Guia do Administrador de Redes Linux

Olaf Kirch

Traduzido pela Conectiva Informática Ltda

Conectiva S/A
http://www.conectiva.com.br

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

```
Kirch, Olaf
   Guia do Administrador de Redes Linux / Olaf Kirch;
   tradução de Conectiva Informática.
   Curitiba : Conectiva, 1999.

   Título original: The Linux Network Administrator's Guide.
   Bibliografia.

1. LINUX
2. Sistemas operacionais (Computador)
3. Redes de computadores
4. UNIX (Sistema operacional de computador)
I. Título

98-4856                         CDD-005.71369
ISBN:  85-87118-01-3
```

Índices para catálogo sistemático:
1. LINUX : Redes de Computadores :
Processamento de dados 005.71369

Conectiva Informática
Rua Rubens Elke Braga, 558
CEP: 80.220.320, Parolin - Curitiba - Paraná

Telefone/Fax: (041) 332-2074

Aos que sonharam e aprenderam a construir...

Aviso Legal

Sumário

Prefácio **1**

Documentação sobre Linux . . . 3

Sobre este livro . . . 4

A Versão Oficial Impressa . . . 5

Mais Informações . . . 6

Sobre os autores . . . 7

Agradecimentos . . . 7

Convenções Tipográficas . . . 9

O Projeto de Documentação do Linux . . . 10

Padrões do Sistema de Arquivos . . . 11

O Guia do Administrador de Redes em Português . . . 11

1 Introdução às Redes **13**

1.1 História . . . 13

1.2 Redes UUCP . . . 14

1.2.1 Como utilizar o UUCP . . . 15

1.3 Redes TCP/IP . . . 17

1.3.1 Introdução a Redes TCP/IP . . . 17

1.3.2 Ethernets . . . 19

1.3.3 Outros Tipos de Hardware 20
1.3.4 O Protocolo Internet 21
1.3.5 IP sobre Linhas Seriais 23
1.3.6 O Protocolo de Controle de Transmissão 23
1.3.7 O Protocolo de Datagrama do Usuário 24
1.3.8 Mais sobre Portas 25
1.3.9 A Biblioteca de Conexão 26
1.4 Redes `Linux` 27
1.4.1 Diferentes Formas de Desenvolvimento 28
1.4.2 Onde conseguir os códigos fontes 28
1.5 Mantendo seu Sistema 29
1.5.1 Sistema de Segurança 30
1.6 Perspectiva dos Capítulos Seguintes 32

2 Redes TCP/IP **35**
2.1 Interfaces de Rede 35
2.2 Endereços IP 36
2.3 Resolução de Endereços 38
2.4 Roteamento IP 39
2.4.1 Redes IP 39
2.4.2 Sub-redes 40
2.4.3 Ponto de Passagem 41
2.4.4 A Tabela de Roteamento 43
2.4.5 Valores de Métrica 45
2.5 O Protocolo de Controle de Mensagens Internet 46
2.6 O Sistema de Nomes de Domínios 47
2.6.1 Resolução de Nomes de Máquinas 47

2.6.2 Entradas DNS 48
2.6.3 Resolução de nomes com DNS 51
2.6.4 Servidor de Nomes do Domínio 52
2.6.5 A Base de Dados DNS 53
2.6.6 Resolução Reversa 55

3 Configurando Hardware de Rede 59

3.1 Dispositivos, Programas de Controle e Outros 59
3.2 Configuração do Kernel 62
3.2.1 Opções do Kernel no `Linux` 1.0 e Acima 63
3.2.2 Opções do kernel no `Linux` 2.0 e Acima 64
3.3 Programas de Controle de Dispositivos de Rede 67
3.4 Instalação Ethernet 68
3.4.1 Cabeamento Ethernet 68
3.4.2 Placas Suportadas 68
3.4.3 Detecção automática da placa Ethernet 70
3.5 O Programa de Controle PLIP 72
3.6 Os Programa de Controle de Dispositivos SLIP e PPP 73

4 Configurando o Hardware Serial 75

4.1 Software de Comunicação para Ligações Via Modem 76
4.2 Introdução sobre Dispositivos Seriais 77
4.3 Acessando Dispositivos Seriais 78
4.4 Hardware Serial 79

5 Configurando Redes TCP/IP 83

5.1 Configurando o Sistema de Arquivos `proc` 84
5.2 Instalando os Binários 85

5.3 Outro Exemplo 85
5.4 Configurando o Nome de Máquina 86
5.5 Definindo Endereços IP 87
5.6 Os Arquivos `hosts` e `networks` 89
5.7 Configuração de Interfaces 91
5.7.1 A Interface Local de Rede 92
5.7.2 Interfaces Ethernet 94
5.7.3 Roteamento Através de um Caminho Padrão 97
5.7.4 Configurando um roteador 98
5.7.5 A interface PLIP 98
5.7.6 A Interface SLIP e PPP 100
5.7.7 A Interface Fantasma 100
5.8 Tudo Sobre o `ifconfig` 101
5.9 Verificação Com o Comando `netstat` 104
5.9.1 Mostrando a Tabela de Roteamento 104
5.9.2 Mostrando as Estatísticas de Interface 105
5.9.3 Mostrando Conexões 106
5.10 Verificando as Tabelas ARP 107
5.11 O Futuro 109

6 Servidor de Nomes e Resolvedor de Endereços 111

6.1 A Biblioteca Resolver 112
6.1.1 O arquivo `host.conf` 112
6.1.2 Variáveis de Ambiente do Resolvedor 114
6.1.3 Pesquisas no Servidor de Nomes — `resolv.conf` 114
6.1.4 A Robustez do Resolvedor 116
6.2 Executando o `named` 117

6.2.1 O arquivo `named.boot` 118
6.2.2 Os Arquivos da Base de Dados do DNS 120
6.2.3 Criando Arquivos Master 124
6.2.4 Verificando a Configuração do Servidor de Nome 127
6.2.5 Outras Ferramentas Úteis 130

7 IP em Linha Serial 131

7.1 Requisitos Gerais 131
7.2 Operação do SLIP 132
7.3 Usando `dip` 134
7.3.1 Um Programa Exemplo 135
7.3.2 Referências `dip` 137
7.4 Executando no Modo Servidor 142

8 O Protocolo Ponto a Ponto 145

8.1 Desvendado os P 145
8.2 PPP no `Linux` 147
8.3 Executando o `pppd` 148
8.4 Usando Arquivos de Opções 149
8.5 Discando Com o Programa `chat` 150
8.6 Depurando a Configuração do PPP 153
8.7 Opções de Configuração IP 154
8.7.1 Escolhendo Um Endereço IP 154
8.7.2 Roteamento Através de Uma Conexão PPP 155
8.8 Opções de Controle de Conexão 157
8.9 Considerações Gerais de Segurança 159
8.10 Autenticação Com PPP 159

8.10.1 CHAP versus PAP 159

8.10.2 O Arquivo de Segredos do CHAP 161

8.10.3 O Arquivo de Segredos do PAP 163

8.11 Configurando um Servidor PPP 164

9 Importantes Funcionalidades de Rede 167

9.1 O Superservidor `inetd` 167

9.2 A Funcionalidade `tcpd` de Controle de Acesso 171

9.3 Os Arquivos `services` e `protocols` 173

9.4 RPC - Chamada de Procedimento Remoto 174

9.5 Configurando os Comandos `r` 176

10 O NIS - Sistema de Informações em Rede 181

10.1 Conhecendo o NIS 182

10.2 NIS versus NIS+ 185

10.3 O Cliente NIS 186

10.4 Servidor NIS 186

10.5 Segurança em Um Servidor NIS 188

10.6 Configurando um Cliente NIS com NYS 189

10.7 Escolhendo os Mapas Corretos 191

10.8 Usando os Mapas `passwd` e `group` 193

10.9 Utilizando NIS com Suporte a Senhas Sombra 195

10.10 Utilizando o Tradicional Código NIS 196

11 O Sistema de Arquivos de Rede 199

11.1 Preparando o NFS 201

11.2 Montando um Volume NFS 202

11.3 Os Servidores NFS 205

11.4 O Arquivo `exports` 206

11.5 O AutoMontador `Linux` 208

12 Gerenciando o Taylor UUCP 209

12.1 História 209

12.1.1 Maiores Informações Sobre o UUCP 211

12.2 Introdução 211

12.2.1 Transportadores UUCP e Execução Remota 211

12.2.2 O Trabalho Interno do `uucico` 213

12.2.3 Opções de Linha de Comando do `uucico` 214

12.3 Arquivos de Configuração UUCP 215

12.3.1 Introdução ao Taylor UUCP 215

12.3.2 O que o UUCP Necessita Saber 219

12.3.3 Nomeando Sites 220

12.3.4 Arquivos de Configuração de Taylor 221

12.3.5 O Arquivo `config` - Opções Gerais de Configuração . . . 222

12.3.6 O Arquivo `sys` - Como Dizer ao UUCP Sobre os Outros Sistemas 222

12.3.7 O Arquivo `port` - O Que São Dispositivos Seriais 228

12.3.8 O Arquivo `dial` - Como Discar 230

12.3.9 UUCP Sobre TCP 232

12.3.10 Usando Uma Conexão Direta 233

12.4 Ajustando Permissões 233

12.4.1 Execução de Comandos 233

12.4.2 Transferência de Arquivos 234

12.4.3 Reenvio 235

12.5 Configurando O Sistema Para o Recebimento de Ligações 236

12.5.1 Configurando `getty` 236
12.5.2 Provendo Contas UUCP 237
12.5.3 Protegendo-se Contra Invasores 239
12.5.4 Verificação de Seqüência de Chamadas - Seja Paranóico 240
12.5.5 UUCP Anônimo 241
12.6 Protocolos UUCP de Transferência 242
12.6.1 Visão Geral do Protocolo 242
12.6.2 Ajustando o Protocolo de Transmissão 244
12.6.3 Selecionando Protocolos Específicos 245
12.7 Problemas & Soluções 246
12.8 Arquivos de Históricos 248

