentar imprimir páginas no formato norte-americano em papel A4, talvez as laterais dos seus documentos sejam cortadas (embora o problema seja menos sério).

Pacotes de softwares individuais podem ter seus próprios padrões no que diz respeito ao tamanho do papel. Por exemplo, GNU **enscript** é mantido na Finlândia por Markku Rossi e assume o padrão do papel A4. Se você for norte-americano e sua distribuição não tiver compilado **enscript** com um padrão diferente, uma opção é pegar o código-fonte e reconfigurá-lo. Em geral, porém, é mais fácil configurar o tipo de papel na linha de comando ou em um arquivo de configuração. Se os seus documentos forem impressos com as extremidades ou as laterais cortadas, uma possível explicação seria os conflitos de tamanho de papel.

Talvez você também consiga ajustar o tamanho do papel padrão para muitas tarefas de impressão com o comando **paperconfig**, com a variável de ambiente PAPERSIZE ou com o conteúdo do arquivo **/etc/papersize**. (Nota: **paperconfig** != **paperconf**).

Comandos de compatibilidade

Antigamente, havia dois sistemas de impressão concorrentes: um encontrado nos sistemas BSD UNIX e outro nos sistemas System V UNIX. Os dois sistemas mantinham filas de impressão relativamente simples e forneciam comandos para criar, excluir, iniciar, parar e pausar as filas e para enfileirar ou desenfileirar trabalhos individuais.

Você poderia perguntar: "Por que dois sistemas e quais eram as diferenças significativas entre eles?". Levante-se no meio de uma reunião em um grupo de usuários Linux e grite: "Qualquer um que utiliza **vi** é idiota!" — e então faça essa pergunta novamente.

Sabiamente, o CUPS fornece comandos de compatibilidade que substituem os dois sistemas. Parte da motivação é facilitar para os veteranos acostumados com antigos sistemas, mas compatibilidade com os softwares existentes também é um objetivo importante.

Mais precisamente, esses comandos nem sempre fazem tudo o que os originais faziam e alguns comandos menos utilizados e específicos do fornecedor ainda não foram implementados. Contudo, vários scripts que utilizam esses comandos funcionam bem com o CUPS. Pense no que está faltando como uma oportunidade: se quiser contribuir para a paz mundial e para a "eficiência de Pareto", ainda resta código a ser escrito.

A Tabela 23.2 lista os comandos CLI que acompanham o CUPS e os classifica de acordo com suas origens.

Tabela 23.2 Utilitários de linha de comando CUPS e suas origens

	Comando	Função
CUPS	**lpinfo**	Mostra os dispositivos ou drivers disponíveis
	lpoptions	Exibe ou configura as opções e os padrões de impressora
	lppasswd	Adiciona, altera ou exclui senhas
	cupsdconf[a]	É uma ferramenta de configuração CUPS
	cups-config[a]	Imprime informações da API, compilador, diretório e de link do CUPS
System V	**lp:**	Imprime arquivos
	cancel	Cancela trabalhos
	accept, reject	Aceita ou rejeita envios de fila
	disable, enable	Pára ou inicia impressoras e classes
	lpstat	Imprime informações do status CUPS
	lpadmin	Configura impressoras e classes CUPS
	lpmove	Move um trabalho para um novo destino
BSD	**lpr**	Imprime arquivos
	lprm	Cancela trabalhos de impressão
	lpq	Exibe status de fila de impressora
	lpc	É um programa de controle de impressora geral

a. Não confunda esses nomes semelhantes. **cupsdconf** é uma ferramenta gráfica no KDEPrint e **cups-config** é uma ferramenta CLI incluída no CUPS.

Software comum para impressão

Há mais na impressão além de spool e impressão de trabalhos. Mesmo em um sistema Ubuntu, o comando

```
$ man -k . | egrep -i 'ghostscript|cups|print(er|ing| *(job|queue|filter))'
```

lista mais de 88 páginas man relacionadas à impressão — e isso é apenas uma pesquisa rápida e grosseira. (Quanto aos comandos relacionados à impressão, considere o fato de que o comando **print** não tem nada a ver com impressão.) Vale a pena conhecer vários desses comandos e ferramentas.

pr é uma das mais antigas ferramentas de impressão. Ela reformata arquivos de texto para a página impressa. Ele quebra sua entrada em páginas de 66 linhas, adiciona cabeçalhos e rodapés e pode dobrar o espaço do texto. Ele é perfeito para manipulações menores dos arquivos de texto a caminho da impressora.

O comando **enscript** da Adobe realiza conversões semelhantes com muito mais penduricalhos; sua saída também é PostScript. O GNU **enscript** é uma versão de código-fonte aberto desse comando que é retrocompatível com o da Adobe; mas, o GNU **enscript** oferece vários recursos novos, incluindo destaque sensível à linguagem, suporte para vários tamanhos de papel, download de fontes e cabeçalhos definidos pelo usuário.

Uma das principais alegações à fama do **enscript** era sua implementação da impressão 2-up. Se você ainda utiliza **enscript** por causa desse recurso, tente a opção **-o number-up=2** do CUPS para **lpr**.

No ponto mais alto do espectro da complexidade está o Ghostscript, originalmente escrito por L. Peter Deutsch para que ele pudesse imprimir documentos PostScript em impressoras PCL baratas. Hoje, o Ghostscript interpreta tanto PostScript como PDF. O CUPS o utiliza como um filtro, mas o Ghostscript também cria imagens de página na tela, por conta própria ou com ajuda de front ends como **gv**, GNOME Ghostview (**ggv**) ou KGhostView do KDE.

Todas as distribuições Linux vêm com uma versão gratuita do Ghostscript; para informações adicionais, consulte www.ghostscript.com. Uma versão comercial do Ghostscript com suporte está disponível na Artifex Software.

Documentação do CUPS

Não há falta de documentação do CUPS, mas às vezes você tem de caçá-la. Páginas man, como aquelas para **lpr**, podem ser incompletas e superficiais. Se você não localizar algo em uma página man, não desista; faça uma pesquisa no Google.

A instalação CUPS vem com muitos manuais nos formatos PDF e HTML. Um lugar para pesquisar esses manuais é conectar-se a um servidor CUPS e clicar no link para ajuda on-line. Infelizmente, isso não ajuda se seu problema for conectar-se ao servidor CUPS.

A mesma documentação pode ser encontrada em www.cups.org. Ela também deve estar sob **/usr/share/doc/cups**. Se sua distribuição não tiver instalou aí, experimente

```
$ locate doc | grep cups
```

Uma outra opção é consultar o gerenciador de pacotes da sua distribuição.

23.5 DICAS PARA SOLUÇÃO DE PROBLEMAS

Sempre se lembre de reiniciar **cupsd** depois de alterar o arquivo de configuração. Sua melhor aposta para reiniciar é executar **/etc/init.d/cups restart** (**/etc/init.d/cupsys restart** no Debian e no Ubuntu). Você também pode reiniciar o daemon por meio do aplicativo KDE Print Manager. Teoricamente, você também pode enviar ao **cupsd** um sinal HUP, mas isso parece simplesmente eliminar o daemon nos sistemas SUSE.

Registro em log do CUPS

CUPS mantém três logs: um log de página, um log de acesso e um log de erro. O log de página é uma lista de páginas impressas. Os outros dois são parecidos com o log de acesso e o log de erro do Apache. Não surpreende, uma vez que o servidor CUPS é um servidor Web.

O arquivo **cupsd.conf** especifica o nível do registro em log e as localizações dos arquivos de log. Em geral, todos são mantidos abaixo de **/var/log**.

Eis um trecho de um arquivo de log que corresponde a um único trabalho de impressão:

```
I [26/Jul/2006:18:59:08 -0600] Adding start banner page "none" to job 24.
I [26/Jul/2006:18:59:08 -0600] Adding end banner page "none" to job 24.
I [26/Jul/2006:18:59:08 -0600] Job 24 queued on 'Phaser_6120' by 'jsh'.
I [26/Jul/2006:18:59:08 -0600] Started filter /usr/libexec/cups/filter/pstops (PID 19985) for job
   24.
I [26/Jul/2006:18:59:08 -0600] Started backend /usr/libexec/cups/backend/usb (PID 19986) for job
   24.
```

Problemas com a impressão direta

Para verificar a conexão física de uma impressora local, você pode executar diretamente o back-end da impressora. Por exemplo, eis o que obtivemos ao executar o back-end para uma impressora conectada via USB:

```
$ /usr/lib/cups/backend/usb
direct usb "Unknown" "USB Printer (usb)"
direct usb://XEROX/Phaser%206120?serial=YGG210547 "XEROX Phaser 6120" "Phaser 6120"
```

Quando o cabo USB desconecta-se acidentalmente (ou quebra), a linha para essa impressora é removida da saída do back-end:

```
$ /usr/lib/cups/backend/usb
direct usb "Unknown" "USB Printer (usb)"
```

Problemas de impressão em rede

Antes de começar a monitorar um problema de impressão em rede, certifique-se de que pode imprimir a partir da máquina que efetivamente hospeda a impressora. Seu "problema de impressão em rede" pode ser apenas um "problema de impressão". Também se certifique de que a rede está ativa.

Em seguida, tente conectar ao **cupsd** hospedeiro com um navegador Web (*hostname*:631) ou por meio do comando **telnet** (**telnet** *hostname* **631**).

Se tiver problemas para depurar uma conexão da impressora de rede, tenha em mente que há uma fila para o trabalho em alguma máquina, uma maneira de decidir aonde enviar o trabalho e um método de enviar o trabalho à máquina que hospeda a fila de impressão. Na máquina que faz a impressão, deve haver um lugar para colocar o trabalho em fila, permissões suficientes para permitir que ele seja impresso e uma maneira de gerar saída para o dispositivo.

Para monitorar esses problemas, talvez você precise examinar vários lugares:

- Arquivos de log do sistema na máquina remetente, em busca de problemas relacionados à permissão e à determinação de nomes
- Arquivos de log do sistema no servidor de impressão, em busca de problemas de permissão

- Arquivos de log do CUPS na máquina de envio, em busca de filtros ausentes, impressoras desconhecidas, diretórios ausentes etc.
- Arquivos de log do CUPS na máquina do servidor de impressão, em busca de mensagens sobre nomes de dispositivos ruins, formatos incorretos etc.

As localizações dos arquivos de log do sistema são especificadas em **/etc/syslog.conf**. As localizações dos arquivos de log do CUPS são especificadas em **/etc/cups/cupsd.conf**.

Problemas específicos da distribuição

CUPS ainda está em evolução e correções de bug são distribuídas com freqüência. Alguns problemas são piores do que outros e outros têm implicações sobre a segurança. Em algumas versões mais antigas do Red Hat, o CUPS funciona muito mal. A solução correta para esses sistemas é uma atualização de sistema operacional. Mas se você não puder instalar uma versão mais recente do Red Hat ou do Fedora, tente obter a versão atual do CUPS.

A Easy Software Products vende uma versão comercial do CUPS chamada ESP PrintPro que suporta um espectro muito mais amplo de impressoras do que a versão gratuita. Se precisar suportar uma impressora incomum e não puder localizar os drivers necessários na Web, a ESP talvez já esteja executando essa impressora. Essa empresa também vende suporte. Confira-os em www.easysw.com.

23.6 RECURSOS AVANÇADOS DE IMPRESSORA

Lidar com impressoras pode trazer problemas e frustrações. Eis algumas diretrizes gerais para ajudar a limitar estes. Quando tudo o mais falhar, sinta-se feliz pelo fato de você não mais utilizar uma impressora matricial conectada via uma porta serial RS-232. A menos que, naturalmente, você ainda use.

Seleção de impressora

Antes de você comprar uma impressora ou aceitar uma impressora "gratuita" que outra pessoa está se desfazendo, visite o banco de dados Foomatic em linuxprinting.org e verifique como essa impressora é suportada sob o Linux. Esse banco de dados classifica as impressoras em quatro categorias que variam de Paperweight a Perfectly; escolha Perfectly.

O CUPS gosta de impressoras PostScript. A configuração dessas impressoras em geral é fácil.

Impressoras não-PostScript também são suportadas, mas não tão bem. Para imprimir nestas, você precisa de software que converta os trabalhos de impressão em PDL ou no formato de dados preferido da impressora. Provavelmente esse software estará disponível na distribuição CUPS ou em um dos outros locais mencionados neste capítulo.

Impressoras GDI

O Windows ainda tem uma vantagem em algumas áreas, uma das quais é o suporte a impressoras não sofisticadas. As impressoras baratas utilizadas em sistemas Windows são conhecidas coletivamente como impressoras GDI ou WinPrinters. Essas impressoras têm muito pouca inteligência predefinida e não têm interpretadores para nenhum PDL real. Elas esperam que a rasterização seja realizada pelo computador host.

Algumas das informações necessárias para comunicação com as impressoras GDI estão ocultas no código proprietário específico do Windows. Esses segredos atrapalham as tentativas de desenvolvimento do suporte Linux para esses dispositivos, embora a comunidade Linux tenha demonstrado uma capacidade admirável em termos de engenharia reversa. O CUPS suporta muitas WinPrinters.

Uma segunda área de domínio do Windows é seu suporte a impressoras novas em folha. Assim como ocorre com as novas placas de vídeo e de áudio, as impressoras novas são primeiro lançadas com drivers Windows, que suportam completamente os recursos documentados e não-documentados de todos os modelos. O suporte Linux geralmente fica para trás. Se comprar uma sofisticada impressora recém-lançada porque você precisa de recursos avançados, talvez você tenha de conformar em operá-la temporariamente a partir do Windows.

Impressão frente e verso

Um duplexador é um componente de hardware que permite que uma impressora imprima na frente e no verso da página. Algumas impressoras incluem esses duplexadores por padrão e outras os suportam como um suplemento opcional.

Se não tiver acesso a (ou não puder comprar) uma impressora que imprima frente e verso, você pode primeiro imprimir as páginas ímpares e então virar o papel e imprimir as páginas pares. Teste com um documento de duas páginas para descobrir a maneira de virar o papel e então passe as instruções para a impressora.

Uma variedade de softwares de impressão pode ajudar a fazer isso; por exemplo, Ghostview (**gv**) tem ícones que permitem marcar uma série e uma opção para imprimir apenas as páginas marcadas. As versões CUPS do **lp** e **lpr** tratam essa

tarefa com as opções **-o page-set=odd** e **-o page-set=even**. Você pode armazenar essas opções em uma "instância de impressora" se utilizá-las com freqüência; veja a página 528.

Algumas impressoras, especialmente impressoras a laser baratas, não foram projetadas considerando a impressão frente e verso. Os fabricantes dessas impressoras costumam alertar contra danos irreparáveis que podem advir da impressão nos dois lados da página. Na realidade, nunca vimos esse tipo de dano, mas certamente os fabricantes de impressora não forneceriam uma recomendação incorreta. Ou forneceriam?

Outros acessórios de impressora

Além dos duplexadores, muitas impressoras permitem adicionar memória, bandejas extras de papel, discos rígidos e outros acessórios. Essas atualizações podem imprimir trabalhos que, de outro modo, seriam difíceis de imprimir, ou podem fazer com que os trabalhos sejam impressos com mais eficiência. Se tiver problemas para imprimir trabalhos, revise os logs de erro para ver se mais memória de impressão poderia ajudar a resolver o problema. Consulte a seção *Registro em log do CUPS*, na página 537.

Impressoras seriais e paralelas

Se sua impressora estiver ligada diretamente ao seu computador com um cabo, ela está utilizando alguma forma de conexão serial ou paralela.

Embora o padrão paralelo não tenha amadurecido com elegância, ele de fato fornece portas que requerem relativamente poucos ajustes. Se você tiver uma impressora paralela, provavelmente será fácil configurá-la — contanto que seu computador também tenha uma porta paralela.

Uma conexão serial no hardware do Mac poderia ser FireWire, mas conexões seriais no mundo Linux em geral utilizam USB. Verifique o banco de dados de dispositivos USB suportados em www.qbik.ch/ usb/devices ou www.linux-usb.org para ver a situação de seu hardware.

É quase certo que você não tem uma impressora serial RS-232 ultrapassada. Se tiver, talvez exija algumas configurações extras trabalhosas. O software spooler tem de saber os valores apropriados para taxa de transmissão e outras opções seriais para poder se comunicar apropriadamente com a impressora. Você especifica todas essas opções no URI do dispositivo. Consulte *CUPS Software Administrators Manual* on-line para detalhes. Poderia ser mais rápido comprar um tipo diferente de impressora do que descobrir a combinação exata da mágica serial necessária para colocar as coisas em funcionamento.

Impressoras de rede

Muitas impressoras contêm interfaces de rede totalmente maduras que permitem que elas se conectem diretamente numa rede e aceite tarefas por meio de uma ou mais redes ou protocolos de impressão. Os dados podem ser enviados para impressoras conectadas à rede muito mais rapidamente do que para impressoras conectadas a portas paralelas seriais.

23.7 OUTROS CONSELHOS SOBRE IMPRESSORAS

Algumas questões administrativas relacionadas à impressão transcendem os detalhes do Linux e do CUPS. A maioria dessas questões surge porque as impressoras são dispositivos mecânicos temperamentais que têm um custo toda vez que eles são utilizados.

Utilize páginas de rosto (banner) somente se necessário

O CUPS pode imprimir páginas inicial (*header*) e final (*trailer*) para cada trabalho que mostram o título do trabalho e o usuário que o enviou. Essas páginas de rosto são às vezes úteis para separar trabalhos em impressoras utilizadas por muitas pessoas, mas, na maioria dos casos, elas são um desperdício de tempo, toner e papel.

Sugerimos desativar as páginas de rosto globalmente na GUI do CUPS (ou executando **lpadmin**) e então ativá-las para trabalhos individuais que talvez se beneficiem dessas páginas:

```
$ lpr -o job-sheets=confidential gilly.ps
```

Você também pode ativar páginas de rosto para usuários individuais usando **lpoptions**. Outra alternativa a considerar é uma instância de impressora que adiciona páginas de rosto aos trabalhos (veja a seção *Instâncias de impressora* na página 528).

Se necessário, crie páginas de rosto personalizadas copiando uma das existentes de **/usr/share/cups/banners** e a modificando-a. Adicione a nova página às outras sob um novo nome.

Forneça lixeiras

Todos os tipos de papel para computador são recicláveis. Você pode utilizar as caixas de papelão como lixeiras; o papel cabe perfeitamente nelas. Coloque um aviso dizendo que nenhum material estranho (como grampos, clipes e jornais) deve ser jogado ali.

Utilize visualizadores

Os usuários muitas vezes imprimem um documento, encontram um pequeno erro de formatação, o corrigem e, então, reimprimem o trabalho. Esse desperdício de papel e tempo pode ser facilmente evitado com um software que permite que os usuários vejam, na tela, como será a saída impressa.

Só ter os visualizadores não é suficiente; seus usuários têm de saber usá-los. Normalmente eles ficam felizes em aprender. Um emprego dos registros contábeis é a verificação de casos nos quais o mesmo documento vem sendo impresso repetidamente. Isso é às vezes um indicador para um usuário que não conhece visualizadores.

A visualização é incorporada a muitos editores WYSIWYG modernos, navegadores e agregadores de trabalho de impressão. Para outros tipos de documentos, suas opções variam. Ferramentas como o Ghostview (**gv**) visualizam documentos PostScript e PDF aleatórios. Para **roff**, canalize a saída do **groff** no Ghostview; para saída TEX, tente **xdvi**, **kdvi** ou Evince.

Compre impressoras baratas

A tecnologia de hardware de impressão é madura. Você não precisa gastar muito para obter impressão de boa qualidade e mecânica confiável.

Não esbanje em uma impressora de "grupo de trabalho" cara a menos que seja realmente necessário. Se estiver imprimindo apenas texto, uma impressora "pessoal" barata pode produzir uma saída de boa qualidade, ser quase tão rápida e confiável e custar bem menos. Uma impressora de 10 páginas por minuto pode atender cerca de cinco usuários em tempo integral. Sua melhor escolha poderia ser a compra de cinco impressoras de US$ 250 para um grupo de 25 escritores do que uma impressora de US$ 1.250.

Em geral, você não deve comprar uma impressora (ou um disco rígido ou memória) de um fabricante de computadores. As impressoras deles normalmente provêm de outros fabricantes famosos, mas custando o dobro do preço. As impressoras PostScript fabricadas para os mercados PC e Macintosh e vendidas de maneira independente normalmente são os melhores negócios. (Algumas empresas, como a HP, produzem computadores e impressoras. Eles são bons.)

Mesmo se você só utilizar as principais marcas, nenhum fabricante individual é uma aposta universalmente segura. Tivemos excelentes experiências com impressoras a laser da HP. Elas são produtos sólidos e a HP tem sido bem dinâmica no suporte tanto do Linux como do CUPS. Mesmo assim, algumas impressoras da HP foram desastres completos. Procure avaliações na Internet antes de comprar uma.

Aqui, também, custo baixo é uma vantagem: é mais fácil se recuperar de um erro de US$ 250 do que de um de US$ 1.250.

Mantenha cartuchos de toner extras no estoque

Ocasionalmente, os cartuchos do toner das impressoras a laser precisarão ser substituídos. Áreas apagadas ou em branco na página são dicas de que o toner da impressora precisa ser trocado. Compre cartuchos de substituição antes de precisar efetivamente deles. Em uma hora de aperto, remova o cartucho da impressora e o agite suavemente a fim de redistribuir as partículas remanescentes do toner. Freqüentemente é possível imprimir outras cem páginas fazendo isso.

Riscos e manchas provavelmente significam que você deve limpar os cartuchos da sua impressora. Examine a impressora para ver se há um ciclo "limpo". Se não houver, ou se o ciclo de limpeza não ajudar, leia atentamente as instruções de limpeza do fabricante, ou pague por esse serviço.

Os fabricantes de impressora odeiam usar cartuchos reciclados e eles farão o possível para impedir isso. Muitos dispositivos utilizam descartáveis "com chaves" cujas identidades são detectadas — eletrônica ou fisicamente — pela impressora. Mesmo se duas impressoras parecerem idênticas (como a Xerox Phaser 6120 e a Konica-Minolta Magicolor 2450), isso não significa necessariamente que você pode utilizar os mesmos cartuchos em ambas.

Às vezes você pode fazer uma operação para converter os cartuchos de um fornecedor nos cartuchos de uma outra impressora, mas ajuda saber o que você está fazendo. Normalmente, isso só cria uma confusão. Se espalhar o pó do toner, use um aspirador de pó para limpar a maior quantidade de pó possível e lave o restante com água fria. Contrário à crença comum, o toner de uma impressora a laser não é um perigo à saúde ou ao meio ambiente, embora, como ocorre com todo tipo de pó, seja melhor evitar inspirar o pó do toner.

Ao substituir um cartucho, guarde a caixa e a sacola do novo cartucho para a reciclagem do cartucho vazio. Então procure na lista telefônica ou na Web uma empresa que trabalhe com cartuchos para que você possa se livrar dos antigos.

Cartuchos descartáveis com chave estimulam o crescimento de empresas que recarregam os cartuchos antigos por uma fração do preço do novo cartucho. Recicladores de cartuchos normalmente também são empresas de recarga, portanto você pode reciclar seus cartuchos antigos e, ao mesmo tempo, obter substituições.

Opiniões sobre a qualidade e tempo de vida dos cartuchos reciclados variam. Uma empresa de recarga que conhecemos não recarrega cartuchos toner colorido nem vende cartuchos remanufaturados, pois acha que a economia não paga os custos de manutenção para as impressoras que os utilizam.

Preste atenção ao custo por página

Fabricantes de impressoras utilizam o que MBAs chamam "mensuração" a fim de tornar o custo total da escala de um produto a mais linear possível com relação à quantidade de uso que o cliente pode tirar dele. Essa é a razão pela qual toner e tinta são excessivamente caros e hardware de impressora sofisticado às vezes é vendido abaixo do custo de produção.

Quando escrevíamos este livro, um fabricante vendia uma impressora a laser colorida por US$ 299. Um conjunto completo de cartuchos de substituição para essa impressora custa US$ 278. Você pode comprar uma impressora de jato de tinta por menos de US$ 50, mas não levará muito tempo para que seja necessário comprar um conjunto de cartuchos de tinta de substituição que custa mais que a impressora.

Você pode achar isso ultrajante, mas os fabricantes de impressora têm de tirar seus lucros de algo. Cartuchos mais baratos significariam apenas impressoras mais caras. Uma boa regra geral é que as impressoras de jato de tinta são baratas contanto que você não as utilize para imprimir; impressoras a laser têm um custo inicial mais alto, mas os cartuchos são mais baratos e duram mais tempo.

Uma página colorida em uma impressora de jato de tinta pode custar entre 20 e 50 vezes o custo de uma impressão análoga em uma impressora a laser. Ela também requer papel especial e imprime mais lentamente. Cartuchos de jato de tinta esvaziam rapidamente e com freqüência apresentam defeitos ou estragam. A tinta escorre quando molhada — não use uma impressora de jato de tinta para imprimir um livro de receitas para uso na cozinha. Por outro lado, agora é possível obter impressões com qualidade de foto em uma jato de tinta com um visual tão bom quanto revelações fotográficas em um laboratório profissional. Fotos em uma laser colorida? Não tão interessantes.

Todas as impressoras têm partes mecânicas propensas a falhas. Impressoras baratas quebram mais rapidamente.

Em outras palavras, tudo se resume a uma relação custo/benefício. Para uso pessoal em pequenos volumes — imprimir uma ou duas páginas Web por dia ou imprimir alguns rolos de filme por mês — uma impressora jato de tinta de baixo custo e de uso geral é uma excelente escolha.

Na próxima vez que decidir comprar uma impressora, calcule por quanto tempo você quer manter sua impressora, o volume de impressão que você faz e o tipo de impressão que você precisa antes de comprar uma. Avalie quantitativamente o custo de longo prazo por página de cada possível impressora. E pergunte na sua loja local de recarga se ela remanufatura cartuchos para essa impressora e qual é o preço disso.

Considere a contabilidade de impressora

Em instalações de médio a grande porte, considere o uso da contabilidade de impressora mesmo se você não planejar cobrar pelo uso de uma impressora. O custo por trabalho individual não tem importância e você verá quem está utilizando a impressora. Informações demográficas sobre as fontes de trabalhos de impressão são valiosas ao planejar a implantação de novas impressoras.

Vários pacotes de contabilidade de impressora (como **accsnmp** e PyKota) foram desenvolvidos para o CUPS. A ESP tem uma lista pesquisável dos links para esses e outros produtos relacionados ao CUPS em www.easysw.com/~mike/cups/links.php

Proteja suas impressoras

Impressoras de rede em geral suportam gerenciamento remoto. Você pode configurá-las e monitorá-las pela rede por meio do IPP ou SNMP ou a partir de um navegador Web utilizando HTTP. Por meio da interface remota, você pode configurar parâmetros como o endereço IP, o gateway padrão, o servidor de syslog, o nome de comunidade SNMP, opções de protocolo e a senha administrativa da impressora.

Como padrão, a maioria das impressoras administráveis remotamente é desprotegida e deve ter uma senha (ou quem sabe um "nome de comunidade" SNMP) atribuída como parte do processo de instalação. Os manuais de instalação do seu fabricante de impressora devem explicar como fazer isso em uma determinada impressora, mas ferramentas gráficas de administração no CUPS e o KDE Print Manager conseguem cada vez mais ocultar as variações entre os fornecedores. Essa tendência deve continuar.

23.8 IMPRIMINDO NO KDE

Mencionamos rapidamente o KDE várias vezes neste capítulo.[3] Os recursos de impressão do KDE são realmente bem interessantes, porém, merecem uma discussão um pouco mais detalhada. O KDE se empenhou muito para tornar suas ferramentas e interfaces de impressão independentes do sistema de impressão subjacente. Ele foi construído depois que o CUPS tornou-se popular, portanto ele pode tratar todos os recursos do CUPS. Ele, porém, funciona com tudo do LPRng a um programa externo genérico.

O desenvolvimento dos recursos de impressão do GNOME demorou mais que os do KDE, mas os desenvolvedores do GNOME também queriam que os usuários tivessem uma boa experiência de impressão. O desenvolvimento ainda prossegue rapidamente; no momento em que você estiver lendo isso, os recursos de impressão do GNOME talvez compitam com os do KDE. Um dos leitores de um rascunho anterior deste capítulo observou a ironia do fato de o CUPS ter substituído os "padrões de impressão conflituosos que não tinham nenhuma razão de viver, mas se recusavam a morrer", somente para preparar o caminho para a concorrência entre conjuntos de utilitários de impressão desktop.

KDEPrint é a principal estrutura para impressão sob o KDE. KDEPrint fornece ferramentas para adicionar impressoras, administrar trabalhos de impressão, reiniciar servidores de impressão e assim por diante. Sim, o CUPS também permite fazer tudo isso; as ferramentas KDEPrint estão aí por duas razões.

Primeiro, elas têm uma aparência e um comportamento do KDE, o que oferece consistência aos usuários do KDE. Por exemplo, a ferramenta **kghostview** empacota o Ghostview em uma pele KDE mais apropriada. (Você provavelmente observou que mesmo os *nomes* dos utilitários KDE têm aparência e comportamento distintos. Alguém recentemente perguntou se **ksh** era um aplicativo KDE.)

Segundo, KDEPrint é independente de spooler. Se por alguma razão você não executa o CUPS (ou pior, tem de mudar de um lado para o outro entre sistemas de impressão), você ainda pode utilizar o KDEPrint para gerenciar sua impressão. Esteja prevenido de que o CUPS tem maior capacidade do que outros sistemas de impressão, assim se tiver trocá-lo por um sistema de impressão alternativo mais barato, algumas funcionalidades do KDEPrint podem desaparecer.

Por que você deve se preocupar com todas essas interfaces gráficas se o trabalho de impressão é feito no shell? Bem, seus usuários provavelmente não utilizarão a interface de shell, portanto, algum conhecimento sobre a interface KDE pode acabar sendo necessário apenas para dar suporte a esses usuários.

Eis os componentes fundamentais do KDEPrint que você deve conhecer:

- **kprinter**, uma ferramenta gráfica que envia trabalhos de impressão
- O assistente Add Printer, que autodetecta impressoras de rede (JetDirect, IPP e SMB) e algumas impressoras conectadas localmente. O assistente Add Printer também permite adicionar e configurar impressoras que ele não autodetecta.
- O Print Job Viewer, que move e cancela trabalhos de impressão e mostra as informações sobre o status dos trabalhos de impressão
- O *KDEPrint Handbook*, que documenta o sistema. Ele está disponível no KDE Help Center, mas pode ser irritantemente difícil localizá-lo. Uma rota mais fácil é invocar algo como **kprinter** e clicar em Help. Uma outra alternativa é executar **konqueror help:/kdeprint**. Uma outra fonte de documentação do KDEPrint é printing.kde.org
- O Print Manager, que é a principal ferramenta de gerenciamento gráfica para o sistema de impressão. Ele também pode ser um pouco difícil de localizar. Você pode examinar o menu principal da área de trabalho, embora o local na árvore de menus varie entre distribuições. Uma outra opção é executar **kcmshell printmgr** ou **konqueror print:/manager**

O assistente Add Printer e o Print Job Manager estão acessíveis por meio do **kprinter** ou do KDE Print Manager. (Sem mencionar os URLs **print:/manager** e **print:/printers** no Konqueror.)

Informações por usuário individual para o KDEPrint são armazenadas sob **~/.kde**. Os arquivos são legíveis por humanos, mas projetados para que sejam alterados pelo Print Manager. Manipule esses arquivos ao seu próprio risco.

kprinter: imprimindo documentos

kprinter é uma substituição gráfica para o **lpr**. Ele pode ser utilizado da linha de comando de maneiras semelhantes. Você pode até mesmo suprimir a interface gráfica;

```
$ kprinter --nodialog -5 -P lj4600 riley.ps gillian.pdf zoe.prn
```

é equivalente a:

```
$ lpr -5 -P lj4600 riley.ps gillian.pdf zoe.prn
```

---

3. O KDE é um conjunto de bibliotecas e padrões de interface com usuário para interfaces gráficas que executam sob o X Window System, a tecnologia em que todas as interfaces gráficas do Linux estão baseadas. É uma alternativa ao sistema GNOME, que é o padrão da maioria das distribuições. Apesar das aparências, não é realmente necessário escolher entre o KDE e o GNOME. Para uma descrição mais geral sobre o GNOME e o KDE, consulte a página 522.

Seus usuários provavelmente querem uma interface gráfica. Mostre a eles como arrastar arquivos de um gerenciador de arquivos ou área de trabalho para a caixa de diálogo do **kprinter** para então imprimir o lote inteiro. Substitua **lpr** por **kprinter** na caixa de diálogo de impressão dos navegadores deles e os usuários terão uma caixa de diálogo de impressão gráfica. Ensine-os a clicar na caixa de seleção "Keep this dialog open after printing" (Manter essa caixa de diálogo aberta depois de imprimir) e eles nem mesmo terão o atraso de reiniciar o programa toda vez que quiserem imprimir.

Observe o menu "Print system currently in use", evidência da neutralidade do sistema do KDEPrint. Também note que **kprinter** oferece funções de impressão para fax e impressão para PDF mesmo sem uma impressora real. As opções avançadas também merecem ser examinadas; você pode enfileirar uma retomada da sua impressão e especificar que ela seja impressa depois do seu chefe ir embora.

Konqueror e impressão

Muitos navegadores Web reconhecem um conjunto de URIs de uso especial que atua como gateways para funcionalidades idiossincráticas. É provável que você pelo menos já tenha tentado about:config e about:mozilla no Firefox. De maneira semelhante, a família print: de URIs é o gateway do Konqueror para KDEPrint.

O URL print:/ mostra todas as possibilidades. print:/jobs monitora os trabalhos de impressão e print:/manager inicia o Print Manager dentro do Konqueror.

Observe que você não está lidando com o CUPS aqui, pelo menos não diretamente. Isso tudo faz parte da camada KDEPrint.

23.9 Leitura recomendada

SWEET, Michael. *CUPS: Common UNIX Printing System*. Indianapolis, Indiana: Sams Publishing, 2001. Essa é a bíblia do CUPS, diretamente da fonte.

Já mencionamos linuxprinting.org várias vezes neste capítulo. É uma vasta coleção dos recursos de impressão do Linux e um bom lugar para começar a responder a perguntas. Esse site também tem um interessante tutorial sobre o CUPS que inclui uma seção de solução de problemas.

Tanto o Wikipedia com o SUSE fornecem uma boa visão geral do CUPS:

en.opensuse.org/SDB:CUPS_in_a_Nutshell

en.wikipedia.org/wiki/Common_Unix_Printing_System

Você pode encontrar vários grupos de notícias relacionados ao CUPS em cups.org/newsgroups.php. Esse é um bom lugar para fazer perguntas, mas primeiro faça sua lição de casa e pergunte educadamente.

O KDE contém páginas man para os comandos KDEPrint e o *KDEPrint Handbook*. Você pode encontrar informações adicionais em printing.kde.org. Todas essas fontes contêm referências úteis a outra documentação. (Mesmo se você não tiver o KDE, a documentação do KDE contém boas informações gerais sobre o CUPS.)

23.10 Exercícios

E23.1 Utilizando um navegador Web, visite um servidor CUPS na sua rede. O que impede que você faça alterações administrativas nas impressoras desse servidor?

E23.2 Encontre alguém que não seja um literato em computação (estudante de artes, sua mãe ou talvez um Microsoft Certified Professional) e ensine essa pessoa a imprimir um documento PDF em um sistema Linux. Seu aluno achou algum dos passos confuso? Como você poderia tornar o processo mais fácil para outros usuários?

E23.3 Visite uma loja real ou virtual de uma grande empresa como o Sam's Club ou Amazon.com e liste as impressoras que você pode comprar por menos de US$ 200. Se você tivesse de comprar uma dessas impressoras para sua empresa amanhã, qual seria e por quê? Justifique sua análise com dados do banco de dados linuxprinting.org.

E23.4 Você foi solicitado a projetar o software de sistema para uma impressora a laser baseada em Linux destinada ao mercado corporativo. Com qual distribuição Linux você começará? Quais softwares adicionais você incluirá e quais softwares você terá de escrever? Como você acomodará clientes Windows e clientes Mac OS? (Dica: confira as distribuições Linux projetadas para "sistemas embutidos".)

24 Manutenção e ambiente

Com ao aumento do uso das estações de trabalho desktop e da diminuição do uso da computação baseada em mainframes, parecia que os dias de uma sala central de máquinas (conhecida como "centro de dados") estariam contados. Não tema! Ao longo da última década, esses sistemas desktop tornaram-se cada vez mais dependentes de um núcleo de servidores centrais que executam sistemas operacionais como o Linux. Como resultado, um grande número de servidores agora pode ser encontrado nessas salas de máquinas uma vez abandonadas.

Nunca foi tão importante assegurar um ambiente bem mantido e saudável para esses servidores. De fato, os requisitos elétricos e de ar-condicionado de um rack para os mais recentes servidores 1 U[1] freqüentemente são iguais ou excedem as demandas dos mainframes que eles substituem.

Este capítulo oferece algumas dicas sobre a manutenção de hardware e sobre como dar a ele uma boa moradia.

24.1 PRINCÍPIOS BÁSICOS DA MANUTENÇÃO DE HARDWARE

A manutenção de hardware era tradicionalmente coberta por um caro contrato anual de manutenção. Embora esses contratos ainda estejam disponíveis, hoje é mais comum e mais econômico usar a abordagem baseada na intuição e na experiência.

Consulte a página 635 para informações adicionais sobre a aposentadoria de hardware.

Se você mantiver um livro de registros, uma rápida olhada nos últimos seis a doze meses lhe dará uma idéia das taxas de falhas. É uma boa idéia manter um registro cuidadoso das falhas e substituições para que você possa avaliar de forma acurada as diversas opções de manutenção que você tem à disposição. Algumas peças dão problema com mais freqüência do que o previsto pelo fabricante, de modo que contratos de manutenção são algumas vezes não somente convenientes, como também financeiramente vantajosos. Mas, lembre-se, chegará uma época quando todo o hardware deverá ser substituído e não mais mantido. Conheça o seu hardware e aposente-o sem pena quando o tempo dele finalmente tiver chegado. Você até poderia considerar a doação de equipamentos obsoletos a uma escola ou universidade local. Para eles, raramente os equipamentos são tão antigos assim para não serem úteis.

Ao planejar sua estratégia de manutenção, considere quais componentes têm maior probabilidade de envelhecimento precoce. Os dispositivos que incluem partes móveis tendem a ser bem menos confiáveis que os dispositivos sólidos como CPUs e memória. Eis alguns candidatos comuns à aposentadoria:

- Unidades de fita
- Autocarregadores e trocadores de fita
- Unidades de disco rígido
- Ventoinhas
- Teclados
- Mouses
- Monitores CRT

24.2 CONTRATOS DE MANUTENÇÃO

Vários dos principais fornecedores oferecem manutenção de hardware para equipamentos computacionais que eles próprios não vendem. Esses fornecedores normalmente estão ávidos para desbancar o fabricante original e se plantam na sua porta, por assim dizer. Algumas vezes você pode negociar contratos de manutenção muito atraentes colocando um fabricante contra um fornecedor independente. Se possível, obtenha referências sobre todos os fornecedores de manutenção em potencial, preferivelmente de pessoas que você conheça e confie.

---

1. Um "U" tem 1,75 polegadas e é a unidade padrão de medida para o espaço de rack.

Manutenção local

Se você tiver um contrato de manutenção local, um técnico de serviços trará peças de reposição diretamente para onde se encontra sua máquina. O tempo de atendimento garantido varia de 4 a 24 horas; em geral, isso está explicitamente no contrato. Os tempos de atendimento durante o horário comercial podem ser menores do que em outros períodos da semana.

Se estiver considerando um contrato de manutenção com rápido tempo de atendimento, normalmente vale a pena calcular o custo de manter à mão alguns sistemas de backup completos para que você possa rapidamente substituir computadores com problemas. Uma troca total de sistema lhe dará um tempo de reparação mais rápido do que o mais sofisticado dos contratos de manutenção, e, com os baixos preços do hardware hoje em dia, muitas vezes o investimento é mínimo.

Manutenção para troca de placas

Um programa para troca de placas requer que você e a sua equipe diagnostiquem problemas, talvez com a ajuda do pessoal de suporte de seu fabricante. Após os diagnósticos, você liga para um número de manutenção, descreve o problema e faz o pedido da(s) placa(s) que precisa(m) ser substituída(s). Normalmente elas são despachadas imediatamente e chegam às suas mãos no dia seguinte. Você então instala a placa, faz com que o hardware volte a funcionar e devolve a placa defeituosa na mesma caixa em que veio a placa nova.

O fabricante normalmente irá querer atribuir um número de autorização de retorno de mercadoria (ARM) à transação. Certifique-se de escrever esse número nos documentos de remessa ao devolver a placa defeituosa.

Garantias

A extensão da garantia do fabricante deve desempenhar um papel significativo no cálculo do custo de propriedade de uma máquina durante seu tempo de vida. Na maioria dos casos, o melhor esquema de manutenção provavelmente seja a estratégia "garantia seletiva". Os fabricantes de unidades de disco oferecem garantias de até cinco anos e alguns módulos de memória chegam a oferecer até garantia por toda a vida. Uma garantia de um ano é padrão para computadores, mas garantias de vários anos ou mais não são raras. Ao adquirir equipamento novo, pesquise em busca da melhor garantia — isso lhe poupará dinheiro em longo prazo.

Em um ambiente universitário, parece ser mais fácil obter fundos governamentais para equipamentos de primeira linha do que para pessoal de suporte ou manutenção. Ocasionalmente pagamos por uma opção de "garantia estendida" em hardware novo (que também poderia ser descrita como manutenção pré-paga) para converter dinheiro destinado a equipamentos em dinheiro destinado à manutenção.

Com tantas peças de hardware, os maiores problemas de manutenção e de confiabilidade ocorrem quase sempre logo após a instalação. Falhas de hardware que ocorrem dentro de um a dois dias da instalação são conhecidas como "mortalidade infantil".

24.3 EXPERIÊNCIA COM O TRATAMENTO DE EQUIPAMENTOS ELETRÔNICOS

Placas de circuito devem ser manipuladas com cuidado, não podem cair no chão, não se pode derramar café nem ter livros empilhados sobre elas etc. A maioria dos engenheiros de campo (aquele pessoal de reparos que vêm com o seu contrato de manutenção) são dez vezes mais bruscos com as placas do que parece ser razoável.

Eletricidade estática

Os componentes eletrônicos são sensíveis à eletricidade estática. Para manusear as placas com segurança, você tem de criar um isolamento para você próprio antes e durante a instalação. Uma pulseira de aterramento usada no pulso e conectada ao "cabo-terra" (normalmente disponível como o terceiro pino de uma tomada elétrica) fornece proteção adequada.

Lembre-se de que você precisa se preocupar com estática ao abrir pela primeira vez o pacote contendo um componente eletrônico e todas as vezes que esse componente é manuseado — não apenas quando você por fim o instala. Tome especial cuidado se a sala onde você recebe sua correspondência (e onde provavelmente você se sentirá tentado a abrir os seus pacotes) tiver carpete; o carpete gera mais eletricidade estática do que um piso frio.

Uma forma de reduzir a estática em pisos acarpetados é adquirir um borrifador e enchê-lo com uma parte de amaciante de roupas e 30 partes de água. Borrife-o sobre o carpete (mas não sobre o computador) uma vez por mês para manter baixos os níveis de estática. Esse procedimento também deixa seu escritório com aquele aroma primaveril.

Reassentando placas

Ocasionalmente você mesmo pode resolver um problema de hardware apenas desligando o equipamento, limpando os contatos dos conectores de borda das placas de interface (SCSI, Ethernet etc.), reassentando as placas e ligando o sistema

novamente. Se isso funcionar temporariamente, mas o mesmo problema voltar daqui a uma semana ou um mês, o contato elétrico entre o cartão e a placa-mãe provavelmente está com problemas.

Os contatos podem ser limpos com uma solução de limpeza especial e kit de limpeza ou com uma borracha comum. Não use uma borracha velha e dura. Se sua borracha não funcionar bem para apagar marcas de lápis em uma folha de papel, ela também não funcionará bem em contatos elétricos. Tente manter seus dedos longe dos contatos. Apenas "apague-os" com a borracha (um abrasivo brando), escove os restos de borracha e reinstale a placa.

24.4 MONITORES

Ao longo dos últimos anos, tivemos a sorte de ver os preços dos monitores LCD caírem a um nível em que eles podem ser amplamente instalados. Embora o custo inicial dos LCDs seja um pouco mais alto do que o dos CRTs, esses dispositivos usam menos energia, menos manutenção e em geral causam menor estresse ocular que os seus predecessores baseados em CRT. Se você ainda tem monitores CRT na sua empresa, um bom plano de manutenção é simplesmente substituí-los por monitores LCD.

Se você ainda é forçado a manter monitores CRT, esteja ciente de que vários deles têm ajustes de brilho e de convergência só acessíveis a partir das placas de circuitos. Infelizmente, os monitores, em geral, usam cargas internas de dezenas de milhares de volts que podem persistir muito depois de terem sido desconectados da alimentação. Devido ao risco de choque elétrico, recomendamos que você sempre use técnicos qualificados para ajustar os monitores. Não tente fazer você mesmo esse serviço.

24.5 MÓDULOS DE MEMÓRIA

O hardware de hoje aceita memória na forma de SIMMs (Single Inline Memory Modules), DIMMs (Dual Inline Memory Modules) ou RIMMs (Rambus Inline Memory) em vez de chips individuais. Esses módulos variam de 32 MB a 4 GB, todos em um pequeno "pente".

Se precisar adicionar memória à estação de trabalho ou servidor, normalmente você mesmo poderá comprá-la de terceiros e instalá-la. Seja cauteloso ao comprar memória de fornecedores de informática; os preços costumam variar bastante de um fornecedor para outro.[2] Ao adicionar memória, pense grande. O preço da memória está caindo continuamente, mas o número de slots de expansão disponíveis em uma placa-mãe típica também está caindo correspondentemente.

Vale a pena verificar com atenção redobrada a documentação de seu sistema antes de adquirir memória para ter certeza de que você tem uma idéia clara dos tipos de módulos de memória que os seus sistemas aceitarão. Normalmente você pode aumentar o desempenho instalando memória que suporte uma taxa de transferência de barramento mais alta ou recursos especiais como DDR (Double Data Rate). Certifique-se do número de slots de memória que cada sistema tem disponível e se há ou não qualquer restrição no acréscimo de módulos novos. Alguns sistemas requerem que os módulos sejam adicionados aos pares; outros não exigem isso estritamente, mas podem oferecer maior desempenho quando os módulos são utilizados em pares.

Entenda perfeitamente como os módulos de memória antigos e novos interagirão. Na maioria dos casos, somente os recursos ou velocidades comuns a todos os módulos poderão realmente ser usados. Algumas vezes pode valer a pena eliminar a memória original de um sistema ao fazer uma atualização.

Se você mesmo instalar a memória, tenha em mente que a memória é mais sensível do que qualquer outra coisa à eletricidade estática. Certifique-se de estar bem aterrado antes de abrir uma embalagem cheia de módulos de memória.

Os módulos de memória normalmente são candidatos para a tecnologia de limpeza com borracha, descrita anteriormente neste capítulo.

24.6 MANUTENÇÃO PREVENTIVA

Pode parecer primitivo (e alguns de nós pensávamos que iríamos superar essa aflição), mas muitos hardwares têm filtros de ar que devem ser regularmente limpos ou trocados. Filtros bloqueados impedem o fluxo de ar e podem resultar em superaquecimento, uma causa importante de falha em equipamentos. É importante manter as saídas de ar em todos os equipamentos abertas e desobstruídas, mas preste atenção especialmente aos servidores densamente empacotados em pequenas aéreas de 1 U ou 2 U. A refrigeração desses sistemas depende da ventilação. Sem ela, você pode ter certeza de que ocorrerá um superaquecimento.

---

2. É uma história diferente se a memória for parte da oferta de um pacote; algumas dessas ofertas são muito boas.

Qualquer coisa com partes móveis pode precisar de lubrificação, limpeza e manutenção das correias feitas regularmente. Observe se não há ruídos estranhos provenientes de equipamentos mais antigos e trate-os de acordo.

Em sistemas servidores, as partes que falham com maior freqüência são a ventoinha e o módulo de fonte de alimentação — especialmente em PCs, em que ele normalmente é uma única unidade que pode ser substituída em campo. Verifique periodicamente seus servidores para ter certeza de que suas ventoinhas principais estão girando de forma rápida e firme. Se isso não ocorrer, geralmente é necessário substituir toda a fonte de alimentação de energia. Caso contrário, você corre o risco de superaquecer seu equipamento. Não tente lubrificar a própria ventoinha; esse procedimento poderia adiar a quebra inevitável, mas também poderia acelerar o problema ou provocar danos a outros componentes.

Muitos gabinetes de PC fornecem um local conveniente para montagem de uma segunda ventoinha (e conexões elétricas para alimentá-la). Se o ruído não for um problema, sempre é aconselhável instalar uma segunda ventoinha.[3] Além de reduzir a temperatura operacional dos componentes, a ventoinha extra atua como um backup se a ventoinha principal falhar. Ventoinhas extras são baratas; tenha em mãos algumas delas como peças de reposição.

Um computador em um ambiente empoeirado queimará componentes com uma freqüência muito maior do que um computador em um ambiente relativamente limpo. A poeira bloqueia filtros, resseca partes lubrificadas, bloqueia partes móveis (ventoinhas) e recobre os componentes com uma camada "isolante" de poeira que reduz a capacidade desses componentes de dissipar calor. Todos esses efeitos tendem a aumentar as temperaturas operacionais. Talvez você precise fazer ocasionalmente uma boa limpeza interna do sistema quando este estiver em ambientes desfavoráveis. (Qualquer ambiente dotado de carpete é forte candidato a ser desfavorável.)

Utilizar um aspirar de pó é a melhor maneira de remover poeira, mas mantenha o motor do aspirador a pelo menos um metro e meio dos componentes e discos para minimizar a interferência magnética. Deve-se limpar com um aspirador de pó o ambiente das máquinas regularmente, mas certifique-se de que essa tarefa seja executada por pessoas treinadas a respeitar distâncias apropriadas para não danificar os equipamentos (o pessoal de limpeza normalmente não é candidato aceitável para essa tarefa).

As unidades de fita normalmente também requerem limpeza regular. Você limpa a maioria das unidades, inserindo um cassete especial de limpeza.

24.7 Ambiente

Assim como os seres humanos, os computadores trabalham melhor e por mais tempo se estiverem felizes em seus ambientes de trabalho. Embora não liguem muito para ter uma janela com vista, eles realmente prestam atenção a outros aspectos do ambiente em que residem.

Temperatura

A temperatura ideal para operação de equipamentos de informática fica na faixa dos 17° a 20°C, com cerca de 45% de umidade. Infelizmente, essa temperatura não coincide com a temperatura de operação ideal dos usuários de computadores. Temperaturas ambientes acima de 27°C na sala dos equipamentos implicam cerca de 50°C dentro das máquinas. Chips comerciais apresentam um intervalo operacional de até cerca de 50°C, ponto em que eles param de funcionar; acima de 71°C, eles quebram. As temperaturas de entrada são críticas; o ar quente de uma máquina nunca deve fluir em direção à entrada de ar de uma outra máquina.

Umidade

A umidade ideal para a maior parte do hardware de informática está na faixa de 40% a 55%. Se a umidade for muito baixa, a eletricidade estática se torna um problema. Se ela for muito alta, pode se formar condensação sobre as placas, provocando curtos e oxidação.

Refrigeração do escritório

Nos dias de hoje, muitos computadores residem nas salas das pessoas e têm de sobreviver sob ar-condicionado (normalmente desligado à noite e nos fins de semana) e têm de suplantar uma boa dose de papéis e de livros sobre suas aberturas para ventilação. Ao colocar um computador em uma sala, tenha em mente que ele roubará ar-condicionado destinado aos seres humanos. Se você se encontrar em uma função em que poderá influenciar na capacidade de refrigeração, uma regra prática é que cada ser humano na sala produz 300 BTUH de calor, ao passo que um PC produz em média 1.100 BTUH. Não deixe os técnicos esquecerem de levar em conta a carga solar proveniente de janelas que recebem diretamente luz solar.

---

3. Ou conheça as últimas novidades sobre ventoinhas supersilenciosas em www.silentpcreview.com.

Refrigeração das salas de máquinas

Se tiver "sorte" o bastante para poder tirar servidores daquelas extravagantes salas de piso falso construídas na década de 1980, que tinham capacidade para refrigerar todo o seu equipamento *e* o Estado da Bahia juntos, então sua maior preocupação provavelmente será encontrar algum material instrutivo sobre manutenção de sistemas de refrigeração. Para o restante de nós, dimensionar corretamente um sistema de refrigeração é o que faz a diferença em longo prazo. Uma sala de máquinas bem refrigerada é uma sala de máquinas feliz.

Descobrimos que é uma boa idéia verificar com atenção redobrada as estimativas de carga de refrigeração fornecida pelo pessoal da área de sistemas de refrigeração, especialmente ao instalar um sistema para uma sala de máquinas. Certamente você precisará de um especialista na área para ajudá-lo a calcular a carga de refrigeração que tetos, paredes e janelas (não se esqueça da carga solar) contribuem para o ambiente. Os especialistas em sistemas de refrigeração normalmente têm muita experiência com esses componentes e devem estar aptos a lhe dar uma estimativa precisa. A parte que você precisa verificar é a carga de calor interna da sala de máquinas.

Você precisará determinar as parcelas de contribuição de carga de calor para os seguintes componentes:

- Teto, paredes e janelas (consulte um especialista em sistemas de refrigeração para essa estimativa)
- Equipamentos eletrônicos
- Luminárias
- Operadores (pessoas)

Equipamentos eletrônicos

Você pode estimar a carga de calor produzida pelos seus servidores (e outros dispositivos eletrônicos) determinando seu consumo de potência. Medição direta de consumo de potência é de longe o melhor método para obter essas informações. O eletricista do seu bairro freqüentemente pode ajudar, ou você pode comprar um medidor barato para que você mesmo faça isso.[4] O consumo máximo de energia da maioria dos equipamentos é medido em watts, mas o consumo típico tende a ser significativamente menor que o máximo. Você poderá converter esses números em unidades de calor padrão, BTUH, multiplicando por 3.413 BTUH/watt. Por exemplo, se quiséssemos construir uma sala de máquinas que fosse abrigar 25 servidores com consumo estimado em 450 watts cada, o cálculo seria:

$$\left(25\ \text{servidores}\right) \times \left(\frac{450\ \text{watts}}{\text{servidor}}\right) \times \left(\frac{3{,}412\ \text{BTUH}}{\text{watt}}\right) = 38.385\ \text{BTUH}$$

Luminárias

Como acontece com os equipamentos eletrônicos, você pode estimar a carga calorífica de luminárias com base no consumo de energia. As luminárias típicas para escritório contêm quatro lâmpadas fluorescentes de 40 watts cada. Se a sua sala tivesse seis dessas luminárias, o cálculo seria:

$$\left(6\ \text{luminárias}\right) \times \left(\frac{160\ \text{watts}}{\text{luminária}}\right) \times \left(\frac{3{,}412\ \text{BTUH}}{\text{watt}}\right) = 3.276\ \text{BTUH}$$

Operadores

De vez em quando, seres humanos precisarão entrar na sala das máquinas para fazer alguma coisa. Calcule 300 BTUH para cada ocupante. Para permitir quatro humanos na sala das máquinas ao mesmo tempo:

$$\left(4\ \text{humanos}\right) \times \left(\frac{300\ \text{BTUH}}{\text{humano}}\right) = 1.200\ \text{BTUH}$$

Carga calorífica total

Assim que tiver calculado a carga calorífica para cada componente, some tudo para determinar a carga calorífica total. Para nosso exemplo, supomos que nosso especialista em refrigeração estimou uma carga calorífica total proveniente de teto, paredes e janelas em 20.000 BTUH.

---

4. O KILL A WATT produzido pela P3 é uma escolha popular em torno de US$ 30.

20.000	BTUH para o teto, paredes e janelas
38.385	BTUH para servidores e outros equipamentos eletrônicos
3.276	BTUH para luminárias
1.200	BTUH para operadores
62.861	BTUH total

A capacidade de um sistema de refrigeração é tipicamente expressa em toneladas. Podemos converter BTUH para toneladas dividindo por 12.000 BTUH/ton. Você também deve permitir um fator de pelo menos 50% para levar em conta erros e crescimento futuro:

$$\left(62.681\ \text{BTUH}\right) \times \left(\frac{1\ \text{ton.}}{12.000\ \text{BTUH}}\right) \times \left(1{,}5\right) = 7{,}86 \text{ toneladas de refrigeração são necessários}$$

Veja em que grau sua estimativa coincide com aquela fornecida pelo pessoal especializado em refrigeração.

Monitoramento de temperatura

Se você estiver fornecendo suporte para um ambiente computacional crucial, é uma boa idéia monitorar a temperatura (e outros fatores ambientais como ruído e potência) na sala de máquinas mesmo quando você não estiver lá. É muito frustrante chegar na segunda-feira pela manhã e encontrar um monte de plástico derretido no chão da sala de máquinas. Felizmente, monitores de salas de máquinas automatizados podem observar as coisas enquanto você está fora. Usamos e recomendamos a família de produtos Phonetics Sensaphone. Esses computadores baratos monitoram variáveis de ambiente como temperatura, ruído e potência e telefonam para você (ou seu pager) quando for detectado algum problema. Para contatar a Phonetics em Aston, PA, Estados Unidos, visite o site em www.sensaphone.com.

24.8 ENERGIA ELÉTRICA

Os equipamentos de informática gostam de receber alimentação contínua, estável e limpa. Em uma sala de máquinas, isso significa um estabilizador de energia, um caro equipamento que filtra picos e pode ser ajustado para fornecer níveis de tensão e fases corretos. Em escritórios, protetores contra surtos de voltagem, ou estabilizadores, colocados entre as máquinas e as tomadas ajudarão a isolar os equipamentos dos picos de tensão.

Servidores e equipamentos de infra-estrutura de rede poderiam ser colocados em um no-break. No-breaks de qualidade possuem uma interface RS-232, Ethernet ou USB, que pode ser conectada à máquina a qual eles fornecem alimentação. Essa conexão permite que o no-break alerte o computador de que a alimentação falhou e que ele deve se desligar sozinho e de forma ordenada antes que as baterias acabem.

Consulte a página 28 para informações adicionais sobre procedimentos de desligamento.

Um estudo estimou que 13% da energia elétrica consumida nos Estados Unidos é utilizada para alimentar computadores. Tradicionalmente, computadores UNIX se baseavam em hardware e software que esperavam que a alimentação estivesse disponível 24 horas por dia. Nos dias de hoje, somente servidores e dispositivos de rede realmente precisam estar ativos todo o tempo. As máquinas desktop podem ser desligadas à noite se houver um modo fácil para os seus usuários fazerem isso (e se você confiar nos seus usuários para fazê-lo corretamente).

Ocasionalmente você pode se deparar com a situação na qual tenha de ligar/desligar regularmente um servidor devido a uma falha de hardware ou de kernel. Ou talvez você tenha servidores não-Linux em sua sala de máquinas mais suscetíveis a esse tipo de problema. Em ambos os casos, talvez seja interessante instalar um sistema que permita ligar e desligar servidores problemáticos por controle remoto.

Uma solução razoável é produzida pela American Power Conversion (APC). O seu produto MasterSwitch é similar a uma régua de alimentação, exceto pelo fato de poder ser controlado por um navegador Web por meio de sua porta Ethernet embutida. Para entrar em contato com a APC, ligue para (00xx1) (401) 789-0204 ou visite seu site em www.apcc.com.

24.9 RACKS

Já se foram os dias das salas de máquinas com piso elevado — em que a fiação elétrica, conexões de rede e cabos do sistema de refrigeração e das linhas telefônicas permaneciam ocultos embaixo do piso. Alguma vez você já tentou puxar um cabo que passa por baixo do piso de um desses labirintos? Nossa experiência é que embora pareça elegante através de um vidro, uma "clássica" sala com o piso elevado é um ninho oculto de ratos. Hoje em dia, você deve utilizar um piso elevado para ocultar fios elétricos, distribuir ar-condicionado e *nada mais*.

Se seu objetivo é operar seu equipamento de maneira profissional, uma sala de máquinas dedicada para servidores é essencial. Uma sala de servidores não somente fornece uma temperatura ambiente apropriada e controlada para suas máquinas, como também atende a suas necessidades de segurança física.

Em uma sala de máquinas dedicada, armazenar equipamentos em racks (em oposição a, digamos, dispô-los em mesas ou no chão) é a única opção profissional e de fácil manutenção. Hoje, os melhores esquemas de armazenamento utilizam racks que são interconectados em um sistema de trilhos suspensos para passagem de cabos. Essa abordagem fornece aquela sensação irresistível de alta tecnologia sem sacrificar a organização ou a manutenção.

O melhor sistema de trilhos suspensos é fabricado pela Chatsworth Products (Chatsworth, CA, (818) 882-8595). Utilizando racks de telecomunicação de trilha única padrão de 19", você pode montar servidores em prateleiras e em racks. Dois racks de telecomunicação de 19" com suas partes traseiras encostadas umas às outras compõem um rack "tradicional" de aspecto de alta tecnologia (para casos em que você precise conectar hardwares tanto pela frente como pela traseira do equipamento). A Chatsworth fornece os racks, canaletas para cabos e outros componentes para gerenciamento de cabos, bem como o hardware necessário para montá-los em seu prédio. Como os cabos repousam em trilhas visíveis, eles são fáceis de serem rastreados e você naturalmente será motivado a mantê-los organizados.

24.10 PADRÕES DOS CENTROS DE DADOS

As salas de servidor tornaram-se tão disseminadas que alguns grupos produziram padrões para especificá-las. Esses padrões em geral especificam atributos como a diversidade da conectividade da rede externa, o sistema elétrico e de refrigeração disponível (juntamente com planos de backup para esses recursos) e o tempo de inatividade anual para a manutenção dos recursos. O Uptime Institute publica uma série desses padrões; suas categorias estão resumidas na Tabela 24.1.

Tabela 24.1 Padrões de servidor do Uptime Institute

Camada	Tempo de funcionamento	Energia elétrica/refrigeração	Redundância
I	99,671%	Caminho único	Sem componentes redundantes
II	99,741%	Caminho único	Alguma redundância de componente
III	99,982%	Múltiplos caminhos, 1 ativo	Componentes redundantes, manutenção concorrente
IV	99,995%	Múltiplos caminhos, >1 ativo	Componentes redundantes, completamente tolerantes a falhas

Além de fornecer uma descrição detalhada de cada uma dessas camadas e descrever como alcançá-las, o Uptime Institute fornece informações e estatísticas sobre as melhores práticas para uma variedade de tópicos relevantes à infra-estrutura dos centros de dados tolerantes a falhas. Você pode acessá-las em www.upsite.com.

24.11 FERRAMENTAS

Um administrador de sistemas bem equipado é um administrador de sistemas eficiente. Ter uma caixa de ferramentas dedicada é fundamental para minimizar o tempo de inatividade em uma emergência. A Tabela 24.2 lista alguns itens para manter na sua caixa de ferramentas ou pelo menos dentro de fácil alcance.

Tabela 24.2 A caixa de ferramentas de um administrador de sistemas

Ferramentas genéricas	
Chaves Phillips: nº. 0, nº. 1 e nº. 2	Pinças
Chaves de fenda: 1/8", 3/16" e 5/16"	Tesoura
Faca de eletricista ou canivete suíço	Kit de chaves para aperto de soquetes
Alicates, comum e de bico	Pequena lanterna LED
Chaves de fenda de joalheiro	Kit de chaves sextavadas
Martelo de bola, 120 g	Kit de chaves de torção

Tabela 24.2 A caixa de ferramentas de um administrador de sistemas *(continuação)*.

Itens especiais relacionados com equipamentos de informática	
Descascador de fios (com um cortador de fios integrado)	Analisador portável de rede
Presilhas para cabos (e seus primos em velcro)	Cabo de energia sobressalente
Cabos cruzados Categoria RJ-45 sobressalentes	Crimpador RJ-45
Conectores RJ-45 sobressalentes (para cabos comuns e cabos trançados)	Terminadores SCSI
Multímetro digital	
Pulseira antiestática	
Diversos	
Lista de manutenção dos contatos de emergência[a]	Cotonetes de algodão
Números de telefones residenciais e de pager do pessoal de suporte por chamada	Fita isolante
Kit de primeiros socorros	Espelhinho de dentista
Uma caixa da melhor cerveja[b]	Telefone celular

a. E números de contrato de manutenção se aplicáveis.
b. Mínimo recomendável.

24.12 Leitura recomendada

As fontes a seguir apresentam informações adicionais sobre os padrões dos centros de dados.

Telecommunications Infrastructure Standard for Data Centers. ANSI/TIA/EIA 942.

ASHRAE INC. *ASHRAE Thermal Guidelines for Data Processing Environments*. Atlanta, GA: ASHRAE, Inc., 2004.

EUBANK, HUSTON; SWISHER, JOEL; BURNS, CAMERON; SEAL, JEN E EMERSON, BEN . *Design Recommendations for High Performance Data Centers*. Snowmass, CO: Rocky Mountain Institute, 2003.

24.13 Exercícios

E24.1 Por que você iria querer montar seus computadores em um rack?

★★ E24.2 Fatores ambientais afetam tanto pessoas quanto máquinas. Amplie os fatores listados neste livro com alguns fatores próprios (por exemplo, poeira, ruído, luminosidade, entupimentos etc). Escolha quatro fatores e avalie a adequação de sua instalação para as pessoas e as máquinas.

★ E24.3 Uma estação de trabalho puxa 0,8 A, e seu monitor puxa 0,7 A em 120 V.

a) Qual a potência em watts consumida por esse sistema? (Dica: P = EI.)

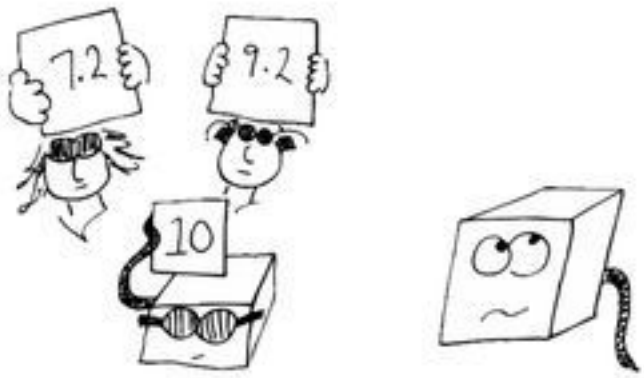
b) Com o custo da eletricidade por volta dos US$ 0,10/kWh, quanto custa deixar esse sistema ligado o ano inteiro?

c) Qual a quantia que pode ser poupada anualmente desligando-se o monitor por uma média de 16 horas por dia (seja manualmente ou usando-se recursos Energy Star como o Display Power Management Signaling)?

d) Qual o custo anual para refrigerar esse sistema? (Exponha suas suposições referentes aos custos de refrigeração e mostre os seus cálculos.)

★★ E24.4 Projete um novo laboratório de informática para a sua instalação. Exponha suas suposições referentes a espaço, números de máquinas e tipo e carga de potência de cada máquina. A seguir, calcule as necessidades de potência e de refrigeração para o laboratório. Inclua tanto os servidores quanto às estações de trabalho. Inclua também o layout da sala, a iluminação e a carga calorífica referentes às pessoas que ocuparão a sala.

25 Análise de desempenho



Este capítulo focaliza o desempenho dos sistemas que são utilizados como servidores. Sistemas desktop normalmente não experimentam os mesmos tipos de problemas de desempenho pelos quais os servidores passam, e a resposta à questão de como aumentar o desempenho em uma máquina desktop é quase sempre "Atualize o hardware". Os usuários gostam dessa resposta, pois isso significa que eles ganham sistemas novinhos em folha com maior freqüência.

Uma das maneiras de o Linux se diferenciar dos outros principais sistemas operacionais está na quantidade de dados que ele disponibiliza sobre seu próprio funcionamento interno. Há informações detalhadas para todos os níveis do sistema, e os administradores controlam uma variedade de parâmetros ajustáveis. Se você ainda tiver problemas para identificar a causa de um problema de desempenho, o código-fonte está sempre disponível para revisão. Por essas razões, o Linux costuma ser o sistema operacional preferido pelos consumidores mais preocupados com o desempenho.

Infelizmente, o ajuste de desempenho do Linux nem sempre é fácil. Usuários e administradores igualmente costumam pensar que, se eles soubessem a "mágica" certa, seus sistemas seriam duas vezes mais rápidos. Uma fantasia comum envolve o ajuste fino das variáveis de kernel que controlam o sistema de paginação e os reservatórios de buffers. Hoje em dia, o kernel das principais distribuições é pré-ajustado para atingir desempenho razoável (embora, admitidamente, não ótimo) sob uma série de condições de carga. Se você tentar otimizar o sistema tomando como base uma medida de desempenho em particular (por exemplo, utilização de buffer), muito provavelmente estará distorcendo o comportamento do sistema em relação a outras métricas de desempenho e condições de carga.

As questões mais sérias de desempenho em geral residem dentro de aplicativos e tem pouco a ver com o sistema operacional subjacente. A menos que tenha aplicativos ditos desenvolvidos internamente, você pode ter sorte ou estar destinado a gastar muito tempo no telefone com a equipe de suporte do fornecedor do aplicativo. Este capítulo discute o ajuste de desempenho no nível de sistema e deixa o ajuste no nível do aplicativo para outros.

Em todo caso, seja cético em relação a tudo o que você lê na Web. Na área de desempenho de sistemas, você verá argumentos superficialmente convincentes sobre todo tipo de assunto. Mas a maioria dos proponentes dessas teorias não possui conhecimento, disciplina e tempo necessários para elaborar experimentos válidos. Suporte popular significa muito pouco; para cada proposta estúpida, você pode esperar ver um coro grego: "Eu multipliquei o tamanho do meu cache de buffers por um fator de dez vezes, exatamente como disse o Zé e o meu sistema está MUITO mais rápido!!!". Certo.

O ajuste de desempenho é um trabalho árduo e exige paciência, uma abordagem metódica e análise cuidadosa. Eis algumas regras que você deve ter em mente:

- Coletar e revisar informações *históricas* sobre o seu sistema. Se o sistema estava executando bem há uma semana, é provável que um exame dos aspectos do sistema que mudaram leve a provas incontestáveis. Mantenha tendências e linhas de referência regularmente em sua manga para usá-la em uma emergência. Revise primeiro os arquivos de log para determinar se um problema de hardware desenvolveu.

  O Capítulo 19, *Gerenciamento e depuração de redes*, discute algumas ferramentas de análise de tendência que também são aplicáveis ao monitoramento de desempenho. O utilitário **sar** discutido na página 560 também pode ser utilizado como a ferramenta de análise de tendência dos pobres.
- Sempre ajuste seu sistema de maneira que permita a comparação dos resultados com a linha de referência anterior do sistema.
- Sempre se certifique de que tem um plano de reversão em caso de sua correção mágica realmente piorar as coisas.
- Não sobrecarregue intencionalmente os sistemas ou a rede. O Linux dá a cada processo uma ilusão de recursos infinitos. Mas tão logo 100% dos recursos do sistema estiverem em uso, o Linux terá de trabalhar muito para manter essa ilusão, atrasando processos e normalmente ele próprio consumindo uma parte razoável dos recursos.

25.1 O QUE VOCÊ PODE FAZER PARA MELHORAR O DESEMPENHO

Eis algumas coisas específicas que podem ser feitas para melhorar o desempenho:

- Certifique-se de que o sistema tem memória suficiente. Como veremos na próxima seção, o tamanho da memória tem uma influência significativa sobre o desempenho. A memória é tão barata hoje em dia que normalmente se pode pagar para ter carga máxima de memória em todas as máquinas sensíveis a questões de desempenho.
- Verifique duas vezes a configuração do sistema e de aplicativos individuais. Muitos aplicativos podem ser ajustados de tal maneira que ganhos incríveis de desempenho são obtidos (por exemplo, distribuindo dados entre os discos, não realizar pesquisas DNS em tempo de execução ou executando um número maior de instâncias de um servidor popular).
- Corrigir problemas de utilização, tanto aqueles causados pelos usuários (muitas tarefas sendo executadas ao mesmo tempo, práticas de programação ineficiente, tarefas executadas com prioridade excessiva e tarefas volumosas executadas em momentos inapropriados do dia) quanto aqueles provocados pelo sistema (cotas, contabilidade de CPU, daemons indesejados).
- Se estiver usando Linux como um servidor Web ou como algum outro tipo de servidor de aplicativo de rede, é recomendável distribuir tráfego entre vários sistemas com um equipamento de balanceamento de carga comercial como o Content Services Switch da Cisco (www.cisco.com), o ServerIron da Foundry (www.foundry.com) ou o Alteon Application Switch da Nortel (www.nortel.com).[1] Essas máquinas fazem vários servidores físicos parecerem um único servidor lógico para o mundo externo. Eles balanceiam a carga de acordo com um dos vários algoritmos selecionáveis pelo usuário como "servidor mais responsivo" ou "round robin" (rodízio).
- Esses balanceadores de carga também fornecem redundância prática caso um servidor tenha de ser desligado. Eles são realmente muito necessários se a sua instalação tiver de lidar com picos de tráfego inesperados.
- Organize discos rígidos e sistemas de arquivos de modo que a carga seja igualmente equilibrada, maximizando o throughput de E/S. Certifique-se de selecionar o agendador de E/S Linux apropriado para o disco (consulte a página 560 para detalhes). Para aplicativos específicos como bancos de dados, você pode utilizar uma sofisticada tecnologia de múltiplos discos como striped RAID para otimizar transferências de dados. Consulte seu fornecedor de bancos de dados para recomendações.
- É importante notar que os diferentes tipos de aplicativos e bancos de dados respondem de modo diferente quando distribuídos entre vários discos. Há muitas formas de RAID; leva tempo para determinar que forma (se houver) é apropriada para seu aplicativo em particular.
- Monitorar a rede para certificar-se de que ela não esteja saturada com tráfego e que a taxa de erros seja baixa. Uma enorme quantidade de informações sobre rede se encontra disponível por meio do comando **netstat**, descrito na página 449. Consulte também o Capítulo 19, *Gerenciamento e depuração de redes*.
- Identificar situações em que o sistema seja fundamentalmente inadequado para satisfazer às demandas exigidas.

Esses passos são listados mais ou menos em ordem de eficiência. Adicionar memória e balancear tráfego entre os servidores pode muitas vezes fazer uma enorme diferença em termos de desempenho. A eficácia de outras medidas varia de notáveis a nenhuma.

A análise e otimização de algoritmos e estruturas de dados de software quase sempre levam a ganhos de desempenho significativos. Mas a menos que você tenha uma base substancial de software local, esse nível de projeto normalmente fica fora de controle.

25.2 FATORES QUE AFETAM O DESEMPENHO

O desempenho percebido é determinado pela eficiência com a qual os recursos do sistema são alocados e compartilhados. A definição exata de um "recurso" é um tanto vaga. Ela pode incluir itens como contextos em cache no chip da CPU e entradas na tabela de endereços do controlador de memória. Entretanto, para uma primeira abordagem, somente os quatro recursos a seguir têm grande efeito sobre o desempenho:

- Tempo de CPU
- Memória
- E/S de disco rígido
- E/S de rede

1. Uma alternativa gratuita (mas relativamente menos estável), é o software Linux Virtual Server disponível em linuxvirtualserver.org.

Todos os processos consomem parte dos recursos do sistema. Se ainda sobrarem recursos depois de processos ativos terem obtido o que queriam, o desempenho do sistema será quase máximo.

Se não houver recursos suficientes para utilizar, os processos têm de ser executados em filas. Um processo que não tem acesso imediato a recursos de que ele precisa tem de aguardar sem fazer nada. A quantidade de tempo gasta aguardando é uma das medidas básicas de degradação de desempenho.

O tempo de CPU é um dos recursos mais fáceis de serem medidos. Uma quantidade constante de poder de processamento está sempre disponível. Teoricamente, essa quantidade é 100% dos ciclos de CPU, mas o overhead e diversas ineficiências fazem com que o número prático seja mais provavelmente 95%. Um processo que esteja utilizando mais do que 90% dos recursos de CPU estão inteiramente limitados pela CPU e consumindo grande parte do poder de processamento disponível no sistema.

Muitas pessoas supõem que a velocidade da CPU é o fator mais importante que afeta o desempenho global de um sistema. Dadas às quantidades infinitas de todos os demais recursos ou de certos tipos de aplicativos (por exemplo, simulações numéricas), uma CPU mais rápida *faz* uma diferença significativa. Mas, no mundo prático, a velocidade de CPU não é tão importante.

Um gargalo comum de desempenho em sistemas Linux é a largura de banda de disco. Pelo fato de os discos rígidos serem sistemas mecânicos, leva muitos milissegundos para se localizar um bloco em disco, recuperar o seu conteúdo e acordar o processo que está aguardando por ele. Atrasos dessa magnitude ofuscam cada uma das demais fontes de degradação de desempenho. Cada acesso a disco dispara centenas de milhões de instruções de CPU.

Pelo fato de o Linux fornecer memória virtual, a largura de banda em disco e a memória estão diretamente relacionadas. Em um sistema carregado com uma quantidade limitada de RAM, normalmente temos de gravar uma página em disco para obter uma página atualizada de memória virtual. Infelizmente, isso significa que utilizar memória é normalmente tão caro quanto utilizar o disco. O swap e a paginação causada pelo software inchado são os principais inimigos do desempenho na maioria das estações de trabalho.

A largura de banda de rede lembra a largura de banda em disco em diversos aspectos, devido às latências envolvidas. Entretanto, as redes são atípicas no sentido de que envolvem comunidades inteiras em vez de computadores individuais. Elas também são suscetíveis a problemas de hardware e a servidores sobrecarregados.

25.3 VERIFICAÇÃO DE DESEMPENHO DO SISTEMA

A maioria das ferramentas para análise de desempenho nos informa sobre o que está acontecendo em um determinado ponto no tempo. Entretanto, o número e o caráter das cargas geralmente mudam ao longo do dia. Certifique-se de coletar um grupo representativo de dados antes de tomar uma ação. As melhores informações sobre desempenho de sistema normalmente se tornam claras apenas após um longo período (um mês ou mais) de coleta de dados. É particularmente importante coletar dados durante períodos de pico de uso. As limitações de recursos e de configurações incorretas de sistema normalmente são visíveis apenas quando a máquina se encontra sob uma carga pesada.

Analisando o emprego de CPU

Provavelmente você vai querer coletar três tipos de dados de CPU: utilização global, médias de cargas e consumo de CPU por processo. A utilização global pode ajudar a identificar sistemas em que a velocidade da CPU em si é o gargalo. As médias de carga oferecem uma idéia do desempenho do sistema como um todo. Os dados de consumo da CPU por processo podem identificar processos específicos que estão devorando recursos.

Você pode obter informações sintetizadas com o comando **vmstat**. **vmstat** aceita dois argumentos: o número de segundos para monitorar o sistema para cada linha de saída e o número de relatórios a fornecer. Se você não especificar o número de relatórios, **vmstat** será executado até você pressionar <Control-C>. A primeira linha de dados retornada por **vmstat** fornece médias desde o momento em que o sistema foi inicializado. As linhas subseqüentes são medidas dentro do período de amostragem anterior, cujo padrão é de cinco segundos. Por exemplo:

```
$ vmstat 5 5
procs-----------memory----------  --swap-- ----io--   --system--  ----cpu----
 r b  swpd     free buff    cache   si so    bi  bo    in   cs  us sy id wa
 1 0   820  2606356 428776 487092    0  0  4741  65  1063 4857  25  1 73  0
 1 0   820  2570324 428812 510196    0  0  4613  11  1054 4732  25  1 74  0
 1 0   820  2539028 428852 535636    0  0  5099  13  1057 5219  90  1  9  0
 1 0   820  2472340 428920 581588    0  0  4536  10  1056 4686  87  3 10  0
 3 0   820  2440276 428960 605728    0  0  4818  21  1060 4943  20  3 77  0
```

O tempo de usuário, o tempo de sistema (kernel), o tempo de inatividade e o tempo de espera para E/S são mostrados nas colunas `us`, `sy`, `id` e `wa` à direita. Os números CPU que são pesados em tempo de usuário geralmente indicam compu-

tação e os números altos de sistema indicam que os processos estão fazendo uma grande quantidade de chamadas de sistema ou executando E/S.

Uma regra geral para servidores de computação de propósitos gerais que nos serviu bem por anos é a de que o sistema deve gastar aproximadamente 50% de seu tempo não-ocioso em espaço de usuário e 50% em espaço de sistema; a porcentagem de tempo ocioso total deve ser não-zero. Se você estiver dedicando um servidor para um único aplicativo que utiliza intensamente a CPU, a maior parte do tempo deve ser gasta em espaço de usuário.

A coluna `cs` ilustra alternações de contexto por intervalo, o número de vezes que o kernel mudou enquanto o processo estava sendo executado. O número de interrupções por intervalo (normalmente gerado por dispositivos de hardware ou componentes do kernel) é ilustrado na coluna `in`. Valores `cs` ou `in` extremamente elevados em geral indicam um dispositivo de hardware se comportando mal ou está mal configurado. As demais colunas são úteis para análise de memória e de disco, que discutiremos posteriormente neste capítulo.

Médias de longo prazo das estatísticas de CPU permitem que você determine se fundamentalmente há poder de processamento suficiente para satisfazer a demanda. Se a CPU normalmente gasta parte de seu tempo no estado ocioso, há ciclos de reserva. Atualizar para uma CPU mais rápida não irá aumentar muito o desempenho global do sistema, embora ele possa acelerar operações individuais.