13 Correio Eletrônico 251

13.1 O Que É Uma Mensagem de Correio Eletrônico? 253
13.2 Como Uma Mensagem É Enviada? 255
13.3 Endereço de Correio Eletrônico 257
13.4 Como Funciona o Roteamento de Mensagens? 259
13.4.1 Roteamento de Mensagens Na Internet 259
13.4.2 Roteamento de Mensagens no Mundo UUCP 260
13.4.3 Misturando-se UUCP e RFC 822 262
13.5 Formato dos Arquivos Caminhos Alternativos e Mapas 264
13.6 Configurando o `elm` 267
13.6.1 Opções Globais do Programa `elm` 267
13.6.2 Conjunto de Caracteres Nacionais 268

14 Configurando e Executando o `smail` 271

14.1 Configuração UUCP 272

14.2 Configuração em Uma Rede Local 274
14.2.1 Gravando os Arquivos de Configuração 275
14.2.2 Executando `smail` 277
14.3 Quando as Coisas Não Funcionam... 278
14.3.1 Compilando o `smail` 280
14.4 Modos de Entrega de Mensagens 280
14.5 Opções Diversas do Arquivo `config` 282
14.6 Roteamento de Mensagens e Entrega 282
14.7 Roteando Mensagens 283
14.7.1 A Base de Dados `paths` 286
14.8 Entregando Mensagens para Endereços Locais 286
14.8.1 Usuários Locais 287
14.8.2 Reenvio 288
14.8.3 Aliases de Arquivos 289
14.8.4 Listas de Mensagens 290
14.9 Transportes Baseados em UUCP 290
14.10 Transportes Baseados em SMTP 291
14.11 Definição de Nome de Máquina 292

15 Sendmail+IDA 295

15.1 Introdução ao Sendmail + IDA 295
15.2 Visão Geral do Arquivo de Configuração 296
15.3 O Arquivo `sendmail.cf` 297
15.3.1 Um Exemplo do Arquivo `sendmail.m4` 297
15.3.2 Parâmetros Tipicamente Usados no Arquivo `sendmail.m4` 298
15.4 Um Tour Pelas Tabelas Sendmail+IDA 303

15.4.1 `mailertable` 304

15.4.2 `uucpxtable` 306

15.4.3 `pathtable` 307

15.4.4 `domaintable` 307

15.4.5 `aliases` 308

15.4.6 Tabelas Raramente Utilizadas 309

15.5 Instalando o `sendmail` 310

15.5.1 Extraindo a Distribuição Binária 310

15.5.2 Construindo `sendmail.cf` 311

15.5.3 Testando o Arquivo `sendmail.cf` 312

15.5.4 Integrando Todos os Componentes - Testando o Arquivo `sendmail.cf` e as Tabelas 315

15.6 Dicas de Administração e Outros Detalhes 317

15.6.1 Reenviando Mensagens Para Um Servidor 317

15.6.2 Forçando Mensagens em um Site Mal Configurado . . . 318

15.6.3 Forçando a Transferência de Mensagens Via UUCP . . . 319

15.6.4 Evitando Que Mensagens Sejam Enviadas Via UUCP . . 319

15.6.5 Filas de Mensagens Por Demanda 320

15.6.6 Relatórios de Estatísticas de Mensagens 320

15.7 Misturando Distribuições 321

15.8 Onde Obter Mais Informações 322

16 Notícias na Internet 323

16.1 História da Usenet 323

16.2 O Que é Usenet? 324

16.3 Como a Usenet Lida com Notícias? 326

17 C News 329

17.1 Entregando Notícias 329
17.2 Instalação 331
17.3 O Arquivo `sys` 334
17.4 O Arquivo `active` 337
17.5 Loteando Artigos 339
17.6 Expiração de Notícias 342
17.7 Arquivos Diversos 345
17.8 Mensagens de Controle 347
17.8.1 A Mensagem de `cancelamento` 348
17.8.2 `newgroup` e `rmgroup` 348
17.8.3 A Mensagem `checkgroups` 348
17.8.4 `sendsys`, `version` e `senduuname` 350
17.9 C News em Um Ambiente NFS 350
17.10 Tarefas e Ferramentas de Manutenção 351

18 Descrição do NNTP 355

18.1 Introdução 355
18.2 Instalando O Servidor NNTP 357
18.3 Restringindo o Acesso ao NNTP 358
18.4 Autorização NNTP 360
18.5 `nntpd` Interação com C News 360

19 Configuração do Leitor de Notícias 363

19.1 Configuração do Programa `tin` 364
19.2 Configuração do Programa `trn` 365
19.3 Configuração do Programa `nn` 367

A Cabo Nulo Para PLIP **371**

B Exemplo de Arquivos de Configuração do `smail` **373**

C COMO FAZER - DNS **381**

C.1 Preâmbulo 381

C.1.1 Aspectos Legais 381

C.1.2 Créditos e Pedidos de Ajuda 381

C.1.3 Dedicatória 382

C.2 Introdução. 382

C.3 Um Servidor de Nomes Somente Para Cache 384

C.3.1 Iniciando o named 388

C.4 Um Domínio *Simples*. 389

C.4.1 Mas primeiro um pouco de teoria 390

C.4.2 Nosso Próprio Domínio 394

C.4.3 A zona reversa 402

C.5 Um Exemplo de Domínio Real 404

C.5.1 /etc/named.conf (ou /var/named/named.conf) 405

C.5.2 /var/named/root.hints 406

C.5.3 /var/named/zone/127.0.0 407

C.5.4 /var/named/zone/land-5.com 407

C.5.5 /var/named/zone/206.6.177 409

C.6 Manutenção 411

C.7 Converter da versão 4 para versão 8 412

C.8 Perguntas e Respostas 414

C.9 Como tornar-se um administrador DNS. 417

D Como Fazer - NFS **419**

D.1 Preâmbulo 419
D.1.1 Nota Legal 419
D.1.2 Outros Assuntos 419
D.1.3 Dedicatória 420
D.2 LEIAME Antes! 420
D.3 Configurando um Servidor NFS 421
D.3.1 Pré-Requisitos 421
D.3.2 Primeiros Passos 421
D.3.3 O Portmapper 421
D.3.4 Mountd e nfsd 422
D.4 Configurando um cliente NFS 424
D.4.1 Opções de Montagem 425
D.4.2 Otimizando NFS 426
D.5 NFS Sobre Linhas de Baixa Velocidade 427
D.6 Segurança e NFS 430
D.6.1 Segurança do Cliente 431
D.6.2 Segurança no Servidor: nfsd 431
D.6.3 Segurança no Servidor: o portmapper 432
D.6.4 NFS e Firewalls 434
D.6.5 Resumo 434
D.7 Pontos de Verificação de Montagem 435
D.8 FAQ 436
D.9 Exportando Sistemas de Arquivos 438
D.9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX 438
D.9.2 Solaris 2 439

D.10 PC-NFS 439

E Licença Pública GNU **441**

E.1 Introdução 441

E.1.1 Termos e Condições 442

E.2 Como Aplicar Estes Termos 447

E.3 BSD Copyright 448

E.3.1 X Copyright 449

Glossário **451**

Bibliografia **459**

Livros 459

Livros sobre a Internet 459

Administração 459

Suporte 462

Como Fazer 463

O que são os Como Fazer `Linux`? 463

Onde Obter os Como Fazer do `Linux` 463

Índice dos Como Fazer Disponíveis 464

Itens Diversos e Notícias Legais 483

RFCs 483

Lista de Figuras

1.1 Os três passos para se enviar um datagrama de `jacare` a `jaburu` 22

2.1 Criando sub-redes em uma rede classe B 41

2.2 Parte da topologia de rede da Universidade do Pantanal 43

2.3 Parte do Espaço de Nome de Domínio 49

3.1 O relacionamento entre programas de controle, interfaces e o hardware 60

5.1 Cervejaria e Vinícola Virtuais – duas sub-redes 89

12.1 Interação dos Arquivos de Configuração do Taylor UUCP 218

16.1 Fluxo de noticias através da Universidade do Pantanal 326

17.1 Fluxo de Notícias Através do `relaynews` 331

Prefácio

Com o alarde feito após o advento da Internet e com todo tipo de pessoas e empresas e outras entidades presentes e navegando pela “Super Estrada da Informação”, as redes de computadores parecem estar se movendo na direção de terem um status de aparelhos de TV e fornos de microondas. A Internet está tendo uma cobertura incomum na mídia e as ciências sociais estão chegando a grupos de discussão da Internet para conduzir pesquisas sobre a “Cultura da Internet.” Companhias de telecomunicações estão trabalhando para introduzir novas técnicas de transmissão, como por exemplo ATM que oferece uma largura de banda muitas vezes maior que a conexão média disponível hoje em dia.