Como você pode observar nesse exemplo, a CPU geralmente fica alternando entre uso pleno e ociosidade completa. Portanto, é importante observar esses números como uma média ao longo do tempo. Quanto menor for o intervalo de monitoramento, menos consistentes serão os resultados.

Em máquinas multiprocessadas, a maioria das ferramentas Linux apresenta uma média de estatística de processador entre todos os processadores. O comando **mpstat** gera saída parecida com **vmstat** para cada processador. O flag **-P** permite que você especifique um determinado processador sobre o qual será gerado o relatório. **mpstat** é útil para depuração de software que suporta multiprocessamento simétrico (Symmetric MultiProcessing – SMP) — também é esclarecedor ver se o sistema está utilizando processadores múltiplos de maneira eficiente ou ineficiente.

Em uma estação de trabalho com somente um usuário, a CPU geralmente gasta 99% de seu tempo no estado ocioso. Portanto, quando você for rolar uma das janelas em seu vídeo de mapa de bits, a CPU é derrubada por um curto período. Nessa situação, as informações sobre emprego médio de CPU em longo prazo não são significativas.

O segundo tipo de informações estatísticas sobre CPU útil para caracterizar a carga no sistema é a "média de carga", que representa o número médio de processos que podem ser executados. Isso dá uma boa idéia do número de partes em que a torta de CPU está sendo dividida. A média de carga é obtida com o comando **uptime**:

```
$ uptime
11:10am  up 34 days, 18:42, 5 users, load average: 0.95, 0.38, 0.31
```

São dados três valores, correspondentes a médias de 5, 10 e 15 minutos. Em geral, quanto maior for a média de carga, mais importante se torna o desempenho agregado do sistema. Quando há somente um processo executável, esse processo normalmente é limitado por um único recurso (comumente, a largura de banda de disco ou a CPU). A demanda de pico para esse recurso em particular se torna o fator determinante no desempenho.

Quando um número maior de processos estiver compartilhando o sistema, as cargas podem ou não ser igualmente distribuídas. Se os processos no sistema consumirem uma mistura de CPU, de disco e de memória, o desempenho do sistema será menos provavelmente dominado por restrições em um único recurso. Nessa situação, torna-se mais importante observar as medidas médias de consumo como a utilização total de CPU.

Consulte a página 39 para informações adicionais sobre prioridades.

Em geral, os sistemas Linux modernos estão ocupados com uma média de carga 3 e não se dão bem com médias de carga acima de aproximadamente 8. Uma média de carga dessa magnitude é uma dica de que você deve começar a procurar maneiras de distribuir artificialmente a carga, como utilizar **nice** para configurar prioridades de processo.

A média de carga do sistema é uma excelente métrica a rastrear como parte de uma linha de referência do sistema. Se você souber a média de carga de seu sistema em um dia normal e essa estiver na mesma faixa da de um dia ruim, isto é indicativo de que você deve procurar algum outro fator como causa dos problemas de desempenho (como, por exemplo, a rede). Uma média de carga acima da norma esperada sugere que você deve examinar os processos que executam no próprio sistema Linux.

Outra maneira de ver o uso de CPU é executar o comando **ps** com os argumentos **aux**, que mostram quanto da CPU cada processo está utilizando. Em um sistema ocupado, pelo menos 70% da CPU normalmente é consumida por apenas um ou dois processos. (Lembre-se de que o próprio **ps** consome um pouco dos recursos da CPU.) Adiar a execução de grandes consumidores de recursos de CPU ou reduzir suas prioridades faz com que a CPU se torne mais disponível para outros processos.

Consulte a página 46 para informações adicionais sobre **top**.

Uma excelente alternativa ao **ps** é um programa chamado **top**. **top** apresenta praticamente as mesmas informações que **ps**, mas em um formato "vivo" que lhe permite observar o estado da mudança do sistema ao longo do tempo.[2]

Como o Linux gerencia memória

Antes de nos aprofundarmos nos detalhes do gerenciamento de memória no Linux, vale a pena mencionar mais uma vez que o kernel do Linux evolui mais rapidamente do que qualquer outro sistema operacional. O sistema de gerenciamento de memória do Linux passou por um período de fluxo especialmente rápido, mesmo para padrões Linux, por muitos anos.

O kernel 2.6 tem um sistema VM vastamente aprimorado.[3] Entretanto, no nível da discussão neste capítulo, a versão exata de kernel não é realmente importante. Esses conceitos se aplicam independentemente da versão exata do kernel em uso.

Como o UNIX, o Linux gerencia memória em unidades chamadas de páginas. O tamanho atual de uma página de memória no PC é de 4 KB.[4] O kernel do Linux aloca páginas virtuais para processos à medida que requisitam memória. Cada página virtual é associada a armazenamento real, seja RAM ou "espaço de reserva" em disco. (O espaço de reserva, *backing store*, é normalmente o espaço na área de swap, mas, para páginas que contêm texto de programa executável, o espaço de reserva é o arquivo executável original.) O Linux utiliza uma "tabela de páginas" para controlar a associação entre essas páginas virtuais e páginas reais de memória.

O Linux pode alocar efetivamente tanta memória quanto os processos solicitarem ampliando a RAM real com espaço de swap. Como os processos esperam que suas páginas virtuais se associem à memória real, o Linux está constantemente atarefado movendo páginas entre a RAM e o disco. Essa atividade é conhecida como paginação.[5]

O Linux tenta gerenciar a memória do sistema para que as páginas que foram acessadas recentemente sejam mantidas em memória e páginas menos ativas sejam colocadas em disco. Esse esquema é conhecido como sistema LRU, já que as páginas utilizadas menos recentemente são aquelas excluídas. Seria muito ineficiente para o kernel realmente controlar todas as referências de memória, de modo que o Linux utiliza um cache de páginas para decidir que páginas excluir. Esse sistema é muito mais barato do que um sistema LRU verdadeiro, mas produz resultados similares.

O Linux mantém um controle de uma "idade" para cada página de memória virtual. Toda vez que o Linux examina uma página e descobre que ela foi recentemente referida, ele incrementa a idade da página. (O termo "idade" é um tanto incorreto, pois o valor realmente indica a freqüência e o quão recente foi o acesso. Quanto maior a idade, mais recente será a página.) Enquanto isso, o Linux executa o processo **kswapd**, que diminui regularmente as idades de páginas não-referenciadas.

O kernel mantém várias listas de páginas de memória. Páginas com uma idade maior que zero são marcadas como "ativas" e estão contidas na "lista de páginas ativas" do cache de páginas. Se a idade de uma página atingir zero, **kswapd** intercambiará a página para a "lista de páginas inativas". **kswapd** remove páginas inativas da tabela de páginas e as considera elegíveis para serem movidas para disco. Embora páginas inativas não sejam mais imediatamente acessíveis por meio da tabela de páginas, o kernel irá recuperá-las da memória ou disco e as colocará de volta na tabela de páginas sob demanda.

Quando a memória está baixa, o kernel tenta descobrir quais páginas na lista de páginas inativas foram utilizadas menos recentemente. Se essas páginas tiverem sido modificadas por um processo, o Linux as considera "sujas" e tem de movê-las para o disco antes de a memória poder ser reutilizada. As páginas que foram lavadas dessa maneira, ou que jamais ficaram sujas antes de qualquer coisa, estão "limpas" e podem ser recicladas pelo Linux para emprego em algum outro lugar.

Quando uma página na lista de páginas inativas for referida por um processo, o kernel retorna seu mapeamento de memória para a tabela de páginas, reinicializa a idade da página e a transfere da lista de páginas inativas para a lista de páginas ativas. Páginas que foram gravadas em disco têm de ser movidas para a memória antes de poderem ser reativadas. Uma "falha leve" ocorre quando um processo se refere a uma página inativa na memória, e uma "falha grave" resulta de uma referência a uma página não-residente. Em outras palavras, uma falha grave requer que uma página seja lida do disco e uma falha leve, não.

A demanda por memória varia, para que o kernel possa executar **kswapd** em diferentes velocidades. Quando a demanda para memória é extrema, o algoritmo é executado mais freqüentemente e, portanto, as páginas devem ser referenciados mais freqüentemente para evitar serem movidas para disco porque sua idade diminui para zero.

Para identificar páginas ativas entre as inativas, o sistema de memória virtual (*virtual memory* – VM) depende do atraso entre o momento em que uma página é colocada numa lista de páginas inativas e o momento em que ela é efetivamente

---

2. Atualizar a saída de **top** muito rapidamente pode por si só ser um voraz consumidor de recursos de CPU; portanto, seja criterioso no emprego de **top**.
3. O kernel 2.6 inclui a opção de executar o sistema *sem* memória virtual — uma opção que é particularmente útil para sistemas embarcados. Se optar por executar sem memória virtual, você deve assegurar que haja memória física suficiente para satisfazer todas as possíveis demandas.
4. Você também pode utilizar páginas de 4 MB configurando HPAGE_SIZE no kernel 2.6.
5. Note que a "paginação" e o "swap" são implementados da mesma maneira e utilizam a mesma área de "swap" para armazenar páginas quando as páginas não estão em RAM.

"paginada", isto é, movida para o disco. Portanto, o sistema VM tem de prever atividade de paginação futura para decidir com que freqüência deve executar o algoritmo de **kswapd**. Se o algoritmo de **kswapd** não for executado com freqüência suficiente, talvez não haja páginas inativas limpas suficientes para satisfazer a demanda. Se ele executar com muita freqüência, o kernel gastará muito tempo processando falhas de páginas leves.

Como o algoritmo de paginação é previsível, não necessariamente há uma correspondência unívoca entre eventos de paginação e alocações de páginas por parte de processos em execução. O objetivo do sistema é manter memória livre suficiente para que processos não tenham efetivamente de aguardar por uma página cada vez que eles fizerem uma nova alocação. Se a paginação aumentar significativamente quando o sistema estiver muito ocupado, talvez seja interessante ter mais RAM.

Você pode ajustar o parâmetro "swappiness" do kernel (**/proc/sys/vm/swappiness**) para dar uma dica ao kernel sobre como ele deve rapidamente fazer com que páginas físicas elegíveis sejam reivindicadas de um processo em caso de uma falta de memória. Por padrão, esse parâmetro tem um valor de 60. Se você configurá-lo como 0, o kernel recorre à reivindicação de páginas que foram atribuídas a um processo apenas quando ele esgotou todas as outras possibilidades. Se estiver configurado com um valor maior que 60 (o valor máximo é 100), o kernel provavelmente reivindicará páginas de um processo quando não tiver mais memória. Em todos os casos, se você se vir tentado a modificar esse parâmetro, talvez esteja na hora de comprar mais RAM para o sistema.

Se o kernel preenche tanto a RAM como o swap, toda a VM foi esgotada. O Linux usa um "eliminador de memória insuficiente" para tratar dessa condição. Essa função seleciona e elimina um processo para liberar memória. Embora o kernel tente eliminar o processo menos importante em seu sistema, esgotar a capacidade de memória sempre é algo a ser evitado. Nessa situação, é provável que uma porção substancial dos recursos de sistema esteja sendo dedicada à limpeza de memória em vez de a trabalho útil.

Até mesmo processos sendo executados com prioridade de CPU baixa podem surrupiar páginas. Suponha, por exemplo, que você esteja rodando um cliente SETI[6] com prioridade muito baixa (alto valor de **nice**) em sua estação de trabalho e ao mesmo tempo lendo e-mails em uma janela de terminal. Ao fazer uma pausa para ler uma mensagem, o emprego de CPU cai para zero e a simulação é permitida. Ele traz todas as suas páginas, forçando o shell, o servidor de janelas, o programa de leitura de e-mails e emulador de terminal a serem retirados da memória. Ao avançar para a próxima mensagem, há um atraso, já que uma grande parte da memória do sistema é movida para disco. Na vida real, um alto valor de **nice** não é garantia de que um processo não provocará problemas de desempenho.

Analisando o emprego de memória

Três números quantificam a atividade de memória: a quantidade total de memória virtual ativa e as taxas de swap e de paginação. O primeiro número informa a demanda total por memória e os dois seguintes sugerem a proporção dessa memória que está sendo utilizada ativamente. O objetivo é reduzir a atividade ou aumentar a memória até que a paginação permaneça em um nível aceitável. A paginação ocasional é inevitável; não tente eliminá-la completamente.

Você pode utilizar o comando **free** para determinar a quantidade de memória e de espaço de swap que estão efetivamente em uso. Use-o com o flag **-t** para calcular automaticamente a quantidade total de memória virtual:

```
$ free -t
                 total       used       free     shared    buffers     cached
Mem:            127884      96888      30996      46840      57860      10352
-/+ buffers/cache:          28676      99208
Swap:           265032       3576     261456
Total:          392916     100464     292452
```

A coluna `free` indica o número de kilobytes na lista livre do sistema; valores menores do que 3% da memória total do sistema geralmente são indicativos de problemas. Utilize o comando **swapon** para determinar exatamente que arquivos e partições estão sendo utilizados para o espaço de swap:

```
$ swapon -s
Filename        Type        Size    Used   Priority
/dev/sdb7       partition   265032  3576   -1
```

Esse sistema utiliza uma partição de disco, sdb7, para swap. Como a atividade de paginação pode ser um vazamento significativo em termos de largura de banda de disco, sempre é melhor, se possível, distribuir o espaço de swap por vários discos físicos. O emprego de vários discos de swap diminui o tempo necessário que um disco deve gastar lidando com dados swap e aumenta a largura de banda de armazenamento de swap.

---

6. SETI (Search for ExtraTerrestrial Intelligence); visite o site http://www.seti.org/science/setiathome.html.

Em sistemas UNIX tradicionais, o comando **vmstat** revela informações sobre paginação e swapping. Entretanto, o comando **procinfo** que vem com a maioria das distribuições Linux é uma fonte de informações melhor. Embora **procinfo** venha com todos os sistemas utilizados como exemplo em nosso livro, ele não é necessariamente incluído na instalação padrão, de modo que talvez você precise adicionar o pacote **procinfo** manualmente.

Consulte a página 600 para informações adicionais sobre o sistema de arquivos /proc.

procinfo não possui uma maneira especial de obter informações sobre seu sistema; ele simplesmente formata os dados dos arquivos em seu sistema de arquivos **/proc**. Sem ele, interpretar os arquivos **/proc** pode se tornar um desafio considerável. Para atualizações contínuas a cada cinco segundos, execute **procinfo -n5**.

```
$ procinfo
Linux 2.6.9-34.EL (root@bull) (gcc 3.4.3 20050227 ) #1 1CPU [main]

Memory:        Total      Used       Free   Shared    Buffers
Mem:          463728    371512      92216        0     144042
Swap:        1866470       218    1626252

Bootup: Sat Jul 29 08:47:20 2006    Load average: 0.08 0.06 0.00

user   :        1:20:32.12         1.3%        page in :       0
nice   :        0:02:36.30         0.0%       page out :       0
system :        3:06:46.90         0.8%        swap in :       0
idle   :      1d 7:23:49.50       97.9%       swap out :       0
steal  :        0:00:00.00         0.0%
uptime :      4d 1:51:43.64                    context :       55465717

irq 0:     5748875398    timer              irq  8:               3  rtc
irq 1:              3                       irq 11:        75300822  eth0
irq 2:              0    cascade [4]        irq 12:               3
irq 4:              4                       irq 14:         7529735  ide0
```

Algumas das informações na saída de **procinfo** sobrepõem aquelas de **free**, **uptime** e **vmstat**. Além disso, **procinfo** fornece informações sobre sua distribuição de kernel, paginação de memória, acesso a disco e atribuições de IRQ. Você pode utilizar **procinfo -a** para ver até mais informações de seu sistema de arquivos **/proc**, inclusive opções de boot de kernel, módulos carregáveis de kernel, dispositivos de caracteres e sistemas de arquivos.

Quaisquer inconsistências aparentes entre as colunas relacionadas com memória são em grande parte ilusórias. Algumas colunas contam páginas e outras contam kilobytes. Todos os valores são médias arredondadas. Além disso, algumas são médias de quantidades escalares e outras são deltas médios. Por exemplo, você não poderia calcular o próximo valor de `Free` a partir do `Free` atual e das informações de paginação atuais, pois os eventos de paginação que determinam o próximo valor médio de `Free` ainda não ocorreram.

Utilize os campos `page in/out` e `swap in/out` para avaliar a paginação do sistema e o comportamento de swapping. Um page-in não representa necessariamente uma página que está sendo recuperada da área de swap. Ele poderia ser um código executável sendo paginado a partir de um sistema de arquivos ou uma página de "cópia na gravação" sendo duplicada, ambos os quais são ocorrências normais que não indicam necessariamente uma falta de memória. Por outro lado, page-outs sempre representam dados gravados em disco depois de serem forçadamente expulsos pelo kernel.

Se seu sistema tiver um fluxo constante de page-outs, é possível que você se beneficie de mais memória. Mas se a paginação acontece apenas ocasionalmente e não produz soluços irritantes ou reclamações por parte dos usuários, você pode ignorá-la. Se seu sistema se encaixa em um meio-termo, a análise adicional dependerá se você está tentando otimizar para desempenho interativo (por exemplo, uma estação de trabalho) ou tentando configurar uma máquina com muitos usuários simultâneos (por exemplo, um servidor).

Se metade das operações forem page-outs, você poderá calcular que cada 100 page-outs provoca cerca de um segundo de latência. Se 150 page-outs devem ocorrer para lhe permitir rolar uma janela, você aguardará cerca de 1,5 segundo. Uma regra prática utilizada por desenvolvedores de interfaces é que um usuário mediano percebe que o sistema está "lento" quando os tempos de resposta forem maiores do que sete décimos de segundo.

Também vale a pena ressaltar que **procinfo** fornece algumas informações de CPU que não são visíveis com **vmstat** ou **uptime**. Além de relatar a média de carga em períodos de 5, 10 e 15 minutos, **procinfo** lista o número instantâneo de processos em execução, o número total de processos e o ID do último processo executado. Por exemplo, eis um trecho da saída gerada por **procinfo** em um servidor atarefado:

```
Load average:        2.37        0.71        0.29        3/67        26941
```

Esse servidor tem 67 processos no total, dos quais três são executáveis. O último processo a ser executado tinha PID 26941 (neste caso, era o processo de shell por meio do qual **procinfo** era executado).

Analisando a E/S de disco

Você pode monitorar o desempenho de disco com o comando **iostat**. Assim como **vmstat**, ele aceita argumentos opcionais para especificar um intervalo em segundos e uma contagem de repetição, e sua primeira linha de saída é um sumário desde o momento da inicialização do sistema. Como **vmstat**, ele também o informa como o tempo de CPU está sendo gasto. Eis um rápido exemplo que omite algumas informações que não são específicas de discos:

```
$ iostat
...
Device:     tps     Blk_read/s   Blk_wrtn/s   Blk_read     Blk_wrtn
hdisk0     0.54           0.59         2.39     304483      1228123
hdisk1     0.34           0.27         0.42     140912       216218
hdisk2     0.01           0.02         0.05       5794        15320
hdisk3     0.00           0.00         0.00          0            0
```

iostat coleta informações do sistema de arquivos **/proc** para produzir uma linha de saída para cada disco físico em seu sistema. Infelizmente, o Linux mantém apenas informações estatísticas mínimas sobre o disco e até mesmo as informações que são preservadas são de uso limitado. Cada disco rígido possui as colunas `tps`, `Blk_read/s`, `Blk_wrtn/s`, `Blk_read` e `Blk_wrtn`, indicando transferências de E/S por segundo, blocos lidos por segundo, blocos gravados por segundo, total de blocos lidos e total de blocos gravados. Linhas de zeros são informadas se o sistema tiver menos de quatro discos.

Em geral, os blocos de disco têm um tamanho igual a 1 KB; portanto, você pode rapidamente determinar o desempenho real do disco em kilobytes. Por outro lado, as transferências são definidas de maneira relativamente nebulosa. Uma requisição de transferência pode incluir várias requisições de E/S lógicas em relação a vários setores; portanto, esses dados também são mais úteis para identificar tendências ou comportamento irregular.

O custo de busca é o fator mais importante afetando o desempenho da unidade de disco. Para uma primeira aproximação, a velocidade rotacional do disco e a velocidade do barramento com o qual ele está conectado têm relativamente pouco impacto. Discos modernos podem transferir dezenas de megabytes de dados por segundo caso eles sejam lidos de setores contíguos, mas eles podem realizar somente 100 a 300 buscas por segundo. Se você transferir um setor por busca, poderá facilmente constatar menos do que 5% do desempenho de pico da unidade de disco.

Buscas são mais custosas quando elas fazem com que as cabeças do disco percorram uma longa distância. Se você tem um disco com várias partições de disco e os arquivos são lidos de cada partição em uma ordem aleatória, as cabeças têm de ficar indo e voltando por um longo percurso para alternar entre partições. Por outro lado, arquivos dentro de uma partição são relativamente locais entre si. Ao particionar um disco novo, talvez você queira considerar as implicações de desempenho e colocar arquivos que são acessados juntos no mesmo sistema de arquivos.

Para efetivamente atingir o máximo de desempenho em disco, você deve colocar os sistemas de arquivos que são utilizados juntos em discos diferentes. Embora a arquitetura de barramento e os drivers de dispositivo afetem a eficiência, a maioria dos computadores pode gerenciar vários discos independentemente, aumentando assim o throughput de modo significativo. Por exemplo, muitas vezes vale a pena distribuir dados de servidor Web freqüentemente acessados e entrar no sistema escolhendo um de vários discos.

É especialmente importante distribuir, se possível, a área de swap entre vários discos, uma vez que a paginação tende a reduzir o desempenho de todo o sistema. Essa configuração é suportada pelo comando **swapon**. O kernel do Linux pode utilizar tanto partições de swap dedicadas como arquivos de swap em um sistema de arquivos formatado.

Sistemas Linux modernos permitem que você configure vários "sistemas de arquivos baseados em memória" que são, essencialmente, a mesma coisa que discos de RAM de PCs. Um driver especial atua como um disco, mas na verdade armazena dados em uma memória de alta velocidade. Muitas instalações utilizam um disco de RAM para seus sistemas de arquivos **/tmp** ou para outros arquivos de alta atividade como logs de servidor Web ou spools de e-mails. Utilizar um disco de RAM pode reduzir a memória disponível para uso geral, mas torna incrivelmente mais rápidas a leitura e a gravação de arquivos temporários. Geralmente é um bom negócio.

A maioria dos kernels Linux foi compilada com suporte a disco de RAM. Em sistemas que suportam discos de RAM, o diretório **/dev** contém vários arquivos de dispositivos de disco de RAM como **/dev/ram0** e **/dev/ram1**. O número de dispositivos pode variar, mas normalmente são pelo menos cinco.

Para usar um disco de RAM, primeiramente formate o sistema de arquivos em um dispositivo de RAM não utilizado, então o monte em um diretório existente:

```
# mke2fs /dev/ram12
# mount /dev/ram12 /tmp/fastdisk
```

O tamanho de RAM padrão é somente de 4 MB, o que realmente não é grande o suficiente para **/tmp**. Infelizmente, o procedimento para alterar o tamanho é um pouco complicado, pois ele é configurado na forma de uma variável de kernel. Você poderá adicionar uma linha como `ramdisk_size=100000` na configuração da rotina de inicialização para passar o novo valor no momento do boot (o tamanho é dado em blocos de 1 KB) ou configurar o driver de disco de RAM como um módulo que pode ser carregado dinamicamente. Neste último caso, você adicionaria o argumento necessário ao arquivo **/etc/modules.conf** ou o passaria na forma de um argumento para **insmod**.

Alguns programas degradam o desempenho do sistema retardando operações básicas. Dois exemplos são cotas de disco e contabilidade de CPU. As cotas requerem que um sumário de uso de disco seja atualizado à medida que os arquivos são gravados e apagados. A contabilidade de CPU grava um registro de dados em um arquivo de contabilidade cada vez que um processo é completado. O cache em disco ajuda a minimizar o impacto desses recursos, mas eles ainda podem ter um ligeiro efeito sobre o desempenho e não devem ser ativados a menos que você realmente os use.

Escolhendo um agendador de E/S

Um algoritmo de agendamento de E/S atua como um árbitro entre processos que competem por E/S de disco. Ele otimiza a ordem e a sincronização de solicitações para fornecer o melhor desempenho geral de E/S possível.

Quatro algoritmos de agendamento diferentes estão integrados no kernel 2.6. Você pode fazer sua escolha. Infelizmente, o algoritmo de agendamento é configurado pelo argumento de kernel **elevator**=*algoritmo* e, então, só pode ser configurado em tempo de inicialização. Se estiver normalmente configurado no arquivo **grub.conf**.

Os algoritmos disponíveis são:

- Completely Fair Queuing (**elevator=cfq**): esse é o algoritmo padrão e, normalmente, é a melhor escolha para servidores de uso geral. Ele tenta distribuir igualmente o acesso a largura de banda E/S.
- Deadline (**elevator=deadline**): esse algoritmo tenta minimizar latência para cada requisição. O algoritmo reordena requisições para aumentar o desempenho.
- NOOP (**elevator=noop**): esse algoritmo implementa uma fila simples FIFO. Ele assume que requisições de E/S são otimizadas ou reordenadas pelo driver ou dispositivo (como talvez seja feito por um controlador inteligente). Essa opção pode ser a melhor escolha em alguns ambientes SAN.
- Anticipatory (**elevator=as**): esse algoritmo retarda requisições de E/S na esperança de ordená-las para eficiência máxima. Esse talvez seja uma boa escolha para uma estação de trabalho desktop, mas, raramente, seria ótima para um servidor.

Determinando que algoritmo de agendamento é mais apropriado para seu ambiente — e você pode precisar tentar os quatro — você pode ser capaz de afetar diretamente o desempenho de E/S de disco.

sar: coletando dados e gerando relatórios estatísticos ao longo do tempo

Uma ferramenta de monitoração de desempenho que se tornou comum em sistemas Linux é o **sar**. Esse comando tem suas raízes no antigo AT&T UNIX, mas foi completamente reescrito para utilização sob Linux.

À primeira vista, o **sar** parece exibir quase as mesmas informações que **procinfo**, **vmstat** e **iostat**. Entretanto, há uma diferença importante: o **sar** pode informar dados históricos bem como os atuais.

Sem opções, o comando **sar** informa a utilização de CPU em intervalos de 10 minutos durante o dia desde a meia-noite, como mostrado abaixo. Essa coleta de dados históricos torna-se possível pelo script **sal**, que faz parte do pacote **sar** (às vezes chamado de **sysstat**) e deve ser configurado para executar a partir do **cron** em intervalos periódicos. O **sar** armazena os dados que ele coleta em **/var/log/sa** em um formato binário.

```
$ sar
Linux 2.6.9-11.ELsmp (bull.atrust.com)  08/04/2006

12:00:01 AM   CPU   %user   %nice   %system   %iowait   %idle
12:10:01 AM   all    0.10    0.00      0.04      0.06   99.81
12:20:01 AM   all    0.04    0.00      0.03      0.05   99.88
12:30:01 AM   all    0.04    0.00      0.03      0.04   99.89
12:40:01 AM   all    0.09    0.00      0.03      0.05   99.83
12:50:01 AM   all    0.04    0.00      0.03      0.04   99.88
01:00:01 AM   all    0.05    0.00      0.03      0.04   99.88
```

Além das informações de CPU, o **sar** também pode informar a métrica, por exemplo, a atividade de disco e de rede. Utilize **sar -d** para um resumo da atividade de disco de hoje ou **sar -n DEV** para estatística de interface de rede. **sar -A** relata todas as informações disponíveis.

Consulte a página 460 para um exemplo de gráfico do Cacti.

O **sar** tem algumas limitações, mas é uma boa aposta para informações históricas rápidas e simples. Se você estiver seriamente interessado em monitoração de desempenho, sugerimos que você configure uma plataforma de coleta de dados e de representação gráfica como o Cacti. O Cacti provém do mundo de gerenciamento de rede, mas pode realmente representar graficamente métricas de sistema arbitrárias como informações da CPU e da memória.

oprofile: profiler abrangente

oprofile é um profiler de sistema integrado incrivelmente poderoso para sistemas Linux que executam no kernel 2.6 ou versão superior. É possível gerar um perfil de todos os componentes de um sistema Linux: rotinas de tratamento de interrupção de hardware e software, módulos de kernel, o próprio kernel, bibliotecas compartilhadas e aplicativos.

Se você tiver bastante tempo extra e quiser saber exatamente como os recursos do sistema estão sendo utilizados (até o menor nível de detalhe), considere a execução do **oprofile**. Essa ferramenta é particularmente útil se você estiver desenvolvendo internamente seus próprios aplicativos ou código de kernel.

Um módulo de kernel e um conjunto de ferramentas no nível do usuário estão incluídos na distribuição **oprofile**, que está disponível para cópia em oprofile.sourceforge.net.

No Red Hat, você também precisa instalar o pacote **kernel-debuginfo** para obter os arquivos necessários para executar **oprofile**.

25.4 SOCORRO! MEU SISTEMA ESTÁ MUITO LENTO!

Em seções anteriores falamos principalmente sobre questões relacionadas ao desempenho médio de um sistema. Soluções para esses problemas de longo prazo geralmente assumem a forma de ajustes de configuração ou de atualizações.

Entretanto, você descobrirá que mesmo sistemas configurados apropriadamente tornam-se às vezes mais lentos do que o usual. Felizmente, problemas transitórios normalmente são fáceis de diagnosticar. Noventa por cento das vezes, eles são provocados por um processo voraz que simplesmente está consumindo tanto poder de processamento, espaço em disco ou largura de banda de rede que os demais processos ficam emperrados. Ocasionalmente, processos mal-intencionados consomem vorazmente recursos disponíveis para intencionalmente deixar lento um sistema ou uma rede, um esquema conhecido como ataque de negação de serviço (*denial of service* – DOS).

Normalmente você pode informar qual recurso está sendo consumido vorazmente mesmo sem executar um comando de diagnóstico. Se o sistema parecer "emperrado" ou você perceber um acesso anormal a disco, o problema mais provável é falta de memória ou espaço em disco.[7] Caso o sistema pareça "emperrado" (tudo demora e os aplicativos não carregam mais rápido após a primeira execução), o problema pode ser a CPU.

A primeira etapa no diagnóstico é executar **ps auxww** ou **top** para procurar processos descontrolados óbvios. Qualquer processo que estiver utilizando mais de 50% da CPU provavelmente deve estar com problemas. Se nenhum processo estiver consumindo individualmente uma parcela excessiva de CPU, verifique quantos processos estão utilizando pelo menos 10% dos recursos. Se houver mais do que dois ou três (não conte o próprio **ps**), a carga média provavelmente deve estar bem alta. Por si só, esse problema é uma causa de desempenho sofrível. Verifique a carga média com **uptime** e use **vmstat** ou **top** para ver se a CPU alguma vez torna-se ociosa.

Se nenhuma disputa por CPU for evidente, execute **vmstat** ou **procinfo** para ver quanta paginação está ocorrendo. Toda atividade em disco é interessante: um número grande de paginação para o disco pode indicar disputa de memória; e tráfego de disco na ausência de paginação pode significar que um processo está monopolizando o disco fazendo constantes leituras e gravações de arquivos.

Não há como associar operações de discos a processos, mas **ps** pode reduzir o número de possíveis suspeitos para você. Qualquer processo que estiver gerando tráfego em disco tem de estar utilizando alguma quantidade de tempo da CPU. Normalmente é possível fazer uma estimativa fundamentada sobre qual dos processos ativos é o verdadeiro culpado.[8] Use **kill -STOP** para suspender o processo e testar sua teoria.

---

7. Isto é, demora alternar entre aplicativos, mas o desempenho é aceitável quando um aplicativo está repetindo uma tarefa simples.
8. Um grande espaço de endereços virtual ou um grande conjunto residente costumavam ser um sinal de suspeita, mas bibliotecas compartilhadas tornaram esses números menos úteis. **ps** não é muito inteligente em fazer uma separação entre o tempo gasto com as bibliotecas de sistema compartilhadas e o espaço de endereços de cada processo individual. Muitos processos parecem ter erroneamente dezenas de megabytes de memória ativa.

Suponha que você ache que um determinado processo esteja com problemas — o que fazer? Em geral, nada. Algumas operações simplesmente requerem tantos recursos que são propensas a tornar o sistema lento. Isso não significa necessariamente que elas são ilegítimas. Em geral, é aceitável aplicar **renice** a um processo incômodo que é limitado pelos recursos da CPU. Mas certifique-se de pedir ao proprietário para utilizar o comando **nice** no futuro. Algumas vezes, o ajuste fino de aplicativos pode reduzir significativamente a demanda de um programa por recursos de CPU; esse efeito é especialmente visível com software de servidor de rede personalizado como os aplicativos Web.

Processos que são consumidores vorazes de disco ou de memória não são tão fáceis de tratar. **renice** geralmente não ajuda. Você tem a opção de eliminar ou interromper o processo, mas não recomendamos esse tipo de procedimento caso a situação não seja uma emergência. Como com os consumidores vorazes de recursos de CPU, você pode usar a solução rudimentar de solicitar ao dono do processo para executá-lo mais tarde.

O kernel permite que um processo restrinja seu próprio uso de memória física com a chamada de sistema **setrlimit**. Essa facilidade também se encontra disponível no shell C por meio do comando embutido **limit**. Por exemplo, o comando:

```
% limit memoryuse 32m
```

faz com que todos os comandos subseqüentes que o usuário executar tenham sua memória física limitada a 32 megabytes. Esse recurso é grosseiramente equivalente ao comando **renice** para processos limitados pela memória. Você poderia sugerir diplomaticamente que transgressores contumazes coloquem uma linha como essa em seus arquivos **.cshrc**.

Se um processo descontrolado não parece ser a fonte de mau desempenho, investigue duas outras possíveis causas. A primeira delas é uma rede sobrecarregada. Muitos programas estão tão intimamente vinculados à rede que é difícil dizer onde termina o desempenho do sistema e onde começa o desempenho da rede. Veja o Capítulo 19 para informações adicionais sobre as ferramentas utilizadas para monitorar redes.

Alguns problemas de rede são difíceis de diagnosticar, pois vêm e vão rapidamente. Por exemplo, se cada máquina da rede executar um programa relacionado com rede e baseado em **cron** num determinado momento do dia, normalmente haverá uma breve, mas significativa disfunção. Toda máquina da rede ficará paralisada por cinco segundos e depois o problema desaparecerá tão rapidamente quanto surgiu.

Atrasos relacionados com servidores é uma outra possível causa de crises de desempenho. Sistemas Linux estão constantemente consultando servidores remotos em busca de NFS, NIS, DNS e uma série de outros recursos. Se um servidor não responder ou algum outro problema tornar difícil à comunicação com ele, os efeitos podem se propagar para os sistemas clientes.

Por exemplo, em um sistema atarefado, um processo poderia utilizar a rotina de biblioteca **gethostent** a cada poucos segundos. Se uma disfunção de DNS fizer essa rotina levar dois segundos para se completar, é provável que você perceba uma diferença no desempenho geral. Problemas de configuração de pesquisa de DNS direta e inversa são responsáveis por um número inesperado de questões de desempenho de servidor.

25.5 LEITURA RECOMENDADA

EZOLT, PHILLIP G. *Optimizing Linux Performance.* Upper Saddle River, NJ: Prentice Hall PTR, 2005.

JOHNSON, S., ET AL. *Performance Tuning for Linux Servers.* Indianapolis, IN: IBM Press, 2005.

LOUKIDES, MIKE E MUSUMECI, GIAN-PAOLO D. *System Performance Tuning. 2. ed.* Sebastopol: O'Reilly & Associates, 2002.

25.6 EXERCÍCIOS

E25.1 Faça uma estimativa fundamentada sobre o problema exposto em cada um dos casos a seguir:

a) Ao alternar entre aplicativos, o disco é utilizado excessivamente e há um retardo perceptível.

b) Um programa de simulação numérica leva mais tempo do que o normal, mas a memória do sistema está em grande parte livre.

c) Usuários em uma rede local muito ativa reclamam de acesso NFS lento, mas a média de carga no servidor está muito baixa.

d) A execução de um comando (qualquer comando) normalmente indica "memória insuficiente".

E25.2 Execute **procinfo** em uma máquina Linux disponível e discuta os resultados obtidos. Quais, se algum, recursos parecem estar sendo utilizados pesadamente? Que recursos parecem não estar sendo utilizados? Inclua a listagem IRQ em sua análise.

★ E25.3 O balanceamento de carga pode ter um impacto significativo no desempenho do servidor a partir da perspectiva do mundo externo. Discuta vários mecanismos que poderiam ser utilizados para realizar o balanceamento de carga.

★ E25.4 Liste os quatro principais recursos que afetam o desempenho. Para cada recurso, dê um exemplo de um aplicativo que poderia facilmente levar à exaustão do recurso em questão. Discuta maneiras de aliviar parte do esforço associado a cada cenário.

★ E25.5 Utilizando a Web e/ou as páginas man, procure o comando **hdparm**. Que opções ele oferece para testar velocidades de acesso a disco? Como **hdparm** pode aumentar as velocidades de acesso a disco em alguns casos?

★ ★ E25.6 Selecione dois programas que utilizem uma quantidade notável de recursos de sistema. Use **vmstat** e as demais ferramentas mencionadas no capítulo para dar um perfil dos dois aplicativos. Faça uma argumentação sobre o que cada programa faz que o torna um voraz consumidor de recursos. Baseie suas afirmações em dados.

26 Cooperando com o Windows

É bem provável que seu ambiente inclua tanto sistemas Microsoft Windows como Linux. Se esse for o caso, há muitas maneiras de esses sistemas operacionais ajudarem um ao outro. Entre outras façanhas, aplicativos Windows podem executar a partir de um desktop Linux ou acessar impressoras e arquivos de um servidor Linux. Os aplicativos Linux podem exibir suas interfaces com usuário em um desktop Windows.

As duas plataformas têm suas forças e podem ser colocadas para trabalhar juntas. O Windows é uma plataforma desktop popular e repleta de recursos, capaz de preencher a lacuna entre o usuário e o cabo de rede que sai da parede. Por outro lado, o Linux é uma plataforma de infra-estrutura confiável e expansível. Portanto, sem brigas, OK?

26.1 Conectando-se ao sistema Linux a partir do Windows

Consulte a página 481 para obter informações adicionais sobre SSH.

Usuários podem muitas vezes se achar em uma situação em que se dirigem para as encostas cobertas de neve de uma boa sessão C ou de uma sessão **bash** sem abandonar a máquina Windows nas suas escrivaninhas. Da perspectiva do cliente, a maneira mais fácil de realizar isso é utilizar o programa **telnet** que a Microsoft distribui com o Windows. Hoje em dia, porém, na maioria dos sistemas Linux os servidores **telnet** (prudentemente) permanecem desativados por questões de segurança. As pessoas que querem se conectar precisam de um emulador de terminal que entenda SSH.

Há várias implementações SSH para Windows. Nossa implementação atual favorita, PuTTY, de código- fonte aberto, é simples e efetiva. O PuTTY suporta vários recursos que você esperava de um aplicativo terminal nativo como o **xterm**. A única reclamação que ouvimos em relação ao PuTTY é que ele não trata caracteres de múltiplos bytes corretamente; usuários asiáticos podem se beneficiar da combinação do Tera Term Pro com o plug-in TTSSH, embora essa configuração suporte apenas a versão 1 do protocolo SSH. Você pode encontrar o PuTTY aqui:

www.chiark.greenend.org.uk/~sgtatham/putty

e o Tera Term Pro aqui:

hp.vector.co.jp/authors/VA002416/teraterm.html

SSH também suporta transferência de arquivo, e o PuTTY inclui dois clientes de linha de comando para esse propósito: **psftp** e **pscp**. Usuários Windows "hard-core" do tipo "nunca toquei em uma linha de comando" talvez prefiram o cliente WinSCP gráfico em winscp.sf.net.

Uma outra opção plausível é instalar o pacote Cygwin "UNIX-on-Windows" mais geral para executar seus utilitários SSH a partir do **rxvt**. Há informações adicionais sobre o Cygwin a partir da página 567.

Uma nova implementação Java do SSH chamada MindTerm está disponível em AppGate (www.appgate.com). Ela é gratuita para uso pessoal. Essa implementação executa em qualquer sistema que suporta Java e pode ser configurada de várias maneiras.

Um recurso interessante do SSH é sua capacidade de encaminhar portas TCP de um lado para o outro entre o cliente e o servidor. Por exemplo, esse recurso permite configurar no cliente uma porta local que encaminha a conexão de entrada a uma porta diferente em uma máquina só acessível a partir do servidor. Embora esse recurso abra um mundo de novas possibilidades, ele também é potencialmente perigoso e algo de que você precisa estar ciente ao conceder acesso SSH ao seu servidor. Felizmente, o recurso de encaminhamento de porta pode ser desativado no lado servidor a fim de limitar SSH ao acesso de terminal e à transferência de arquivos.

26.2 ACESSANDO DESKTOPS REMOTOS

Desktops gráficos no Linux estão associados ao X Window System gratuito, que não está de nenhuma maneira relacionado ao Microsoft Windows. O X Windows foi desenvolvido no MIT em meados da década de 1980 e foi adotado como um padrão por todos os fabricantes de estações de trabalho UNIX e distribuições Linux. Ele passou por várias atualizações importantes, mas uma base estável foi alcançada com a versão 11, primeiramente publicada no início dos anos 90. O número da versão do protocolo X foi acrescentado ao X para formar o X11, o nome pelo qual ele é mais comumente conhecido. (O nome "Windows" por si só sempre se refere ao Microsoft Windows, tanto neste capítulo como no mundo real.)

O DNS é um sistema cliente/servidor. O servidor X é responsável por exibir os dados na tela do usuário e por adquirir a entrada de teclado e de mouse do usuário. Ele se comunica com aplicativos clientes pela rede. O servidor e os clientes não precisam estar em execução na mesma máquina.

Uma discussão mais detalhada da arquitetura do X Windows pode ser encontrada no Capítulo 22, *O X Window System*, a partir da página 511.

Executando um servidor X em um computador Windows

O X11 é um protocolo rico que incorporou muitas extensões ao longo dos anos. A implementação de um servidor X é, portanto, bastante complexa. Contudo, há agora implementações do servidor X para quase todos os sistemas operacionais. O próprio X Windows é agnóstico quanto ao sistema operacional, portanto, clientes X11 que executam em uma máquina Linux podem ser exibidos em um servidor X executando em Microsoft Windows e permitir que um usuário os controle como se esse usuário estivesse em frente ao console de sistema.

Infelizmente, os projetistas originais dos protocolos X não deram muita atenção à segurança. Cada programa que se conecta ao seu servidor X pode ler tudo que você digita no teclado e ver tudo exibido na sua tela. Para piorar as coisas, os programas remotos nem mesmo têm de exibir uma janela ao acessar seu servidor X; eles simplesmente podem mover-se furtivamente no segundo plano.

Vários métodos para proteger o X11 foram propostos ao longo dos anos, mas todos tendem a ser um pouco complexos. A questão é que seria melhor bloquear todas as conexões remotas com o servidor X a menos que você esteja absolutamente seguro do que está fazendo. A maioria dos servidores X é configurada para recusar conexões remotas por padrão, então você deve estar seguro contanto que não execute o programa **xhosts** (ou seu equivalente) para conceder acesso remoto.

Consulte a página 481 para informações adicionais sobre SSH.

Infelizmente, conceder acesso remoto é exatamente o que você precisa fazer quando busca executar programas no Linux e exibir suas interfaces no Windows. Portanto, como executar um aplicativo remoto sem conceder acesso remoto ao servidor X? O método mais comum é utilizar um recurso do protocolo SSH que é especificamente projetado para suportar o X11. Esse esquema cria um túnel seguro entre clientes X em execução no host remoto e no servidor X local. Programas iniciados no host remoto serão automaticamente exibidos na máquina local, mas, por meio da mágica do SSH, o servidor X local irá percebê-los como se fossem originados localmente.

Observe que o encaminhamento X só funcionará se os recursos X de encaminhamento foram ativados tanto no servidor SSH como no cliente SSH. Se você utilizar o cliente PuTTY SSH no Windows, simplesmente ative o recurso de encaminhamento X11 na tela de configuração. No lado servidor SSH (isto é, o lado cliente X11; a máquina Linux), certifique-se de que o arquivo **/etc/ssh/ sshd_config** contém a linha

```
X11Forwarding yes
```

Se modificar a configuração do servidor SSH, certifique-se de reiniciar o processo **sshd** para ativar a nova configuração.

Embora a Apple forneça um servidor X gratuito para o Mac OS X, não há infelizmente nenhuma oferta dessas por parte da Microsoft. Há, entretanto, um servidor X gratuito disponível a partir do projeto Cygwin (cygwin.com) que funciona muito bem depois de configurado. Infelizmente, a configuração pode ser um desafio para as pessoas que não sabem configurar um servidor X no Linux. Servidores X comerciais para o Windows incluem o eXeed e o X-Win32. Estes oferecem uma configuração muito mais simples a um preço um pouco mais alto.

VNC: Virtual Network Computing

No final da década de 1990, algumas pessoas nos AT&T Labs em Cambridge, Reino Unido, desenvolveram um sistema para acesso remoto de desktop chamado VNC. A idéia era casar a simplicidade de um terminal "burro" com o mundo moderno dos sistemas de janelas. Comparado com o X11, o protocolo VNC não lida com aplicativos individuais. Em vez disso, ele cria uma desktop virtual completo (ou fornece acesso remoto a um desktop existente) como uma unidade. Sob o VNC, um servidor X11 especial executa na máquina central e um aplicativo visualizador é utilizado para acessar esse servidor.

A AT&T publicou o software VNC sob uma licença de código-fonte aberto. Isso permitiu que outras pessoas aderissem a essa onda e criassem mais implementações de servidores e visualizadores adicionais, e aprimorassem o protocolo para

reduzir a largura de banda. Hoje, há visualizadores VNC disponíveis para a maioria dos dispositivos que fornece algum meio de exibição gráfica. Os servidores VNC para UNIX/Linux e Windows estão amplamente disponíveis.

A implementação do servidor VNC Linux é essencialmente um emulador de adaptador de imagens gráficas que é conectado ao servidor X.Org X Windows. Executar um **vncserver** a partir da sua conta Linux cria um novo desktop virtual que executa no mundo autocontido da máquina Linux. Você então pode utilizar um visualizador VNC para acessar esse desktop remotamente.

O protocolo VNC é baseado em mapa de bits e não tem informações de estado. Portanto, os visualizadores podem se conectar e desconectar livremente. É possível até que vários visualizadores acessem o mesmo servidor VNC ao mesmo tempo. Esse último recurso é especialmente útil para suporte remoto e configurações de teste. Ele também facilita o acesso compartilhado via console à administração de sistemas.

Os servidores VNC no mundo Windows normalmente não criam um desktop extra; eles simplesmente exportam o desktop Windows padrão como ele é exibido na tela. A principal aplicação dessa tecnologia é o suporte remoto.

Atualmente, os autores originais do VNC administram uma empresa própria, RealVNC (realvnc.com). O projeto Ultr@VNC (ultravnc.sf.net) se concentra no domínio Windows com uma implementação do servidor VNC baseada no Windows muito rápida e cheia de recursos, e a TightVNC (tightvnc.com) está trabalhando para aprimorar as taxas de compactação. Esses grupos conversam entre si, assim há uma tendência de os recursos serem utilizados entre as várias implementações.

O protocolo VNC foi projetado com a extensibilidade em mente. Todas as combinações de visualizadores e servidores podem trabalhar juntas; elas selecionam a melhor variante do protocolo que ambos os lados entendem. Recursos específicos à implementação (como transferência de arquivos) normalmente só podem ser acessados executando um servidor e um cliente no mesmo projeto.

Windows RDP: Remote Desktop Protocol

A partir do Windows 2000 Server, cada máquina Windows tem a capacidade técnica de fornecer acesso remoto gráfico a vários usuários ao mesmo tempo. O componente de acesso remoto é chamado Remote Desktop e utiliza um protocolo chamado Remote Desktop Protocol (RDP) para a comunicação entre o cliente e o servidor.

Com um hardware de PC decente, você pode facilmente executar 50 sessões concorrentes do Microsoft Office em um servidor Windows 2003. Devido a restrições de licença, essa maravilhosa capacidade só é utilizável para uma única sessão no Windows XP. Se você se conectar a uma máquina remota, você será desconectado localmente.

Se tiver uma instalação do Windows Server, você poderá ativar o serviço Terminal Server e utilizar o programa **rdesktop** de código-fonte aberto (www.rdesktop.org) para acessar um desktop Windows diretamente da sua estação de trabalho Linux. (Há também clientes para Windows, Mac OS e Windows CE diretamente da Microsoft.) Como o VNC, o protocolo RDP tem informações de estado. Portanto, é possível reconectar-se a uma sessão desconectada ou realizar suporte remoto.

O RDP permite até mesmo que recursos de uma impressora local e recursos de disco sejam encaminhados à sessão Windows remota. Teste a configuração e se divirta — só a instale depois de ter a licença correta. O Terminal Server emitirá uma licença de teste de 30 dias a quem se conectar a ele, mas, depois de 30 dias, ele não permitirá mais conexões a partir desse cliente a menos que você tenha comprado uma Terminal Server Client Access License apropriada.

Todo o conceito do acesso remoto de uma máquina Windows parece confundir os departamentos de licenciamento da Microsoft e da maioria dos outros fornecedores de software Windows. As condições da licença continuam mudando enquanto tentam chegar a uma abordagem conclusiva, portanto certifique-se de obter a palavra oficial sobre o preço atual antes de deixar seus usuários com água na boca. O esquema de licenciamento que gostaríamos de ver (licença por número de usuários concorrentes) não parece estar na agenda de ninguém. Atualmente, a escolha é entre comprar licenças para cada computador ou para cada usuário individual que acessará o servidor.

26.3 Executando o Windows e aplicativos do tipo Windows

O produto comercial VMware (vmware.com) permite executar múltiplos sistemas operacionais simultaneamente no hardware de PC. O VMware emula "máquinas convidadas" virtuais completas na parte superior de um sistema operacional hospedeiro, que deve ser Linux ou Windows. Independentemente do sistema operacional hospedeiro, você pode instalar a maioria dos sistemas operacionais compatíveis com Intel em uma das máquinas virtuais do VMware.

O VMware é uma ferramenta ideal para ambientes de desenvolvimento ou teste. Ele vem até com um recurso que estabelece uma rede com seus sistemas operacionais virtuais de modo que eles possam se comunicar entre si e compartilhar a conexão Ethernet física do sistema.

Uma abordagem diferente é adotada pelo sistema Wine, em winehq.org. O Wine implementa a API de programação Windows no ambiente Linux, permitindo executar aplicativos Windows diretamente na parte superior do Linux e do X.

Esse software gratuito converte chamadas nativas da API do Windows em suas contrapartes Linux e pode fazer isso sem usar nenhum código da Microsoft. O Wine oferece suporte para TCP/IP, dispositivos seriais e saída de som.

O projeto Wine foi iniciado em 1993 por Bob Amstadt. Desde então o Wine percorreu um longo caminho. Um grande número de aplicativos Windows executa sem problemas e outros podem funcionar com alguns truques; acesse o site Web para detalhes. Infelizmente, costuma não ser tão simples fazer com que um aplicativo execute sob o Wine. As talentosas pessoas no codeweavers.com escreveram um sistema instalador comercial que é capaz de fazer alguns aplicativos Windows mais falhos funcionar corretamente.

Se sua ferramenta preferida for suportada pelo CodeWeavers, excelente. Mas, mesmo se não for, dê uma chance ao aplicativo — talvez você tenha uma surpresa agradável. Mas se o aplicativo não funcionar por conta própria e você não puder encontrar nenhuma dica já publicada, esteja preparado para investir muitas horas vagas se você estiver determinado a colocar esse aplicativo em funcionamento. Se seu orçamento permitir, você poderá pensar em contratar a CodeWeavers para ajudá-lo.

Win4Lin é uma alternativa comercial ao Wine da NeTraverse. O Win4Lin afirma ser mais estável do que o Wine e de suportar alguns aplicativos Windows a mais. Entretanto, ele requer modificações no kernel que o Wine não requer. Win4Lin está disponível em netraverse.com.

Inicialização dupla ou por que você não deve fazer isso

Se nunca instalou o Linux em um computador que já era utilizado como uma máquina Windows, sem dúvida você tem a opção de especificar uma configuração de inicialização dupla. Hoje em dia, essas configurações funcionam muito como o prometido. É até mesmo possível montar partições Windows sob o Linux e acessar o sistema de arquivos Linux sob o Windows. Leia tudo sobre como ajustar uma configuração de inicialização dupla na página 21.

Mas espere! Se você estiver fazendo trabalho real e precisar de acesso tanto ao Windows como ao Linux, seja muito séptico a respeito da inicialização dupla como possível solução. Configurações de inicialização dupla representam a Lei de Murphy no seu pior aspecto: elas sempre parecem ser inicializadas no sistema operacional errado, e a tarefa mais tranqüila em geral exige várias reinicializações. Computadores são tão baratos e o acesso remoto tão fácil que normalmente não há razões de passar por essa interminável tortura.

A alternativa OpenOffice.org

Há alguns anos, a Sun distribuiu uma versão de código-aberto do StarOffice, o seu conjunto de aplicativos do tipo Microsoft Office, sob o nome OpenOffice.org. O Openoffice.org inclui uma planilha, um processador de texto, um pacote de apresentação e um aplicativo de desenho. Essas ferramentas podem ler e gravar arquivos gerados pelos seus análogos Microsoft. Você pode fazer a cópia desse conjunto a partir do openoffice.org.

O openoffice.org está disponível nas principais plataformas, incluindo Windows, Linux, Solaris e a maioria da outras versões do UNIX. Há também uma versão para o Mac OS X. Se estiver procurando um pacote com um contrato de suporte comercial, você também pode comprar o StarOffice da Sun, que é essencialmente o Openoffice.org em uma máquina com melhor verificação ortográfica e um banco de dados.

26.4 Utilizando ferramentas de linha de comando com o Windows

O que muitas pessoas sentem falta no Linux ao trabalhar em sistemas Windows é o querido **xterm**. Não apenas qualquer aplicativo terminal antigo ou a abominação conhecida como modo DOS, mas um adequado **xterm** com suporte a redimensionamento, cores, controle de mouse e todas as sofisticadas seqüências de escape **xterm**.

Embora não haja nenhuma porta nativa independente (isto é, sem o X) do **xterm** para o Windows, um pequeno programa elegante chamado **rxvt** chega bem próximo. Ele faz parte do sistema Cygwin, cujo download pode ser feito em cygwin.com. Se instalar o servidor X do Cygwin, você poderá utilizar o **xterm** real.

Esse sistema, distribuído sob a GNU General Public License, contém um extenso complemento de comandos UNIX comuns e também uma biblioteca para portar código que implementa as APIs POSIX sob o Windows. A maneira de o Cygwin reconciliar as convenções de linha de comandos e sistema de arquivos do UNIX e do Windows é bem bolada e é capaz de trazer vários itens básicos de um shell UNIX para comandos Windows nativos. Além de fazer com que os usuários Linux se sintam em casa, a Cygwin facilita a execução de software Linux e UNIX sob o Windows. Visite cygwin.com para detalhes.

O MKS Toolkit é uma alternativa comercial ao Cygwin. Veja o site da MKS em www.mkssoftware.com para mais informações.

Uma lista crescente de softwares UNIX agora também executa nativamente no Windows, incluindo Apache, Perl, BIND, PHP, MySQL, Vim, Emacs, Gimp, Wireshark e Python.

26.5 CONFORMIDADE DO WINDOWS COM PADRÕES DE E-MAIL E DA WEB

Em um mundo ideal, todos usariam padrões de código-fonte aberto para comunicação e todos seriam muito felizes. Mas esse não é um mundo ideal e várias pessoas acusam o Windows de ser uma confusão de protocolos proprietários e implementações desconexas dos padrões Internet. Isso talvez seja parcialmente verdadeiro, contudo há muitas áreas em que o Windows pode desempenhar um papel muito bom no mundo dos padrões. Duas dessas áreas são os serviços de e-mail e os serviços Web.

Na impetuosa história da Web, várias corporações tentaram abraçar e estender a Web de maneira que permitissem que elas tivessem uma vantagem sobre a concorrência e dessem um poderoso impulso aos seus próprios negócios. A Microsoft continua empenhada nessa batalha no nível de navegador com suas inúmeras extensões peculiares ao Internet Explorer. Mas, no nível subjacente do protocolo HTTP, o Windows e os navegadores Windows são relativamente agnósticos quanto à plataforma.

A Microsoft fornece seu próprio servidor Web, IIS, mas, historicamente, o desempenho desse servidor está bem abaixo daquele do Apache em execução no Linux. A menos que você esteja preso a uma tecnologia do lado servidor, como o ASP, não há nenhuma razão para utilizar máquinas Windows como servidores Web. Como clientes Web, porém, elas irão absorver sem problemas todo o conteúdo Web que os seus servidores UNIX e Linux podem servir.

Para e-mail, a Microsoft promove seu produto Exchange Server como a tecnologia do lado servidor preferida. Verdade seja dita: as capacidades do Exchange Server superam aquelas dos sistemas de correio padrão da Internet, particularmente quando os clientes de correio consistem em máquinas Windows em execução no Microsoft Outlook. Mas não se preocupe: o Exchange Server também pode conversar com o SMTP para correio interno e externo e ele pode servir correio para clientes Linux, utilizando os protocolos padrão IMAP e POP. No lado cliente, tanto o Outlook como o seu irmão mais jovem gratuito, o Outlook Express, podem se conectar a servidores IMAP e POP. Mescle e combine como você preferir. Mais informações sobre o POP e o IMAP, a partir da página 376.

26.6 COMPARTILHANDO ARQUIVOS COM SAMBA E CIFS

No começo da década de 1980, a IBM projetou uma API que permitia que computadores na mesma sub-rede da rede conversassem um com o outro utilizando nomes em vez de números obscuros. O resultado foi o Network Basic/Input Output System, ou NetBIOS. A combinação do NetBIOS e seu original protocolo de transporte de rede subjacente foi chamado NetBIOS Extended User Interface, ou NetBEUI. A API do NetBIOS tornou-se bem popular e foi adaptada para uso na parte superior de vários diferentes protocolos de rede como IPX, DECNet e TCP/IP.

A Microsoft e a Intel desenvolveram um protocolo de compartilhamento de arquivos sobre o NetBIOS e o chamaram "protocolo básico". Mais tarde, ele foi renomeado como o protocolo Server Message Block, ou SMB. Uma evolução posterior do protocolo SMB conhecida como Common Internet File System (CIFS) é essencialmente uma versão SMB que foi aprimorada e ajustada para operação em redes remotas. O CIFS é a atual língua franca do compartilhamento de arquivos Windows.

No mundo Windows, um sistema de arquivos ou diretório disponibilizado na rede é conhecido como "compartilhamento". Soa um pouco estranho aos ouvidos UNIX, mas seguimos essa convenção ao fazer referência aos sistemas de arquivos CIFS.

Samba: servidor CIFS para UNIX

O Samba é um pacote de software tremendamente popular, disponível sob a licença pública GNU, que implementa o lado servidor do CIFS em hosts Linux. Ele foi criado originalmente por Andrew Tridgell, um australiano, que aplicou engenharia reversa sobre o protocolo SMB e publicou o código resultante em 1992.

Hoje o Samba tem bom suporte e está sendo atualizado ativamente para expandir sua funcionalidade. Ele fornece um mecanismo estável para integrar máquinas Windows em uma rede Linux. A verdadeira beleza do Samba é que você só precisa instalar um pacote no servidor; não é necessário nenhum software especial no lado Windows.[1]

O CIFS fornece cinco serviços básicos:

- Compartilhamento de arquivos
- Impressão em rede
- Autenticação e autorização
- Conversão de nomes
- Anúncio de serviços ("navegação" de impressoras e de servidor de arquivos)

1 A máquina Windows deve ter sido configurada como "rede Microsoft".

O Samba não só serve arquivos por meio do CIFS, como também pode realizar as funções básicas de um controlador de domínio primário Windows.[2] O Samba suporta alguns recursos avançados, incluindo logins de domínio Windows, perfis do usuário Windows móveis e spool de impressão CIFS.

A maior parte da funcionalidade do Samba é implementada por dois daemons: **smbd** e **nmbd**. O **smbd** implementa serviços de arquivo e serviços de impressão, bem como autenticação e autorização. O **nmbd** fornece os outros principais componentes CIFS: conversão de nomes e anúncio de serviços.

Diferentemente do NFS, que requer suporte no nível do kernel, o Samba não exige nenhuma modificação no driver do kernel e executa inteiramente como um processo de usuário. Ele se vincula aos sockets usados para requisições CIFS e aguarda que um cliente requisite acesso a um recurso. Assim que a requisição foi autenticada, o **smbd** cria uma instância de si mesmo que é executada como o usuário que está fazendo as requisições. Conseqüentemente, todas as permissões de acesso a arquivos Linux (inclusive permissões de grupos) são obedecidas. A única funcionalidade especial que o **smbd** acrescenta a isto é um serviço de bloqueio a arquivos que dá aos PCs clientes a semântica de bloqueio com a qual eles estão acostumados.

Instalação do Samba

O Samba vem com todas as distribuições do Linux vistas neste livro. Patches, documentação e outros recursos úteis estão disponíveis em samba.org. Certifique-se de utilizar os pacotes mais atuais do Samba disponíveis para seu sistema, visto que bugs poderiam ser a causa potencial da perda de dados ou problemas de segurança.

Em todos os sistemas, você precisará editar o arquivo **smb.conf** (que deve estar localizado em **/etc/samba/smb.conf** ou **/etc/smb.conf**) para instruir o Samba como ele deve se comportar. Nesse arquivo, você especifica os diretórios e impressoras que devem ser compartilhados, seus direitos de acesso e parâmetros operacionais genéricos do Samba. O pacote Samba vem com um modelo de arquivo **smb.conf** bem comentado que geralmente é um excelente ponto de partida para configurações novas. Observe que assim que o Samba estiver em execução, ele sempre verifica o arquivo de configuração e carrega todas as alterações que você faz.

É importante estar atento às implicações de segurança do compartilhamento de arquivos e outros recursos por meio de uma rede. Para uma instalação típica, você precisa de duas coisas para garantir um nível de segurança básico:

- Especificar explicitamente quais clientes podem acessar os recursos compartilhados pelo Samba. Essa parte da configuração é controlada pela cláusula `hosts allow` no arquivo **smb.conf**. Certifique-se de que ele contém apenas os endereços IP (ou intervalos de endereços) que ele deve ter.
- Bloquear acesso ao servidor fora da sua empresa. O Samba só utiliza criptografia para autenticação de senha. Ele não utiliza criptografia para seu transporte de dados. Dependendo da natureza dos dados armazenados no seu servidor Samba, é recomendável bloquear acesso ao servidor fora da sua empresa a fim de impedir que as pessoas façam acidentalmente a cópia de arquivos de uma maneira desprotegida. Isso em geral é feito no nível do firewall de rede; o Samba utiliza as portas UDP 137 a 139 e as portas TCP 137, 139 e 445.

A partir da distribuição da versão 3 do Samba, há uma excelente documentação disponível on-line em samba.org.

O Samba vem com padrões sensatos para a maioria das suas opções de configuração e a maioria das instalações só precisará de um pequeno arquivo de configuração. Utilize o comando **testparm -v** para obter uma listagem de todas as opções de configuração do Samba e dos valores com os quais elas estão atualmente configurados. Essa listagem incluirá suas configurações do arquivo **smb.conf** e também os valores padrão.

Evite configurar as opções no arquivo **smb.conf** a menos que elas sejam diferentes dos valores padrão e que você saiba de fato por que quer modificá-las. A vantagem dessa abordagem é que sua configuração automaticamente se adaptará às configurações recomendadas pelos autores do Samba quando você atualizar para uma mais versão recente do Samba.

Tendo dito isso, certifique-se de que a criptografia de senha está ativada:

```
encrypt passwords = true
```

Essa opção criptografa a troca de senhas entre clientes Windows e o servidor Samba. Atualmente ela é o padrão, mas não há nenhuma situação concebível em que seria recomendável desativá-la.

O recurso de criptografia requer que o servidor Samba armazene um hash de senha Windows especial para cada usuário. As senhas Windows funcionam de maneira fundamentalmente diferente das senhas UNIX e, portanto, não é possível utilizar as senhas em **/etc/shadow**.

O Samba fornece uma ferramenta especial para configurar essas senhas, **smbpasswd**. Por exemplo, vamos adicionar uma senha e configurá-la para o usuário tobi:

---

2 Em combinação com o LDAP, o Samba também pode servir como um controlador de domínio de backup. Consulte *Samba-3 by Example*, de John H. Terpstra (citação na página 578), para uma excelente discussão. Consulte o Capítulo 17, *Compartilhando arquivos de sistema*, para informações adicionais sobre o LDAP.

```
$ sudo smbpasswd -a tobi
New SMB password: <senha>
Retype new SMB password: <senha>
```

Os usuários também podem alterar suas próprias senhas Samba com **smbpasswd**:

```
$ smbpasswd -r smbserver -U tobi
New SMB password: <senha>
Retype new SMB password: <senha>
```

Esse exemplo altera a senha Samba do usuário tobi no de servidor smbserver.

Codificação do nome de arquivo

A partir da versão 3.0, o Samba codifica todos os nomes de arquivo em UTF-8. Se seu servidor for executado em uma localidade que usa o UTF-8, essa é uma excelente combinação.[3] Se estiver na Europa e ainda estiver utilizando uma das localidades ISO 8859 no servidor, você descobrirá que nomes de arquivo com caracteres especiais como ä, ö, ü, é, ou è parecerão mais estranhos quando você digitar **ls** em um diretório em que esses arquivos foram criados com o Samba utilizando o UTF-8. A solução é informar o Samba para usar a mesma codificação de caracteres do seu servidor:

```
unix charset = ISO8859-15
display charset = ISO8859-15
```

É importante certificar-se de que a codificação dos nomes de arquivos esteja correta desde o início. Do contrário, arquivos com nomes de arquivo codificados de maneira estranha se acumularão e sua correção pode ser uma tarefa bem complexa posteriormente.

Navegação pelo ambiente de rede

A primeira rede Windows era muito simples. Só era necessário conectar todas as máquinas Windows a um segmento Ethernet e os sistemas começavam a ver um ao outro de acordo com o nome e conseguiam compartilhar recursos. Essa façanha era realizada por meio de um sistema auto-organizável baseado em broadcasts e em eleições; uma máquina Windows era selecionada para ser a mestra e essa mestra então mantinha uma lista dos recursos disponíveis.

Quando o Windows começou a falar TCP/IP, esse sistema foi portado para utilizar broadcasts UDP. O efeito era mais ou menos o mesmo, mas com a sub-rede IP sendo a unidade operativa em vez da Ethernet. Os hosts descobertos aparecem no desktop Windows sob Network Neighborhood ou My Network Places, dependendo da versão específica do Windows. Para estruturar um pouco a visibilidade, cada máquina Windows também pode ser atribuída a um grupo de trabalho.

O Samba participa desse sistema por meio do daemon **nmdb**. O **nmbd** pode até mesmo tornar-se o mestre de navegação designado e assumir o papel de organizador de todos os sistemas Windows na sub-rede local. Infelizmente, assim que a rede começa a tornar-se mais complexa, com várias sub-redes e servidores, broadcasts não serão suficientes para permitir que máquinas Windows vejam umas as outras.

Para corrigir esse problema, configure um servidor Samba por sub-rede individual como um servidor WINS (Windows Internet Naming Service) especificando

```
wins support = yes
```

no **smb.conf** e então vinculando esses servidores às opções `wins server`.

Nossa experiência com o sistema de navegação da Microsoft foi variada, mesmo em redes Windows puras. Freqüentemente nos deparamos com situações estranhas em que um host não responde ao seu próprio nome ou em que um computador nas proximidades não poderia ser localizado mesmo que estivéssemos certos de que ele estava ativo e em execução. Esses problemas são ainda mais freqüentes em grandes redes com múltiplas sub-redes.

Se você servir clientes Windows modernos (Windows 2000 e superiores), disponibilize entradas DNS a todos os seus hosts e não exija navegação pelo Network Neighborhood, então você pode remover suporte NetBIOS do seu servidor Samba, se quiser. Será uma coisa a menos para se preocupar.

```
[global]
disable netbios = yes
name resolve order = host
```

3 Digite **echo $LANG** para ver se seu sistema está executando no modo UTF-8.

Sem o NetBIOS, o Windows terá de utilizar o DNS para descobrir o endereço IP de um servidor Samba, da mesma forma como com qualquer outro recurso na rede. Se executar o Samba sem suporte NetBIOS, não haverá necessidade de iniciar **nmbd**. Remova-o do script de inicialização do Samba em **/etc/init.d**.

Para conectar um cliente a um servidor Samba que desativou o suporte NetBIOS, é melhor digitar o nome DNS completo do servidor ao se conectar a um compartilhamento. Por exemplo:

```
\\server.example.com.\myshare
```

Observe o ponto no final do nome do servidor. Ele pede para Windows não se incomodar em tentar encontrar a máquina solicitada via o NetBIOS, mas, em vez disso, ir direto para o DNS.

Autenticação de usuário

Nos sistemas de autenticação Windows, o cliente não confia no servidor; a senha do usuário nunca percorre a rede como texto simples. Em vez disso, o Windows utiliza um método de desafio/resposta para a autenticação.

Se fornecer um nome de usuário e senha ao efetuar o login em um sistema Windows, o Windows tentará utilizar essas informações para autenticar o que quer que seja apresentado com uma requisição de autenticação. Por causa do método de desafio/resposta, não há nenhum perigo nesse comportamento além do fato de que a senha em texto simples é armazenada em algum lugar na RAM. O aspecto interessante é que, se um usuário tiver a mesma combinação nome de usuário e senha na sua máquina Windows no seu servidor Samba, o Samba aparentemente concederá acesso sem senha aos compartilhamentos Samba apropriados. Toda a autenticação acontece transparentemente no segundo plano.

A desvantagem da abordagem desafio/resposta é que o servidor tem de armazenar senhas equivalentes em texto simples. Na realidade, as cópias do servidor das senhas são criptografadas localmente, mas isso é principalmente uma precaução contra navegação casual. Um invasor que ganha acesso às senhas criptografadas pode utilizá-las para acessar as contas associadas sem a necessidade de outros tipos de quebra de senha. As senhas Samba devem ser protegidas mais vigorosamente do que o arquivo **/etc/shadow**.

Em ambientes complexos com múltiplos servidores Samba, faz sentido operar um serviço de diretório centralizado que se certifica de que a mesma senha está ativa em todos os servidores. O Samba suporta NIS,[4] LDAP e Windows como serviços de autenticação. NIS e LDAP são discutidos no Capítulo 17, *Compartilhando arquivos de sistema*.

Para mesclar os sistemas de autenticação do Windows e do Linux, você tem duas opções básicas. Primeiro, você pode configurar um servidor Samba para atuar como um Windows NT4 Primary Domain Controller (não como um servidor Active Directory, ainda). Alternativamente, é possível instalar o pGina (pgina.xpasystems.com) nos seus clientes Windows. Esse aplicativo inteligente substitui o sistema de login padrão do Windows por uma estrutura que suporta todos os tipos de serviços de autenticação padrão, incluindo LDAP e NIS.

Compartilhamento de arquivos básico

Se cada usuário tiver um diretório inicial, eles podem ser do tipo "compartilhado por volume":

```
[homes]
comment = Home Directories
browseable = no
valid users = %S
writeable = yes
guest ok = no
```

Essa configuração, por exemplo, permitirá que o usuário oetiker acesse seu diretório inicial por meio do caminho \\sambaserver\oetik a partir de qualquer sistema Windows.

Em algumas instalações, as permissões padrão nos diretórios iniciais do Linux permitem que as pessoas pesquisem os arquivos uma da outra. Como o Samba conta com as permissões de arquivo UNIX para implementar restrições de acesso, os usuários Windows acessando via o CIFS também serão capazes de ler os diretórios iniciais um do outro. Entretanto, a experiência mostra que esse comportamento tende a confundir os usuários Windows e fazer com que eles se sintam expostos. A linha `valid users` no trecho de configuração anterior instrui o Samba a bloquear conexões com diretórios iniciais das outras pessoas. Deixe-a fora se isso não for o que você quer.

O Samba utiliza sua seção mágica `[homes]` como um último recurso. Se houver um compartilhamento explicitamente definido na configuração do diretório inicial de um usuário particular, os parâmetros configurados aí sobrescreverão os valores configurados pelo `[homes]`.

---

4 Mesmo que continue listado em algumas documentações, não há suporte NIS+ na versão 3 do Samba. Parece que ninguém estava disposto a manter o código, assim esse recurso foi removido (pelo menos por enquanto).

Compartilhamentos de grupos

O Samba pode mapear listas de controle de acesso (Access Control List – ACLs) do Windows para permissões de arquivo ou ACLs (se o sistema de arquivos subjacente suportá-las). Na prática, achamos que o conceito das ACLs tende a ser muito complexo para a maioria dos usuários. Portanto, em geral só configuramos um compartilhamento especial para cada grupo de usuários que requer um e configuramos o Samba para que ele cuide da configuração das permissões apropriadas. Sempre que um usuário tentar montar esse compartilhamento, o Samba verifica para se certificar que o requerente está no grupo Linux apropriado e, então, alterna seu UID efetivo para o proprietário designado do compartilhamento de grupo (um pseudo-usuário criado para esse propósito). Por exemplo:

```
[eng]
comment = Group Share for engineering
; Everybody who is in the eng group may access this share.
; People will have to log in using their Samba account.
valid users = @eng
; We have created a special user account called "eng". All files
; written in this directory will belong to this account as
; well as to the eng group.
force user = eng
force group = eng
path = /home/eng

; Disable NT Acls as we do not use them here.
nt acl support = no

; Make sure that all files have sensible permissions.
create mask = 0660
force create mask = 0660
security mask = 0000
directory mask = 2770
force directory mask = 2770
directory security mask = 0000

; Normal share parameters
browseable = no
writeable = yes
guest ok = no
```

Um efeito semelhante pode ser alcançado por meio da opção `inherit permissions` do Samba. Se essa opção estiver ativada em um compartilhamento, todos os novos arquivos e diretórios herdam suas configurações dos seus diretórios pai:

```
[eng]
comment = Group Share for engineering
path = /home/eng
nt acl support = no
browseable = no
writeable = yes
inherit permissions = yes
```

Como agora o Samba propaga as configurações do diretório pai, é importante configurar as permissões na raiz do compartilhamento apropriadamente:

```
$ sudo chmod u=rw,g=rws,o= /home/eng
$ sudo chgrp eng /home/eng
$ sudo chown eng /home/eng
```

Observe que essa configuração ainda requer a criação de um pseudo-usuário para que ele atue como o proprietário do diretório compartilhado.

Redirecionamento transparente com MS DFS

O Distributed File System da Microsoft (MS DFS) permite que diretórios dentro de um compartilhamento desencadeiem clientes para automontar transparentemente outros compartilhamentos assim que eles são acessados. Para *habitués* do UNIX e Linux isso não parece algo importante, mas, para o Windows, todo o conceito é bastante revolucionário e inesperado.

Isso talvez seja parte da razão por que ele não é amplamente utilizado, embora o DFS seja suportado por todas as versões do Windows desde o Windows 98.

Eis um exemplo:

```
[global]
; Ativa o suporte MS DFS para este servidor Samba.
host msdfs = yes
...
[mydfs]
; Esta linha instrui o Samba a examinar
; DFS symlinks no diretório do seu compartilhamento.
msdfs root = yes
path = /home/dfs/mydfs
```

Você cria links simbólicos no **/home/dfs/mydfs** para configurar as automontagens reais. Por exemplo, o comando a seguir faz do "diretório" **jump** um link para um dos dois diretórios em outros servidores. (Observe as aspas. Elas são necessárias para proteger as barras normais.)

```
$ sudo ln -s 'msdfs:serverX\shareX,serverY\shareY' jump
```

Se mais de uma origem for fornecida (como aqui), o Windows escolherá uma delas. Os usuários que acessam **\\server\mydfs\jump** agora estarão realmente lendo arquivos a partir do shareX no serverX ou shareY no serverY, dependendo da disponibilidade. Se os sistemas de arquivos forem exportados como leitura/gravação, você deve certificar-se de que haja algum mecanismo correto para sincronizar os arquivos. O **rsync** pode ser útil para isso.

Com o Samba, também é possível redirecionar todos os clientes que acessam um compartilhamento particular para um servidor diferente. Isso é algo que um servidor Windows não pode fazer.

```
[myredirect]
msdfs root = yes
msdfs proxy = \\serverZ\shareZ
```

Note que o DFS só funcionará para usuários com o mesmo nome de usuário e senha em todos os servidores envolvidos.

smbclient: um cliente CIFS simples

Além dos seus muitos recursos do lado servidor, o pacote Samba inclui um programa simples de transferência de arquivos a partir da linha de comando chamado **smbclient**. Você pode usar esse programa para acessar diretamente qualquer servidor Windows ou Samba. Por exemplo:

```
$ smbclient //redmond/joes -U joe
Password: <senha>
Doman=[REDMOND] OS=[Windows 5.0] Server=[Windows 2000 LAN Manager]
smb: \>
```

Depois de efetuar o login com sucesso no servidor de arquivos, você utiliza os comandos no estilo **ftp** padrão (como **get**, **put**, **cd**, **lcd** e **dir**) para navegar e transferir arquivos.

O sistema de arquivos smbfs

O Linux inclui suporte direto do lado cliente para o sistema de arquivos SMB/CIFS. Você pode montar um compartilhamento CIFS na árvore do seu sistema de arquivos de maneira muito semelhante com a qual você pode montar qualquer outro sistema de arquivos que seja diretamente entendido pelo kernel. Por exemplo:

```
# mount -t smbfs -o username=joe //redmond/joes /home/joe/mnt
```

Embora esse recurso seja útil, tenha em mente que o Windows conceitua montagens de rede como sendo estabelecidas por um determinado usuário (daí a opção anterior **username=joe**), enquanto o Linux as considera, de modo mais geral, pertencentes ao sistema como um todo. Em geral, os servidores Windows não podem lidar com o conceito de que várias pessoas diferentes poderiam acessar um compartilhamento Windows montado.

Da perspectiva do cliente Linux, todos os arquivos no diretório montado parecem pertencer ao usuário que o montou. Se você montar o compartilhamento como root, então todos os arquivos pertencerão ao root, e os usuários comuns podem não ser capazes de gravar arquivos no servidor Windows.

As opções de montagem **uid**, **gid**, **fmask** e **dmask** permitem ajustar essas configurações de modo que os bits de posse e de permissão estejam mais sintonizados com a diretiva de acesso pretendida para esse compartilhamento. Verifique a página manual do **mount.smbfs** para informações adicionais sobre esse comportamento.

Para permitir que os usuários montem um compartilhamento Windows por conta própria, você pode adicionar uma linha no seguinte formato ao seu arquivo **/etc/fstab**:

```
//redmond/joes /home/joe/mnt smbfs username=joe,fmask=600,dmask=700,user,noauto 0 0
```

Por causa da opção `user` especificada aqui, os usuários agora podem montar o sistema de arquivos simplesmente executando o comando

```
$ mount /home/joe/mnt
```

mount solicitará que o usuário forneça uma senha antes de montar o compartilhamento.

Consulte o Capítulo 16 para informações adicionais sobre NFS.

Embora o NFS seja o padrão UNIX para o serviço de arquivo de rede, em algumas situações talvez faça mais sentido utilizar o Samba e o CIFS para compartilhar arquivos entre computadores UNIX e Linux. Por exemplo, é perigoso permitir que usuários realizem montagens NFS dos sistemas de arquivos corporativos a partir dos seus laptops pessoais.[5] Mas você pode seguramente utilizar o CIFS para fornecer a esses laptops acesso aos diretórios iniciais dos seus proprietários.

A IBM tem investido recursos para tornar o acesso CIFS a partir do Linux o mais transparente possível, com o objetivo de estabelecer o Samba como uma alternativa robusta ao NFS. Um resultado inicial desse esforço é que o Samba 3.x pode ser configurado para fornecer "extensões UNIX". Com esse recurso ativado, você pode ver todos os atributos de arquivos, posses e tipos de arquivo UNIX mesmo ao montar um compartilhamento por meio do Samba.

Juntamente com as modificações do lado servidor, há também um novo módulo no lado cliente dos sistemas de arquivos em desenvolvimento, chamado cifs. O desenvolvimento atual focaliza o módulo cifs, que já apresenta melhor suporte à depuração e à configuração que o smbfs tradicional.

26.7 COMPARTILHANDO IMPRESSORAS COM O SAMBA

A abordagem simples para compartilhamento de impressoras é adicionar uma seção `[printers]` ao arquivo **smb.conf**; isso faz com que o Samba compartilhe todas as impressoras locais. O Samba utiliza os comandos do sistema de impressão para fazer seu trabalho, mas, como a impressão no Linux não está bem padronizada, talvez você precise informar ao Samba qual sistema de impressão particular está em uso no seu servidor configurando a opção `printing` como um valor apropriado. Verifique a página man do **smb.conf** para a lista dos sistemas de impressão que são atualmente suportados.

```
[printers]
; Where to store print files before passing them to the printing system?
path = /var/tmp
; Everybody can use the printers.
guest ok = yes
; Let Samba know this share is a printer.
printable = yes
; Show the printers to everyone looking.
browseable = yes
; Tell samba what flavor of printing system the system is using.
printing = LPRNG
```

Consulte o Capítulo 23 para informações adicionais sobre impressão.