Naturalmente, as redes existem há um longo tempo. Conectar computadores para formar uma rede local é uma prática comum, mesmo em pequenas instalações, assim como conexões de longo alcance utilizando linhas públicas de telefone ou linhas dedicadas de dados. Um conglomerado de redes de alcance mundial tem, no entanto, possibilitado ligações entre a aldeia global e viabilizado acessos mesmo para pequenas organizações sem fins lucrativos ou até mesmo para usuários particulares. Como a configuração de um servidor Internet com funcionalidades de correio eletrônico e notícias, oferecendo acesso discado a qualquer usuário, tornou-se algo ao alcance de qualquer usuário, e com o advento do ISDN[1], esta tendência mantém um ritmo acelerado de crescimento.

Falar sobre redes de computadores freqüentemente significa falar sobre UNIX. Naturalmente o UNIX não é o único sistema operacional com funcionalidades de rede, nem será sempre somente o servidor, mas está presente no ramo de redes há um longo tempo e certamente continuará a estar por ainda muito mais tempo.

O que torna esta plataforma interessante para usuários particulares é a atividade que tem havido para trazer sistemas operacionais tipo UNIX gratuitos para o PC, como os 386BSD, FreeBSD — e `Linux`. Embora o `Linux` *não* seja um UNIX no sentido mais comercial da palavra, ele é compatível com os padrões abertos definidos para aquele sistema. Unix é uma marca registrada de quem quer que mantenha os direitos sobre ela (Univel, no momento em que estou escrevendo este Guia), enquanto `Linux` é um sistema operacional que busca oferecer toda a funcionalidade dos padrões POSIX necessários a um sistema operacional tipo UNIX, como uma completa reimplementação deste.

O kernel do `Linux` foi escrito, na sua maior parte, por Linus Torvalds, que iniciou este trabalho como um projeto pessoal para aprimorar o seu conhecimento do processador Intel i386, e para "desenvolver um MINIX melhor que o disponível nos idos de 1991." MINIX era então outro popular sistema operacional para PC, oferecendo ingredientes vitais da funcionalidade do `Unix`, escrito pelo professor Andrew S. Tanenbaum.

O `Linux` está sob a Licença Pública Geral (GPL) GNU, a qual permite livre distribuição do código (por favor leia o GPL no apêndice E para uma definição do significado de "software de livre distribuição"). Em um crescente meteórico e com uma grande e sempre em expansão base de aplicações gratuitas, Linux tornou-se a escolha para milhões de usuários de PC. O kernel e a biblioteca C são tão boas que a maioria dos softwares padrão com pouco esforço, algo não conhecido em qualquer outro conhecido sistema `Unix`, pode ser portado com relativa facilidade para esta plataforma. Uma grande quantidade de distribuições empacotadas do `Linux` permitem que a sua utilização requeira somente a sua instalação em disco rígido.

[1] N.T. Que começou a ser disponibilizado recentemente no Brasil

Documentação sobre Linux

Uma das freqüentes necessidades que são levantadas sobre Linux (e softwares gratuitos de maneira geral) é a carência ou total ausência de documentação. No início não era raro um pacote vir repleto de arquivos LEIAME[2] e instruções de instalação. Eles deram ao usuário mais experiente do Unix informações suficientes para instalar o software com sucesso e poder executá-lo, mas deixou o usuário menos experiente em maus lençóis.

Voltando ao final de 1992, Lars Wirzenius e Michael K. Johnson sugeriram a criação do Projeto de Documentação do Linux, ou LDP, o qual tem como objetivo prover um conjunto coerente de manuais. Pequenas pausas para responder questões como "Como?", ou "Por quê?", ou "Qual o significado da vida, universo e todo o resto?" ocorreram, porém como resultado final buscou produzir estes manuais, que tentam cobrir a maioria dos aspectos de execução e utilização de um sistema Linux, mesmo para usuários iniciantes.

Entre os objetivos alcançados pelo LDP estão o *Guia de Instalação e Utilização do Linux*, escrito por Matt Welsh, o *Guia do Kernel* escrito por Michael K. Johnson e o projeto de páginas de manual on-line coordenado por Rik Faith, o qual até agora nos forneceu um conjunto de aproximadamente 450 páginas de manual para a maioria dos sistemas e chamadas a bibliotecas C. O *Guia de Administração do Sistema Linux*, escrito por Lars Wirzenius, já encontra-se traduzido e disponível em português, distribuído pela Conectiva Informática Ltda.

Este livro, o *Guia do Administrador de Redes Linux*, é também parte da série do LDP.

Entretanto, os livros do LDP não são a única fonte de informação sobre Linux. Até o momento, há mais de meia centena de documentos chamados "Como Fazer"[3] que são postados regularmente em comp.os.linux.announce e arquivados em vários sites FTP. Os documentos "Como Fazer" são pequenos documentos de poucas páginas que nos dão uma breve introdução sobre tópicos como suporte Ethernet sob Linux, ou a configuração de um software de notícias Usenet, bem como respondem à perguntas mais freqüentes. Estes normalmente fornecem a mais atualizada e precisa informação disponível sobre o tópico. Uma lista dos "Como fazer" disponíveis está reproduzida na "Bibliografia" anotada no final deste livro.

No Brasil, diversas iniciativas de tradução dos "Como Fazer" foram iniciadas, sen-

[2] README
[3] Conhecidos como HOWTOs

do que a Conectiva Informática está disponibilizando dezenas de "Como Fazer" via Internet ou em sua publicação denominada "Guia do Servidor Linux". Os documentos "Como Fazer" podem ser encontrados em `http://ldp-br.conectiva.com.br/documentos/comofazer/html/HOWTO-INDEX.html`.

Sobre este livro

Quando me juntei ao Projeto de Documentação do `Linux` em 1992, escrevi dois pequenos capítulos sobre UUCP e `smail`, através dos quais pretendia contribuir com o Guia do Administrador de Sistemas. O desenvolvimento de redes TCP/IP estava apenas começando e quando aqueles "pequenos capítulos" começaram a crescer, comecei a imaginar se não seria uma boa opção ter-se um Guia de Redes. "Boa", "Vai nessa", todos disseram. Iniciei-o então e escrevi uma pequena versão do Guia de Rede, que foi lançada em Setembro de 1993.

O Guia de Redes que você está lendo agora é uma completa reedição, a qual contém algumas novas aplicações que se tornaram disponíveis para usuários de `Linux` após o primeiro lançamento.

O livro está organizado em uma seqüência de passos que poderão ser seguidos para configurar um sistema para trabalhar em rede. Começa discutindo os conceitos básicos de redes, e particularmente as redes baseadas em TCP/IP. Pausadamente apresentamos o caminho desde a configuração dos dispositivos TCP/IP até instalação de aplicações como `rlogin` e similares, o Sistema de Arquivos de Rede e o Sistema de Informações em Rede. Estes são seguidos por um capítulo sob como instalar sua máquina como um nó UUCP. Uma boa parte deste livro é dedicado às duas mais importantes aplicações que rodam sobre TCP/IP e UUCP: correio eletrônico e notícias.

A parte de correio eletrônico contém uma introdução às mais internas formas de transporte de correio e roteamento, e os esquemas de endereçamento que podem ser encontrados. Este descreve a configuração e gerenciamento do `smail`, um agente de transporte de correio utilizado normalmente em sites menores, e do `sendmail`, este para pessoas que desejem fazer roteamentos mais complexos, ou têm que trabalhar com um grande volume de mensagens. O capítulo referente ao `sendmail` é uma contribuição de Vince Skahan.

A parte de notícias tem como objetivo fornecer uma visão geral de como os grupos de notícias Usenet funcionam, cobrindo C news, o mais largamente utilizado software de transporte de notícias e o uso do NNTP que fornece acesso à leitura de notícias em uma rede local. O livro fecha com um pequeno capítulo citando os mais populares leitores de notícias em `Linux`.

A Versão Oficial Impressa

No outono de 1993, Andy Oram, que esteve navegando pelas listas de correio do LDP quase desde o início, solicitou a publicação do meu livro na O'Reilly e Associados. Eu estava animado com isto; eu nunca tinha imaginado meu livro como sendo de tanto sucesso. Nós finalmente concordamos que a O'Reilly produziria uma versão melhorada da Versão Oficial Impressa do Guia de Rede, enquanto eu reteria os direitos autorais do original, garantindo assim que a fonte do livro pudesse ser livremente distribuída.[4] Isto significa que se pode escolher livremente: obter a fonte LaTeX distribuída na rede (ou a versão pré-formatada DVI ou PostScript, para este propósito) e imprimí-la em uma impressora local, ou adquirir a versão oficial impressa ou outras autorizadas.

Então, por que pagar por algo que pode ser obtido gratuitamente? Tim O'Reilly estava louco ao publicar algo que todo mundo pode imprimir? Ou há alguma diferença entre as versões?

A resposta é "depende," "não, definitivamente não," e "sim e não". Caso este projeto seja superavitário, eu acredito que servirá como um exemplo de como o mundo do software gratuito e as companhias podem cooperar para produzir algo de que todos se beneficiem. Na minha visão, o grande serviço que a O'Reilly está fazendo para a comunidade Linux (apesar do livro estar disponibilizado na sua loja mais próxima) é que este poderá ajudar o Linux a ser reconhecido como algo que possa ser encarado corporativamente: uma alternativa viável e útil para os sistemas operacionais comerciais PC.

E então a respeito das diferenças entre a versão impressa e a versão on-line? Andy Oram fez grandes esforços para transformar os meus primeiros rabiscos em algo que valesse a pena ser impresso. (Ele também tem revisado os outros livros a serem apresentados no Projeto de Documentação do Linux, tentando contribuir com a comunidade Linux).

Desde que Andy iniciou a revisão do Guia de Redes e editou as cópias que eu lhe enviei, o livro tem melhorado vastamente daquilo que era há meio ano atrás. Não

[4] O aviso de direitos autorais está reproduzido na página imediatamente seguinte ao título.

estaria perto de lugar algum onde está agora sem a sua contribuição.

O mesmo é verdade sobre Stephen Spainhour, que esteve copiando e editando o livro por quase um mês para que ele ficasse com a forma que está agora. Todas estas edições foram alimentadas na versão on-line, então não há diferença no conteúdo. Ainda, a versão da O'Reilly *é* diferente: por uma lado, o pessoal da O'Reilly trabalhou bastante na aparência, produzindo um layout mais agradável do que poderia ser obtido no padrão LaTeX. Entre outras coisas, Chris Reilley gentilmente refez todas as figuras da versão original e adicionou algumas figuras extras. Ele fez um grande trabalho ao melhorar consideravelmente o que eu originalmente quis dizer com os meus desenhos amadores feitos no `xfig`.

Mais Informações

Se você seguir as instruções deste livro, e de maneira alguma algo funcionar, por favor seja paciente. Alguns de seus problemas podem ser devidos a erros estúpidos cometidos por mim, mas também podem ser causados por mudanças nos softwares de rede. Portanto, primeiramente pergunte a `comp.os.linux.help`[5]. Há uma boa chance de que você não seja o único com esse problema, então uma solução ou ao menos uma proposta para tal pode ser conhecida. Se você tiver oportunidade, poderá obter o último kernel e a última versão de rede de um dos sites de FTP do `Linux`. Muitos problemas são causados por softwares em diferentes estágios de desenvolvimento, os quais falham ao trabalharem juntos. Porém lembre-se `Linux` é "trabalho em progresso".

Outro bom lugar para se informar sobre os desenvolvimentos correntes é o "Como Fazer - Redes". Ele é mantido por Terry Dawson[6]. Este é atualizado e divulgado em `comp.os.linux.announce` uma vez por mês, e contém as mais atualizadas informações. A versão corrente pode ser obtida (além de outras) em `tsx-11.mit.edu`, no caminho `/pub/linux/doc`. Para problemas que não possam ser resolvidos de forma alguma, pode-se ainda contatar o autor deste livro no endereço dado no prefácio. Porém não se acanhe em pedir ajuda aos desenvolvedores. Eles já estão devotando a maior parte do seu tempo ao `Linux`, e ocasionalmente têm uma vida além da rede `:-)`.

[5] Pode-se ainda utilizar as listas disponíveis em `http://listas.conectiva.com.br/listas`

[6] Terry Dawson pode ser contatado em `terryd@extro.ucc.su.oz.au`

Sobre os autores

Olaf tem sido um usuário de UNIX e administrador de redes em meio período por alguns anos quando estudava matemática. No momento, ele está trabalhando como um programador UNIX e escrevendo um livro. Um de seus esportes favoritos é fazer coisas em `sed` que possam ser utilizados por outras pessoas em seu interpretador `perl`. Ele se diverte tanto com isto como faz caminhadas e acampamentos nas montanhas.

Vince Skahan administrou um grande número de sistemas UNIX desde 1987 e atualmente executa sendmail+IDA em aproximadamente 300 estações de trabalho UNIX para mais de 2.000 usuários. Ele admite que perdeu consideráveis horas de sono editando alguns arquivos `sendmail.cf` da 'maneira difícil' antes de descobrir o sendmail+IDA em 1990. Ele também admite que está aguardando ansiosamente pela entrega da primeira versão do sendmail baseada em perl para ainda mais divertimento[7]...

Olaf pode ser contactado nos seguintes endereços:

Olaf Kirch
Kattreinstr. 38
64295 Darmstadt
Germany

`okir@monad.swb.de`

Vince pode ser contactado nos seguintes endereços:

Vince Skahan
`vince@victrola.wa.com`

Nós estamos abertos para suas perguntas, comentários, mensagens, etc., porém pedimos que *não* nos telefonem, a menos que seja realmente muito importante.

Agradecimentos

Olaf agradece: Este livro deve muito às numerosas pessoas que despenderam seu tempo a nos auxiliarem na correção de erros, técnicos e gramaticais (nunca tomei

[7] Vince, você não acha que nós poderíamos fazer isto com `sed`?

conhecimento de algo chamado particípio). O mais vigoroso entre eles foi Andy Oram da O'Reilly e Associados.

Eu também estou em dívida com Andres Sepúlveda, Wolfgang Michaelis, Michael K. Johnson, e todos os desenvolvedores que usaram o seu tempo para checar as informações fornecidas neste Guia de Rede. Eu também gostaria de agradecer a todos aqueles que leram a primeira versão do Guia de Redes e enviaram correções e sugestões. É possível encontrar uma lista completa de contribuidores no arquivo `Agradecimentos` na distribuição on-line. Finalmente este livro não seria possível sem o suporte de Holger Grothe.

Eu também gostaria de agradecer aos seguintes grupos e companhias que imprimiram a primeira edição do Guia de Redes e doaram dinheiro ou para minha pessoa, ou para o Projeto de Documentação do `Linux` como um todo.

- Linux Support Team, Erlangen, Alemanha

- S.u.S.E. GmbH, Fuerth, Alemanha

- Linux System Labs, Inc., Estados Unidos

Vince agradece: Agradecimento a Neil Rickert e Paul Pomes por sua ajuda por todos estes anos nas implementações do sendmail+IDA e a Rich Braun por fazer a conexão inicial do sendmail+IDA para `Linux`. O maior agradecimento até agora vai para minha esposa Susan por todo o seu suporte neste e em outros projetos.

Convenções Tipográficas

Ao escrever este livro, um número de convenções tipográficas foram empregadas para marcar os comandos no interpretador de comandos, variáveis, etc., as quais são explicadas a seguir.

Negrito Usada para marcar endereços de servidores e de correio eletrônico, bem como novos conceitos e avisos.

Itálico Usada para marcar nomes de arquivos, comandos UNIX, e teclas chave nos arquivos de configuração. Também usada para *enfatizar* textos.

`Courier` Usada para representar interações na tela, como entrada de informações pelo usuário ao executar um programa. Também usada para exemplos de código, no caso de um arquivo de configuração, um programa interpretado, etc..

`Courier Slanted` Usada para marcar variáveis "meta" no texto, especialmente nas representações de linhas de comando. Por exemplo,

```
$ ls l teste
```

onde *teste* deverá ser substituída por um nome de diretório, como por exemplo `/tmp`.

`Tecla` Representa uma tecla a ser pressionada. Você freqüentemente a verá nesta forma:

Pressione `enter` para continuar .

◇ Uma pequena bomba na margem, significa "perigo" ou "cuidado". Leia os parágrafos assim marcados cuidadosamente.

$ e # Quando precedidas de um comando interpretado a ser digitado, denotam uma linha do interpretador de comandos. O símbolo '`$`' é usado quando o comando for executado como um usuário normal; '`#`' significa que o comando requer privilégios de superusuário.

O Projeto de Documentação do Linux

O Projeto de Documentação do `Linux` ou LDP, é constituído por uma equipe livre de escritores, leitores e profissionais que estão trabalhando juntos para fornecer uma completa documentação do sistema operacional `Linux`. O coordenador geral deste projeto é Matt Welsh, que está sendo auxiliado fortemente por Lars Wirzenius e Michael K. Johnson.

Este manual é um de uma série a ser distribuída pelo LDP, incluindo o Guia de Usuários do Linux, o Guia do Administrador de Sistemas[8], o Guia do Administrador de Redes, e o Guia do Kernel. Estes manuais estarão disponíveis no formato fonte LaTeX no formato `.dvi`, e a saída postscript em FTP anônimo em `nic.funet.fi`, no diretório `/pub/OS/Linux/doc/doc-project`, e em `tsx-11.mit.edu`, no diretório `/pub/linux/docs/guides`.

Nós encorajamos qualquer pessoa com habilidades para escrita que se junte a nós, melhorando esta documentação do Linux. Se você tiver acesso a um email da Internet, você pode acessar o canal `DOC` da lista de correio `Linux-Activists` enviando uma mensagem para

```
linux-activists-request@niksula.hut.fi
```

com a linha

```
X-Mn-Admin: join DOC
```

no cabeçalho ou como a primeira linha do corpo da mensagem. Uma mensagem vazia sem linha de cabeçalho adicional fará o servidor de correio retornar uma mensagem de ajuda. Para deixar o canal, envie uma mensagem para o mesmo endereço incluindo a linha

```
X-Mn-Admin: leave DOC
```

[8] Traduzido e disponível nas versões impressas e on-line em `http://www.conectiva.com.br`

Padrões do Sistema de Arquivos

No passado, um dos problemas que atingiu as distribuições do Linux além da coleção de pacotes, era a falta de um único padrão de sistema de arquivos. Isto resultou em incompatibilidades entre diferentes pacotes, e desafiou usuários e administradores na tarefa de instalar vários arquivos e programas.