Os clientes Windows agora podem utilizar essas impressoras como impressoras em rede, como se elas fossem hospedadas em um servidor Windows. Há, porém, um pequeno problema. O cliente Windows desejará saber o tipo de impressora que ele está utilizando e solicitará ao usuário que ele selecione um driver de impressora apropriado. Isso resulta em uma grande quantidade de solicitações de suporte por parte de usuários que não sabem prosseguir nessa situação. Se a impressora particular em questão exigir um driver que não está incluído no Windows, essa situação demandará ainda mais suporte.

5 A segurança NFS baseia-se na idéia de que o usuário não tem nenhum acesso root no cliente e de que há UIDs correspondentes no cliente e no servidor. Esse normalmente não é o caso para máquinas autogerenciadas.

Felizmente, o Samba pode ser configurado para fornecer os drivers de impressora Windows necessários para os clientes Windows. Mas, para que isso funcione, você deve fazer alguns preparativos. Primeiro, certifique-se de que o Samba comporta-se como um servidor de impressão adicionando entradas apropriadas à seção `[global]` do arquivo **smb.conf**:

```
[global]
; Who is our printer admin
printer admin = printadm
; The following have the right value by default.
disable spoolss = no
; Don't bother showing it; you cannot add printers anyway
show add printer wizard = no
; Assuming you want everybody to be able to print
guest ok = yes
browseable = no
```

Agora o Samba sabe que esse é um servidor de impressão e aceitará o usuário printadm como seu administrador de impressoras.

Se você fornecer drivers de impressora aos seus clientes Windows, deverá haver um lugar para armazenar os drivers. Isso é feito por meio de um compartilhamento especial chamado `[print$]`.

```
[print$]
comment = Printer Driver Area
; Place to store the printer drivers
path = /var/lib/samba/printers
browseable = yes
guest ok = yes
read only = yes
; Who can administer the printer driver repository
write list = printadm
```

Antes que você possa começar a carregar os drivers de impressora para o novo servidor de impressão, você deve cuidar de alguns outros detalhes no nível do sistema. Certifique-se de que a conta printadm existe e tem permissão de acessar o Samba.

```
$ sudo useradd printadm
$ sudo smbpasswd -a printadm
```

O Samba só pode armazenar drivers de impressora se a estrutura de diretórios apropriada existir e for possuída por printadm (como definida na opção `write list`):

```
$ sudo mkdir -p /var/lib/samba/printers
$ sudo cd /var/lib/samba/printers
$ sudo mkdir W32X86 WIN40
$ sudo chown -R printadm .
```

Nesse ponto há duas opções: você pode usar uma máquina Windows e carregar os drivers de impressora a partir dela ou utilizar as ferramentas do Samba para fazer tudo isso a partir da linha de comando. Infelizmente, não há nenhuma maneira simples de saber o que exatamente tem de ser instalado para um driver particular, portanto, recomendamos a primeira abordagem na maioria das circunstâncias. Somente se você precisar instalar repetidamente um driver em vários servidores vale a pena examinar a instalação e entender como replicá-la com as ferramentas de linha de comando.

Instalando um driver de impressora a partir do Windows

Para instalar os drivers a partir de um cliente Windows, abra uma conexão com o servidor Samba digitando \\samba-server.example.com na caixa de diálogo Start -> Run. O Windows solicitará o login para o servidor Samba. Efetue o login como o usuário printadm. Se tudo der certo, uma janela abre com uma lista dos compartilhamentos fornecidos pelo servidor.

Dentro da subpasta Printers talvez você possa ver todas as impressoras que você compartilhou a partir do seu servidor. Clique com o botão direito do mouse no espaço em branco perto dos ícones de impressora para ativar o diálogo Server Properties, adicione então seus drivers de impressora favoritos via a guia Drivers.

Os drivers carregados acabam sendo colocados no diretório especificado para o compartilhamento `[print$]`. Nesse ponto, talvez você queira examinar rapidamente as propriedades do driver que você acabou de carregar. Essa lista de arquivos é o que você terá de fornecer à ferramenta de linha de comando do Samba se quiser automatizar o carregamento do driver.

Depois de os drivers adequados serem carregados, você agora pode associá-los com impressoras específicas. Exiba o painel Properties de cada impressora sucessivamente (clicando com o botão direito do mouse e marcando Properties) e selecione os drivers apropriados na guia Advanced. Abra então o diálogo Printing Defaults e modifique as configurações.

Mesmo se você estiver satisfeito com as configurações padrão, faça pelo menos uma pequena alteração para forçar o Windows a armazenar as estruturas de dados de configuração no servidor Samba. O Samba irá então fornecer esses dados aos clientes que acessam a impressora. Se não seguir esse último passo, você talvez faça com que os clientes travem porque nenhuma configuração padrão válida poderá ser localizada quando eles tentarem utilizar a impressora.

Instalando um driver de impressora a partir da linha de comando

Como talvez você já tenha adivinhado, alguns desses passos são difíceis de replicar sem utilizar o Windows, especialmente a configuração dos padrões de impressora. Mas, se quiser configurar centenas de impressoras em um servidor Samba, é recomendável tentar fazer isso a partir da linha de comando. A configuração a partir da linha de comando funciona particularmente bem para impressoras PostScript porque o driver de impressora PostScript do Windows funciona corretamente sem informações da configuração padrão.

Se você escreveu uma nota dos arquivos necessários para um determinado driver, você poderá instalar o driver a partir da linha de comando. Primeiro, copie os arquivos necessários para o compartilhamento `[print$]`:

```
$ cd ~/mydriver
$ smbclient -U printadm '//samba-server/print$' -c 'mput *.*'
```

Em seguida, atribua o driver a uma impressora particular. Vamos supor que temos uma impressora PostScript simples com um arquivo PPD personalizado:

```
$ rpcclient -U printadm -c "\
adddriver \"Windows NT x86\" \"Our Custom PS:\
PSCRIPT5.DLL:CUSTOM.PPD:PS5UI.DLL:PSCIPT.HLP:NULL:NULL:PSCRIPT.NTF\"" \
samba-server
```

As barras normais no final das linhas permitem que o comando seja dividido em várias linhas por uma questão de clareza; você pode omitir estas e inserir o comando em uma única linha, se preferir. As barras normais antes das aspas duplas distinguem os conjuntos aninhados de aspas.

A longa string no exemplo anterior contém as informações listadas no diálogo de propriedade do driver de impressora que são vistas quando o driver de impressora está sendo instalado a partir do Windows:

- Nome longo da impressora
- Nome do arquivo de driver
- Nome do arquivo de dados
- Nome do arquivo de configuração
- Nome do arquivo Help
- Nome do monitor de linguagem (configure isso como NULL se você não tiver um)
- Tipo de dados padrão (configure isso como NULL se não houver nenhum)
- Lista separada por vírgulas dos arquivos adicionais

Para configurar uma impressora para utilizar um dos drivers carregados, execute

```
$ rpcclient -U printadm -c "\
set driver \"myprinter\" \"Our Custom PS\"" samba-server
```

26.8 Depurando o Samba

O Samba normalmente é executado sem exigir muita atenção. Entretanto, se você tiver um problema, poderá consultar duas fontes primárias para informações de depuração: os arquivos de log por cliente individual e o comando **smbstatus**. Certifique-se de que você tem as configurações apropriadas do arquivo de log no seu arquivo de configuração:

```
[global]
; The %m causes a separate file to be written for each client.
log file = /var/log/samba.log.%m
max log size = 1000
; How much info to log. You can also specify log levels for components
; of the system (here, 3 generally, but level 10 for authentication).
log level = 3 auth:10
```

Níveis de log mais altos produzem informações adicionais. O login leva tempo, portanto, não solicite detalhes demais a menos que você esteja depurando. A operação pode tornar-se significativamente lenta.

O exemplo a seguir mostra as entradas de log geradas por uma tentativa de conexão malsucedida seguida por uma bem-sucedida.

```
[2004/09/05 16:29:45, 2] auth/auth.c:check_ntlm_password(312)
  check_ntlm_password:  Authentication for user [oetiker] -> [oetiker] FAILED with error
    NT_STATUS_WRONG_PASSWORD
[2004/09/05 16:29:45, 2] smbd/server.c:exit_server(571)
  Closing connections
[2004/09/05 16:29:57, 2] auth/auth.c:check_ntlm_password(305)
  check_ntlm_password:  authentication for user [oetiker] -> [oetiker] -> [oetiker] succeeded
[2004/09/05 16:29:57, 1] smbd/service.c:make_connection_snum(648)
  etsuko (127.0.0.1) connect to service oetiker initially as user oetiker (uid=1000, gid=1000) (pid
    20492)
[2004/09/05 16:29:58, 1] smbd/service.c:close_cnum(837)
  etsuko (127.0.0.1) closed connection to service oetiker
[2004/09/05 16:29:58, 2] smbd/server.c:exit_server(571)
  Closing connections
```

O comando **smbcontrol** é útil para alterar o nível de depuração sob um servidor Samba em execução sem alterar o arquivo **smb.conf**. Por exemplo,

```
$ sudo smbcontrol smbd debug "4 auth:10"
```

O exemplo anterior configuraria o nível de depuração global como 4 e o nível de depuração para questões relacionadas à autenticação como 10. O argumento **smbd** especifica que todos os daemons **smbd** no sistema terão seus níveis de depuração configurados. Para depurar uma conexão estabelecida específica, você pode utilizar o comando **smbstatus** para descobrir qual daemon **smbd** trata a conexão e então passar seu PID para **smbcontrol** a fim de depurar apenas essa conexão. Com níveis de log acima de 100 você começará a ver senhas (criptografadas) nos logs.

O **smbstatus** mostra as conexões atualmente ativas e os arquivos bloqueados. Essas informações podem ser especialmente úteis quando você estiver monitorando problemas de bloqueios (por exemplo: "Qual usuário tem o arquivo xyz aberto de forma exclusiva para leitura/gravação?"). A primeira parte da saída lista os recursos com os quais um usuário se conectou. A segunda parte lista todos os bloqueios de arquivos ativos.[6]

```
Samba version 3.0.5

PID                 Username      Group           Machine
-------------------------------------------------------------------------------
12636               zauck         ee              zhaka (192.168.1.228)
29857               milas         guests          beshil (192.168.1.123)

Service             pid           machine         Connected at
-------------------------------------------------------------------------------
milasa              29857         beshil          Fri Sep  3 17:07:39 2004
zaucker             12636         zhaka           Thu Sep  2 12:35:53 2004

Locked files:

Pid                 DenyMode     Access   R/W    Oplock  Name
-------------------------------------------------------------------------------
29857               DENY_NONE    0x3      RDWR   NONE    /home/milasa/hello.dba
12636               DENY_NONE    0x2019f  RDWR   NONE    /home/zaucker/aufbest.doc
```

Se eliminar o **smbd** associado com certo usuário, todos os seus bloqueios desaparecerão. Alguns aplicativos tratam isso elegantemente e só irão readquirir um bloqueio se precisarem dele. Outros (como o MS Access) se congelarão e sofrerão uma morte horrível com a necessidade de um grande volume de cliques no lado Windows apenas para serem capazes de fechar o aplicativo malsucedido. Tão dramático quanto isso pode parecer, ainda temos de verificar toda corrupção de arquivo resultante desse procedimento. Em qualquer caso, tenha cuidado quando o Windows afirma que os arquivos foram bloqueados por algum outro aplicativo. Em geral o Windows está certo e você deve corrigir o problema no lado cliente fechando o aplicativo em vez de tratá-lo com força bruta a partir do servidor.

---

6 A saída gerada por **smbstatus** contém algumas linhas muito longas; nós a condensamos por uma questão de clareza.

26.9 LEITURA RECOMENDADA

TERPSTRA, JOHN H. *Samba-3 by Example: Practical Exercises to Successful Deployment. 2. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2006. (uma versão online deste livro está disponível em: samba.org.)

TERPSTRA, JOHN H. E VERNOOIJ, JELMER R. *The Official Samba-3 HOWTO and Reference Guide. 2. ed.* Upper Saddle River, NJ: Prentice Hall PTR, 2006. (uma versão online deste livro está disponível em: samba.org.)

26.10 EXERCÍCIOS

E26.1 Por que você iria querer bloquear acesso Internet às portas 137–139 e 445 em um servidor Samba?

E26.2 Instale o software Cygwin em uma máquina Windows e utilize **ssh** no **rxvt** para conectar-se a uma máquina Linux. Quais diferenças entre o PuTTY você encontra?

★★E26.3 No laboratório, compare o desempenho de um cliente que acesse arquivos por meio do Samba com um que acesse arquivos de um servidor CIFS nativo (isto é, uma máquina Windows). Se seus dois servidores de teste tiverem hardware diferente, imagine uma maneira de ajustar a diferença de hardware para que a comparação seja mais indicativa do desempenho do software de servidor. (Pode exigir acesso de root.)

★★E26.4 No laboratório, utilizando um analisador de pacotes (packet sniffer) como **tcpdump** ou Wireshark, monitore uma sessão **telnet** entre um servidor Windows e um servidor Linux. Obtenha e instale o software PuTTY e repita o monitoramento. Em cada caso, o que você pode observar com o analisador de pacotes? (Exige acesso de root.)

★★E26.5 Configure um servidor de impressão Samba que fornece os drivers de impressora Windows para todas as impressoras que ele compartilha. Certifique-se de que as impressoras são fornecidas com uma configuração padrão funcional.

27 Dispositivos seriais

Desde 1969, o padrão da interface serial RS-232C forneceu uma maneira para que dispositivos de muitos tipos e de muitos fabricantes se comunicassem. O RS-232C é o padrão de interface de computador que permaneceu praticamente intocado ao longo dos seus quase 40 anos de história e continua em uso hoje em dia.

As portas seriais são utilizadas com uma série de dispositivos, inclusive impressoras, terminais e outros computadores. Elas também são encontradas em uma ampla gama de equipamentos personalizados, de hobby e de baixo volume de vendas (trocadores automáticos de mídia, sensores de temperatura, receptores de GPS e até mesmo máquinas de costura). Um dispositivo pode ser conectado ao sistema diretamente (por meio de um cabo) ou via linha telefônica com modems nas duas pontas.

Este capítulo descreve como conectar dispositivos ao seu sistema e explica como configurar seu software para tirar proveito deles. Utilizamos modems e impressoras como exemplos específicos, mas outros dispositivos são basicamente semelhantes.

As primeiras seções tratam de questões de cabeamento e hardware serial. Em seguida, a partir da página 587, discutimos a infra-estrutura de software que tem sido historicamente utilizada para dar suporte a terminais de hardware. Os terminais raramente são utilizados hoje em dia, mas seus fantasmas sobrevivem na manipulação de pseudoterminais e sistemas com janelas do Linux. O restante do capítulo (começando na página 592) fornece algumas informações gerais sobre modems, depuração de dispositivos seriais e USB (Universal Serial Bus).

27.1 O PADRÃO RS-232C

A maioria das portas seriais segue alguma variante do padrão RS-232C. Esse padrão especifica as características elétricas e o significado de cada fio de sinal, bem como as atribuições de pinos no tradicional conector serial (DB-25) de 25 pinos mostrado na Ilustração A.

Ilustração A Um conector DB-25 macho

O padrão RS-232[1] completo é excessivamente grande para a maioria das situações práticas, pois define numerosos sinais que não são necessários para comunicação básica. Os conectores DB-25 também são inconvenientemente grandes. Como resultado, uma série de conectores alternativos acabou se tornando amplamente usada. Esses são descritos na seção *Conectores alternativos*, começando na página 581.

1 Para ser tecnicamente correto, esse padrão deveria ser chamado de EIA-232-E. Mas ninguém terá a mínima idéia sobre o que você está falando.

O RS-232 tradicional utilizava cabo de par trançado blindado (*shielded twisted-pair* – STP), normalmente fios de calibre 22 trançados. Hoje em dia, o mesmo par de cabo trançado não-blindado (*unshielded twisted pair* – UTP), que é o padrão para conexões Ethernet, é utilizado para o cabeamento serial. Tecnicamente, esse cabo não segue a especificação R-232, mas, na prática, costuma funcionar razoavelmente.

As voltagens de sinal do RS-232 original eram de ±12 volts DC, mas ±5 volts são mais comuns hoje em dia. Algumas vezes, é utilizado ± 3 volts. Voltagens maiores são menos suscetíveis à interferência. Todas essas voltagens atendem à especificação RS-232, de forma que é perfeitamente correto conectar dispositivos que utilizem padrões de voltagem diferentes.

O RS-232 não é um sistema eletricamente "balanceado"; ele utiliza um único condutor para os dados que trafegam em cada direção. Portanto, as propriedades elétricas especiais de cabos de par trançado podem ser menos significativas para comunicação serial do que são para, digamos, redes Ethernet. De fato, os cabos de par trançado podem realmente reduzir a confiabilidade e a amplitude de uma conexão serial caso as duas linhas de dados (TD e RD) sejam colocadas juntas em um único par. Portanto, não faça isso.

Não há nenhum padrão sobre quais sinais RS-232 deveriam percorrer juntos em um cabo de par trançado. Algumas fontes recomendam parear sinais terra com TD e RD, mas esse pareamento significa um condutor adicional e fornece vários caminhos para o sinal terra. Pelo que nos consta, não há nenhuma razão que force o uso dessa convenção.

Os conectores DB-25 podem ser ou machos (com os pinos salientes, chamados de DB25P) ou fêmeas (com orifícios correspondentes aos pinos, chamados de DB25S). Há minúsculos números quase invisíveis próximos dos pinos ou orifícios numerando-os de 1 a 25. Você pode ver melhor os números colocando o conector contra a luz e observando-o de maneira enviesada. Algumas vezes somente os pinos 1, 13, 14 e 25 são numerados.

A Ilustração A mostra um conector DB-25 macho. Como acontece com todos os conectores seriais, os números de pinos em um conector fêmea representam uma imagem espelhada daqueles contidos em um conector macho para que pinos de mesma numeração se encaixem. A ilustração é desenhada com base na orientação mostrada (como se estivéssemos olhando para o final do cabo, prestes a plugar o conector em sua parte frontal).

Note que, na Ilustração A, somente sete pinos são realmente instalados. Isto é típico do mundo real. Os sinais RS-232 e suas atribuições de pinos em um conector DB-25 são mostrados na Tabela 27.1. Somente os sinais sombreados são utilizados na prática (pelo menos em sistemas computacionais genéricos); todos os demais podem ser ignorados.

Tabela 27.1 Atribuições de sinais e de pinos R-232 em um DB-25

Pino	Nome	Função	Pino	Nome	Função
1	FG	Terra de frame	14	STD	TD secundário
2	TD	Dados transmitidos	15	TC	Relógio para transmissão
3	RD	Dados recebidos	16	SRD	RD secundário
4	RTS	Sinal de solicitação de envio	17	RC	Relógio para recebimento
5	CTS	Transmissão pode prosseguir	18	–	Não empregado
6	DSR	Pronto para dados	19	SRTS	RTS secundário
7	SG	Sinal terra	20	DTR	Pronto para receber uma transmissão
8	DCD	Detecção de portadora de dados	21	SQ	Detector de qualidade de sinal
9	–	Tensão positiva	22	RI	Indicador de anel
10	–	Tensão negativa	23	DRS	Seletor de velocidade de dados
11	–	Não empregado	24	SCTE	Relógio de transmissão externo
12	SDCD	DCD secundário	25	BUSY	Ocupado
13	SCTS	CTS secundário			

As duas configurações de interface para equipamentos seriais são DTE (Data Terminal Equipment) e DCE (Data Communications Equipment). DTE e DCE compartilham as mesmas pinagens, mas especificam interpretações diversas dos sinais RS-232.

Cada dispositivo é configurado como DTE ou como DCE; alguns poucos dispositivos suportam ambos, mas não simultaneamente. Computadores, terminais e impressoras geralmente são DTE, e a maioria dos modems é DCE. As portas seriais DTE e DCE podem se comunicar entre si em qualquer combinação, mas combinações diversas exigem cabeamentos diferentes.

Não há razão lógica para existirem tanto o DTE como o DCE; todos os equipamentos poderiam utilizar o mesmo esquema de fiação. A existência das duas convenções é meramente um dos diversos legados históricos sem sentido do RS-232.

DTE e DCE podem se tornar bastante confusos se você pensar muito no assunto. Quando isso acontece, simplesmente respire fundo e releia os seguintes pontos:

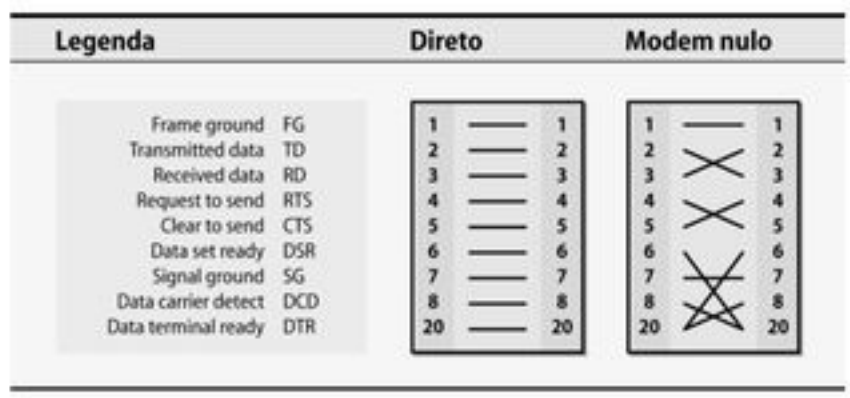
- A pinagem do RS-232 para um determinado tipo de conector é sempre a mesma, independentemente de o conector ser macho ou fêmea (números de pinos concordantes sempre coincidem) e independentemente de o conector estar em um cabo, em um dispositivo DTE ou em um dispositivo DCE.
- Toda a terminologia RS-232 se baseia no modelo de uma conexão direta de um dispositivo DTE para um dispositivo DCE. (Por "direta" queremos dizer que TD na ponta DTE é conectado ao TD na ponta DCE e assim por diante. Cada pino se conecta com o pino de mesma numeração do outro lado.)
- Os sinais recebem nomes sob a perspectiva do dispositivo DTE. Por exemplo, o nome TD (*transmitted data*, ou dados transmitidos) na realidade significa "dados transmitidos do DTE para o DCE". Apesar do nome, o pino TD é uma *entrada* em um dispositivo DCE. De maneira semelhante, o RD é uma entrada para o DTE e uma saída para o DCE.
- Ao conectarmos equipamento DTE a equipamento DTE (computador-terminal ou computador-computador), temos de enganar cada dispositivo fazendo com que eles acreditem que o outro é o DCE. Por exemplo, ambos os dispositivos DTE têm a expectativa de transmitir em TD e receber em RD; temos de conectar os fios de maneira invertida para que o pino de transmissão de um dispositivo vá para o pino de recepção do outro dispositivo e vice-versa.
- Três conjuntos de sinais devem ser cruzados desta maneira para comunicação DTE-DTE (se optarmos por conectá-los todos). TD e RD devem ser cruzados. RTS e CTS devem ser cruzados. E o pino DTR de cada lado tem de estar conectado tanto ao pino DCD como ao DSR do correspondente.
- Para confundir ainda mais, um cabo cruzado para comunicação DTE-DTE é normalmente chamado de cabo "modem nulo". Você pode ser tentado a utilizar um cabo modem nulo para conectar um modem, porém, como os modems são DCE, isso não funcionará! Um cabo para modem é chamado de "cabo modem" ou "cabo direto".

Como a questão DTE *versus* DCE é tão confusa, ocasionalmente talvez você veja tentativas bem-intencionadas, mas incorretas, de dar alguma luz à nomenclatura definindo DTE e DCE como se tivessem pinagens distintas (por exemplo, renomeando o pino TD de um DCE para RD e vice-versa). Nesse universo alternativo, as pinagens variam, mas as conexões de cabos (por nome de sinal), não. Sugerimos que você ignore qualquer material que fale sobre "pinagem DTE" ou "pinagem DCE"; é pouco provável que seja uma fonte de informação confiável.

Originalmente, os dispositivos DTE deveriam supostamente ter conectores machos e os dispositivos DCE serem conectores fêmea. Os projetistas de hardware acabaram descobrindo que os conectores machos eram mais frágeis. Hardware computacional caro normalmente tem agora conectores fêmeos e a maioria dos cabos é de machos em ambas as extremidades.[2]

A Ilustração B mostra as atribuições de pinos e de conexões tanto para cabos modem nulo como para cabos diretos. Somente sinais utilizados no mundo real são mostrados.

Ilustração B Atribuições de pinos e conexões para cabos DB-25



27.2 CONECTORES ALTERNATIVOS

As seções a seguir descrevem a maioria dos sistemas alternativos de conectores: mini DIN-8, DB-9 e RJ-45. Apesar de suas diferenças físicas, todos esses conectores fornecem acesso aos mesmos sinais elétricos do que um conector DB-25. Dispositivos que utilizam conectores diferentes sempre são compatíveis quando o tipo correto de cabo conversor é utilizado.

2 Na Qwest, os termos "macho" e "fêmea" são considerados inadequados. Os funcionários são encorajados a utilizar as palavras "plugue" e "receptáculo". Os nomes DB25P e DB25S de conectores padrão na verdade derivam de uma terceira convenção: "pino" e "soquete".

A variante mini DIN-8

Cabos Mini DIN-8 são encontrados em muitos laptops e estações de trabalho. Esse conector quase circular e extremamente compacto fornece conexões para sete sinais. Ele é mostrado na Ilustração C.

Ilustração C Um conector mini DIN-8 macho

Lojas de informática normalmente vendem cabos conversores DB-25 para mini DIN 8 moldados por injeção. Não tente você mesmo confeccioná-los, pois um mini DIN-8 é tão pequeno que é um desafio estabelecer-se conexões seguras utilizando as mãos. As atribuições de pino são mostradas na Tabela 27.2 a seguir.

Tabela 27.2 Pinos de um cabo direto mini DIN-8 para DB-25

DIN-8	DB-25	Sinal	Função
3	2	TD	Dados transmitidos
5	3	RD	Dados recebidos
6	4	RTS	Sinal de solicitação de envio
2	5	CTS	Transmissão pode prosseguir
4,8	7	SG	Sinal terra
7	8	DCD	Detecção de portadora de dados
1	20	DTR	Pronto para receber uma transmissão

A variante DB-9

Comumente encontrado em PCs, esse conector de nove pinos (que se parece com um conector DB-25 "júnior") fornece os oitos sinais mais comumente utilizados.

Ilustração D Um conector DB-9 macho

Lojas de informática em sua região devem ter disponíveis cabos conversores DB-9 para DB-25 pré-fabricados. A Tabela 27.3 mostra as atribuições de pinos.

Tabela 27.3 Pinos para um cabo direto DB-9 para DB-25

DB-9	DB-25	Sinal	Função
3	2	TD	Dados transmitidos
2	3	RD	Dados recebidos
7	4	RTS	Sinal de solicitação de envio
8	5	CTS	Transmissão pode prosseguir
6	6	DSR	Pronto para dados
5	7	SG	Sinal terra
1	8	DCD	Detecção de portadora de dados
4	20	DTR	Pronto para receber uma transmissão

A variante RJ-45

Um RJ-45 é um conector modular de telefone de 8 fios. Ele é semelhante ao conector RJ-11 padrão comumente utilizado para fiação telefônica, mas um RJ-45 tem oito pinos (um RJ-11 tem apenas quatro).[3] Em muitas situações, os conectores RJ-45 facilitam a administração das comunicações seriais por meio da planta de cabeamento existente do seu edifício (que provavelmente foi instalado tendo um par trançado Ethernet em mente).

Ilustração E Um conector RJ-45 macho

Os plugues RJ-45 para conexões seriais normalmente não são encontrados em computadores ou equipamentos seriais diversos, mas são freqüentemente utilizados como conectores intermediários para rotear linhas seriais por meio de painéis de conexão. Os RJ-45 são algumas vezes utilizados com cabo telefônico simples em vez de pares trançados. Qualquer uma das formas de cabo é aceitável para conexões seriais, embora o cabo de par trançado geralmente resulte em melhor qualidade de sinal a certa distância. Cabo de telefone simples nunca é aceitável para conexões Ethernet, um grande número de instalações adotou o padrão de cabo de par trançado para reduzir o risco de que um usuário que não saiba lidar com cabos selecione o tipo de cabo errado acidentalmente.

Os RJ-45 são compactos, seguros e baratos. Eles são crimpados no cabo com uma ferramenta especial. Leva menos do que um minuto para fixar um. Caso esteja projetando um sistema de cabeamento grande da estaca zero, os RJ-45 são uma boa opção para conectores intermediários.

Vários sistemas mapeiam os pinos em um conector RJ-45 para um DB-25. A Tabela 27.4 a seguir mostra o padrão RS-232D oficial, que só é utilizado casualmente.

Uma maneira alternativa de empregar RJ-45 é por meio do sistema de Dave Yost, que adiciona um soquete RJ-45 para cada dispositivo e utiliza somente um único tipo de cabo conector RJ-45 independentemente de os dispositivos serem DTEs ou DCEs. O sistema de Dave é mais do que simplesmente um plugue com pinos; ele é um sistema de cabeamento completo que atende a vários problemas diferentes. Com a gentil permissão dele, a descrição do padrão de Dave é reproduzida na próxima seção.

---

3 Uma análise cuidadosa revelará que um plugue RJ-11 na verdade possui seis aberturas nas quais poderíamos ter pinos, mas que somente quatro condutores estão instalados. Um verdadeiro conector telefônico de quatro condutores é um RJ-10. Um RJ-11 com todos os seis pinos instalados é conhecido como RJ-12.

Tabela 27.4 Pinos para um cabo direto RJ-45 para DB-25

RJ-45	DB-25	Sinal	Função
1	6	DSR	Pronto para dados
2	8	DCD	Detecção de portadora de dados
3	20	DTR	Pronto para receber uma transmissão
4	7	SG	Sinal terra
5	3	RD	Dados recebidos
6	2	TD	Dados transmitidos
7	5	CTS	Transmissão pode prosseguir
8	4	RTS	Sinal de solicitação de envio

O padrão Yost para cabeamento RJ-45

Essa especificação foi escrita por Dave Yost (Dave@Yost.com).

Eis um esquema que oferece soluções para vários problemas envolvendo RS-232:

- Todos os conectores de cabos devem ser do mesmo tipo e sexo (macho RJ-45)
- Não há nenhuma distinção entre DTE e DCE
- Você precisa de apenas um tipo de cabo conector
- Você pode produzir cabos em massa rapidamente, utilizando somente um crimpador

Cada porta serial em cada equipamento obtém seu adaptador DB-25 ou DB-9 para RJ-45 com a fiação apropriada. Esse adaptador é permanentemente aparafusado na porta. A porta agora apresenta a mesma interface conectora, RJ-45 fêmeo, independentemente de seu conector implícito ser DB-25 ou DB-9, DTE ou DCE, macho ou fêmea. Além disso, toda porta serial agora transmite e recebe dados nos mesmos pinos.

Assim que tiver colocado esses adaptadores em suas portas RS-232, poderá conectar qualquer coisa com qualquer coisa sem utilizar modems nulos ou terminais nulos, mudar pinos em conectores de cabos ou construir cabos especiais. Podemos conectar um modem a um computador, modem a terminais, terminais a computadores, terminais a terminais, computador a computador etc., tudo isso com apenas um tipo de cabo.

Na visão original de Yost, os cabos são de fita e têm 8 fios. Os conectores em cada extremidade são fixados no cabo com um crimpador, de forma que não há nenhum tipo de soldagem ou de confusão com os pinos. Por causa da aversão a esse tipo de cabo, algumas empresas utilizam o padrão Yost com o cabeamento UTP.

Há três fios de sinais (um de dados e dois de controle) indo em cada direção, mais um para sinais terra. Os cabos não são conectados normalmente (isto é, com cada pino do conector ligado ao pino correspondente no outro lado do cabo). Eles são ligados "cruzados", "espelhados" ou "invertidos", ou seja lá o que você chame isso. Isto é, o pino 1 no final do cabo se liga ao pino 8 na outra extremidade etc. Esse esquema funciona porque o layout dos sinais no cabo de fita é simétrico. Isto é, cada pino de transmissão tem o seu pino de recepção correspondente na posição inversa do cabo flat.[4]

Cabos RJ-45 prontos são em geral ligados diretamente. Para usá-los com esse sistema, você tem de remover o conector de uma extremidade e crimpá-la em uma nova extremidade com os fios invertidos. Conectores RJ-45 fêmea-fêmea (butt blocks) são disponíveis para estender cabeamento, mas lembre-se do seguinte: dois cabos trançados unidos com tal acoplador criam um cabo direto.

Muitos fornecedores fabricam adaptadores DB-25 para RJ-45. A codificação de cores interna não coincide com as cores do cabo. Os adaptadores, os conectores de fios e a fiação se tornaram disponíveis nas lojas de eletrônica, infelizmente sem qualquer ajuda sobre como utilizá-los para RS-232.

Consulte a página 247 para informações adicionais sobre o cabo Categoria 5.

O esquema Yost foi projetado para uso com cabo de fita, no qual todos os fios estão lado a lado. Os cabos de par trançado, ao contrário, têm quatro pares de fios, e cada par é trançado ao longo do comprimento do cabo. Se você utiliza cabos de par trançado (como o cabo Categoria 5), não deve ligar seus cabos como normalmente faria para o RJ-45 (por exemplo, para 10BaseT, telefone etc.). Em vez disso, você deve ligá-los de modo que os fios 3:4 e fios 5:6 formem pares. Outros pareamentos são suscetíveis à conversação cruzada do sinal de dados. O pareamento dos fios restantes não é importante, mas 1:2 e 7:8 serão tão bons quanto qualquer outro.

Dentro de um adaptador temos um soquete RJ-45 com oito fios saindo dele. Esses fios têm pinos RS-232 (ou soquetes de pinos, conforme apropriado) crimpados neles. Simplesmente empurramos esses pinos para os orifícios apropriados no conector RS-232 e depois fechamos o alojamento do adaptador. Utilize as pinagens mostradas na Tabela 27.5.

---

4 Dave não diz isso explicitamente, mas, se você estiver utilizando um cabo fita, na realidade você deve usar o cabo sem um cruzamento para alcançar o efeito "cruzado". Da maneira como são construídos, sua numeração de pinos é automaticamente invertida.

Tabela 27.5 Fiação para um adaptador Yost RJ-45 para DB-25 ou DB-9

Cabo RJ-45			Adaptador	Conexão com pinos DCE DB-25	DB-9	Sinal	Conexão com pinos DTE DB-25	DB-9	Sinal
1	Marrom	(para Cinza)	Azul	4	7	RTS	5	8	CTS
2	Azul	(para Laranja)	Laranja	20	4	DTR	8	1	DCD
3	Amarelo	(para Preto)	Preto	2	3	TD	3	2	RD
4	Verde	(para Vermelho)	Vermelho	7	5	GND	7	5	GND
5	Vermelho	(para Verde)	Verde	7	5	GND	7	5	GND
6	Preto	(para Amarelo)	Amarelo	3	2	RD	2	3	TD
7	Laranja	(para Azul)	Marrom	8	1	DCD	20	4	DTR
8	Cinza	(para Marrom)	Branco	5	8	CTS	4	7	RTS

Mas há um problema: ambos os pinos terra devem ser encaixados no mesmo orifício DB-25 ou DB-9 (pino 7 ou 5, respectivamente). Para que esses fios formem um único pino, você pode crimpar esse fios e utilizar uma pequena peça plástica produzida pela AMP chamada "Tel-splice connector ½ tap dry", número da peça 553017-4. Até agora, essa peça aparentemente só está disponível em volumes de mil peças por aproximadamente US$ 80.

Alguns dispositivos DTE requerem que o sinal DSR esteja ativo antes de eles enviarem dados. Esse sinal é usualmente fornecido pelo dispositivo DCE, mas você pode simulá-lo ligando os pinos 20 e 6 (4 e 6 em um conector DB-9). Dessa maneira, o dispositivo DTE receberá o sinal DSR dele próprio sempre que ele clamar DTR.

Em algumas impressoras DCE, o pino 7 do adaptador RJ-45 (o fio marrom) deve ser conectado à linha DSR (pino 6 tanto no DB-25 como no DB-9). Leia a documentação de sua impressora para descobrir se ela fornece sinais de handshaking úteis no DSR em vez de DCD.

Dave Yost. Los Altos, CA; julho de 1999.

Não utilize nenhuma dessas conexões DSR como uma seqüência lógica e natural; adicione-as apenas a dispositivos específicos que pareçam precisar delas (ou que não pareçam operar com a configuração padrão). Alguns dispositivos comuns toleram as conexões extras, mas outros ficam confusos.

27.3 PORTADORA VIA HARDWARE E SOFTWARE

O Linux espera que o sinal DCD, detecção de portadora, fique alto (tensão positiva) quando um dispositivo é conectado e ligado. Esse sinal é conduzido no pino 8 do conector DB-25 padrão. Se seu cabo serial tiver uma linha DCD e seu computador realmente prestar atenção a isso, você estará usando o que é conhecido como uma portadora via hardware. A maioria dos sistemas também permite uma portadora via software, isto é, o computador simula que a DCD sempre está declarada.

Para determinados dispositivos (particularmente terminais), a portadora via software é uma grande dádiva. Você pode escapar das conseqüências do uso de apenas três linhas para cada conexão serial: transmitir, receber e sinal terra. Entretanto, as conexões de modem realmente precisam do sinal DCD. Se um terminal está conectado por meio de um modem e o sinal de portadora for perdido, o modem deve suspender (especialmente em uma chamada interurbana!).

Normalmente especificamos portadora via software para uma porta serial no arquivo de configuração para qualquer software cliente que estivermos utilizando em conjunto com a porta (por exemplo, **/etc/gettydefs** ou **/etc/inittab** para um terminal de login ou **/etc/printcap** para uma impressora). Também podemos utilizar **stty -clocal** para ativar a portadora via software no processo. Por exemplo,

```
# stty -clocal < /dev/ttyS1
```

ativa a portadora via software para a porta **ttyS1**.

27.4 CONTROLE DE FLUXO VIA HARDWARE

Os sinais CTS e RTS garantem que um dispositivo não envie dados mais rápido do que o receptor é capaz de processar. Por exemplo, se um modem está prestes a esgotar seu espaço de buffer (talvez porque a conexão com a instalação remota é mais lenta do que o link serial entre a máquina local e o modem), ele poderá instruir o computador a se calar até que haja mais espaço disponível no buffer.

O controle de fluxo é essencial para modems de alta velocidade e também é muito útil para impressoras. Em sistemas que não suportam controle de fluxo via hardware (seja porque as portas seriais não o entendem ou porque o cabo serial deixa os sinais CTS e RTS desconectados), o controle de fluxo pode algumas vezes ser simulado em software com os caracteres ASCII XON e XOFF. Entretanto, o controle de fluxo via software tem de ser explicitamente suportado por software de alto nível e mesmo assim ele não funciona bem.

XON e XOFF são, respectivamente, <Control-Q> e <Control-S>. Isto é um problema para usuários do **emacs**, pois <Control-S> é a associação de teclas padrão para o comando de pesquisa do **emacs**. Para resolver esse problema, associe o comando de pesquisa a alguma outra tecla ou utilize **stty start** e **stty stop** para modificar o significado de XON e XOFF para o driver de terminal.

A maioria dos terminais ignora os sinais CTS e RTS. Jumpeando os pinos 4 e 5 no final do cabo do terminal, você pode enganar os poucos terminais que exigem um handshake por meio desses pinos antes de eles se comunicarem. Quando o terminal envia um sinal no pino 4 dizendo "Estou pronto", ele obtém o mesmo sinal de volta no pino 5 dizendo "Vá em frente". Você também pode jumpear o handshake do DTR/ DSR/DCD dessa maneira.

Como acontece com a portadora via software, o controle de fluxo via hardware pode ser manipulado por arquivos de configuração ou configurados via **stty**.

27.5 COMPRIMENTO DO CABO

O padrão RS-232 especifica um comprimento máximo de cabo de 22,5 m em uma velocidade de 9.600 bps. Os padrões são normalmente conservadores e o RS-232 não é nenhuma exceção. Utilizamos rotineiramente cabos RS-232 em distâncias muito maiores, até cerca de 300 m. Alcançamos o limite em torno de 240 m a 300 m, mas descobrimos que a marca específica dos dispositivos em cada ponta faz uma grande diferença.

Os drivers de linha ou repetidores são utilizados algumas vezes com RS-232 para se atingir o comprimento máximo de uma conexão. Infelizmente, esses dispositivos normalmente só amplificam os pinos RD e TD; portanto, outros sinais talvez precisem ser jumpeados.

27.6 ARQUIVOS DE DISPOSITIVOS SERIAIS

As portas seriais são representadas pelos arquivos de dispositivos em ou sob **/dev**. A maioria dos computadores tem duas portas seriais: **/dev/ttyS0** e **/dev/ttyS1**. As distribuições do Linux em geral pré-instalam um conjunto completo de arquivos de dispositivos (64 ou mais) para portas seriais adicionais, mas até que adicionemos mais hardware ao sistema, os arquivos extras são supérfluos e devem ser ignorados.

Como sempre, os nomes dos arquivos de dispositivos realmente não importam. O mapeamento de dispositivo é determinado pelos números de dispositivos principais e secundários, e os nomes de dispositivos são meramente uma conveniência para usuários humanos.

27.7 SETSERIAL: CONFIGURANDO PARÂMETROS DE PORTA SERIAL

As portas seriais em um PC podem ser configuradas para utilizar vários endereços de portas de E/S e níveis de interrupção (IRQs) diferentes. Essas configurações são normalmente acessadas por meio da BIOS do sistema no momento em que a máquina é ligada. A razão mais comum para alterá-las é a de acomodar algum hardware excêntrico que é caprichoso em relação às suas próprias configurações e somente funcionará de forma correta quando tiver cooptado as configurações normalmente utilizadas por uma porta serial. Infelizmente, talvez o driver serial não seja capaz de detectar essas mudanças de configuração sem a sua ajuda.

A resposta de sistemas UNIX tradicionais a essa diversidade é permitir que sejam especificados parâmetros de porta serial quando o kernel é compilado. Felizmente, o Linux permite pular esse passo entediante e alterar os parâmetros no processo com o comando **setserial**.

setserial -g mostra as configurações atuais:

```
# setserial -g /dev/ttyS0
/dev/ttyS0, UART: 16550A, Port: 0x03f8, IRQ: 4
```

Para configurar os parâmetros, você especifica o arquivo de dispositivos e depois uma série de parâmetros e valores. Por exemplo, o comando:

```
# setserial /dev/ttyS1 port 0x02f8 irq 3
```

configura o endereço de porta de E/S e a IRQ para **ttyS1**. É importante ter em mente que esse comando não altera de maneira alguma a configuração de hardware; ele simplesmente informa o driver serial Linux da configuração. Para alterar realmente as configurações de hardware, consulte a BIOS de seu sistema.

setserial altera somente a configuração atual, e as configurações não persistem após reinicializações. Infelizmente, não há uma maneira padrão de tornarmos permanentes as modificações; cada uma de nossas distribuições de exemplo o faz de forma diferente.

O script **/etc/rc.d/rc.sysinit** do Red Hat e do Fedora verifica se existe **/etc/rc.serial** e o executa no momento da inicialização caso ele exista. Não é fornecido nenhum modelo de arquivo, de modo que você mesmo tem de criar o arquivo caso queira utilizar esse recurso. Basta listar os comandos **setserial** que você quer executar, um em cada linha. Para ser completo, provavelmente é uma boa idéia tornar o arquivo executável e colocar **#!/bin/sh** na primeira linha; entretanto, esses *toques de elegância* não são estritamente exigidos.

O script **/etc/init.d/serial** do SUSE trata a inicialização da porta serial. Infelizmente, esse script não tem nenhum arquivo de configuração; você tem de editá-lo diretamente para refletir os comandos que deseja executar. SuSE malvado! O script utiliza sua própria pequena metalinguagem para construir as linhas de comando **setserial**, mas felizmente há uma série de linhas de exemplos comentadas para podermos escolher.

O Debian fornece um arquivo de configuração gentilmente comentado, **/etc/serial.conf**, que é lido por **/etc/init.d/setserial**. Esse arquivo também está disponível para o Ubuntu no pacote **setserial**, mas não é instalado por padrão. Os recursos avançados desse pacote (como autoconfiguração persistente) talvez só sejam úteis para os projetistas da distribuição. Para casos simples, basta retirar os comentários (#) das linhas que você quiser ou então adicionar suas próprias linhas no formato **setserial** (omita o nome do comando).

27.8 CONFIGURAÇÃO DE SOFTWARE PARA DISPOSITIVOS SERIAIS

Assim que um dispositivo tiver sido conectado ao cabo apropriado, o software na máquina host deve ser configurado para tirar proveito dele. As tarefas de configuração para um dispositivo novo dependem do tipo de dispositivo e dos usos que ele terá:

- Para um terminal cabeado, temos de instruir o sistema a ouvir logins na porta do terminal. Especificamos a velocidade e os parâmetros da conexão serial. A configuração para terminais é descrita na próxima seção.
- Modems de discagem são configurados de forma similar aos terminais cabeados. Entretanto, o procedimento exato pode ser ligeiramente diferente em alguns sistemas.
- Para configurar um modem de discagem para utilização por humanos, você deve colocar entradas em **/etc/remote** para utilização pelos comandos **tip** e **cu**. Esse procedimento é descrito na página 593.
- Para ver como utilizar um modem a fim de se conectar a uma rede remota via PPP, consulte a página 223.
- Consulte o Capítulo 23, *Impressão*, para informações adicionais sobre como configurar uma impressora serial. Algumas impressoras somente recebem dados; outras são bidirecionais e podem retornar informações de estado para o computador host.
- Um dispositivo serial personalizado que você utiliza somente a partir de seu próprio software não necessita de nenhuma configuração especial. Você pode simplesmente abrir o arquivo de dispositivos para acessar o dispositivo. Consulte a página man **termios** para aprender a configurar a velocidade, bits de flag e modo de buffering de uma porta serial.

27.9 CONFIGURAÇÃO DE TERMINAIS CABEADOS

Ao longo das duas últimas décadas, computadores baratos substituíram praticamente todos os terminais ASCII. Entretanto, mesmo as janelas de "terminal" numa tela gráfica usam os mesmos drivers e arquivos de configuração do que os verdadeiros terminais, de forma que os administradores ainda precisam entender como essa tecnologia arcaica funciona.

A configuração de terminais envolve duas tarefas principais: certificar-se de que um processo está conectado a um terminal para aceitar logins e de que informações sobre o terminal estão disponíveis assim que um usuário tiver se conectado.

O processo de login

O processo de login envolve vários programas diferentes.

Consulte a página 23 para informações adicionais sobre o daemon init.

O processo de login envolve vários programas diferentes, sendo o mais importante deles o daemon **init**. Uma das tarefas de **init** é gerar um processo, conhecido genericamente como **getty**, em cada porta de terminal ativada no arquivo **/etc/ttys** ou **/etc/inittab**. **getty** configura as características iniciais da porta (como velocidade e paridade) e imprime um prompt de login.

O verdadeiro nome do programa **getty** varia entre as distribuições Linux, e algumas distribuições incluem várias implementações. O Red Hat, o Fedora e o SUSE usam uma versão simplificada chamada **mingetty** para lidar com logins em consoles virtuais. Para gerenciar terminais e modems discados, esses arquivos fornecem a implementação **mgetty** de Gert Doering. O Debian e o Ubuntu usam um único **getty** escrito por Wietse Venema e outros; essa versão também está disponível nos sistemas SUSE sob o nome **agetty**. Uma implementação mais antiga chamada **uugetty** foi suplantada de longe por **mgetty**.

Para fazer a distinção entre essa plenitude de **gettys**, imagine-os em ordem de complexidade. **mingetty** é o mais simples e é basicamente um substituto para **getty**. Ele é capaz de tratar logins somente em consoles Linux virtuais. **agetty** é um pouco mais aprimorado e trata tanto de portas seriais como de modems. **mgetty** é atualmente o mais popular. Ele trata recebimento de fax e logins, e executa o bloqueio e coordenação apropriados para que o mesmo modem possa ser utilizado tanto para receber como para fazer chamadas.

A seqüência de eventos em um login completo é a seguinte:

- **getty** imprime o conteúdo do arquivo **/etc/issue** juntamente com um prompt de login
- Um usuário introduz um nome de login no prompt de **getty**
- **getty** executa o programa **login** com o nome especificado como argumento
- **login** solicita uma senha e a valida contra o arquivo **/etc/shadow**[5]
- **login** imprime a mensagem do dia a partir de **/etc/motd** e executa um shell
- O shell executa os arquivos de inicialização apropriados[6]
- O shell imprime um prompt e aguarda entrada por parte do usuário

Quando o usuário sai do sistema, o controle retorna para **init**, que desperta e gera um novo **getty** na porta do terminal.

A maioria das configurações nessa cadeia de eventos se concentra em **/etc/inittab**, onde o complemento normal do sistema de **gettys** é definido, e em **/etc/gettydefs**, onde algumas versões de **getty** procuram informações adicionais de configuração.

O arquivo /etc/inittab

init suporta vários "níveis de execução" que determinam quais recursos de sistema são ativados. Há sete níveis de execução numerados de 0 a 6, com "s" reconhecido como sinônimo para o nível 1 (operação monousuário). Ao deixar o modo monousuário, **init** lhe solicita para entrar em um nível de execução, a menos que exista um campo `initdefault` em **/etc/inittab** conforme descrito abaixo. **init** varre então o arquivo **inittab** em busca de todas as linhas que atendem ao nível de execução especificado.

Normalmente, os níveis de execução são configurados de modo que você tenha um nível em que apenas o console é ativado e um outro nível que ativa todos os terminais **getty**. Você pode definir os níveis de execução da maneira mais apropriada ao seu sistema; mas recomendamos que você não se desvie muito dos padrões.

Entradas em **inittab** têm a forma:

```
id:níveis-de-execução:ação:processo
```

Eis alguns exemplos simples de entradas **inittab**.

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
```

Nesse formato, *id* é uma string de um ou dois caracteres que identifica a entrada; ela pode ser nula. Para entradas de terminais, costuma-se utilizar o número do terminal como o *id*.

níves-de-execução enumera os níveis de execução ao qual a entrada pertence. Se não for especificado nenhum nível de execução (como acontece na primeira linha), então a entrada é válida para todos os níveis de execução. *ação* informa como manipular o campo *processo*; a Tabela 27.6 lista alguns dos valores mais comumente utilizados.

---

5 Se as senhas sombra não estiverem sendo utilizadas, a senha poderá provir diretamente de **/etc/passwd**. Além disso, **/etc/passwd** poderia ser substituído ou complementado por um sistema de banco de dados administrativo como o NIS. Consulte o Capítulo 17 para informações adicionais.

6 **.profile** para **sh** e **ksh**; **.bash_profile** e **.bashrc** para **bash**; **.cshrc** e **.login** para **csh/tcsh**.

Tabela 27.7 Valores comuns para o campo *ação* de /etc/inittab

Valor	Esperar?	Significado
`initdefault`	–	Configura o nível de execução inicial
`boot`	Não	Executado quando **inittab** é lido pela primeira vez
`bootwait`	Sim	Executado quando **inittab** é lido pela primeira vez
`ctrlaltdel`	Não	Executado em resposta ao acionamento das teclas <Control-Alt-Delete>
`once`	Não	Inicia o processo uma vez
`wait`	Sim	Inicia o processo uma vez
`respawn`	Não	Sempre mantém o processo em execução
`powerfail`	Não	Executado quando **init** recebe um sinal de falha de energia
`powerwait`	Sim	Executado quando **init** recebe um sinal de falha de energia
`sysinit`	Sim	Executado antes de se acessar a o console

Se um dos *níveis-de-execução* coincidir com o nível de execução atual e o campo *ação* indicar que a entrada é relevante, **init** utilizará **sh** para executar (ou terminar) o comando especificado no campo *processo*. A coluna Esperar? da Tabela 27.6 informa se **init** aguardará pelo término do comando antes de prosseguir.

Nas linhas **inittab** do exemplo anterior, as duas últimas linhas geram processos **mingetty** nos primeiros dois consoles virtuais (acessado com <Alt-F1> e <Alt-F2>). Se você adicionar terminais cabeados ou modems de discagem, as linhas **inittab** apropriadas serão semelhantes a essas. Mas você tem de utilizar **mgetty** ou **getty** (**agetty** no SUSE) com esses dispositivos, pois **mingetty** não é sofisticado o bastante para lidar com eles corretamente. Em geral, `respawn` é a ação correta e 2345 é um conjunto de níveis apropriado.

O comando **telinit -q** faz com que **init** releia o arquivo **inittab**.

Diferentes **gettys** exigem diferentes procedimentos de configuração. A versão **getty/agetty** encontrada em SUSE, Debian e Ubuntu geralmente é um pouco mais limpa do que **mgetty**, pois aceita todas as informações de configuração na linha de comando (em **/etc/inittab**).

O modelo genérico é:

```
/sbin/getty porta velocidade tipo-de-terminal
```

Consulte a página 590 para informações adicionais sobre o banco de dados terminfo.

onde *porta* é o arquivo de dispositivos da porta serial relativa a **/dev**, *velocidade* é a taxa de transmissão (por exemplo, 38400) e *tipo-de-terminal* identifica o tipo de terminal padrão para a porta. *tipo-de-terminal* refere-se a uma entrada no banco de dados **terminfo**. A maioria dos emuladores simula um terminal VT100 da DEC, denotado por **vt100**. A maioria das várias outras opções menos importantes está relacionada ao tratamento de modems discados.

mgetty fornecido pelo Red Hat, Fedora e SUSE é um pouco mais sofisticado do que **agetty** em seu tratamento com os modems e integra uma capacidade tanto para receber como para transmitir faxes. Infelizmente, sua configuração é um pouco mais difusa. Além de outros flags de linha de comando, **mgetty** pode aceitar uma referência opcional a uma entrada em **/etc/gettydefs** que especifica detalhes de configuração para o driver serial. A menos que esteja configurando uma configuração de modem sofisticada, normalmente você pode virar sem uma entrada **gettydefs**.

Em sistemas Red Hat, use **man mgettydefs** para localizar a página man para o arquivo **gettydefs**. Ele é chamado assim para evitar conflitos com uma página man **gettydefs** mais antiga que já não existe mais em nenhum sistema Linux.

Uma linha de comando **mgetty** simples para um terminal cabeado se parece com isto:

```
/sbin/mgetty -rs velocidade dispositivo
```

velocidade é a taxa de transmissão (por exemplo, 38400) e *dispositivo* é o arquivo de dispositivos para a porta serial (utilize o nome de caminho completo).

Se quiser especificar um tipo de terminal padrão para uma porta ao utilizar **mgetty**, você tem de fazer assim em um arquivo distinto, **/etc/ttytype**, e não na linha de comando **mgetty**. O formato de uma entrada em **ttytype** é:

```
tipo-de-terminal dispositivo
```

dispositivo é o nome abreviado para o arquivo de dispositivos representando a porta e *tipo-de-terminal* é o nome da entrada **terminfo** apropriada (consulte a próxima seção). Por exemplo:

```
linux      tty1
linux      tty2
vt100      ttyS0
vt100      ttyS1
```

Suporte a terminais: os bancos de dados termcap e terminfo

O Linux suporta muitos diferentes tipos de terminais auxiliados por um banco de dados de capacidades de terminal que especifica os recursos e as peculiaridades da programação de cada marca de terminal. Há, historicamente, dois tipos de formatos de bancos de dados concorrentes: **termcap** e **terminfo**. Para maior compatibilidade, as distribuições do Linux geralmente fornecem ambos. O banco de dados **termcap** é contido no arquivo **/etc/termcap** e o banco de dados **terminfo** é armazenado em **/usr/share/terminfo**. Os dois bancos de dados são semelhantes e utilizam o mesmo nome para cada tipo de terminal, de modo que a distinção entre eles não é importante.

Da forma como são distribuídos, ambos os bancos de dados contêm entradas para centenas de terminais diferentes. Nesta era em que os terminais estão em extinção, a maioria deles é irrelevante. Uma boa regra prática é que tudo emula um terminal VT100 até prova em contrário. Muitos emuladores também suportam "ansi"; "linux" e "xterm" são úteis para, respectivamente, consoles Linux e janelas **xterm** (terminal X Windows).

Consulte a página 591 para informações adicionais sobre como configurar terminais em tempo de login.

Os programas Linux examinam a variável de ambiente TERM para determinar o tipo de terminal que você está utilizando. O terminal pode então ser pesquisado em **termcap** ou **terminfo**. Alternativamente, a entrada **termcap** de terminal pode ser colocada diretamente no ambiente sob a variável de ambiente TERMCAP. Os usuários normalmente planejam para que suas variáveis TERMCAP e TERM sejam configuradas em tempo de login. Normalmente o sistema configura a variável TERM para nós no momento do login, de acordo com os argumentos da linha de comandos para **getty/agetty** ou com o conteúdo de **/etc/ttytype**.

27.10 CARACTERES ESPECIAIS E O DRIVER DE TERMINAL

O driver de terminal suporta várias funções especiais que você pode acessar digitando determinadas teclas (geralmente as teclas de controle). A associação exata de funções às teclas pode ser configurada por meio dos comandos **tset** e **stty**. A Tabela 27.7 lista algumas dessas funções, juntamente com suas associações de teclas básicas.

Tabela 27.7 Caracteres especiais para o driver de terminal

Nome	Padrão	Função
erase	<Control-?>	Apaga um caractere da entrada
werase	<Control-W>	Apaga uma palavra da entrada
kill	<Control-U>	Apaga a linha inteira da entrada
eof	<Control-D>	Envia uma indicação "EOF" (final de arquivo)
intr	<Control-C>	Interrompe o processo que está em execução no momento
quit	<Control-\>	Encerra o processo atual com um dump de núcleo
stop	<Control-S>	Pára a saída da tela
start	<Control-Q>	Reinicia a saída para a tela
susp	<Control-Z>	Suspende o processo atual
lnext	<Control-V>	Interpreta o próximo caractere literalmente

Por padrão, as versões PC do kernel do Linux geram um caractere Delete (<Control-?>) quando a tecla Backspace é pressionada. (Essa tecla pode estar identificada como "backspace" ou "delete" ou poderá então apresentar apenas o desenho de uma seta invertida. Depende do teclado.) No passado, muitos sistemas UNIX usavam o caractere backspace (<Control-H>) para essa função. Infelizmente, a existência de dois padrões para a função cria uma série de problemas.

Você pode utilizar **stty erase** (consulte a próxima seção) para informar ao driver de terminal qual código de tecla sua configuração está realmente gerando. Entretanto, alguns programas (como os editores de texto e shells com recursos de edição de comandos) têm uma concepção própria do que representa o caractere backspace e nem sempre eles prestam atenção à configuração do driver de terminal. Em um truque útil, mas confuso, alguns programas obedecem tanto ao caractere backspace como ao caractere delete. Você também poderá descobrir que sistemas remotos nos quais você faz o login por meio da rede fazem suposições completamente diferentes daquelas do seu sistema local.

Solucionar esses pequenos conflitos pode ser por si só uma empreitada para um fim de semana inteiro. Geralmente não há nenhuma solução simples e universal. Cada pacote de software deve ser avaliado individualmente. Dois recursos úteis que ajudam nessa tarefa são *Linux Backspace/Delete mini-HOWTO*, disponível em www.tldp.org, e um excelente artigo de Anne Baretta, disponível em www.ibb.net/~anne/keyboard.html.

27.11 STTY: OPÇÕES DE CONFIGURAÇÃO DE TERMINAL

stty permite alterar diretamente e consultar as diversas configurações do driver de terminal. Há zilhões de opções; estas estão documentados na página man para o driver **tty** (normalmente **tty**(4) ou **tty**(5), não **tty**(1), que é um programa simples que informa o terminal ou pseudoterminal em que você está conectado).

As opções de linha de comando do **stty** podem aparecer em qualquer ordem e em qualquer combinação. Um traço antes de uma opção, a nega. Por exemplo, para configurar um terminal para operação em 9600 bps com paridade par e sem tabulações de hardware, utilize o seguinte comando:

```
$ stty 9600 even -tabs
```

Uma boa combinação de opções para ser utilizada em um terminal típico seria:

```
$ stty intr ^C kill ^U erase ^H ? -tabs
```

Aqui, **-tabs** impede que o driver de terminal tire proveito do mecanismo de tabulação embutido do terminal, o que é útil, pois vários emuladores não são muito inteligentes quanto às tabulações. As demais opções configuram os caracteres de interromper, terminar e apagar associados, respectivamente, a <Control-C>, <Control-U> e a <Control-?> (delete).

Você pode utilizar **stty** para examinar os modos atuais do driver de terminal, bem como para configurá-los. **stty** sem nenhum argumento gera uma saída como a mostrada a seguir:

```
$ stty
speed 38400 baud; line = 0;
-brkint -imaxbel
```

Para um relatório de estado mais prolixo, use a opção **-a**:

```
$ stty -a
speed 38400 baud; rows 50; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
```

O formato da saída é similar, mas apresenta um número maior de informações. O significado da saída deveria ser intuitivamente óbvio (caso você tenha escrito um driver de terminal recentemente).

stty opera no descritor de arquivo da entrada padrão ou da saída padrão (dependendo do seu sistema), assim você pode configurar e consultar os modos de um terminal, além do atual, utilizando os caracteres de redireção de shell ">", " <" e o caractere de redireção de entrada do shell (<). Na maioria dos sistemas, você deve ser o superusuário para alterar os modos do terminal de uma outra pessoa.

27.12 TSET: CONFIGURANDO OPÇÕES AUTOMATICAMENTE

tset inicializa o driver de terminal para um modo apropriado para um determinado tipo de terminal. O tipo pode ser especificado na linha de comandos; se ele não for especificado, **tset** utiliza o valor da variável de ambiente TERM.

tset suporta uma sintaxe para mapeamento de valores da variável de ambiente TERM a outros valores. Esse recurso é útil caso você entre freqüentemente no sistema via modem ou um switch de dados e quiser o driver de terminal configurado corretamente para o terminal que você está de fato usando na outra ponta da conexão em vez de algo genérico e inútil como "discado".

Suponha, por exemplo, que você utilize **xterm** em casa e que o sistema para o qual você está discando seja configurado para pensar que o tipo de terminal de um modem é "discado". Adicionar o comando

```
tset -m dialup:xterm
```

ao seu arquivo **.login** ou **.profile** configura o driver de terminal apropriadamente para o **xterm** sempre que você se conecta.

Infelizmente, o comando **tset** não é realmente tão simples quanto parece ser. Para fazer com que **tset** ajuste as variáveis de ambiente além de configurar os modos de terminal, precisaremos de algo como o seguinte:

```
set noglob
eval `tset -s -Q -m dialup:xterm`
unset noglob
```

Essa mágica suprime as mensagens que **tset** normalmente envia (o flag **-Q**) e solicita que os comandos de shell configurem para que as variáveis de ambiente sejam geradas como saída em seu lugar (o flag **-s**). Os comandos de shell emitidos por **tset** são capturados pelas aspas simples invertidas e alimentados para o shell como entrada por meio do comando embutido **eval** fazendo com que os comandos tenham o mesmo efeito do que se estivessem sendo digitados pelo usuário.

set noglob impede que o shell expanda quaisquer metacaracteres, como "*" e "?", que sejam incluídos na saída de **tset**. Esse comando não é necessário para usuários **sh/ksh** (nem o comando **unset noglob** para desfazê-lo), já que esses shells normalmente não expandem caracteres entre aspas simples invertidas. O comando **tset** é o mesmo independentemente de o shell utilizado; **tset** examina a variável de ambiente SHELL para determinar que tipo de comandos deve apresentar.

27.13 DESTRAVANDO TERMINAIS

Alguns programas (como o **vi**, por exemplo) promovem alterações drásticas no estado do driver de terminal enquanto estão sendo executados. Essa intervenção normalmente é invisível para o usuário, uma vez que o estado de terminal é cuidadosamente restaurado sempre que o programa é fechado ou suspenso. Mas um programa poderia travar ou ser eliminado sem realizar essa etapa de faxina. Quando isso acontece, o terminal pode se comportar de maneira muito estranha: ele pode falhar ao tratar linhas novas corretamente, ecoar caracteres digitados ou executar comandos de forma apropriada.

Uma outra maneira comum de confundir um terminal é executar acidentalmente **cat** ou **more** em um arquivo binário. A maioria dos arquivos binários contém uma mistura de caracteres especiais capazes de certamente mandar para o espaço emuladores menos robustos.

Para corrigir essa situação, utilize **reset** ou **stty sane**. Na verdade, **reset** é apenas um link para **tset** em muitos sistemas e pode aceitar a maioria dos argumentos do **tset**. Entretanto, ele normalmente é executado sem argumentos. Tanto **reset** como **stty sane** restauram o funcionamento correto do driver de terminal e enviam um código de reinicialização apropriado a partir de **termcap/terminfo**, caso um deles esteja disponível.

Em muitos casos para os quais um **reset** é apropriado, o terminal foi deixado em um modo no qual nenhum processamento estava sendo feito nos caracteres digitados pelo usuário. A maioria dos terminais gera retornos de carro em vez de caracteres de linha nova quando a tecla Return ou Enter é pressionada; sem processamento de entrada, essa tecla gera caracteres <Control-M> em vez de enviar o comando atual para ser executado. Para inserir novas linhas diretamente, use <Control-J> ou a tecla de quebra de linha (se houver uma) em vez da tecla Return.

27.14 MODEMS

Um modem converte o sinal serial digital produzido por um computador em um sinal análogo compatível para transmissão em uma linha telefônica padrão. Felizmente, o advento do acesso via banda larga à Internet transformou esses dispositivos, antigamente essenciais, em candidatos a museus em muitas localizações.

Modems externos têm um conector RJ-11 no lado analógico e uma interface RS-232 de algum tipo no lado digital — usualmente uma DB-25 fêmea. Na parte frontal eles normalmente têm uma série de leds que exibem o estado atual e nível de atividade do modem. Esses leds são incrivelmente úteis para depuração, de modo que os modems devem geralmente ser localizados em algum ponto de fácil visão.

Modems internos são normalmente vistos somente em PCs. Eles são encaixados em um slot ISA, PCI ou PCM-CIA e têm um conector RJ-11 que fica na parte externa traseira do gabinete do computador uma vez que o modem é instalado. Eles são mais baratos do que um modem externo, mas mais problemáticos para serem configurados e geralmente não têm os leds indicadores.

Se você estiver pensando em utilizar um modem interno, certifique-se de que ele é suportado pela sua versão do Linux. CPUs rápidas tornaram possível simplificar o hardware dos modems, executando algumas tarefas de processamento de sinais no processador do computador. Infelizmente, modems que funcionam dessa maneira (conhecidos genericamente como Winmodems) requerem drivers sofisticados e não são universalmente suportados sob as versões baseadas em PC do Linux. Consulte *Linmodem HOWTO*, de Sean Walbran e Marvin Stodolsky (disponível em www.tldp.org), para uma visão geral do suporte Winmodem sob o Linux. (O *Modem HOWTO* também é muito útil para uma perspectiva mais ampla sobre o gerenciamento de modems.)

Modems internos são normalmente feitos para parecer como se estivessem conectados por meio de uma porta serial fantasma sob a perspectiva do software em nível de usuário. Essa convenção ajuda a isolar a função lógica do modem de sua implementação de hardware. Pacotes de software padrão podem acionar o modem sem ter de saber nada sobre suas peculiaridades.

Os modems variam um pouco em termos de robustez geral, mas essa característica é difícil de ser avaliada sem experiência direta. No passado, constatamos que alguns modems eram significativamente mais tolerantes em termos de ruído de linha do que outros. Hoje, a maioria dos projetos utiliza um chipset padrão de um dos vários grandes fabricantes, de maneira que provavelmente as variações entre os modems não são tão grandes quanto foram no passado.

Os modems de alta velocidade requerem firmware complexo e esse firmware eventualmente apresenta bugs. Os fabricantes compartilham firmware entre modelos quando possível; portanto, uma linha inteira de produtos pode ter um bom firmware ou não. Por essa razão, ainda recomendamos utilizar as marcas mais conhecidas.

Modulação, correção de erros e protocolos de compactação de dados

Há muito tempo, era importante verificar os protocolos exatamente suportados por um modem, pois os padrões estavam em constante mudança, e os fabricantes de modems nem sempre implementavam um conjunto de protocolos completo. Atualmente, todos os modems suportam praticamente os mesmos padrões. A única diferença real entre eles está na qualidade do firmware, na eletrônica e no suporte.

A taxa de transmissão (*baud rate*) de um protocolo é a taxa em que o sinal de portadora é modulado. Se houver mais de dois níveis de sinal, mais do que um bloco de informações pode ser enviado por transição e a velocidade em bits por segundo será maior do que a taxa de transmissão. Historicamente, a velocidade dos dados e a velocidade de sinalização dos modems eram as mesmas, induzindo a uma casual confusão dos termos "baud" e "bps" (bits por segundo).

A maioria dos modems atual utiliza o padrão V.90 "56K" , o que não significa que ele forneça realmente uma capacidade de 56 Kb/s. Na melhor das hipóteses, ele possibilita uma taxa de 33,6 Kb/s do computador para o provedor de acesso Internet e de 53 Kb/s na direção oposta. Uma atualização recente ao padrão V.90, V. 92, envia 48 Kb/s em vez de 33,6 Kb/s ao provedor de acesso. O V.90 e o V.92 alcançam velocidades próximas aos limites teóricos e legais da sinalização por linhas telefônicas de voz comuns e não é esperado que esses padrões sejam substituídos tão cedo.

O ruído de linha pode introduzir um número significativo de erros em uma conexão de modem. Diversos protocolos de correção de erros foram desenvolvidos para empacotar os dados transmitidos e fornecer correção de erros baseados em técnicas de verificação de soma, isolando o usuário ou aplicativo das falhas da linha. Antigamente, você tinha de saber isso para poder configurar um modem corretamente, mas nos dias de hoje ele simplesmente funciona.

Os algoritmos de compactação de dados podem ser utilizados para dividir o número de bits que devem ser transmitidos entre extremidades analógicas. O nível de compactação varia de pior do que nada (quando se transmite dados que já foram compactados) até uma razão de 4:1. Um valor mais típico é de 1,5:1. Em geral, a configuração média apresenta um melhor desempenho quando algum desses algoritmos de compactação é ativado.

minicom: iniciando conexões

Os programas de discagem UNIX tradicionais **tip** e **cu** são relativamente impopulares nos sistemas Linux embora ambos tenham sido portados (**cu** é normalmente empacotado com UUCP, um sistema de comunicação telefônica obsoleto). Mais comum sob o Linux são pacotes abrangentes como **kermit** e **minicom** que fornecem emulação de terminais e suporte para protocolos de transferência de dados. Para depuração e uso ocasional, recomendamos o **minicom**, em grande parte porque é o que tem maior possibilidade de já estar pré-instalado. O Debian não o instala por padrão; execute **apt-get install minicom** se achar que ele está faltando.

Por sorte ou por azar, o **minicom** está mais para software tipo PC do que para Linux. Se suas memórias de 1986 incluírem conexão a 1.200 bauds por meio de um emulador de terminal MS-DOS, o **minicom** fará com que você se sinta em casa. Para configurar o software, execute **minicom -s** como root e entre no menu "serial port setup". Configure o arquivo de dispositivos para o modem, ative o controle de fluxo via hardware, configure a codificação para 8N1 (8 bits de dados, sem paridade, e 1 bit de parada) e certifique-se de que a velocidade esteja OK. Retorne para o menu principal e selecione "save settings as dfl" para gravar suas alterações.

Se estiver familiarizado com a linguagem de comandos Hayes utilizada pela maioria dos modems, você pode simplesmente introduzir os comandos diretamente (por exemplo, "ATDT5551212" para discar 555-1212). Para que o **minicom** faça isso para você, digite <Control-A> e D para entrar no menu de discagem.

Modems bidirecionais

Freqüentemente é útil utilizar um único modem tanto para iniciar como para receber chamadas, particularmente se quiser utilizar suporte a fax. Essa configuração exige que **getty** manipule a porta serial de uma maneira especialmente solícita, uma vez que ele não pode simplesmente cuidar da porta e bloquear todos os demais processos. O compartilhamento de portas é alcançado por meio de opções para a chamada de sistema **open** e envolve recursos tanto do driver serial como do kernel. Todos os programas que compartilham a porta devem obedecer ao protocolo apropriado.

No passado, configurar um modem para uso bidirecional era um grande problema que exigia muitos ajustes de sistema específicos e em geral não funcionava muito bem. Felizmente, os principais pacotes de software Linux se dão relativamente

bem uns com os outros da forma como vêm configurados originalmente desde que você utilize **mgetty** ou **uugetty**. Em geral, prefere-se **mgetty**; é o comportamento padrão para compartilhamento; portanto, basta você ligar e usar.

27.15 DEPURANDO UMA LINHA SERIAL

Depurar linhas seriais não é difícil. Alguns erros típicos são:

- Esquecer-se de informar a **init** para reler seus arquivos de configuração
- Esquecer-se de configurar a portadora via software ao utilizar cabos de três fios
- Usar um cabo com a nulidade incorreta
- Soldar ou crimpar conectores DB-25 de ponta-cabeça
- Conectar um dispositivo ao fio errado devido a associações de fiação incorretas ou inexistentes
- Configurar as opções de terminal incorretamente

Um dispositivo de interconexão (*breakout box*) é uma ferramenta indispensável para depurar problemas de cabeamento. Ele é conectado na linha serial e exibe os sinais em cada pino à medida que passam através do cabo. Os melhores dispositivos desse tipo têm tanto conectores macho como fêmea em cada lado e são totalmente flexíveis e bissexuais em seus posicionamentos. LEDs associados a cada pino "interessante" (pinos 2, 3, 4, 5, 6, 8 e 20) mostram quando o pino está ativo.

Alguns dispositivos de interconexão são somente para leitura e apenas permitem o monitoramento de sinais; outros permitem refazer a conexão e garantir uma determinada tensão em um dado pino. Por exemplo, se suspeitar que um cabo precisa ser anulado (cruzado), você pode utilizar o dispositivo de interconexão para anular a fiação real e inverter os pinos 2 e 3 e também os pinos 6 e 20.

Consulte a página 258 para informações adicionais sobre a Black Box.

Um dispositivo de interconexão ruim pode ser pior do que simplesmente não utilizar nenhum dispositivo de interconexão. Nossa implementação favorita é a BOB-CAT-B produzida pela Black Box. É um dispositivo fácil de utilizar que custa cerca de US$ 250. Para entrar em contato com a Black Box: (724)746-5500 ou www.blackbox.com.

27.16 OUTRAS PORTAS DE E/S COMUNS

Antigamente não havia concorrentes ao padrão de portas seriais para conectar periféricos de baixa velocidade aos sistemas UNIX, mas hoje em dia a maioria dos hardwares de PC é distribuída com portas USB (Universal Serial Bus) que servem a essa função.

Portas USB superam de longe as tradicionais portas seriais (e paralelas). Elas são mais rápidas (até 480 Mb/s) e arquitetonicamente elegantes e empregam cabos padronizados que são baratos e simples. O USB rapidamente tornou-se o padrão para a maioria dos dispositivos externos.

Como um ajuste extra, você pode comprar adaptadores com uma porta USB em uma das extremidades e uma porta serial RS-232C na outra. Esses adaptadores permitem que os PCs mais recentes sem portas seriais embutidas se comuniquem com dispositivos seriais legados. Tivemos um bom desempenho com os adaptadores USB-to-RS-232C vendidos pela www.keyspan.com. Os drivers para os adaptadores Keyspan são incluídos nas versões 2.4.22 e superiores do kernel do Linux.

As portas paralelas eram populares na década de 1980 e são semelhantes em conceito às portas seriais, mas transferem oito bits de dados ao mesmo tempo em vez de apenas um bit. As interfaces paralelas são tradicionalmente encontradas nas impressoras, mas no mundo Windows elas também foram historicamente utilizadas para conectar unidades Zip e unidades de fita. Esses usos bizarros foram há muito tempo substituídos pelo USB, e o suporte Linux a dispositivos paralelos que não impressoras é raro.

USB: o barramento serial universal

Para mais informações sobre USB, acesse o site www.usb.org.

O USB é um sistema de interconexão de periféricos genérico desenvolvido pela Compaq, DEC, IBM, Intel, Microsoft, NEC e Northern Telecom. O primeiro padrão USB foi publicado em 1996. A aceitação do USB no mundo Windows rapidamente cresceu como uma bola de neve ao longo da última década. Todos os novos PCs têm portas USB, e a maioria dos periféricos de computadores está disponível nas versões USB.

O USB é um excelente sistema e acreditamos que provavelmente ele permaneça em uso por muitos anos. Ele praticamente tem todas as propriedades e recursos que alguém poderia imaginar para um barramento de comunicações utilitário.

- É extremamente barato.
- Podem ser conectados até 127 dispositivos.
- Os cabos têm somente quatro fios: alimentação, terra e dois fios de sinal.

- Os conectores e o sexo dos conectores são padronizados.
- Os conectores são pequenos e os cabos são finos e flexíveis.
- Os dispositivos podem ser conectados e desconectados sem a necessidade de desligar o sistema.
- Velocidades de até 480 Mb/s são possíveis (USB 2.0).
- Dispositivos paralelos e seriais antigos podem ser conectados via adaptadores.

O USB pode até mesmo ser utilizado com tecnologia LAN, embora não seja realmente projetado para isso.

O Linux já tem um amplo e sólido suporte a dispositivos USB. O padrão USB define interfaces padrão para várias classes de equipamentos comuns (como mouses, modems e dispositivos de armazenamento de massa), de forma que o emprego destes dispositivos em geral é relativamente direto. Por outro lado, dispositivos como câmeras e scanners podem ocasionalmente requerer drivers de chipset específicos.

Informações sobre dispositivos USB podem ser encontradas em **/proc/bus/usb**. Arquivos numerados (como **001**, que representa a controladora USB do próprio host) correspondem a dispositivos USB individuais. O arquivo **/proc/bus/usb/devices** fornece informações bem detalhadas sobre o censo atual dos dispositivos, agora o arquivo **/proc/bus/usb/drivers** contém os nomes dos drivers USB registrados atualmente (independentemente de eles estarem sendo utilizados ou não). Ambos os arquivos podem ser inspecionados com **less** ou seu editor de textos predileto.

Embora o sistema de arquivos **/proc/bus/usb** suporte autoconfiguração e ajude na depuração, em geral ele não é usado diretamente pelos drivers no nível do kernel. A maioria dos dispositivos USB é acessada por arquivos de dispositivos no estilo UNIX tradicional sob **/dev**.

O USB é mais comumente utilizado com dispositivos de armazenamento de dados como unidades de memória flash. Consulte a página 104 para mais detalhes sobre como montar unidades USB. Para informações atualizadas sobre outros tipos de dispositivos, consulte a lista de dispositivos em www.linux-usb.org.

27.17 EXERCÍCIOS

E27.1 O que é um cabo modem nulo? Como ele é utilizado para conectar dispositivos seriais DCE e DTE?

E27.2 Podemos utilizar um cabo serial de 3 fios para uma conexão de modem serial? Para uma impressora serial? Por que sim ou por que não?

E27.3 Como o tradicional controle de fluxo de hardware serial funciona? O que pode ser feito caso um sistema não entenda o controle de fluxo de hardware?

E27.4 O que é um pseudoterminal? Quais programas utilizam pseudoterminais?

E27.5 Elabore entradas **inittab** que:

a) Executem um programa chamado **server-fallback**, espere sua finalização e depois pare imediatamente o sistema se a energia falhar.

b) Gere novamente um servidor chamado **unstable-srv** caso ele trave.

c) Execute um script chamado **clean-temp** que elimine todos os arquivos temporários cada vez que o sistema é inicializado.

E27.6 Você conectou um novo dispositivo USB ao seu computador. Como você pode verificar se ele foi reconhecido e está acessível?

★ E27.7 Compare os padrões serial RS-232 e USB.

★ E27.8 Um amigo descuidadamente foi embora e deixou seu terminal conectado pela madrugada toda no laboratório Linux e agora o laboratório está passando por problemas estranhos ao executar aplicativos shell. Os programas se fecharam ou foram suspensos, e entradas anteriores desaparecem quando certos comandos e entradas são feitos; mas algumas coisas parecem funcionar normalmente. O que um usuário malicioso poderia ter feito para provocar tal comportamento? Explique como você poderia testar sua resposta. Como o problema poderia ser corrigido? Quem poderia ter feito essa travessura?

28 *Drivers e o kernel*

O kernel é responsável por ocultar o hardware do sistema debaixo de uma interface de programação abstrata, de alto nível. Ele fornece muitos dos recursos que os usuários e os programas no nível de usuário tomam como certos. Por exemplo, o kernel cria todos os conceitos a seguir a partir dos recursos de hardware de nível mais baixo:

- Processos (compartilhamento de tempo, espaços de endereços protegidos)
- Sinais e semáforos
- Memória virtual (swapping, paginação, mapeamento)
- O sistema de arquivos (arquivos, diretórios e espaço de nomes)
- Entrada/saída geral (hardware, teclado, mouse especiais)
- Comunicação entre processos (pipes e conexões de rede)

O kernel contém drivers de dispositivo que gerenciam sua interação com peças de hardware específicas; o restante do kernel é, em grande parte, independente de dispositivo. O relacionamento entre o kernel e seus drivers de dispositivo é semelhante ao relacionamento entre processos em nível de usuário e o kernel. Quando um processo pede para o kernel: "Leia os primeiros 64 bytes de **/etc/passwd**", o kernel poderia traduzir essa solicitação em uma instrução de driver de dispositivo como "Recupere o bloco 3.348 do dispositivo 3". O driver subdividiria ainda mais esse comando em seqüências de padrões de bits a ser apresentadas aos registradores de controle do dispositivo.

O kernel é, em grande parte, escrito em C, com uma pitada de linguagem assembly para ajudá-lo a interfacear com funções específicas de hardware ou de chips que não são acessíveis por meio das diretivas de compilador normais.

Uma das vantagens do ambiente Linux é que a disponibilidade de código-fonte torna relativamente fácil o desenvolvimento dos seus próprios drivers de dispositivos e módulos de kernel da estaca zero. Nos primórdios do Linux, ter habilidades nesse campo era uma necessidade, pois era difícil administrar sistemas Linux efetivamente sem ser capaz de "moldar" o sistema para um ambiente específico.

Hoje, os administradores de sistemas podem ser perfeitamente eficientes sem nunca precisarem lidar com o maçante código de kernel. Na realidade, pode-se argumentar que tais atividades são mais bem tratadas pelas mãos de programadores e que os administradores deveriam se concentrar mais nas necessidades como um todo da comunidade de usuários. Os administradores de sistemas podem ajustar o kernel ou adicionar módulos preexistentes conforme descrito neste capítulo, porém eles não precisam fazer um curso aprofundado de programação em C ou linguagem assembly para sobreviverem.

O objetivo final é que você não confunda a administração de ambientes Linux modernos com a "economia doméstica" pouco explorada de alguns anos atrás.

28.1 ADAPTAÇÃO DO KERNEL

Os sistemas Linux vivem em um mundo que poderia incluir potencialmente qualquer uma das dezenas de milhares de dispositivos de hardware de computadores disponíveis hoje. O kernel tem de se adaptar a seja lá qual for o hardware presente na máquina na qual estiver sendo executado.

Um kernel pode aprender sobre o hardware do sistema de uma série de maneiras. As mais básicas são você informar explicitamente ao kernel sobre o hardware que ele deve esperar encontrar (ou espera não encontrar, como pode ser o caso). Além disso, o kernel procura alguns dispositivos por conta própria, em tempo de inicialização ou dinamicamente (depois que o sistema entra em execução). O último método é o mais comum para dispositivos modernos como aqueles que residem no barramento serial universal (Universal Serial Bus – USB), incluindo *memory sticks*, modems, câmeras digitais e impressoras. Felizmente, o Linux tem suporte razoável a uma ampla gama desses dispositivos.

Na plataforma PC, em que o Linux é popular, o desafio de criar um inventário acurado do hardware do sistema é particularmente difícil (e, algumas vezes, impossível). O hardware de PC seguiu uma linha evolutiva não muito diferente da nossa na qual protozoários primitivos agora deram vida a tudo, de cães selvagens a abelhas assassinas. Essa diversidade é

composta pelo fato de que fabricantes de PCs costumam não nos oferecer muita informação sobre os sistemas que eles vendem, de modo que muitas vezes você tem de "desmontar" seu sistema e inspecionar visualmente as peças para responder a perguntas como: "Que chipset Ethernet minha placa de rede usa?".

Sistemas Linux modernos sobrevivem sob uma dieta híbrida de componentes de kernel estáticos e dinâmicos, sendo a mistura entre os dois ditada basicamente pelas limitações de hardware do PC. É provável que em algum ponto durante a sua carreira de administrador de sistemas você precisará dar uma mão na forma de criar uma configuração de kernel.

28.2 DRIVERS E ARQUIVOS DE DISPOSITIVO

Um driver de dispositivo é um programa que gerencia a interação do sistema com um dispositivo de hardware. O driver faz a tradução entre os comandos de hardware compreendidos pelo dispositivo e a interface de programação estilizada utilizada pelo kernel. A camada de driver ajuda a manter o Linux razoavelmente independente de dispositivo.

Drivers de dispositivos fazem parte do kernel; eles não são processos de usuários. Entretanto, um driver pode ser acessado tanto de dentro do kernel como do espaço de usuário. O acesso em nível de usuário para dispositivos é normalmente por meio de arquivos de dispositivos especiais que residem no diretório **/dev**. O kernel transforma operações nesses arquivos em chamadas para o código do driver.

A plataforma PC tem sido uma fonte de caos no mundo do administrador de sistemas. Um conjunto desconcertante de hardwares e "padrões" com níveis variáveis de suporte por parte de sistemas operacionais se encontram à disposição. Veja:

- Atualmente, há mais de 30 chipsets SCSI diferentes disponíveis e cada um é empacotado e vendido por pelo menos o dobro do preço de muitos fornecedores
- Mais de 200 interfaces de rede diferentes se encontram lá, cada uma delas sendo comercializada por vários fornecedores diferentes sob nomes diversos
- Tipos de hardware mais recentes, melhores e menos caros estão sendo desenvolvidos todo o tempo. Cada um requer um driver para que ele possa funcionar com seu Linux preferido

Com esse ritmo alucinante no qual novos dispositivos de hardware estão sendo desenvolvidos, é praticamente impossível manter atualizadas as principais distribuições de sistemas operacionais em relação ao hardware mais recente. Ocasionalmente, você precisará adicionar um driver de dispositivo ao seu kernel para suportar um novo hardware.[1]

Apenas drivers de dispositivos para uso com o Linux (e normalmente, uma versão específica do kernel do Linux) podem ser instalados com sucesso em um sistema Linux. Drivers para outros sistemas operacionais (por exemplo, Windows) não funcionarão, portanto, ao comprar novo hardware, tenha isso em mente. Além disso, os dispositivos variam em termos de nível de compatibilidade e de funcionalidade quando utilizados com o Linux; ou seja, é sensato prestar atenção aos resultados obtidos por outras instalações com qualquer hardware que você esteja considerando comprar.

Os fornecedores estão se tornando mais cientes dos mercados UNIX e Linux e eles freqüentemente fornecem drivers Linux com seus produtos. Na melhor situação, seu fornecedor disponibilizará tanto instruções sobre os drivers como sobre a instalação. Ocasionalmente, você só encontrará o driver necessário em alguma página Web não comentada. Em qualquer caso, essa seção mostra o que está realmente acontecendo quando adicionamos um driver de dispositivo.

Arquivos de dispositivo e números de dispositivo

Muitos dispositivos têm um arquivo correspondente em **/dev**; exceções notáveis nos sistemas operacionais modernos são os dispositivos de rede.

Pela vantagem de serem arquivos de dispositivos, os arquivos em **/dev** possuem um número de dispositivo principal e um secundário associado a eles. O kernel utiliza esses números a fim de mapear referências aos arquivos de dispositivo para o driver correspondente.

O número de dispositivo principal identifica o driver com o qual o arquivo está associado (em outras palavras, o tipo de dispositivo). O número de dispositivo secundário identifica qual instância em particular de um dado tipo de dispositivo deve ser tratada. O número de dispositivo secundário é algumas vezes chamado de número de unidade.

Podemos ver os números de dispositivo principal e secundário de um arquivo de dispositivo pelo comando **ls -l**:

```
$ ls -l /dev/sda
brw-rw---- 1 root    disk     8,   0 Jan 5 2005 /dev/sda
```

Esse exemplo ilustra o primeiro disco SCSI em um sistema Linux. Ele tem um número principal 8 e um secundário igual a 0.

---

1 No hardware de PC, você pode utilizar o comando **lspci** para visualizar os dispositivos atualmente conectados ao barramento PCI e reconhecidos pelo kernel.

O número de dispositivo secundário é algumas vezes utilizado pelo driver para selecionar a característica particular de um dispositivo. Por exemplo, uma única unidade de fita pode ter vários arquivos em **/dev** representando-a em várias configurações de densidade de gravação e características de rebobinamento. O driver é livre para interpretar o número de dispositivo secundário da maneira que preferir. Consulte a página man do driver para determinar que convenção ele está adotando.

Na verdade, há dois tipos de arquivos de dispositivo: arquivos de dispositivo de blocos e arquivos de dispositivo de caracteres. Um dispositivo de blocos é lido ou gravado um bloco por vez (um grupo de bytes, normalmente um múltiplo de 512); um dispositivo de caracteres pode ser lido ou gravado um byte por vez. Discos e fitas têm uma vida dupla; terminais e impressoras não.

Algumas vezes é conveniente implementar uma abstração como um driver de dispositivo mesmo quando ele não controla efetivamente nenhum dispositivo. Tais dispositivos fantasmas são conhecidos como pseudodispositivos. Por exemplo, um usuário que faz o login em uma rede recebe um PTY (pseudo-TTY) que se parece e se comporta como uma porta serial sob a perspectiva do software de alto nível. Esse truque permite que programas escritos na época em que todo o mundo usava uma TTY continuem a funcionar no mundo atual das janelas e das redes.

Quando um programa efetua uma operação sobre um dispositivo de arquivo, o kernel captura automaticamente a referência, procura o nome de função apropriado em uma tabela e transfere o controle para ela. Para executar uma operação incomum que não tenha um análogo direto no modelo de sistema de arquivos (por exemplo, ejetar um disquete), um programa pode utilizar a chamada de sistema **ioctl** para passar uma mensagem diretamente do espaço de usuário para o driver.

Criando arquivos de dispositivo

Os arquivos de dispositivo podem ser criados manualmente com o comando **mknod**, com a sintaxe

```
mknod nomeDoArquivo tipo principal secundário
```

onde *nomeDoArquivo* é o arquivo de dispositivo a ser criado, *tipo* é **c** para um dispositivo de caracteres ou **b** para um dispositivo de blocos e *principal* e *secundário* são os números de dispositivo principal e secundário. Se você estiver criando um arquivo de dispositivo manualmente que faz referência a um driver já presente no seu kernel, consulte a página man desse driver a fim de localizar os números dos dispositivos primários e secundários apropriados.

Historicamente, os arquivos de dispositivo eram criados manualmente em **/dev** pelo administrador do sistema. A maioria dos sistemas fornecia um script, chamado **MAKEDEV**, no diretório **/dev** para ajudar nessa tarefa. **MAKEDEV** às vezes, mas nem sempre, sabia criar os arquivos de dispositivo corretos para um componente particular. Era um processo entediante no melhor dos casos.

Na versão 2.6 do kernel do Linux,[2] o sistema `udev` gerencia dinamicamente a criação e a remoção dos arquivos de dispositivo de acordo com a presença real (ou ausência) dos dispositivos. O daemon **udevd** ouve as mensagens do kernel relacionadas às alterações no status do dispositivo. Com base nas informações sobre a configuração nos subdiretórios **/etc/udev/udev.conf**, o **udevd** pode tomar várias ações se um dispositivo é descoberto ou está desconectado. Por padrão, **udevd** cria os arquivos de dispositivo em **/dev**. Ele também tenta executar scripts de configuração de rede quando novas interfaces de rede são detectadas.

`sysfs`: uma janela para o reino dos dispositivos

Outro recurso introduzido na versão 2.6 do kernel é o `sysfs`. Isso é um sistema de arquivos virtual que fornece informações bastante detalhadas e bem organizadas sobre os dispositivos livres, suas configurações e seus estados. Essas informações estão acessíveis tanto de dentro do kernel como do espaço de usuário.

Você pode explorar o diretório **/sys**, onde `sysfs` é em geral montado, para descobrir tudo, do IRQ que um dispositivo está utilizando ao número de blocos enfileirados a ser gravado em uma controladora de disco. Um dos princípios orientadores do `sysfs` é que cada arquivo em **/sys** deve representar somente um atributo do dispositivo subjacente. Essa convenção impõe um pouco de estrutura a um conjunto de dados que seria caótico em outros aspectos.

Originalmente, as informações sobre a configuração do dispositivo seriam encontradas no sistema de arquivos **/proc** (`procfs`, discutido mais adiante neste capítulo) se estivessem absolutamente disponíveis. Embora o **/proc** continue a manter informações de tempo de execução sobre processos e o kernel, sabemos que as informações específicas ao dispositivo serão movidas para **/sys** ao longo do tempo.

---

2 O `udev` é uma substituição completa para `devfs`, uma outra tentativa recente de alcançar funcionalidades semelhantes. O `udev` não implementa todos os recursos do `devfs`, mas acredita-se que, arquitetonicamente, ele seja mais puro. Você não sabia que todos esses convenientes recursos extras do devfs eram simplesmente vícios auto-indulgentes?

Como o `sysfs` é relativamente novo, boa parte do seu potencial permanece hoje não-utilizado. Conseqüentemente, talvez seja possível configurar um dispositivo em tempo real por meio do `sysfs`. Em longo prazo, ele até mesmo poderia substituir todo ou parte do **/dev**. Só o tempo dirá.

Convenções de nomeação para dispositivos

As convenções para atribuição de nomes para dispositivos são um tanto aleatórias. Normalmente, elas são remanescentes da maneira como era feita no UNIX em um DEC PDP-11, por mais arcaico que isso possa parecer nos dias de hoje.

Consulte o Capítulo 27 para informações adicionais sobre portas seriais.

Os arquivos de dispositivos seriais são chamados **ttyS** seguidos por um número que identifica a interface específica à qual a porta está conectada. TTYs são algumas vezes representados por mais de um arquivo de dispositivo; os arquivos extras normalmente fornecem acesso a métodos de controle de fluxo alternativos ou a protocolos de bloqueio.

Os nomes dos dispositivos de fita costumam incluir não apenas uma referência à própria unidade como também uma indicação de se o dispositivo está rebobinado depois de o dispositivo de fita ser fechado.

Os dispositivos de disco rígido IDE recebem nomes **/dev/hd**LP, onde L é uma letra que identifica a unidade (com **a** sendo o mestre na primeira interface IDE, **b** sendo o escravo nesta mesma interface, **c** sendo o mestre na segunda interface IDE e assim por diante) e P é o número de partição (começando com 1). Por exemplo, a primeira partição do primeiro disco IDE é tipicamente **/dev/hda1**. Os discos SCSI recebem nomes de forma similar, mas com o prefixo **/dev/sd** em vez de **/dev/hd**. Você pode eliminar o número de partição em ambos os tipos de dispositivos para poder acessar o disco inteiro (por exemplo, **/dev/hda**).

As unidades de CD-ROM SCSI são designadas pelos arquivos **/dev/scd**N, onde N é um número que distingue várias unidades de CD-ROM. As unidades de CD-ROM modernas IDE (ATAPI) são designadas da mesma maneira que os discos rígidos IDE (por exemplo, **/dev/hdc**).

28.3 POR QUE E COMO CONFIGURAR O KERNEL?

Quando o sistema é instalado, ele vem com uma configuração genérica que é projetada para executar a maioria dos aplicativos em quase todos os hardwares. A configuração genérica inclui muitos drivers de dispositivos diferentes e pacotes opcionais e ela possui valores de parâmetros ajustáveis escolhidos para "emprego genérico". Examinando-se atentamente essa configuração e ajustando-a para suas necessidades exatas, você pode ser capaz de otimizar o desempenho, a segurança e até mesmo a confiabilidade de seu sistema.

Os kernels do Linux modernos são melhores do que seus antecessores ao descarregarem drivers indesejados da memória, porém opções compiladas não como módulos sempre estarão ativas. Embora reconfigurar o kernel para fins de eficiência seja menos importante do que costumava ser, um bom caso ainda pode justificar a configuração.

Instruções para adicionar um novo driver iniciam na página 602.

Outra razão para reconfigurar o kernel é a de adicionar suporte a novos tipos de dispositivos (por exemplo, adicionar novos drivers de dispositivos). O código do driver não pode simplesmente ser enfiado no kernel como um monte de massinha de modelar; ele tem de ser integrado às tabelas e estruturas de dados do kernel. Em alguns sistemas, esse procedimento pode exigir que você retorne aos arquivos de configuração para o kernel e adicione o novo dispositivo, reconstruindo o kernel da estaca zero. Em outros sistemas, talvez você só precise executar um programa desenvolvido para fazer automaticamente essas mudanças de configuração.

A configuração do kernel não é difícil; só é difícil corrigir uma vez que danificamos o kernel.

Você pode utilizar qualquer um dos quatro métodos básicos para configurar um kernel do Linux. Provavelmente, você acabará tendo a oportunidade de tentar todos eles. Os métodos são:

- Modificar parâmetros de configuração de kernel ajustáveis (dinâmicos)
- Criar um kernel da estaca zero (isso, de fato, significa compilá-lo a partir de código-fonte, possivelmente com modificações e acréscimos)
- Carregar novos drivers e módulos em um kernel existente durante a execução
- Fornecer diretivas operacionais em tempo de inicialização por meio da rotina de inicialização do kernel, LILO ou GRUB; consulte a página 19 para mais informações sobre esses sistemas

Cada um desses métodos é aplicável em situações ligeiramente diferentes. Modificar parâmetros ajustáveis é a maneira mais fácil e mais comum, ao passo que criar um kernel de arquivos de código-fonte é o jeito mais difícil e, em geral, menos exigido. Felizmente, todas essas abordagens acabam sendo incorporadas com um pouco de prática.

28.4 AJUSTANDO OS PARÂMETROS DO KERNEL DO LINUX

Muitos módulos e drivers no kernel foram desenvolvidos com o conhecimento de que uma condição não atende a tudo. Para aumentar a flexibilidade, ganchos especiais permitem que parâmetros como o tamanho de uma tabela interna ou o comportamento do kernel em uma determinada circunstância sejam ajustados pelo administrador de sistemas com o sistema em execução. Esses ganchos são acessíveis por meio de uma extensa interface kernel-userland representada pelos arquivos no sistema de arquivos **/proc** (conhecido como `procfs`). Em muitos casos, um grande aplicativo em nível de usuário (especialmente um aplicativo de infra-estrutura como um banco de dados) exigirá o ajuste de parâmetros para atender às suas necessidades específicas.

Arquivos especiais em **/proc/sys** permitem visualizar e configurar as opções do kernel em tempo de execução. Esses arquivos simulam arquivos Linux padrão, porém eles são realmente as portas dos fundos para o kernel. Se um desses arquivos tiver um valor que você pretende alterar, você poderá tentar gravar nele. Infelizmente, não é possível gravar em todos os arquivos (independentemente das suas aparentes permissões) e não há muita documentação disponível. Se tiver a árvore do fonte do kernel disponível, você conseguirá ler alguns valores e seus significados no subdiretório **Documentation/syscnt**.

Por exemplo, para alterar o número máximo de arquivos que o sistema pode abrir de uma vez, experimente algo como

```
# echo 32768 > /proc/sys/fs/file-max
```

Assim que se acostumar com essa interface não ortodoxa, você achará bem útil, especialmente para modificar opções de configuração. Uma palavra de alerta, porém: mudanças não são lembradas após reinicializações. A Tabela 28.1 lista alguns parâmetros comumente ajustados.

Tabela 28.1 Arquivos em /proc/sys para alguns parâmetros de kernel ajustáveis

Dir[a]	Arquivo	Padrão	Função e comentário
C	**autoeject**	0	Ejeção automática de CD-ROM na desmontagem? Bom para dar um susto nos operadores da sala de máquinas às 3h00 da manhã.
F	**file-max**	4096	Configura o número máximo de arquivos abertos. Em um sistema que trata um grande número de arquivos, tente aumentar isso para 16384.
F	**inode-max**	16384	Configura o número máximo de inodes aberto por processo. Poderia ser útil experimentar isso se você estiver escrevendo um aplicativo que abre dezenas de milhares de handles de arquivos.
K	**ctrl-alt-del**	0	Reinicialização na seqüência Ctrl-Alt-Del? Isso pode ser uma questão de preferência pessoal ou pode aumentar a segurança nos consoles de servidor que não são protegidos fisicamente.
K	**printk_ratelimit**	5	Segundos mínimos entre as mensagens de kernel
K	**printk_ratelimit _burst**	10	Número de mensagens sucessivas antes de o limite de taxa **printk** ser realmente imposto.
K	**shmmax**	32M	Configura a quantidade máxima de memória compartilhada. Sintonize se você tiver aplicativos que exigem memória compartilhada significativa.
N	**conf/default/rp_filter**	0	Permite a verificação da rota de origem. Esse mecanismo de anti-spoofing faz com que o kernel descarte pacotes recebidos de caminhos "impossíveis".
N	**icmp_echo_ignore_all**	0	Ignora o ping de ICMP quando configurado como 1. Bom se você quiser que o sistema não responda ao comando ping.
N	**icmp_echo_ignore_ broadcasts**	0	Ignora pings de broadcast quando configurados como 1. Quase sempre uma boa idéia é configurar isso como 1.
N	**icmp_ignore_bogus_ error_responses**	0	Ignora erros ICMP incorretamente formatados. Quase sempre uma boa idéia é configurar isso como 1.
N	**ip_forward**	0	Permite encaminhamento de IP quando configurado como 1. Aprimora a segurança quando configurado como 0; configure-o como 1 se você estiver utilizando a máquina Linux como um roteador.
N	**ip_local_port_range**	32768 61000	Especifica o intervalo de porta local alocado durante a configuração da conexão. Para servidores que iniciam muitas conexões externas, expanda isso para 1024–65000 para desempenho aprimorado.
N	**tcp_fin_timeout**	60	Especifica a espera em segundos para um pacote FIN final. Configurado como um valor mais baixo (~20) nos servidores de alto tráfego para aumentar o desempenho.
N	**tcp_syncookies**	0	Protege contra ataques de inundação SYN. Ative-o se suspeitar de ataques negação de serviço (DOS).

a. F = **/proc/sys/fs**, N = **/proc/sys/net/ipv4**, K = **/proc/sys/kernel**, C = **/proc/sys/dev/cdrom**

Uma maneira mais permanente de modificar esses mesmos parâmetros pode ser encontrada na maioria dos sistemas na forma do comando **sysctl**. **sysctl** pode configurar variáveis individuais seja pela linha de comando seja lendo uma lista de pares *variável=valor* de um arquivo. Como padrão, o arquivo **/etc/sysctl.conf** é lido em tempo de inicialização e seu conteúdo utilizado para configurar os valores de parâmetros iniciais (personalizados).

Por exemplo, o comando:

```
# sysctl net.ipv4.ip_forward=0
```

desativa o encaminhamento de IPs. Observe que você forma os nomes de variáveis utilizados por **sysctl** substituindo as barras na estrutura do diretório **/proc/sys** por pontos.

28.5 Construindo um kernel do Linux

Como o Linux está evoluindo tão rapidamente, é possível que você por fim tenha de enfrentar a necessidade de construir um kernel do Linux. Patches de kernel, drivers de dispositivos e novas funcionalidades entram continuamente em cena. Isso realmente é uma faca de dois gumes. De um lado, é conveniente sempre suportar o "mais recente e o melhor", por outro, pode levar muito tempo manter-se atualizado com tamanho fluxo de material novo. Mas depois da construção bem-sucedida de um kernel, você se sentirá fortalecido e pronto para fazê-lo de novo.

É menos provável que você precise construir um kernel por conta própria caso esteja usando uma versão "estável". Originalmente, o Linux adotava um esquema de numeração de versões no qual a segunda parte do número da versão indicava se o kernel era estável (números pares) ou ainda estava em desenvolvimento (números ímpares). Por exemplo, o kernel versão 2.6.6 seria um kernel "estável", ao passo que 2.7.4 seria um kernel "em desenvolvimento". Hoje, esse esquema não é religiosamente seguido, portanto é melhor verificar a home page em kernel.org para a palavra oficial sobre essa questão. O site kernel.org também é a melhor fonte para código-fonte do kernel do Linux se você não estiver contando com uma distribuição particular (ou fornecedor) a fim de obter um kernel.

Se ele estiver funcionando, não o corrija

Com as novas versões do kernel do Linux entrando em cena em um intervalo de poucos meses e novos drivers e patches sendo lançados a cada dia, é fácil se tornar viciado em patches e atualizações. Afinal de contas, o que há de mais excitante do que informar à sua comunidade de usuário que você acabou de descobrir um novo patch de kernel e que estará desativando o servidor de e-mails à tarde para instalá-lo? Alguns administradores justificam suas existências dessa maneira; todo mundo gosta de bancar o herói.

Um bom administrador de sistemas pesa cuidadosamente necessidades e riscos ao planejar atualizações e patches a serem aplicados no kernel. Certamente, a nova versão pode ser a mais recente e a mais sensacional, porém é tão estável quanto a versão atual? Essa atualização ou patch poderia ser adiada e instalada com um outro grupo de patches no final do mês? É importante resistir à tentação de ser um hacking de kernels prevalecer sobre os interesses de sua comunidade de usuários.

Uma boa regra prática é atualizar ou aplicar patches apenas quando os ganhos de produtividade que você espera obter (normalmente medidos em termos de confiabilidade e de desempenho) forem exceder o esforço e o tempo gastos para realizar a instalação. Caso esteja com problemas em quantificar o ganho específico, isto é um bom sinal de que o patch pode esperar mais alguns dias.

Configurando opções do kernel

A configuração do kernel do Linux trilhou um longo caminho, mas ainda parece primitiva comparada com o procedimento utilizado em alguns outros sistemas. Historicamente, o fonte do kernel do Linux era armazenado em **/usr/src/linux** e os privilégios root eram exigidos para construir um kernel. As versões 2.4, 2.6 e superiores do kernel podem ser construídas como um usuário não-privilegiado e conseqüentemente agora é mais politicamente correto armazenar o código-fonte do kernel em um diretório específico à instalação como **/usr/local/src/kernel** ou *~nomeDoUsuário/***kernel**.

Neste capítulo utilizamos *caminho_para_o_fonte_do_kernel* como um marcador de lugar para o diretório em que você escolheu colocar o código-fonte do kernel. Em todos os casos, você precisa instalar o pacote de kernel original antes de poder construir um kernel em um sistema; consulte a página 164 para dicas sobre a instalação de pacotes.

O processo de configuração do kernel ocorre no arquivo **.config** da raiz do diretório do fonte do kernel. Todas as informações sobre a configuração do kernel são especificadas nesse arquivo, mas seu formato é um pouco obscuro. Utilize o guia de decodificação em

*caminho_para_o_fonte_do_kernel/***Documentation/Configure.help**

para descobrir o significado das várias opções.

Para poupar os amigos de ter de editar o arquivo **.config** diretamente, o Linux possui várias diretivas do **make** que permitem configurar o arquivo kernel com diferentes interfaces. Se estiver usando o KDE, a melhor interface de configuração

será fornecida por **make xconfig**. Da mesma forma, se você estiver executando GNOME, **make gconfig** talvez seja a melhor opção. Esses comandos acionam uma tela de configuração gráfica na qual podemos selecionar os dispositivos a ser adicionados ao kernel (ou compilar como módulos carregáveis).

Se não estiver executando o KDE ou o GNOME, você poderá utilizar uma alternativa baseada em **curse**[3] chamada por meio de **make menuconfig**. Finalmente, o estilo **make config** mais antigo lhe apresenta um prompt para responder a cada opção de configuração disponível sem permitir retornar e mudar de idéia. Recomendamos **make xconfig** ou **make gconfig** se seu ambiente suportá-los; do contrário, utilize **make menuconfig**. Evite **make config**.

Se você estiver migrando uma configuração de kernel existente para uma nova versão do kernel (ou árvore), utilize o comando **make oldconfig** para ler o arquivo de configuração anterior e fazer apenas as perguntas que são novas.

Essas ferramentas são bem simples e objetivas no que diz respeito às opções que podem ser ativadas, porém, infelizmente, elas são difíceis de usar se você quiser manter várias versões do kernel para várias arquiteturas ou configurações de hardware.

As diversas interfaces de configuração descritas anteriormente geram um arquivo **.config** que se parece com o seguinte:

```
# Automatically generated make config: don't edit
# Code maturity level options

CONFIG_EXPERIMENTAL=y
#
# Processor type and features
#
# CONFIG_M386 is not set
# CONFIG_M486 is not set
# CONFIG_M586 is not set
# CONFIG_M586TSC is not set
CONFIG_M686=y
CONFIG_X86_WP_WORKS_OK=y
CONFIG_X86_INVLPG=y
CONFIG_X86_BSWAP=y
CONFIG_X86_POPAD_OK=y
CONFIG_X86_TSC=y
CONFIG_X86_GOOD_APIC=y
...
```

Como você pode ver, o conteúdo é um tanto enigmático e não fornece nenhuma descrição sobre o significado das tags `CONFIG`. Algumas vezes é possível descobrir o significado. Basicamente, cada linha `CONFIG` se refere a uma opção de configuração de kernel específica. O valor **y** compila a opção no kernel; **m** o ativa, mas como um módulo carregável.

Algumas coisas podem ser configuradas como módulos e outras não. Você só precisa saber quem é quem; não fica claro pelo arquivo **.config**. Os tags `CONFIG` também não são facilmente mapeados para informações significativas.

Construindo o kernel binário

Configurar um arquivo **.config** apropriado é a parte mais importante do processo de configuração do kernel do Linux, mas você deve passar por muitas outras etapas para transformar esse arquivo em um kernel finalizado.

Eis uma descrição do processo todo:

- Utilizar **cd** a fim de ir para ao nível superior do diretório do fonte do kernel
- Executar **make xconfig**, **make gconfig** ou **make menuconfig**
- Executar **make dep** (não exigido para as versões 2.6.x e superiores do kernel)
- Executar **make clean**
- Executar **make**
- Executar **make modules_install**
- Copiar **arch/i386/boot/bzImage** para **/boot/vmlinuz**
- Copiar **arch/i386/boot/System.map** para **/boot/System.map**
- Editar **/etc/lilo.conf** (LILO) ou **/boot/grub/grub.conf** (GRUB) a fim de adicionar uma linha de configuração para o novo kernel
- Se estiver usando LILO, execute **/sbin/lilo** para instalar a rotina de inicialização reconfigurada

---

3. **curse** é uma biblioteca de antigamente utilizada para criar GUIs baseadas em texto que executam em uma janela terminal.

A etapa **make clean** nem sempre é estritamente necessária, porém em geral é uma boa idéia começar com um ambiente de construção limpo. Na prática, muitos problemas podem ser atribuídos a não realização desse passo.

28.6 ADICIONANDO UM DRIVER DE DISPOSITIVO LINUX

Em sistemas Linux, os drivers de dispositivo são distribuídos tipicamente em uma das três formas:

- Uma correção (patch) em oposição a uma versão de kernel específica
- Um módulo carregável
- Um script de instalação ou pacote que instala os patches apropriados

A forma mais comum é o script ou pacote de instalação. Se tiver sorte suficiente em ter um destes para seu novo dispositivo, você conseguirá seguir as instruções para executar o script ou instalar o pacote, e seu novo dispositivo será reconhecido pelo kernel.

Nas situações em que você tem um patch contra uma versão do kernel específica você pode, na maioria das vezes, instalar o patch com o procedimento a seguir:[4]

```
# cd caminho_para_o_fonte_do_kernel ; patch -p1 < arquivo_do_patch
```

Diferenças em relação a uma versão secundária diferente do kernel podem falhar, mas o driver ainda deve funcionar.

Se não funcionar, é provável que você esteja em uma situação em que será necessário integrar manualmente o novo driver de dispositivo à árvore do fonte do kernel. Se tiver sorte, isso ocorre porque você está escrevendo seu próprio driver de dispositivo, e não porque você foi vítima de um fornecedor de dispositivos comerciais que não conseguiu fornecer os scripts de instalação apropriados. Nas páginas a seguir, demonstramos como adicionar manualmente um driver "snarf" de rede hipotético ao kernel. Na verdade, o Linux torna esse um processo bem entediante, especialmente se comparado com algumas outras versões do UNIX.

Dentro do subdiretório **drivers** da árvore de código-fonte do kernel, você pode localizar o subdiretório que corresponde ao tipo de dispositivo que você tem. Uma listagem de diretórios de drivers se parece com o seguinte:

```
$ ls -F caminho_para_o_fonte_do_kernel/drivers
acorn/     char/         i2c/          Makefile      net/          s390/         telephony/
acpi/      dio/          ide/          md/           nubus/        sbus/         usb/
atm/       fc4/          ieee1394/     media/        parport/      scsi/         video/
block/     gsc           input/        message/      pci/          sgi/          zorro/
bluetooth/ hil/          isdn/         misc/         pcmcia/       sound/
cdrom/     hotplug/      macintosh/    mtd/          pnp/          tc/
```

Os diretórios mais comuns aos quais os drivers são adicionados são **block**, **char**, **net**, **scsi**, **sound** e **usb**. Esses diretórios contêm drivers para dispositivos de blocos (como as unidades de disco IDE), dispositivos de caracteres (como as portas seriais), dispositivos de rede, placas SCSI, placas de som, dispositivos USB, respectivamente. Alguns dos outros diretórios contêm drivers para os barramentos em si (por exemplo, **pci**, **nubus** e **zorro**); é pouco provável que você precisará adicionar drivers a esses diretórios. Alguns diretórios contêm drivers de plataforma específica como **macintosh**, **s390** e **acorn**. Outros diretórios contêm dispositivos especiais como **atm**, **isdn** e **telephony**.

Como nosso exemplo de dispositivo é um relacionado à rede, adicionaremos o driver ao diretório **drivers/net**. Modificaremos estes arquivos:

- **drivers/net/Makefile**, para que nosso driver seja compilado
- **drivers/net/Kconfig**, para que nosso dispositivo apareça nas opções de configuração

Após colocarmos os arquivos **.c** e **.h** para o driver em **drivers/net/snarf**, adicionaremos o driver a **drivers/net/Makefile** A linha que adicionamos (quase no final do arquivo) é

```
obj-$(CONFIG_SNARF_DEV) += snarf/
```

Essa configuração adiciona o driver snarf (armazenado no diretório **snarf/**) ao processo de construção.

Após adicionarmos o dispositivo ao **Makefile**, temos de ter certeza de que podemos configurar o dispositivo ao configurarmos o kernel. Todos os dispositivos de rede devem ser listados no arquivo **drivers/net/Kconfig**. Para adicionar o dispositivo para que ele possa ser construído como um módulo ou como parte do kernel (consistente com o que afirmamos no **Makefile**), acrescentamos esta linha:

---

4. Obviamente, o pacote de código-fonte do kernel tem de ser instalado antes de modificarmos a árvore do kernel.

```
config SNARF_DEV
tristate 'Snarf device support'
```

O primeiro token depois de `config` é a macro de configuração, que deve corresponder ao token depois do `CONFIG_` no **Makefile**. A palavra-chave `tristate` significa que podemos criar o dispositivo como um módulo. Se o dispositivo não puder ser criado como um módulo, use a palavra-chave `bool` em vez de `tristate`. O próximo *token*, ou identificador, é a string a ser exibida na tela de configuração. Ele pode ser qualquer texto arbitrário, mas deve identificar o dispositivo que está sendo configurado.

Detecção de dispositivos

Tendo conseguido conectar um novo driver de dispositivo ao kernel, como você instrui o kernel de que ele precisa utilizar o novo driver? Nas versões do kernel anteriores a 2.6, essa era uma tarefa entediante que exigia conhecimento de programação. Como parte das recentes alterações arquitetônicas feitas ao modelo de driver de dispositivo, agora há uma maneira padrão para que os próprios drivers se associem ao kernel.

Está além do escopo deste capítulo explicar detalhadamente a maneira como isso acontece, mas o resultado é que os drivers de dispositivo escritos para a versão 2.6 (e superiores) se registrem com a macro `MODULE_DEVICE_TABLE`. Essa macro faz as conexões apropriadas nos bastidores para que outros utilitários como **modprobe** (discutidos a seguir) possam ativar novos dispositivos no kernel.

28.7 MÓDULOS DE KERNEL CARREGÁVEIS

O suporte a módulos de kernel carregáveis (Loadable Kernel Module – LKM) permite que um driver de dispositivo — ou qualquer um outro serviço de kernel — seja linkado e removido do kernel enquanto ele está em execução. Esse recurso torna muito mais fácil a instalação de drivers, uma vez que o binário de kernel não precisa ser modificado. Ele também possibilita que o kernel seja menor, pois os drivers não são carregados a menos que necessário.

Módulos carregáveis são implementados por um ou mais "ganchos" documentados no kernel nos quais drivers de dispositivo adicionais podem se "pendurar". O comando em nível de usuário **insmod** se comunica com o kernel e o informa para carregar novos módulos na memória. O comando **rmmod** descarrega drivers.

Embora os módulos carregáveis sejam convenientes, eles não são cem por cento seguros. Toda vez que você carrega ou descarrega um módulo, você corre o risco de causar pânico no kernel.[5] Não recomendamos carregar ou descarregar um módulo não-testado quando você não quer travar a máquina.

Sob o Linux, quase tudo pode ser construído como um módulo de kernel carregável. As exceções ficam por conta do tipo de sistema de arquivos-raiz, o dispositivo no qual o sistema de arquivos-raiz reside e o driver de mouse PS/2.

Módulos de kernel carregáveis são convencionalmente armazenados sob **/lib/modules/***version*, onde *version* é a versão do seu kernel do Linux domodo retornado por **uname -r**. Você pode inspecionar os módulos atualmente carregados com o comando **lsmod**:

```
# lsmod
Module        Size  Used by
ppp          21452  0
slhc          4236  0 [ppp]
ds            6344  1
i82365       26648  1
pcmcia_core  37024  0 [ds i82365]
```

Estão carregados nessa máquina os módulos da controladora PCMCIA, o driver PPP e os módulos de compactação de cabeçalho PPP.

Como um exemplo de como carregar manualmente um módulo de kernel, eis como inseririamos o módulo snarf que configuramos na seção anterior:

```
# insmod /caminho/para/snarf.ko
```

Também podemos passar parâmetros para os módulos de kernel carregáveis; por exemplo,

```
# insmod /caminho/para/snarf.ko io=0xXXX irq=X
```

5. Esse risco é muito pequeno. Ele *pode* ser tão pequeno como ser considerado insignificante no seu ambiente.

Uma vez que um módulo de kernel carregável foi inserido manualmente no kernel, ele só será removido se solicitarmos de maneira explícita sua eliminação ou se o sistema for reinicializado. Poderíamos usar **rmmod snarf** para remover nosso módulo snarf.

Você pode utilizar **rmmod** a qualquer momento, mas ele só funciona se o número de referências atuais ao módulo (listado na coluna `Used` da saída gerada pelo comando **lsmod**) for 0.

Também podemos carregar LKMs Linux de forma semi-automática com **modprobe**, um envoltório para **insmod** que compreende as dependências, opções e procedimentos de instalação/remoção. **modprobe** utiliza o arquivo **/etc/modules.conf** para descobrir como tratar cada um dos módulos.

Você pode gerar dinamicamente um arquivo **/etc/modprobe.conf** que corresponde a todos os seus módulos atualmente instalados executando **modprobe -c**. Esse comando gera um longo arquivo que se parece com isto:

```
#This file was generated by: modprobe -c
path[pcmcia]=/lib/modules/preferred
path[pcmcia]=/lib/modules/default
path[pcmcia]=/lib/modules/2.6.6
path[misc]=/lib/modules/2.6.6
...
# Aliases
alias block-major-1 rd
alias block-major-2 floppy
...
alias char-major-4 serial
alias char-major-5 serial
alias char-major-6 lp
...
alias dos msdos
alias plip0 plip
alias ppp0 ppp
options ne io=x0340 irq=9
```

As declarações `path` informam onde determinado módulo pode ser encontrado. Você pode modificar ou adicionar entradas desse tipo caso queira manter seus módulos em uma localização não padrão.

A declaração `alias` mapeia entre números principais de dispositivos de blocos, números principais de dispositivos de caracteres, sistemas de arquivos, dispositivos de redes e protocolos de rede e os nomes dos módulos correspondentes.

As linhas `options` não são geradas dinamicamente. Elas especificam opções que deveriam ser passadas para um módulo quando ele é carregado. Poderíamos, por exemplo, usar a linha a seguir para informar ao módulo snarf seu endereço de I/O e vetor de interrupção apropriados:[6]

```
options snarf io=0xXXX irq=X
```

modprobe também entende as instruções `install` e `remove`. Essas instruções permitem que comandos sejam executados quando um módulo específico é inserido ou removido do kernel em execução.

28.8 CONEXÃO AUTOMÁTICA

Embora os módulos de kernel carregáveis atendam à necessidade de o kernel adicionar e remover drivers de dispositivo dinamicamente, os recursos de conexão automática do Linux exportam informações sobre a disponibilidade de dispositivos para o espaço do usuário. Esse recurso permite que os processos do usuário respondam a eventos como a conexão de uma câmera digital ou PDA por meio do USB. Por exemplo, talvez você queira copiar imagens automaticamente para uma unidade local ou sincronizar automaticamente o calendário do seu PDA. A filosofia do projeto de conexão automática é a de que nenhuma entrada adicional de usuário deve ser necessária além da conexão do dispositivo.

A partir da versão 2.6 do kernel, a conexão automática está disponível nos barramentos e drivers que foram projetados para usar `sysfs`. A conexão automática, `sysfs`, e o registro dos drivers de dispositivo antes discutidos estão intimamente linkados.

---

6. Se estiver usando hardware PC excêntrico, pode ser um desafio criar uma configuração na qual os vetores de solicitação de interrupção (IRQs) e portas de E/S não sejam coincidentes. Você pode ver as atribuições atuais em um sistema examinando, respectivamente, o conteúdo de **/proc/interrupts** e **/proc/ioports**. A parte coincidente em geral não é uma questão com o atual hardware de PC mais representativo.

Na implementação atual da conexão automática, o kernel executa o processo de usuário especificado pelo parâmetro **/proc/sys/kernel/hotplug** (normalmente **/sbin/hotplug**) sempre que ele detectar que um dispositivo foi adicionado ou removido. **/sbin/hotplug** é um script de shell que chama um agente específico ao tipo de dispositivo a partir do diretório **/etc/hotplug/** para atuar no evento. Por exemplo, se o evento fosse a adição de uma interface de rede, o script **/etc/hotplug/net.agent** seria executado a fim de colocar a interface on-line. Você pode adicionar ou editar scripts no diretório **/etc/hotplug** para personalizar o comportamento da conexão automática do seu sistema.

Por razões de segurança e outras, não recomendável que o sistema de conexão automática atue sobre um evento. Nessas situações, você pode adicionar dispositivos ao arquivo **/etc/hotplug/blacklist** para impedir que seus eventos desencadeiem ações.

Por outro lado, você pode forçar ações de conexão automática criando arquivos "handmap", como **/etc/hotplug/***type***.handmap**. (Certifique-se de que *type* não entra em conflito com um nome existente se você estiver criando algo novo.)

28.9 CONFIGURANDO AS OPÇÕES DE INICIALIZAÇÃO

Assim que tiver um kernel funcionando e operacional, talvez você precise passar opções de configuração especiais a ele em tempo de inicialização, como o dispositivo-raiz que ele deve usar ou uma instrução para sondar várias placas Ethernet. A rotina de partida (LILO ou GRUB) é responsável pela transferência dessas opções para o kernel.

Para especificar as opções que devem ser utilizadas toda vez que o sistema inicializa, você pode adicionar configurações estáticas a **/etc/ lilo.conf** ou **/boot/grub/grub.conf**, dependendo da rotina de partida que você utiliza. Consulte a página 19 para informações adicionais.

Se não for possível editar o arquivo de configuração da rotina de inicialização (talvez você tenha quebrado algo e a máquina não consegue fazer a inicialização), você pode passar as opções manualmente. Por exemplo, no prompt de inicialização do LILO poderíamos digitar

```
LILO: linux root=/dev/hda1 ether=0,0,eth0 ether=0,0,eth1
```

para informar o LILO para carregar o kernel especificado pelo tag "linux", para usar o dispositivo-raiz **/dev/hda1** e para sondar duas placas Ethernet.

Um exemplo similar usando GRUB seria parecido com o seguinte:

```
grub> kernel /vmlinuz root=/dev/hda1 ether=0,0,eth0 ether=0,0,eth1
grub> boot
```

Uma outra situação comum em que é útil utilizar opções em tempo de inicialização é ao investigar números de unidade lógica (*logical unit numbers* – LUNs) em uma rede de área de armazenamento (*storage area network* – SAN). Por padrão, o kernel do Linux investiga apenas o LUN 0, o que talvez não seja adequado se seu ambiente apresentar áreas lógicas de armazenamento como diferentes LUNs. (Entre em contato com seu administrador ou fornecedor da rede de área de armazenamento para determinar se esse é o caso.) Nessa situação, você precisa informar ao kernel quantos LUNs investigar, visto que a investigação ocorre durante a inicialização.

Por exemplo, se quiser que um kernel 2.4.x investigue os 8 primeiros LUNs, você poderia utilizar uma linha de inicialização como esta:

```
grub> kernel /vmlinuz root=/dev/hda1 max_scsi_luns=8
grub> boot
```

Nas versões 2.6.x e superiores do kernel, o nome do parâmetro mudou:

```
grub> kernel /vmlinuz root=/dev/hda1 max_luns=8
grub> boot
```

28.10 LEITURA RECOMENDADA

BOVET, DANIEL P. E CESATI, MARCO. *Understanding the Linux Kernel. 3. ed.* Sebastopol, CA: O'Reilly Media, 2006.

CORBET, JONATHAN, ET AL. *Linux Device Drivers. 3. ed.* Sebastopol, CA: O'Reilly Media, 2005. Este livro também está disponível on-line em: lwn.net/Kernel/LDD3.

LOVE, ROBERT. Linux Kernel Development. *2. ed.* In: Novell Press, 2005.

28.11 EXERCÍCIOS

E28.1 Descreva o que o kernel faz. Explique a diferença entre carregar um driver como um módulo e vinculá-lo estaticamente ao kernel.

★ E28.2 Examine os valores de vários parâmetros da Tabela 28.1, utilizando tanto o método **/proc** como o método **sysctl**. Modifique dois dos valores com um método e depois os leia novamente com o outro método. Verifique se o comportamento do sistema mudou efetivamente em resposta aos ajustes que você fez. Emita um relatório escrito de seu experimento. (Exige acesso de root.)

★ E28.3 Em uma feira de usados e raridades, você fez um excelente negócio comprando uma placa de laptop que lhe dá conectividade Ethernet por meio da porta paralela. Que etapas você precisaria realizar para fazer com que o Linux reconhecesse essa nova placa? Você deveria compilar suporte diretamente no kernel ou adicioná-lo na forma de um módulo? Por quê? (Pergunta extra: se você cobra R$ 80/hora de consultoria, estime o valor do trabalho necessário para fazer com que essa placa de rede baratinha funcione.)

★ E28.4 Acaba de ser lançada uma nova versão do kernel do Linux e você quer atualizar todas as máquinas no laboratório local (cerca de 50 máquinas, nem todas idênticas). Que questões você deve considerar? Que procedimento você deve seguir? Que problemas poderiam surgir e como você os trataria?

★ ★ E28.5 No laboratório, configure um kernel com **xconfig** ou **menuconfig** e construa um binário de kernel. Instale e execute o novo sistema. Extraia a saída gerada por **dmesg** do kernel antigo e do novo e destaque as diferenças. (Exige acesso de root.)

29 *Daemons*

Um daemon é um processo em segundo plano que realiza uma função ou tarefa de sistema específica. Dentro da filosofia de modularidade do UNIX e do Linux, os daemons são programas em vez de partes do kernel. Muitos iniciam no momento da inicialização e continuam a ser executados enquanto o sistema estiver ligado. Outros são iniciados quando necessário e só executam se forem úteis.

"Daemon" foi utilizado primeiramente como um termo computacional por Mick Bailey, um cidadão britânico que estava trabalhando na equipe de programação do CTSS no MIT durante o início dos anos 60.[1] Mick baseou-se na citação do *Oxford English Dictionary* tanto em termos de significado quanto de pronúncia da palavra. As palavras "daemon" e "demon" provêm da mesma raiz, mas "daemon" é uma forma mais antiga e seu significado é um tanto diverso. Um daemon é um espírito acompanhante que influencia o caráter ou a personalidade de uma pessoa. Os daemons não são subordinados do mal *ou* do bem; eles são criaturas de pensamento e vontade independentes. Os daemons trilharam seu caminho do CTSS para o Multics, depois para o UNIX e finalmente para o Linux, no qual são tão populares que precisam de um superdaemon (**inetd** ou **xinetd**) para gerenciá-los.

Este capítulo apresenta uma visão geral dos daemons Linux mais comuns. Nem todos os daemons listados aqui são fornecidos com todas as versões de distribuições Linux e nem todo daemon fornecido com alguma versão de distribuição Linux é listado aqui. Além de deixá-lo mais ciente sobre o funcionamento do Linux, um conhecimento do que todos os diversos daemons fazem irá torná-lo realmente preparado para quando um de seus usuários perguntar: "O que **klogd** faz?".

Antes de **inetd** ter sido escrito, todos os daemons iniciavam no momento do boot e rodavam continuamente (ou, mais precisamente, eles paravam aguardando novo trabalho). Ao longo do tempo, um número cada vez maior de daemons foi adicionado ao sistema. A população de daemons se tornou tão grande que começou a provocar problemas de desempenho. Em resposta, os gurus da Berkeley desenvolveram o **inetd**, um daemon que inicia outros daemons quando estes são necessários. **inetd** popularizou com sucesso esse modelo de superdaemon, que continua sendo uma maneira comum de minimizar o número de processos que executam em um servidor. A maioria das versões UNIX e Linux agora utiliza uma combinação de **inetd** e daemons em execução permanente.

Há vários daemons com os quais os administradores de sistemas devem estar intimamente familiarizados, seja porque eles requerem uma grande dose de administração seja por desempenharem um grande papel na operação diária do sistema. Alguns daemons descritos aqui em uma ou duas linhas possuem um capítulo inteiro dedicado a eles neste livro. Fornecemos referências cruzadas onde apropriado.

Começamos este capítulo introduzindo uma série de daemons de sistema importantes (**init** e **cron**) e depois avançamos para uma discussão de **inetd** e **xinetd**. Por fim, descrevemos brevemente a maioria dos daemons com os quais provavelmente o administrador de sistemas irá se deparar nas três distribuições adotadas neste livro.

29.1 INIT: O PROCESSO PRIMORDIAL

init é o primeiro processo a ser executado depois de o sistema ser inicializado, e sob diversos aspectos ele é o daemon mais importante. Ele sempre possui um PID igual a 1 e é um ancestral de todos os processos de usuário e de todos, exceto alguns poucos, os processos de sistema.

Na inicialização, **init** coloca o sistema em um modo monousuário ou começa a executar os scripts necessários para trazer o sistema para o modo multiusuário. Ao inicializar o sistema no modo monousuário, **init** executa os scripts de inicialização depois de o shell monousuário ter sido terminado digitando-se **exit** ou <Control-D>.

No modo multiusuário, **init** é responsável pela certificação de que os processos se encontram disponíveis para manipular logins em cada dispositivo ativado por login. Em geral, os logins em portas seriais são tratados por alguma variante de **getty** (por exemplo, **agetty**, **mgetty** ou **mingetty**; consultar a página 589 para detalhes). **init** também supervisiona um procedimento de login gráfico que permite aos usuários conectarem-se diretamente ao X Windows.

---

1. Esse pequeno histórico vem de Jerry Saltzer, no MIT, via Dennis Ritchie.

Além dos seus deveres de gerenciamento de login, **init** também tem a responsabilidade de exorcizar processos zumbis que, de outro modo, se acumulariam no sistema. O papel **init** nesse processo é descrito na página 40.

Consulte a página 588 para informações adicionais sobre o arquivo ***inittab****.*

init define vários "níveis de execução" que determinam que conjunto de recursos de sistema deve ser ativado. Há sete níveis numerados de 0 a 6. O nome "s" é reconhecido como um sinônimo para nível 1 (modo monousuário). As características de cada nível de execução são definidas no arquivo **/etc/inittab**.

init normalmente lê seu nível de execução inicial a partir do arquivo **/etc/inittab**, mas o nível de execução também pode ser passado como um argumento a partir do carregador de inicialização (rotina de inicialização [*boot loader*].) Se for especificado "s", **init** entrará no modo monousuário. Caso contrário, ele varrerá **/etc/inittab** em busca de entradas que se aplicam ao nível de execução solicitado e executa seus comandos correspondentes.

O comando **telinit** altera o nível de execução de **init** assim que o sistema estiver ativo. Por exemplo, **telinit 4** força **init** ao nível de execução 4 (que não é utilizado em nossos sistemas de exemplo). O argumento mais útil do **telinit** é **q**, que faz com que **init** releia o arquivo **/etc/inittab**.

As distribuições Linux geralmente implementam uma camada adicional de abstração sobre o mecanismo básico de nível de execução fornecido por **init**. Essa camada extra permite que pacotes de software individuais instalem seus próprios scripts de inicialização sem modificar o arquivo **inittab** genérico do sistema. Trazer **init** para um novo nível de execução faz com que os scripts apropriados sejam executados com os argumentos **start** ou **stop**.

Uma discussão mais completa sobre o **init** e os scripts de inicialização começa na página 23.

29.2 CRON E ATD: AGENDANDO COMANDOS

O daemon **cron** (conhecido como **crond** no Red Hat) é responsável pela execução de comandos em momentos predeterminados. Ele aceita arquivos de agendamento ("crontabs") tanto de usuários como de administradores.

cron é empregado freqüentemente para fins administrativos, inclusive gerenciamento de arquivos de log e limpeza do sistema de arquivos. De fato, **cron** é tão importante para os administradores de sistema que foi dedicado um capítulo inteiro a ele. Esse capítulo, *Processos periódicos*, começa na página 106.

O daemon **atd** executa comandos agendados com o comando **at**. A maioria das versões do Linux também inclui o agendador **anacron**, que executa trabalhos em intervalos de tempo em vez de em horários específicos. **anacron** é particularmente útil em sistemas que nem sempre estão ligados, como laptops.

29.3 XINETD E INETD: DAEMONS DE GERENCIAMENTO

xinetd e **inetd** são daemons que gerenciam outros daemons. Eles iniciam seus daemons clientes quando eles têm trabalho e permitem que os clientes morram elegantemente uma vez que suas tarefas foram completadas.

A versão tradicional de **inetd** vem para nós do mundo UNIX, mas a maioria das distribuições Linux migrou para o **xinetd** de Panos Tsirigotis , uma alternativa "envenenada" que incorpora recursos de segurança semelhantes àqueles primeiramente alcançados pelo uso de **tcpd**, o pacote "empacotadores TCP". **xinetd** também fornece melhor proteção contra ataques de negação de serviço, melhores recursos de gerenciamento de log e uma linguagem de configuração mais flexível.

Infelizmente, o arquivo de configuração **inetd** não é diretamente compatível com o do **xinetd**. Discutimos primeiramente o **inetd** tradicional e depois damos uma olhada em **xinetd** em uma seção separada.

Entre nossas distribuições de exemplo, somente o Debian e o Ubuntu utilizam o padrão **inetd**; o RHEL, o Fedora e o SUSE todos assumem o padrão de **xinetd**. Você pode converter qualquer sistema para utilizar o gerenciador de daemons não padrão, mas não há realmente nenhuma razão para fazer isso.

xinetd e **inetd** só trabalham com daemons que fornecem serviços na rede. Descobrir quando alguém está tentando acessar um dos clientes deles, **xinetd** e **inetd** se anexam às portas de rede que normalmente seriam gerenciadas pelos daemons quiescentes. Quando ocorre uma conexão, **inetd** inicia o daemon apropriado e conecta seus canais de E/S padrão à porta da rede. Os daemons devem ser escritos com essa convenção em mente se tiverem de ser compatíveis.

Alguns daemons (como aqueles associados ao NIS e NFS) dependem de uma camada adicional de indireção conhecida como sistema Remote Procedure Call (RPC). O RPC foi originalmente desenvolvido e implementado pela Sun como uma maneira de promover o compartilhamento de informações em um ambiente de rede heterogêneo. As atribuições de porta para daemons que utilizam RPC são gerenciadas pelo daemon **portmap** (também conhecido como **rpcbind** em alguns sistemas), que é discutido mais adiante, neste capítulo.

Alguns daemons podem ser usados da maneira tradicional (na qual são inicializados uma só vez e continuam a ser executados até que o sistema seja desligado) ou via **xinetd/inetd**. Os daemons discutidos neste capítulo são marcados com o símbolo Ⓣ caso sejam compatíveis com **xinetd/inetd**.

Pelo fato de **xinetd/inetd** ser responsável pelo gerenciamento de vários serviços comuns baseados em rede, ele desempenha um papel importante na proteção do seu sistema. É importante verificar que apenas os serviços de que você precisa e

em que confia foram ativados. Em um sistema novo, você quase certamente precisará modificar sua configuração padrão para desativar serviços que são desnecessários ou indesejáveis em seu ambiente.

Configurando xinetd

O arquivo de configuração principal do **xinetd** é tradicionalmente o **/etc/xinetd.conf**, embora distribuições também forneçam comumente um diretório de configuração **/etc/xinetd.d**. Pacotes individuais podem inserir seus arquivos de configuração nesse diretório sem se preocuparem em sobrescrever as configurações de outros pacotes.

O exemplo abaixo mostra a definição de parâmetros padrão e a configuração de um serviço FTP em um sistema Red Hat Enterprise.

```
defaults
{
    instances             = 60
    log_type              = SYSLOG authpriv
    log_on_success        = HOST PID
    log_on_failure        = HOST
    cps                   = 25 30
}

service ftp
{
    # Instâncias ilimitadas porque wu.ftpd faz seu próprio gerenciamento de carga
    socket_type           = stream
    protocol                    = tcp
    wait                  = no
    user                  = root
    server                = /usr/sbin/wu.ftpd
    server_args           = -a
    instances                   = UNLIMITED
    only_from                   = 128.138.0.0/16
    log_on_success        += DURATION
}

includedir /etc/xinetd.d
...
```

A Tabela 29.1 fornece um miniglossário de parâmetros.

Tabela 29.1 Parâmetros de configuração xinetd (uma lista não-exaustiva)

Parâmetro	Valor	Significado
bind	*endereçoIP/host*	Interface na qual se deve disponibilizar esse serviço
cps	*tempo de espera num*	Limita o número de conexões por segundo
disable	sim/não	Desabilita serviço; mais fácil do que comentá-lo
include	*caminho*	Lê o caminho listado como um arquivo de configuração suplementar
includedir	*caminho*	Lê todos os arquivos no diretório especificado
instances	*número* ou UNLIMITED	Número máximo de instâncias simultâneas de um determinado serviço
log_on_failure	*especial*[b]	Informações para log de falhas ou recursos de acesso[b]
log_on_success	*especial*[b]	Informações para registrar em log conexões bem-sucedidas[b]
log_type	*especial*[b]	Configura arquivo de log ou parâmetros de syslog
max_load	*num*	Desativa o serviço se a carga média for > limiar
nice	*número*	Valor **nice** dos processos de servidor gerados
no_access	*listaDeCorrespondência*	Nega serviço a endereços IP especificados
only_from	*listaDeCorrespondência*	Só aceita requisições de endereços especificados
per_source	*número*	Limita o número de instâncias por máquina remota
protocol	tcp/udp	Protocolo de serviços
server	*caminho*	Caminho para o binário de servidor
server_args	*string*	Argumentos de linha de comando para o servidor[c]

Parâmetro	Valor	Significado
socket_type	stream/dgram	Usa stream para serviços TCP, dgram para UDP
user	*nomeDoUsuário*	Usuário (UID) como qual o serviço deve ser executado
wait	yes/no	**xinetd** deve parar de se intrometer até que o daemon seja encerrado?

a. Um ou mais valores de uma lista definida muito longa para merecer uma reprodução nesta tabela.
b. Note que a diretiva USERID utilizada com esses parâmetros fará com que **xinetd** realize consultas IDENT nas conexões, normalmente resultando em atrasos significativos.
c. Diferentemente de **inetd**, **xinetd** não requer que o comando de servidor seja o primeiro argumento.

Alguns parâmetros de **xinetd** podem aceitar atribuições na forma += ou -= (conforme visto no valor log_on_success para o servidor FTP) para modificar os valores padrão em vez de substituí-los totalmente. Apenas alguns parâmetros são realmente necessários para cada serviço.

Podem ser especificadas listas de concordância de endereços para os parâmetros only_from e no_access em diversos formatos. Os mais úteis são os endereços IP no formato CIDR com uma máscara explícita (conforme mostrado no exemplo) e nomes de host ou domínios como boulder.colorado.edu e .colorado.edu — observe o ponto precedente. Várias especificações podem ser separadas com um espaço (como em todas as listas **xinetd**).

xinetd pode tanto registrar diretamente em um arquivo de log como enviar entradas de log para syslog. Já que o volume de informações de log pode ser potencialmente elevado em um servidor atarefado, pode ser que faça sentido usar registro direto em arquivo por questões de desempenho. Tenha em mente que registrar informações de log num arquivo é menos seguro do que enviá-las para um servidor remoto por meio de syslog, pois um hacker que ganhe acesso ao sistema local pode adulterar os arquivos de log.

xinetd é capaz de fornecer alguns serviços interessantes como encaminhamento de requisições a um host interno que não é visível ao mundo externo. Vale a pena revisar a página de manual referente a **xinetd** para ter uma idéia de seus recursos.

Configurando inetd

O Debian e o Ubuntu são as únicas distribuições Linux importantes que ainda utilizam o tradicional **inetd**. Essa versão de **inetd** consulta **/etc/inetd.conf** para determinar em que portas de rede ele que deve ouvir. O arquivo de configuração inclui quase as mesmas informações do **xinetd.conf**, mas utiliza um formato tabular (em vez de uma lista de atributos/valores). Eis um exemplo (sintetizado) de um sistema Debian:

```
# Example /etc/inetd.conf - from a Debian system

#:INTERNAL: Internal services
#echo      stream tcp    nowait root   internal
#echo      dgram  udp    wait   root   internal
...
#time      stream tcp    nowait root   internal
#time      dgram  udp    wait   root   internal

#:STANDARD: These are standard services.
#:BSD: Shell, login, exec and talk are BSD protocols.

#:MAIL: Mail, news and uucp services.
imap2      stream tcp    nowait root   /usr/sbin/tcpd /usr/sbin/imapd
imaps      stream tcp    nowait root   /usr/sbin/tcpd /usr/sbin/imapd

#:INFO: Info services
ident      stream tcp    wait   identd /usr/sbin/identd identd
...
#:OTHER: Other services
swat       stream tcp    nowait.400     root   /usr/sbin/swat swat
# finger   stream tcp    nowait nobody /usr/sbin/tcpd in.fingerd -w
391002/1-2 stream rpc/tcp wait   root   /usr/sbin/famd fam
...
```

A primeira coluna contém o nome do serviço. **inetd** associa nomes de serviços a números de portas consultando o arquivo **/etc/services** (para serviços TCP e UDP) ou então o arquivo **/etc/rpc** e o daemon **portmap** (para serviços RPC). Os serviços RPC são identificados por nomes da forma *nome/número* e a designação rpc na coluna três. No exemplo anterior, a última linha é um serviço RPC.

O único serviço RPC que é comumente controlado por **inetd** é **mountd**, o daemon de montagem NFS. As distribuições Linux parecem executar esse daemon da maneira antiga (iniciando-o no momento da inicialização), de forma que talvez você não tenha serviços RPC nenhum em seu arquivo **inetd.conf**.

Em um host com mais de uma interface de rede, você pode anteceder o nome do serviço com uma lista de endereços IP separados por vírgulas ou nomes de host simbólicos para especificar as interfaces nas quais **inetd** deve escutar requisições de serviços. Por exemplo, a linha

```
inura:time stream tcp    nowait root   internal
```

fornece o serviço time apenas na interface associada com o nome inura em DNS, NIS ou o arquivo **/etc/hosts**.

A segunda coluna determina o tipo de soquete que o serviço usará e é invariavelmente `stream` ou `dgram`. `stream` é usado com serviços TCP (orientados a conexões) e `dgram` é usado com UDP; mas, alguns serviços usam ambos, por exemplo, **bind**.

A terceira coluna identifica o protocolo de comunicação utilizado pelo serviço. Os tipos admissíveis são listados no arquivo **protocols** (usualmente em **/etc**). O protocolo é quase sempre `tcp` ou `udp`. Os serviços RPC prefixam `rpc/` com o tipo de protocolo, como com `rpc/tcp` e `rpc/udp` no exemplo anterior.

Se o serviço que está sendo descrito é capaz de processar várias requisições por vez (e não processar uma requisição e sair), a coluna quatro deve ser configurada como `wait`. Essa opção permite que o daemon gerado assuma o controle da porta enquanto ele estiver em execução; **inetd** aguarda o daemon sair antes de retomar seu monitoramento da porta. O contrário de `wait` é `nowait`; ele faz com que **inetd** monitore continuamente e crie uma cópia do daemon cada vez que ele receber uma requisição. A seleção de `wait` ou `nowait` tem de corresponder ao verdadeiro comportamento do daemon e não deve ser configurada arbitrariamente. Ao configurar um novo daemon, consulte o arquivo **inetd.conf** para ter um exemplo de linha de configuração ou consulte as páginas de manual para o daemon em questão.

A forma `nowait.400`, utilizada na linha de configuração para **swat**, significa que **inetd** deve gerar no máximo 400 instâncias do daemon de servidor por minuto. O padrão é mais conservador, 40 instâncias por minuto. Dada a natureza desse serviço (uma ferramenta administrativa para o Samba), não fica claro por que o limiar regulador foi aumentado.

A quinta coluna dá o nome de usuário sob o qual o daemon deve ser executado. É sempre mais seguro executar um daemon como um usuário diferente do root se isso for possível. No exemplo anterior, **in.fingerd** teria executado como o usuário convidado nobody (se a linha não fosse comentada).

Os campos remanescentes dão o nome de caminho totalmente qualificado do daemon e seus argumentos de linha de comando. A palavra-chave `internal` indica serviços cujas implementações são fornecidas pelo próprio **inetd**.

A maioria das entradas de serviço nesse exemplo executa seus daemons via **tcpd** em vez de executá-los diretamente. **tcpd** registra em log tentativas de conexão e implementa controle de acesso baseado na origem da tentativa de conexão. Em geral, todos os serviços devem ser protegidos com **tcpd**. Esse modelo de configuração apresenta um problema de segurança em potencial já que **swat**, um utilitário de configuração para compartilhamento de arquivos, não é protegido.[2]

No padrão **inetd.conf** distribuído com o Debian, os servidores de **rlogin**, **telnet**, **finger** e **rexec** nem mesmo são mais listados. Consulte a seção *Problemas de segurança diversos* na página 473 para mais informações relacionadas com segurança.

Consulte o Capítulo 10 para informações adicionais sobre syslog.

Depois de editar **/etc/inetd.conf**, envie um sinal HUP para **inetd** para pedir para ele reler seu arquivo de configuração e implementar qualquer alteração que você tenha feito. Depois de feita a sinalização, aguarde um momento e verifique os arquivos de log em busca de mensagens de erro relacionadas às suas alterações (**inetd** registra os erros em syslog sob o recurso "daemon"). Teste qualquer serviço novo que você tenha adicionado para garantir que funcione corretamente.

O arquivo services

Após adicionar um novo serviço a **inetd.conf** ou **xinetd.conf**, você também pode precisar criar uma entrada para ele no arquivo **/etc/services**. Esse arquivo é utilizado por várias rotinas de biblioteca padrão que mapeiam entre nomes de serviços e números de portas. **xinetd** na verdade permite que você especifique diretamente o número da porta, mas sempre é uma boa idéia manter uma lista mestra das portas no arquivo **services**.

Por exemplo, ao digitar o comando

```
$ telnet anchor smtp
```

telnet procura o número da porta para o serviço "smtp" no arquivo **services**. A maioria dos sistemas vem com os serviços comuns já configurados; você só precisa editar o arquivo **services** caso acrescente algo novo.

---

2. Se não estiver utilizando **tcpd** para proteger um serviço, o primeiro argumento de linha de comando do daemon sempre deve ser o nome reduzido do próprio daemon. Essa exigência não é uma peculiaridade de **inetd**, mas uma convenção tradicional do UNIX que normalmente é oculta pelo shell.

O arquivo **services** só é utilizado para serviços TCP/IP de boa-fé; informações semelhantes de serviços RPC são armazenadas em um arquivo de configuração separado, normalmente **/etc/rpc**.

Eis algumas linhas selecionadas de um arquivo **services** (o original tem cerca de 570 linhas de comprimento):

```
tcpmux     1/tcp                # TCP port multiplexer
echo       7/tcp
echo       7/udp
...
ssh        22/tcp               #SSH Remote Login Protocol
ssh        22/udp               #SSH Remote Login Protocol
smtp       25/tcp  mail
rlp        39/udp  resource     # resource location
name       42/tcp               # IEN 116
domain     53/tcp               # name-domain server
domain     53/udp
...
```

O formato de uma linha é