Para melhorar esta situação, em Agosto de 1993, algumas pessoas formaram o Grupo do Padrão do Sistema de Arquivos, ou o grupo FSSTND (em inglês) para abreviar, coordenado por Daniel Quinlan. Após seis meses de discussões, o grupo apresentou uma proposta do que parecia ser uma estrutura de sistema de arquivos coerente e definiu a localização dos programas essenciais e dos arquivos de configuração.

Este padrão foi implementado pela maioria das distribuições e pacotes do Linux[9]. Através deste livro, iremos assumir que qualquer arquivo discutido reside no local especificado por este padrão; apenas onde houver uma longa tradição que conflite com as especificações, serão então mostradas localizações alternativas.

O Padrão de Sistema de Arquivos do Linux pode ser obtido na maioria dos sites de FTP do Linux ou em seus espelhos; de imediato, ele pode ser encontrado em `metasite.unc.edu`, sob o caminho `/pub/linux/docs`. Daniel Quinlan, o coordenador do FSSTND pode ser contatado em `quinlan@bucknell.edu`. Uma outra alternativa, em português, pode ser o resumo deste trabalho, disponível no Manual do Usuário do Conectiva Linux, disponível em `http://www.conectiva.com.br`.

O Guia do Administrador de Redes em Português

É com prazer e imensa satisfação que a Conectiva Informática traduziu este documento, disponibilizando este Guia em formato impresso ou para livre recepção via Internet. Esperamos assim contribuir com o imenso esforço de prover informações e meios para que o Linux possa cumprir o seu papel nos quatro cantos do planeta.

Esta primeira tradução faz parte de uma trilogia composta pelo Guia do Administrador de Sistemas Linux, também traduzido do LDP e do Guia do Servidor Linux, uma coletânea dos principais documentos "Como Fazer", além de um guia da interface gráfica de configuração de serviços no Linux denominada Linuxconf. Todos foram atualizados, comentados pela Conectiva, e foi incluída ainda parte

[9] O Conectiva Linux segue o padrão FSSTND

da documentação da versão Conectiva Linux - Edição Servidor.

Caso você encontre qualquer incorreção ou tenha sugestões sobre esta publicação, por favor entre em contato conosco através do email `doc@conectiva.com.br`. A Conectiva pode ser localizada ainda no seguinte endereço:

Conectiva Informática
R. Prof. Rubens Elke Braga, 558
Parolin
80220-320 Curitiba(PR)
Brasil

Fone/Fax 41 332 2074

Equipe de Desenvolvimento Conectiva Linux

Capítulo 1

Introdução às Redes

1.1 História

A idéia de redes é provavelmente tão velha quanto as telecomunicações. Considere as pessoas vivendo na idade da pedra, onde tambores eram usados para transmitir mensagens entre indivíduos. Imagine que o homem das cavernas A quer convidar o homem das cavernas B para um jogo de arremesso de pedras um no outro, mas ele vive muito longe de B para que este ouça A bater no tambor. Então qual são as opções de A? Ele poderia 1) andar até a casa de B, 2) comprar um tambor maior, ou 3) pedir a C, que vive no meio do caminho entre os dois, para passar a mensagem adiante. Esta última opção é chamada de rede de comunicação.

Naturalmente houve um longo caminho desde as atividades primitivas e os dispositivos dos nossos antepassados. Hoje em dia, temos computadores conversando um com o outro sobre uma vasta construção de cabos, fibras óticas, microondas, e similares fazendo contatos para um jogo de futebol no Sábado.[1] Na seqüência, vamos tratar dos significados e caminhos nos quais isto é feito, mas deixemos de fora os cabos, bem como a parte do futebol. Nós descreveremos dois tipos de redes neste guia: aquelas baseadas em UUCP, e aquelas baseadas em TCP/IP. Estes são protocolos e pacotes de software que fornecem meios de transporte de dados entre computadores. Neste capítulo, descreveremos ambos os tipos de redes, e discutiremos os princípios de suas camadas fundamentais.

[1] O espírito original deste jogo ainda ocorre em alguns lugares da Europa.

Definimos rede como uma coleção de *máquinas* que são capazes de se comunicarem umas com as outras, freqüentemente confiando em serviços de máquinas dedicadas que reenviam dados para os participantes. Estações são freqüentemente computadores, mas não necessariamente. Podemos pensar também em terminais X ou impressoras inteligentes como estações de rede. Pequenas aglomerações de máquinas são também chamadas de *sites*.

Comunicação é impossível sem algum tipo de linguagem ou código preestabelecida. Nas redes de computadores, estas linguagens são coletivamente chamadas de *protocolos*. Não se deve pensar em protocolos escritos, mas sim em códigos de comportamento altamente formalizados, observados quando, por exemplo, chefes de Estado se encontram. De uma forma bem similar, protocolos utilizados em redes de computadores são nada mais que regras bem restritas para a troca de mensagens entre dois ou mais equipamentos.

1.2 Redes UUCP

UUCP é uma abreviação para "Cópia Unix para Unix"[2]. Foi iniciado como um pacote de programas para transferir arquivos em linhas seriais, agendar estas transferências e realizar a execução de programas em sites remotos. Foi submetido a grandes mudanças desde sua primeira implementação, no final dos anos 70, mas ainda é espartano nos serviços que oferece. Sua aplicação é ainda em redes geograficamente distribuídas baseadas em conexões telefônicas do tipo discado.

O UUCP foi primeiramente desenvolvido pelos Laboratórios da Bell em 1977, para a comunicação entre os seus sites de desenvolvimento UNIX. Em meados de 1978, esta rede era composta por mais de 80 sites. Executava email como aplicação, bem como impressão remota, embora o uso central do sistema fosse a distribuição de novos softwares e correção de problemas.[3] Hoje, o UUCP não está mais confinado ao ambiente `Unix`. Há portes disponíveis tanto gratuitos como comerciais para uma variedade de plataformas, incluindo AmigaOS, DOS, TOS da Atari, etc..

Uma das principais desvantagens das redes UUCP é sua baixa largura de banda. Por um lado o equipamento telefônico coloca um limite justo na taxa máxima de transferência. Por outro lado, as conexões UUCP são raramente permanentes; pelo contrário, os servidores preferem discar uns para os outros em intervalos regulares. Por esta razão, a maior parte do tempo gasto para que uma mensagem de email

[2] Unix-to-Unix Copy
[3] Não que isto tenha mudado muito...

trafegue em uma rede UUCP é gasta com a mensagem aguardando em algum disco de servidor a próxima conexão a ser estabelecida.

Apesar destas limitações, ainda há muitas redes UUCP operando por todo o mundo, as quais oferecem a usuários particulares acesso a redes maiores por preços razoáveis. A principal razão para a popularidade do UUCP reside em ele ser bem mais barato do que se ter um computador conectado à Internet. Para fazer de um computador um nó UUCP, tudo o que se precisa é um modem, uma implementação rodando UUCP, e outro nó UUCP que alimente o servidor local com correio eletrônico e notícias.

1.2.1 Como utilizar o UUCP

A idéia por trás do UUCP é bem simples: e como seu nome indica, basicamente copia arquivos de um servidor para outro, porém permite também certas ações a serem realizadas no servidor remoto.

Imagine que uma máquina local está permitindo acesso a um servidor hipotético chamado `piraquara`, o qual deve executar o comando de impressão `lpr`. Pode-se então simplesmente digitar o seguinte na linha de comandos, para por exemplo ter um arquivo impresso por `piraquara`:[4]

```
$ uux -r piraquara!lpr !guia.ps
```

`uux` é um comando, do conjunto de comandos disponíveis no UUCP, que define uma *tarefa* para `piraquara`. Esta tarefa consiste na transferência do arquivo `guia.ps`, e na solicitação para enviar este arquivo para o comando de impressão `lpr`. O indicador `-r` diz a `uux` para não acionar o sistema remoto imediatamente, mas sim que armazene a tarefa em outro lugar até que a conexão seja estabelecida posteriormente. Isto é chamado de *armazenamento de tarefas temporárias*[5].

Outra propriedade do UUCP consiste na possibilidade de transferir tarefas adiante através de diversos servidores, contando com a sua cooperação. Por exemplo, presumindo-se que a máquina `piraquara` do exemplo acima tenha uma conexão UUCP com `pantanal`, a qual mantém um grande arquivo de aplicações `Unix`. Para que a máquina local receba o arquivo `conec-1.0.tar.gz`, através de `pantanal` deve-se executar o seguinte comando:

[4] Quando se estiver utilizando o `bash`, o GNU Bourne Again Shell, não se deve utilizar o caractere de exclamação, porque este é usado como caractere de histórico.

[5] spooling

```
$ uucp -mr piraquara!pantanal!~/security/conec-1.0.tar.gz conec.tgz
```

O serviço criado solicitará a `piraquara` que traga o arquivo disponível em`pantanal`, e o envie a seu site, onde o UUCP o armazenará em `conec.tgz` e notificará o usuário via correio sobre a chegada do arquivo. Isto é feito em três passos. Primeiro o site envia a tarefa a `piraquara`. Da próxima vez que `piraquara` estabelecer contato com `pantanal`, o arquivo será transferido. O passo final é a transferência de `piraquara` para a máquina do usuário.

Os mais importantes serviços fornecidos pelo UUCP hoje em dia são correio eletrônico e notícias. Retornaremos a estes mais tarde, porém forneceremos agora uma pequena introdução. Correio eletrônico – ou email para abreviar – é o serviço de rede que permite a troca de mensagens com usuários em servidores remotos, sem a necessidade de sabere como acessá-los. O serviço de direcionamento de uma mensagem do site local ao site de destino é realizado inteiramente pelo sistema de tratamento de correio. Em um ambiente UUCP, o correio é freqüentemente transportado utilizando-se o comando `rmail` em um servidor próximo, passando a este o endereço do receptor e a mensagem de correio. O `rmail` remeterá então a mensagem ao servidor seguinte, e assim por diante, até que este atinja o servidor de destino. Este tema será detalhado no capítulo 13.

Notícias podem ser descritas como um sistema de distribuição de boletins. Porém freqüentemente, este termo refere-se a Notícias Usenet, esta até agora a mais largamente conhecida rede de troca de notícias com um número estimado de 120.000 sites participantes. As origens do Usenet remontam ao ano de 1979, quando após o lançamento do UUCP com o novo Unix V7, três estudantes tiveram a idéia de criarem um sistema de troca de informações dentro da comunidade Unix. Fizeram isso juntando alguns programas, os quais tornaram-se o primeiro sistema de notícias de rede. Em 1980, esta rede conectou `duke`, `unc`, e `phs`, a duas Universidades na Carolina do Norte. Embora originada em uma rede baseada em UUCP, não está mais confinada a um único tipo de rede.

A unidade básica de informação é o artigo, o qual pode ser postado em uma hierarquia de grupos dedicados a tópicos específicos. A maioria dos sites recebe apenas uma seleção de todos os grupos, os quais carregam em média 60 MB de arquivos por dia.

No mundo UUCP, notícias são geralmente enviadas por uma conexão UUCP coletando-se todos os artigos de todos os grupos solicitados, e empacotando-os através de *execuções programadas*. Estes são enviados ao site receptor, onde eles são enviados para o comando `rnews` para desempacotamento e processamento futuro.

Finalmente, o UUCP é uma opção para muitos sites que sejam repositórios de arquivos e que ofereçam acesso público. É possível acessá-los através de linhas discadas e com o UUCP, identificando-se como um visitante e transferindo informações de acesso público a partir da área de arquivos. Estas contas de visitantes freqüentemente têm um nome de acesso e senha similares a `uucp/nuucp`.

1.3 Redes TCP/IP

Embora o UUCP pode ser uma escolha razoável para uma rede discada de baixo custo, há muitas situações nas quais técnicas de armazenamento e envio se provam inflexíveis, como por exemplo em Redes Locais (LANs). Estas são normalmente feitas de um pequeno número de máquinas localizadas no mesmo local, ou ainda no mesmo andar, interconectadas a fim de fornecerem um ambiente de trabalho homogêneo. Tipicamente se faz uma troca de arquivos entre os servidores, ou se executam aplicações distribuídas em máquinas diferentes.

Estas tarefas requerem um enfoque diferente de rede. Ao invés de se enviar arquivos inteiros adiante com uma descrição da tarefa, todos os dados são quebrados em pequenos pedaços (pacotes), os quais são enviados imediatamente ao servidor de destino, onde eles são remontados. Este tipo de rede é chamada de *rede de troca de pacotes*. Entre outras coisas esta permite rodar aplicações interativas sobre a rede. O custo disto é, naturalmente, um grande aumento na complexidade do software.

A solução que sistemas `Unix` e muitos sistemas não `Unix` têm adotado é conhecida como TCP/IP. Nesta seção, daremos uma visão geral destes conceitos.

1.3.1 Introdução a Redes TCP/IP

O TCP/IP tem sua origem em um projeto de pesquisa fundado pelos Estados Unidos chamado DARPA (Agência de Projetos Avançados de Pesquisa de Defesa) em 1969. A ARPANET foi uma rede experimental, a qual foi convertida em uma rede operacional em 1975, após ser provado o seu sucesso.

Em 1983, o novo conjunto de protocolos TCP/IP foi adotado como um padrão, e todos os servidores na rede deveriam passar a utilizá-lo. Quando a ARPANET finalmente cresceu e se tornou a Internet (resultando na finalização de sua existência em 1990), o uso do TCP/IP espalhou-se a diversas redes muito além da Internet. As mais notáveis são as redes locais `Unix`. Porém com o advento de equipamentos telefônicos digitais mais rápidos, como o ISDN, o TCP/IP tem ainda um futuro promissor como protocolo de transporte para redes discadas.

Como algo concreto para comentarmos enquanto discutimos o TCP/IP através das seções seguintes, nós consideraremos a fictícia Universidade do Pantanal, situada em algum lugar na Região Centro-Oeste como nosso exemplo. A maioria dos departamentos têm a sua rede local própria, enquanto outros dividem uma, e outros ainda possuem muitas redes. Elas estão todas interconectadas e estão ligadas a Internet através de uma única conexão de alta velocidade.

Imaginemos uma máquina `Linux` conectada a uma LAN com outras máquinas `Linux` no Departamento de Matemática, e que seu nome seja `jacare`. Para acessar um servidor no Departamento de Física, digamos `jaburu`, deve-se informar o seguinte comando:

```
$ rlogin jaburu.fisica
Bem-Vindo ao Departamento de Física da Universidade do Pantanal
(ttyq2) login:
```

Na linha de comando, deve-se informar o nome de acesso, digamos `lroberto` e sua senha. A seguir, caso o acesso seja permitido, você estará utilizando um interpretador de comandos em `jaburu`, no qual se pode digitar como se estivesse à frente da console do sistema. Após sair do interpretador de comandos, volta-se à linha de comandos da máquina local. Neste caso foi utilizada somente uma das aplicações interativas e instantâneas que o TCP/IP fornece: o acesso remoto.

Enquanto se estiver conectado a `jaburu`, pode-se também executar uma aplicação baseada em X11 (em modo gráfico), como por exemplo um programa para imprimir funções ou um revisor de PostScript. Para avisar esta aplicação que se deseja ter suas janelas apresentadas na tela do servidor, deve-se ajustar a variável de ambiente `DISPLAY`:

```
$ export DISPLAY=jacare.mat:0.0
```

Ao se iniciar agora uma aplicação, esta contatará o servidor X da máquina remota ao invés do servidor X de `jaburu`, e mostrará todas as janelas na tela da máquina remota. Naturalmente, isto requer que se tenha X11 rodando em `jacare`. O ponto aqui é que o TCP/IP permite que `jaburu` e `jacare` enviem pacotes X11 para todos os lados, dando a impressão de que se está em um único sistema. A rede é quase transparente aqui.

Outra aplicação muito importante no TCP/IP é o NFS, o qual significa Sistema de Arquivos de Rede. É outra forma de uso transparente da rede, permitindo basicamente que se acesse diretórios e arquivos hierarquicamente que estão localizados em outras máquinas, aparentando como se fossem arquivos locais do sistema. Por exemplo, todos os diretórios podem estar em uma máquina servidora central do qual todos os outros servidores na LAN montam seus diretórios. O resultado disto é que todos os usuários podem utilizar qualquer máquina da rede, encontrando sempre o mesmo conjunto de diretórios. Da mesma maneira é possível instalar aplicações que necessitem grandes quantidades de espaço em disco (como TEX) em apenas uma máquina, e disponibilizar estes diretórios para as demais máquinas da rede. Nós retornaremos ao NFS no capítulo 11.

Naturalmente, estes são apenas exemplos do que se pode fazer em redes TCP/IP. As possibilidades são quase ilimitadas.

Agora vamos dar uma olhada a fundo na maneira como o TCP/IP trabalha. Isso se faz necessário para que se possa entender como e porque se deve configurar uma máquina. Iniciaremos pelo exame do hardware, e depois lentamente trabalharemos o caminho restante.

1.3.2 Ethernets

O tipo de hardware mais largamente usado pelas redes locais é aquele normalmente conhecido como *Ethernet*. Este consiste em um cabo com máquinas sendo conectadas a ele através de conectores, chaves ou transceptores. Ethernets simples são muito baratas, o que, junto com a sua capacidade de chegarem a taxas de transferência de até 100 Megabits por segundo, as tornam muito populares.

As Ethernets vêm em três "sabores": chamados *thick* e *thin*, respectivamente, e *par trançado*. A Ethernet thin e thick usam cabo coaxial, diferentes em largura e na maneira como se pode conectar um servidor neste cabo. A Thin Ethernet usa um conector tipo-T "BNC", no qual pode ser inserido o cabo, e que é inserido em um conector na parte traseira do computador. A Thick Ethernet requer que se faça um pequeno buraco no cabo, e se conecte um transceptor usando uma "chave vampira". Um ou mais servidores podem então ser conectados ao transceptor. Os cabos das Ethernet Thin e Thick podem ter no máximo de 200 e 500 metros, respectivamente, e são portanto chamadas de 10base-2 e 10base-5. Par trançado, praticamente um padrão em redes nos dias atuais, usa um cabo com diversos fios de cobre. É também conhecido como 10base-T, para velocidades de 10 Mbps ou 100base-T para velocidades de 100 Mbps.