```
nome       porta/proto    aliases# comentário
```

Em geral, os serviços são listados em ordem numérica, embora essa ordem não seja solicitada. *nome* é o nome simbólico do serviço (o nome que você utiliza no arquivo **inetd.conf** ou **xinetd.conf**). *porta* é o número da porta em que o serviço normalmente ouve; se o serviço for gerenciado por **inetd**, ele é a porta em que **inetd** irá ouvir.[3]

proto estipula o protocolo utilizado pelo serviço; na prática, é sempre `tcp` ou `udp`. Se um serviço puder usar tanto UDP quanto TCP, terá de ser incluída uma linha para cada (como acontece com o serviço `ssh` anterior). O campo *alias* contém nomes adicionais para o serviço (por exemplo, `whois` também pode ser pesquisado como `nicname`.).

portmap: mapeando serviços RPC para portas TCP e UDP

portmap mapeia números de serviços RPC para portas TCP/IP nas quais seus servidores estão escutando. Quando um servidor RPC é inicializado, ele se registra com **portmap**, listando os serviços que suporta e a porta na qual pode ser contatado. Clientes consultam **portmap** para descobrir como manter contato com o servidor apropriado.

Esse sistema permite que uma porta seja mapeada para um nome de serviço simbólico. Ele é basicamente mais um nível de abstração acima do arquivo **services**, muito embora um que introduza complexidade adicional (e problemas de segurança) sem solucionar quaisquer problemas práticos. Muitos programas RPC são de fato iniciados pelo **inetd**, fornecendo assim não uma, mas duas camadas de indireção e desvirtuamento.

Se o daemon **portmap** morrer, todos os serviços que dependem dele (inclusive **inetd** e NFS) terão de ser reinicializados. Em termos práticos, isso significa que é chegado o momento de reinicializar o sistema. **portmap** tem de ser iniciado antes de **inetd** para **inetd** manipular serviços RPC corretamente.

29.4 DAEMONS DE KERNEL

Por razões arquitetônicas, algumas partes do kernel do Linux são controladas como se fossem processos de usuário. Em kernels mais antigos, esses processos poderiam ser identificados por seus PIDs baixos e nomes que iniciam com **k**, como **kupdate**, **kswapd**, **keventd** e **kapm**. A atribuição de nomes é menos consistente sob os kernels 2.6, mas **ps** sempre mostra os nomes de threads de kernel entre colchetes.

Em grande parte, esses processos lidam com vários aspectos da E/S, gerenciamento de memória e sincronização do cache de disco. Eles não podem ser manipulados pelo administrador de sistemas e devem ser deixados de lado.[4]

A Tabela 29.2 a seguir resume brevemente as funções dos daemons importantes. Os daemons que incluem um parâmetro N em seus nomes (como mostrado por **ps**) executam separadamente em cada CPU de um sistema de múltiplas CPUs; o N informa qual cópia vai com qual CPU.

---

3. Os números das portas não são arbitrários. Todas as máquinas têm de concordar sobre quais serviços se associam a quais portas; caso contrário, as requisições serão constantemente direcionadas para a porta errada. Se você estiver criando um serviço específico a um site, selecione um número de porta alto (maior do que 1023) que já não se encontre listado no arquivo **services**.

4. Se estiver familiarizado com a implementação do kernel, é ocasionalmente útil alterar essas prioridades de execução dos processos. Entretanto, essa não é uma tarefa administrativa padrão.

Tabela 29.2 Daemons de kernel importantes (kernels 2.6)

Daemon	Função
ksoftirqd/*N*	Trata interrupções de software quando a carga for alta
kacpid	Lida com o subsistema ACPI
kblockd/*N*	Bloqueia o trabalho do subsistema
aio/*N*	Tenta novamente as E/Ss assíncronas
kswapd*N*	Move páginas para fazer swap
ata/*N*	Faz processamento para suporte ATA serial
scsi_eh_*N*	Realiza tratamento de erro SCSI
kjournald	Suporta sistemas de arquivos de journaling
events/*N*	Faz processamento de fila de trabalho genérico

Um outro daemon de sistema nessa categoria, muito embora um com um nome não padrão, é **mdrecoveryd**. Ele faz parte da implementação "dispositivos múltiplos", mais comumente conhecida como RAID.

klogd: lendo mensagens do kernel

klogd é responsável pela leitura de entradas log do buffer de mensagem do kernel e pelo seu encaminhamento para o syslog a fim de elas possam ser roteadas para o destino final. Ele também pode processar mensagens se configurado para fazer isso. Consulte a seção *O kernel e o registro em log em tempo de inicialização*, na página 147, para mais informações.

29.5 DAEMONS DE IMPRESSÃO

Vários sistemas de impressão estão em uso comum e cada um tem sua própria família de comandos e daemons que fornecem serviços relacionados à impressão. Em alguns casos as famílias foram hibridizadas; em outros, várias variantes são executadas em um único sistema.

cupsd: agendador para o Common UNIX Printing System

Consulte o Capítulo 23 para informações adicionais sobre CUPS.

O CUPS fornece um recurso de impressão portável implementando a versão 1.1 do Internet Printing Protocol. Ele permite que os usuários remotos imprimam para seus escritórios (ou vice-versa) utilizando uma interface Web. O CUPS tornou-se bem popular e é o gerenciador de impressão padrão da maioria dos sistemas. É suficientemente flexível para permitir autenticação remota.

lpd: gerenciando impressão

lpd é responsável pelo sistema de spool de impressão BSD no estilo antigo. Ele aceita tarefas de usuários e cria processos que realizam a impressão em si. **lpd** também é responsável pela transferência de tarefas de impressão a partir de e para sistemas remotos. **lpd** pode travar alguma vezes e precisar ser reiniciado manualmente.

Seu sistema pode ter a versão original do **lpd** ou a novíssima versão que faz parte do pacote LPRng. Veja o Capítulo 23, *Impressão*, para obter informações adicionais sobre essas alternativas.

29.6 DAEMONS DE SERVIÇOS DE ARQUIVOS

Os daemons a seguir fazem parte do sistema de compartilhamento de arquivos NFS ou Samba. Aqui damos apenas uma breve descrição de suas funções. O NFS é descrito em detalhes no Capítulo 16, *O sistema de arquivos de rede*, e o Samba é discutido a partir da página 568.

rpc.nfsd: servindo arquivos

rpc.nfsd executa em servidores de arquivos e trata requisições de clientes NFS. Na maioria das implementações do NFS, **nfsd** é realmente apenas uma parte do kernel que foi elegantemente vestida como um processo por motivos de escalonamento. O Linux na verdade ostenta duas implementações diferentes, uma das quais segue essa convenção e outra que é executada no espaço de usuário. A implementação no kernel é mais popular e é o padrão para a maioria das distribuições.

rpc.nfsd aceita um único argumento que especifica quantas cópias de si próprio ele deve criar. Há certa magia negra envolvida na escolha do número correto de cópias; consulte a página 342.

rpc.mountd: respondendo a requisições de montagem

rpc.mountd aceita requisições de montagem de sistemas de arquivos de clientes NFS em potencial. Ele verifica se cada cliente tem permissão para montar os diretórios solicitados. **rpc.mountd** consulta o arquivo **/var/state/nfs/xtab** para determinar que candidatos são legítimos.

amd e automount: montando sistemas de arquivos sob demanda

amd e **automount** são automontadores NFS, daemons que aguardam até que um processo tente usar um sistema de arquivos antes de eles realmente o montarem. Os automontadores posteriormente desmontam os sistemas de arquivos caso não tenham sido acessados em um período especificado.

O emprego de automontadores é muito útil em ambientes grandes em que dezenas ou centenas de sistemas de arquivos são compartilhados na rede. Os automontadores aumentam a estabilidade da rede e reduzem a complexidade da configuração, já que todos os sistemas da rede podem compartilhar a mesma configuração **amd** ou **automountd**. Discutimos o uso do automontador Linux padrão detalhadamente a partir da página 346.

rpc.lockd e rpc.statd: gerenciando bloqueios NFS

Embora **rpc.lockd** e **rpc.statd** sejam daemons distintos, eles sempre executam como uma equipe. **rpc.lockd** mantém bloqueios consultivos (*a la* **flock** e **lockf**) em arquivos NFS. **rpc.statd** permite que processos monitorem o status de outras máquinas que estão executando NFS. **rpc.lockd** utiliza **rpc.statd** para decidir quando tentar se comunicar com uma máquina remota.

rpciod: blocos NFS de cache

rpciod armazena em cache requisições de leitura e de gravação em clientes NFS. Ele realiza o armazenamento em buffer tanto de leitura antecipada (*read-ahead*) como de gravação retardada (*write-behind*) e aprimora muito o desempenho do NFS. Esse daemon é análogo aos daemons **biod** e **nfsiod** encontrados em outros sistemas, embora seja algo estruturalmente diferente.

rpc.rquotad: servindo cotações remotas

rpc.rquotad permite que usuários remotos verifiquem suas cotas em sistemas de arquivos que eles montaram com NFS. A verdadeira implementação das restrições de cotas ainda é realizada no servidor; **rpc.rquotad** simplesmente faz com que o comando **quota** funcione corretamente.

smbd: fornecendo serviço de arquivo e impressão para clientes Windows

smbd é o servidor de arquivos e de impressão na família Samba. Ele fornece serviços de compartilhamento de arquivos e de impressão por meio do protocolo Windows conhecido de forma variada como SMB ou CIFS. Consulte a página 568 para mais detalhes.

nmbd: servidor de nome NetBIOS

nmbd é mais um componente do Samba. Ele responde a requisições de serviços de nomes NetBIOS geradas por máquinas Windows. Ele também implementa o protocolo de navegação que máquinas Windows usam para preencher a pasta Meus Locais de Rede e torna visíveis os discos compartilhados no host local. **nmbd** também pode ser usado como servidor WINS.

29.7 DAEMONS DE BANCO DE DADOS ADMINISTRATIVOS

Vários daemons estão associados com o sistema de bancos de dados administrativos NIS da Sun, que é descrito no Capítulo 17, *Compartilhando arquivos de sistema*. Embora o NIS tenha se originado na Sun, agora ele também é utilizado em sistemas de muitos outros fabricantes, inclusive o Linux.

ypbind: localizando servidores NIS

O daemon **ypbind** funciona em todos os clientes e servidores NIS. Ele localiza um servidor NIS para o qual consultas podem ser direcionadas. **ypbind** não processa realmente as requisições em si; ele simplesmente informa aos programas clientes que servidor usar.

ypserv: servidor NIS

ypserv é executado em todos os servidores NIS. **ypserv** aceita consultas de clientes e responde com as informações solicitadas. Consulte a página 359 para informações sobre como configurar as máquinas que executam **ypserv**.

rpc.ypxfrd: transferindo bancos de dados NIS

rpc.ypxfrd transfere eficientemente bancos de dados NIS para servidores escravos. Um escravo inicia uma transferência com o comando **ypxfr**. Toda vez que um banco de dados for alterado no mestre, ele deve ser imediatamente transferido para todos os escravos de forma que os servidores NIS permaneçam consistentes entre si.

lwresd: servidor de biblioteca de conversão leve

lwresd fornece um método rápido de armazenamento em cache de pesquisas de endereço de endereço-hostname e hostname-endereço. Ele é contatado por um conversor de stub que faz parte das bibliotecas padrão do sistema e é chamado diretamente por muitos programas. A biblioteca e o daemon se comunicam por um protocolo UDP simples.

nscd: daemon de cache de serviço de nomes

nscd armazena em cache os resultados de chamadas das rotinas de biblioteca C padrão nas famílias **getpw***, **getgr*** e **gethost***, que pesquisam dados tradicionalmente armazenados nos arquivos **passwd**, **group** e **hosts**. Nos dias de hoje, a gama de fontes potenciais é maior e inclui opções como NIS e DNS. **nscd** realmente não sabe de onde os dados provêm; ele simplesmente coloca os resultados em cache e os usa para curto-circuitar chamadas de bibliotecas subseqüentes. A política de autenticação é configurada no arquivo **/nscd.conf**.

29.8 DAEMONS DE E-MAILS

Além do **sendmail** básico e dos sistemas de entrega de correio Postfix, ambos os quais têm amplo uso, vários daemons facilitam o acesso remoto a caixas de correio.

Ⓣ sendmail: transporte de e-mail

As tarefas de **sendmail** incluem aceitar mensagens de usuários e sites remotos, regravar endereços, expandir nomes alternativos e transferir e-mails por meio da Internet. **sendmail** é um daemon importante e muito complexo. Consulte o Capítulo 18, *E-mail*, para informações completas.

Ⓣ smtpd: daemon Simple Mail Transport Protocol

smtpd ouve na porta 25 as mensagens de e-mails que chegam e as encaminha para seu sistema back-end de transporte para mais processamento. Consulte as páginas 374 e 431 para informações adicionais sobre o uso de **smtpd** no **sendmail** e sistemas Postfix.

Ⓣ popd: servidor básico de caixa de correio

O daemon **popd** implementa o POP (Post Office Protocol). Esse protocolo é comumente utilizado por sistemas não-Linux para receber e-mails.

Ⓣ imapd: servidor de luxo de caixa de correio

O daemon **imapd** implementa o IMAP (Internet Mail Access Protocol, de acesso a e-mails na Internet), que é uma alternativa mais festiva e cheia de recursos do que o POP. Ele permite que usuários baseados em PC (ou usuários Linux com leitores de e-mail preparados para o padrão IMAP) acessem seus e-mails de uma série de localidades, com pastas de e-mail armazenadas no servidor Linux. Confira www.imap.org para mais informações sobre o IMAP.

29.9 DAEMONS DE EXECUÇÃO DE LOGIN E COMANDO REMOTO

A capacidade de efetuar login na rede e nela executar comandos tornou-se uma das primeiras motivações para o desenvolvimento de rede UNIX, e esse recurso ainda é um componente básico da administração dos sistemas atuais. Infelizmente, levou várias décadas para a comunidade UNIX alcançar uma apreciação madura das implicações de segurança dessa tecnologia. Os modernos sistemas de produção devem empregar o SSH (**sshd**) e quase mais nada.

Ⓣ sshd: servidor seguro de login remoto

sshd fornece serviços similares a **in.rlogind**, mas suas sessões são transportadas (e autenticadas) por meio de um duto criptografado. Uma série de algoritmos criptográficos está disponível. Por causa do ambiente da Internet hoje, você deve permitir acesso de shell a partir da Internet *somente* por meio de um daemon como este — *não* **in.rlogind** ou **in.telnetd**. Você pode localizar informações adicionais sobre **sshd** a partir da página 481.

in.rlogind: servidor obsoleto de login remoto

in.rlogind foi o padrão inicial para tratar logins remotos. Quando invocado por **inetd**, ele tenta autenticar automaticamente o usuário remoto examinando o arquivo **~/.rhosts** do usuário local e o **/etc/hosts.equiv** no nível de sistema. Se a autenticação automática for bem-sucedida, o usuário é conectado diretamente. Caso contrário, **in.rlogind** executa o programa **login** para solicitar ao usuário uma senha. Devido à sua fácil e barata autenticação, **in.rlogind** é um tanto preocupante em termos de segurança. Consulte a página 473 para mais comentários sobre o assunto.

in.telnetd: ainda outro servidor de login remoto

in.telnetd é similar a **in.rlogind**, exceto pelo fato de usar o protocolo TELNET. Esse protocolo permite que ambos os lados (cliente e servidor) negociem configurações duplex e controle de fluxo, tornando-o uma melhor opção do que **in.rlogind** para links que são lentos ou não confiáveis. Da mesma forma que **rlogin**, **telnet** transmite senhas em texto puro por meio da rede. Portanto, o seu emprego é desencorajado em redes modernas. Entretanto, muitos sistemas não-Linux suportam **telnet**.

in.rshd: servidor de execução de comando remoto

in.rshd trata requisições de execução de comando remota a partir de **rsh** e **rcmd**

O processo de autenticação aplicado por **in.rshd** é similar àquele de **in.rlogind**, exceto pelo fato de que se a autenticação automática não funcionar, **in.rshd** negará a solicitação sem permitir que o usuário forneça uma senha. **in.rshd** também é o servidor para **rcp** (cópia remota). Como **in.rlogind**, **in.rshd** tornou-se algo como um albatroz da segurança e está invariavelmente desativado. Consulte a página 473 para informações adicionais.

29.10 DAEMONS DE INICIALIZAÇÃO E CONFIGURAÇÃO

Nos anos 80, o mundo UNIX foi varrido por uma mania de estações de trabalho sem disco ("diskless"). Essas máquinas eram inicializadas completamente pela rede e realizavam todas as suas operações de E/S por meio de uma tecnologia de sistemas de arquivos remotos como o NFS. À medida que os preços dos discos foram caindo e suas velocidades aumentando, o interesse por estações de trabalho sem disco desapareceu rapidamente. Eles poderiam voltar à moda a qualquer momento, mas, como os sapatos de plataforma da década de 1970. Os dois principais remanescentes da era sem disco são uma pletora de daemons destinados ao suporte de sistemas sem disco e à bizarra organização de sistemas de arquivos da maioria dos fornecedores.

Para os curiosos, discutimos os sistemas sem disco em si em mais detalhes a partir da página 164.

Embora as estações de trabalho sem disco não sejam mais comuns hoje em dia, seus protocolos de inicialização foram usurpados por outros dispositivos. A maioria das impressoras de rede e hubs de rede gerenciáveis faz a inicialização por meio de alguma combinação dos serviços listados nesta seção.

dhcpd: atribuição de endereço dinâmica

O DHCP (Dynamic Host Configuration Protocol) fornece aos PCs, laptops e outras plataformas "móveis" informações sobre seus endereços IP, gateway padrão e servidor de nomes em tempo de inicialização. **dhcpd** é o daemon que implementa esse serviço no Linux. Você pode encontrar informações adicionais sobre DHCP na página 217. Uma elaboração mais sofisticada de DHCP chamada PXE (Pre-boot eXecution Environment) ajuda máquinas compatíveis a inicializar a partir da rede sem a necessidade de um dispositivo de inicialização local; consulte a página 160 para mais detalhes.

in.tftpd: servidor de transferência de arquivos trivial

in.tftpd implementa um protocolo de transferência de arquivos similar àquele do **ftpd**, mas muito, muito mais simples. Muitos sistemas sem disco usam TFTP para fazer o download de seus kernels em um servidor. **in.tftpd** não executa autenticação, mas ele normalmente é restrito a servir arquivos num único diretório (normalmente **/tftpboot**). Como qualquer coisa colocada no diretório TFTP passa a ser acessível a toda rede, o diretório deve conter somente arquivos de inicialização e não deve ser gravável por todos.

rpc.bootparamd: suporte vital sem disco avançado

rpc.bootparamd usa o arquivo **/etc/bootparams** para informar clientes sem disco onde encontrar seus sistemas de arquivos. O serviço **rpc.bootparamd** é normalmente utilizado por máquinas que obtêm seus endereços IP usando RARP e que utilizam NFS para montar seus sistemas de arquivos.

hald: daemon da Hardware Abstraction Layer (HAL)

hald coleta informações sobre o hardware do sistema a partir de várias fontes. Ele fornece uma lista de dispositivo vital por meio do D-BUS.

udevd: serializando avisos de conexão de dispositivo

udevd é uma parte menor do sistema de atribuição de nomes de dispositivo dinâmico **udev**. Ele permite a serialização adequada de eventos de conexão automática (*hot-plug events*), com o qual às vezes o kernel pode se comunicar fora de ordem com o espaço de usuário.

29.11 OUTROS DAEMONS DE REDE

Todos os daemons a seguir utilizam protocolos de Internet para tratar requisições. Entretanto, muitos desses daemons de "Internet" realmente passam a maior parte do tempo servindo requisições locais.

Ⓣ talkd: serviço de chat de rede

As solicitações de conexão do programa **talk** são manipuladas por **talkd**. Ao receber uma requisição, **talkd** negocia com a outra máquina para estabelecer uma conexão de rede entre os dois usuários que executaram **talk**.

Há duas versões do **talk**: a original (na porta 517) e uma mais nova a partir de 4.3BSD (**ntalk**, na porta 518). **ntalk** não é retrocompatível e não aceitará conexões de clientes **talk**. Embora o 4.3BSD tenha sido lançado em 1986, muitos sistemas ainda utilizam a versão mais "antiga" do talk (15 anos depois!).

snmpd: fornecendo serviço de gerenciamento de rede remoto

snmpd responde a requisições que usam o protocolo SNMP (Simple Network Management Protocol). O SNMP padroniza algumas operações de rede comuns. Consulte a página 455 para mais informações sobre o SNMP.

Ⓣ ftpd: servidor de transferência de arquivos

Consulte a página 506 para informações adicionais sobre o ftpd.

ftpd é o daemon que manipula solicitações de **ftp**, o programa de transferência de arquivos da Internet. Muitos sites o desativam, normalmente porque se preocupam com a segurança. **ftpd** pode ser configurado para permitir que qualquer pessoa transfira arquivos para e a partir de sua máquina.

Uma variedade de implementações **ftpd** se encontra disponível para sistemas Linux. Se planeja usar um servidor com tráfego intenso ou precisa de recursos avançados como gerenciamento de carga, pode ser sensato investigar as alternativas para o **ftpd** padrão de sua distribuição.

O WU-FTPD, desenvolvido na Universidade de Washington, é uma das alternativas mais populares ao padrão **ftpd**. Consulte o site www.wu-ftpd.org para informações adicionais.

Ⓣ rsyncd: sincronizando arquivos entre múltiplos hosts

rsyncd é realmente apenas um link ao comando **rsync**; a opção **--daemon** o transforma em um processo de servidor. **rsyncd** facilita a sincronização de arquivos entre os hosts. Ele é, essencialmente, uma versão eficiente e atenta à segurança de **rcp**. **rsync** é um verdadeiro achado para os administradores de sistemas, e neste livro descrevemos seu uso em uma série de contextos diferentes. Consulte a página 353 para informações genéricas e algumas dicas sobre como usar **rsync** para compartilhar arquivos de sistema. **rsync** também é um grande participante dos processos de instalação internos de muitos sites.

routed: mantendo tabelas de roteamento

routed mantém as informações de roteamento usadas por TCP/IP para enviar e encaminhar pacotes em uma rede. **routed** lida somente com roteamento dinâmico; rotas que são definidas estaticamente (isto é, embutidas na tabela de roteamento do sistema com o comando **route**) nunca são modificadas por **routed**. **routed** é relativamente estúpido e ineficiente; só recomendamos seu uso em algumas poucas situações específicas. Consulte a página 239 para uma discussão mais detalhada de **routed**.

gated: mantendo tabelas de roteamento complicadas

gated compreende vários protocolos de roteamento, inclusive RIP, o protocolo utilizado por **routed**. **gated** transforma informações de roteamento entre os diversos protocolos e é muito maleável em termos de configuração. Ele também pode ser muito mais gentil com sua rede do que **routed**. Consulte a página 239 para obter informações adicionais sobre **gated**.

named: servidor de DNS

named é o servidor mais popular para DNS (Domain Name System). Ele mapeia nomes de host em endereços de rede e realiza muitos outros recursos e truques, todos usando um banco de dados distribuído mantido por **nameds** em toda parte. O Capítulo 15, *DNS: o Domain Name System*, descreve os cuidados e como alimentar **named**.

syslogd: processando mensagens de log

Consulte a página 149 para informações sobre syslog.

syslogd atua como uma alfândega para informações de estados e mensagens de erro geradas pelos daemons e software de sistema. Antes de **syslogd** ter sido desenvolvido, os daemons gravavam suas mensagens de erro diretamente na console do sistema ou mantinham seus próprios arquivos de log privados. Agora eles usam a rotina de biblioteca **syslog** para transferir as mensagens para **syslogd**, que as organiza de acordo com regras estabelecidas pelo administrador do sistema.

Ⓣ in.fingerd: pesquisando usuários

in.fingerd fornece informações sobre os usuários conectados ao sistema. Se solicitado, ele também pode fornecer mais detalhes sobre usuários individuais. **in.fingerd** realmente não faz muito por si só: ele simplesmente aceita linhas de entrada e as passa ao programa **finger** local.

finger é capaz de retornar uma quantidade razoável de informações sobre um usuário, inclusive o estado de login do usuário, o conteúdo do campo GECOS do usuário em **/etc/passwd** e o conteúdo dos arquivos **~/.plan** e **~/.project** do usuário.

Se estiver conectado à Internet e usando **in.fingerd**, qualquer pessoa no mundo poderá obter essas informações. **in.fingerd** realmente ativou muitos serviços interessantes (como as listas telefônicas na Internet), mas também permitiu que pessoas implementassem uma série de esquemas fraudulentos, como localizar pessoas e ligar para elas para tentar vender coisas ou serviços e para localizar endereços candidatos a receber spam. Alguns sites responderam a essa invasão desativando **in.fingerd**, ao passo que outros simplesmente restringiram a quantidade de informações por ele retornada. Não suponha que porque **in.fingerd** é simples, ele seja necessariamente seguro — um ataque buffer overflow (estouro de buffer) contra esse daemon foi explorado pelo worm Internet original de 1988.

Ⓣ httpd: servidor World Wide Web

httpd permite que seu site se torne um servidor Web. **httpd** é capaz de enviar texto, imagens e som a seus clientes. Veja o Capítulo 21, *Hospedagem Web e servidores Internet*, para informações adicionais sobre como servir páginas Web.

29.12 NTPD: DAEMON DE SINCRONIZAÇÃO DE TEMPO

À medida que os computadores ficaram cada vez mais interdependentes, tornou-se cada vez mais importante para eles compartilhar uma idéia de tempo consistente. Relógios sincronizados são essenciais para correlacionar entradas de arquivos de log no caso de uma brecha de segurança e eles também são importantes para uma série de aplicativos de usuário final, desde projetos conjuntos de desenvolvimento de software até o processamento de transações financeiras.

ntpd[5] implementa o Network Time Protocol, que permite aos computadores sincronizar seus relógios a milissegundos um do outro. A primeira implementação NTP começou por volta de 1980 com uma exatidão de várias centenas de milissegundos. Hoje, um novo modelo de relógio de kernel pode manter tempo com uma precisão de até um nanossegundo. A última versão do protocolo (versão 4, documentado na RFC2783) mantém compatibilidade com as versões anteriores e acrescenta fácil configuração e alguns recursos de segurança.

Os servidores NTP são organizados em uma hierarquia, da qual cada nível é chamado de "estrato". O tempo em servidores estrato 1 é em geral escravizado a um clock de referência externa como um receptor de rádio ou relógio atômico. Os servidores estrato 2 configuram seus relógios usando servidores estrato 1 e atuam como centros de distribuição de tempo. São oferecidos até 16 estratos. Para determinar seu próprio estrato, um servidor de tempo simplesmente adiciona 1 ao estrato do servidor mais alto numerado com o qual ele se sincroniza. Uma pesquisa da rede NTP de Nelson Minar, de 1999, indicou que havia (nessa época) 300 servidores no estrato 1; 20.000 servidores no 2; e mais de 80.000 servidores no estrato 3.[6]

Clientes podem acessar uma série de padrões de tempo de referência, como aqueles fornecidos pelo WWV e GPS. Uma lista autorizada dos servidores de tempo da Internet dos Estados Unidos mantida pela National Institute of Standards and Technology pode ser localizada em

www.boulder.nist.gov/timefreq/service/time-servers.html

---

5. Esse daemon também era conhecido como **xntpd** nas primeiras encarnações.
6. Acesse www.media.mit.edu/~nelson/research/ntp-survey99

A maioria dos ISPs mantém seu próprio conjunto de servidores de tempo, que devem ser mais próximos em termos de rede de seus clientes downstream (e se NTP funcionar de modo correto, eles devem ser igualmente precisos).

ntpd implementa tanto o lado cliente como o lado servidor do protocolo NTP. Ele lê **/etc/ntp.conf** na inicialização. No arquivo de configuração você pode especificar acesso, redes de cliente, servidores de tempo, clientes de multicast, configuração geral e autenticação; mas não se assuste — é muito autoexplicativo.

O Debian e o Ubuntu não parecem incluir **ntpd** por padrão, mas ele está disponível por **apt-get**. Você também pode obter o software atual a partir de ntp.isc.org.

Você também pode utilizar o utilitário rápido e sujo, **ntpdate**, para configurar o relógio do sistema a partir de um servidor NTP. Essa é uma solução menos desejável que **ntpd**, porque pode fazer o fluxo de tempo parecer descontínuo. É muito prejudicial configurar o relógio de volta repentinamente, uma vez que os programas às vezes assumem que o tempo é uma função que aumenta monotonicamente.

O **ntpd** utiliza a chamada de sistema **adjtimex** mais suave para o ajuste do relógio do sistema e impede saltos grandes para frente ou para trás. **adjtimex** induz a velocidade do clock do sistema de modo que gradualmente cai em alinhamento correto. Quando o horário do sistema concordar com o horário objetivo atual, o ganho é cancelado e o relógio anda normalmente.

29.13 EXERCÍCIOS

★ E29.1 Utilizando **ps**, determine quais daemons estão sendo executados em seu sistema. Também determine que daemons estão disponíveis para execução por **inetd.conf** ou **xinetd.conf**. Combine as listas e descreva o que cada daemon faz, onde ele é iniciado, se múltiplas cópias podem ser executadas (ou realmente executam) ao mesmo tempo e qualquer outro atributo que você puder coletar.

★ E29.2 No laboratório, instale e configure o daemon de tempo de rede, **ntpd**.

a) Como você sabe se o seu sistema está com o horário correto?

b) Utilizando o comando **date**, configure manualmente seu tempo de sistema com uma lentidão de 15 segundos. Quanto tempo leva (ou levará) para o horário se tornar correto?

c) Configure manualmente o horário de seu sistema um mês adiantado. Como **ntpd** responde a essa situação?

(Exige acesso de root.)

★ ★ E29.3 No laboratório, use uma ferramenta como **netstat** para determinar quais portas estão em um estado de "escuta" em sua máquina.

a) Como você pode reconciliar as informações de **netstat** com aquilo que é encontrado em **inetd.conf** ou em **xinetd.conf**? Se houver uma discrepância, o que mais está ocorrendo?

b) Instale a ferramenta **nmap** em uma máquina diferente. Execute uma varredura de portas com o seu sistema como alvo para verificar o que você aprendeu no item a. Que informações adicionais (se realmente houver alguma) você obteria de **nmap** que não fossem óbvias de **netstat**? (Consulte a página 475 para informações adicionais sobre **nmap**.)

30 Gerenciamento, diretivas e políticas

Você pode ter a equipe mais capacitada de administradores possível, mas se sua gerência técnica for inadequada, você e seus usuários irão odiar a vida. Neste capítulo discutiremos os aspectos não-técnicos de dirigir de uma bem-sucedida empresa de suporte à tecnologia da informação (TI), juntamente com alguns detalhes técnicos (projeto da infra-estrutura, sistemas de registro de problemas etc.) que ajudam a sustentar o objetivo gerencial do sistema de administração.

A maioria dos tópicos e idéias apresentadas neste capítulo não é específica a um ambiente particular. Eles se aplicam tanto a um administrador de sistemas de meio período como a um grande grupo de profissionais de tempo integral responsável por uma importante instalação de TI. Como uma boa alimentação, esses tópicos e idéias são bons independentemente do que você está tentado alcançar. (Para instalações muito grandes com centenas de funcionários TI, também descreveremos brevemente a Information Technology Interface Library, uma arquitetura de gerenciamento de TI focada no processo que é adequada principalmente para grandes instalações. Consulte a página 657.

Além das dicas de gerenciamento, também incluímos seções sobre a documentação (página 638), sistemas TTS (trouble ticket systems) (página 640), recuperação após desastre (página 643), desenvolvimento de diretivas (página 646), questões jurídicas (página 650), patentes de softwares (página 655), esforços de padronização (página 656), cultura Linux (página 658), organizações profissionais relacionadas ao UNIX/Linux, conferências e oportunidades de treinamento para administradores de sistemas (página 660).

30.1 FAÇA TODO MUNDO FELIZ

A administração de sistemas é uma indústria de serviços, e tanto pessoas como computadores são os destinatários desses serviços. Alguns administradores com uma tendência mais técnica acham que os usuários dos seus sistemas não são nada além de aborrecimentos que interferem na atividade real. Esse é um ponto de vista inútil e impreciso; nosso trabalho é fornecer aos usuários a infra-estrutura de TI necessária para eles realizarem seus trabalhos eficiente e confiavelmente.

Todos nós já estivemos, em algum momento de nossas carreiras, do outro lado da mesa. Portanto, coloque-se no lugar dos seus clientes: como foi essa experiência e o que você mudaria?

Os usuários ficam felizes quando:

- Computadores estão configurados e em funcionamento e eles conseguem efetuar o login
- Arquivos de dados permanecem intactos
- Softwares aplicativos estão instalados e funcionando como deveriam
- Ajuda informativa e amigável está disponível quando necessária

Os usuários querem tudo isso 24 horas por dia, 7 dias por semana. Preferivelmente de forma gratuita. Os usuários odeiam

- Tempo de inatividade, agendado ou não
- Atualizações que introduzem alterações repentinas e incompatíveis
- Mensagens incompreensíveis do sistema ou de administradores
- Longas explicações do por quê as coisas não estão funcionando

Quando algo não funciona, os usuários querem saber quando isso será corrigido. É isso aí. Para eles não importa qual ou por que um disco rígido ou gerador quebrou; deixe essas informações para seus relatórios administrativos.

Da perspectiva do usuário, nenhuma notícia é uma boa notícia. O sistema funciona ou não, e se não funcionar, não importa por quê. Nossos clientes ficam felizes quando eles nem mesmo percebem nossa existência! Triste, mas é verdade.

Os usuários são nossos principais clientes, mas é igualmente importante manter nossos funcionários felizes. Bons administradores são difíceis de encontrar e suas necessidades devem ser levadas em consideração ao projetar os sistemas administrativos da sua instalação.

Administradores de sistema e outros funcionários técnicos ficam felizes quando:

- Seus computadores e sistemas de suporte estão ativos e em execução

- Eles têm os recursos necessários para que possam realizar seus trabalhos
- Eles têm as melhores e as mais recentes ferramentas de software e de hardware
- Eles podem trabalhar sem interrupções constantes
- Eles conseguem ser criativos sem o chefe interferir e tentar gerenciar pequenos detalhes
- Suas horas trabalhadas e nível de estresse estão dentro de limites sensatos

Funcionários técnicos precisam mais do que apenas um cheque no final do mês para que continuem empenhados. Eles precisam sentir que têm algum controle criativo sobre suas atividades e quando são reconhecidos pelos seus companheiros, chefes e usuários.

Os requisitos para clientes e funcionários TI felizes têm alguns fatores em comum. Mas algumas coisas parecem ser antagônicas ou até mesmo estar em conflito direto. O chefe precisa assegurar que todas essas diferentes expectativas sejam compatíveis e atingíveis.

Em algumas empresas, o chefe atua como uma interface com os clientes a fim de proteger as principais atividades da equipe técnica contra interrupções, deixando-a livre para que possa fazer o trabalho "real". Nossa experiência é a de que esse arranjo é normalmente uma idéia ruim. Ele afasta a equipe técnica das necessidades dos clientes e freqüentemente resulta em promessas feitas pelo chefe que a equipe não pode cumprir.

30.2 COMPONENTES DE UMA EMPRESA DE TI FUNCIONAL

À medida que a empresa de suporte de um administrador de sistemas começa a crescer, torna-se óbvio que nem todo mundo no grupo pode ou deve saber tudo sobre a infra-estrutura inteira. Também não é prático (na realidade, impossível) que uma única pessoa tome todas as decisões diárias. Com o crescimento, as subdivisões organizacionais a seguir tendem a evoluir:

- *Gerência*: define a estratégia geral e coordena a empresa
- *Administração*: discute as necessidades dos clientes, negocia contratos, envia faturas, faz pedidos de compras, paga os fornecedores, lida com os processos administrativos da contratação e demissão de funcionários
- *Desenvolvimento*: projeta, implementa e testa novos produtos e serviços antes de eles serem implantados para os clientes
- *Produção*: fornece serviços diretos a clientes, tanto gerenciamento do sistema como suporte e treinamento do usuário final — subdividimos essa área em *operações* (lidar com máquinas) e *suporte* (lidar com pessoas)

Essas divisões funcionais não necessariamente são representadas por indivíduos. Mesmo em uma pequena operação com um único administrador de sistemas, as quatro áreas de tarefas são simplesmente diferentes aspectos do trabalho que o administrador de sistemas deve realizar.

O tamanho relativo de cada divisão reflete o tipo de empresa que o grupo administrativo atende. Uma universidade de pesquisa pode ser uma diferente combinação de uma empresa de desenvolvimento de softwares e de uma fábrica. Nossa experiência em uma universidade de pesquisa sugere que o gerenciamento seja aproximadamente 5% do trabalho, a administração seja aproximadamente 10%, o desenvolvimento seja 10% a 20 % e a produção seja responsável pelo restante (65%-75%).

Sugerimos que, em empresas de pequeno e médio porte, todos os funcionários trabalhem pelo menos meio período na divisão de suporte. Essa convenção expõe em primeira mão os problemas, frustrações e desejos dos clientes à equipe técnica. O feedback dos clientes pode então ser utilizado para aprimorar o ambiente atual ou projetar sistemas e serviços futuros. Como efeito colateral, ela também torna alguns funcionários mais experientes e acessíveis aos clientes. (Já ficou preso no nível um do suporte técnico?)

Vinte por cento pode parecer alto para o desenvolvimento, mas sem ele suas soluções não atenderão às exigências à medida que aumentar o número e a complexidade dos sistemas suportados. Seus conceitos originais sobre gerenciamento se estagnarão e se tornarão tão ultrapassados que não haverá esperanças de repará-los.

O número ideal de funcionários que uma pessoa deve gerenciar diretamente parece estar entre 5 e 10. Além desse número, pense em ampliar a hierarquia.

Discutiremos todas as cinco divisões de tarefas nas seções a seguir.

30.3 O PAPEL DO GERENCIAMENTO

O objetivo geral do gerente é facilitar o trabalho dos funcionários técnicos, que são os especialistas no que diz respeito à atividade "real" da empresa. Várias tarefas e responsabilidades caem sobre os ombros do gerente:

- Liderar o grupo, gerar idéias e fornecer os recursos necessários
- Contratar, demitir, avaliar a equipe e desenvolver habilidades

- Atribuir tarefas aos funcionários e monitorar o progresso
- Orientar a gerência
- Tratar problemas: conflitos na equipe, usuários imaturos, hardware antigo etc.
- Atuar como uma "autoridade superior" com a qual os usuários podem superar gradativamente os problemas
- Supervisionar o desenvolvimento de uma infra-estrutura escalonável
- Planejar-se contra desastres e emergências
- Extrair a documentação da mente de administradores de sistemas excêntricos

Talvez pareça que a tarefa de fazer uma interface com os clientes está ausente da lista. Entretanto, acreditamos que esse papel é realmente mais bem desempenhado pelos membros da equipe técnica. Os gerentes normalmente não têm o conhecimento técnico para avaliar a dificuldade e a viabilidade dos requisitos dos clientes. Provavelmente deve haver menos surpresas nos dois lados da mesa se a equipe técnica que faz o trabalho real tiver conhecimento dos produtos finais e das agendas prometidas aos clientes.

Discutimos a seguir algumas dessas funções administrativas em mais detalhes. Algumas delas extrapolam os limites das áreas multifuncionais e estão em outros locais. Por exemplo, o desenvolvimento de uma infra-estrutura com capacidade de expansão é um projeto em conjunto das funções de gerenciamento e desenvolvimento; consulte a página 630 para nossos comentários. De maneira semelhante, a recuperação após desastre é uma questão administrativa e também uma operacional; isso será discutido a partir da página 643. A documentação é tão importante e tão controversa que há uma seção própria que começa na página 638.

Liderança

A liderança é difícil de descrever. Mas quando ausente ou mal executada, sua falta torna-se imediatamente aparente. De certa maneira, a liderança é a "administração de sistemas" das empresas; ela estabelece a direção, assegura que os componentes trabalhem juntos e mantém todo o sistema em execução com o menor número possível de mensagens de erro.

Infelizmente, as habilidades técnicas que tornam uma pessoa um excelente administrador de sistemas não se traduzem necessariamente em liderança, o que exige um conjunto de habilidades mais centrado nas pessoas. A arte de comandar pessoas é muito mais difícil do que a de dominar o Perl.

Para novos gerentes com uma forte formação técnica, pode ser particularmente difícil se concentrar no trabalho de gerenciamento e evitar a tentação de fazer trabalho de engenharia. É mais confortável e mais divertido mergulhar na solução de um problema técnico do que ter uma longa e necessária conversa com um membro "difícil" da equipe. Mas o que é mais valioso para a empresa?

Uma verificação simples (e talvez surpreendente) do seu nível de liderança é a seguinte. Crie uma lista das tarefas nas quais sua empresa está trabalhando. Utilize uma cor para marcar as áreas em que você é o timoneiro e uma cor diferente para marcar as áreas em que você está remando ou puxando o barco. Qual cor domina o quadro?

Contratação, demissão e gerenciamento de pessoal

O gerenciamento de pessoal pode ser particularmente desafiador. Como parte da sua função administrativa, você negocia tanto com o lado técnico como com o lado pessoal dos funcionários. Seguindo o estereótipo, administradores de sistemas tecnicamente brilhantes muitas vezes não sabem se comunicar e às vezes tendem a conviver melhor com máquinas do que com pessoas. Como gerente desses administradores, você precisa fazer com que eles consigam um equilíbrio entre tecnicidade e comunicação.

Crescimento técnico é relativamente fácil de promover e de quantificar, mas o crescimento pessoal é igualmente importante. A seguir há algumas perguntas importantes a fazer durante a avaliação da interface entre o usuário e o funcionário:

- O comportamento dessa pessoa é adequado ao nosso ambiente de trabalho?
- Como essa pessoa interage com autoridades, clientes e fornecedores?
- Essa pessoa tem um bom relacionamento com os outros membros da equipe?
- Essa pessoa tem habilidades de liderança que devem ser desenvolvidas?
- Como essa pessoa responde a críticas e desavenças técnicas?
- Essa pessoa trabalha ativamente para ampliar seu conhecimento?
- Como são as habilidades de comunicação dessa pessoa?
- Essa pessoa pode planejar, implementar e demonstrar o projeto de um cliente?

Contratação

É importante fazer essas avaliações para novas contratações e também para os funcionários existentes. As qualidades pessoais dos candidatos são freqüentemente menosprezadas ou subestimadas. Não siga atalhos nessa área — você certamente se lamentará mais tarde!

Uma entrevista pessoal pode responder a algumas dessas perguntas. Uma conversa telefônica com referências em geral lhe dirá muito mais. Ouça atentamente, muitas pessoas não gostam de dizer coisas ruins a respeito de um antigo empregado ou companheiro de trabalho, de forma que eles encontram maneiras inteligentes de lhe dizer (caso você esteja ouvindo atentamente) que um potencial funcionário tem problemas. Suspeite se o candidato não incluir empregadores recentes como referências.

Há duas formas de montar uma equipe de administradores de sistemas:

- Contratar gente experiente
- Desenvolver sua própria equipe

Em geral, o pessoal experiente engrena mais rapidamente, mas você quer sempre que eles desaprendam certas coisas. Para fazer seus trabalhos, eles precisarão de acesso root. Mas você ainda não os conhece e não vai querer se arriscar colocando imediatamente os dados da empresa em suas mãos. Dividi-las pode requerer um pouco de espionagem e auditoria à medida que sua confiança aumenta.

Requer bastante tempo e empenho treinar um administrador de sistemas, e as redes de produção não são um terreno ideal para treinamento. Mas com a pessoa certa (inteligente, interessada, curiosa, cuidadosa etc.), o resultado final costuma ser melhor.

Algumas das qualidades de um bom administrador de sistemas são contraditórias. Um administrador de sistemas tem de ser impetuoso o bastante para tentar soluções inovadoras quando estiver preso em um problema, mas também tem de ser cauteloso o suficiente para não tentar nada realmente destrutivo. Tanto as habilidades interpessoais como as habilidades de resolução de problemas são importantes, embora pareçam estar em eixos ortogonais entre os muitos administradores de sistema que conhecemos. Um dos nossos revisores sugeriu que um "administrador de sistemas sociável" fosse um oximoro.

Desenvolvemos duas ferramentas de avaliação para candidatos experientes. Costumávamos chamá-las de "testes", mas descobrimos que algumas instituições (como universidades estaduais nos Estados Unidos) não têm permissão de testar os candidatos. Não testamos mais; avaliamos e estimamos.

A primeira avaliação escrita solicita que os candidatos avaliem suas experiências e conhecimentos em vários sistemas e tarefas de rede. A escala de familiaridade varia de 0 a 4:

- Nunca ouvi falar a respeito (0)
- Ouvi falar a respeito, mas nunca fiz (1)
- Já o fiz, poderia fazê-lo sob supervisão (2)
- Poderia fazê-lo sem supervisão (3)
- Conheço isso bem, poderia ensinar alguém a fazê-lo (4)

No meio das perguntas há diversos alarmes. Por exemplo, na seção de hardware logo depois de uma pergunta sobre conectores RJ-45 segue-se uma sobre "conectores MX" (MX refere-se a um registro do servidor de correio no DNS, não a uma rede ou conector serial). Essas "pegadinhas" permitem medir o fator "B" (de "besteira") nas respostas de um candidato. Um 3 nos conectores MX seria suspeito. Na entrevista, depois da avaliação escrita, você poderia inocentemente perguntar: "Então, para que *você* utiliza conectores MX?".

A segunda avaliação é destinada a uma entrevista por telefone. As perguntas são feitas para extrair respostas rápidas de candidatos que realmente entendem do seu ofício. Damos +1 para uma resposta correta, 0 para um "Não sei" e -1 para besteira óbvia ou para quem digitar **man xxx** escondido. Você também poderia dar um +1 para uma resposta razoável do tipo "Não sei, mas sei como poderia descobrir". Muitas vezes, saber onde procurar algo é tão importante quanto entender os detalhes de cabeça.

Esses dois esquemas têm sido uma boa métrica para nós. A porcentagem das "pegadinhas" que utilizamos foi determinada pelo pessoal de RH; uma ou duas perguntas não são suficientes para desqualificar ninguém.

Demissão

Se cometer um engano ao contratar alguém, demita-o logo. Você pode até perder alguns potenciais talentos, mas manter pessoas que ainda não são capazes de realizar bem o seu trabalho alienará os demais membros da equipe que terão de assumir o que essas pessoas não fazem ou fazem mal. Seus clientes também perceberão que o "mais ou menos" não funciona e começam a exigir um administrador de sistemas específico para seus trabalhos. Você não quer que seus clientes interfiram na gerência dos seus negócios diários, quer?

Em muitas empresas é bastante difícil demitir alguém, especialmente após o período de experiência ter terminado. Certifique-se de que as avaliações iniciais foram feitas de maneira séria. Mais tarde, talvez você tenha de coletar dados revelando a incompetência desse funcionário, dar avisos prévios, definir objetivos de desempenho e assim por diante. Em situações extremas, a única maneira de demitir um funcionário incompetente, mas astuto, é eliminar seu trabalho.

Teste completo e garantia de controle de qualidade

O gerente determina o tom do que uma tarefa finalizada significa. Um administrador de sistemas inexperiente normalmente pensará que um problema foi resolvido, apenas para receber vários outros relatos de problemas conforme a tarefa é lenta, completa e corretamente feita. Esse cenário pode ocorrer porque o usuário que primeiro informou o problema não o descreveu claramente, ou talvez porque o usuário sugeriu uma solução imprópria (ou incorreta) que o administrador de sistemas aceitou sem fazer um diagnóstico independente. Da mesma forma, isso pode ocorrer porque o administrador de sistemas não testou a solução cuidadosamente.

Alguns equívocos comuns incluem

- Páginas man e documentação não instaladas para um software novo
- Software não instalado em todos os pontos devidos
- Software que se revela como de propriedade do administrador de sistemas ou instalado com permissões incorretas

Testar é chato, mas um administrador de sistemas atarefado pode cortar a produtividade pela metade se pular essa fase. Cada relatório de problemas custa tempo e esforços, tanto para usuários como para o administrador de sistemas.[1] O trabalho só está concluído quando todos os problemas operacionais surgem, são solucionados e o cliente fica satisfeito.

Muitas vezes um usuário informa que "X não funciona na máquina Y" e o administrador de sistemas examina a máquina Y e tenta o comando X e ela funciona bem. A resposta para o relato retorna: "funcionou comigo", com um pouco de ironia nas entrelinhas.

Se o administrador de sistemas realmente tentasse o comando da maneira descrita e enviada pelo usuário (por exemplo, executando **sudo su - username** em frente do comando), o problema talvez fosse identificado na primeira tentativa. O argumento "-" para **su** faz com que o shell resultante use o ambiente do usuário ao qual **su** está sendo aplicado. Portanto, você pode realmente reproduzir a situação que foi relatada como não estando funcionando.

Às vezes é útil esquematizar seu teste. Por exemplo, se você suspeita que o problema P é causado pelo agente X, remova o agente X e verifique se o problema P desaparece. Então reintroduza X e certifique-se de que P retorna. Por fim, remova X e verifique se P é solucionado. Essa seqüência fornece duas chances de você perceber se P e X não estão de fato relacionados.

Os usuários podem ficar irritados quando um problema não é solucionado completamente na primeira tentativa. Tente definir as expectativas deles apropriadamente. Muitas vezes vale a pena ter o usuário que relatou o problema ao seu lado para tentar solucioná-lo, especialmente se o problema for referente a um pacote de software com o qual você não está familiarizado. Você obterá informações adicionais e será menos provável que o usuário o encare como um adversário.

Gerenciando sem se intrometer

Como um gerente tecnicamente competente, você estará constantemente tentado a aconselhar os funcionários como eles devem fazer seus trabalhos. Mas, se fizer isso, pode estar negando a chance de eles crescerem e se tornarem completamente responsáveis pelos seus trabalhos. É ainda pior se você não só aconselhar como também decidir a maneira como as tarefas devem ser feitas. Esse gerenciamento dos pormenores resulta em uma equipe frustrada incapaz de utilizar perícia e criatividade próprias.

Se você realmente souber tudo melhor que a sua equipe, há algo de errado com sua empresa (você é responsável por um campo muito estreito) ou você tem a atividade errada ou a equipe errada. Ou talvez você simplesmente seja muito controlador.

Premie bom comportamento com descanso remunerado

As universidades norte-americanas têm licenças, normalmente remuneradas, conhecidas como sabáticas — um ano ou seis meses durante o qual não há atividades pedagógicas e quando é possível conduzir pesquisas em outras universidades ou trabalhar em novos projetos de interesse pessoal. A idéia é para que os professores possam ensinar e pesquisar melhor; um determinado departamento precisa interagir com outras pessoas e instituições.

Os administradores de sistema também precisam dessas licenças, mas eles freqüentemente são tidos como sendo tão indispensáveis que não podem ser cedidos por um período de tempo tão longo. Isto é, até eles "surtarem" e conseqüentemente pedirem demissão. Permutar administradores de sistemas com outra empresa durante o verão (ou até por mais tem-

---

1 Às vezes há um custo alto associado a um sistema de registro de problemas ou chamada ao help desk. Por exemplo, o suporte terceirizado atualmente, nos Estados Unidos, é de cerca de US$ 75 para cada chamada ou novo registro.

po) pode servir como uma recompensa aos funcionários mais importantes e libera parte da pressão do serviço ininterrupto da linha de frente. Pode demorar um pouco para que os administradores de sistemas na nova empresa acompanhem o fluxo de trabalho na infra-estrutura local, mas eles trazem idéias e procedimentos a partir dos seus ambientes originais que podem aprimorar as práticas locais dos seus administradores de sistemas.

Se uma permuta cruzar fronteiras internacionais, talvez seja necessário criativamente pensar em uma forma de financiamento para evitar a burocracia necessária para vistos de trabalho oficiais. Fizemos algumas permutas em que os administradores de sistemas foram pagos pelas instituições originais. Em um caso, eles permutaram apartamentos, carros, animais de estimação, plantas etc. — tudo, exceto namoradas!

Atribuindo e monitorando tarefas

Um dos aspectos mais fundamentais da supervisão de um projeto é assegurar que cada tarefa tenha um proprietário claramente definido.

Os perigos congênitos a evitar são:

- Tarefas malsucedidas porque todo mundo pensa que uma outra pessoa é responsável por elas
- Desperdício de recursos pela duplicação de esforços quando várias pessoas ou grupos trabalham no mesmo problema sem coordenação

O trabalho pode ser compartilhado, mas, na nossa experiência, a responsabilidade é menos receptiva à difusão; cada tarefa deve ter claramente definida para uma única pessoa responsável. Essa pessoa não precisa ser o supervisor ou o gerente — apenas alguém disposto a agir como um coordenador ou gerente do projeto.

Um efeito colateral importante dessa abordagem é que fica implicitamente claro quem implementou o ou quem fez quais alterações. Essa transparência torna-se importante se quiser descobrir por que algo foi feito de certa maneira ou por que algo repentinamente está funcionando diferentemente ou não mais funciona.

Ser "responsável" por uma tarefa não necessariamente significa ser um "bode expiatório" se surgirem problemas. Se sua empresa definir uma tarefa dessa maneira, você descobrirá que o número de proprietários de projetos disponíveis cai rapidamente. O objetivo é simplesmente remover ambigüidades sobre quem deve resolver um dado problema ou uma questão qualquer.

Do ponto de vista do cliente, um bom sistema de atribuições é um que direciona os problemas a um funcionário experiente que possa solucionar esses problemas rápida e completamente. Mas, da perspectiva gerencial, ocasionalmente as atribuições precisam ser desafiadoras e de responsabilidade de quem recebeu a atribuição, assim a equipe continua a crescer e a aprender no curso dos seus trabalhos. Seu trabalho é equilibrar a necessidade de administrar as capacidades dos funcionários e a necessidade de manter os funcionários atentos, tudo enquanto mantém tanto os clientes como os funcionários felizes.

Para tarefas administrativas rotineiras, a chave para atribuições de tarefas é um sistema de registro de problemas. Esses sistemas asseguram que relatórios de problemas sejam respondidos e sejam eficientemente encaminhados à pessoa que pode melhor abordá-los. Os sistemas de registro de problemas têm um tópico próprio; consulte a página 640 para uma discussão mais detalhada.

Tarefas maiores podem ser qualquer coisa e até mesmo incluir projetos de engenharia de software com todos os recursos. Essas tarefas talvez exijam o uso de ferramentas formais de gerenciamento de projeto e princípios de engenharia de software. Não descreveremos essas ferramentas aqui; contudo, elas são importantes e não devem ser ignoradas.

Às vezes, os administradores de sistemas sabem que uma tarefa particular precisa ser feita, mas eles não a realizam porque a tarefa é desagradável. Um funcionário que percebe uma tarefa não-atribuída, negligenciada ou impopular provavelmente irá receber essa tarefa como uma atribuição. Essa situação cria um conflito de interesse porque motiva os funcionários a permanecer silenciosos sobre essas situações. Não deixe isso acontecer na sua instalação.

Gerenciando a hierarquia mais alta

Para efetivamente desonerar seus deveres de gerenciamento (particularmente aqueles relacionados à liderança), você precisa de respeito e suporte da sua própria gerência. Você precisa da capacidade de definir a estrutura e os funcionários do seu grupo, incluindo a autoridade da decisão sobre contratações e demissões. Você precisa controlar as atribuições de tarefas, incluindo a autoridade de decidir quando os objetivos foram alcançados e quando os funcionários podem ser reatribuídos. Por fim, você precisa ser responsável por representar seu grupo tanto dentro da empresa maior como para o mundo como um todo.

A gerência mais alta freqüentemente não tem idéia do que os administradores de sistema fazem. Utilize seu sistema de registro de problemas para fornecer essas informações; isso pode ajudar quando seu chefe tentar contratar novos funcionários ou comprar novos equipamentos.

É sensato manter bons registros mesmo na ausência de um objetivo em particular. Os gerentes, especialmente os gerentes não-técnicos, normalmente estão bem longe da realidade em suas estimativas da dificuldade de uma tarefa ou da quantidade de tempo que levará para completá-la. Essa inexatidão é especialmente notável para tarefas de solução de problemas.

Tente estabelecer expectativas reais. Se você não tiver muita experiência em planejar seu trabalho, duplique ou triplique suas estimativas de tempo para tarefas grandes ou cruciais. Se uma atualização for feita em dois dias em vez de três, a maioria dos usuários irá lhe agradecer em vez de xingá-lo como eles normalmente o fariam caso sua estimativa tivesse sido de um dia.

Os termos de licenciamento de softwares muitas vezes não correspondem com o uso real por parte de uma empresa do software licenciado; a gerência mais alta tende a fechar os olhos para isso. Consulte a página 654 para sugestões sobre como tratar essas discrepâncias entre a diretiva e prática.

É sempre difícil obter o suporte por parte da gerência a diretivas de segurança mais rígidas. Segurança rígida invariavelmente significa inconveniência para os usuários, e os usuários normalmente o superam em número e poder de reclamação. Melhor segurança pode reduzir a produtividade dos usuários; antes de implementar uma proposta de alteração de segurança, faça uma análise de risco a fim de assegurar que a gerência e os usuários entendem o motivo da sugestão.

Certifique-se de que qualquer alteração na segurança que afeta os usuários (por exemplo, converter a partir de senhas em chaves RSA/DSA para logins remotos) seja anunciada bem antecipadamente e que esteja bem documentada e bem suportada no momento em que ocorrer a alteração. A documentação deve ser facilmente entendida e fornecer receitas para lidar com o novo sistema. Reserve horas extras para o staff quando mudar para o novo sistema de modo que possa lidar com os usuários assustados por não terem lido seus e-mails.

Soluções de conflitos

Geralmente várias tarefas sob responsabilidade do gerente dizem respeito a um bom relacionamento com as pessoas (em geral clientes ou funcionários) em situações desagradáveis. Primeiro examinaremos a abordagem geral e então discutiremos o caso especial de como lidar com clientes "rebeldes".

Os conflitos no mundo da administração de sistemas freqüentemente ocorrem entre administradores de sistemas e seus clientes, colegas ou fornecedores. Por exemplo, um cliente não está feliz com os serviços prestados, um fornecedor não entregou os materiais prometidos no prazo, um colega de trabalho não fez o que você esperava ou um departamento de engenharia insiste que ele precisa de controle sobre as configurações do sistema operacional instaladas nos seus desktops.

Mediação

A maioria das pessoas não gosta de falar sobre conflitos nem mesmo admitir que eles existem. Quando as emoções afloram, isso geralmente ocorre porque os conflitos só foram solucionados tarde demais, depois de uma situação insatisfatória se estender por muito tempo. Durante essa fase de preparação, as parte envolvidas têm a oportunidade de resolver vários ressentimentos e refletir sobre os motivos de aborrecimento um do outro.

Uma reunião face a face com a presença de mediador neutro pode às vezes diminuir a tensão dessa situação. Tente restringir a sessão a um único tópico e limitar o tempo alocado a meia hora. Essas medidas reduzem a chance de a reunião se degenerar em uma interminável sessão de reclamação.

Uma vez que o problema foi definido e os desejos de cada parte foram expostos, tente alcançar um acordo sobre como impedir uma recorrência do problema. Depois de a solução ser implementada, reexamine a situação a fim de certificar-se de que a solução é eficaz e prática em longo prazo.

Usuários e departamentos rebeldes

O processo de introduzir sistemas bem gerenciados costuma causar conflitos. Usuários com tendência técnica (e, algumas vezes, departamentos inteiros) podem achar que a administração de sistema centralizado não é capaz de atender adequadamente às suas necessidades de configuração ou de autonomia no controle dos computadores por eles usados.

Seu primeiro impulso pode ser o de tentar forçar tais usuários "rebeldes" a aceitar configurações padrão para minimizar o custo e o tempo necessários para suportá-los. Entretanto, tal abordagem tirânica normalmente acaba criando insatisfação tanto por parte dos usuários quanto dos administradores do sistema. Tenha em mente que os anseios dos usuários rebeldes são em geral perfeitamente legítimos e que é o papel dos administradores de sistemas suportá-los ou, pelo menos, de abster-se de aumentar a insatisfação dos usuários.

A solução mais desejável para esse problema é identificar as razões subjacentes da relutância por parte do usuário rebelde em aceitar os sistemas gerenciados. Em muitos casos, você pode atender às necessidades desses usuários e resolver as diferenças.

A alternativa à estratégia de integração é a de negociar suporte por autonomia. Permita que usuários ou grupos rebeldes façam o que eles bem entendam com o entendimento explícito de que eles também devem assumir a responsabilidade de manter os sistemas personalizados funcionando. Instale um firewall para proteger os sistemas que você controla contra quaisquer invasões ou vírus que talvez se originem na rede dos usuários rebeldes. E não "limpe a sujeira" do usuário rebelde.

Não deixe de fazer todos os residentes na rede do usuário rebelde assinarem um documento das diretivas que especificam algumas regras de segurança. Por exemplo, se os sistemas desses usuários interferirem no restante da empresa, sua conexão de rede poderá ser desativada até o problema ser corrigido e não mais se chocar com a rede de produção.

Administração de sistemas criativa é necessário para lidar com o número crescente de laptops utilizados no trabalho. Você deve descobrir maneiras de fornecer serviço a esses dispositivos (não-confiáveis) sem arriscar a integridade dos seus sistemas. Uma rede separada talvez seja uma boa idéia.

Todas as instalações têm seus usuários "de ponta" que são obcecados em adquirir os materiais mais recentes assim que eles são lançados. Esses usuários estão preparados para conviver com a inconveniência das versões beta e pré-releases instáveis contanto que estejam com a última versão do software. Tente vê-las como recursos úteis e não como pessoas problemáticas. Eles são candidatos ideais para testar um novo software e freqüentemente estão dispostos a gerar relatórios de bugs para que você possa corrigir os problemas.

30.4 O PAPEL DA ADMINISTRAÇÃO

Nesse contexto, "administração" inclui todas as tarefas que não contribuem diretamente com a produção da empresa, como contabilidade, recursos humanos, trabalho burocrático, vendas, compras e outros processos logísticos. Ela também está bem separada da "administração" na "administração de sistemas".

Dependendo da empresa, as tarefas administrativas podem ser de sua total responsabilidade, realizadas pela matriz ou em algum ponto intermediário. Mas, seja como for, todas precisam ser feitas.

Em grupos pequenos e autônomos, essas tarefas tendem a ser distribuídas entre os vários membros da equipe. Mas é provável que esse não seja um uso eficiente dos seus especialistas de TI altamente qualificados e difíceis de encontrar. Além disso, muitas pessoas odeiam trabalhos administrativos. Requisitos legais e organizacionais impõem regras estritas que nem sempre são bem aceitas por profissionais criativos. Uma tarefa importante para o chefe é certificar-se de que as tarefas administrativas são atribuídas aos cargos mais altos, realizadas por um pessoal dedicado, terceirizado ou, pelo menos, distribuídas igualmente entre a equipe do administrador de sistemas.

Discutimos apenas algumas tarefas administrativas comuns aqui porque a maioria delas é ditada pela empresa matriz e não é negociável.

Vendas

Se pensar em assumir um projeto para um novo cliente ou um cliente existente, você precisará do equivalente a uma equipe técnica de vendas que entenda as potenciais necessidades dos clientes, conheça os serviços e produtos que seu grupo pode fornecer e esteja ciente da capacidade do grupo. Nada é pior do que fazer promessas que não podem ser cumpridas. Se o chefe participar das negociações, ele deverá receber de antemão um adequado relatório da equipe técnica. Melhor ainda, deixe a equipe técnica fazer a negociação enquanto o chefe observa.

Compras

Em muitas instalações, a equipe de administração de sistemas e a equipe de compras são totalmente separadas. É ruim se a equipe de compras toma decisões sem consultar os administradores de sistemas. Pior ainda se as decisões sobre as compras forem tomadas pela alta gerência jogando golfe ou assistindo a um jogo de futebol em cadeiras cativas compradas pelo fornecedor.

Uma história típica: o vice-presidente do departamento de TI de uma empresa de armazenamento de fitas pensou que sabia qual era o hardware de armazenamento de disco necessário de um grande projeto de servidor. A caminho de uma partida de golfe em um domingo, ele assinou um contrato com a IBM sem consultar sua equipe de administração de sistemas. Os conjuntos de discos da IBM deveriam ser conectados a um servidor da Sun e o vendedor da IBM jurou que isso funcionaria bem. No final, descobrimos que os drivers de dispositivo consistiam em código de qualidade alfa que mal funcionava. O servidor de missão crítica da empresa ficou mais parado do que funcionando nos três meses seguintes enquanto a IBM trabalhava nos drivers. Prazos de entrega do projeto crítico não foram cumpridos e o departamento de TI foi tido como imbecil pelo restante da empresa. O desastre do armazenamento também causou um impacto sobre a principal atividade da empresa — ela passou de uma das maiores no seu mercado para uma das menores porque estava atrasada em disponibilizar produtos cruciais.

Os administradores de sistemas podem fornecer boas informações sobre compatibilidades no ambiente local, sobre a competência dos fornecedores (especialmente revendedores independentes) e sobre a confiabilidade de certos tipos de equipamentos. A confiabilidade das informações é especialmente importante no mundo dos PCs, onde o preço é tão competitivo que a qualidade costuma ser totalmente ignorada.

Os administradores de sistemas precisam acompanhar os pedidos de um novo hardware para que possam determinar como integrá-lo à infra-estrutura atual e também prever quais projetos e recursos são necessários para suportá-lo.

Observe que não é trabalho dos administradores de sistema decidir *se* um sistema pode ser suportado. Você pode fazer recomendações aos seus clientes, mas se eles precisarem de um equipamento específico para que possam realizar suas atividades, é seu trabalho colocar esse equipamento em funcionamento. Talvez seja necessário contratar novos funcionários ou deixar outros sistemas de lado. Forneça ao seu chefe as escolhas e deixe que os superiores responsáveis "pelos pepinos" decidam o que eles preferem fazer.

A participação de um administrador de sistema na especificação dos sistemas sendo comprados é especialmente valiosa em empresas que, por padrão, devem comprar pelos menores preços (por exemplo, instituições governamentais e universidades estaduais). A maioria dos sistemas de compras permite definir critérios de avaliação. Não deixe de incluir cláusulas de escape como "deve ser compatível com o ambiente existente" ou "deve ser capaz de executar bem o pacote XYZ".

O impacto e o custo incremental de uma parte adicional de hardware (ou software, algumas vezes,) não são constantes. Ela é a 60ª dessa arquitetura ou a primeira deste tipo? Ela possui espaço em disco local suficiente para os sistemas de arquivos? Ela possui memória suficiente para executar os pesados aplicativos de hoje em dia? Há uma porta de rede reserva para conexão? Trata-se de um sistema operacional completamente novo?

Perguntas como essas tendem a enfatizar uma questão mais fundamental: você fica estagnado e compra equipamento do fornecedor atual ou você tenta a novidade mais incrementada de uma empresa que poderá ter estrondoso sucesso ou, quem sabe, vir a falir após um ano? A natureza de sua empresa deve responder a essa pergunta. Não é um simples sim ou não; normalmente é necessário fazer uma comparação complexa entre o mais recente e arrojado equipamento e as máquinas com as quais você já está acostumado e compreende. Essas são decisões de negócio e devem ser resolvidas de maneira estruturada e imparcial.

Se lhe for permitido negociar com fornecedores (oficialmente ou de alguma outra maneira) em geral você pode ter um desempenho muito melhor do que seu próprio departamento de compras. Não fique constrangido de fazer cotações de preços com outros fornecedores de equipamentos equivalentes, e seja otimista em relação ao custo real das compras esperadas para o ano seguinte. Afinal de contas, a equipe de vendas está otimista quanto ao valor do seu produto.

Comprar no final de um trimestre ou um pouco antes ou depois da mudança na linha de produtos de um fornecedor pode resultar em alguns bons descontos. Os vendedores freqüentemente precisam aumentar a quantidade de vendas trimestrais ou reduzir o estoque dos modelos que estão em vias de se tornarem defasados.

Na nossa experiência, é difícil encontrar fornecedores com uma equipe tecnicamente qualificada cujo suporte e consultoria estão disponíveis para você. Se encontrar um, mantenha um bom relacionamento.

Depois que os itens foram comprados, a pessoa que especificou o pedido original deve revisar o pedido final, a lista de pacotes e talvez até mesmo a fatura do fornecedor a fim de confirmar se o equipamento correto foi entregue e faturado. Na maioria das empresas, os itens também devem ser inseridos em um sistema local de controle de estoque de acordo com a diretiva da empresa.

Contabilidade

Manter suas finanças em ordem é uma outra tarefa administrativa importante. Mas essa tarefa é tão específica à instalação que não podemos abrangê-la em detalhes. É imperativo que você conheça quais são os requisitos legais e organizacionais e que mantenha seus registros em dia. Auditorias são trabalhosas.

Pessoal

Já discutimos as questões da contratação e demissão (ver a página 623), mas integrar um novo funcionário à sua infraestrutura exige mais do que simplesmente publicar uma vaga de trabalho. Você deve conhecer e honrar as regras da sua empresa com relação a anúncios de vagas, períodos de experiência, revisões etc. Uma outra série de tarefas abrange a mecânica do ambiente de trabalho que define a mesa, a escrivaninha, o computador, as chaves, as contas, o acesso **sudo** e, assim por diante, para um novo funcionário.

Talvez mais importante, quando um funcionário anuncia sua intenção de deixar a empresa, você deve desfazer todos esses processos. Nos Estados Unidos, é comum um aviso prévio de duas semanas antes da saída. Algumas instalações (por exemplo, empresas de segurança) se abstêm do período de duas semanas e dispensam o funcionário, revogando imediatamente todo acesso físico e de rede. Esse talvez seja um bom momento para uma rápida averiguação interna para ver se o funcionário que está saindo não colocou alguma porta dos fundos nos servidores. Alguns pacotes de softwares comerciais (por exemplo, o produto de diretório da Novell) proclamam a capacidade de eliminar ex-funcionários instantaneamente.

A equipe de administração de sistema normalmente não é consultada quando uma empresa planeja demissões em massa, mas deveria ser. O dia da demissão chega e a gerência angustiadamente solicita que seja feito o backup das máquinas dos ex-funcionários, que o acesso seja revogado etc. O hardware desaparece por causa de funcionários descontentes e planejamento ruim.

Marketing

A administração de sistema é um negócio engraçado. Se você fizer bem seu trabalho, os usuários não se darão conta do seu ambiente de computação transparente e ninguém notará o que você faz. Mas no mundo de vírus, spam, aplicativos pesados e total dependência da Internet de hoje em dia, o pessoal de TI é uma parte indispensável da empresa.

Clientes felizes são seu melhor esquema de marketing. Entretanto, há outras maneiras de ganhar visibilidade dentro da sua empresa e dentro da comunidade mais ampla. Com base em nossa longa experiência, sugerimos os métodos a seguir como particularmente efetivos.

- Organize reuniões em que os usuários possam expressar suas preocupações e fazer perguntas sobre a infra-estrutura computacional. Você poderia preparar essa reunião analisando as solicitações de suporte dos usuários e abri-la com uma breve apresentação dos tópicos mais problemáticos que você identificou. Forneça comes e bebes para garantir uma reunião agradável.
- Deixe bastante tempo para perguntas e certifique-se de que o pessoal experiente está disponível para respondê-las. Não tente, porém, escapar de perguntas inesperadas. Se você não souber a resposta imediatamente, é melhor admitir e rever isso mais tarde.
- Projete uma série de seminários direcionada aos seus pares na comunidade de administração de sistemas ou aos usuários finais dentro da sua empresa. Agende reuniões a cada dois ou três meses e publique os tópicos a ser apresentados com antecedência .[2]
- Participe de conferências sobre a administração de sistemas, profira palestras ou escreva artigos sobre as ferramentas que você desenvolve. Essas apresentações não só fornecem um feedback dos seus pares, mas também mostram aos seus clientes (e seu chefe) a qualidade do seu trabalho.

Embora essas técnicas de marketing estejam listadas na seção sobre administração, toda a equipe precisa estar claramente envolvida no esforço de marketing.

A administração de sistemas versa, em última instância, sobre como lidar com as pessoas e suas necessidades. Relacionamentos pessoais são igualmente importantes em qualquer negócio. Converse com seus clientes e parceiros e reserve tempo para discussões e trocas pessoais.

Se oferecer suporte a múltiplos clientes, considere encarregar uma pessoa específica a cada cliente para atuar como um gerente de contas. Esse canal de comunicação deve assumir a responsabilidade pela felicidade geral do cliente e estar em contato regular com os usuários finais do cliente. Encaminhe notícias e informações sobre as alterações no ambiente de computação por meio desse canal a fim de criar oportunidades adicionais de contato.

Tarefas administrativas diversas

O grupo de administração fica em dificuldades com uma grande quantidade de pequenos itens logísticos:

- Manutenção do estoque
- Aquisição de móveis e outros equipamentos para o escritório
- Manutenção da máquina de café e organização do estoque de café, açúcar e leite. O estoque de café não deve vir do orçamento do grupo para garantir que cada funcionário tenha seu melhor desempenho com a dose certa de cafeína. Algumas empresas também fornecem aperitivos
- Regar plantas e flores
- Fazer uma interface com o departamento de manutenção do edifício (ou empresas terceirizadas) para organizar a limpeza, manutenção do sistema de refrigeração etc.
- Organização das chaves ou outro meio de controle de acesso

30.5 O PAPEL DO DESENVOLVIMENTO

Grandes instalações precisam de uma robusta infra-estrutura computacional com capacidade de expansão para que os administradores de sistemas não gastem todo o tempo lutando com um sistema inflexível, mal projetado e que demanda muita mão-de-obra. Em termos práticos, isso significa que você precisa de um software personalizado, de uma estrutura arquitetônica e de uma equipe que seja formalmente responsável por projetar e manter tudo isso.

---

2. Você pode convidar parceiros da comunidade de administradores de sistemas para que se apresentem na sua série de seminários. Certifique-se de que suas apresentações tenham a mesma alta qualidade dos seus interlocutores internos.

À medida que sua instalação cresce e aumenta o número de computadores, você ficará cada vez mais empolgado ou começará a elaborar maneiras de otimizar e sistematizar o processo de administração. A ferramenta de otimização mais comum (e mais poderosa) é a automação. Ela reduz o trabalho manual e assegura que as tarefas sejam realizadas consistentemente.

Infelizmente, essa automação com freqüência ocorre de uma maneira não-estruturada, com a triste conseqüência de que a instalação acumula gradativamente uma grande e aleatória série de correções rápidas e de fragmentos de script. Esse corpo do software acabará, por si só, se tornando um pesadelo de manutenção não-documentado.

A controvérsia irá persistir quanto ao número de máquinas que uma instalação precisa ter antes que faça sentido começar a pensar em termos de um plano global. Nossa percepção é realmente de que isso não é tanto uma questão do número de profissionais envolvidos, mas de profissionalismo e satisfação no trabalho. Uma vida de severos testes e dramas não é divertida e invariavelmente está correlacionada a uma qualidade ruim do serviço. Os requisitos também variam bastante; o que é apropriado para uma instalação que suporta usuários do mercado financeiro será inadequado a um laboratório para estudantes em uma universidade e vice-versa. Suas soluções devem ser apropriadas às necessidades dos seus usuários.

O inconveniente termo "arquitetura de infra-estrutura" é às vezes utilizado para descrever o trabalho conceitual de alto nível do sistema de administração, em oposição às tarefas rotineiras. A parte "infra-estrutura" é uma referência a quando o departamento de TI torna-se um serviço de utilidade pública. Como ocorre com o sistema de telefonia ou o sistema de abastecimento de água, ele deve funcionar bem. Quando a infra-estrutura não funciona, deve ser fácil localizar e substituir o componente deficiente. A parte "arquitetura" significa que uma infra-estrutura é projetada para um propósito específico e é personalizada para se ajustar às necessidades dos seus clientes.

Nesta seção delineamos alguns recursos de uma arquitetura de infra-estrutura funcional. Para uma discussão mais detalhada desse tópico, confira as referências no final do capítulo. Indicações úteis sobre a implementação podem ser encontradas no Capítulo 11, *Gerenciamento de software e configuração*.

Princípios arquitetônicos

As seções a seguir apresentam uma seleção das "melhores práticas" arquitetônicas a considerar no planejamento do projeto de TI da sua instalação. Esses princípios são particularmente importantes quando a configuração a qual você fornecerá suporte for nova ou incomum, uma vez que essas situações podem ser difíceis de medir em relação às correspondentes no mundo real. Processos bem projetados incorporam ou fomentam a adesão a esses princípios.

Torne os processos reproduzíveis

A administração de sistemas não é uma arte performática. O que quer que seja feito, deve ser feito consistente e repetidamente. Em geral, isso significa que o nível mais baixo das alterações deve ser realizado por scripts ou programas de configuração em vez de pelos administradores de sistema. Variações na configuração devem ser capturadas nos arquivos de configuração do seu software administrativo.

Por exemplo, um script que configura uma nova máquina não deve fazer perguntas sobre números IP e pacotes a ser instalados. Em vez disso, ele deve verificar o diretório de configuração de um sistema para determinar o que fazer. Esse script pode apresentar essas informações para confirmação, mas as escolhas devem ser preestabelecidas. Quanto menos interação de usuário, menor a chance de erro humano.

Mas vamos deixar claro: não estamos descrevendo uma instalação em que autoridades administrativas de alto nível tomam decisões sobre a diretiva a ser executada por pessoas incompetentes. A reprodutibilidade é igualmente relevante se você for o único administrador na sua instalação. Geralmente não é uma boa idéia tomar decisões de configuração que não possam ser acompanhadas e que não deixam nenhuma trilha de auditoria. Se algo precisa ser alterado, mude as informações da configuração central e as propague daí.

Deixe uma trilha de migalhas de pão

Quem fez isso e por quê? Se houver problemas com seu sistema, a correção será muito mais rápida quando for possível voltar ao último estado funcional ou pelo menos descobrir o que mudou desde então. Além do "o quê" também é importante saber "quem" e "por quê". Falar com a pessoa que implementou uma alteração complexa freqüentemente leva a um insight importante. Talvez você seja capaz de rapidamente desfazer a alteração, mas às vezes a alteração foi feita por uma boa razão e a desfazer somente irá piorar as coisas.

Os sistemas de controle de revisão fornecem uma maneira útil de monitorar as alterações; eles são discutidos em detalhes a partir da página 174. Esses sistemas fornecem tanto um registro do histórico dos dados reais ao longo do tempo como informações sobre qual administrador de sistemas realizou a alteração. Se usada corretamente, cada modificação é acompanhada por um comentário que explica o raciocínio por atrás dela. Ferramentas automatizadas podem fazer uma verificação nos arquivos de configuração que elas modificam e se auto-identificarem no comentário. Dessa maneira, é fácil identificar um script que não funciona e reverter as alterações que ele realizou.

Um outro recurso útil é algum tipo de diário de e-mail. Isso pode ser um registro permanente das mensagens enviadas a uma lista real de contatos administrativos na sua instalação ou uma pequena lista de correio que só existe para registrar as

atualizações do diário. O diário fornece um lugar bem conhecido para informações sobre alterações na configuração a serem registradas e o fato de que é implementada via e-mail significa que as atualizações não são trabalhosas para os administradores de sistemas. A desvantagem de um diário de e-mail é que não há como estruturar as informações além de cronologicamente.

Reconhecer a importância da documentação

De fato, a documentação é tão importante para uma infra-estrutura escalonável que criamos uma seção especial para ela, iniciando na página 638.

Personalizar e escrever código

Utilizar as ferramentas existentes é uma virtude e você deve usá-las sempre que possível. Mas nenhuma instalação no mundo é exatamente igual a sua e é certo que sua empresa terá alguns requisitos únicos. Uma infra-estrutura de TI que preenche precisamente as necessidades da empresa dá uma vantagem competitiva e melhora a produtividade de todo mundo.

Com a excelente e ampla possibilidade de criação de scripts das ferramentas de código-fonte aberto, o Linux é a base ideal para uma infra-estrutura bem ajustada. Na nossa visão, um grupo de administração de sistemas sem uma função de desenvolvimento de softwares é um deficiente.

Manter o sistema limpo

O gerenciamento de sistemas não é apenas sobre instalar, adicionar e configurar; ele também é sobre saber o que manter, o que descartar e o que reciclar. Chamamos esse conceito "gerenciamento sustentável". É fantástico conseguir adicionar um novo computador ao seu ambiente em 5 minutos e é excelente conseguir criar uma nova conta de usuário em 10 segundos. Mas se você olhar para frente, é igualmente importante ser capaz de encontrar e de remover contas e computadores antigos de uma maneira organizada. A sustentabilidade no gerenciamento de sistemas significa que você tem as ferramentas e conceitos necessários para administrar sua operação por um longo período de tempo de uma forma organizada.

Anatomia de um sistema de gerenciamento

Seu sistema de gerenciamento deve conter estes elementos fundamentais:

- **Configuração automatizada das novas máquinas.** Isso não é só a instalação do sistema operacional; ela também inclui software adicionais e a configuração local necessária para permitir que uma máquina entre em uso de produção. É inevitável que sua instalação precise suportar mais de um tipo de configuração, portanto você deve incluir múltiplos tipos de máquinas nos seus planos logo no início. *Instalação básica do Linux*, iniciando na página 159, discute os vários sistemas que ajudam com a configuração.
- **Correção sistemática e atualização das máquinas existentes.** Ao identificar um problema na sua configuração, você precisa de uma maneira padronizada e fácil de implementar as atualizações para todas as máquinas afetadas. Observe que como os computadores não permanecem ligados todo o tempo (mesmo que devessem), seu esquema de atualização deve tratar corretamente as máquinas que não estão on-line quando a atualização é iniciada. Você pode verificar as atualizações em tempo de inicialização ou em tempo de atualização de acordo com uma agenda regular; ver a página 182 para informações adicionais.
- **Um sistema de monitoração.** Seus usuários não precisam ligar para você para informá-lo de que o servidor está parado. Isso não só é antiprofissional, como também você não terá nenhuma idéia de quanto tempo o sistema estava parado. É provável que a primeira pessoa a contatá-lo não seja a primeira a passar por problemas. Você precisa de algum tipo de sistema de monitoração que dispare um alarme assim que os problemas forem evidentes. Mas alarmes são complicados. Se houver um número excessivamente grande, os administradores de sistemas começam a ignorá-los; se houver um excessivamente pequeno, problemas importantes passam despercebidos.
- **Um sistema de comunicação.** Não se esqueça de manter contato com as necessidades dos seus usuários; fornecer suporte ao cliente é o objetivo final de tudo que você faz como um administrador de sistema. Um sistema de rastreamento de solicitações é uma necessidade (consulte a página 640). Uma localização central onde os usuários possam obter informações de contato e o status do sistema (em geral na Web) também é útil.

A caixa de ferramentas do administrador de sistema

Bons administradores de sistemas são preguiçosos (inteligentes) e muitas vezes tentam automatizar seus trabalhos. A empresa bem-sucedida de um administrador de sistemas escreve scripts para preencher as lacunas entre as ferramentas

herdadas dos fornecedores e as ferramentas cujas cópias foram feitas a partir da rede, portanto certifique-se de que esses scripts estejam disponíveis onde quer que sejam necessários.

A maioria dos locais tem um script **adduser** próprio e provavelmente também um script **rmuser**. Se for o único administrador de sistemas, você provavelmente irá salvar essas pequenas jóias no seu próprio diretório **~/bin** e viver feliz para sempre depois disso. Se fizer parte de uma equipe de duas ou mais pessoas, a vida torna-se mais complexa. Mensurações das substituições temporárias — por exemplo, copiar diferentes ferramentas ou modificar caminhos de pesquisa para que eles apontem para os diretórios **~/bin** de outras pessoas — ajudarão em curto prazo. Entretanto, você por fim precisará encontrar algum tipo de solução sustentável.

Nossa solução para esse problema é a "caixa de ferramenta do administrador de sistemas". Essa caixa de ferramenta contém todos os softwares da administração de sistemas desenvolvidos localmente e reside em uma árvore de diretórios especial (por exemplo, **/usr/satools**). Acima de tudo, temos um documento de diretivas que descreve como escrever ferramentas para essa caixa de ferramenta. Um dos administradores de sistemas é o "proprietário" da caixa de ferramenta e toma todas as decisões sobre a diretiva que não foram abrangidas pelo documento de diretivas.

Ao criar uma diretiva como essa, é importante certificar-se de que cada elemento da diretiva tenha uma razão clara de estar aí. Apenas se as pessoas entenderem a motivação por trás das regras é que elas serão capazes de seguir o espírito da diretiva e não apenas suas regras literais.

Algumas entradas de diretivas de exemplo estão listadas a seguir.

- Ferramentas publicadas estão disponíveis em **/usr/satools/bin**. As informações sobre a configuração são armazenadas em **/usr/satools/etc**. Os arquivos de log e quaisquer outros arquivos de dados que as ferramentas criam enquanto em execução são colocados em **/usr/satools/var**. Arquivos de dados estáticos estão em **/usr/satools/share**. Razão: padronização.
- As ferramentas devem ser escritas em Perl ou em Python. Razão: padronização; facilita a reutilização de código entre os administradores.
- Cada ferramenta deve ter sua própria página man. Razão: se uma ferramenta não estiver documentada, somente seu autor entenderá como ela funciona.
- Ao desenvolver uma ferramenta, escreva primeiro a documentação e faça com que o "mestre em ferramentas" a revise antes da codificação. Razão: captura redundância e questões de conformidade com a diretiva antes de qualquer trabalho ser concluído.
- Cada ferramenta deve aceitar uma opção **--no-action** que faz com que ela mostre o que ela faria sem realmente fazer nada. Se essa opção não puder ser implementada, a ferramenta precisa pelo menos falhar com uma mensagem apropriada. Razão: ajuda a evitar catástrofes, especialmente com ferramentas mais drásticas como **rmuser** ou **mvhomedir**.
- Todas as ferramentas devem estar sob controle de revisão em **svn://server/satools** (um repositório Subversion; consulte a página 178). Razão: várias pessoas podem trabalhar nas ferramentas e um controle de revisão transparente simplifica o gerenciamento original e o tratamento de bugs.

Um documento de diretivas prático poderia ter algumas páginas.

Talvez você queira escrever uma pequena ferramenta de gerenciamento da caixa de ferramentas que instale novas ferramentas do repositório Subversion para sua árvore **/usr/satools**. Para controle de qualidade, a ferramenta de gerenciamento poderia também tentar fazer algum teste de conformidade com a diretiva. Por exemplo, ela pode verificar se cada ferramenta instala a documentação.

Princípios da engenharia de software

Como scripts administrativos nem sempre parecem software "real", você poderia pensar que eles não precisam do grau de previsão e formalismo encontrado nos típicos projetos de engenharia de software. Isso pode ou não ser verdadeiro. À medida que seu sistema cresce em complexidade e importância dentro da empresa, você precisará empregar esses princípios de engenharia de software, por mais difíceis que eles sejam.

O livro clássico de Frederick P. Brooks Jr. sobre engenharia de software, *The Mythical Man-Month: Essays on Software Engineering*, é uma leitura fácil e uma boa referência. É interessante ver que os problemas de engenharia de software que ele discute a partir das décadas de 1960 e 1970 continuam atuais. Eis uma lista de verificação dos princípios e armadilhas do projeto de software (sem uma ordem específica):

- Projetos devem ter um escopo claramente definido e um cronograma bem definido para conclusão.
- Responsabilidades devem ser claramente definidas. O administrador que trabalha no projeto, o gerente do administrador de sistemas e o cliente final, todos devem concordar quanto aos objetivos, escopo e riscos potenciais.
- Documentação preliminar deve ser escrita antes de a codificação começar.
- As interfaces entre os vários componentes do software devem ser bem definidas e documentadas antecipadamente.

- Trechos importantes do código devem ser prototipados e testados primeiro. É importante obter feedback das pessoas que utilizarão o software o mais cedo possível na etapa de desenvolvimento.
- Informações sobre o status devem ser comunicadas honestamente. Ninguém gosta de surpresas, e notícias ruins sempre aparecem no final.
- Incorpore uma análise *post-mortem* a cada projeto. Aprenda a partir de erros e sucessos.

O gerenciamento ótimo dos projetos de softwares sempre é debatido acaloradamente. Ao longo da última década, os métodos "ágeis" que salientam prototipagem rápida, implantação prévia, adição incremental de funcionalidades e a eliminação de overheads receberam atenção significativa dos desenvolvedores. A importância desses métodos continua longe de apresentar um acordo universal, mas muitas das idéias parecem senso comum e a abordagem leve é bem apropriada a projetos menores encontrados no domínio da administração de sistemas. Leia mais em en.wikipedia.org/wiki/Agile_methods..

30.6 O PAPEL DAS OPERAÇÕES

A divisão das operações é responsável pela instalação e manutenção da infra-estrutura de TI. Junto com a divisão de suporte, ela é um componente da função de "produção" descrita na página 622. Como regra geral, as operações lidam com computadores e fios e o suporte lida com as pessoas.

As operações são apenas os serviços existentes; todas as "coisas novas" devem ser projetadas, implementadas e testadas pela divisão de desenvolvimento antes de elas serem incorporadas em um ambiente de produção.

As operações focalizam a criação de um ambiente estável e seguro para os clientes. A disponibilidade e a confiabilidade são suas principais preocupações. A equipe de operações não deve realizar experimentos ou fazer correções rápidas ou melhorais em uma sexta-feira à tarde. A chance de falhas (e de ninguém, exceto os clientes, perceber os problemas no fim de semana) é simplesmente muito alta.

Minimizando tempo de inatividade

Muitas pessoas dependem da infra-estrutura computacional que fornecemos. Talvez o departamento de uma universidade possa viver temporariamente sem seu site, mas uma empresa de compras on-line como a Amazon.com não pode. Algumas pessoas não perceberão se seu servidor de impressão está parado, mas um aluno ou um professor com um prazo final de entrega apertado para enviar um artigo científico ou uma proposta ficará de fato muito infeliz. Mesmo em um ambiente universitário, não ter acesso ao e-mail normalmente torna todo o mundo ranzinza. Servidores de arquivos centrais são outra fonte potencial de desastres.[3]

Em algumas instalações você precisará fornecer um serviço para emergências. Em um ambiente comercial, isso talvez signifique suporte 24/7 no local pela equipe experiente de administradores de sistemas. Mas mesmo em um ambiente não comercial como um departamento universitário, é ruim quando ocorre grave no sistema em um fim de semana prolongado.

Mesmo se não tiver o orçamento para fornecer explicitamente suporte 24/7, você deverá estar preparado para tirar proveito de quaisquer administradores disponíveis tarde da noite ou nos finais de semanas. Um pager ou outro sistema de notificação freqüentemente pode proporcionar suporte "razoavelmente bom" para emergências. Certifique-se de que os usuários possam acessar esse suporte de alguma maneira fácil e bem-conhecida; por exemplo, um alias de correio eletrônico chamado pager de suporte que retransmite ao pager móvel.

Dependências de documentos

Para fazer reivindicações precisas quanto à disponibilidade ou o tempo de funcionamento, você não só deve conhecer suas próprias forças e fraquezas (incluindo a confiabilidade do hardware que você implanta), mas também as dependências dos sistemas de TI de outros hardwares, softwares e pessoal. Por exemplo:

- Potência: alimentadores e circuitos independentes de energia, proteção contra surtos e curto-circuito, sistemas sobressalentes como geradores e no-breaks, fiação elétrica do edifício, plantas da energia fornecida a partes específicas do equipamento
- Rede: cabeamento do edifício, linhas sobressalentes, serviço de atendimento ao cliente para ISPs, topologia de rede, informações de contato para outros grupos dentro da empresa com uma função de gerenciamento de rede própria
- Hardware: sistemas de alta disponibilidade e procedimentos para utilizá-los, hot/cold standbys (máquinas de backup automático/manual), peças sobressalentes, contratos de manutenção de hardware

---

3 Se um servidor de arquivos parar de funcionar por um longo período de tempo, os usuários descontentes talvez insistam em usar seus próprios discos rígidos em vez do servidor de arquivos para armazenamento (pelo menos até "surtarem" por causa da falha do disco local, sem nenhum backup).

Nova função ou eliminação do hardware mais antigo

Para manter sua infra-estrutura, você deve comprar máquinas novas, dar uma nova função para as máquinas mais velhas e descartar as mais antigas. Discutimos a aquisição na seção sobre administração a partir da página 628, mas descartar equipamentos antigos sempre parece ser mais uma questão de operações.

Toda vez que o desempenho do hardware melhora, normalmente quando o tamanho ou a complexidade de um software aumenta, essa melhoria cai por terra. Por exemplo, o Windows 95 e 98 sobreviviam bem com 32 MB de memória. O Windows 2000 executava mal a menos que você tivesse pelo menos 128 MB, e um laptop executando o Windows XP agora parece precisar mais de 256 MB apenas para executar o gerenciador de tarefas. Naturalmente, esse inchaço não é apenas um problema do Windows; ambientes desktop Linux atuais também precisam de muita memória.

Como os usuários e a gerência costumam ficar relutantes à atualização de equipamentos obsoletos, você às vezes tem de tomar essa iniciativa. Informações financeiras são a evidência mais persuasiva. Se puder demonstrar no papel que o custo da manutenção de um equipamento antigo excede o custo da substituição, você poderá remover boa parte das objeções intelectuais à atualização. Às vezes também é útil substituir hardware heterogêneo simplesmente para poupar tempo e esforços necessários para manter atualizados todos os diferentes sistemas operacionais e versões de software.

Hardware barato da Intel/PC é a base da arquitetura padrão na área de desktops, especialmente agora que a Apple usa hardware da Intel. O predomínio dos PCs ao longo dos anos fez com que as despesas em computação passassem do hardware para o software e para o suporte.

Você pode facilitar essa transição para os usuários de sistemas antigos mantendo esses sistemas disponíveis. Deixe os sistemas antigos ligados, mas reduza o nível de suporte que seu grupo administrativo fornece. Interrompa a manutenção do hardware nas máquinas antigas e permita que elas parem gradativamente de funcionar até serem desativadas.

Você pode utilizar máquinas mais antigas para serviços dedicados que não requerem muitos recursos de hardware (por exemplo, impressão, DHCP ou servidores de licença). Apenas certifique-se de que você tem um plano para corrigir ou substituir as máquinas ou mover serviços para um outro servidor no caso de problemas.

Universidades freqüentemente recebem doações de componentes de computadores antigos de empresas que querem dedução de impostos. Em geral, a resposta correta é: "Não, obrigado, não precisamos de 2.000 unidades de fita de nove trilhas e racks para armazená-las". Há muitos anos uma universidade em Budapeste recebeu como doação um mainframe IBM. Em vez de dizer não e comprar PCs rápidos, eles dedicaram todo o orçamento para transporte e fiação elétrica no primeiro ano e manutenção nos anos seguintes. Resultado, não valeu a pena. Por outro lado, muitas universidades estabelecem fortes vínculos com a indústria de informática local e obtêm uma série de hardware valioso como doação — são bons modelos do ano passado. Considere o custo total de propriedade e saiba quando dizer: "Não, obrigado".

Se o seu caso for o de uma empresa e tiver máquinas sobrando, considere a possibilidade de doá-las para escolas ou faculdades de sua região; elas normalmente não têm verba suficiente para manter seus laboratórios atualizados e podem aceitar com muito prazer os seus PCs mais antigos.

Computadores obsoletos que não podem ser passados para uma instituição de caridade precisam ser adequadamente reciclados; eles contêm materiais perigosos (por exemplo, chumbo) e devem ser separados do lixo normal. Se sua empresa não tem um departamento de recursos que pode lidar com equipamentos antigos, talvez você tenha de levar esses equipamentos a um centro de reciclagem local para descarte. A taxa de remoção de lixo nos Estados Unidos é de aproximadamente US$ 10 por sistema.

30.7 O TRABALHO DE SUPORTE

A tarefa do departamento de suporte é lidar com seres humanos que usam e dependem dos sistemas de informática que você mantém. Além de oferecer o help desk normal ou o suporte hotline, esse departamento também pode oferecer serviços de consultoria aos clientes ou produzir seminários de treinamento sobre tópicos específicos. Idealmente, esses serviços auxiliares aumentam a auto-suficiência dos clientes e reduzem o número de solicitações de suporte.

Disponibilidade

Bom suporte de TI significa que uma equipe qualificada está disponível para ajudar um cliente sempre que ele precisar.

A maioria dos problemas é menos importante e pode entrar seguramente na fila de serviços. Outros são problemas que interrompem a produção e merecem atenção imediata. Respostas automatizadas de um sistema de rastreamento de solicitações e mensagens telefônicas gravadas informando o horário comercial só causam aborrecimentos. Certifique-se de que os usuários podem ter sempre algum tipo de acesso como um último recurso se a necessidade surgir. Um celular disponível que transfere chamadas entre a equipe do administrador de sistemas fora do horário comercial é normalmente suficiente.

Infelizmente, suporte excelente cria dependências. É fácil os usuários se habituarem a consultar o help desk mesmo quando isso não é necessário. Se reconhecer que alguém está utilizando o sistema de suporte para respostas que poderiam ser obtidas tão facilmente como uma consulta em páginas man ou no Google, comece a responder às perguntas citando a página man ou URL relevante. Essa tática funciona bem para estudantes; ela provavelmente é menos efetiva para executivos.

Escopo do serviço

Você deve definir claramente os serviços que seu grupo de suporte fornecerá; do contrário, as expectativas dos usuários não corresponderão à realidade. Eis algumas questões a serem consideradas:

- Tempo de resposta
- Serviço nos fins de semana e fora do horário
- Chamadas telefônicas feitas a partir de casa (suporte para máquinas domésticas)
- Hardware estranho
- Hardware antigo
- Sistemas operacionais suportados
- Configurações padrão
- Expiração das fitas de backup
- Software de propósito especial
- Tarefas de zelador (limpeza de telas e de teclados)

Em uma instalação Linux, cada usuário tem acesso ao código-fonte e pode facilmente criar um kernel personalizado para sua própria máquina. Você tem de considerar essa possibilidade em sua política corporativa e provavelmente tentar manter uma padronização com algumas poucas configurações de kernel específicas. Caso contrário, seu objetivo de manutenção e de expansibilidade com a empresa encontrará sérios obstáculos. Encoraje seus funcionários criativos e que gostam de fuçar no sistema operacional a sugerir modificações de kernel de que eles precisem para suas atividades. Faça com que eles tomem parte do processo de padronização para que você não tenha de fazer a manutenção de kernels distintos para cada engenheiro.

Além de saber quais serviços são fornecidos, os usuários também têm dê conhecer o esquema de prioridade utilizado para gerenciar a fila de tarefas. Os esquemas de prioridade sempre têm espaço variável, mas tente elaborar um que cubra a maioria das situações com poucas ou nenhuma exceção. Algumas variáveis relacionadas a prioridades são listadas a seguir:

- O nível de serviço para o qual o cliente pagou/fez o contrato
- O número de usuários afetados
- A importância dos usuários afetados (essa é uma difícil, mas em geral não é recomendável irritar a pessoa que assina o cheque de pagamento)
- O barulho feito pelos usuários afetados (rodinhas das cadeiras guinchando)
- A importância do prazo final de entrega (lição de casa tardia *versus* proposta ou contrato de concessão de pesquisa)

Embora todos esses fatores influenciem suas avaliações, recomendamos um conjunto simples de regras juntamente com algum senso comum para lidar com as exceções. Basicamente, utilizamos estas prioridades:

- Muitas pessoas não podem trabalhar
- Uma pessoa não pode trabalhar
- Pedidos de melhorias

Se duas ou mais solicitações tiverem máxima prioridade e não puderem ser atendidas paralelamente, baseamos nossa decisão sobre qual problema atacar primeiro na gravidade das questões (por exemplo, e-mail que não funciona deixa quase todo mundo infeliz, enquanto indisponibilidade temporária de algum serviço Web talvez afete apenas algumas pessoas). As filas nas prioridades mais baixas normalmente são tratadas de uma maneira primeiro a entrar, primeiro a sair.

Talvez você queira calibrar o senso comum da sua equipe de suporte. Normalmente fazemos perguntas como: "O que você fará se o chefe entrar no seu escritório enquanto você estiver trabalhando em um problema de prioridade 1 e ele quiser que você arrume a impressora dele porque ele precisa imprimir alguns slides para uma reunião. O táxi para o aeroporto sai em 15 minutos e, naturalmente, todo mundo no departamento de TI, exceto você, está almoçando". Qualquer pessoa sensata no suporte perceberá (pelo menos depois de pensar melhor) que simplesmente não há razões para argumentar com essa pessoa que já está sob muita pressão. Para se defender, é razoável perguntar se não há problemas em interromper o trabalho no servidor central de arquivos. Se a resposta for sim, é apropriado ignorar tudo e imprimir os slides.

Em seguida perguntamos: "O que você fará quando o chefe o colocar em uma situação semelhante algumas semanas mais tarde?". Mais uma vez não há razões em argumentar, mas dessa vez talvez seja uma boa idéia agendar uma reunião com ele depois de ele retornar a fim de discutir as necessidades de suporte do chefe. Talvez ele precise de uma pessoa específica para seu suporte pessoal e, se ele estiver disposto a pagar por isso, qual é o problema? Só não tente agendar essa reunião se ele ainda estiver preocupado com o horário do vôo.

Os usuários geralmente partem do pressuposto de que seus dados importantes são armazenados em fitas de backup que serão arquivadas para sempre. Mas as fitas de backup não são eternas; a mídia magnética tem tempo de vida finito após o qual a leitura de dados se torna difícil. (Você precisa regravar seus dados periodicamente; é bem possível que em uma mídia mais nova, se quiser mantê-los por muito tempo.) Talvez um dia você precise recuperar arquivos das fitas de backup; nessa hora você *vai* querer que os dados estejam lá e sejam recuperáveis. É melhor trabalhar com as pessoas responsáveis por essas decisões a fim de redigir um acordo por escrito que especifique por quanto tempo os backups devem ser mantidos, se várias cópias devem ser criadas (requeridas? permissíveis? nunca?) e se essas cópias devem ser armazenadas em locais diferentes.

Você deve disponibilizar essas informações aos usuários dos seus sistemas porque isso promove expectativas realistas em relação aos backups. Também notifique os usuários de que eles mesmos devem tomar precauções se perceberem que precisam de melhor proteção para seus dados.

Conjuntos de habilidades

Não há nada mais irritante para um usuário experiente do que uma pessoa no suporte perguntando: "Você conectou o cabo de força?" — enquanto ao mesmo tempo pesquisa freneticamente um banco de dados do serviço de atendimento ao cliente. Por outro lado, é um desperdício de recursos fazer com que seu administrador mais experiente explique a um usuário iniciante como localizar a tecla Delete em algum sistema de processamento de texto.

No nosso ambiente, um número de horas de suporte por semana é atribuído a cada membro da equipe.[4] A equipe de suporte reserva esse tempo no seu planejamento semanal, e, quando um funcionário recebe uma solicitação de suporte, o tempo gasto nessa chamada é descontado desse planejamento. As atribuições são feitas de acordo com as habilidades necessárias para corrigir o problema e de acordo com o tempo remanescente disponível ao suporte semanal de cada pessoa.

Para que esse esquema funcione com sucesso, certifique-se de que sua lista de habilidades está equilibrada. Em longo prazo, todo o mundo na equipe de suporte deve ser capaz de cumprir suas horas alocadas ao suporte. Em geral, um membro da equipe com muitas entradas na lista de habilidades é mais "precioso". Entretanto, não há nada de errado em uma equipe com menos habilidades, contanto que ela tenha trabalho suficiente.

Uma lista de habilidades exata ajuda a verificar se você tem pessoal qualificado suficiente para lidar com férias e doenças. Você pode construir a lista de habilidades à medida que os problemas aparecem e são resolvidos pelos membros da equipe. Inclua a tarefa, o nome do membro da equipe e o nível demonstrado de perícia.

As habilidades devem ser definidas em um nível apropriado de abstração, nem muito específico nem muito geral. A lista de habilidades de exemplo a seguir demonstra o nível apropriado de granularidade:

- Criar usuários, remover usuários, configurar senhas, mudar cotas
- Criar contas CVS ou SVN
- Restaurar arquivos a partir dos backups
- Integrar novos drivers de hardware no RIS (Windows)
- Empacotar um aplicativo Windows no formato MSI
- Criar e instalar pacotes de aplicativos software no Linux
- Analisar os arquivos de log
- Depurar problemas de servidor de correio
- Depurar problemas de impressão
- Depurar problemas gerais de hardware
- Criar entradas de DNS
- Gerenciar licenças de software
- Responder a questões sobre a segurança no Windows
- Responder a questões sobre a segurança no Linux
- Resolver solicitações relacionadas ao Samba
- Configurar o DHCP
- Configurar um servidor LDAP
- Adicionar ou remover sites Web (configurar o Apache)

---

4 Por padrão, atribuímos a cada administrador de sistemas 25% de tempo no suporte. Quando um projeto exige atenção em tempo integral por parte de um administrador de sistemas particular, desviamos do esquema padrão removendo essa pessoa do esquema de atendimento de suporte até o final do projeto.

Gerenciamento do tempo

A administração de sistemas envolve mais alterações no contexto em um dia do que a maioria dos trabalhos tem em um ano, e a equipe de suporte ao usuário precisa conviver com esse caos. Cada administrador precisa de boas habilidades de gerenciamento do tempo. Sem elas, você não conseguirá assumir suas responsabilidades diárias e ficará frustrado e deprimido. (Ou, se já estiver frustrado e deprimido, ficará ainda mais frustrado e deprimido.)

O stress de administradores de sistemas é violento. A maioria dos administradores permanece ativa por apenas alguns anos. Ninguém quer estar constantemente de plantão e ouvir reclamações continuamente. Descobrir maneiras de gerenciar seu tempo eficientemente e manter seus clientes felizes é uma situação em que todos ganham.

No seu livro recém-publicado, *Time Management for System Administrators,* Tom Limoncelli sugere maneiras de evitar essas armadilhas. Uma referência completa é fornecida no final deste capítulo.

30.8 DOCUMENTAÇÃO

Assim como a maioria das pessoas aceita os benefícios dos exercícios físicos e de uma alimentação baseada em verduras e legumes, todo mundo aprecia uma boa documentação e tem uma idéia vaga da sua importância.

Infelizmente, isso não significa necessariamente que elas escreverão ou atualizarão a documentação sem algum tipo de estímulo.

Por que devemos nos preocupar realmente?

- A documentação reduz a probabilidade de um ponto de falha único. É maravilhoso ter ferramentas que implantam estações de trabalho rapidamente e distribuem patches com um único comando, mas essas ferramentas são quase inúteis se não houver uma documentação e se o especialista estiver de férias ou pediu demissão.
- A documentação ajuda a reprodutibilidade. Se práticas e procedimentos não forem armazenados na memória institucional, há poucas probabilidades de eles serem seguidos consistentemente. Quando administradores não conseguem localizar informações sobre como fazer algo, eles têm de improvisar.
- A documentação economiza tempo. Ao escrevê-la parece que você não está economizando tempo, mas depois de passar alguns dias resolvendo um problema que já foi atacado, mas cuja solução foi esquecida, a maioria dos administradores se convence de que o tempo é bem investido.
- Por fim e, acima de tudo, a documentação aprimora a inteligibilidade de um sistema e permite que modificações subseqüentes sejam feitas de maneira consistente com a maneira como o sistema deve funcionar. Quando as modificações são feitas com base em um entendimento parcial, elas freqüentemente não se adaptam à arquitetura. A entropia aumenta ao longo do tempo e até mesmo os administradores que trabalham no sistema chegam a vê-lo como uma coleção desordenada de quebra-galhos. O resultado final muitas vezes é o desejo de jogar tudo fora e começar de novo da estaca zero.

A documentação local serve para muitas coisas. Alguma vez você já teve de entrar em uma sala de máquinas para reinicializar um servidor, só para se deparar com racks e racks de hardware, todos parecidos entre si, todos diferentes e todos sem nenhuma identificação? Ou teve de instalar um componente de hardware com o qual já lidou no passado, mas tudo o que você consegue se lembrar agora a respeito da tarefa é de que ela era difícil de ser feita?

A documentação local deve ser mantida em um lugar bem definido. Dependendo do tamanho da sua operação, esse lugar poderia ser um diretório em um servidor de arquivos montado em todas as suas máquinas ou talvez até mesmo no diretório inicial de uma conta especial de usuário do sistema.

Documentação padronizada

Nossa experiência sugere que a maneira mais fácil e mais efetiva de manter a documentação é padronizar de acordo com documentos leves e curtos. Em vez de escrever um manual de gerenciamento de sistemas para sua empresa, escreva vários documentos de uma página, cada um dos quais abrangendo um único tópico. Inicie com a visão geral e então o divida em partes que contêm informações adicionais. Se em alguma parte você precisar ser mais detalhista, escreva um documento extra de uma página que focaliza os passos que são particularmente difíceis ou complicados.

Essa abordagem tem várias vantagens:

- Seu chefe provavelmente só está interessado na configuração geral do seu ambiente. Isso é tudo que é necessário para responder às perguntas anteriores ou conduzir uma discussão administrativa. Não adicione detalhes demais ou você só fará com que seu chefe fique tentado a interferir neles.
- O mesmo é verdadeiro para clientes.

- Um novo funcionário ou alguém assumindo novas responsabilidades dentro da sua empresa precisa de uma visão geral da infra-estrutura a fim de tornar-se produtivo. Não é útil fornecer a essas pessoas uma grande quantidade de informações.
- É mais eficiente utilizar o documento correto do que navegar por um grande documento.
- É mais fácil manter a documentação atual quando você pode fazer isso atualizando uma única página.

Esse último ponto é particularmente importante. Manter a documentação em dia é um desafio enorme; normalmente, é a primeira coisa a ser descartada quando o tempo é curto. Descobrimos que algumas abordagens específicas mantêm a documentação fluindo.

Primeiro, especifique a expectativa de que a documentação deve ser concisa, relevante e direta. Vá direto ao assunto; o importante é obter as informações. Nada faz a documentação empacar mais rapidamente do que tentar escrever uma dissertação com especificação de mil páginas sobre a teoria de projetos. Solicite uma grande quantidade de documentação e talvez você não obtenha nada.

Segundo, integre a documentação aos processos. Os comentários nos arquivos de configuração estão entre as melhores formas de documentação. Eles sempre estão onde são necessários e praticamente não exigem tempo de manutenção. A maioria dos arquivos de configuração padrão permite comentários e mesmo aqueles que não são particularmente amigáveis a comentários costumam aceitar informações extras. Por exemplo, o conteúdo padrão do campo GECOS do arquivo **passwd** (escritório, número de telefone etc.), em geral, não é a informação mais útil que você poderia colocar aí. Sinta-se à vontade para definir suas próprias convenções no nível global da instalação. (Mas, nesse caso específico, lembre-se de que os usuários podem alterar o conteúdo desse campo, portanto, nem sempre acredite nas informações armazenadas aí.)

Ferramentas construídas localmente podem exigir documentação como parte das suas informações de configuração padrão. Por exemplo, uma ferramenta que configura um novo computador pode exigir informações sobre a localização, status do suporte e informações de cobranças do proprietário do computador mesmo se esses fatos não afetarem diretamente a configuração do software da máquina.

A documentação não cria informações redundantes. Por exemplo, se você mantiver um arquivo de configuração mestre no nível global da instalação que liste as máquinas e seus endereços IP, não deverá haver nenhum outro lugar onde essas informações são atualizadas manualmente. Fazer atualizações em vários locais não só é um desperdício de tempo como também é certeza de que aparecerão inconsistências ao longo do tempo. Quando essas informações são necessárias em outros contextos e arquivos de configuração, escreva um script que obtenha essas informações (ou atualizações) a partir da configuração mestre. Se não puder eliminar completamente as redundâncias, pelo menos deixe claro qual fonte é a autorizada. E escreva ferramentas para capturar inconsistências, talvez executadas regularmente a partir de **cron**.

Identificação de hardware

Alguma documentação é mais apropriada quando anotada em um pedaço de papel ou gravada em um hardware. Por exemplo, procedimentos de emergência para uma completa falha de sistema ou de rede são particularmente úteis se armazenados em uma máquina morta ou inacessível.

Cada computador deve ser identificável sem que ninguém precise ligá-lo ou efetuar o login, pois essas atividades nem sempre serão possíveis. Rotule cada estação de trabalho de uma forma única (nome de host, endereço IP) e coloque um rótulo nela que inclua as informações de contato para o help desk.

Em uma sala de servidor, todos os sistemas e seus dispositivos externos devem ser rotulados com um nome de host (máquinas completamente qualificadas a partir de domínios diferentes são localizadas aí). É útil colocar esses rótulos tanto na parte frontal como na parte de trás das máquinas (especialmente em racks com pouco espaço) de modo que você possa encontrar facilmente o botão de energia da máquina que pretende ligar.

Se seu ambiente incluir diferentes tipos de sistemas, talvez seja útil adicionar informações extras como arquitetura, instruções de inicialização, seqüências especiais de teclas, ponteiros para documentação adicional, o serviço de atendimento do fornecedor ou o número de telefone da pessoa responsável. Registrar seqüências de tecla pode parecer um pouco ridículo, mas os servidores costumam ser conectados a um terminal ou servidor de console mais antigo em vez de a um monitor dedicado.

Certifiquem-se de que seus registros centrais ou dados de inventário contenham uma cópia das informações em todas essas pequenas etiquetas. Será útil se gerenciar suas máquinas por meio de uma conexão TCP/IP com seu servidor de console em vez de trabalhar o dia todo em uma sala de máquinas barulhenta.

Também grave o nome de host com outros dispositivos de hardware relacionados a cada máquina: unidades de disco, modems, impressoras, unidades de fita etc. Se vários subsistemas externos idênticos estiverem anexados ao mesmo host, certifique-se

de que os rótulos são únicos.[5] Na nossa instalação, mesmo os cabos SCSI e Fibre Channel são rotulados (e, obviamente, as várias portas no servidor) de modo que realmente possamos saber que dispositivo está conectado a qual interface.

Se um host for um cidadão importante (por exemplo, um servidor fundamental ou um roteador crucial), inclua a localização do disjuntor. Se for necessário um disquete ou cartão de memória flash para inicialização, aponte para sua localização. Os servidores de arquivos principais devem ter informações sobre nomes de dispositivos de disco, tabelas de partição, pontos de montagem e as localizações de superblocos de backup prontamente disponíveis. Coloque as informações nas próprias unidades de fita ou as armazene em um local bem conhecido, talvez na sala de máquinas.

As unidades de fita devem ser identificadas com os arquivos de dispositivos e comandos necessários para acessá-las. É uma boa idéia listar o tipo de fitas exigido por unidade, o local mais próximo para adquiri-las e até o preço aproximado.

As impressoras devem ser rotuladas com seus nomes, breves instruções de impressão e os hosts dos quais elas dependem. Normalmente as impressoras vêm com interfaces de rede e são membros completos da rede, mas elas ainda dependem de um host Linux para spooling e configuração.

A interligação de redes deve ser documentada cuidadosamente. Rotule todos os cabos, identifique painéis de ligação e tomadas e marque os dispositivos de rede. Sempre facilite as condições para que o técnico responsável pelas ligações consiga manter a documentação atualizada; mantenha lápis e formulários pendurados na parede do gabinete de fiação para facilitar as anotações de que um cabo foi transferido de um dispositivo para outro. Posteriormente, você deve transferir esses dados para armazenamento on-line.

Contudo, há um outro nível de complexidade se seus dispositivos de rede (por exemplo, roteadores e switches) puderem ser reconfigurados pela rede. Embora você agora possa mover máquinas entre sub-redes a partir do seu confortável escritório, a documentação torna-se ainda mais importante. Seja ainda mais cuidadoso, pois você pode bagunçar uma parte muito maior da sua infra-estrutura mais rápida e completamente.

Documentação de usuário

É interessante preparar um documento impresso que pode ser dado a usuários novos. Ele deve documentar personalizações específicas, procedimentos para descrever problemas, os nomes e as localizações de impressoras, seus cronogramas de backup e de desligamento programado e assim por diante. Esse tipo de documento pode lhe poupar uma quantidade enorme de tempo de serviços de usuários e de administradores de sistemas. Você também deve tornar essas informações disponíveis na Web. É mais provável que um documento impresso seja lido pelos usuários novos, porém uma página Web é mais fácil de ser acessada nos momentos em que surgirem dúvidas. Crie ambas e as mantenha regularmente atualizadas. Não há nada mais irritante do que uma documentação on-line ou FAQs ultrapassadas.

Além de documentar seu ambiente computacional local, talvez seja interessante preparar algum material introdutório sobre o Linux. Tal material é essencial em um ambiente universitário no qual a comunidade de usuários é transiente e muitas vezes não conhece Linux. Fornecemos planilhas de uma página impressas que listam os comandos e aplicativos comumente necessários para nossa comunidade de usuários.

30.9 SISTEMAS DE RASTREAMENTO DE SOLICITAÇÕES E DE GERAÇÃO DE RELATÓRIOS DE PROBLEMAS

Se tiver uma grande equipe, você descobrirá que é útil impor alguma estrutura formal à sua fila de trabalhos com um sistema de rastreamento de solicitações. Mesmo se sua empresa de TI for uma instalação com uma única pessoa, esse sistema pode ser bastante útil para rastrear todos os problemas e solicitações pendentes. Os rastreadores de solicitação também são uma excelente fonte de informações históricas sobre o trabalho que sua equipe fez; e eles são uma maneira fácil de monitorar suas listas de tarefas e projetos internos. Gerentes sempre querem saber o que você está fazendo; a maioria dos sistemas pode gerar relatórios que mostram sua carga de trabalho, seus recursos e sua real e exata necessidade de mais pessoal e de um orçamento maior.

Esses sistemas têm vários nomes; os mais comuns são "sistemas de rastreamento de solicitações", "sistemas de registro de problemas" ou "rastreadores de bugs". Esses nomes são para o mesmo tipo de coisa. O site Web linas.org/linux/pm.html lista e categoriza uma variedade de implementações de código-fonte aberto.

Funções comuns dos sistemas de registro de problemas

Um sistema de registro de problemas aceita solicitações por meio de várias interfaces (e-mail, formulários Web e linhas de comando são as mais comuns) e as monitora desde o envio até a solução. Os gerentes podem atribuir registros a grupos

5 Um dos nossos colaboradores recentemente substituiu a bateria de backup do cache de gravação do array RAID errado. Felizmente, esse erro particular só resultou na degradação do desempenho. Mas imagine se você substituísse o disco errado em um array RAID5, não percebesse isso e então um segundo disco falhasse.

de pessoas ou a membros individuais da equipe. A equipe pode consultar o sistema para verificar a fila de registros pendentes e talvez resolver alguns deles. Os usuários podem descobrir o status de uma solicitação e verificar quem está trabalhando nela. Os gerentes podem extrair informações de alto nível como:

- O número de registros abertos
- O tempo médio para fechar um registro
- A produtividade dos administradores de sistemas
- A porcentagem de registros não resolvidos (corrompidos)
- A distribuição da carga de trabalho de acordo com o tempo para a solução

O histórico de solicitações armazenado no sistema de registro torna-se um histórico dos problemas com sua infraestrutura de TI e as soluções para esses problemas. Se esse histórico for facilmente pesquisável, ele torna-se um recurso inestimável para a equipe do administrador de sistemas.

Mensagens de problemas solucionados podem ser enviadas aos administradores de sistemas iniciantes e aos estagiários, inseridas em um sistema de FAQs ou simplesmente registradas em log. Pode ser bem útil os novos membros da equipe verificarem os registros fechados porque esses registros incluem não só informações técnicas como também exemplos do tom e do estilo de comunicação apropriado para uso com os clientes.

Como ocorre com todos os documentos, os dados do histórico do seu sistema de registro de problemas podem ser utilizados potencialmente contra sua empresa em um tribunal. Siga as diretrizes de retenção de documentos especificadas pelo seu departamento jurídico.

A maioria dos sistemas de rastreamento de solicitações confirma automaticamente as novas solicitações e atribui a elas um número de rastreamento que aqueles que as enviam podem utilizar para acompanhar ou investigar o status da solicitação. A mensagem de resposta automatizada deve afirmar claramente que ela é apenas uma confirmação. Ela deve ser seguida imediatamente por uma mensagem proveniente de uma pessoa real que explica o plano para lidar com o problema ou a solicitação.

Aceitação dos usuários aos sistemas de registros de problemas

Receber uma resposta imediata de uma pessoa real é um determinante crucial da satisfação dos clientes, mesmo se essa resposta pessoal não contiver nenhuma informação além da resposta automatizada. Para a maioria dos problemas, é bem mais importante deixar quem enviou a mensagem saber que o registro foi revisado por uma pessoa real do que corrigir o problema imediatamente. Os usuários sabem que os administradores recebem muitas solicitações e estão dispostos a esperar um período de tempo razoável para que possam ser atendidos. Mas eles não querem ser ignorados.

Nosso sistema de registros de problemas utiliza um alias de e-mail chamado "support". Certa vez fomos bombardeados com relatos de problemas que eram incompletos ou incompreensíveis. Criamos um script que fazia ao usuário perguntas específicas como:

- Em que host ocorreu o problema?
- O problema pode ser reproduzido?
- Qual o nível de importância deste problema ser corrigido imediatamente?

A rebelião de usuários se iniciou uma hora depois, e, dentro de um dia, retornamos o sistema ao que era antes. Seu único valor parece ter sido aquele relacionado com o furor em relação ao script; na verdade, muitos usuários leram as perguntas que estávamos fazendo, e a qualidade de nossos relatos de problemas de formato livre aumentou.

Uma outra instalação tratou desse problema enviando uma mensagem que explicava quais informações eram importantes em um relato de problema e deu exemplos de relatos inúteis. Quando um relato inútil era recebido, ele era respondido com uma desculpa ("Desculpe-nos, não temos informações suficientes para …") junto com uma cópia do próprio relato. Os usuários se ligaram rapidinho.

Na nossa experiência, formulários só funcionam para tarefas que são altamente padronizadas, como criação de contas ou solicitações de exclusão. Nesses contextos, seus clientes entendem que as informações solicitadas são necessárias para que você faça seu trabalho. Algumas tarefas podem ser totalmente automatizadas depois que o formulário é preenchido, como configurar uma mensagem de férias para um usuário. Outras, como a criação de contas, podem ser parcialmente automatizáveis, mas exigem autorização ou outro processamento por parte da equipe.

Sistemas de registro de problemas

As Tabelas 30.1 e 30.2 a seguir resumem as características dos vários sistemas de registros de problemas bem conhecidos. A Tabela 30.1 mostra sistemas de código-fonte aberto, e a Tabela 30.2 mostra sistemas comerciais.

Tabela 30.1 Sistemas de registro de problemas de código-fonte aberto

Nome	In[a]	Lang	Back[b]	URL
Mantis	WE	PHP	M	www.mantisbt.org
RT: Request Tracker	W	Perl	M	www.bestpractical.com
Scarab	W	Java	M	scarab.tigris.org
Double Choco Latte	W	PHP	PM	dcl.sourceforge.net
OTRS	WE	Perl	PMOD	www.otrs.org
JitterBug[c]	WE	C	F	www.samba.org/jitterbug
WREQ[c]	WE	Perl	G	math.duke.edu/~yu/wreq

a. Tipos de entrada: W = web, E = e-mail.
b. Back end: M = MySQL, P = PostgreSQL, O = Oracle, D = DB2, F = flat files, G = gdbm.
c. Não mais mantido.

Gostamos bastante do Mantis. Ele foi originalmente desenvolvido para rastrear bugs no software de um videogame. Ele executa no Linux, Solaris, Windows, Mac OS e mesmo no OS/2. É leve, simples, facilmente modificável e personalizável. Ele requer PHP, MySQL e um servidor Web. Mas seu recurso mais importante é uma boa documentação!

Um outro sistema interessante é o OTRS, Open Ticket Request System. O OTRS apresenta interfaces Web tanto para clientes como para administradores de sistemas e também uma interface de e-mail. O OTRS é altamente personalizável (por exemplo, mensagens de saudação configuráveis por fila) e pode até mesmo registrar em log o tempo gasto em um registro. Os pacotes para cada uma das nossas plataformas Linux de exemplo estão disponíveis no site Web do OTRS.

RT: A versão 3 do Request Tracker tem um rico conjunto de recursos, incluindo interfaces Web altamente configuráveis tanto para os usuários como para os administradores de sistemas. Como ele utiliza uma miríade de módulos Perl, pode ser trabalhoso instalá-lo. Sua interface de linha de comando também é básica no melhor dos casos. Alguns sites utilizam uma versão corrigida da versão 1 (consulte ee.ethz.ch/tools) que corrige algumas das deficiências mais irritantes do RT sem adicionar a complexidade das versões mais recentes.

A Tabela 30.2 mostra algumas alternativas comerciais para gerenciamento de solicitações. Como os sites Web das ofertas comerciais são principalmente um modismo de marketing, detalhes como a linguagem de implementação e o back end não são listados.

Tabela 30.2 Sistemas comerciais de registro de problemas

Nome	Escala	URL
Remedy (agora BMC)	Grande	www.bmc.com/remedy
ServiceDesk	Grande	manageengine.adventnet.com
HEAT	Média	www.frontrange.com
Track-It!	Média	www.numarasoftware.com

Algumas ofertas comerciais são tão complexas que precisam de uma ou duas pessoas dedicadas à manutenção, configuração e para mantê-las em execução (Remedy e ServiceDesk). Esses sistemas são apropriados para um site com uma equipe de TI muito grande, mas são inúteis para uma típica equipe de TI, pequena e com excesso de trabalho.

Envio de registros

Em um grupo grande, mesmo um com um impressionante sistema de registros, ainda resta um problema para ser resolvido. Não é eficiente que várias pessoas dividam a atenção entre a tarefa em que elas estão trabalhando no momento e a fila de solicitações, especialmente se as solicitações chegarem por e-mail em uma caixa de correio pessoal. Testamos duas soluções para esse problema.

Nosso primeiro teste foi atribuir turnos de trabalho de meio período das tarefas na fila de problemas aos membros da equipe no nosso grupo de administradores de sistemas. A pessoa encarregada da tarefa tentaria responder ao maior número possível de consultas de entrada durante seu turno. O problema dessa abordagem foi que nem todo o mundo tinha as habilidades para responder a todas as perguntas e corrigir todos os problemas. As respostas às vezes eram inadequadas porque a pessoa encarregada da tarefa era nova e não conhecia os clientes, seus ambientes nem o contrato de suporte específico que esses clientes tinham. O resultado era que um número maior de pessoas experientes tinha de dar atenção aos detalhes e assim não conseguiam se concentrar nos seus próprios trabalhos. No final, a qualidade do serviço era pior e nada era realmente ganho.

Depois dessa experiência, criamos um papel de "despachante" que era alternado mensalmente entre um grupo de administradores seniores. O despachante é responsável pela verificação de novas entradas e tarefas no sistema de registros e por distribuir tarefas para membros específicos da equipe. Se necessário, o despachante entra em contato com os usuários para extrair todas as informações adicionais necessárias para priorizar as solicitações. O despachante utiliza um banco de dados "feito em casa" das habilidades da equipe para decidir quem na equipe de suporte tem o tempo e as habilidades apropriadas. O despachante também garante que as solicitações são resolvidas de maneira oportuna.

30.10 RECUPERAÇÃO APÓS DESASTRE

Sua empresa depende de um ambiente TI funcional. Você não só é responsável pelas operações diárias, mas também deve ter planos para lidar com qualquer eventualidade razoavelmente previsível. A preparação para esses problemas em larga escala influencia tanto sua estratégia geral como a maneira que você define as operações diárias.

Nesta seção analisamos diversos tipos de desastres, os dados que você precisa recuperar graciosamente e os elementos importantes de um plano pós-desastre.

Sugerimos que você crie um catálogo escrito explícito dos potenciais desastres contra os quais você quer se proteger. Nem todos os desastres são idênticos e talvez sejam necessárias várias estratégias diferentes para cobrir todas as possibilidades. Considere as seguintes ameaças:

- Quebras de segurança: antes do ano 2000, cerca de 60% originava-se dentro da própria empresa. Em 2001, o aumento abrupto no número de ataques externos fez com que a porcentagem anterior caísse para algo em torno de 30%. Em 2005, os ataques baseados internamente estavam em torno de 47%.[6]
- Problemas de ambiente: surtos de tensão e falta de energia, falhas no sistema de refrigeração, inundações, furacões, terremotos, meteoros, ataques terroristas ou invasões de alienígenas.
- Erro humano: arquivos e bancos de dados excluídos ou danificados, informações de configuração perdidas
- Panes de hardware espontâneas: servidores que não respondem, discos rígidos "fritos", redes defeituosas.

Pare e pense, quais benefícios nossos usuários obtêm do ambiente de TI? Quais destes são mais importantes? O que coloca em risco esses benefícios? Como podemos nos proteger contra essas ameaças? Obter uma definição correta do problema não é uma tarefa trivial e é crucial que toda sua empresa esteja envolvida, não apenas o grupo de administração de sistema.

Backups e informações off-line

Por razões óbvias, os backups dos dados são uma parte crucial do planejamento contra desastres e uma discutida em mais detalhes no Capítulo 9, *Backups*. Mas criar backups confiáveis fora do site é apenas parte da história. Qual é a rapidez com que você pode acessar esses backups? Como você organizará o hardware substituto, mão-de-obra e dados de backup para substituir um servidor que falhou?

Estamos acostumados a usar a rede para nos comunicarmos e acessarmos documentos. Entretanto, esses recursos poderão ficar indisponíveis ou comprometidos após algum incidente. Guarde todos os contatos e procedimentos relevantes off-line. Saiba onde encontrar as fitas de dump recentes e qual comando **restore** usar sem procurar em **/etc/dumpdates**.

Em todas as situações de desastre, você precisará acessar tanto cópias on-line como off-line de informações essenciais. As cópias on-line devem, se possível, ser mantidas em uma máquina auto-suficiente: uma que tenha um complemento rico de ferramentas, ambientes dos administradores de sistemas chave, que execute seu próprio servidor de nomes, que tenha um arquivo **/etc/hosts** local completo, que não tenha nenhuma dependência de compartilhamento de arquivos, que tenha uma impressora anexada e assim por diante. Não utilize uma máquina velha que não seja útil para nada; a máquina de recuperação após desastre deve ser rápida e deve ter bastante memória e espaço em disco livre para que você possa utilizá-la em restaurações e comparações durante a recuperação. Essa máquina precisa de um ambiente completo de desenvolvimento de modo que ela possa corrigir e recompilar qualquer software comprometido. Ajuda se essa máquina também tiver interfaces para todos os tipos de unidades de disco utilizados na sua instalação (IDE, SATA, SCSI, FC-AL etc.)

Eis uma lista dos dados úteis a serem mantidos na máquina de backup e na forma impressa:

- Esquema do procedimento contra desastres: pessoas que devem ser acionadas, quando acioná-las, o que dizer
- Números de telefone de empresas contratadas e números de clientes
- Números de telefones locais importantes: equipe, polícia, bombeiros, chefe, agência de empregos

---

6 Esses dados são provenientes do FBI e do Computer Security Institute.

- Dados sobre as configurações de hardware e software: versão do sistema operacional, níveis de correção, tabelas de partição, configurações de hardware de PC, IRQs, DMAs e coisas do gênero[7]
- Fitas de backup e programa de agendamento de backups que as gerou
- Mapas das redes
- Números de série de software, dados de licenciamento e senhas
- Informações para contato com o fornecedor para esse disco de emergência de que você precisa imediatamente

Preparando sua equipe contra desastres

Decida com antecedência quem estará encarregado no caso de um incidente catastrófico relacionado com a segurança do sistema. Estabeleça uma cadeia de comando e mantenha fora de computadores ("off-line") os nomes e números de telefones dos superiores. Pode ser que a melhor pessoa para ser encarregada disso seja um administrador de sistemas das trincheiras, não o diretor de TI (que normalmente é uma péssima opção para esse papel).

A pessoa encarregada deve ser alguém que tenha autoridade e poder para tomar decisões difíceis com base em informações mínimas (por exemplo, uma decisão para desconectar um departamento inteiro da rede). A capacidade de tomar essas decisões, comunicá-las de maneira sensata e liderar a equipe em uma crise é provavelmente mais importante do que ter um insight teórico sobre o gerenciamento do sistema e da rede. Guardamos um pequeno cartão laminado com nomes e números de telefones importantes impressos em tipo microscópico. Muito prático — e cabe na carteira de bolso.

Uma suposição importante, mas muitas vezes não dita, feita na maioria dos planos pós-desastre é que a equipe de administradores de sistemas estará disponível para tratar da situação. Infelizmente, as pessoas ficam doentes, têm festas de formatura, saem de férias, mudam de emprego e em momentos de estresse podem até mesmo se tornar hostis. Vale a pena considerar o que você deve fazer caso precise de ajuda emergencial extra. (Não ter administradores de sistemas suficientes à disposição pode, algumas vezes, por si só se constituir em uma emergência caso seus sistemas sejam frágeis ou seus usuários não sejam muito dotados.)

Talvez você precise tentar formar uma espécie de pacto OTAN com uma empresa de consultoria local ou universidade que tenha talentos em administração de sistemas que possam ser compartilhados. Obviamente, você tem de estar disposto a retribuir colocando o seu pessoal à disposição caso eles venham a ter problema. Mais importante ainda, não opere próximo do limite em sua rotina diária. Contrate administradores de sistemas suficientes e não espere que eles trabalhem 12 horas por dia.

Força e sistemas de ar-condicionado

Teste o seu plano de recuperação pós-desastre antes de precisar utilizá-lo efetivamente. Teste seus geradores e no-breaks mensal ou trimestralmente, dependendo do grau de risco que seu gerenciamento está disposto a aceitar. Verifique se tudo com o que você se preocupa está ligado em um no-break, que as baterias do no-break estejam boas e que o mecanismo de falha esteja funcionando. Para testar um no-break individual, basta desligá-lo da tomada. Para certificar-se de que os equipamentos cruciais tenham uma especificação de no-breaks apropriada, talvez você precise usar disjuntores. Conheça as dependências e os pontos de falha do seu sistema de energia.

No-breaks também precisam de manutenção. Essa função provavelmente está fora do escopo dos seus deveres de administrador de sistemas, mas você é responsável por assegurar que ela seja realizada.

A maioria dos problemas de alimentação é de curta duração, mas planeje para poder suportar por duas horas via bateria para ter tempo suficiente para desligar todas as máquinas apropriadamente no caso de uma falta de energia mais prolongada. Alguns no-breaks possuem uma porta serial ou interface Ethernet para você poder iniciar um desligamento adequado de máquinas não cruciais após 5 minutos (tempo configurável) de falta de energia.

Tire proveito das faltas de energia para fazer atualizações rápidas que você já testou, mas ainda não implementou. O sistema está fora do ar de qualquer maneira, de forma que as pessoas já têm uma expectativa de serem incomodadas. Em algumas instalações, 5 minutos extras durante uma falta de energia é mais fácil do que aceitar um tempo de inatividade programado com uma semana de antecedência. Se tiver máquinas antigas que você acredita não estarem mais em uso, deixe-as desligadas até que alguém reclame. Pode ser que passe semanas — ou que jamais aconteça — dessa máquina "ausente" ser percebida.

Consulte a página 534 para informações adicionais sobre questões de ambiente.

Sistemas de refrigeração normalmente possuem um sistema de aviso que pode informá-lo caso a temperatura se torne muito elevada. Ajuste o valor de "muito alto" de forma que, após o sistema de refrigeração lhe enviar um bip, você tenha tempo suficiente antes de as máquinas realmente começarem a fritar; nós usamos 25°C em vez de 32°C, mas moramos nas montanhas a 45 minutos de distância (no verão; tempo

7 Ferramentas de descoberta de rede e de inventário podem compilar boa parte desses dados automaticamente. Duas candidatas a considerar são H-Inventory (sourceforge.net/projects/h-inventory) e LANsurveyor (www.neon.com).

indeterminado no inverno). Mantenha um par de termômetros mecânicos ou operados por bateria na sala de máquinas — falta de energia significa perder todos aqueles graciosos indicadores eletrônicos que normalmente indicam a temperatura.

Uma instalação em um grande órgão governamental americano recentemente construiu uma sala de máquinas extravagante e colocou um cluster Alpha de 256 nós nela para rodar enormes modelos científicos. Tudo estava ligado em um no-break e todos os recursos eram o estado-da-arte. Infelizmente, uma simples falta de energia elétrica derrubou o centro por horas. Por quê? O PC que controlava o sistema de ventilação (ar-condicionado) não estava conectado ao no-break. Ele falhou e bagunçou todo o sistema de ar-condicionado. Teste cuidadosamente.

Se você instalar um equipamento em uma empresa de serviços de *co-location* (o cliente coloca seu servidor nas instalações do fornecedor), verifique os recursos de energia elétrica sobressalentes da instalação hospedeira antes de assinar um contrato. Verifique se o gerador é real e é testado regularmente. Peça para estar presente no próximo teste de gerador; se puder ou não ver um teste real, provavelmente você obterá informações úteis.

Redundância de rede

Os ISPs ocasionalmente são absorvidos como parte de uma incorporação. Essas fusões demoliram os planos cuidadosamente elaborados de muitas empresas de manterem conexões redundantes com a Internet. Um ISP pós-fusão normalmente consolida circuitos que anteriormente pertenciam a empresas independentes. Clientes que antigamente tinham caminhos independentes para a Internet poderão passar a ter as duas conexões percorrendo um único conduíte e mais uma vez estarem à mercê de um único cabo de fibra ótica.

Os ISPs também eram famosos por oferecerem "circuitos redundantes ou "conexões de backup" questionáveis. Em uma inspeção mais rigorosa talvez você descubra que sim, há duas fibras, mas ambas estão no mesmo conduíte ou a conexão de backup transita em uma nuvem ATM já saturada. Faça uma revisão anual com seus ISPs para verificar se você ainda tem redundância genuína.

Incidentes de segurança

A segurança de sistema é abrangida em detalhes no Capítulo 20, *Segurança*. Entretanto, ela também merece ser mencionada aqui porque as considerações de segurança têm um impacto sobre a maioria das tarefas administrativas. Não há nenhum aspecto da estratégia de gerenciamento da sua instalação que possa ser projetado sem a devida consideração à segurança. Para a maioria deles, o Capítulo 20 se concentra em como evitar a ocorrência de incidentes de segurança. Mas pensar como poderia ser possível a recuperação a partir de um incidente relacionado à segurança é uma parte igualmente importante do planejamento de segurança.

O seqüestro do seu site Web é um tipo de invasão particularmente embaraçosa. Para o administrador de sistemas em uma empresa de hospedagem Web, um seqüestro pode ser um evento calamitoso, especialmente nos sites que administram dados de cartões de crédito. Choverão ligações telefônicas dos clientes, da mídia, dos VIPs da empresa que acabaram de ver as notícias sobre o seqüestro na CNN. Quem atenderá a essas ligações? O que essa pessoa deverá dizer? Quem será o responsável? Que papel cada pessoa terá? Se estiver em um ramo de alta visibilidade, realmente vale a pena pensar em uma situação deste tipo, bolando algumas respostas pré-planejadas e, talvez, até mesmo realizar uma sessão prática para ajustar os detalhes.

Sites que aceitam dados de cartão de crédito têm requisitos legais a serem administrados depois de um seqüestro. Certifique-se de que o departamento jurídico da sua empresa esteja envolvido no planejamento contra incidentes de segurança e certifique-se também de que há nomes de contato e números de telefone relevantes a chamar em um momento de crise.

Quando a CNN ou a Slashdot anuncia que seu site Web está fora do ar, o mesmo efeito que faz com que o trânsito de uma rodovia fique lento provocado pela curiosidade dos motoristas em observar um acidente faz com que seu tráfego Internet aumente enormemente, muitas vezes a ponto de romper qualquer coisa que você havia corrigido. Se o seu site Web não é capaz de lidar com um aumento no tráfego na ordem de 25% ou mais, considere a possibilidade de fazer com que o seu software para balanceamento de carga direcione o excesso de conexões a um servidor que apresente uma única página dizendo: "Desculpe-nos, estamos com carga excessiva para poder atender à sua solicitação no momento".

Histórias de segunda mão do World Trade Center

Ouvimos algumas histórias instrutivas relacionadas aos ataques em 11 de setembro de 2001 ao World Trade Center, em Nova York. Infelizmente, os administradores de sistemas que "possuíam" essas histórias foram incapazes de obter a permissão de seus gerentes para compartilhá-las. Em vez de desistirmos, incluímos aqui algumas histórias de segunda ou terceira mão cuja autenticidade não foi verificada. Elas certamente não são a palavra final sobre o gerenciamento de desastres, mas não obstante são interessantes.

Um ISP direcionou todas as suas chamadas e tráfego de rede por meio de uma instalação localizada em um dos menores prédios do complexo do World Trade Center. O prédio do ISP sobreviveu ao ataque e a instalação continuou a operar usando geradores de emergência (embora tivesse tido problemas de dissipar calor, já que o sistema de ar-condicionado do prédio estava inoperante). Infelizmente, o pessoal do ISP estava impossibilitado de adentrar o prédio para reabastecer os

tanques de combustível dos geradores. Moral da história: certifique-se de ter capacidade de gerador suficiente para alguns dias para situações de emergência prolongada.

Uma outra empresa localizada em uma das torres do WTC costumava manter sua instalação para recuperação em situações emergenciais na outra torre. Felizmente, a empresa repensou sua definição de "off-site" antes do ataque. Uma terceira empresa foi capaz de recuperar todos os seus dados cruciais que eram replicados em vários dispositivos em diversos locais diferentes, espelhados sincronicamente via fibra ótica. Entretanto, essa empresa em particular perdeu uma série de documentos em papel — nem tudo estava na forma computadorizada.

Uma empresa do mercado financeiro localizada nas torres estava terceirizando o armazenamento dos seus backups fora do site. Entretanto, essa empresa não sabia que o local do armazenamento também estava em uma das torres. É importante saber onde seus materiais são mantidos

O site Web da CNN passou por tráfego intenso em 11 de setembro que desafiou qualquer mensuração. O sistema telefônico em Nova York ficou sobrecarregado e os canais de TV que costumavam transmitir de antenas sobre o WTC saíram do ar. Todo o mundo pulou para a Internet e castigaram o site cnn.com. Inicialmente o site ficou impotente diante da sobrecarga. Após tentar contornar o problema deslocando servidores, a CNN acabou reduzindo a complexidade de sua home page para uma única imagem e texto puro sem links. Após algumas horas, o site estava recuperado e conseguindo atender à demanda.

A experiência desse pico intenso no tráfego Web resultou em alterações na diretiva da CNN relacionada ao momento em que a complexidade da home page poderia ser reduzida e quem poderia decidir sobre essa alteração. O poder e a responsabilidade passaram da hierarquia da gerência para os administradores de sistemas.

30.11 DIRETIVA ESCRITA

Ao pesquisarmos este capítulo, falamos com maiorais do mundo da administração de sistemas, segurança em computadores, comunidade de padrões e legislação de informática. Ficamos surpresos que todos eles mencionaram que "políticas por escrito e assinadas" são essenciais para uma organização saudável.

Devem existir vários documentos de diretivas:

- Diretivas de serviços administrativos
- Direitos e responsabilidades dos usuários
- Diretivas referentes a administradores de sistema (usuários com privilégios especiais)
- Diretiva de contas de convidados

Procedimentos na forma de listas de verificação ou receitas podem ser usados para codificar a prática existente. Eles são úteis tanto para administradores de sistemas novos como para veteranos. Melhor ainda são os procedimentos na forma de scripts executáveis. Os diversos benefícios dos procedimentos-padrão são os seguintes:

- As tarefas são sempre feitas da mesma maneira
- As listas de verificação reduzem a probabilidade de erros ou de etapas esquecidas
- É mais rápido para o administrador de sistemas trabalhar com base em uma receita
- As alterações são autodocumentáveis
- Os procedimentos por escrito fornecem um padrão mensurável de correção

Hoje, o Linux está substituindo os grandes mainframes do passado e realizando funções cruciais no mundo corporativo. Em grandes instalações, as listas de verificação, muitas vezes chamadas de "livros executáveis", servem como documentação para tarefas comuns. Elas normalmente são mantidas on-line e também na forma de manuais impressos. Os administradores de sistemas que escrevem e mantêm os livros executáveis normalmente são uma camada à parte do pessoal de suporte que os usa, mas tal organização e padronização vale a pena em longo prazo.

Eis algumas tarefas comuns para as quais você pode querer configurar procedimentos:

- Adicionar um host
- Adicionar um usuário
- Localizar uma máquina
- Configurar backups para uma máquina nova
- Proteger uma máquina nova
- Reiniciar um trecho de software complicado
- Reviver um site Web que não está respondendo ou não está servindo nenhum dado
- Descongestionar e reiniciar uma impressora

- Atualizar o sistema operacional
- Corrigir software
- Instalar um pacote de software
- Instalar software pela rede
- Atualizar software crucial (**sendmail**, **gcc**, **named** etc.)
- Fazer backups e restaurar arquivos
- Expirar fitas de backup
- Realizar desligamentos de emergência (todos os hosts, todos os hosts exceto hosts importantes etc.)

Muitas questões residem exatamente entre políticas e procedimentos. Por exemplo:

- Quem pode ter uma conta?
- O que acontece quando eles saem da empresa?

As resoluções de tais questões precisam ser colocadas por escrito de modo que você possa continuar consistente e evitar tornar-se vítima do manjado e infantil subterfúgio "A mamãe disse não, vamos perguntar ao papai!". Muitas vezes, a porção "se" é a diretiva e a porção "como" é o procedimento.

Algumas decisões políticas serão ditadas pelo software que você está usando ou pelas diretivas de grupos externos, como os ISPs. Algumas políticas são obrigatórias caso a privacidade dos dados de seus usuários precise ser protegida. Chamamos esses tópicos de "diretiva inegociável".

Em particular, acreditamos que endereços IP, hostnames, UIDs, GIDs e nomes de usuários devem ser gerenciados no nível global da instalação. Alguns sites (empresas multinacionais, por exemplo) são claramente tão grandes para se implementar uma diretiva dessas, porém se você conseguir fazê-lo, o gerenciamento válido para todo o site torna as coisas muito mais simples. Conhecemos uma empresa que impõe gerenciamento no nível global da instalação a 35.000 usuários e 100.000 máquinas, portanto o limiar em que uma empresa torna-se muito grande para gerenciamento no nível global da instalação deve ser bem alto.

Outras questões importantes talvez tenham um escopo maior do que aquele no seu grupo de administradores de sistemas local:

- Tratamento de brechas de segurança
- Controles de exportação de sistemas de arquivos
- Critérios para seleção de senhas
- Remoção de logins motivadas
- Material com direitos autorais (por exemplo, MP3s e DVDs)
- Pirataria de software

Manter bons canais de comunicação entre grupos administrativos em um site grande pode impedir problemas e ajudar a desenvolver confiança e cooperação. Considere a possibilidade de realizar uma festa como um veículo de comunicação. Alguns grupos de administradores de sistemas utilizam um IRC do tipo MUD ou MOO ou outro sistema de bate-papo como um veículo de comunicação. As pessoas podem passar muito tempo nesses sistemas de bate-papo, mas se eles forem utilizados apropriadamente podem fazer com que as atividades da empresa caminhem de forma mais fluida, especialmente se existirem pessoas que trabalham fora da empresa ou em casa. Bate-papos podem ser espionados, assim tenha cuidado para não enviar informações sigilosas sobre sua rede e sobre sua empresa por uma rede de terceiros.

Diretivas de segurança

O que você quer proteger? Seus dados? Seu hardware? Sua capacidade de recuperar rapidamente após uma quebra? Você deve considerar vários pontos de negociação ao projetar uma diretiva de segurança de uma instalação:

- Serviços oferecidos *versus* segurança fornecida (mais serviços = menos segurança)
- Facilidade de uso e conveniência *versus* segurança (segurança = 1/conveniência)
- Custo de segurança *versus* risco (custo) de perda

A RFC2196, *Site Security Handbook*, é um documento de 75 páginas escrito em 1997 por um subgrupo do IETF (Internet Engineering Task Force). Ela alerta os administradores de sistemas sobre vários aspectos relacionados à segurança de sistemas, políticas de usuários e procedimentos. Ela não tem uma receita para proteger um site Internet, mas contém efetivamente algumas informações inestimáveis. As últimas 15 páginas são um conjunto maravilhoso de referências, tanto on-line como impressas.

A RFC2196 sugere que seus documentos sobre diretivas tenham os seguintes pontos:

- *Diretrizes de compra* para hardware e software. Pode ser uma grande vitória envolver administradores de sistemas no processo de determinação destas diretrizes, pois eles muitas vezes conhecem defeitos de hardware, limitações de software e problemas de suporte que não são anunciados pelas equipes de marketing dos fabricantes.
- Uma *diretiva de privacidade* que estabelece expectativas referentes ao monitoramento dos e-mails e dos comandos dados pelos usuários e estabelece diretivas para lidar com arquivos de usuários.
- Uma *diretiva de acesso*: quem pode ter acesso, o que eles podem fazer com seus acessos, que hardware e software eles podem instalar etc. Esse documento deve incluir os mesmos alertas sobre uso desautorizado e monitoramento de linha que são inclusos na diretiva de privacidade.
- Uma *diretiva contábil* que diga claramente as responsabilidades tanto dos usuários como dos administradores de sistemas. A diretiva deve informar explicitamente quem pode espionar o tráfego de rede, ler e-mails dos usuários e investigar outras áreas igualmente sigilosas. Ela também deve delinear as circunstâncias em que essas investigações poderiam acontecer.
- Uma *diretiva de autenticação* que estabelece diretrizes para senhas e acesso remoto.
- Uma *diretiva de disponibilidade* que descreve quando o sistema deve ser ativo, lista horários de manutenção programados, dá instruções para relatar problemas e estabelece expectativas referentes a tempos de resposta.
- Uma *diretiva de manutenção* que inclui regras sobre terceirização e especifique procedimentos para dar acesso ao pessoal de manutenção de terceiros.

Notadamente está faltando na RFC2196 uma diretiva de autorização que especifique quem pode autorizar contas novas e privilégios estendidos. A RFC1244 original, *Site Security Handbook*, continha listas de questões concretas em vez de tipos de diretivas, o que poderia ser um pouco mais útil sob o ponto de vista do administrador de sistemas. A RFC mais nova inclui recomendações para cada tipo de serviço que uma máquina poderia executar e descreve os problemas dos serviços e possíveis soluções.

A ISO 117999 é uma referência mais atualizada e está sendo utilizada maciçamente na indústria. O COBIT é um outro padrão popular (e um que abrange mais do que apenas a segurança).

Seja lá quais forem as diretivas adotadas, elas devem ser explícitas, por escrito, entendidas e assinadas por todos os usuários e administradores de sistemas. O cumprimento das diretivas deve ser consistente, até mesmo quando os usuários são clientes que estão pagando por serviços de informática. A falha na aplicação uniforme de diretivas enfraquece sua validade legal e perceptível.

Acordos referentes a diretivas de usuários

No departamento de Ciências da Computação da Universidade de Colorado, a diretiva para usuários é emitida na forma de um shell inicial que imprime a diretiva e exige que os usuários concordem com ela e a "assinem" antes de obterem um verdadeiro shell para que possam usar suas contas. Esse esquema poupa tempo e dores de cabeça, porém consulte os *seus próprios* advogados antes de implementá-lo em sua instalação.

Eis algumas questões explícitas que devem ser tratadas em um acordo sobre diretivas para usuários:

- Compartilhar contas entre amigos e parentes (nossa sugestão: nunca)
- Executar programas para descobrir a senha local do sistema
- Executar programas para descobrir senhas de outros sites
- Interromper serviços
- Invadir contas alheias
- Fazer mau uso e forjar e-mails
- Uso geral de e-mails e de mídias eletrônicas
- Bisbilhotar arquivos de outros usuários (se puderem ser lidos? gravados? convidados?)
- Importar software da Internet (jamais? sempre? se o usuário verificar?)
- Empregar obrigatoriamente um firewall e um software antivírus nos hosts Windows
- Utilizar recursos de sistema (impressoras, espaço em disco, modems, CPU)
- Duplicar material com direitos autorais (software, música, filmes etc.)

- Permitir que outras pessoas dupliquem material com direitos autorais
- Utilizar recursos para atividades privadas ou comerciais
- Praticar atividades ilegais (fraude, difamação etc.)
- Buscar atividades que são legais em alguns lugares, mas não em outros (por exemplo, pornografia, atividade política)

Você deve perceber, porém, que nenhum acordo de diretiva pode evitar que alguém a viole. Você pode usá-lo como uma arma para descobrir o culpado, encerrar a conta deste, demiti-lo ou expulsá-lo ou o que parecer apropriado no seu ambiente.

Eis um exemplo de um acordo de diretiva breve e simples que o departamento de ciência da computação na Universidade de Melbourne exige que os alunos assinem para que possam utilizar os computadores da universidade:

Eu, abaixo-assinado, DECLARO PELO PRESENTE INSTRUMENTO que aceito e serei fiel às regras estabelecidas a seguir:

- Utilizarei os recursos de informática e de rede do Departamento exclusivamente para fins acadêmicos diretamente relacionados aos meus estudos das matérias de Ciências da Computação.
- *Entendo que o Departamento concede contas de computador para uso exclusivo do usuário dessa conta. Portanto, não autorizarei ou facilitarei o emprego de minha conta ou arquivos por nenhuma outra pessoa, nem divulgarei minha senha para qualquer outra pessoa.*
- *Não acessarei ou tentarei ganhar acesso a nenhum computador, conta de computador, rede ou arquivos sem a devida e explícita autorização. Tal acesso é ilegal segundo as leis estaduais e federais e é contrário às normas da Universidade. Informarei imediatamente à Secretaria do Departamento de Ciências da Computação caso fique ciente de que tal acesso ocorreu.*
- *Entendo que alguns softwares e dados residentes nos sistemas de arquivos que eu possa vir a acessar são protegidos por direitos autorais e outras leis e também por licenças e outros acordos contratuais; portanto, não violarei essas restrições.*
- *Não utilizarei os recursos da Universidade para obter, criar, executar ou distribuir cópias desautorizadas do software.*
- *Comprometer-me-ei a manter confidencial qualquer revelação a mim feita pela Universidade do software (inclusive métodos ou conceitos utilizados neste particular) licenciado à Universidade para uso em seus computadores e eu, por meio desse, salvaguardo e isento de quaisquer responsabilidades por perdas e danos a Universidade em relação a reivindicações de qualquer natureza que possam vir a surgir por motivo de qualquer revelação de minha parte a terceiros do dito software em violação deste comprometimento.*
- *Comprometo-me a manter o mais elevado padrão de honestidade e de integridade pessoal em relação ao emprego por minha parte dos recursos de informática e de rede do Departamento. Garanto, além disso, que evitarei quaisquer ações em relação ao emprego por minha parte dos recursos de informática e de rede do Departamento que possam vir a suscitar qualquer sintoma de má reputação ao Departamento ou à Universidade.*

Entendo que estou sujeito ao Regulamento 8.1.R7 da Universidade de Melbourne (descrito no Diário do Estudante), que também governa e regula emprego por minha parte dos recursos de informática e de rede da Universidade.

Entendo que o descumprimento de quaisquer dos princípios estabelecidos acima incorrerá em penas severas, inclusive reprovação numa matéria ou numa tarefa, suspensão ou revogação do direito de acesso aos recursos computacionais da Universidade, suspensão ou expulsão da Universidade, imposição de multas e/ou ação judicial, previstas no Crimes (Computer) Act 1988.[8]

Atenção especial às palavras evasivas em relação à honestidade, integridade pessoal e a não suscitar qualquer sintoma de má reputação à Universidade. Exigências vagas como essas dão a você algum espaço de manobra posterior e ajudam a abranger quaisquer detalhes que possam ter sido deixados de fora da diretiva. Embora o verdadeiro peso legal provavelmente seja insignificante, é uma boa idéia incluir tais exigências em seus acordos sobre diretivas adotadas.

Acordos referentes a diretivas para administradores de sistemas

O acordo de diretiva dos administradores de sistemas deve especificar as diretrizes para o uso de privilégios root e para honrar a privacidade dos usuários. Os gerentes de TI devem certificar-se de que a equipe dos administradores de sistemas entende e apóia as implicações desse documento. Ao mesmo tempo, é difícil responder à reclamação de um usuário de que o correio não funciona sem examinar as mensagens que rebateram. (Entretanto, uma cópia dos cabeçalhos costuma ser suficiente para caracterizar e resolver o problema.)

---

8. Tenha em mente que essa é uma lei australiana, embora legislação similar referente a computadores e software tenha passado nos Estados Unidos.

*Consulte a página 35 para informações adicionais sobre o **sudo** e página 477 para o John the Ripper.*

Se a sua instalação utiliza uma ferramenta como o **sudo** para acesso de root, é essencial que seus administradores de sistemas usem senhas fortes e não compartilhem seus logins com *ninguém*. Considere executar o John the Ripper contra as senhas dos administradores de sistemas regularmente. Também é essencial que os administradores não executem **sudo sh** ou utilizem um escape de shell depois de executarem **sudo** nos seus editores de textos favoritos; estes são simplesmente usos alternativos do **sudo** que driblam o recurso de login de **sudo**. O log não está aí para espionar os administradores; ele está aí para ajudar a reconstruir a seqüência de eventos quando algo dá errado.

Para alguns administradores de sistemas, a ânsia de demonstrar poderes de root ultrapassa o bom senso. De maneira gentil, mas firme, sugira outras carreiras alternativas.

Em algumas instalações, ter a senha root é um símbolo de status, talvez mais valioso que uma chave para o banheiro dos executivos ou acesso ao estacionamento interno. Freqüentemente, os funcionários que têm a senha são os engenheiros que não precisam dela ou não deveriam tê-la. Uma instalação que conhecemos ofereceu a senha root a todos os engenheiros, mas estipulou que todos que as possuíssem deveriam usar um bip e ajudar os outros quando necessário. As solicitações pela dita senha caíram vertiginosamente.

Uma outra técnica que usamos e que foi bem-sucedida é lacrar a senha root em um envelope e escondê-lo em um local conhecido pelo pessoal da administração de sistemas. Os administradores de sistemas geralmente utilizam **sudo** para realizarem seus trabalhos; caso efetivamente precisem da senha root por algum motivo, eles abrirão o envelope. Eles devem então alterar a senha root e armazená-la secretamente em um novo envelope. O procedimento é simples e direto mesmo que ainda seja suficientemente trabalhoso para motivar o uso do **sudo**. É importante manter as senhas antigas disponíveis temporariamente para as máquinas que estavam paradas ou off-line quando uma nova senha for escolhida.

Se sua equipe conhecer a senha root, você deverá modificá-la sempre que um membro da equipe sair. Pode haver várias senhas administrativas na sua instalação: para computadores, impressoras, hardware de rede, PROMs ou BIOSs, bancos de dados etc. Anote todas as coisas que devem ser alteradas (e como fazer isso) se um membro da equipe sair.

30.12 QUESTÕES JURÍDICAS

O governo federal e vários estados norte-americanos possuem leis referentes a crimes relacionados à informática. Em nível federal, há duas leis do início dos anos 90 e duas mais recentes:

- Federal Communications Privacy Act
- Computer Fraud and Abuse Act
- No Electronic Theft Act
- Digital Millennium Copyright Act

Algumas questões importantes na arena jurídica são as responsabilidades dos administradores de sistemas, dos operadores de rede e dos sites de hospedagem Web; a criptografia forte para comércio eletrônico; as redes não-hierárquicas e suas ameaças à indústria de entretenimento; as questões de direitos autorais; e as questões de privacidade. Os tópicos desta seção discutem essas questões e uma variedade de outras controvérsias legais relacionadas à administração de sistemas.

Criptografia

A necessidade de criptografia no comércio eletrônico e nas comunicações é clara. Entretanto, a criptografia é contra a lei em alguns países. Órgãos para cumprimento das leis não querem que cidadãos sejam capazes de armazenar dados que eles (a polícia) não conseguem decifrar.

Nos Estados Unidos, as leis referentes à criptografia estão mudando. No passado era ilegal exportar qualquer forma de tecnologia de criptografia forte. As empresas tinham de criar duas versões de software que incorporava criptografia: uma para vendas no mercado doméstico norte-americano e uma versão modificada para exportação. Um efeito colateral dessa diretiva foi o fato de que muitos projetos de software relacionados à criptografia acabaram sendo baseados em outros países. O absurdo patente dessa diretiva (o resto do mundo possuía tecnologia de criptografia há um longo tempo) e as necessidades do comércio eletrônico motivaram o governo a mudar tal postura.

Embora as restrições à exportação não tenham desaparecido completamente ainda, as mudanças introduzidas em 2004 tornaram a situação nos Estados Unidos melhor do que costumava ser. Para ver como isso poderia afetá-lo, acesse www.bis.doc.gov/Encryption.

O IETF trabalhou em padrões na área das comunicações seguras de ponta a ponta em nível de protocolo — a iniciativa IPsec —, e os fabricantes começam agora a fornecer sistemas que a incluem. Em geral, a parte referente à autenticação está embutida, mas a parte referente à criptografia em geral é instalada separadamente. Essa arquitetura preserva a flexibilidade para países em que a criptografia não pode ser usada.

Em uma recente (2006) decisão judicial de um tribunal norte-americano, o código de software foi equiparado ao direito de liberdade de expressão e julgado como sendo protegido pela primeira emenda da Constituição dos Estados Unidos. Em 1990, Dan Bernstein desenvolveu um novo algoritmo criptográfico enquanto aluno de graduação na U.C. Berkeley. Entretanto, ele não teve permissão de discutir o algoritmo publicamente pelo U.S. State Department, que classificou essa tecnologia como "munição". A Electronic Frontier Foundation (eff.org) foi atrás desse caso e acabou ganhando. Demorou apenas 16 anos!

Direitos autorais

As indústrias fonográfica e cinematográfica perceberam com alguma consternação que computadores domésticos são capazes de reproduzir música de CDs e de exibir filmes em DVD. Isto é tanto uma oportunidade como uma ameaça para eles. O resultado final dependerá da maneira como essas indústrias responderão, isto é, de forma preventiva ou reativa; infelizmente, parece que eles estão mais voltados para o caminho da reação.

O formato DVD utiliza uma chave criptográfica para bagunçar o conteúdo de um disco com um algoritmo chamado CSS (Content Scrambling System). A idéia era limitar a capacidade de reproduzir DVDs a aparelhos aprovados e licenciados. Os aparelhos de DVD dos consumidores incluem a chave de decodificação apropriada, assim como os reprodutores via software que vêm com a maioria das unidades de DVD para computadores.

Um estudante da Noruega aplicou engenharia reversa ao processo de criptografia CSS e postou um programa chamado DeCSS para a Web. O programa não evitava o esquema de criptografia dos DVDs; ele simplesmente utilizava a chave de decriptografia de um reprodutor Windows legítimo para decodificar o fluxo de dados do DVD e o salvava em disco para que pudesse executá-lo no Linux em vez de ter de usar o Windows.

A Motion Picture Association of America e a DVD Copy Control Association instauraram processos judiciais contra vários "distribuidores" do software DeCSS; todo o mundo que tivesse um site Web com link para uma cópia do DeCSS foi considerado um distribuidor. Os processos legais alegaram que os réus não estavam envolvidos no furto de materiais com direitos autorais, mas na distribuição de segredos comerciais e na "violação da proteção contra cópias"[9]. Nos Estados Unidos, esta atividade foi redefinida como uma forma de infração de direitos autorais pelo Digital Millennium Copyright Act (DMCA) de 1998. O MPAA ganhou esse caso; o DeCSS foi considerado culpado por violar o DMCA. Atualmente (2006), nenhum player de DVD de código-fonte aberto é legal para uso nos Estados Unidos.

Em dois casos recentes, empresas tentaram utilizar os direitos autorais e aspectos da engenharia anti-reversa da lei DMCA para asfixiar seus concorrentes: A Sears tentou proibir que abridores de portas de garagens de terceiros fossem utilizados para operar portas de garagens com a marca Sears e a Lexmark tentou utilizar essa lei para proibir que empresas de reciclagem de cartuchos vazios das impressoras a laser da Lexmark fizessem o refil e os comercializassem a um custo mais baixo do que os novos. Nos dois casos, a EFF representou a parte menos poderosa e ganhou a causa.

Um outro caso relacionado à DMCA, MGM contra Grokster e Streamcast, chegou à Suprema Corte norte-americana. Decisões anteriores (por exemplo, o caso Sony Betamax precedente da década de 1970) haviam afirmado que vender tecnologia capaz de infringir o material com direitos autorais de uma pessoa não era ilegal; somente o uso dessa tecnologia foi considerado uma infração e isso o tornaria culpado. No caso Grokster, a Streamcast teve lucros anunciando que seu software era capaz de duplicar música e filmes com direitos autorais protegidos. Esse incentivo à infração foi julgado como uma violação ao DMCA e a decisão final do tribunal foi majoritariamente contra a Grokster (9-0) e favorável à MGM.

Redes não-hierárquicas, como a BitTorrent, talvez estejam protegidas contra esse argumento "incentivador" uma vez que muitos aplicativos de compartilhamento de arquivos não envolvem material com direitos autorais protegidos. Entretanto, os detalhes continuam bem nebulosos.

A SCO sacudiu o mercado de provedores Linux ao utilizar bits com direitos autorais do código UNIX nas suas distribuições. Essa empresa entrou com milhares de processos legais contra indivíduos e um contra a IBM. ODSL, Open Source Development Labs, estabeleceu um fundo legal de 10 milhões de dólares para proteger os usuários Linux. Algumas empresas que suportam o ODSL são a Cisco, Dell, Ericsson, Fujitsu, Hitachi, HP, IBM, Intel, Nokia, Red Hat, Sun e Toshiba. Uma decisão judicial recente (julho de 2006) deu causa à ação da IBM de que a maioria dos mais de 250 processos da SCO era muito geral e não deveria ser considerada; acesse groklaw.net para detalhes adicionais. A forma dessa decisão desencorajou uma apelação, assim talvez as ações judiciais da SCO tenham chegado a um beco sem saída e talvez toda essa estupidez não tenha continuidade.

---

9 O DMCA proíbe a "violação" de medidas concebidas para manter o conteúdo digital seguro, mesmo se a razão dessa violação cair sob as isenções da legislação tradicional do "uso lícito" dos direitos autorais. Obviamente, essa modificação representa um compêndio significativo das isenções do uso lícito.

Privacidade

A privacidade sempre foi difícil de ser salvaguardada, porém com o advento da Internet, ela está em perigo mais do que nunca. Os registros médicos dos pacientes no sistema de planos de saúde da Universidade de Michigan foram inadvertidamente publicados na Internet. Os dados ficaram disponíveis à vontade por meses até um estudante ter percebido o descuido. Os bancos de dados cheios de números de cartões de crédito foram comprometidos. E-mails falsos que chegam quase diariamente parecem provenientes do seu banco e alegam que problemas com sua conta exigem que você verifique os dados da sua conta. Uma leitura mais minuciosa desse tipo de e-mail, em vez de deixar que ele seja interpretado com uma página Web, mostra que os dados seriam enviados a um hacker na Europa oriental ou na Ásia e não ao seu banco.

DoubleClick

Um outro grande escândalo referente à privacidade, esse intencional, envolveu a DoubleClick.net, uma agência de propaganda que cria a maioria dos anúncios de banners em páginas Web. A DoubleClick prometeu por anos que os usuários em seu sistema jamais seriam rastreados ou identificados. Recentemente, porém, eles compraram uma empresa que faz garimpagem de dados e começaram a coletar dados de cada usuário que visitava uma página Web contendo um anúncio da DoubleClick. O furor resultante fez com que a DoubleClick revogasse o projeto e contratasse dois advogados de peso para cargos na área de "privacidade" para descobrirem uma maneira de a empresa imobilizar legalmente os usuários que estavam sujeitos aos seus anúncios.

Rootkits da Sony

A Sony incluiu um programa chamado XCP (eXtended Copy Protection) em vários produtos de CD de música. O software se auto-instalava no computador do proprietário, ocultava seus arquivos de modo que o usuário não percebesse isso e gerava um relatório para a Sony sobre a música que estava sendo ouvida. Como o XCP é uma forma de rootkit, os arquivos instalados poderiam expor a máquina aos hackers; consulte a página 475.

Mark Russinovich comprou um CD da Sony com o DRM (Digital Rights Management) na amazon.com e o instalou no seu sistema Windows. Mais tarde, ele descobriu o rootkit da Sony e analisou seu impacto e extensão.

> "Não apenas a Sony havia adicionado um software ao meu sistema que usava técnicas comumente empregadas por um malware para mascarar sua presença, como também esse software havia sido mal escrito e não fornecia nenhuma maneira de desinstalá-lo. Pior, a maioria dos usuários que encontrava os arquivos ocultos (*cloaked files*) via uma varredura contra rootkits que danificava seus computadores quando tentavam o óbvio passo de excluir esses arquivos ocultos.
>
> Embora eu acredite no direito da indústria de entretenimento em utilizar mecanismos de proteção contra cópias a fim de evitar cópias ilegais, não acho que ainda encontramos o equilíbrio certo entre o uso lícito e a proteção contra cópias. Esse é um caso claro da Sony em levar o DRM longe demais".

O fato de o rootkit estar aí e não poder ser desinstalado é mencionado na licença EULA que os usuários devem assinar ao instalar o CD nos seus computadores.[10] No Reino Unido, isso é uma violação clara do Computer Misuse Act. Também é ilegal na Califórnia sob o California Business & Protections Code e punível com uma multa de US$ 1.000 por computador infectado. A EFF entrou com uma ação coletiva contra a Sony; para reivindicar danos, visite www.sonybmgcdtechsettlement.com.

Por que os administradores de sistemas devem se importar com isso? Bem, se houver um rootkit da Sony instalado nos seus PCs desktop, toda a sua instalação ficará potencialmente vulnerável a explorações.

Registros de chamadas e logs Web

Três das maiores empresas de telefonia norte-americanas foram acusadas de fornecer os telefones e registros Internet dos seus 200 milhões de clientes à National Security Agency. A Qwest, uma outra grande empresa de telefonia norte-americana, havia recusado cooperar com a NSA devido a questões de privacidade. Essa empresa manteve sua posição quando a NSA sugeriu veementemente que a Qwest poderia não receber contratos do governo norte-americano. Bom para a Qwest; a EFF está processando as três outras empresas de telefonia.

A história ilustra o abuso potencial que pode ocorrer em registros de chamadas. Em 2002, um computador do cartel de drogas colombiano foi confiscado pela polícia norte-americana.[11] Quando os dados contidos nos discos rígidos desse computador foram analisados, eles revelaram que, em 1994, os registros das chamadas telefônicas haviam sido utilizados para determinar a "proximidade" entre funcionários desse cartel e as autoridades. Vários funcionários que estavam "muito próximos" foram eliminados. A análise do tráfego é uma ferramenta poderosa e está sendo utilizada.

---

10. 10. EULA significa End-User License Agreement (Acordo de licensa de usuário).
11. Acesse amsterdam.nettime.org/Lists-Archives/nettime-l-0207/msg00015.html.

Vários dos grandes sistemas de pesquisa e portais de sites Web (por exemplo, Yahoo!) forneceram ao governo dos Estados Unidos dados das pesquisas na Web dos usuários. O Google foi o único a não atender às demandas do governo norte-americano. Mas o Google forneceu um sistema de pesquisa deficiente ao governo chinês, que queria censurar os resultados das pesquisas na Web por toda a China. Há informações de que o Google está revendo essa decisão.

A privacidade na rede de telefonia celular nos Estados Unidos também está causando incômodos na área de privacidade. À medida que você viaja nos Estados Unidos com um celular no bolso, esse celular verifica automaticamente outras celulares de modo que chamadas possam ser roteadas para você. Infelizmente, o governo não precisa de uma autorização nem mesmo de uma causa provável para rastrear a localização do seu celular.

A questão legal subjacente é a definição dos chamados dados do "pen-register" (dispositivo que registra os números de telefones discados). No contexto dos telefones, são as informações sobre quem e quando chamou, independente do conteúdo real da conversa. Em 1984, o Electronic Privacy Communications Act diferenciou os dados pen-register do conteúdo (os dois estavam inicialmente protegidos) e afirmou que não havia problemas para que as empresas de telefonia revelassem os dados de pen-register com base em uma ação judicial fácil de ser impetrada. Em 2001, o Patriot Act estendeu a definição de "números de telefone" a fim de incluir programas de software e informações relacionadas à Internet, por exemplo, endereços IP e URLs.

Conseqüentemente, os URLs visitados e as pesquisas na Web que você realiza agora são classificados como dados de pen-register. Eles não estão explicitamente protegidos como informações privadas. Antes que um administrador de sistemas possa entregar e-mails ou arquivos dos usuários é necessário um mandado, mas o conteúdo da maioria dos arquivos de log só exige uma ordem judicial.

Um outro recurso dos dados pen-register é que eles podem ser utilizados nos procedimentos jurídicos mesmo se obtidos ilegalmente pelas autoridades. Tempos assustadores.

EULAs baseadas em clique

EULA é o acrônimo para End User License Agreements que você precisa aceitar ao instalar um software que você comprou. A maioria das pessoas não lê esses acordos cuidadosamente, nem mesmo os lêem, elas clicam em Sim visto que já pagaram pelo software, abriram o CD e começaram o processo de instalação — é muito tarde para devolver o software e pedir o dinheiro de volta. Algumas EULAs são simplesmente malignas; eis algumas coisas com as quais você pode ser solicitado a concordar:

- Permitir que o fornecedor monitore ou audite seu computador (Red Hat!)
- Não criticar o produto nem o fornecedor (Microsoft)
- Não submeter o produto a um benchmark ou transmitir os resultados (McAfee)
- Renunciar ao direito de impor uma diretiva de privacidade declarada pela empresa (McAfee)
- Abster-se de personalizar ou reparar seu computador
- Permitir atualizações automáticas de software
- Não fazer reengenharia reversa no produto
- Não remover o produto do seu computador
- Não utilizar o produto junto com o produto de um concorrente

Uma lista longa e, seguramente, não discutimos várias outras...

Impondo uma diretiva

Os arquivos de log podem provar a você, sem sombras de dúvida, que a pessoa X fez a coisa errada Y, mas, para um tribunal, tudo isto é apenas evidência de boatos. Proteja-se com diretivas por escrito. Os arquivos de log às vezes incluem registros de data/hora que são úteis, mas não necessariamente admissíveis como uma evidência a menos que seu computador esteja executando o Network Time Protocol (NTP) para manter o clock sincronizado com um padrão de referência.

Talvez você precise de uma diretiva de segurança para processar alguém por uso indevido. Ela deve incluir uma declaração do tipo: *O uso desautorizado dos sistemas computacionais da Universidade pode envolver não só a transgressão da diretiva da Universidade, como também uma violação das leis estaduais e federais. O uso desautorizado é um crime e pode envolver penalidades criminais e civis; isso será processado com todo o rigor da lei.*

Aconselhamos que você coloque um alerta em **//etc/motd** (a mensagem do arquivo do dia) que informe os usuários sobre sua diretiva de "xereta". O nosso diz o seguinte:

```
As informações digitadas pelo teclado podem estar sendo monitoradas
para o caso de um incidente de segurança real ou suspeito.
```

Talvez você queira garantir que os usuários vejam a notificação pelo menos uma vez incluindo-a nos arquivos de inicialização que você dá a usuários novos. Se precisar usar **ssh** para fazer o login (e você deve usá-lo), configure então **sshd.config** para sempre exibir o arquivo **motd**.

Não deixe de especificar que, pelo simples ato de usarem suas contas, os usuários aceitam a diretiva por escrito imposta por você. Explique onde os usuários podem obter cópias adicionais dos documentos da diretiva e envie documentos-chave em um boletim apropriado. Inclua também a penalidade específica pelo descumprimento da diretiva (eliminação da conta etc.).

Controle = responsabilidade

Os ISPs têm, tipicamente, uma diretiva AUP (Appropriate Use Policy) ditada pelos provedores acima e a exigem de seus clientes abaixo. Esse "fluxo de cima para baixo" de obrigações atribui responsabilidade pelas ações dos usuários aos próprios usuários, não ao ISP ou ao provedor acima do ISP. Essas diretivas foram utilizadas como uma tentativa de controlar spam (e-mail comercial não solicitado) e proteger os ISPs nos casos em que os clientes armazenam material ilegal ou com direitos autorais em suas contas. Verifique as leis de sua região; essas podem variar muito.

Suponha que algo impróprio seja enviado via e-mail ou postado na Web a partir do seu site. Se você for a CompuServe (agora parte da AOL), isso é um problema. Em um caso chamado *Cubby versus CompuServe*, algo difamatório foi postado. O juiz determinou que a CompuServe não era culpada, porém definiu que o moderador do grupo de notícias para o qual ele foi postado foi negligente. Quanto mais você tenta controlar informações, mais responsável você se torna.

Esse princípio é maravilhosamente ilustrado pela história de uma empresa texana fundada por um de nossos estudantes empreendedores, Cheeser. Ele escreveu scripts Perl para garimpar os grupos de notícias Usenet, coletar imagens indecentes e construir um site Web pago baseado nesse conteúdo. Ele cobrava US$ 12/mês dos assinantes e estava ganhando muito dinheiro com isso.

Cheeser tentou ser um pornógrafo responsável e não se inscreveu em grupos de notícias sabidamente de pedofilia. Ele também monitorava vários grupos de notícias que estavam na crista da onda, algumas vezes com conteúdo ilegal, outras não. Esse pequeno descuido e sua opção de sediar seu negócio em um Estado tão conservador quanto o Texas o levaram à ruína.

Baseada em uma denúncia anônima (talvez de um concorrente), a polícia local confiscou os seus computadores. Certamente eles encontraram algumas amostras de pornografia infantil que havia sido postada a um dos grupos de notícias "mais seguros". O caso criminal jamais foi a julgamento, mas, durante a negociação da contenda, ficou claro que o juiz achava Cheeser culpado — não por ter criado aquele tipo de conteúdo, mas por não ser um censor suficientemente bom. A implicação era de que se Cheeser não tivesse feito nenhuma espécie de censura, ele estaria legalmente certo. Nunca censure sua pornografia.

Esse princípio também se aplica às demais interações com o mundo externo. Sob um ponto de vista legal, quanto mais você monitora o uso da Internet por parte de seus usuários, maior o seu nível de envolvimento e de responsabilidade pelas ações ou postagens de seus usuários. Caso saiba de alguma atividade ilegal ou acionável, você tem uma incumbência legal de investigá-la e de relatar suas descobertas à justiça.

Por essa razão, alguns sites limitam os dados que eles registram em log, o período de tempo no qual esses arquivos de log são mantidos e a quantidade de histórico de log mantido em fitas de backup. Alguns pacotes de software (por exemplo, o cache de Web Squid) ajudam na implementação dessa diretiva incluindo níveis de registro em log que auxiliam o administrador de sistemas a depurar problemas, mas sem violar a privacidade do usuário. Mas esteja ciente do tipo de registro em log que talvez seja exigido pela sua legislação local.

Os administradores de sistemas devem estar familiarizados com todas as diretivas relevantes da empresa ou da universidade e devem garantir que elas sejam cumpridas. Diretivas inconsistentes e não impostas são piores do que nenhuma, tanto sob o ponto de vista prático como legal.

Licenças de software

Muitos sites pagaram por cópias K de um pacote de software e têm N cópias em uso diariamente, onde K << N. Ser pego em tal situação poderia ser prejudicial à empresa, talvez mais prejudicial do que o custo dessas N menos K licenças adicionais. Outros sites recebem uma cópia de demonstração de um pacote de software caro e o burlam (reinicializam a data na máquina, descobrem o código de licenciamento etc.) para fazer com que ele continue a funcionar após a expiração do período de demonstração. Como um administrador de sistemas deve lidar com casos de violação dos acordos de licenciamento e fazer cópias do software em máquinas que não possuem licença? O que fazer ao descobrir que máquinas pelas quais você é responsável têm software pirata? O que dizer do shareware que nunca foi pago?

É uma questão muito difícil. Seus superiores normalmente não o apoiarão em seus pedidos de eliminação e pagamento de cópias não licenciadas de software. Normalmente, é um administrador de sistemas que assina o acordo para eliminar as cópias de demonstração após uma determinada data, mas é um gerente que toma a decisão de eliminá-las.

Até mesmo se o trabalho for o melhor que você já teve, sua integridade pessoal e profissional estão na mira. Felizmente, no mercado de trabalho atual, a demanda por administradores de sistemas bem qualificados está em alta e a sua recolocação

será rápida. Sabemos de vários casos nos quais o gerente imediato de um administrador de sistemas não queria se envolver na situação e disse ao administrador de sistemas para não complicar mais a situação. O administrador escreveu então um memorando ao chefe pedindo para corrigir a situação e documentando o número de cópias do software que estavam licenciadas e o número de cópias em uso. O administrador pegou algumas frases da licença de uso e mandou cópias do memorando para o presidente da empresa e para os gerentes de seu chefe. Em um caso, esse procedimento funcionou, e o gerente do administrador de sistemas foi mandado embora. Em outro caso, o administrador de sistemas pediu demissão quando até mesmo a gerência mais alta se recusou a fazer a coisa certa. Não importa o que você faz nessa situação, escreva tudo. Peça uma resposta por escrito ou se tudo o que você obtiver forem palavras faladas, escreva um memorando curto que documente o entendimento que você tem das suas instruções e o envie à pessoa encarregada.

Conformidade reguladora

Uma série de escândalos envolvendo os altos escalões de departamentos financeiros de empresas como a Enron e a WorldCom (MCI) inspirou a passagem de uma lei americana de 2002 chamada Sarbanes-Oxley Act, conhecida coloquialmente como SOX. A SOX especifica regras da área financeira para empresas com ações na bolsa nos Estados Unidos e torna os diretores corporativos pessoalmente responsáveis pela validade dos seus dados financeiros.

Como os computadores monitoram os registros da maioria das empresas, a SOX afeta a administração dos sistemas. Algumas medidas impostas pela SOX são fáceis (logs de e-mail, trilhas de auditoria, controles internos), mas outras são um pouco mais complexas. Por exemplo, a SOX exige a determinação de uma linha de base das atividades normais para um repositório de dados (por exemplo, banco de dados) e a criação de ferramentas para detectar comportamentos ou acessos anômalos. Você deve também documentar tudo e provar que utiliza regularmente os controles que você implementou. A conformidade com a SOX tem um preço sobre produtividade e a moral do administrador de sistemas.

Por outro lado, a Sarbanes-Oxley é uma grande vitória, se você vende dispositivos de armazenamento ou serviços de contabilidade. Para grandes empresas, o custo médio da conformidade com essa lei foi de 4 milhões de dólares no ano passado e esse custo tende a aumentar. Agora que executivos são diretamente responsáveis pelas violações e podem ser presos, às vezes eles exigem novos procedimentos estranhos para a administração de sistemas em nome da conformidade com a SOX.

A SOX afeta empresas privadas e também organizações sem fins lucrativos. Acordos de diretiva escritos que documentam por quanto tempo os dados são mantidos e quando exclusões agendadas ocorrerão tornaram-se importantes tanto para documentos impressos como para documentos eletrônicos.

Um administrador de sistemas estava fazendo algum teste de penetração para uma grande empresa de informática sob a direção de uma firma internacional de contabilidade. Ele provou que quase qualquer pessoa poderia invadir o banco de dados Oracle Financials de uma empresa como root e bagunçar os livros dessa empresa sem ser detectado — exatamente o tipo de coisa que a conformidade com a SOX deveria supostamente garantir e que é impossível. A firma de contabilidade quis que ele provasse isso alterando os dados financeiros do cliente. Ele recusou, ela persistiu e, depois de examinar cuidadosamente a questão, ele decidiu denunciar a firma de contabilidade ao escritório de conduta de negócios do cliente. No final a firma de contabilidade foi demitida e o administrador de sistemas perdeu seu trabalho logo depois.

Auditoria e governança de TI são questões importantes hoje em dia. Regulamentos e semipadrões para especificar, medir e apoiar a conformidade geraram vários acrônimos: SOX, ITIL, COBIT e ISO 17799, para citar alguns. Infelizmente, essa sopa de letras está deixando um gosto ruim na boca dos administradores de sistemas e atualmente não há um software para implementar todos os controles considerados necessários pela legislação recente.

30.13 Patentes de softwares

No início, a agência de patentes norte-americana determinava que você não poderia patentear um teorema matemático. O teorema tornou-se então um algoritmo e mesmo assim ele não poderia ser patenteado. Em seguida, o algoritmo foi implementado no hardware e isso certamente poderia ser patenteado. Firmware, talvez. Softwares, ainda não. Mas patentes podem são atraentes, e uma das instâncias jurídicas baixas gostava de patentes de software. Contra sua vontade, a agência de patentes começou a emiti-las, em alguns casos para aplicativos que haviam entrado com um pedido 10 a 15 anos antes.

Infelizmente, a agência de patentes historicamente tinha uma compreensão deficiente do estado-da-arte nos softwares e emitiu muitas patentes impróprias (alguns diriam, estúpidas). Há cinco patentes diferentes para o algoritmo de compactação de dados Lempel-Ziv. Esse algoritmo foi publicado em uma revista de matemática e foi implementado e distribuído no Berkeley UNIX. O conceito de um arquivo de inclusão foi patenteado. O conceito de um cursor foi patenteado. A subtração foi patenteada como uma medida de corrigir softwares que não eram compatíveis com o Y2K. O processo de copiar uma imagem da memória para uma janela na tela foi patenteado, e também o uso da operação XOR para tratar janelas sobrepostas. Vários padrões de criptografia de dados foram patenteados. O conceito de incorporar material publicitário a uma interface com o usuário foi patenteado.

De acordo com lei de patentes norte-americana, se certa invenção for patenteável, cada pequena melhoria agregada a essa invenção também poderá ser patenteada. No caso dos softwares, quase tudo pode ser visto como um passo no sentido de patentear. A ignorância do U.S. Patent and Trademark Office em relação a softwares e sua persistente incapacidade em julgar o que constitui uma arte ou obviedade já estabelecida fez com que essa agência emitisse milhares de patentes duvidosas de software a cada ano.

Nos Estados Unidos, uma ameaça ainda maior surgiu: a patente de software para métodos de negócios. As empresas convenceram a agência de patentes de que praticamente todo método de computadorizar um procedimento de negócio padrão é uma invenção patenteável. Patentes foram emitidas para atividades mundanas como examinar a conta de um cliente em um banco de dados computadorizado quando esse cliente faz uma chamada a um help desk. A Amazon.com obteve uma patente da prática de negócios sobre a tecnologia "1-click"; ela obteve uma injunção que exigia que a Barnes and Noble fizesse com que seus clientes dessem pelo menos dois cliques com o mouse para comprar livros.[12]

Por outro lado, a União Européia determinou em julho de 2005 que qualquer coisa que consistir em "esquemas, regras e métodos para realizar atos mentais, brincar com jogos, fazer negócios e criar programas para computadores" não é uma invenção e, portanto, não é patenteável. A votação foi um retumbante 648 a 14.

A agência de patentes norte-americana está tentando "limpar sua barra", mas parece que o dano já foi feito em muitos casos. Um marco importante foi o recall de 1994 de uma patente pertencente à New Media da Compton, que envolvia sistemas de recuperação para dados armazenados em CD-ROM. Alguns analistas consideraram isso amplo o bastante para abranger 80% de todos os produtos de CD-ROM existentes, embora isso seja provavelmente um exagero. No final, cada uma das 41 reivindicações foi invalidada por meio de uma campanha cara e demorada por parte dos fornecedores de softwares a fim de demonstrar a existência de uma arte já estabelecida.

A descoberta de uma arte já estabelecida é a fraqueza real no processo da agência de patentes norte-americana. As solicitações de patente são mantidas em segredo, e com muito pouca perícia de software na agência de patentes é difícil para essa agência saber quais aplicativos realmente representam uma nova tecnologia. Os processos legais irão conseqüentemente chegar a um fim, sendo os advogados os vencedores finais de cada caso.

No sistema norte-americano, mesmo a falha da agência de patentes em conceder uma patente para uma invenção proposta não irá impedir que ações fraudulentas relacionadas a patentes cheguem aos tribunais. O NTP, Inc. requisitou cinco patentes na ampla área de e-mail sem fio. A agência de patentes rejeitou todas; duas das cinco eram rejeições finais sem possibilidade de apelação. Ainda baseada nessas cinco patentes fictícias e rejeitadas, a NTP processou a empresa canadense Research in Motion, fabricante do celular BlackBerry para recebimento de e-mail.

RIM estava claramente em uma posição sólida. Ela havia feito todas as suas próprias pesquisas e desenvolvimento e os pedidos das patentes pela NTP haviam sido rejeitados pela agência de patentes. Contudo, o juiz que julgou o caso parecia inclinado a conceder à NTP uma injunção que tornaria ilegal que a RIM vendesse novos BlackBerrys nos Estados Unidos. Mais importante, essa injunção tornaria impossível que os clientes existentes continuassem a utilizar seus Blackberries para e-mail.

A RIM avaliou o risco de uma injunção como sendo muito grande; ela encerrou o processo legal com um pagamento de uma única vez de US$ 612.500.000 em troca de uma licença permanente para utilizar a tecnologia abrangida pelos cinco requerimentos das patentes rejeitadas da NTP. Um dia feliz para usuários do BlackBerry como nós, mas um dia triste para pequenas empresas com boas idéias e bons produtos.

Confira os repositórios de arquivos da Electronic Frontier Foundation em www.eff.org para mais especificidades. Uma boa fonte de notícias de primeira mão é a área de patentes da slashdot.org.

30.14 Padrões

O processo de padronização ajuda em alguns casos (interfaces Wi-Fi de diferentes fabricantes podem conversar entre si) e prejudica em outros casos (protocolos de rede OSI, milhões de dólares da engenharia de software desperdiçados). Os comitês de padrões devem codificar e formalizar práticas existentes, não inventar.

Os padrões destinam-se a nivelar o campo de atuação e a tornar possível para clientes adquirir produtos compatíveis de fornecedores concorrentes. Algumas das partes envolvidas no processo de padronização realmente só querem codificar práticas existentes. Outras têm uma finalidade mais política: para retardar um concorrente ou reduzir a quantidade de trabalho necessária para fazer com que os seus próprios produtos entrem no padrão.

Organizações governamentais são normalmente os maiores interessados em sistemas e aplicativos baseados em padrões. O emprego de padrões permite que eles comprem de forma competitiva sem favorecer uma determinada marca.

---

12 Talvez a Doubleclick.net pudesse derrotá-la patenteando um clique duplo…

Mas, alguns cínicos utilizam padrões como uma barreira comercial não-monetária — empresas retardam seus processos de padronização até que seus produtos "peguem" no mercado.

Há vários organismos de padronização, tanto formais como informais. Cada um deles tem regras diferentes para associação, votação e jogo político. Sob o ponto de vista de um administrador de redes ou de sistemas, os organismos mais importantes são o POSIX (Portable Operating System Interfaces), IETF (Internet Engineering Task Force, descrito na página 192).

LSB: o Linux Standard Base

O Free Standards Group é uma organização sem fins lucrativos que promove os padrões de código-fonte aberto — em particular, a iniciativa de padronização do Linux. Hoje em dia, aproximadamente 20 corporações colaboram com essa iniciativa, incluindo fabricantes importantes como HP, IBM e Silicon Graphics. Distribuidores importantes do Linux como Red Hat, Novell (SUSE), Debian, Mandriva e Turbolinux também participam. O objetivo é tornar mais fácil para os desenvolvedores de software independentes a criação de um produto de software para o Linux e transformá-lo em um único produto, não 20 ou 50 pequenas variações diferentes do mesmo produto.

A lista de itens canalizada por meio do processo de padrões inclui:

- Interfaces de bibliotecas, incluindo bibliotecas compartilhadas comuns
- Arquivos de configuração
- Comandos de sistema
- Um formato comum de pacote e das diretrizes de instalação
- Uma API para interfaces de sistema
- Padrões da hierarquia do sistema de arquivos

Quando escrevíamos este livro, o LSB 3.1, distribuído em meados de 2006, incorporava o padrão ISO do LSB básico (ISO/IEC 23360) e pela primeira vez também incluía funcionalidades de desktop que padronizavam os conjuntos de ferramentas GUI Gtk e Qt. O governo chinês, que está criando um padrão nacional para o Linux na China, assinou um acordo para basear seu Linux padrão no LSB.

Infelizmente, nenhum fornecedor distribuiu um aplicativo compatível com o LSB — nem ao menos um. Uma grande idéia com bom suporte da indústria que não chegou a lugar nenhum.

POSIX

POSIX, uma ramificação do IEEE que se encarrega das cópias de seus padrões, se engajou durante os últimos anos na definição de um padrão genérico para UNIX. A principal iniciativa no passado se concentrava nos comandos e chamada de sistema ou interfaces de bibliotecas. O Linux é compatível com o POSIX e segue os padrões POSIX.

ITIL: a Information Technology Interface Library

Isso é um padrão procedural em vez de um padrão técnico para o Linux, mas é um relevante para as organizações de TI maiores.

Na década de 1980, o sistema ITIL foi desenvolvido na Inglaterra para ajudar a gerenciar computadores mainframe e o desenvolvimento de softwares em grandes instalações. O foco eram os processos em vez das pessoas de modo que as tarefas dentro da empresa pudessem ser executadas de modo consistente independentemente de quem pediu demissão ou saiu de férias. Doze áreas funcionais que foram identificadas foram bem avaliadas quanto a necessidades de alto nível do TI na maioria das empresas européias (recuperação após desastre, controle de alteração, planejamento de capacidades etc.).

O ITIL permaneceu latente até que a Internet incentivou o nascimento da tecnologia da informação como um componente obrigatório na maioria das empresas. Grandes empresas precisam de alguma estrutura nos seus processos de TI e o ITIL não é pior do que uma outra estrutura qualquer. As melhores práticas da indústria foram definidas para cada uma das áreas ITIL.

Para diminuir o caos, o ITIL classifica cada evento ou atividade TI operacional como uma "alteração", um "incidente" (algum tipo de evento inesperado) ou um "problema" (algo a ser corrigido em longo prazo).

Há agora um padrão ISO (ISO 20000) que é semelhante ao ITIL, portanto, ocasionalmente, você verá instalações anunciando que são compatíveis com ITIL ou ISO. Em geral, o modelo ITIL/ISO não é exatamente o correto para sua empresa e você precisa contratar uma agência de consultoria para personalizá-lo. Vários dos sistemas comerciais de registro de problemas mencionados na página 642 afirmam ter compatibilidade parcial com o ITIL.

Muitas corporações grandes que operam nos Estados Unidos vêem o ITIL como uma forma de ajudá-las a resolver problemas dos seus bancos de dados e da administração de sistemas relacionados à lei SOX . Consulte a página 655 para informações adicionais sobre a lei SOX.

COBIT: Control Objectives for Information e tecnologias relacionadas

Como ocorre com o ITIL, o COBIT é uma estrutura para gerenciamento de informações e baseia-se nas melhores práticas da indústria. A missão do COBIT é "pesquisar, desenvolver, publicar e promover uma série internacional autorizada, atualizada de objetivos de controle da tecnologia da informação geralmente aceitos para uso cotidiano por parte de gerentes e auditores de negócios".

O COBIT é um filho da Information Systems Audit and Control Association (ISACA) e do IT Governance Institute (ITGI). Acesse www.isaca.org/cobit para detalhes.

A primeira edição dessa estrutura foi publicada em 1996 e agora estamos na versão 4.0, publicada em 2005. Essa última iteração foi fortemente influenciada pelos requisitos da Sarbanes-Oxley Act. Ela apresenta 34 objetivos de alto nível que abrangem 215 "objetivos de controle categorizados em quatro domínios: Planejamento e organização, aquisição e implementação, entrega e suporte e monitoração e avaliação. (Mas espere, isso não são oito domínios?)

30.15 A CULTURA LINUX

Nos anos 80 e 90 podíamos diferenciar usuários de PCs de usuários de Macs pelos seus níveis de intensidade. Os usuários de PCs achavam seus computadores uma ferramenta útil, mas os usuários de Mac adoravam suas máquinas. O computador de um usuário de Mac fazia parte da família, como o animal de estimação preferido.

A mesma intensidade que permeava o mundo Mac agora é muito forte na comunidade Linux. Os usuários do Linux não só gostam de seus sistemas — eles estão preparados para lutar por eles, consertá-los e torná-los melhores, mais rápidos e mais seguros do que um computador Windows jamais sonhou ser. Energia está fluindo no Linux em uma velocidade incrível, e a cultura Linux tem suas próprias éticas, mitos, deuses e heróis.

Um grupo de entusiastas do Linux do Bergen Linux User Group em Bergen, Noruega, estava intrigado por uma destas RFCs tipo Dia da Mentira do IETF: RFC1149, *A Standard for the Transmission of IP Datagrams on Avian Carriers.* Essa RFC define a CPIP (Carrier Pigeon Internet Protocol), que a equipe implementara utilizando um bando de pombos-e-mail de um vizinho. Eis um teste da implementação deles (extraído do site do projeto em blug.linux.no/rfc1149):