A maioria das pessoas prefere a Ethernet par trançado, porque esta é muito barata e mais eficiente: placas de PC saem por menos de US$ 50, e cabos estão na faixa de alguns centavos por metro. Por exemplo, a Ethernet no Departamento de Matemática da UP é par trançado 100-BaseT, onde o tráfego não precisa ser interrompido cada vez que se deseje adicionar um novo servidor à rede, diferentemente dos outros "sabores".

Uma das desvantagens da tecnologia Ethernet é a sua limitação de comprimento de cabo, o qual o exclui de qualquer outro uso que não seja em LANs, embora muitos segmentos Ethernet possam ser ligados uns aos outros utilizando-se repetidores, pontes ou roteadores. Repetidores simplesmente copiam os sinais entre dois ou mais segmentos, então todos os segmentos juntos agirão como se fossem uma única Ethernet. Devido a problemas de temporização, não se pode ter mais do que quatro repetidores entre dois servidores na rede. Pontes e roteadores são mais sofisticados. Eles analisam dados de entrada e os enviam apenas quando o servidor receptor não estiver na Ethernet local.

A Ethernet trabalha como um sistema de vias, onde um servidor pode enviar pacotes (ou *quadros*) de até 1500 bytes a outro servidor na mesma rede Ethernet. Um servidor é endereçado por um código de seis bytes que são gravados no firmware da placa Ethernet. Estes endereços são normalmente escritos como uma seqüência de dois dígitos hexadecimais separados por dois pontos, como em `aa:bb:cc:dd:ee:ff`.

Um quadro enviado a uma estação é transmitido para todas as estações conectadas na rede, mas apenas o servidor de destino pode apanhá-lo e processá-lo. Caso duas estações tentem enviar dados simultaneamente ocorrerá uma *colisão*, a qual é solucionada através do cancelamento da transmissão por ambas as máquinas, e a tentativa de reenvio após um certo período aleatório de tempo.

1.3.3 Outros Tipos de Hardware

Em grandes instalações, como a Universidade do Pantanal, a Ethernet não é o único tipo de equipamento utilizado. Na UP, cada rede local de departamento é ligada à rede do campus, o qual possui um cabo de fibra ótica rodando FDDI (*Interface de Fibra de Dados Distribuídos*). O FDDI usa um método diferente para transmitir dados, o qual basicamente envolve o envio de um número de *"bastões"*[6] pela rede, com a estação podendo transmitir dados somente quando receber um bastão. A principal vantagem do FDDI é a velocidade acima dos 100 Mbps e um comprimento máximo do cabo de mais de 200 km.

Para conexões de rede a longas distâncias, um diferente tipo de equipamento é utilizado, o qual é baseado em um padrão chamado X.25. Muitas das chamadas Redes Públicas de Dados, como a Tymnet nos Estados Unidos, ou a Datex-P na Alemanha, oferecem este serviço. O X.25 requer um hardware especial, chamado de Montador/Desmontador de Pacotes ou *PAD*. O X.25 define um conjunto de protocolos de rede próprio, mas freqüentemente é usado para conectar redes que executem TCP/IP ou outros protocolos. Considerando que pacotes IP não podem ser mapeados em X.25 e vice-versa, eles são simplesmente encapsulados em pacotes X.25 e enviados pela rede.

Freqüentemente, rádios amadores usam seus equipamentos para colocar seus computadores em rede, isto é chamado de *rádio pacote* ou *rádio ham*. O protocolo utilizado pelos rádios ham é chamado de AX.25, derivado do X.25.

Outras técnicas envolvem o uso de linhas seriais baratas para acesso discado. Estas requerem ainda outro protocolo para transmissão de pacotes, como SLIP ou PPP, os quais são descritos nos capitulos 7 e 8 respectivamente.

1.3.4 O Protocolo Internet

Naturalmente, pode ocorrer que uma rede não deva ficar limitada a apenas uma Ethernet. O ideal seria que se estivesse apto a usar uma rede não importando qual hardware está rodando e de quantas subunidades esta seja constituída. Por exemplo, em grandes instalações como a Universidade do Pantanal, normalmente se terá um grande número de Ethernets separadas que deverão estar conectadas de alguma forma forma. Na UP, o Departamento de Matemática tem duas Ethernets: uma com máquinas lentas para alunos e outra rede com máquinas mais rápidas para os professores e alunos graduados. Ambas estão ligadas à rede FDDI do campus.

Esta conexão é tratada por um servidor dedicado, chamado também de *ponto de passagem*[7], o qual trata os pacotes que estão entrando e saindo copiando-os entre as duas Ethernets e o cabo de fibras óticas. Por exemplo, se você estiver no Depar-

[6] Como nas corridas de revezamento; do inglês tokens.

[7] gateway

tamento de Matemática, e quiser acessar `jaburu` na rede local do Departamento de Física a partir de uma máquina `Linux`, o software de rede não pode enviar pacotes diretamente à máquina jaburu, porque este não está no mesmo barramento Ethernet. Portanto, deve-se utilizar os pontos de passagem para enviá-los. O ponto de passagem (chamado de `dourado`) envia então estes pacotes para o seu ponto de passagem chamado `piranha` no Departamento de Física, usando a rede do campus, com `dourado` podendo visualizar `piranha`, a máquina de destino. O fluxo de dados entre `jacare` e `jaburu` é mostrado na figura 1.1 (Com desculpas a Guy L. Steele).

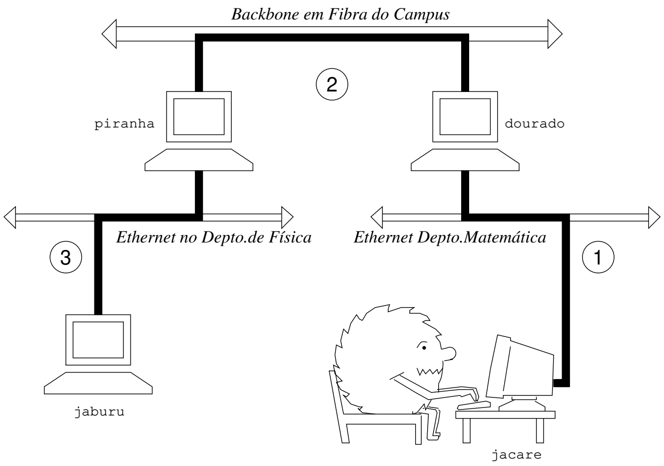

Figura 1.1: Os três passos para se enviar um datagrama de `jacare` a `jaburu`

Este esquema de direcionamento de dados a um servidor remoto é chamado de *roteamento*, onde pacotes são normalmente chamados de *datagramas* neste contexto. Para facilitar as coisas, a troca de datagramas é coordenada por um único protocolo que é independente do hardware utilizado: IP, ou *Protocolo Internet*. No capítulo 2, cobriremos o protocolo IP e os tópicos de roteamento em maiores detalhes.

O principal benefício do IP reside no fato dele transformar redes fisicamente diferentes em uma rede aparentemente homogênea. Isto é chamado de "entre redes", e o resultado da "meta-rede" é chamado de *internet*. Deve-se notar a diferença entre *uma* internet e *a* Internet. A última é o nome oficial de uma internet global.

Naturalmente, o IP também requer um esquema de endereçamento independente de hardware. Isto é feito atribuindo-se a cada servidor um único número de 32 bits, chamado de *endereço IP*. Um endereço IP é normalmente representado por quatro números decimais, um para cada porção de 8 bits, separados por pontos. Por exemplo, `jaburu` deve ter um endereço de IP de `0x954C0C04`, o qual deve ser escrito como `149.76.12.4`. Este formato é também chamado de notação *de quatro segmentos*.

Note que agora temos três tipos de endereços: primeiro há o nome do servidor, como `jaburu`, depois há um endereço IP, e finalmente há os endereços de hardware, como os endereços Ethernet de 6 bytes. Estes devem ser de alguma maneira compatíveis, para que ao se digitar `rlogin jaburu`, o software de rede possa fornecer o endereço IP de `jaburu`; e quando o IP entregar qualquer dado à Ethernet do Departamento de Física, este de alguma maneira possa descobrir qual endereço Ethernet corresponde ao endereço IP. Isto pode ser um tanto complexo.

Não entraremos neste tópico aqui, trataremos disso no capítulo 2. Por hora, basta saber que a sistemática de se encontrar um endereço a partir de um nome é chamada de *resolução de nomes*, e *resolução de endereços* é o mapeamento de endereços IP em endereços de hardware.

1.3.5 IP sobre Linhas Seriais

Em linhas seriais, um padrão de fato conhecido como PPP ou *Protocolo Ponto a Ponto* é freqüentemente usado. Outros protocolos como SLIP ou CSLIP podem ser utilizados. PPP tem muito mais facilidades que SLIP, incluindo-se a negociação de conexão. A sua maior vantagem está na possibilidade de transmitir qualquer tipo de datagrama.

1.3.6 O Protocolo de Controle de Transmissão

Obviamente enviar e receber datagramas de uma máquina para outra não é a história completa. Caso se esteja conectado a `jaburu`, é desejável ter-se uma conexão confiável entre um processo `rlogin` a ser executado em `jacare` e o interpretador de comandos em `jaburu`. A informação enviada de/e para a outra máquina deve ser dividida em pacotes pelo emissor, e remontada em um conjunto de caracteres pelo receptor. Apesar de parecer simples isso envolve um grande número de tarefas complexas.