```
Script started on Sat Apr 28 11:24:09 2001
vegard@gyversalen:~$ /sbin/ifconfig tun0
tun0      Link encap:Point-to-Point Protocol
          inet addr:10.0.3.2  P-t-P:10.0.3.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:150  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:88 (88.0 b)  TX bytes:168 (168.0 b)

vegard@gyversalen:~$ ping -i 900 10.0.3.1[13]
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
vegard@gyversalen:~$ exit

Script done on Sat Apr 28 14:14:28 2001
```

Pode-se dizer que essas pessoas tinham muito tempo nas mãos, porém é exatamente esse tipo de criatividade e de entusiasmo que torna o movimento de código-fonte aberto (e o fonte do Linux participa deste) tão poderoso e tão divertido. O Linux é o máximo.

13. Note a utilização apropriada de espaço de endereçamento privado da RFC1918. Claramente, uma implementação do NAT baseada em pigeon seria necessária para conectar a rede de teste ao mundo exterior.

30.16 A ONDA LINUX

O Linux é hoje o sistema operacional que apresenta a mais rápida evolução e é executado em tudo, de mainframes a relógios de pulso. Entretanto, ele começa a sofrer um pouco das incompatibilidades e diferenças menores que, no auge do UNIX, eram empecilhos.

Era comum ouvirmos frases como "Ah, não podemos usar Linux — ele não é suportado!" nos círculos corporativos. Mas o mundo tem se alterado de forma tão rápida que chega a assustar a Microsoft. O Linux está abrindo suas trilhas pelo mundo comercial. Ele normalmente chega com alguns universitários recém-contratados que haviam utilizado Linux na faculdade e que o utilizam de maneira aberta ou de maneira clandestina em suas máquinas desktop. Após estabelecerem um posto avançado, o Linux normalmente se torna a plataforma preferida para servidores de e-mail ou servidores Web, nos quais sua segurança, desempenho e expansibilidade o tornam preferível em relação a soluções proprietárias da Microsoft. O custo da posse também é um grande fator no lado do Linux.

O "free" no software gratuito tem dois significados: gratuito como em cerveja de graça — você não precisa pagar por ele — e livre, por ter relativamente poucos limites sobre o que você tem permissão de fazer com ele. A GNU Public License impõe esses dois significados e esse fato assusta alguns departamentos de TI corporativos. Derivações dos trabalhos licenciados pela GPL herdam as estipulações GPL se você os distribuir. Alguns gerentes vêem os desenvolvedores em Linux como hippies com cabelos longos e acolhem a GPL com temor, incerteza e dúvidas. Mas o Linux parece estar mudando essa situação, lentamente, porém de maneira segura.

Nos últimos anos, a www.distrowatch.com avaliou as 100 melhores distribuições do Linux com base nos visitantes do seu site Web. Há muita agitação nas posições relativas das nossas cinco distribuições de exemplo. A Tabela 30.3 mostra alguns números.

Tabela 30.3 Ranking de popularidade da Distrowatch.com ao longo do tempo (em setembro de 2006)

Distribuição	Últimos 6 meses	Últimos 12 meses	2005	2004	2003
Ubuntu	1	1	1	13	–
SUSE	2	2	3	4	7
Fedora	3	4	4	2	–
Debian	7	7	6	5	5
Red Hat	29	24	23	12	2

O número real de hits por dia em geral varia de menos de 100 a mais de 3.000. Mais importante, essas avaliações demonstram claramente o crescimento do Ubuntu e a divisão entre o Red Hat Enterprise Linux e o Fedora.

Alguns parceiros corporativos importantes agora estão distribuindo e suportando o desenvolvimento do Linux: IBM, Hewlett-Packard, Compaq, Silicon Graphics e Sun, para citar alguns. É interessante observar o casamento entre esses fabricantes enormes e pesados com uma cultura Linux dirigida por engenheiros de software jovens e inexperientes cuja energia normalmente compensa sua inexperiência. A IBM teve de passar por uma mudança de paradigma para se aclimatar à vida no mundo anárquico do código-fonte aberto do Linux. Ele contribuiu portando suas extensas bibliotecas de código bem-testado. Acesse ibm.com/linux and ibm.com/developerworks/opensource para informações adicionais.

A SGI iniciou o Linux Test Project, uma série de conjuntos de teste e métodos de teste de carga que validam a confiabilidade, robustez e estabilidade do Linux. A IBM agora o mantém. A última distribuição (junho de 2006 no momento em que escrevíamos este livro) incluía mais de 2.900 testes, mais as ferramentas de documentação de teste.

O Linux Documentation Project (www.tldp.org) fornece documentação em muitos níveis de perícia técnica, incluindo livretos introdutórios relativamente longos, FAQs, HOWTOs para várias tarefas comuns e páginas man para todos os comandos e pacotes básicos do Linux.[14] O LDP também suporta duas revistas on-line, *Linux Gazette* e *Linux Focus*. Toda a documentação se encontra à disposição gratuitamente via Internet.

Pelo menos um fornecedor de sistemas de segurança contra invasão de computadores reconhece essa qualidade, cobrando cerca de 5% a 15% menos para assegurar servidores Web Linux do que para assegurar sistemas Microsoft equivalentes. Um artigo de agosto de 2001 da ZDNet fez os seguintes comentários:[15]

14. O site Web costumava ser o linuxdoc.org, mas um escândalo interno envolvendo um webmaster desatento ou malicioso, causou uma disputa em relação à posse do nome e um site Web comercial plagiado chamado linuxdoc.com motivou o projeto a mudar seu domínio.

15. http://www.zdnet.com/zdnn/stories/news/0,4586,2805929,00.html.

"A corretora de seguros J.S. Wurzler Underwriting Managers começou a cobrar até 15 por cento a mais em prêmios dos clientes que utilizam o software Internet Information Server da Microsoft, em que o worm Red Cod fez a festa... A Wurzler, que vende seguros contra hackers desde 1998, baseou sua decisão em mais de 400 análises de segurança feitas pela sua firma nos últimos três anos. Wurzler descobriu que os administradores de sistema trabalhando com sistemas de código-fonte aberto tendem a ser mais bem treinados e permanecem mais tempo com seus funcionários do que em empresas que usam o software Windows."

Das nossas cinco distribuições de exemplo, duas são suportadas por grandes empresas comerciais: o Red Hat por si só e o SUSE pela Novell; o Ubuntu é suportado por uma pequena empresa fundada pelo sul-africano Mark Shuttleworth chamada Canonical Ltd. O Debian e o Fedora, por enquanto, são os menos suportados.

Dentro da comunidade do Linux, criou-se uma divergência entre o Debian e as distribuições mais centradas em empresas como o Red Hat e o SUSE. O pessoal do Debian teve problemas em fazer com que suas boas idéias fossem levadas seriamente e incorporadas aos esforços de padronização do LSB. Esperamos que a popularidade recente do Ubuntu ajude a integrar o Debian de maneira um pouco mais sólida ao Linux mais representativo.

A distribuição CentOS (Community ENTerprise Operating System), baseada na Inglaterra, está ganhando popularidade nos ambientes que querem um sistema Linux estável sem o envolvimento do Red Hat. Ela é essencialmente um clone do Red Hat Enterprise Linux. A GPL requer que o Red Hat disponibilize seus patches de software ao público. O pessoal da CentOS corrige as árvores-fonte, remove logotipos e marcas comerciais do Red Hat e reconstrói a distribuição a partir do zero, o que resulta em algo semelhante ao RHEL sem nenhum custo ou contratos de suporte. Naturalmente, o suporte não é de todo desprezível, e o RHEL garante um suporte de 7 anos — quase uma eternidade no mundo Linux.

Para a maioria das empresas, o custo total da posse também é importante. Inicialmente, a maioria das distribuições é gratuita e também mais barata de manter graças a sua inerente gerenciabilidade pela rede. O Microsoft XP Professional custa atualmente cerca de US$ 275/máquina somente para o sistema operacional. Depois que os aplicativos são adicionados, o custo para uma grande empresa pode ser desestimulante. O Google.com atualmente é executado em milhares de máquinas Linux; imagine comprar uma licença Windows para cada uma dessas máquinas.

A pirataria de software é violenta no Extremo Oriente. Em um esforço para "limpar sua barra", o governo chinês preferiu adotar o Linux ao Windows porque ele é de código-fonte aberto e, portanto, não sujeito a pirataria. Há várias distribuições Linux com capacidades do idioma chinês. O Red Flag Linux é uma das mais conhecidas e vangloria-se de ter mais de 200 milhões de usuários.

O Linux é utilizado por muitas agências do governo norte-americano incluindo Nasa, NERSC, NIH, NOAA e USGS. Ele também é usado no Fermilab, Los Alamos, Oak Ridge e Sandia National Laboratories. Os computadores mais poderosos hoje em dia ainda são sistemas grandes — mas clusters Beowulf (veja www.beowulf.org) de máquinas Linux estão cada vez mais comuns. Em setembro de 2004, a IBM anunciou que tinha produzido, naquela época, o supercomputador mais rápido do mundo, um cluster de 16.250 CPUs que cabia em apenas oito racks, em execução sob o Linux com um desempenho sustentado de 36 teraflopes.[16]

30.17 ORGANIZAÇÕES, CONFERÊNCIAS E OUTROS RECURSOS

Muitos e grupos de suporte do UNIX e do Linux — tanto geral como específico a fornecedores — o ajudam a criar uma rede com outras pessoas que estão utilizando o mesmo software. A Tabela 30.4 apresenta uma lista resumida de organizações. Há uma quantidade enorme de grupos nacionais e regionais que não são apresentados nesta tabela.

Algumas organizações realizam conferências sobre tópicos relacionados com o Linux embora a maioria das conferências seja mais abrangente do que simplesmente Linux ou UNIX e incluem também eventos para Windows.

O Linux International e o Linux Professional Institute promovem o Linux de várias maneiras — o LI por meio de esforços de marketing para trazer o Linux para a comunidade comercial (e financiar algumas das iniciativas para o desenvolvimento de código-fonte aberto), e o LPI, por meio de seu programa de certificação de administração de sistemas Linux.

O site do Linux International possui bons subsídios para convencer gerentes de sistemas relutantes ao emprego de sistemas operacionais de código-fonte aberto. Eles patrocinam um estande Linux em diversas feiras, entre as quais a LinuxWorld, a Linux Expo e a CeBit. O LI também está conduzindo um projeto de internacionalização para trazer os conjuntos de caracteres de todas as línguas para o Linux. O programa de internato nas férias de verão reúne estudantes com desenvolvedores Linux para dar aos estudantes experiência prática na criação de programas de alta qualidade para ambientes de produção.

---

16 Um teraflop é um trilhão de cálculos de pontos flutuantes por segundo.

Tabela 30.4 Organizações patrocinadoras do Linux e do UNIX

Nome	URL	O que ela é
LPI	www.lpi.org	Linux Professional Institute
LI	www.li.org	O Linux International talvez se torne um grupo de usuários
FSF	www.fsf.org	Free Software Foundation, patrocinadora do projeto GNU
OSDL	www. osdl.org	Open Source Development Lab, para desenvolvedores
USENIX	www.usenix.org	Grupo de usuários UNIX; bem técnico
SAGE	www.sage.org	System Administrators Guild associada ao USENIX; sedia a conferência LISA todos os anos
LOPSA	www.lopsa.org	League of Professional System Administrators, uma subsidiária da USENIX/SAGE
SANS	www.sans.org	Patrocina conferências sobre segurança e administrador de sistemas ; menos técnica
EUROPEN	www.europen.org	Costumava ser as Nações Unidas dos grupos de usuários nacionais, mas agora está abandonada; remanescentes permanecem em NLUUG, DUUG, UKUUG, GUUG e outros
AUUG	www.auug.org.au	Australian UNIX Users Group, abrange aspectos tanto técnicos quanto de gerenciamento
SAGE-AU	www.sage-au.org.au	SAGE australiana, sedia anualmente conferências na Austrália.
SANE	www.sane.nl	O grupo System Administration and Network Engineering mantém conferências anuais na Europa do norte

A importância do Linux Internacional agora é menor já que seu objetivo de incorporar o Linux ao mundo corporativo foi alcançado. O LI talvez, no futuro próximo, se transforme em um algum tipo de grupo de usuários Linux.

FSF, a Free Software Foundation, é a patrocinadora do projeto GNU ("GNU's Not Unix" um acrônimo recursivo). O "free" no nome da FSF é a liberdade de expressão e não o da cerveja gratuita. A FSF também é a origem da GNU Public License, que abrange a maioria dos softwares Linux. O GPLv3 está atualmente sob desenvolvimento e esboços estão disponíveis para comentários públicos.

OSDL, Open Source Development Lab, é um grupo sem fins lucrativos cujo objetivo é acelerar a implantação do Linux no mercado corporativo empresarial. O OSDL é suportado pelos principais clientes Linux e fornece teste e suporte técnico à comunidade de desenvolvimento do Linux.

Conferências e feiras de exposições

A Tabela 30.5 lista os detalhes para as três conferências anuais sobre desenvolvimento Linux.

Tabela 30.5 Conferências sobre o desenvolvimento Linux

Conferência	URL	Comprimento	Onde:
Linux Symposium	www.linuxsymposium.org	4 dias	Ottawa, Canadá
linux.conf.au	linux.conf.au	5 dias	Austrália
Linux Kongress	www.linux-kongress.org	4 dias	Alemanha

A conferência de Ottawa, que normalmente acontece em agosto, inclui três séries de artigos e uma série paralela de tutoriais. A conferência australiana de desenvolvimento oferece tutoriais, seminários, idéias básicas e miniconferências, uma das quais é dedicada à administração de sistemas. Essa conferência ocorre em várias cidades australianas, normalmente em janeiro. A conferência alemã, Linux Kongress, incluindo dois dias de tutoriais e dois dias de apresentações técnicas, principalmente em inglês. Ela acontece em várias cidades alemãs e costuma ocorrer em setembro.

Linux Weekly News, lwn.net, é uma revista on-line dedicada ao Linux. Ela foi iniciada por Elizabeth Coolbaugh e Jonathan Corbet, em 1997. Ela é suportada por assinaturas, com membros obtendo acesso imediato a todos os artigos e não-membros que precisam esperar aproximadamente uma semana. A taxa da assinatura varia com base na sua categoria. "Starving hacker" é a mais barata a U$S 2,50 por mês, seguida pela "profissional hacker" a US$ 5 por mês e "project leader" a US$ 10 por mês.

A USENIX, uma organização de usuários de Linux, UNIX e outros sistemas operacionais de código-fonte aberto, sedia uma conferência geral e vários workshops e conferências especializadas (menores) a cada ano. A conferência tem um espaço

paralelo dedicado a sistemas abertos que apresenta desenvolvimento de sistemas operacionais em andamento nas comunidades Linux e BSD.

O grande evento para administrador de sistemas é a conferência USENIX LISA (Large Installation System Administration) que ocorre no terceiro trimestre de cada ano. As feiras normalmente são associadas a essas conferências.

Nos últimos anos, a USENIX também dedicou um dos seus workshops ao desenvolvimento do kernel do Linux como um serviço à comunidade Linux. O acesso a esse evento de dois dias é apenas por meio de convite.

A SAGE, System Administrators Guild (Comunidade de Administradores de Sistema) da USENIX, é a primeira organização internacional para administradores de sistemas. Ela promove a administração de sistemas como profissão, patrocinando conferências e programas informais. Acesse www.sage.org para mais detalhes.

Em 2005, uma controvérsia entre a USENIX e a SAGE colocou o futuro da SAGE em dúvida. O resultado foi que alguns antigos participantes na organização SAGE formaram uma empresa separada chamada LOPSA, League of Professional System Administrators, www.lopsa.org. Ela ainda não dá conferências, mas talvez comece a fazer isso em breve. A SAGE tinha um programa de certificação de administradores de sistemas, mas desistiu disso; esperamos que a LOPSA retome esse trabalho.

Muitas áreas locais têm grupos de usuários UNIX regionais, Linux, ou grupos de usuários de sistemas de código-fonte aberto. Alguns destes são afiliados ao USENIX e outros não. Os grupos locais normalmente têm reuniões regulares, workshops com palestrantes locais ou visitantes e freqüentemente jantam juntos antes ou depois das reuniões. Eles são uma boa maneira de contatar outros administradores de sistemas na sua área.

A maior feira de exposições do Linux é a conferência e exposição LinuxWorld Expo, que agora acontece uma vez por ano nos Estados Unidos, normalmente em San Francisco, no outono. Ela também é repetida na Europa e na Ásia por todo o ano; eventos futuros estão previstos em Pequim, Moscou e Cingapura.

A principal feira de exposições para a indústria de rede é a Interop; sua série de tutoriais também é de alta qualidade. A Interop costumava ser um evento anual que era ansiosamente aguardado pelos técnicos, bem como fabricantes da área. A Interop agora acontece várias vezes ao ano — poderíamos dizer um circo itinerante sobre redes. As comissões dos palestrantes foram cortadas pela metade, porém parece que a qualidade das palestras sobreviveu ao fato.

LPI: o Linux Professional Institute

O LPI certifica administradores de sistemas Linux. Isto é, ele administra testes que medem capacidade e conhecimento dos administradores de sistemas sobre vários aspectos do Linux. Há três níveis de certificação disponíveis, cada um dos quais formado por um teste de duas partes. Os testes até então (Níveis 1 e 2) são no formato múltipla escolha e são administrados via Web pelo Vue Electronic Testing Service (www.vue.com/lpi).

O Nível 1 parece cobrir comandos do superusuário do Linux e tarefas bastante básicas da administração de sistemas Linux. O Nível 2 é mais aprofundado e inclui redes. O Nível 3 foi definido, mas ainda não foi completamente desenvolvido. Infelizmente, o status do Nível 3 não mudou desde a última edição deste livro há cinco anos. O site Web da LPI (www.lpi.org) descreve os tipos de conhecimentos esperados em cada um dos níveis de certificação e fornece exemplos de perguntas de cada um dos testes.

A certificação é importante e difícil de fazer bem. A administração de sistemas é um tipo de ciência prática (ou ela é uma arte?). Um teste de múltipla escolha, embora fácil de ser aplicado e avaliado em termos de nota, não é muito bom na medição de habilidades para resolução de problemas que são a marca registrada de um bom administrador de sistemas. Esperamos que o Nível 3 da família de testes LPI envolva testes práticos em laboratório, mais parecido com a certificação CCIE da Cisco do que com o exame da MCSE da Microsoft.

O Red Hat e o SUSE têm programas de certificação próprios. A certificação do Ubuntu está atualmente na esteira da certificação LPI.

Listas de discussão e recursos Web

Os administradores de sistemas possuem acesso a uma imensa variedade de listas de discussão e mecanismos de busca na Web. Listamos alguns de nossos favoritos no Capítulo 1, *Por onde começar*. Além disso, há extensas listas de discussão relacionadas ao Linux disponível em www.linux.org/docs/lists.html.

Para questões específicas ao Linux, uma pesquisa específica sobre o Linux no Google pode ser bem eficaz. Acesse www.google.com/linux e veja como isso funciona.

Há também listas de discussão que focalizam distribuições particulares; a Tabela 30.6 mostra algumas especificidades para nossas distribuições de exemplo.

Tabela 30.6 Listas de discussão específicas a cada distribuição

Distribuição	URL
Red Hat	www.redhat.com/mailing-lists
Fedora	www.redhat.com/mailing-lists
SUSE	www.suse.com/us/support/online_help/mailinglists/
Debian	www.debian.org/MailingLists/
Ubuntu	lists.ubuntu.com

O Red Hat hospeda zilhões de listas de discussão que abrangem tanto a distribuição Enterprise Linux como Fedora; www.redhat.com/mailing-lists inclui o nome da lista e um breve resumo de cada uma. Além do URL mostrado nessa tabela, o SUSE também tem informações sobre listas de discussão em www.suse.com/us/business/mailinglists.ht. Os repositórios de arquivos dessas listas podem ser localizados em lists.suse.com/archives.

Pesquisas com administradores de sistemas

A SAGE (System Administrators Guild), associada à USENIX Association e o SANS (System Administration, Networking and Security Institute) realizam pesquisas anuais que se concentram principalmente nos salários dos administradores de sistemas. Essas pesquisas estão disponíveis em www.sage.org/salsurv e em www.sans.org/salary2005. Pesquisas recentes da SAGE (50–60 páginas) só estão disponíveis aos membros; o download das pesquisas antigas pode ser feito por qualquer pessoa. O URL da SANS produz apenas um resumo em PDF de cinco páginas dos resultados das suas pesquisas mais recentes.

Algumas das estatísticas coletadas são:

- Dados sobre salários: salário médio, histórico de aumentos, bônus e pagamento de horas extras
- Combinação de plataformas
- Horas trabalhadas por semana
- Demanda do mercado por administradores de sistema
- Nível de experiência, nível de educação, anos no emprego
- Distribuição geográfica, por sexo e por idade dos administradores
- As partes do trabalho mais apreciadas e aquelas mais maçantes

É interessante ver como sua empresa se compara à indústria da maneira retratada por esses resultados. Entretanto, os dados dessa pesquisa devem ser interpretados cuidadosamente. Os resultados são influenciados pelas condições locais e pelo estado da economia. Salários, custo de vida e impostos podem variar bastante entre países ou mesmo regiões.

30.18 LEITURA RECOMENDADA

Infra-estrutura

TRAUGOTT, STEVE. "Bootstrapping and Infrastructure." Boston: LISA 1998. Disponível em: <www.infrastructures.org/papers/bootstrap/bootstrap.html>.

SCHWEIKERT, DAVID. "ISGTC: an alternative to ~bofh/bin." Amsterdam, Netherlands: SANE 2004. Localizado em: isg.ee.ethz.ch/publications/papers/isgtc-sane.pdf

OETIKER, TOBIAS. "TemplateTree II: The Post-Installation Setup Tool." San Diego, CA: LISA 2001. Localizado em: isg.ee.ethz.ch/tools/tetre2/pub/tetre-lisa.pdf

BURGESS, MARK S. *Principles of Network and System Administration. 2. ed.* Hoboken, NJ: Wiley, 2003.

Muitas das ferramentas e conceitos descritos na seção sobre gerenciamento deste capítulo podem ser localizados em isg.ee.ethz.ch/tools.

Gerenciamento

LIMONCELLI, THOMAS A. E HOGAN, CHRISTINE. *The Practice of System and Network Administration.* Boston, MA: Addison-Wesley, 2001.

Esse livro tem um site Web associado, www.everythingsysadmin.com, que também é interessante. Uma nova edição está prevista para 2007.

MACHIAVELLI, NICCOLÒ. *The Prince*. 1513. Disponível em: <www.gutenberg.org/etext/1232>.

LIMONCELLI, THOMAS A. *Time Management for System Administrators*. Sebastopol, CA: O'Reilly Media, 2005.

BROOKS, FREDERICK P., JR. *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley, 1995.

Um pequeno livro maravilhoso, originalmente publicado em 1975 e em grande parte ainda verdadeiro.

McCONNELL, STEVE. *Software Project Survival Guide: How to Be Sure Your First Important Project Isn't Your Last*. Redmond, WA: Microsoft Press, 1998.

Esse livro é inteligente e bem escrito.

PRESSMAN, Roger S. *Software Engineering: A Practitioner's Approach. 6. ed.* Boston, MA: McGraw-Hill, 2005.

SOMMERVILLE, IAN. *Software Engineering. 8. ed.* New York, NY: Addison-Wesley, 2006.

O site www.itil-toolkit.com é um bom ponto de partida se você busca entender os vários jargões e o idioma de gerenciamento associado com processos e padrões ITIL.

Diretiva e segurança

RFC2196, *Site Security Handbook* e sua predecessora, RFC1244. O conteúdo é bem diferente, portanto vale revisar ambas as versões.

Diretivas locais, padrões e procedimentos para o San Diego Supercomputer Center estão disponíveis em: security.sdsc.edu/help/SGs.shtml. Essa é uma excelente coleção de documentos sobre diretivas, mas já passaram alguns anos sem uma atualização.

Questões legais, patentes e privacidade

O site da Electronic Frontier Foundation, eff.org, é um excelente lugar para encontrar comentários das questões mais recentes sobre privacidade, criptografia e legislação. Sempre uma leitura interessante.

www.groklaw.net é um excelente site para notícias sobre questões legais relacionadas à tecnologia da informação. As questões estão descritas e resumidas em inglês em vez de em um idioma jurídico.

Notícias gerais sobre a indústria

slashdot.org é uma boa fonte de notícia sobre tecnologias, mas costuma estar um ou dois dias atrasados em relação a fontes como as lista de discussão sobre segurança.

HARROW, JEFFREY R. *The Harrow Technology Report*. Disponível em: <www.theharrowgroup.com>.

Esse site hospeda alguns artigos interessantes sobre questões relacionadas à tecnologia. Ele é uma combinação de notícias e conteúdo editorial. J. R. Harrow é autor do extinto periódico *The Rapidly Changing Face of Computing* que foi inicialmente publicado pela Compaq. O site foi mantido até o final de 2005.

www.heise.de é um excelente site para notícias. Ele está em alemão, mas www.heise.de/english fornece uma tradução para o inglês.

www.theregister.co.uk ou www.theregister.com é bom site de notícias sobre TI.

30.19 EXERCÍCIOS

E30.1 Quais são os procedimentos recorrentes da sua empresa? Quais são raramente realizados e reinventados toda vez? Quais são arriscados?

E30.2 Quais são suas dependências em relação a provedores externos? Você precisa e tem um plano B? Explique por que ou por que não. Descreva o plano B, se houver um.

E30.3 Entreviste brevemente vários clientes internos para determinar suas expectativas com relação à disponibilidade da infra-estrutura computacional. As expectativas são consistentes? Elas são razoáveis? Elas são consistentes com os objetivos declarados do grupo de administração de sistemas?

E30.4 Qual infra-estrutura organizada para o gerenciamento de sistemas já foi estabelecida na sua instalação? Identifique as partes que ainda não estão presentes.

E30.5 Um dos seus colaboradores está saindo para almoçar amanhã e não retorna, mas você ainda não sabe qual deles. Quais procedimentos críticos talvez sejam afetados e como sua empresa está preparada para substituir o membro ausente da equipe? Qual documentação teria de existir a fim de evitar a interrupção de um serviço?

E30.6 O que aconteceria se *você* não aparecesse no trabalho nos próximos três meses? Quanto seus colegas o odiariam quando você por fim voltasse e por quê? O que você poderia fazer nas duas próximas semanas para reduzir o trauma desse evento?

★ E30.7 Seu chefe pede que corte o orçamento da administração de sistemas em 30% até o final do ano atual. Você pode quantificar as conseqüências desse corte? Apresente um resumo que permitirá que o chefe tome uma decisão informada com relação aos serviços a reduzir ou descontinuar.

★ E30.8 Quais são alguns dos principais apoiadores corporativos atuais do Linux? Quais são os seus interesses e motivações? Que tipo de contribuições eles oferecem?

★ E30.9 Você está fazendo a limpeza do sistema após um problema no disco e percebe a presença de arquivos no diretório **lost+found**. Ao investigar com maior cuidado, descobre que alguns dos arquivos são mensagens de e-mail que foram trocadas entre dois estudantes que estão armando um backdoor em torno do firewall do departamento para poderem guardar arquivos MP3 no servidor de arquivos remoto. O que você deve fazer? Há diretivas ou regulamentos corretos que cobrem esses incidentes?

★ ★ E30.10 Avalie a documentação local de sua instalação para usuários novos, administradores de sistemas, procedimentos de padrões e emergências.

★ ★ E30.11 Tente prever o futuro das diferentes versões comerciais e gratuitas/livres do Linux ao longo dos próximos cinco anos. Como o desenvolvimento e modelos de distribuição atuais se sustentam ao longo do tempo? Qual será o impacto em longo prazo da adoção do Linux pelos fornecedores de hardware? Estabeleça diferenças entre os mercados de servidor e de desktop.

Índice

A

A DNS, registros, 277, 285
AAAA DNS, registros, 283
Abell, Vic, 54
acesso, banco de dados, sendmail, 410
ACLs, DNS, 300, 316, 317
ACLs, firewall, 486, 491
ACLs, sistema de arquivos, 63, 66, 574
acordos de diretivas, 652
acordos de diretivas do usuário, 650
Adams, Rick, 226
Address Resolution Protocol (ARP), 195, 209, 210, 222
adjtimex, chamada de sistema, 622
administração de sistemas, 13, *Ver também* responsabilidades do grupo de administração de sistemas; segurança; hardware
  acordos de diretivas, 651, 652
  automação, 634, 636
  caixa de ferramentas, 552, 634, 635
  comprando hardware, 540, 543, 630, 631
  configurando múltiplas máquinas, 351
  desenvolvimento, 632, 636
  documentação, 640, 642
  fontes de alimentação de energia de emergência, 646
  gerenciamento, 624, 630
  lista de tarefas de e-mail, 369
  listas de verificação, 648
  mantendo usuários felizes, 623, 624
  operações, 636, 637
  organizações e conferências, 662, 664
  papel da, 630, 632
  questões legais, 652, 658
  recuperação após desastre, 116, 645, 648
  recursos Internet, 10
  registro e rastreamento de problemas, 642, 645
  resultados das pesquisas, 665
  scripts locais, 634, 636
  síndrome de personalidade, 13
  SOX (Sarbanes-Oxley Act), 657
  suporte, 637, 640
  tarefas essenciais, 11, 13
  testando soluções, 627
ADSM/TSM, sistema de backup, 141
agendadores de E/S, 562
agendamento em tempo real, 40
agendando comandos, 107, 111
agentes de acesso, e-mail, 372
agentes de entrega, 399
  cyrus, 399
  de e-mail smrsh, 421
  discard, 414
  e-mail, 371
  e-mail (MSA), 372
  error, 411, 414
  fax, 399
  locais, sendmail, 371
  pop, 399
  qpage, 399
agentes de transporte, e-mail, 371
agentes de usuário, e-mail, 370
agetty, processo, 589, 591
aio, daemon, 616
AirPort, 253
AIT, fitas de backup, 118
ajustando
  discos IDE, 92
  kernel, 221, 223, 602, 603
  NFS, 345
  parâmetros de rede, 221, 223
Albitz, Paul, 296
aliases (e-mail)
  para novos usuários, 68
aliases, arquivo, 76, 379, 383
aliases, e-mail, 379, 383, *Ver também* e-mail; sendmail
  abuso, 413
  banco de dados com hash, 383
  distribuição, 111
  e listas de discussão, 384, 386
  exemplos, 381
  formato de arquivo, 379
  globais, 68
  loops, 380, 383
  postmaster, 380
  root, 381
aliases.db, arquivo, 383
alien, 167
Allman, Eric, 150, 369, 388, 394, 414
allow-recursion, opção, DNS, 297
allow-update, cláusula, DNS, 303, 315
always_add_domain, recurso (sendmail), 400
Amanda, sistema de backup, 140
amavisd, filtro de vírus de e-mail, 442
ambientes de área de trabalho, 523, 525
amd, 617
American Power Conversion (APC), 551
American Registry for Internet Numbers (ARIN), 204, 205, 207, 260
Amstadt, Bob, 569
Anaconda, 162
anacron, 111, 611
análise de desempenho
  ferramentas
    mpstat, 557
  tempo de atividade, 557
Analog Apache, analisador de logs, 503
Anderson, Paul, 184
Anvin, H. Peter, 161
Apache. *Ver* hospedagem Web
Apache Software Foundation, 8, 501
APC (American Power Conversion), 551
APNIC (Asia-Pacific Network Information Center), 268
APT, ferramenta de software, 171, 174
apt-ftparchive, 173
apt-get, 171, 174
apt-proxy, 173
ar-condicionado, 549, 551
ARIN (American Registry for Internet Numbers), 204, 205, 207, 260
ARK, linguagem, 185
armazenamento de backup fora do local de trabalho, 115
armazenamentos de mensagens, 371
Armstrong, Jason, 431
ARP (Address Resolution Protocol), 195, 209, 210, 222
arp, comando, 209
ARPANET, 193
arquivos. *Ver também* arquivos de configuração; arquivos de dispositivos; atributos de arquivo; nomes de arquivo
  bloqueio do NFS, 340
  compartilhando com o Samba, 573
  diretório, 56, 57
  dispositivo, 87
  dispositivo de blocos, 56
  dispositivo de caracteres, 56
  excluindo, 56
  links físicos (hard links), 57
  links simbólicos, 56, 58
  links *vs.* arquivos originais, 57
  modos. *Ver* atributos de arquivo
  permissões, 59, 62, 474
  pipes identificados, 56, 58
  posse de, 32, 33
  regulares, 56
  removendo temporários, 110
  servidores, arquivos de sistema, 356
  servidores, NFS dedicado, 347
  sockets de domínio local, 56, 58
  tipos de, 55, 58
arquivos de configuração
  compartilhando, 351, 367
  copiando, 353, 357
  distribuindo, 353
  extraindo, 356
arquivos de descrição de impressora (PPD), 532
arquivos de dispositivos, 57, 58
  atributos, 60
  criando, 600
  de bloco *vs.* de caracteres, 600
  de blocos, 56, 57, 600
  de caracteres, 56, 57, 600
  nomes, 601
  números de dispositivos primários/secundários, 599, 600
  para discos, 87
  para drivers de unidade fita, 121, 601
  para portas seriais, 589, 601
  script MAKEDEV, 57, 600
  segurança, 475
  udev, 57
arquivos de inicialização, 75, 76
arquivos de log, 150, 158, *Ver também* syslog; registro em log
  analisando e pesquisando, 158, 159
  arquivando, 146
  hospedagem Web, 503
  lista de, 146, 147
  listas de, 156
  monitorando, 158, 159
  para Apache, 503
  para console do sistema, 157
  para logins e logouts, 147
  rodízio, 111, 145, 149, 150
  substituindo enquanto em uso, 145
arquivos de mapas, automontador NFS, 349, 350
arquivos regulares, 56
arquivos temporários, removendo, 110
Arusha Project, 185
Asia-Pacific Network Information Center (APNIC), 268
AT&T Bell Labs, 4
at, comando, 611
ata, daemon, 616
ATA/ATAPI, interface. *Ver* IDE
ATAPI CD-ROM, nomes do dispositivo, 601
ataques de negação de serviço (denial of service – DOS), 153
atd, daemon, 611
Atkins, Todd, 158
ATM, redes, 254, 255
atributos de arquivo, 58, 63
  ACLs, 63, 66
  alterando, 58, 61, 62
  bit de pesquisa de diretório, 59
  bits de permissão, 59, 474
  bits setuid/setgid, 33, 59, 60
  chattr, 62
  contagem de links, 60
  em arquivos de dispositivos., 60
  exibindo com ls, 58, 59, 61
  links simbólicos, 58
  lsattr, 62
  número de ID de grupo, 60

número de ID de usuário, 60
número de inode, 60
sticky bit, 59, 60
suplementando, 62
atualizações, 125
atualizando arquivos de zona, DNS, 313, 315
auth.log, arquivo, 147
autofs, script, 348
automontadores
amd, 617
automount, 348, 349, 617
configuração, 348, 350
NFS e, 348, 350
Windows, 574
automount, daemon, 348, 349, 617
autores, contato com, xiv
AutoYaST, 164
AUUG, grupo, 663
avoid-v4-udp-ports, opção, DNS, 299
AWStats Apache, analisador de logs, 503

B

backspace, tecla, 592
backups, 113, 141, *Ver também* Bacula; mídia, backup; software e sistemas, backup
agendas, 122, 123
ajustando na mídia, 114
armazenamento fora do local de trabalho, 115
compactação, 117
configurando, 120, 125
de um disco danificado, 95
dicas, 113, 116
incrementais, 121
intervalo entre, 114
para atualizações, 125
para discos removíveis, 118
para Windows, 140
planejamento contra desastres, 645
programas, 125, 140
projetando dados para, 116
quando fazer, 115
restauração completa, 124, 125
restaurando, 123, 125, 645
segurança, 115, 476
software e sistemas
dump/restore, 125
tar, 125
tamanho de sistema de arquivos, 114
Bacula, 127, 140, *Ver também* Bacula, arquivos de configuração
arquitetura, 128
backup manual, 135
daemon de arquivo de cliente, 134
daemons, iniciando, 134
instalação, 129
manual backup, 136
pools de mídia, 135
restaurando arquivos, 136
restoring files, 139
solucionando problemas, 139, 140
Bacula, arquivos de configuração, 129, 134,
arquivo /etc/bacula/bacula-fd.conf, 134
arquivo /etc/bacula-dir.conf, 130, 133
arquivo bacula-sd.conf, 133
arquivo bconsole.conf, 133
diretório /etc/bacula, 129
Bailey, Mick, 610
baixo nível, formatação, discos, 88
balanceamento de carga
discos e sistemas de arquivos, 555, 561
medida, 557
servidor Web, 500, 501
servidores, 555
banco de dados de acesso, sendmail, 411, 413
bancos de dados. *Ver também* MySQL
administrativos, 352, 357
banco de dados de impressora Foomatic, 532, 540
de dispositivos USB suportados, 541
DNS, 265, 272, 287
NIS, 357, 358
sendmail, 401, 402, 411, 413
Baretta, Anne, 593
bash, shell, 3, 23, 70
.bash_profile, arquivo, 75
.bashrc, arquivo, 75
Bastille Linux, 492
BCP, documentos, 195
Beowulf, clusters, 662
Bergen Linux User Group, 660
Berkeley DB, biblioteca, 179, 180, 383, 401, 436
Berkeley Internet Name Domain, sistema. *Ver* BIND
Berkeley UNIX, 4
BGP, protocolo de roteamento, 238
/bin, diretório, 54
BIND. *Ver também* DNS; servidores de nomes; named
ACLs, 316, 317
arquivo /etc/named.conf, 295, 315, 316, 337
arquivo /etc/resolv.conf, 293, 294
arquivo /etc/rndc.conf, 305
arquivo /etc/rndc.key, 305
arquivo de chave DNSSEC .private, 318
arquivo DNSSEC .key, 318
arquivo root.cache, 304
arquivos de configuração, 295, 296
atualizando arquivos de zona, 313, 315
chaves, gerando, 321
componentes, 288
configuração, 294, 312
configuração de cliente, 293, 294
depurando, 326, 335
desempenho, 335
dicas do servidor-raiz, 304
diretiva $GENERATE, 281, 284
diretiva $INCLUDE, 284
diretiva $ORIGIN, 284
diretiva $TTL, 284
diretório /etc/named, 297
DNS dividido, 306, 308, 311
DNSSEC, 272, 320, 324
dnssec-keygen, 318, 321
dnssec-signzone, 322
doc (domain obscenity control), 334, 335
documentação, 337, 338
encaminhamento de consulta, 299
endereço loopback, 305
estatísticas, 331
/etc/named.conf, arquivo, 312, 330, 336
/etc/syslog.conf, arquivo, 327
exemplo de configuração da zona host local, 307
exemplo de configuração ISC, 311
exemplos de configuração, 312
exemplos de configuração BIND, 307
informações específicas à distribuição, 337
informações específicas de distribuição, 335
interfaces de shell. *Ver* dig e nslookup
KSK (key signing key), 320, 322
listas de concordâncias de endereços, 295
named-checkconf, 295, 319, 335
named-checkzone, 295, 335
nsupdate, 314
opções de notificação, 297
registrando em log, 287, 302, 313, 327, 330
requisitos de hardware, 294
rndc, 305, 306, 313, 330, 331
rndc-confgen, 305
script nanny, 313
segurança, 292, 297, 316, 325
servidor escravo, configurando, 303
servidor mestre, configurando, 303
teste de conversores, 294
transferências de zona, 289, 298, 313, 314
transferências de zona incrementais, 300
transferências de zonas incrementais, 313
transferências incrementais de zonas, 272
versões, 287, 288
zona direta, configurando, 305
zona, comandos, 285
zonas assinadas, criando, 321
zonas stub, configurando, 303
BIOS, 18
em placas SCSI, 95
inicialização e IDE, 81
nas placas SCSI, 95
blackhole, opção, DNS, 299
Blandy, Jim, 179
blocos defeituosos, disco, 88
bogus, diretiva, DNS, 302
/boot, diretório, 54
boot.log, arquivo, 148
boot.msg, arquivo, 148
BOOTP, protocolo, 220
bootparamd, 619
broadcast, endereços, 199
broadcasts direcionados, rede, 223
BSD (Berkeley UNIX), 4
BSD FFS, sistema de arquivo, 86
BugTraq, 493
Burgess, Mark, 184
BUS, sinal, 42

C

CA (Certificate Authority), 505
cabeamento
escritórios, 257
rede, 246
construindo, 257
cabeamento de rede, 642
cabeamento/fiação de rede
analisador de cabos, 257
analisador de rede Wireshark, 257
construindo, 258
escolhas de cabo, 247, 250, 257
manutenção e documentação, 259
para escritórios, 257
cabos. *Ver também* conectores
10*Base*, 247
10Base, 249
cabo blindado de par trançado (shielded twisted-pair – STP), 582
cabo de par trançado não-blindado (unshielded twisted pair – UTP), 582
categoria, 247, 250, 257
categorias de cabos, 249
DB-9 para DB-25, 585
diretos, 583
Ethernet, 197, 248, 250
IDE, 81
mini DIN-8 para DB-25, 584
modem, 583
modem nulo, 583
padrão RJ-45 Yost, 586, 587
RJ-45 para DB-25, 586
rotulando, 259, 642
SATA, 82
SCSI, 82, 84
cabos seriais
diretos, 583
limites de comprimento, 588
modem nulo, 583
padrão RJ-45 Yost, 586, 587

Cacti, ferramenta de monitoramento de desempenho, 461
CAIDA (Cooperative Association for Internet Data Analysis), 206, 281
camada de link, redes, 197, 198
caminho de pesquisa, 10
caracteres de controle
  e driver de terminal, 593
  e o driver de terminal, 592
  em nomes de arquivo, 56
Card, Rémy, 86
carga média, sendmail, 426
carregadores de inicialização, 88
cavalos de Tróia, 477
ccTLDs, 266
cdebconf, 165
cdebootstrap, 165
CentOS, 5
central de dados
  monitoramento da temperatura, 551
centro de dados
  caixa de ferramentas, 552
  energia elétrica, 551
  padrões, 552
  racks, 551
  refrigeração, 550
  trilhos para cabeamento, 552
Cerf, Vint, 194
CERT, 493
Certificate Authority (CA), 505
Certificate Signing Request (CSR), 505
Certified Information Systems Auditor (CISA), 468
cf/cf, diretório, sendmail, 395
cfengine, 184
CGI, script, 500
chage, programa de expiração de senha, 472
ChaosNet, 274
Chapman, Brent, 384
chat, 228
chat scripts, PPP, 227
Chatsworth Products, 552
chattr, 62
chaves, gerando BIND, 321
chaves, SSH, 483
checksendmail, script, 428
chfn, 70
chgrp, 62
chkconfig, 25, 28, 363
chmod, 58, 61, 64
chown, 62
chroot
  para FTP, 508
  para named, 316, 317
  para o Postfix, 434
  para sendmail, 422
chsh, 70, 71
ci, check-in RCS, 176, 177
CIDR (Classless Inter-Domain Routing), 201, 202, 203, 204
CIFS. *Ver* Samba
cilindros, disco, 86
CIM (Common Information Module)
  configuração de sistema, 185
CISA (Certified Information Systems Auditor), 468
Cisco routers, 243
Cisco, roteadores, 244, 486, 494
CiscoWorks, 463
CISSP (Certified Information Systems Security Professional), 468
classes de agendamento, 40
clientes sem disco, 166, 167, 619
clocks, sincronização, 621
clone, chamada de sistema, 41
closelog routine, 157
closelog, rotina, 158
CNAME DNS, registros, 279
co, RCS check out, 176, 178
COBIT (Control Objectives for Information e technologias relacionadas), 660
comandos
  agendamento, 107, 111
  de zona, 284
  localizando, 10
compactação de dados, modems, 595
compartilhamento (Samba), 570
comprando hardware, 540, 543, 630, 631
Computer Systems Research Group (CSRG), 4
concentradores. *Ver* hubs Ethernet
conectando-se a partir do Windows, 566
conectores. *Ver também* cabos
  adaptadores RS-232 para USB, 596
  DB-25, 581, 583
  DB-9, 584
  IDE, 81
  mini DIN-8, 584
  RJ-11, 594
  RJ-45, 585
  RJ-45 fêmea-fêmea, 586
  SCSI, 82, 84
conectores seriais. *Ver* conectores seriais
confCOPY_ERRORS_TO, opção, sendmail, 396
.config, arquivo de configuração para o kernel, 605
.config, arquivo, para o kernel, 603
configuração de rede, 210, 219
  Debian e Ubuntu, 218
  Red Hat e Fedora, 217
  SUSE, 218
configuração de sistema, 181, 185, *Ver também* administração de sistema; Linux, instalação; hardware
  Arusha Project, 185
  cfengine, 184
  CIM (Common Information Module), 185
  gerenciamento, 184, 185
  LCFG (sistema de configuração em larga escala), 184
  Template Tree 2, 185
congestionamentos de transmissões, 212
ConnectionRateThrottle, opção, sendmail, 423
console, efetuando o loging a, 157
CONT, sinal, 42, 43, 44, 49
conta root, 33, 472
contas de usuário
  adicionando, 73, 77, 78
  aliases, globais (e-mail), 68
  arquivos de inicialização, 75
  autenticação sob o Samba, 573
  compartilhadas, 472
  convidados, 649
  daemon, 37
  diretórios iniciais, 54, 70, 75
  excluindo, 78
  ftp, 508
  gerenciamento por toda a instalação, 649
  higiene, 67
  informações GECOS, 70
  máquina do diretório inicial de e-mails, 76
  modificando, 78
  nobody (NFS), 37, 341
  nomes, 68
  número de ID. *Ver* IDs de usuário
  processo de login, 33
  pseudo-usuários, 37
  removendo, 77
  root. *Ver* root, conta
  senhas, 74
  shell de login, 70
  superusuário. *Ver* root, conta
  uso de sendmail, 419
contato com os autores, xiv
controle de fluxo, linha serial, 587, 588
controle de revisão, 175, 181
  CVS, 178, 179
  RCS, 176, 178
  Subversion, 179, 181
controls, instrução, DNS, 305
convenção para atribuição de nomes
  sistemas de arquivos compartilhados, 341
convenções de nomeação
  arquivos de dispositivos, 601
convenções tipográficas, 7, 8
convenções utilizadas neste livro, 7, 8
cookies
  mágicos (NFS), 340
  mágicos (X Windows), 516
  NFS, 340
Cooper, Mendel, 8
Cooperative Association for Internet Data Analysis (CAIDA), 281
Corbet, Jonathan, 228
core, arquivos, 110
correio. *Ver* e-mail
correlação de eventos, 158
cotas de disco, 340
Council of European National Top-level Domain Registries, 268
cpio, 126
CPU
  analisando uso, 556, 560
  estatísticas, 557
  médias de carga, 557
  uso, 558
cracklib, 473
criptografia, 465
  algoritmo MD5, 69, 318
  chave pública, 320
  criptografia de senhas, 67, 68, 378, 571
  em DNS, 272, 318, 320, 324
  em LDAP, 367
  intercâmbio de chaves Diffie-Hellman, 319
  IPSec, 491
  no sendmail, 419, 424
  questões legais, 652
  SSL, 505
criptografia de chave pública, 320, 483
cron, daemon, 107, 111, 611
  automatizando o registro em log, 144
  comandos pulados, 111
  gerenciamento, 109
  usos comuns, 109, 111
crond. *Ver* daemon de cron
crontab, arquivos, 108, 109
crontab, comando, 109
crypt library, rotina, 471
Crypto-Gram, boletim, 494
.cshrc, arquivo, 75
CSLIP protocolo, 226
CSMA/CD (Ethernet), 247
CSR (Certificate Signing Request), 505
CSRG (Computer Systems Research Group), 4
ctime (atributo change time), 60
CTS (Clear to Send), 587
cu, 589, 595
Cubby *vs.* CompuServe, 656
CUPS, 529, 545, 616
  administração, 533, 539
  arquitetura, 529, 533
  arquivos de descrição de impressora (PPD), 532
  autoconfiguração de impressora, 534
  classes de impressora, 535
  comandos de compatibilidade, 537
  comparação com a impressão ATT/BSD, 538
  documentação, 538

exemplos de configuração, 535
filas de impressão, 530
filtros, 532, 533
HTTP e, 531
impressão em rede, 531
imprimindo um documento, 530
registro em log, 539
removendo uma impressora, 535
scripts de inicialização, 533, 539
solucionando problemas, 539, 540
utilitários de linha de comando, 538
cupsd, daemon, 531, 539
cupsdconf, 533
curingas, shell, 7
CVS, 178, 179
Cygwin, ferramentas, para o servidor X, 567
cyrus, agente de entrega, 399

D

dadm, utilitário de gerenciamento RAID, 103
daemons
configuração, 619
e-mails, 618
execução de comando remota, 618
executores de fila do sendmail, 426
impressão, 616
inicializando, 619
kernel, 615
login remoto, 618
NFS, 616, 617
rede, 620, 621
Samba, 616, 617
Dan Geer, 115
DARPA (Defense Advanced Research Project Agency), 194, 493
datasize, opção, DNS, 298
date, 145
Dawson, Terry, 8
DB, biblioteca, 390
DB-25, conectores, 581, 583
DB-9, conectores, 584
dbm/ndbm, biblioteca, 401
dc, atributo, 387
DCD (Carrier Detect), 587
DCD (Data Carrier Detected) sinal, 588
DCE (Data Communications Equipment), interface, 583
dd, 95, 126
DDS/DAT, fitas, 118
.deb, formato de pacote de software, 167
Debian, configuração de rede, 218
debian-installer, 165
DEC VT100, terminal, 591
DeCSS, programa, 653
DEFAULT_HOME, variável em login.defs, 70
DefaultUser, opção, sendmail, 381, 419
DELAY_LA, opção (sendmail), 423, 426
delegações defeituosas, DNS, 329, 333, 334
delegation-only, opção, DNS, 300
delete, caractere, 592
depurando. *Ver* solucionando problemas
Deraison, Renaud, 479
desastre
fontes de alimentação de energia, 646
planejando-se para, 116, 645
recuperação, 492, 493, 645, 648
desempenho, 554, 564
ajuste de unidades IDE, 92
ajuste do kernel para e-mail, 427
aprimorando, 554, 555
BIND, 335
cache Web Squid, 507, 508
CPU, 556, 558, 560
disco, 556, 561
discos, 562
fatores que afetam, 555, 556
flag st_atime, 63
médias de carga, 557
medindo e monitorando, 461
memória, 49, 555, 556, 560
NFS, 345
nice, 44
particionando discos para aprimoramento, 88
PPP, 226
rede, bits TOS, 232
SDSC Secure Syslog, 150
sendmail, 424, 427
sendmail, 427
servidor Web, 500, 501, 503
sincronizando arquivos de log, 153
solucionando problemas, 563, 564
utilizando RAID para aprimorar, 99
desligando o sistema, 28, 30
detecção de invasão, samhain, 480, 481
/dev, diretório, 54, 57, 599, 601
/dev/console, arquivo, 157
/dev/log, soquete, 150
/dev/pts, diretório, 54
/dev/ram0 e /dev/ram1, arquivos, 561
df, 90, 346
DFS (Distributed File System, Windows), 574
DHCP (Dynamic Host Configuration Protocol), 219, 221
BIND e, 314
configuração de cliente, 221
configuração de servidor, 220, 221
daemon, 619
endereços duplicados, 221
retrocompatibilidade, 220
dhcp.leases, arquivo, 220, 221
dhcpcd, daemon, 220, 221
dhcpd, daemon, 619
Diffie-Hellman, intercâmbio de chaves, 319
dig, 287, 304, 317, 332, 334
dig, comando, 286
Digital Millennium Copyright Act (DMCA), 653
directory, instrução, DNS, 296
diretivas
acordos, 76, 650
backups, 645, 646
daemons de diretiva Postfix, 442
documentos, 648, 652
imposição, 655
registro em log, 144
segurança, 649, 650
uso adequado, 656
diretórios, 55, 57
copiando, 125
diretórios iniciais, 54
ausentes, 70
criando, 75
localização, 54
login, 70
removendo, 77
discos. *Ver também* LVM, RAID, SATA, IDE e SCSI
ajuste de desempenho, 92
análise de E/S, 561, 562
arquivos de dispositivos para, 87
balanceamento de carga, 555, 561
carregadores de inicialização, 88
como mídia de backup, 119
conectando, 87
cotas, 340
desempenho, 556, 563
espaço de swap, 559, 561
exibindo espaço livre, 90
falha e RAID, 99
Fibre Channel, 80
formatando, 88
geometria, 85, 86
hot-swappable, 80, 83, 104
IDE, 81, 84, 92
instalando, 87, 92, 95, 98
interfaces, 80, 85
LBA (Logical Block Addressing), 81
partições, 88, 89, 96, 97
PATA. *Ver* IDE
RAID, 555
RAM, 561
realocando espaço de armazenamento, 104
removíveis, 118
rótulos, 88, 89
SCSI, 80, 82, 84
Serial ATA. *Ver* SATA
testando, 88
trilhas e setores, 86
USB, 80, 105, 106, 118
verificando, 93, 95
discos rígidos. *Ver* discos
DISPLAY, variável, 515, 517
dispositivo de interconexão, 596
dispositivos seriais, configuração de software, 589
dispositivos, pseudodispositivos, 600
Distfile, 354, 355
distribuições, Linux, 5, 661
popularidade, 661
Distributed Management Task Force(DMTF), 185
DIX Ethernet II, 197
DMA, ajustando, 93
DMCA (Digital Millennium Copyright Act), 653
dmesg, arquivo, 148
dmesg, comando, 148
DMTF (Distributed Management Task Force), 185
DNAME DNS, registros, 283
DNS, 265, 270, *Ver também* BIND; nomes de domínio, DNS; registros de recursos, DNS; servidores de nomes; zonas, DNS
adicionando uma nova máquina, 262, 263
armazenamento em cache, 270
armazenamento em cache negativo, 270
arquitetura, 290
arquitetura de servidor, 292
arquivo de configuração de servidores-raiz, 269
arquivo service.switch, 215
arquivo switch, arquivo, 216
arquivos de zonas diretas, 265
arquivos de zonas reversas, 265
atualizações dinâmicas, 220, 314, 315
banco de dados, 265, 272, 287
biblioteca de conversão, 290
configuração, 290, 293
configuração de cliente, 215, 216, 294
configuração do conversor, 293, 294
criptografia de chave pública, 320
criptografia em, 272, 318, 324
delegação, 269
delegações defeituosas, 329, 333, 334
dicas de servidor, 290
DNS dividido, 306, 307
doc (domain obscenity control), 334, 335
domínio in-addr.arpa, 277
domínio us, 266
domínios de primeiro nível, 266, 267
domínios de primeiro nível para código de países, 266
domínios de primeiro nível para códigos de países, 266
eficiência, 270
endereços IP, 262, 263, 277, 278
equilíbrio de carga, 270

espaço de nome, 265, 291
espaço de nomes, 267
ferramenta Site Finder da VeriSign, 300
gTLDs (domínios de primeiro nível genéricos), 266
gTLDs (generic top-level domains), 266
hack do CNAME, 280, 281
harmonização TTL, 273
história, 263
implementações, 264, 265
interface de shell do registro SOA, 290
internacionalização, 272
ip6.arpa para mapeamentos IPv6 reversos, 283
lookups, sendmail, 401
mapeamento direto, 265, 268
mapeamento reverso, 265, 268, 277, 283, 311
Microsoft e, 325, 326
MX, registros, 428
negação de serviço (DOS), ataques, 278
nomes de domínio. *Ver* nomes de domínio, DNS
nomes de domínio completamente qualificados, 267
notificação assíncrona de mudanças de zona, 272
número serial do registro SOA, 275
projeto, 290
protocolo, 264, 271
protocolo EDNS0, 272
pseudo-registros SPF (Sender Policy Framework), 282
Punycode, 272
recursão de consulta, 297
referrals, 289
registros de recursos. *Ver* registros de recursos, DNS
resposta negativas, 325
RFCs, 263, 264, 338
roteamento anycast, 297
round-robin, 500
segurança, 292, 297, 316, 325
serviços de registro de nomes de domínio ISP, 268
servidor escravo, 289
servidor mestre de nomes, 289
servidores autorizados, 289, 291
servidores de cache, 291
servidores não autorizados, 289
servidores não recursivos, 289
servidores recursivos, 289
servidores stub, 289
spam, eliminando, 282
subdomínios, 269
suporte IPv6, 283, 284
teste de conversores, 294
tipos de registros, 274
TKEY, 318, 320
traceroute e, 451
TSIG (assinaturas de transação), 311, 318, 320
uso com CIDR, 280, 281
valores de tempo limite do registro SOA, 276
zonas de stub, 285
zonas stub, 286
ZSK (zone-signing keys), 320
DNS A6, registros, 283
DNS dividido, 306, 308, 311
DNSKEY DNS, registros, 320
DNSmapeamento reverso, 278
DNSSEC, 272, 320, 324
dnssec-keygen, 318, 321
dnssec-signzone, 322
doc (domain obscenity control), DNS, 334, 335
documentação
fontes, 8
Linux, 8, 10
local, 12, 640, 642
usuário, 642
Doering, Gert, 590
domain, diretiva (DNS), 294
DOMAIN, macro (sendmail), 398
domainname, comando, 363
domínio broadcast, 248
domínios de primeiro nível, 266
para código de países, 266
domínios genéricos de primeiro nível, 266
domínios virtuais, Postfix, 437, 439
domínios, configurando NIS, 361, 363
DontBlameSendmail, opção, 419, 421
DOS (denial of service), 422
DOS (denial of service), ataques, 423, *Ver* negação de serviço, ataques
DOS (Denial Of Service), ataques, 278
dot, arquivos, 75, 76
DoubleClick.net, 654
dpkg, 169
drivers. *Ver* drivers de dispositivo
diretório, árvore-fonte do kernel, 605
drivers de dispositivo, 57, 598, 599
adicionando ao kernel, 605, 606
conexão automática, 607, 608
detecção de dispositivos, 606
impressora, 528
impressora Windows, 577, 578
MODULE_DEVICE_TABLE, macro, 606
módulos carregáveis, 606, 607
números, 599
números de dispositivos, 600
seriais, 601
terminal e caracteres de controle, 592, 593
drivers seriais
caracteres especiais, 592, 594
opções de configuração, 593
DS DNS, registros, 320, 322
DSL, redes, 256
DSR (data set ready), sinal, 587
DSR (Data Set Ready), sinal, 588
DTE (Data Terminal Equipment), interface, 582, 583
DTR (Data Terminal Ready), sinal, 588
dump, 87, 120, 123
dumpdates, arquivo, 121
dumps. *Ver* backups
duplex, configurando interface. *Ver* mii-tool
DVD Copy Control Association, 653
DVMRP, protocolo, 241

E

E/S, agendadores, 562
e2label, 91
EDITOR, variável de ambiente, 74
EFF (Electronic Frontier Foundation), 658, 666
EIGRP, protocolo, 238, 240
eletricidade estática, 547
.emacs, arquivo, 75
e-mail. *Ver também* sendmail; Postfix; MX DNS, registros
adulteração, 424
agente de envio de e-mail (MSA), 372
agentes de acesso, 372
agentes de entrega, 371
agentes de entrega (MSA), 372
agentes de transporte, 371
agentes de usuário, 370
aliases. *Ver* aliases, e-mail
armazenamentos de mensagens, 371
arquitetura, 376, 378
ataques de negação de serviço (denial of service – DOS), 423
ataques de negação de serviço (DOS), 422
cabeçalhos, 372, 375
caixa de correio inicial, 76
caixa postal inicial, 377
clientes, 370
códigos do status de entrega, 412
comercial não-solicitado, 368
componentes, 370, 372
componentes de sistema, 370, 372
configuração de servidor, 376, 377, 427
diretório de fila, 393
diretório de filas, 392
encaminhamento, 382
encaminhando, 383
endereços, 68, 373
envelope, 372
estrutura da mensagem, 373
estrutura das mensagens, 372
Exim, 431, 433
falsificação, 423
lista de discussão, 384
lista negra, 413
listas de discussão, 386
listas negras, 413
loops, 380, 383
mensagens não-entregáveis, 426
mensagens ricocheteadas, 396
MX, reserva (fallback), 426
para arquivos, 381
para programas, 381
para uma conta desativada, 77
privacidade, 424
projeto do sistema, 376, 378
protocolo IMAP, 372, 378
protocolo POP, 372, 378
proxies, 376
retransmissão, 409, 411
RFCs, 444
SASL, 424
segurança, 381, 419, 424
servidores de backup, ISP, 377
spam. *Ver* spam
tarefas de administração de sistemas, 369
e-mails
daemons, 618
emergência. *Ver* desastre
empilhadores, mídia de fita, 119
ENC TCP, opção, 216
endereços
questões de gerenciamento, 260
endereços de e-mail, 68, 373
endereços IP. *Ver* IP, endereços
endereços, Ethernet (conhecidos como MAC), 198, 207
endereços, SCSI, 84
Engarde Linux, 492
enscript, 537, 538
entrega prolixa, 428
entregadores
locais, 399
equilíbrio de carga
DNS, 270
equipamentos
racks, 551
ESMTP, protocolo, 371, 429
espaço de nome, DNS, 265
espaço de swap, 92, 98, 559, 561
espaço livre, exibindo, 90
estações de trabalho, sem disco, 619
estado de link, protocolos de roteamento, 238
estatísticas
arquivo para e-mail (sendmail), 427
BIND, 331
CPU, 557
desempenho, 562
informando, 562
NFS, 347

rede, 451
redes, 454
sendmail, 427
/etc, diretório, 54
/etc/apache2, diretório, 501
/etc/apt/sources.list, arquivo, 172, 173
/etc/auto.master, arquivo, 348
/etc/bootparams, arquivo, 619
/etc/cron.allow, arquivo, 109
/etc/cron.deny, arquivo, 109
/etc/cups, diretório, 533
/etc/cups/cupsd.conf, arquivo, 531, 533, 539
/etc/cups/mime.convs, arquivo, 533
/etc/cups/mime.types, arquivo, 533
/etc/cups/ppd, arquivo, 532
/etc/dhcpd.conf, arquivo, 220, 221
/etc/dumpdates, arquivo, 121
/etc/exports, arquivo, 342, 344
/etc/fstab, arquivo, 90, 92, 94, 98, 346, 348, 576
/etc/ftpusers, arquivo, 509
/etc/gateways, arquivo, 241
/etc/group, arquivo, 72, 73
definindo grupos, 32
editando, 74
permissões, 475
/etc/group,arquivo
para servidores FTP, 508
/etc/gshadow, arquivo, 73
/etc/host, arquivo, 211
/etc/host.conf, arquivo, 216
/etc/hostname, arquivo, 219
/etc/hosts, arquivo, 199, 211, 294
/etc/hosts.allow, arquivo, 479, 480
/etc/hosts.deny, arquivo, 479, 480
/etc/hosts.equiv, arquivo, 475, 619
/etc/inetd.conf, arquivo, 613, 614
/etc/init.d, diretório, 24, 25, 27, 28
/etc/inittab, arquivo, 24, 27, 589, 591, 611
/etc/interfaces, arquivo, 219
/etc/issue, arquivo, 590
/etc/lilo.conf, arquivo, 20, 22, 608
/etc/login.defs, arquivo, 70, 72
/etc/logrotate.conf, arquivo, 150
/etc/logrotate.d, diretório, 150
/etc/mail/relay-domains, arquivo, 410
/etc/modprobe.conf, arquivo, 607
/etc/motd, arquivo, 590
/etc/named.conf, arquivo, 295, 312, 315, 316, 330, 336, 337
/etc/netgroup, arquivo, 361
/etc/nscd.conf, arquivo, 353, 618
/etc/nsswitch.conf, arquivo, 216, 360, 391
/etc/ntp.conf, arquivo, 622
/etc/papersize, arquivo, 537
/etc/passwd, arquivo, 67, 71, 475
editando, 69, 74
números de ID de grupo, 60
números de ID de usuário, 32, 60
para servidores FTP, 509
permissões, 475
segurança, 471, 473
/etc/ppp, diretório, 227
/etc/profile.d, diretório, 76
/etc/resolv.conf, arquivo, 216, 293, 294
/etc/rndc.conf, arquivo, 305
/etc/rndc.key, arquivo, 305
/etc/securetty, arquivo, 475
/etc/services, arquivo, 199, 486, 614, 615
/etc/shadow, arquivo, 471, 473, 590
/etc/shells, arquivo, 70, 77
/etc/sshd_config, arquivo, 484, 567
/etc/sudoers, arquivo, 35, 36
/etc/sysconfig, diretório, 26, 217, 218
/etc/sysctl.conf, arquivo, 223, 603
/etc/syslog.conf, arquivo, 146, 155, 430
/etc/syslog.conf,arquivo, 151
/etc/termcap, arquivo, 592
/etc/X11, diretório, 514
/etc/X11/xorg.conf, arquivo, 518, 521
/etc/xinetd.conf, arquivo, 612
/etc/xinetd.d, diretório, 612
/etc/yp.conf, arquivo, 357
Ethernet, 246, 252
autonegociação, 213
cabos, 197, 248, 250
cabos UTP, 248, 250, 257
colisões, 248, 451, 452
congestionamento, 250, 251, 259
DIX II, 197
domínio broadcast, 248
encapsulamento, 197
encapsulamento de pacotes, 196
endereços, 199
endereços de hardware, 207
evolução, 247
frames. *Ver* pacotes
hubs/concentradores, 250, 251
padrões de framing, 197
questões de projeto, 258, 260
roteadores, 252
solução de problemas, 256
solucionando problemas, 257
switches, 248, 251, 252
topology, 248
velocidade, 247
velocidade, configurando, 213
Ethernet, endereços, 199
Ethernet, padrões de framing, 197
EUIDs (effective user IDs), 40
EULAs (End User License Agreements), 655
EUROPEN, 663
events, daemon, 616
exec, chamada de sistema, 41
execução, bit, 62
executando programas Linux a partir do Windows, 567
executores de fila, sendmail, 426
Exim, sistema de e-mails, 431, 433
expect, 75
expect, comando, 244
EXPN, comando (SMTP), 409
exportfs, 37, 344
EXPOSED_USER, macro (sendmail), 404, 406
ext2fs, sistemas de arquivos, 62, 86
ext3fs, sistemas de arquivos, 62, 86, 89

F

FAI, 165
FAT, sistemas de arquivos, 86
fator de bloqueio, fita, 126
fax, agente de entrega, 399
FC-AL (Fibre Channel Arbitrated Loop), 80
fcntl, chamada de sistema, 340
fcron, 111
FDDI, redes, 253, 254
fdisk, 96, 97, 100, 101
FEATURE, macro (sendmail), 399, 407
Fedora, configuração de rede, 217
ferramentas de análise de desempenho
iostat, 561
oprofile, 563
sar, 562
ferramentas de pacotes de softwares
alien, 167
alto nível, 169, 175
APT, 171, 174
apt-ftparchive, 173
apt-get, 171, 174
apt-proxy, 173
arquivo /etc/apt/sources.list, 172
dpkg, 169
/etc/apt/sources.list, arquivo, 173
RHN (Red Hat Network), 171
rpm, 168
yum, 174
ferramentas, hardware, 552
Fibre Channel, 80
file, instrução, DNS, 304
filtros, CUPS, 532, 533
find, 52, 110
finger, 70
FireWall-1, 224
firewalls, 486, 488, 491
baseados em host, 224
bloqueando ICMP, 448, 449, 450
filtrando pacotes, 486
informações de estado, 487
Netfilter, 488, 491
proxy, 487
proxy de serviços, 487
tabelas de IP Linux, 488, 491
traceroute e, 450
fitas, backup. *Ver também* mídia, backup
4 mm, 118
8 mm, 118
AIT, 118
arquivos de dispositivos, 121
biblioteca, robótico, 127
copiando, 126
DDS/DAT, 118
DLT/S-DLT, 118
empilhadores, 119
fator de bloqueio, 126
LTO, 119
posicionando, 127
SAIT, 118
VXA/VXA-X, 119
flock, chamada de sistema, 340
Fogel, Karl, 179
fontes de alimentação de energia, emergência, 646
Foomatic, banco de dados, 532, 540
fork, chamada de sistema, 41
formatando discos, 88
formato de pacote de software RPM, 167
fornecedores recomendados, 260, 261
.forward, arquivo, e-mail, 382, 383, 421
forwarders, opção, DNS, 299
fragmentação, IP, 198
frame relay, redes, 255
frames. *Ver* pacotes
Frampton, Steve, 8
Free Software Foundation (FSF), 663
fsck, 17, 91, 93, 95, 98
FSUID, parâmetro de processo, 40
ftp, 620
FTP
anônimo. *Ver* FTP
chrooted, 508
configuração de servidor, 508, 509
e HTTP, comparação, 508
permissões, 509
por firewalls, 487
por meio de firewalls, 487
segurança, 475, 509
ftpd, daemon, 508, 620
functions, script, 149
fuser, 53
FYI, documentos, 195

G

garantias, 547
gated, daemon de roteamento, 241, 620
GDI, impressoras, 540
gdm, 514
GECOS informações, 70
$GENERATE, diretiva, DNS, 281, 284
genericstable, recurso do sendmail, 402
gerenciador de vídeo X, 514, 515
gerenciamento, 624, 630
gerenciamento de energia, 551
gerenciamento de pacotes, 167, 175
alien, ferramenta de conversão, 167

automatizando, 173, 174
dpkg/APT, 167, 169, 171, 174
formato .deb, 167
formato RPM, 168
Red Hat Network, 171
repositórios, 170
rpm/yum, 167, 174, 175
gerenciamento de pessoal, 625, 627
gerenciamento e pacotes
rpm/yum, 170
gerenciamento lógico de volume. *Ver* LVM
getfacl, 64
gethostbyname, rotina, 290, 360
gethostent, rotina, 564
getty, 18
getty, processo, 18, 589, 591
gettydefs, arquivo, 591
Ghostscript, 538
Ghostview, 542
GIAC (Global Information Assurance Certification), 468
GIDs. *Ver* IDs de grupo
globbing, shell, 56
GNOME, 524, 525, *Ver também* X Window System
GNU
Free Software Foundation (FSF), 663
Openwall GNU/*/Linux (Owl), 492
pacote de roteamento Zebra, 242
Public License (GPL), 661
Stow, 187
gravação, bit, 62
greet_pause, recurso, sendmail, 415
grilagem, domínio, 267
grub, 98
GRUB, rotina de inicialização, 19, 20
configuração de inicialização múltipla, 21
modo monousuário, 23
opções, 608
grub.conf, arquivo, 19, 21, 608
grub-install, 19
grupo de administração de sistemas responsabilidades, 624, 630
administração, 630, 632
desenvolvimento, 632, 636
operações, 636, 637
suporte, 637, 640
grupos. *Ver também* /etc/group, arquivo
atributo de arquivo, 60
de um processo, 40
efetivos, 40
números (GIDs), 32
padrão, 70
senhas para, 73
grupos de fila, sendmail, 425
grupos de rede, NIS, 361
gv, 542

H

hald, 620
halt, 30
Hamilton, Bruce, 10
hardware. *Ver também* discos; Ethernet; PC, hardware; manutenção; redes
adaptação de kernel, 598
alimentação de energia, 551
ambiente, 549, 551
aposentando, 546
ar-condicionado, 549
cabeamento, 257, 642
comprando, 540, 543, 630, 631
eletricidade estática, 547
ferramentas, 552
fornecedores. *Ver* fornecedores recomendados
garantias, 547
hubs, 250
investigação, 16
logs, 546
memória, 16, 548, 555
monitoramento da temperatura, 551
monitores de computador, 548
racks para equipamentos, 551
requisitos do BIND, 294
roteadores, 252
rotulando, 641
sistemas de refrigeração, 646
switches, 248, 251, 252, 253
USB. *Ver* USB
wiring, 258
Hayes, linguagem de comandos, 595
Hazel, Philip, 431
hdparm, 92, 93
Hesiod, 274
/home, partição, 89
hospedagem Web, 498, 508
Apache, 501, 507
arquivos de log, 503
balanceamento de carga, 500, 501
cache do Squid, 507
cache Squid, 508
certificados, 505, 507
configuração do Apache, 502, 507
conteúdo estático, 503
desempenho, 500, 501
equilíbrio de carga, 270
httpd, 621
IIS (Windows), 570
instalação do Apache, 501, 502
interfaces virtuais, 503, 505
script CGI, 500
servidor de cache, 507, 508
servidor proxy, 507, 508
SSL, 505, 507
TUX, 503
host, 332
hostname, comando, 211, 244
hosts virtuais, web, 503, 505
hot-swappable, unidades, 80, 83, 104
Hotz, Steve, 332
HTTP
CUPS e, 531
protocolo, 498, 499
servidor. *Ver* hospedagem Web
httpd. *Ver* hospedagem Web
httpd.conf, arquivo, 502, 507
hubs, Ethernet, 250
HUP, sinal, 42, 43
hwconf,, arquivo, 27
HylaFAX, pacote, 399

I

ICANN (Internet Corporation for Assigned Names and Numbers), 194, 204, 205, 260, 268
ICMP, 195
bloqueio de firewall, 448, 449, 450
números de seqüência, 449
pacotes, 490
ping e, 448
redirecionamentos, 208, 223
saída de netstat, 454
tracroute e, 450
TTL e, 450
IDE, 80, 82
ajuste de desempenho, 92
alterando parâmetros de disco, 92, 93
DMA, ajustando, 93
história, 81
nomes do dispositivo, 601
IDENT, protocolo, 423
IDs de grupo, 32, *Ver também* /etc/group, arquivo
globalmente únicos, 73
kernel e, 74
na saída ls, 60
números, 32
real, efetivo e salvo, 32
salvos, 40
IDs de usuário, 32, 40, 69, 74
IDs de usuário efetivos (EUIDs), 40
IEEE 802.* padrões, 197
IEEE 802.*, padrões, 247
IEEE 802.11g, padrões, 252
IEEE 802.1a, padrões, 252
IEEE 802.3, padrões, 250
IETF (Internet Engineering Task Force), 194
ifconfig, 211, 213
adicionando regras com, 236
adicionando rotas, 214
comando, 226
endereços virtuais e, 504
máscaras de subrede, 200
ifdown, 217, 219
ifup, 28, 217, 219
ifup, comando, 230
IGMP (Internet Group Management Protocol), 199
IGRP (Interior Gateway Routing Protocol), 238, 240
IMAP (Internet Message Access Protocol), 372, 378, 618
imapd, 618
impressão. *Ver também* CUPS; impressão; impressoras
arquivos de descrição de impressora (PPD), 532
banco de dados Foomatic, 540
compartilhando impressoras utilizando o Samba, 576, 578
daemons, 616
histórico, 526
instalação do driver a partir do Windows, 577
instalação do driver Windows, 578
KDE Print Manager, 533
páginas banner, 541
sob o KDE, 545
tamanhos de papel, 536
visualizadores, 542
impressoras
arquivos de descrição de impressora (PPD), 532
cartuchos, 542, 543
comprando, 540, 543
contabilidade, 543
drivers, 528
linguagens, 527, 529
rede, 534, 541
segurança, 543
seleção, 540, 542
seriais e paralelas, 541
USB, 534, 539
WinPrinters, 540
imprimindo
banco de dados Foomatic, 532
Konqueror e, 545
no KDE, 544
projeto Gutenprint, 532
software, 538
utilizando kprinter, 544
XHTML, 528
in.fingerd, 621
in.rlogind, 619
in.rshd, 619
in.telnetd, 619
in.tftpd, 619
in-addr.arpa, domínio, 277
INCLUDE, diretiva (DNS) $, 284
include, instrução, DNS, 296
:include:, diretiva para aliases e-mail, 380
inetd, 612, 613, 615
desenvolvimento de, 610

inicialização, 15, 18
automática *vs.* manual, 15
diretamente para o bash, 26
diretório /etc/sysconfig, 26
do kernel, 16
dual, 21, 22, 569
fsck e, 17
investigação de dispositivos, 16
modo monousuário, 15, 17, 22, 23, 24
montando sistemas de arquivos NFS, 346
múltipla, 21, 22
níveis de execução, 23
opções, 608
opções do kernel, 21
questões específicas do PC, 18
scripts
/etc/init.d diretório, 24
scripts de inicialização, 23, 28
threads do kernel, 16
verificação do sistema de arquivos, 94
via rede, 161, 162
init, 18
init, processos, 15, 41, 589, 591, 610, 611
logins e, 610
níveis de execução e, 23, 25, 30, 611
processos zumbis e, 41, 44
scripts de inicialização e, 23, 27, 28
initlog, 148
inodes, 60, 89
insmod, 606, 607
inspeção de informações de estado, firewalls, 487
instalação de sistema
netbooting, 162
instalação, Linux. *Ver* Linux, instalação
instrução controls, DNS, 306
INT, sinal, 42, 43
interfaces
DCE (Data Communications Equipment), 582
rede. *Ver* redes
seriais, DCE *vs.* DTE, 583
serial, DCE *vs.* DTE, 582
International Organization for Standardization (ISO), 249
Internet
conexões discadas. *Ver* PPP, protocolo
controle, 194, 195
história, 193, 194
Internet Control Message Protocol. *Ver* ICMP
Network Information Center (InterNIC), 204
Official Protocol Standards, 195
padrões e documentação, 194
padrões e documentos, 195
protocolo. *Ver* IP
protocolo, classes de endereços, 200
registros, 204
segurança de protocolo (IPSec), 491
série RFCs, 194
Systems Consortium (ISC), 8
Internet Corporation for Assigned Names and Numbers (ICANN), 205
Internet Systems Consortium (ISC), 220
InterNIC (Internet Network Information Center), 204
IOS (Cisco router OS), 244
IOS (OS de roteadores), 243
iostat, 561
IP, 195, *Ver também* IP, endereços; IPv6; roteamento
bits TOS (type-of-service), 232
broadcast direcionado, 223
disfarçando. *Ver* NAT
encaminhamento de pacotes, 214, 223
falsificação, 224
fragmentação, 198
roteamento na origem, 223
IP, endereços, 198, 207, *Ver também* IPv6
alocação, 204, 205
broadcast, 199
CIDR (Classless Inter-Domain Routing), 201, 202, 203, 204
classes, 200
escassez de, 202
interface loopback, 200, 208, 213, 278
máscaras de rede, 200, 202
multicast, 199, 200
nomes de host e, 199, 211
portas, 199
PPP, 227
privados, 205, 206, 286, 291, 306, 325
sub-rede, 200, 202
unicast, 207
ipcalc, 201
IPSec (Internet Protocol security), 491
IPsec, protocolo, 224, 652
iptables, 225, 488, 491
IPv6, 202, 203, 206, 207
suporte DNS, 272, 283, 284
vs. CIDR, 202
IPv6, endereços, 203
ISC (Internet Systems Consortium), 220, 264
ISDN, redes, 255
IS-IS, protocolo, 238, 241
ISO (International Organization for Standardization), 249
ISOC (Internet Society), 194
ISPs
alocação de endereços IP, 204, 207
AOL, 656
serviços de registro de nomes de domínio, 268
ISS, servidor Web, 570
ISSO/IEC 17799, padrão, 469
ITIL (Information Technology Interface Library), 659
!IXGEN_DONT_EDIT_THIS!, 248

J

Jacobson, Van, 194, 226, 232, 449, 455
JFS, sistema de arquivos, 87
John the Ripper, 479
jukeboxes, mídia de fita, 119

K

kacpid, daemon, 616
Kahn, Bob, 194
Kalt, Chrisophe, 187
kblockd, daemon, 616
KDE, 524, 525, *Ver também* X Window System
impressão sob, 545
imprimindo no, 544
Konqueror, 545
Print Manager, 533
kdm, 514
Kerberos, 325, 482
kermit, 595
kernel, 598, 599
ajustando, 221, 223, 427, 602, 603
ARP, cache, 209
arquivo .config, personalizando, 603, 605
árvore de código-fonte, 603
árvore de origem, 603
árvore-fonte, 605
classificação de pacotes baseada em TOS, 232
configuração, 601, 603
construindo, 603, 605
daemons, 615
diretório, 603
drivers de dispositivo, 57, 598, 599
IDs de grupo salvos, 40
inicialização, 16
lista negra para conexões automáticas, 608
localização, 54
módulos carregáveis, 606, 607
módulos de conexão automática, 607, 608
opções, 602, 603, 604
opções de tempo de inicialização, 21, 608
pânicos, 94, 95
parâmetro swappiness, 559
registro em log, 148, 149
threads, 16
variáveis de segurança de rede, 225
KEY DNS, registros, 318, 320
.key, arquivo DNSSEC, DNS, 318
key, instrução, DNS, 301
kghostview, 544
Kickstart, 162, 164
kill, 43
kill, comando, 563
KILL, sinal, 42, 43, 44
killall, 44, 145
Kim, Gene, 428
Kirch, Olaf, 8
kjournald, daemon, 616
klogd, daemon, 148
Knoppix, 5, 165
Kolstad, Rob, 428
Konqueror, 545
Kotsikonas, Anastasios, 385
kprinter, 544
Kristensen, Peter, 187
ks.cfg, arquivo, 162, 164
ksoftirqd, daemon, 616
kswapd, daemon, 558, 616
Kudzu, 27

L

lâminas, discos, 85
LAMP, plataforma, 498
LANs
ATM, 254
casos para, 246
Ethernet, 246, 252
FDDI, 253, 254
lastlog, arquivo, 148
LBA (Logical Block Addressing), 81
LCFG (sistema de configuração em larga escala), 184
LDAP (Lightweight Directory Access Protocol), 363, 367
configuração, 365
documentação, 365
estrutura de dados, 364
IDs de usuário e, 69
nomes de atributo, 364
OpenLDAP, 365
segurança, 367
setup, 366
uso com sendmail, 381, 386, 388, 403, 404
usos de, 364
utilizações de, 365
ldap_routing, recurso, sendmail, 387, 403, 404
LDP (Linux Documentation Project), 8
Leffler, Sam, 399
leitura, bit, 62
Libes, Don, 75
licenças, software, 656
liderança, 625
lilo, 20, 22, 98
LILO, rotina de inicialização, 20, 21
configuração de inicialização múltipla, 22
modo monousuário, 23
LILO, rotina de partida

configurando, 608
limit, 564
linha serial
controle de fluxo, 587, 588
depurando, 596
links
físicos, 57, 58, 60
links físicos (hard links), 57
links simbólicos, 56, 58
Linux
console, 592
cultura, 660, 661
distribuições, 5, 661
documentação, 8, 10
Documentation Project (LDP), 8
falhas de segurança, 465
história, 4
International (LI), 662
listas de discussão, 664
padrões, 658, 659
popularidade, 661
Professional Institute (LPI), 664
recursos, 662, 665
Test Project, 661
vs. UNIX, 4
Linux aprimorado, 492
Linux Professional Institute, 662
Linux Standard Base (LSB), 659
Linux, instalação, 160, 166, *Ver também* administração de sistemas; configuração de sistema
arquivo k.cfg, 162, 164
automatizando a partir de um sistema mestre, 165
automatizando com AutoYaST, 164, 165
automatizando com cdebconf, 165
automatizando com cdebootstrap, 165
automatizando com debian-installer, 165
automatizando com FAI, 165
automatizando com Kickstart, 162, 164
automatizando com system-config-kickstart, 165
inicialização via rede, 161, 162
protocolo PXE, 161, 162
protocolo TFTP, 161
PXELINUX, 161
system-config-netboot, 161
LinuxWorld Expo, conferência, 664
lista de discussão, 380, 664
lista de discussão, software, 386
listas cinzas para spam, 442
listas de buracos negros, spam, 416
listas de concordâncias de endereços, BIND, 295
listas de controle de acesso. *Ver* ACLs
listas de discussão, 384, 386
listas de discussão, software, 384
listmanager, 386
ListProc, 385
LISTSERV Lite, 386
Mailman, 385
SmartList, 385
listas de verificação, administração de sistemas, 648
listas negras, spam, 441
listas telefônicas, 621
listen-on, opção, DNS, 298
listmanager, 386
ListProc, 385
LISTSERV Lite, 386
Liu, Cricket, 296
LMTP, protocolo, 434
ln, 57, 58
LOC DNS, registros, 281
LOCAL_*, macros, sendmail, 407
local_lmtp, recurso, sendmail, 407
local_procmail, recurso, sendmail, 407
localhost, 200
localhost, zona, exemplo de configuração, BIND, 307
local-host-names, arquivo, 400
locate, 11, 532
lockd, 617
lockd, daemon, 340
lockf, chamada de sistema, 340
log, registro em
para sudo, 35
logcheck, 158
logger, 156, 157
logging, instrução, DNS, 302, 327
login. *Ver* contas de usuário
.login, arquivo, 75
login, comando, 33, 590
login, processo, 589
logins
grupos, 70
logrotate, 149, 150
logwatch, 158
loopback
endereço, BIND, 305
interface, 200, 208, 213, 278
sistema de arquivos, 53
LOPSA, 663
lost+found, diretórios, 90, 95
lpd, daemon, 616
lpd-errs, arquivo, 154
lpinfo, 533
ls, 32, 56, 58, 59, 61
lsattr, 62
lsmod, 606
lsof, 54, 346
LTO, fitas de backup, 119
lvcreate, 103
lvextend, 104
LVM, 99, 102, 105
criando, 102, 103
redimensionando, 104, 105
lwresd, 618

M

m4, 394, 396, 407
MAC, endereço, 207
MAC, endereços, 198
Mackerras, Paul, 355
macros, sendmail, 396, 399
mail.local, agente de entrega, 407, 421
mail.local, agente entrega, 371
MAIL_HUB, macro (sendmail), 405, 417
MAILER, macro (sendmail), 399
mailertable, recurso (sendmail), 402
mailq, comando, 430
.mailrc, arquivo, 75
mailstats, 427
main.cf, arquivo, 434, 435
Majordomo, 384, 385, 420
makedbm, comando, 358
MAKEDEV, script, 57, 600
makemap, 401, 402
Manheimer, Ken, 385
MANs, 246
manutenção, 546, 552, 553, *Ver também* hardware
ambiente, 549
contratos, 546, 547
energia elétrica, 551
preventiva, 548, 549
racks de equipamentos, 551
many-answers, opção, DNS, 297
mapas executáveis, automontador NFS, 349
mapeamento direto, DNS, 268
mapeamento reverso, DNS, 268, 277, 311
máquinas multiprocessadas, analisando desempenho, 557
mascaramento, sendmail, 404, 406
máscaras de rede, 200, 202
máscaras de sub-rede. *Ver* redes, máscaras de rede
máscaras em ACLs, 65
MASQUERADE_AS, macro (sendmail), 404, 405, 428
master.cf, arquivof, 433
masters, instrução, DNS, 302, 304
match-clients, cláusula, DNS, 307
max-cache-size, opção, DNS, 298
MaxDaemonChildren, opção, sendmail, 422
MaxMessageSize, opção, sendmail, 422
MaxRcptsPerMessage, opção, sendmail, 423
MBR (master boot record), 18
McKusick, Kirk, 86
MDA (mail delivery agent), 371
mdadm, utilitário de gerenciamento do RAID, 101
mdadm, utilitário de gerenciamento RAID, 102
mdrecoveryd, daemon, 616
médias de carga, 557
memória
analisando uso, 560
armazenando em buffer, 28
discos de RAM, 561
efeito sobre desempenho, 555, 556
gerenciamento, 558, 559
inicialização do kernel e, 16
módulos, 548
paginação, 558, 561, 563
virtual (VM), 559
memória virtual (virtual memory – VM), 89, 92, 558, 559
memória, sistemas de arquivos baseados em, 561
mensagem do dia, 590, 655
mensagens não-entregáveis, sendmail, 426
Merit GateD Consortium, 242
Metcalfe, Bob, 246
mgetty, processo, 589, 591
Microsoft Windows. *Ver* Windows
mídia, backup, 116, *Ver também* fitas
CD e DVD, 117
jukeboxes, 119
magnético, 117
óptica, 117
resumo dos tipos, 119
rotulando, 113
vida de, 116
mídias, backup, 120
verificando, 115
mii-tool, 213
Miller, Todd, 35
miltering, sendmail, 415
MIME (Multipurpose Internet Mail Extensions), 370, 418
Minar, Nelson, 621
mingetty, processo, 589, 591
mini DIN-8, conectores, 584
minicom, 595
mkdir, 57
mke2fs, 89, 97, 98
mkfs, 50
mknod, 57, 58, 600
mkpasswd, 75
MKS Toolkit, 569
mkswap, 98
Mockapetris, Paul, 263
model, arquivo, 532
modem nulo
cabos seriais, 583
modems, 587, 594, 595
modems a cabo, 256
modems bidirecionais, 595
modems de discagem, 589
modo monousuário
entrando, 22
evitando, 17

inicialização manual, 15, 17
inicializando para, 23
remontando o sistema de arquivos raiz, 17, 22
modo multiusuário, 18
modprobe, comando, 607
MODULE_DEVICE_TABLE, macro, 606
módulos carregáveis, 607
módulos de kernel de conexão automática, 607, 608
Mondo Rescue, 140
monitoramento do ambiente, 551
monitorando a integridade, 480
monitorando arquivos de log, 158, 159
monitores de computador, 548
monousuário, modo
inicializando no, 17
montagem
sistemas de arquivos NFS, 344
montando
sistemas de arquivos. *Ver* sistemas de arquivos, montando
Moore, lei de, 194
Morris, Robert, Jr., 465
MOSPF, protocolo, 241
motd, arquivo, 655
Motion Picture Association of America, 653
mount, 53, 90, 91
ativando ACLs de sistema de arquivos, 63
sistemas de arquivos NFS, 346
mount.smbfs, 576
mountd daemon, 342
mpstat, 557
mreport program, 431
MRTG (Multi-Router Traffic Grapher), 461
MSA (mail submission agent), 372
mt, 127
MTA (mail transport agent), 371
MTU (Maximum Transfer Unit), 198, 253
mtx, 127
MUA (mail user agent), 370
multicast, endereços, 199, 200
multiprocessamento simétrico (symmetric multiprocessing – SMP), 557
Multipurpose Internet Mail Extensions (MIME), 370, 418
MX DNS, registros, 278, 279, 428
MX, reserva (fallback), e-mail, 426
MySQL, 127, 129, 264, 498

N

Nagios SNMP, ferramenta de monitoramento, 461
named, 288, 313, *Ver também* BIND; DNS; servidores de nomes
ACLs, 300, 316, 317
arquivo /etc/named.conf, 295, 315, 316, 337
arquivo named.run, 330
arquivo root.cache, 304
atualizando arquivos de zona, 313
atualizando arquivos de zonas, 315
chrooted, 316
cláusula allow-update, 303, 315
cláusula match-clients, 307
cláusula update-policy, 315
compilando com OpenSSL, 321
configuração, 294, 312
confinando com chroot, 317
controls, instrução, 306
depurando, 326, 335
dicas do servidor-raiz, 304
diretiva $TTL, 273, 276
diretiva bogus, 302
diretiva domain, 294
diretiva search, 293
em chroot, 317
/etc/named.conf, arquivo, 312, 330, 336
exemplo de configuração de zona host local, 307
exemplo de configuração ISC, 311
exemplos de configuração, 307, 312
iniciando, 314
instrução acl, 300
instrução controls, 305
instrução directory, 296
instrução include, 296
instrução key, 301
instrução logging, 302, 327
instrução masters, 302, 304
instrução options, 296, 300
instrução rrset-order, 299
instrução server, 301, 314
instrução topology, 299
instrução trusted-keys, 301
instrução view, 307
instrução zone, 302
instrução, arquivo, 304
instruções, lista de, 295
interface de linha de comando. *Ver* named, rndc
mensagens de erro, 329
named-checkconf, 319, 335
named-checkzone, 295, 335
números seriais de zona, 313
opção allow-recursion, 297
opção avoid-v4-udp-ports, 299
opção blackhole, 299
opção datasize, 298
opção delegation-only, 271, 300
opção forwarders, 299
opção listen-on, 298
opção many-answers, 297
opção max-cache-size, 298
opção notify, 297
opção provide-ixfr, 314
opção query-source, 298
opção recursion, 297
opção recursive-clients, 298
opção request-ixfr, 314
opção sortlist, 299
opção transfers-in, 298
opção transfer-source, 298
opção transfers-out, 298
opção transfers-per-ns, 298
opção zone-statistics, 303
opções TTL, 299
registrando em log, 287, 313, 327, 330
requisitos de hardware, 294
rndc, 305, 306, 313, 330, 331
script keep-running, 292
scripts init, 313
servidor escravo, configurando, 303
servidor mestre, configurando, 303
testando, 326, 335
transfer-source, opção, 311
versões, 288, 297
zona direta, configurando, 305
zonas stub, configurando, 303
named.run, arquivo, 330
named_dump.db, arquivo, 331
named-checkconf, 295, 319, 335
named-checkzone, 295, 335
nanny, script, 313
NAT, 205, 206, 225
National Science Foundation (NSF), 267
navegadores, Web, 498
ncftp, comando, 356
negação de serviço (denial of service – DOS), ataques, 422
negação de serviço (DOS), ataques, 278, 423
negação de serviço, ataques, 357
neigh, diretório, 222
Nessus, 479
netBIOS, 570
NetBIOS, 617
Netfilter, 488, 491
NeTraverse, 569
NET-SNMP, 459, 461
netstat, 451, 454
e NFS UDP, overflows, 344
estatísticas de rede, 451, 454
examinando a tabela de roteamento, 208
exemplos, 236, 237
interfaces, 451
monitorando conexões, 452
portas abertas, 453
tabela de roteamento, 453
netstat, comando
examinando interfaces, 212
Network Appliance, Inc., 347
Network Information Service. *Ver* NIS
Network Solutions, Inc., 267
Network Time Protocol (NTP), 621
network unreachable, erro, 214
network-scripts, diretório, 27
newaliases, comando, 383
newfs, 50
newgrp, 73
NFS (Network File System), 339, 350
acesso de root, 341
ajustando, 345
arquivo mestre, 349
bloqueio de arquivo, 340
cliente, 344, 346
configuração, servidor, 342, 344
conta nobody, 37, 341
convenção para atribuição de nomes, 341
cookies, 340
cotas de disco, 340
daemons, 616, 617
e dump, 121
e o arquivo fstab, 91
estatísticas, 347
exportando sistemas de arquivos, 342, 344
firewalls e, 341
montagem, 344
montagens hard *vs.* montagens soft, 345
montando sistemas de arquivos em tempo de inicialização, 346
mount, 346
opção all_squash, 341, 343
opção anongid, 341, 343
opção anonuid, 341, 343
opção insecure, 343, 346
opção no_root_squash, 342, 343
opção secure, 343, 346
opção secure_locks, 343
opção subtree_check, 343
opções comuns, 343
RPC e, 340
secure_locks, opção, 344
segurança, 341, 342, 346, 476
servidores de arquivos dedicados, 347
subtree_check, opção, 344
tamanhos de buffer, 345
TCP *vs.* UDP, 340
utilizando para exportar e-mail, 378
versões de protocolo, 339
nfsd, 616
nfsd, daemon, 342, 344, 346
nfsstat, 347
nice, 44, 45, 564
nice, valor, 40
NIS (Network Information Service), 357, 363
arquitetura, 358, 359
arquivos a compartilhar, 352
arquivos de banco de dados, 357, 358
arquivos de mapa, 358
commands, 359
configurando clientes, 363

configurando opções de controle de acesso, 362
configurando servidores, 361
configurando um domínio, 361, 363
grupos de rede, 361
LDAP *vs.*, 366
procedimento de consulta, 358
segurança, 476
servidor mestre, 357, 358, 361, 362
servidores de escravo, 361
servidores escravos, 357, 358, 359
níveis de execução, 590
alterando, 611
init e, 23, 25, 30, 611
nmap, 477, 479
nmbd, 571, 617
no-break, 551
nocanonify, recurso, sendmail, 400
nohup, 43
nome de domínio, DNS
segundo nível, 268
subdomínios, 269
nomes de arquivo
caracteres de controle em, 56
codificação sob o Samba, 572
colocando entre aspas, 52
correspondência de padrão, 7, 56
curingas de shell, 7
espaços em, 52
removendo falsos, 56
removendo suspeitos, 56
restrições de comprimento, 52
shell globbing, 56
nomes de caminhos, 35, 52
nomes de dispositivos, arquivos
nomes, 601
nomes de domínio, DNS, 265, 269
completamente qualificados, 267
distinção entre maiúsculas e minúsculas, 267
domínio in-addr.arpa, 277
grilagem, 267
hierarquia, 265
internacionalização, 272
ponto final em, 267
primeiro nível, 266, 267
registro, 268
regras, 265, 269
selecionando, 268
sintaxe, 267
nomes de domínios, DNS
registro, 260, 268
nomes de host
mapeando para endereços IP, 199, 211
nomes de hosts totalmente qualificados, 267, 277
nomes de usuário. *Ver* contas de usuário, nomes
notify, opção, DNS, 297
NS DNS, registros, 276, 285
nscd, daemon, 352, 618
NSEC DNS, registros, 325
NSF (National Science Foundation), 267
NSFNET, 194
nsupdate, 314
NTP (Network Time Protocol), 621
ntpd, 621, 622
ntpdate, 622
nullclient, recurso do sendmail, 407
nullclient, recurso, sendmail, 406
número de destino, SCSI, 84
números de dispositivo principais, 57, 599
números de dispositivo secundários, 57, 599
números de dispositivos primários, 600
números de dispositivos secundários, 600
NXT DNS, registros, 320

O

Oetiker, Tobi, 461
Oetiker, Tobias, 185
Oja, Joanna, 8
OpenLDAP, 365
openlog, rotina, 157, 158
OpenOffice.org, 569
Openwall GNU/*/Linux (Owl), 492
oprofile, 563
options, instrução, DNS, 296, 300
$ORIGIN, diretiva, DNS, 284
OSI, camadas, 196
OSPF, protocolo, 238, 240, 241
OSTYPE, macro (sendmail), 397, 398

P

Pack Management Project, 187
pacotes. *Ver* pacotes de software, *Ver também* redes
analisadores, 455, 456
descartados, 449
encaminhamento, 237
encaminhando, 235
encapsulamento, 196, 197
filtrando, 470, 486
gerenciamento, 168
ICMP, 490
rastreando, 449
sniffers, 257
tempo da viagem de ida e volta, 449
tratamento com Netfilter, 488
tratando com Netfilter, 491
pacotes de software
controle de revisão, 175, 181
dependências, 186
espaços de nomes, 186
gerenciamento, 167, 175
instaladores, 167
lista de, 188
localizações, 181, 184
repositórios, 170
RPM, formato, 167
padrões, 658, 659
centro de dados, 552
COBIT (Control Objectives for Information e technologias relacionadas), 660
conformidade do Windows com e-mail e Web, 570
Ethernet, 197, 247
gerenciamento de rede, 457
IEEE 802.*, 197, 247
IEEE 802.11g, 252
IEEE 802.3, 250
Internet, 194
ISSO/IEC 27001, 469
ITIL (Information Technology Interface Library), 659
Linux, 658, 659
LSB (Linux Standard Base), 659
POSIX, 659
segurança, 469
padrões de gerenciamento, redes, 457
paginação, 92, 558, 559, 563
páginas man, 8, 9
páginas, memória, 558, 559
Painter, Mark, 264
PAM (Pluggable Authentication Modules), 473, 474
paper sizes for printers, 537
paperconfig, 537
PAPERSIZE, variável de ambiente, 537
par trançado não-blindado. *Ver* UTP (unshielded twisted-pair), cabos
Parain, Will, 185
parando o sistema, 28, 30
partições, 88, 89, 96, 98, *Ver também* sistemas de arquivos
balanceamento de carga, 561
configurando, 96, 97
redimensionando com LVM, 104, 105
root, 88
swap, 89, 92, 98
troca (swap), 92
passwd, comando, 33, 68, 75
PAT (Port Address Translation), 225
PATA. *Ver* IDE
patches, software, 470
patentes
agência de patentes norte-americana, 657
diretiva de patentes da UE, 658
software, 657, 658
PC hardware
inicialização múltipla, 21, 22
PC, hardware. *Ver também* hardware
BIOS, 18
caractere delete, 592
drivers de dispositivo, 599
inicialização, 18
prioridade de dispositivo de inicialização, 18
vs. hardware de UNIX, 18
PCL, linguagem de impressora, 527, 529
PDF, 528, 529
pdftops, 533
Perl, 3, 10, 107, 498, 500, 569, 635
e syslog, 157
em scripts administrativos, 3
empacotando tarefas cron, 357
exemplo inseguro, 467
scripts de exemplo, 366
verificação de ID de usuário, 472
verificação de senhas nulas, 472
permissões, 62
arquivo, 59, 474
chmod e, 61
importantes, 474
sendmail, 420, 421
umask e, 62
permissões padrão, 62
pesquisa, bit, 62
PGP (Pretty Good Privacy), 424, 482
Phonetics Sensaphone, 551
pico, editor, 3
PIDs, 39
PIM, protocolo, 241
ping, 223, 448, 449
pipes identificados, 56, 58
pipes, identificados, 56, 58
pirataria, 656
PIX, equipametno, 224
PJL, linguagem de impressora, 528
.plan, arquivo, 621
Pluggable Authentication Modules (PAM), 473, 474
poff, comando, 232
pon, comando, 232
ponto de montagem, sistema de arquivos, 53
POP (Post Office Protocol), 372, 378, 570, 618
pop, agente de entrega, 399
popd, 618
pornografia, 656
portadora via hardware, 587
portadora via software, 587
portas bem conhecidas, 486
portas de rede bem conhecidas, 477
portas de rede, privilegiadas, 199
portas privilegiadas, 199
portas seriais, 581, 583
arquivos de dispositivos, 589
controle de fluxo, 587, 588
opções de configuração, 593, 594
parâmetros, configurando, 588
portadora via software/hardware, 587
reconfigurando, 594

redefinindo, 594
portas, network
privileged, 486
portas, rede, 199
bem conhecidas, 477, 486
varredura, 477, 479
portmap, daemon, 341, 611, 615
POSIX, 474, 659
APIS sob Windows, 569
capacidades da conta root, 33
Post Office Protocol (POP), 372, 378, 570, 618
postconf Postfix, ferramenta de configuração, 436
Postel, Jon, 194
Postfix, 433, 443
arquitetura, 433
autenticação, 440
configurando, 434, 441
controle de acesso, 439, 440, 443
controle de spam, 441, 442
daemons de diretiva, 442
debugging, 444
depurando, 443
domínios virtuais, 437, 439
entrega local, 437
enviando e-mail, 434
filtragem de conteúdo, 442
filtragem de vírus, 442
filtro de vírus amavisd, 442
gerenciador de fila, 433
listas cinzas, 442
listas negras, 441
no ambiente chroot, 434
recebendo e-mail, 433
segurança, 434
tabelas de pesquisa, 436
utilitários de linha comando, 434
PostScript, 527, 529
poweroff, 30
PPD (Printer Description Files), 532
PPIDs, 40
PPP, protocolo, 232
comandos, lista de, 228
configuração, 227, 232
pppd, daemon, 227, 228, 230, 232
pppstats, comando, 232
pr, 538
Practical Extraction and Report Language. *Ver* Perl
Pre-boot eXecution Environment (PXE), 161, 162, 619
Preston, W. Curtis, 141
Pretty Good Privacy (PGP), 424, 482, 483
primeiro nível, domínios, 267
printing
paper sizes, 537
prioridade, processos, 40, 44, 45
privacidade, 654
.private, arquivo de chave DNSSEC, 318
privileged ports, 486
/proc, sistema de arquivos, 48, 221, 223, 600, 602
/proc/mdstat, arquivo, 101, 103
/proc/sys, diretório, 602
processo init, 17
inicialização, 18
processos, 39
agendamento, 32
alterando as credenciais de posse, 33
alterando IDs de usuário e de grupo, 33
canais de E/S, 41
descontrolados, 49, 50
EGID (effective group ID), 40
enviando sinais para, 43
espontâneos, 16
estados de execução, 44
EUID (effective user ID), 40
GID (group ID), 40
identidades
real, efetivo e salvo, 32
IDs, 39
monitorando, 45, 47
parâmetro FSUID, 40
parando e iniciando, 44
posse, 40
PPID (PID pai), 40
prioridade, 40, 44, 45
proprietário, 32
órfãos, 41, 44, 45
terminal de controle, 41
zumbi, 611
zumbis, 41, 44, 45
processos descontrolados, 49, 50
processos órfãos, 41, 44, 45
processos zumbis, 611
procinfo, 563
procinfo, comando, 560
procmail, 371, 407, 442
profiler, sistema, 563
programas de discagem, 595
programas, localizando, 10
Project Athena, 513
.project, arquivo, 621
projeto Gutenprint, 532
protocolo SMTP, 570
protocolos de correção de erros, 595
protocolos de roteamento de vetor de distância, 237
provide-ixfr, opção, DNS, 314
proxies, serviço, 487
proxies, servidores Web, 507
ps, 45, 47, 557, 563
pseudodispositivos, 600
pseudo-usuário, 69
PTR DNS, registros, 277, 311
Punycode, 272
PuTTY, 566
pvcreate, utilitário LVM, 102
PXE (Pre-boot eXecution Environment), 161, 162, 619
PXELINUX, 161
Python, 3, 11, 365, 635

Q

qmgr, 434
qpage, agente de entrega, 399
qpopper, servidor de e-mail, 378
qshape, 443
query-source, opção, DNS, 298
questões de direitos autorais, 653
questões legais, 652, 658
criptografia, 652
direitos autorais, 653
EULA (End User License Agreements), 655
licenças de software, 656
pornografia, 656
privacidade, 654
registros de chamadas e logs Web, 654
responsabilidade para dados, 656
QUIT, sinal, 42, 43

R

R-232 para adaptadores USB, 596
R-232, padrão, 581, 583, 588
racks para equipamentos, 551
RAID, 99, 105, 555, 616
RAID de software, 99
raidtools, 101
rc, scripts. *Ver* inicialização, scripts
rc.local, script, 25
rcmd, 619
rcp, 475
rcp, comando, 619
RCPT comando, SMTP, 409
RCS, 176, 178
rcsdiff, 177
rdesktop, 568
rdist, 353, 355
RDP (Remote Desktop Protocol), 568
rdump, 121
RealVNC, 568
reboot, 30
recursion, opção, DNS, 297
recursive-clients, opção, DNS, 298
Red Hat Network, repositório de softwares, 171
Red Hat, configuração de rede, 217
redes. *Ver também* configuração de rede; Ethernet; IP, endereços; roteamento; TCP/IP
adicionando uma máquina a uma rede local, 210, 216
ajustando, 221, 223
ARP (Address Resolution Protocol), 209, 210
ATM, 254, 255
balanceamento de carga, 555
bancos de dados administrativos, 357
cabeamento, 246
CIDR (Classless Inter-Domain Routing), 202, 203
conectando com PPP, 226
conectando e expandindo, 250, 252
configuração de interface, 211, 213
congestionamentos de transmissões, 212
conversão de endereços. *Ver* NAT
daemons, 620, 621
depurando com mii-tool, 213, 214
DHCP (Dynamic Host Configuration Protocol), 219, 221
endereços, 198, 207
equilíbrio de carga, 270
estatísticas, 451, 454
firewalls, 224, 486, 488, 491
IPsec, 652
IPv6, 203
loopback, 200, 208, 213, 278
máscaras de rede, 200, 202
monitorando, 452
MTUs, 198, 253
NAT, 205, 206, 225
pacotes. *Ver* pacotes
padrões de gerenciamento, 457
PAT, 225
ping e, 448, 449
portas, 199
PPP, 232
protocolos de gerenciamento de redes, 457, 459
questões de gerenciamento, 260, 447
questões de projeto, 258, 260
redes privadas virtuais. *Ver* VPNs
redundância, 647
relatórios sobre as atividades da interface, 454
rota padrão, 207, 208, 215, 232, 236
rotas estáticas, 215
scanner, Nessus, 479
sem fio, 197, 252, 253
solucionando problemas, 256, 257, 447, 454
sub-rede, 200, 202
tabelas de roteamento, 453
tempo da viagem de ida e volta, 449
varredura de porta, 479
varredura de portas, 477
VLANs, 251
redes discadas. *Ver* PPP, protocolo
redes privadas virtuais. *Ver* VPNs
redesbancos de dados administrativos, 352
redimensionando
partições de discos, 105
redimensionando partições de disco, 104

redirect, recurso, sendmail, 400
REFUSE_LA, opção, sendmail, 423
registrando em log
for BIND, 330
para BIND, 313
para o BIND, 287, 302, 327
para sendmail, 430
registro de ações, sistemas de arquivos, 86
registro dos nomes de domínio. *Ver* nomes de domínio, registro
registro e rastreamento de problemas, 642, 645
registro em log. *Ver também* syslog; arquivos de log
ao console do sistema, 157
ao servidor central, 153
em servidor central, 155
em tempo de inicialização, 148
falhas de hardware, 546
kernel, 148, 149
para cron, 107
para o CUPS, 539
para sendmail, 431
por meio de syslog, 158
por meio do syslog, 157
tempo de inicializa inicialização, 149
registros de recursos, DNS, 273, 284
A, 277, 285
A6, 283
AAAA, 283
adesão, 286
bloco A, 283
caracteres especiais em, 273
CNAME, 279
cola, 285
DNAME, 283
DNSKEY, 320
DS, 320, 322
formato, 273
KEY, 318, 320
LOC, 281
MX, 278, 279
NS, 285
NSEC, 320, 322, 324
NXT, 320
ponto final em nomes, 273
PTR, 277, 311
RRSIG, 320, 322, 324
SIG, 320
SOA, 275, 276, 313
SRV, 281, 282, 325
time to live, 273
TXT, 282, 297
WKS, 282
registros DNS do bloco A, 283
reinicialização, 29, 594
Reiser, Hans, 86
ReiserFS, sistema de arquivos, 86
Remote Desktop Protocol (RDP), 568
Remote Procedure Call. *Ver* RPC
remoto, arquivo, 589
renice, 44, 45
renice, comando, 564
repositórios, software, 170, 187
request-ixfr, opção, DNS, 314
resize_reiserfs, 105
resolver library, DNS, 290
resource records, DNS
NSEC, 325
restore, 123, 125
comando, 645
retransmissão aberta, e-mail, 409
retransmissão promíscua, sendmail, 409
reverse mapping, DNS, 278
RFCs
Carrier Pigeon Internet Protocol, 660
documentos BCP, 195
documentos FYI, 195
documentos STD, 195
espaço de endereçamento privado, 205, 206
relacionadas ao DNS, 263
relacionadas ao TCP, 202
relacionadas com DNS, 264, 338
relacionadas com LDAP, 365
relacionadas com NFS, 350
relacionadas com rede, 463
visão geral, 194, 195
RHN, pacotes repositórios, 171
.rhosts, arquivo, 475, 619
Riggle, David, 264
RIP, protocolo, 238, 239, 240, 241
RIP-2, protocolo, 240
RJ-11, conectores, 594
RJ-45, conectores, 585
RJ-45S, conectores, 250
rlog, 177
rlogin, 475
rm, 56, 57
rmdir, 57
rmmod, comando, 607
rndc, 305, 306, 313, 330, 331
rndc-confgen, 305
rodízio de arquivos de log, 111, 145, 149, 150
root, conta, 32
acessando, 33, 34, 36
acessando via NFS, 341
arquivo /etc/sudoers, 35
capacidades do POSIX, 33
encurralando, NFS, 341
/etc/sudoers, 36
ID de usuário, 33
restringindo acesso, 475
senhas, 34
rootkits, 477, 654
Rossi, Markku, 537
rota padrão, 215, 232, 236
rotas estáticas, 208, 214, 215
roteadores, 252
roteamento, 207, 209, 235, 245
CIDR (Classless Inter-Domain Routing), 201, 204
com múltiplos ISPs, 239
daemons e protocolos, 237, 242
encaminhamento de pacotes, 214, 223, 235, 237
estratégia, 242
máscaras de rede, 200, 202
métrica do custo, 239
OSPF, 240
pacote Zebra, 242
PPP, 227
protocolo BGP, 238
protocolo EIGRP, 240
protocolo IGRP, 240
protocolo IS-IS, 241
protocolo OSPF, 238, 240, 241
protocolo RIP, 238, 239, 240, 241
protocolos, 239, 241
protocolos de estado de link, 238
protocolos de gateway externos, 239
protocolos de gateway internos, 239, 241
protocolos de vetor de distância, 237
rastreamento de rota, 449
redirecionamentos de ICMP, 237
redirecionamentos ICMP, 208, 223
rota padrão, 207, 208, 215, 232, 236
rotas estáticas, 208, 214, 215
roteadores Cisco, 243, 244
roteamento estático *vs.* dinâmico, 242
sendmail, 405
sistemas autônomos, 239
sub-rede, 200, 202
tabelas, 207, 208, 236, 237, 453
unreachable networks, 214
XORP (eXtensible Open Router Platform), 242
roteamento de origem, 223, 235
rotinas de inicialização, 16, 18, 19, 22, 98
GRUB, 19, 20, 21, 23
LILO, 20, 21, 22, 23
rotinas de inicialização múltipla, 21
route, comando, 208, 214, 215, 218, 620
routed, daemon, 239, 241, 620
Router Discovery Protocol, 241
Rowland, Craig, 158
RPC (Remote Procedure Call)
gerenciando atribuições de porta, 611
mapeando números de serviço para portas, 615
NFS e, 340, 615
rpc.bootparamd, daemon, 619
rpc.lockd, daemon, 617
rpc.mountd, daemon, 342, 617
rpc.nfsd, daemon, 342, 616
rpc.rquotad, daemon, 617
rpc.statd
daemon, 617
rpc.ypxfrd, daemon, 618
rpcbind, daemon, 611
rpciod, daemon, 617
rpm, 168
rquotad daemon, 340
RRDTool, ferramenta de representação gráfica, 461
rrestore, 124
rrset-order, instrução, DNS, 299
RRSIG DNS, registros, 320, 322, 324
rsh, 619
rshd, 619
rsync, 140, 355, 357, 620
rsyncd, daemon, 620
rsyncd.secrets, arquivo, 356
RTS (Request to Send), 587
RunAsUser, conta de usuário de sendmail, 419
Russinovich, Mark, 654
rxvt, 569

S

S/MIME, 424
SafeFileEnvironment, opção, sendmail, 421
SAGE guild, 663, 664, 665
SAGE-AU guild, 663
SAIT, fitas, 118
Samba, 570, 579, *Ver também* aliases; e-mail; spam; Windows
arquivos de log, 578
autenticação de usuário, 573
CIFS, 570
codificação do nome de arquivo, 572
codificação UTF-8, 572
compartilhamentos de grupo, 574
compartilhando arquivos, 570, 573
compartilhando impressoras, 576, 578
configuração, 571
configurando senhas, 571
criptografia de senha, 571
daemon de servidor de arquivo e de impressora, 617
daemons, 571, 616, 617
depurando, 578, 579
exibindo conexões ativas e arquivos bloqueados, 579
instalação, 571, 572
listagem das opções de configuração, 571
navegação pelo ambiente de rede, 572
programa de transferência de arquivos a partir da linha de comando, 575
segurança, 571
servidor WINS, 572
samhain, 480, 481
San (Storage Area Network), servidores, 347
SANE, 663

SANS Institute, 468, 494, 663, 665
sar, 454, 562, 563
Sarbanes-Oxley Act (SOX), 469, 657, 659
SASL (Simple Authentication and Security Layer), 424
SATA (Serial ATA), 80, 82
savelog, 150
Sawyer, Michael, 332
/sbin, diretório, 54
SCA (Single Connector Attachment), plugue, 83
SCO, 653
scp, 483
scripts de chat, PPP, 230, 231
scripts de inicialização, 23, 28
  CUPS, 533, 539
  diretório /etc/init.d, 25
  /etc/init.d diretório, 27, 28
  exemplos, 24, 27
  inicializando, 28
  init e, 15, 23, 27, 28
  servidor NFS, 342
scripts empacotadores para localização, 187
SCSI, 80, 82, 84
  BIOS, 95
  conectores, 82, 84
  fast e wide, 82
  instalando, 95
  nomes do dispositivo, 601
  números de unidade lógica, 84
  solucionando problemas, 84
  vs. IDE, 84
scsi_eh_N, daemon, 616
SDSC Secure Syslog, 150
search, diretiva, DNS, 293
SEC (Simple Event Correlator), 158
secure, arquivo, 147
SecurityFocus.com, 493
segurança
  analisador de pacotes, 456
  analisadores de pacote, 455
  Application Security Checklist, 469
  arquivo .rhosts, 475
  arquivo /etc/securetty, 475
  arquivo /etc/shadow, 471, 473
  arquivo auth.log, 147
  arquivo passwd, 471, 473
  arquivo secure, 147
  arquivos de dispositivos, 475
  arquivos de log, 144, 153
  ataque DOS via syslog, 153
  ataques de negação de serviço (denial of service – DOS), 423
  ataques de negação de serviço (DOS), 422
  BIND, 292, 297, 316, 325
  broadcast direcionado, 223
  cavalos de Tróia, 477
  certificações, 468
  CISA (Certified Information Systems Auditor), 468
  CISSP (Certified Information Systems Security Professional), 468
  conta root, 34, 472
  correções de software, 470
  das impressoras, 543
  das redes sem fio, 253
  de impressoras, 543
  de named, 316, 317
  de sendmail, 416, 424
  de senhas, 34, 67, 69, 471, 472, 479
  de VPNs (virtual private networks), 491
  de VPNs (Virtual Private Networks), 224, 491
  detecção de invasão, 480, 481
  dicas, 470
  diretiva, 650
  diretrizes, 649
  DNS, 292, 297, 316, 325
  DNSSEC, 272, 320, 324
  do arquivo group, 475
  do arquivo passwd, 475
  do Postfix, 434
  do PPP, 227
  do sendmail, 388, 409, 419, 476
  e-mail, 409
  e-mail para programas, 381, 421
  encaminhamento IP, 223
  engenharia social, 466
  executando su, 35
  falhas no Linux, 465
  falhas no UNIX, 465
  falsificação de IP, 224
  ferramentas, 477, 486
  ferramentas poderosas, 481
  firewalls, 486, 488, 491
  firewalls baseados em host, 224
  fontes de informações, 493, 495
  FTP, 475, 509
  GIAC (Global Information Assurance Certification), 468
  higiene de contas, 67
  identificando portas abertas, 453
  informando invasões, 493
  iptables, 488, 491
  Kerberos, 482
  LDAP e, 367
  Linux aprimorado, 492
  lista negra para conexões automáticas, 608
  monitorando, 12, 470, 471, 477, 480, 488
  negação de serviço, 357
  negação de serviço (DOS), ataques, 278
  NFS, 341, 342, 346, 476
  NIS, 362, 476
  nomes de login, unicidade, 68
  padrões, 469
  PAM (Pluggable Authentication Modules), 473, 474
  permissões de arquivo, 474
  programas setuid, 474
  recursos específicos do fornecedor, 494
  rede, 223, 225
  redirecionamentos ICMP, 209, 223
  registro em log de evento remoto, 475
  restringindo acesso root, 475
  rootkits, 477, 654
  roteamento de origem, 223
  Samba, 571
  SDSC Secure Syslog, 150
  SELinux, 481
  senhas de sombra, 67
  senhas sombras, 471
  senhas-sombra, 72
  SNMP, 459
  SSH, 475, 483, 484
  SSL, 505, 507
  stunnel, 484, 486
  syslog, 153
  terminais, 475
  tratando ataques, 492, 493
  TSIG (assinaturas de transação), 311, 318, 320
  variáveis de rede do kernel, 225
  varredura de porta, 477, 479
  vírus, 476, 477
  visão geral, 465
  vs. conveniência, 467
  X Window System, 515, 516, 517, 567
SEGV, sinal, 42
SELinux, 481
sem fio, redes. *Ver* redes sem fio
Sender ID, 416
Sender Policy Framework (SPF), 282, 416
sendmail, 369, 618
  agentes de entrega, 371
  atuando como MSA/MTA, 372
  autenticação e criptografia, 419, 424
  banco de dados de acesso, 411, 413
  cabeçalhos, 372, 375, 413, 414
  configuração, 389, 391, 393, 408, 410, 415
  configuração de depuração, 388
  controlando falsificação, 423
  depurando, 394, 427, 431
  desempenho, 424, 427
  divisão de envelope, 425
  documentação, 394
  e o arquivo de comutação de serviço, 391
  e-mail para uma conta desativada, 77
  entrega prolixa, 428, 429
  estatísticas, 427
  executores de fila, 426
  exemplos de configuração, 416, 419
  filas, 392, 393, 425, 426, 430
  flags de linha de comando, 391
  grupos de filas, 425
  história, 388
  instalação e guia de operação, 444
  instalando, 389
  m4 e, 394, 396, 407
  macro OSTYPE, 397
  mascaramento, 404, 406
  miltering, 415
  modos de entrega, 424
  no ambiente chroot, 422
  opções de configuração, 407
  opções de privacidade, 421, 422
  permissões, 420, 421
  posse, arquivos, 419, 420
  recurso virtusertable, 403
  recursos de controle de spam, 409, 416
  Red Hat, 26
  registro em log, 430, 431
  retransmissão, 409, 411
  script de inicialização, 27
  segurança, 388, 409, 416, 419, 424, 476
  sites de backup MX, 393
  slamming, 415
  tabelas e bancos de dados, 401, 403
  taxa e limites de conexão, 414
  utilizando o SMTP para depurar, 429
  versões, 388
Sendmail, Inc., 369, 424
sendmail.cf, arquivo, 389, 392, 393
sendmail.cw, arquivo, 400
sendmail.st, arquivo, 427
senhas
  criptografia, 67, 68, 69, 571
  expirando, 472
  força, 473, 479
  FTP, 508
  grupo, 73
  iniciais, 74
  quebrando, 479
  root, 34
  rotina de partida, 467
  Samba, 571
  segurança, 34, 471, 473
  seleção, 34, 74, 471, 472
  sombra, 67, 72, 471
  usadas uma única uma vez, 484
  usadas uma única vez, 484
senhas sombras, 67, 72, 471
SEPP, 187
Serial ATA (SATA), 80, 82
server, instrução, DNS, 301, 314
service.switch, arquivo, 215, 216, 391
servidor mestre de nomes, DNS, 289
servidor mestre, NIS, 357, 358, 361, 362
servidor WINS, Samba, 572
servidores
  Apache. *Ver* hospedagem Web
  arquivos de sistema, 356
  backup de e-mail, 377
  balanceamento de carga, 555
  DNS/BIND, 288, 290

equilíbrio de carga, 270
escravos NIS, 357, 358, 361
FTP, 508, 509
HTTP, 501
impressora de rede, 534
Kerberos Windows e DNS, 325
NFS, 342, 344, 347
NIS mestre, 358, 361, 362
NIS slave, 359
nomes. *Ver* BIND, DNS, e named
proxy Web, 507
Squid, 507, 508
Storage Area Network (San), 347
TUX, 503
VNC, 568
Web. *Ver* hospedagem Web
WINS, 572
X Window System para Windows, 567, 569
servidores de nomes. *Ver também* BIND; DNS; named
armazenamento em cache, 270, 291
armazenamento em cache negativo, 270
arquivo switch, 336
atualizações dinâmicas, 314, 315
autorizados, 289, 291
conversor, 293, 294
delegação, 269
delegação de zonas, 285, 286
delegações, 334
delegações defeituosas, 329, 333
dicas, 290
encaminhando, 299
escravos, 289
mestre, 289
não autorizados, 289
números seriais de zona, 313
recursão, 289, 297
resolver, 290
script keep-running, 292
stub, 289
servidores escravos, NIS, 357, 358, 359, 361
setfacl, 64, 65
setores e trilhas, discos, 86
setrlimit, chamada de sistema, 564
setserial, 588
setuid/setgid, atributo de arquivo, 33, 59, 60, 474
Shapiro, Gregory, 424
shell
arquivos de inicialização, 75
curingas para nomes de arquivo, 7
globbing de nome de arquivo, 56
login, 70
SHELL, variável de ambiente, 594
show interfaces, 243
showmount, 344
shutdown, 29, 154
SIG DNS, registros, 320
Simple Network Management Protocol. *Ver* SNMP
sinais, 41, 43, *Ver também* nomes de sinais individuais
capturados, bloqueados ou ignorados, 42
CONT, 44, 49
enviando para um processo, 43
KILL, 43, 44
lista dos importantes, 42
rastreando, 48
STOP, 44, 49
TERM, 43
TSTP, 44
sincronização de clocks, 621
sincronização de tempo, 621, 622
sincronizando arquivos
copiando, 353
rdist, 353, 355
rsync, 355, 356
wget/ftp/expect, 356, 357
sistema de arquivos
raiz, 54
sistema de arquivos raiz, 17, 22, 54
sistema operacional, instalação. *Ver* Linux, instalação
sistemas autônomos, 239
sistemas de arquivos, 51, 52, *Ver também* partições
ativando ACLs, 63
balanceamento de carga, 555, 561
convenção para atribuição de nomes, NFS, 341
convertendo ext2fs em ext3fs, 86
copiando, 126
corrigindo, 95
cotas, 340
criando, 89, 90, 97, 98
danificados, 94, 95
desativando a execução setuid, 474
desmontando, 53
dimensionando para backups, 114
diretórios lost+found, 90, 95
exportando NFS, 342, 344
ext2fs, 62, 86, 89
ext3fs, 62, 86
FAT, 86
fazendo backup, 114
fuser, 53
inodes, 89
journaling, 86
limpando com cron, 110
limpeza, 110
Linux, 86, 87
loopback, 53
lsof, 54
montagem automática, 90
montando, 53, 54, 90
montando em tempo de inicialização, NFS, 346
mountando, 91, 575
organização, 54, 55
raiz, 17, 22
reinicializando, 50
ReiserFS, 86
root, 88
rótulos, 91
smbfs, 575
superblocos, 90
sysfs, 600, 607
verificando e reparando, 17, 91, 93, 95, 98
sistemas de arquivos danificados, 95
sistemas de refrigeração, 646
Site Finder, ferramenta da VeriSign, 300
skel, diretório, 76
slamming, controlando no sendmail, 415
slapd, daemon, 365, 387
SLIP, 226
slurpd, daemon, 365
SMART_HOST, macro (sendmail), 405, 417
SmartList, 385
SMB, protocolo. *Ver* Samba
smb.conf, arquivo, 571, 572, 576, 579
smbclient, 575
smbcontrol, 579
smbd, daemon, 571, 617
smbfs, sistema de arquivos, 575
smbpasswd, 572
smbstatus, 579
SMP (symmetric multiprocessing), 557
smrsh, agente de entrega de e-mail, 371, 407, 421
SMTP, protocolo, 429, 434
smtpd, 618
smtpd/smtpfwdd, 376
smurf, ataques, 223
SNMP, 457, 463, 620
agentes, 459, 460
CiscoWorks e, 463
coleta de dados, 461
ferramentas, 460, 462
MIBs (Management Information Bases), 458
NET-SNMP, 459, 461
OIDs (object identifiers), 457, 458
organização dos dados, 457, 458
RMON MIB, 459
string comunitária, 459
traps, 458
utilizando Cacti, 461
utilizando Nagios, 461
snmpd, daemon, 459, 620
snmpd.conf, arquivo, 459
snmpwalk, 460
SOA DNS, registros, 275, 276, 313
socket, chamada de sistema, 58
sockets de domínio local, 56, 58
sockets, domínio local, 58
software. *Ver também* pacotes de software
compartilhando pelo NFS, 186
correções, 470
desenvolvimento, 632, 636
erros de configuração, 467
ferramentas de gerenciamento, 187
imprimindo, 538
licenças, 656
patentes, 657, 658
pirataria, 656
princípios de engenharia, 635, 636
recomendado, 187
vulnerabilidades, 467
software de lista de discussão
Majordomo, 384
software e sistemas de backup
ADSM/TSM, 141
Amanda, 140
Bacula, 127, 140
cpio, 126
dd, 126
dump/restore, 120
sistemas comerciais, 140, 141
tar, 126
Veritas, 141
software e sistemas, backup. *Ver também* backups; Bacula
solucionando problemas
Bacula, 139, 140
BIND, 326, 335
CUPS, 539, 540
hardware de rede, analisador de linha T-BERD, 257
hardware de rede, analisadores de cabos, 257
hardware de rede, sniffers, 257
impressão em rede, 539
impressoras, 539, 540
linha serial, 596
named, 326, 335
Postfix, 443, 444
processos descontrolados, 49, 50
RAID, 103, 104
redes, 256, 257, 447, 454
redes com mii-tool, 213, 214
Samba, 578, 579
SCSI, 84
sendmail, 427, 431
servidor Xorg X, 521
sistema lento, 563, 564
syslog, 156, 157
terminal travado, 594
X Window System, 521, 523
solucionando problemas:, 523
Sony, rootkits, 654
sortlist, opção, DNS, 299
SOX (Sarbanes-Oxley Act), 469, 657, 659
spam
combate, 416

combatendo, 415
eliminando com DNS, 282
filtro de vírus amavisd, 442
listas cinzas, 442
listas de buracos negros, 416
listas negras, 413
perigo em responder, 409, 416
Postfix, 441, 442
recursos de controle de sendmail, 416
recursos de controle do sendmail, 409
recursos web para lutar contra, 416
retransmissão, 409, 411
Sender ID, 416
Spam Cop, 416
SpamAssassin, 416
spammers móveis, 415
SPF, 282, 416
verificação de cabeçalho de e-mail, 413
verificação de cabeçalho de e-mails, 414
Spam Cop, 416
SpamAssassin, 416
SPF (Sender Policy Framework), 282, 416
Squid, cache Web, 507, 508, 656
SRV DNS, registros, 281, 282, 325
ssh, 483
SSH, 483, 484
clientes Windows, 566
encaminhamento para o X, 516
encaminhamento para X, 517
encaminhamento X, 567
segurança, 475
sshd
daemon, 483, 618
SSL, 505, 507
Stafford, Stephen, 8
star, 140
StarOffice, 569
startx, 514
statd, 617
daemon, 340
STD, documentos, 195
sticky bit, 59, 60
STOP, sinal, 42, 43, 44
Stow, GNU, 187
STP (shielded twisted-pair), cabo, 582
strace, 48
stty, 587, 593, 594
stunnel, 484, 486
su, 34
subdomínios, DNS, 269
submit.cf, arquivo, 389, 393
sub-rede, 200, 202
Subversion, 179, 181
sudo, 35, 36, 69, 147
sudo.log, arquivo, 147
superblocos, 90
superusuário. *Ver* root, conta
SUSE, configuração de rede, 218
svn, 180
svnserve, daemon, Subversion, 179
svnserve.conf, arquivo, 180
swap, 92
swap, espaço, 89
swapon, 91, 98, 559, 561
swatch, 158
switch de serviço, arquivo, 391
switch, arquivo, 294, 336
switches, 248, 251, 252, 253
Swpkg, 187
sync, chamada de sistema, 30, 90
/sys, diretório, 600
sysctl, 603
sysfs, sistema de arquivos virtual, 600, 607
syslog, 150, 157, 158, *Ver também* arquivos de log; registro em log
/etc/syslog.conf, arquivo, 146, 151, 155
ações, 153
alternativas, 150
arquitetura, 150
arquivo syslog.conf, 176
ataques DOS via, 153
bibliotecas, 157, 158
configuração, 153
configurando, 151, 153
daemon syslogd, 146, 150, 153, 621
depuração, 156, 157
e registro em log DNS, 330
e registro em log do DNS, 327
example using Perl, 158
exemplos de configuração, 154, 156
interface de programação, 157, 158
níveis de gravidade, 152
nomes de recursos, 152
registro em log remoto, 475
registros de data/hra, 151
reiniciando, 151
saída, 155
segurança, 153
servidor central, 153, 155
utilizando programas, 156
syslog, rotina, 150
syslogd, daemon, 146, 150, 153, 621
syslog-ng, 150
system-config-kickstart, 165
system-config-netboot, 161

T

talk, 620
talkd, daemon, 620
tamanho, atributo de arquivo, 60
tamanhos de papel para impressoras, 536
Tanenbaum, Andrew S., 4
tar, 125, 126
taxa de transmissão de dados, 595
TCP
empacotadores, 611
estados de conexão, 452
vs. UDP para NFS, 340
TCP/IP, 193, 195, 200
bits TOS, 232
CIDR (Classless Inter-Domain Routing), 201, 202, 204
conjunto de protocolos, 195, 196
encapsulamento, 197
encapsulamento de pacotes, 196
fragmentação, 198, 449
história, 193
interface loopback, 200, 208, 213, 278
IPv6, 202, 206, 207
máscaras de rede, 200, 202
modelo de rede, 196
NAT, 205, 206, 225
opções sofisticadas (SACK, ECN), 216
portas, 199
sub-rede, 200, 202
tcpd, daemon, 611
tcpdump, 455
telas. *Ver também* monitores de computador
telinit, 22, 30, 591, 611
telnet, 243, 499
TELNET, protocolo, 619
telnetd, 619
Tel-splice, conector, 587
Template Tree 2, configuração de sistema, 185
tempo de atividade, 557
Tera Term Pro, 566
TERM
sinal, 42, 43
variável de ambiente, 592, 593
TERMCAP, variável de ambiente, 592
terminadores (SCSI), 83
terminais, 589, 592
bancos de dados de capacidades, 592
caracteres especiais, 592, 594
controle, 41
destravando, 594
opções de configuração, 593, 594
seguros, 475
terminais seguros, 475
terminais seriais, configurando, 589, 592
terminais virtuais e o X, 522
terminal de controle, 41
Terminal Server, serviço, Windows, 568
termtype, arquivo, 591
Terry, Douglas, 264
testando, sistema, 182
testparm, 571
Texinfo, 8
tftp, 244
TFTP, 220, 619
Thomas, Eric, 386
threads, kernel, 16
TIA (Telecommunications Industry Association), 249
TightVNC, 568
tip, 589, 595
TLS. *Ver* SSL
TLT/S-DLT, fitas, 118
/tmp, diretório, 54
/tmp, partição, 89
top, 47, 563
top, comando, 558
topology, instrução, DNS, 299
Torvalds, Linus, 4
trabalhos, agendamento, 611
traceroute, 449
transfers-in, opção, DNS, 298
transfer-source, opção, DNS, 298, 311
transfers-out, opção, DNS, 298
transfers-per-ns, opção, DNS, 298
tríade CIA, 468
Tridgell, Andrew, 355, 570
Troan, Erik, 149
troca (swap) de byte, 126
Trojnara, Michal, 484
trusted-keys, instrução, DNS, 301
TrustedUser, conta de usuário de sendmail, 419
Ts'o, Theodore, 86
tset, 593, 594
TSIG (assinaturas de transação), 311, 318, 320
Tsirigotis, Panos, 611
TSM (Tivoli Storage Manager), 141
TSTP, sinal, 42, 43, 44
TTL (time to live), pacotes, 450
TTL para registros de recursos DNS, 273
$TTL, diretiva, DNS, 273, 276
$TTL, diretiva, 295
tune2fs, 86, 94
TUX, servidores, 503
Tweedie, Stephen, 86
TXT DNS, registros, 282, 297

U

Ubuntu, configuração de rede, 218
udev, 57
sistema, 600
udev.conf, diretório, 600
udevd, 600, 620
UDP (User Datagram Protocol), 193, 195, 340
UIDs, 40, *Ver* IDs de usuário
Ultr@VNC, projeto, 568
Ultra SCSI. *Ver* SCSI
umask, 62, 75
umount, 53, 91, 346
uname, 606
unicast, endereços, 207
unidades de fitas
nomes de dispositivos, 601
UNIX *vs.* Linux, 4
unlink, 58

update-policy, cláusula, DNS, 315
update-rc.d, 28
uptime, 563
URLs, 498, 499
us, domínio, 266
USB, 596, 597
adaptadores RS-232, 596
discos, 80, 105, 106, 118
dispositivos suportados, 541
identificação de dispositivo, 598
impressoras, 534, 539
na placa do SCSI, 82
use_cw_file, recurso, sendmail, 400
USENIX Association, 663, 665
User Datagram Protocol. *Ver* UDP
useradd, 73, 78
userdel, 78
usermod, 71, 78
/usr, diretório, 54
/usr/local hierarchy
distribuição, 183
/usr/local, hierarquia, 181, 184
compilação, 182, 183
distribuição, 183
organizando, 181, 182
testes, 182
/usr/src/linux, diretório, 603
usuários. *Ver também* contas de usuário
acordos de diretiva, 651
documentação, 642
mantendo-os satisfeitos, 623, 624
UTP (unshielded twisted-pair), cabos, 248, 250, 257, 582

V

V.90, padrão de modem, 595
van den Berg, Stephen R., 385, 407
/var, sistema de arquivo, 89
/var, sistema de arquivos, 54
/var/adm, diretório, 146
/var/log, diretório, 146
/var/log/messages, arquivo, 148, 154
/var/spool/mqueue, diretório, 392, 430
/var/yp, arquivo, 358
variáveis, inicializando em arquivos de inicialização, 75
velocidade, configurando para uma interface de rede. *Ver* mii-tool
Venema, Wietse, 433, 590
verificação de cabeçalho, sendmail, 413, 414
Veritas, ferramenta de backup, 141
VERSIONID, macro (sendmail), 397
vetor de distância, protocolos, 240
VFS (Virtual File System), 86
vfstab, arquivo, 348
vgcreate, utilitário LVM, 102
vgdisplay, utilitário LVM, 102, 104
vgscan LVM, utilitário, 102
Viega, John, 385
view, instrução, DNS, 307
.vimrc, arquivo, 75
vipw, 74
Virtual File System (VFS), 86
Virtual Network Computing. *Ver* VNC, protocolo
VirtualHost, cláusula, Apache, 505
virtusertable, recurso, sendmail, 403
vírus, 476, 477
visudo, 36
Vixie, Paul, 107, 264
Vixie-cron. *Ver* daemon de cron
VLANs, 251
vmlinuz, arquivo, 21, 54
vmstat, 563
vmstat, comando, 556, 557
VMware, 568
VNC, protocolo, 567
vncserver, 568
VPNs (virtual private networks), 491
túneis IPSec, 491
túneis SSH, 491
VPNs (Virtual Private Networks), 224, 231, 491
VRFY, comando, 409
VT100, terminal, 591
VXA/VXA-X, fitas de backup, 119

W

wait, chamada de sistema, 41
Wall, Larry, 272
WANs, 246
Ward, Grady, 34
WarGames, 465
Warsaw, Barry, 385
Wassenaar, Eric, 332
Watchguard Firebox, 224
WBEM (Web-Based Enterprise Management) standard, 457
web. *Ver* World Wide Web
Web 2.0, 498
Weeks, Alex, 8
Wheeler, David ., 40
whereis, comando, 11
which, comando, 10
Win4Lin, 569
WINCH, sinal, 42, 43
Windows. *Ver também* Samba
acessando desktops remotos, 567, 568
ACLs, 574
automontador, 574
backups, 140
clientes SSH, 566
compartilhando arquivos, 570
conectando-se a partir de, 566
conformidade com padrões de e-mail e Web, 570
DFS (Distributed File System), 574
encaminhamento X, 567
executando programas Linux a partir de, 567
executando programas Windows sob o Linux, 569
executando sob VMware, 568
executando software UNIX em, 569
IMAP, 570
impressão, 577
inicialização
dual, 569
inicialização múltipla com o Linux, 21, 22
montando sistemas de arquivos Windows, 575
navegação pelo ambiente de rede utilizando o Samba, 572
POP (Post Office Protocol), 570
projeto Wine, 569
RDP (Remote Desktop Protocol), 568
serviço Terminal Server, 568
servidor Kerberos e DNS, 325
servidores VNC, 568
servidores X Window System, 567, 569
sistemas de arquivos FAT, 86
SMTP, 570
xterm para, 569
Wine, projeto, 569
Winmodems, 594
WinPrinters, 540
WinSCP, 566
Wireshark, analisador de pacotes, 257, 456
Wirzenius, Lars, 8
WKS DNS, registros, 282
World Wide Web. *Ver também* hospedagem Web
navegadores, 498
protocolo HTTP, 498, 499
URLs, 498
wtmp, arquivo, 147
WU-FTPD, 620
wvdial, 228
WWW. *Ver* World Wide Web

X

X Cygwin, ferramentas servidor, 569
X Window System. *Ver também* Xorg X, servidor
ambientes de área de trabalho, 523, 525
arquitetura, 513
arquivos de inicialização, 75
autenticação de cliente, 516
cookies mágicos, 516
encaminhamento X, 567
encerrando o servidor X, 522
executando um aplicativo, 515, 517
gerenciador de vídeo, 514, 515
histórico, 513
saída do servidor X, 522, 523
segurança, 515, 516, 517
segurança sob o Windows, 567
servidores Windows, 567, 569
solucionando problemas, 521, 523
SSH e, 516, 517
terminais virtuais, 522
variável de ambiente DISPLAY, 515, 517
X11. *Ver* X Window System
xargs, 52
xauth, 516
.Xclients, arquivo, 75
.Xdefaults, arquivo, 75
xdm, diretório, 514
xdm, programa, 514
xdpyinfo, 523
xdvi, 542
XFS, sistema de arquivos, 87
xhost, 516
XHTML, 528
xinetd, 611, 613
arquivo /etc/services, 614, 615
arquivo /etc/xinetd.conf, 612, 613
configurando, 612, 613
ftpd e, 508
xinit, 514
.xinitrc, arquivo, 75
xntpd, daemon, 45
XON/XOFF, 587, 588
Xorg X, servidor, 518, 521
arquivo /etc/X11/xorg.conf, 518, 521
configurando, 518, 521
depurando, 521
efetuando o login, 522
xorgconfig, 518
XORP (eXtensible Open Router Platform), 242
Xsession, 514
~/.xsession, arquivo, 75, 514
xtab, arquivo, 342, 617
xterm (terminal X Windows), 592
xterm, emulador de console, 569

Y

Yellow Pages. *Ver* NIS
Ylönen, Tatu, 483
Yost, Dave, 586
Yost, sistema de cabeamento serial, 586, 587
yp*, comandos, 362
ypbind, daemon, 617
ypmake, 357
ypserv, daemon, 617
ypxfr, 618
ypxfrd, 618
yum, 174

Z

Zebra, pacote de roteamento, 242
Zhou, Songnian, 264
Zimmermann, Philip, 482
zonas, DNS, 272, 288
  arquivo, 272
  assinadas, criando, 321
  atualizando arquivos, 313, 315
  comandos, 284, 285
  IXFRs, 313
  linkagem, 285, 286
  transferências, 289, 298, 313, 314
  transferências incrementais, 272, 300
zone, instrução, DNS, 302
zone-statistics, opção, DNS, 303
zumbis, processos, 41, 44, 45