Algo importante para se saber sobre IP é que, por definição, ele não é totalmente confiável. Assumindo, por exemplo, que 10 pessoas em uma rede Ethernet iniciem a transferência da última versão do Xfree86 a partir do servidor FTP da UP, a quantidade de tráfego gerada seria demasiada para um simples ponto de passagem poder suportar, por ser muito lento e ter pouca memória. Neste momento caso um pacote seja enviado para `jaburu`, `dourado` poderá estar sem espaço no buffer por um momento, sendo incapaz de retransmití-lo. O protocolo IP resolve este problema simplesmente descartando o pacote, o qual estará irremediavelmente perdido. É de responsabilidade da máquina checar a integridade completa dos dados e retransmití-los em caso de erro.

Isso é realizado por outro protocolo denominado TCP, ou *Protocolo de Controle de Transmissão*, o qual constrói um serviço confiável sob o protocolo IP. O uso adequado do TCP faz com que ele use o protocolo IP para dar a ilusão de uma conexão simples entre dois processos em sua máquina e em uma máquina remota, não sendo necessário assim preocupar-se com a rota que os dados eventualmente utilizem. Uma conexão TCP funciona essencialmente como um conector de duas mãos no qual ambos os processos podem escrever e ler a partir da conexão. Podemos imaginar algo similar a uma conversação através do telefone.

O protocolo TCP identifica os pontos finais da conexão pelo endereço IP das máquinas envolvidas e através do número da porta envolvida em cada máquina. Portas podem ser vistas como pontos de ligação para conexões de rede. Retornando ao exemplo da ligação telefônica, pode-se comparar o endereço IP como o código DDD e os números de portas aos números de telefones.

No exemplo de um programa `rlogin`, a aplicação cliente (`rlogin`) abre uma porta na máquina local, por exemplo `jacare`, e conecta-se à porta 513 em `jaburu`, a qual o programa servidor `rlogind` conhece e monitora. Após este procedimento, é estabelecida então uma conexão TCP. Ao utilizar esta conexão, `rlogind` executa então um procedimento de validação de usuário e após disponibiliza um ambiente interpretador de comandos. A entrada e a saída padrão do interpretador são redirecionadas para a conexão TCP, fazendo com que tudo o que seja digitado em `rlogin` na máquina local seja enviado através de datagramas TCP e seja fornecido ao interpretador de comandos na entrada padrão do servidor remoto.

1.3.7 O Protocolo de Datagrama do Usuário

TCP não é o único protocolo de usuário em uma rede TCP/IP. Apesar de aplicável em programas como `rlogin`, o custo envolvido em aplicações como NFS é proibitivo. Ao invés dele é usado um protocolo derivado do TCP chamado UDP ou *Protocolo de Datagrama do Usuário*. Assim como o TCP, UDP também permite que uma aplicação possa contactar um serviço em uma certa porta em uma máquina remota, mas não estabelece uma conexão para isto. Ao invés disso, você pode usá-lo para enviar pacotes individuais para um serviço de destino (daí o seu nome).

Vamos assumir que se tenha montado o diretório TEX na hierarquia de diretórios do servidor central NFS denominado `capivara`, e se deseje visualizar um documento que descreve como usar LATEX. Inicia-se o editor que primeiramente lerá o arquivo inteiro. Uma conexão TCP com o servidor `capivara` poderá tardar muito para enviar o arquivo e liberar a conexão novamente. Então, ao invés de utilizar o TCP, uma requisição é realizada a `capivara`, que envia o arquivo em alguns pacotes UDP, o que é muito mais rápido. De qualquer forma, o UDP não foi desenvolvido para lidar com perdas de pacotes ou corrupção de dados. Nestes casos a aplicação, no exemplo o NFS, toma conta disto.

1.3.8 Mais sobre Portas

Portas podem ser vistas como pontos de ligação para conexões de redes. Se uma aplicação deseja oferecer determinado serviço, ele conecta-se a uma determinada porta e aguarda os clientes (este processo também é chamado de *"ouvir"* uma porta). Um cliente que deseje usar os serviços aloca uma porta em sua máquina local e se conecta à porta específica na máquina remota, que ofereça o serviço desejado.

Uma propriedade importante das portas consiste em que, após o estabelecimento da conexão entre o cliente e o servidor, outra cópia do servidor possa ser criada e o servidor possa continuar a ouvir na mesma porta. Isso permite, por exemplo, que diversas conexões concorrentes de acessos remotos sejam executadas simultaneamente, todas utilizando a mesma porta 513. O protocolo TCP é capaz de estabelecer estas conexões entre máquinas, porque elas provêm de diferentes de diferentes portas ou máquinas. Por exemplo, caso se acesse duplamente a máquina `jaburu` a partir de `jacare`, o primeiro acesso via `rlogin` usará uma porta local 1023 e a segunda utilizará a porta 1022. De qualquer forma a porta utilizada em `jaburu` será sempre a de número 513.

Este exemplo mostra o uso de portas como pontos desordenados, onde um cliente contata uma porta específica para obter um determinado serviço. Para que um cliente saiba o número apropriado de uma determinada porta, um acordo tem que ser realizado entre os administradores de ambos os sistemas para a definição destes números. Para serviços largamente utilizados, como `rlogin`, estes números são administrados centralizadamente pelo IETF (ou *Força Tarefa de Engenharia Internet*), a qual regularmente publica uma RFC chamada *Números Definidos*. Ela descreve entre outras coisas, os números de portas de serviços *muito utilizados*. `Linux` usa um arquivo de mapeamento de nomes para números, chamado `/etc/services`. Ele é descrito na seção `Os Arquivos services e protocols` no capítulo 9.

É importante frisar que tanto TCP como conexões UDP baseiam-se em portas e que estes números não podem conflitar entre si. Isso significa que a porta TCP 513, por exemplo, é diferente da porta UDP 513. Na verdade, algumas portas servem de pontos de acesso para dois diferentes serviços, denominados `rlogin` (TCP) e `rwho` (UDP).

1.3.9 A Biblioteca de Conexão

Em sistemas operacionais `Unix`, o software que executa todas as tarefas e protocolos descritos acima é normalmente parte integrante do núcleo, e o mesmo ocorre com o `Linux`. A interface de programação mais comum no mundo `Unix` é conhecida como *Biblioteca Socket Berkeley*. Seu nome deriva de uma analogia popular que vê portas como tomadas, conectando-se a cada uma delas como em uma tomada. Ela provê a função denominada (`bind(2)`) para especificar uma máquina remota, um protocolo de transporte e um serviço ao qual um programa pode conectar-se ou ouvir (usando `connect(2)`, `listen(2)`, e `accept(2)`). A biblioteca socket é de alguma forma de caráter mais genérico, pois provê não somente classes de conexões baseadas em TCP/IP (`AF_INET`), mas também classes que administram conexões com a máquina local (a classe `AF_UNIX`). Algumas implementações podem ainda gerenciar outras classes, como o protocolo XNS (*Sistema de Rede Xerox*) ou X.25.

Em `Linux`, a biblioteca socket é parte da biblioteca C padrão denominada `libc`. Atualmente ela suporta somente as classes `AF_INET` e `AF_UNIX`, porém esforços têm sido despendidos para suportar outros protocolos, e eventualmente uma ou mais classes poderão ser adicionadas.

1.4 Redes Linux

Sem o esforço concentrado de programadores ao redor do mundo, o Linux não teria sido viabilizado através da rede mundial. Com esta dispersão no seu desenvolvimento, não é nenhuma surpresa o fato de, em seus primeiros estágios de desenvolvimento, diversas pessoas terem começado a trabalhar em disponibilizar capacidades de rede. Uma implementação de UUCP estava disponível no Linux praticamente no princípio de sua existência. Trabalhos baseados em redes TCP/IP foram iniciados no outono de 1992, quando Ross Biro e outros criaram o que ficou conhecido como Net-1.

Após a finalização do desenvolvimento ativo de Ross em Maio de 1993, Fred van Kempen iniciou um trabalho de reimplementação, reescrevendo as maiores partes do código. Este trabalho de continuação ficou conhecida como Net-2. Uma primeira versão pública, denominada Net-2d, foi liberada no verão de 1992 (como parte do kernel 0.99.10), e desde então tem sido mantida por diversas pessoas, mais notadamente por Alan Cox, como o Net-2d Depurado. Após testes intensos e numerosas implementações no código, o seu nome foi alterado para Net-3 depois da versão Linux 1.0 ter sido liberada.

Net-3 oferece programas de controle para uma grande variedade de placas de rede Ethernet, assim como SLIP e PPP (para envio de tráfego de rede através de linhas seriais) e PLIP (através de portas paralelas). Com o Net-3, o Linux tem uma implementação do TCP/IP que se comporta muito bem em um ambiente de rede local, apresentando uma performance capaz de superar implementações comerciais de diversos Unices. O desenvolvimento concorrente produz a estabilidade necessária para a execução confiável em servidores Internet.

Além destas facilidades, há diversos projetos em desenvolvimento que irão aprimorar a versatilidade do Linux. Dentre os já disponíveis podemos citar o PPP (um protocolo ponto a ponto que melhora a forma de enviar dados através de linhas seriais) e o AX.25, capaz de transmitir dados através de rádio amadores. Alan Cox implementou ainda o protocolo IPX da Novell ©, além do programa samba, um servidor NetBIOS de livre distribuição para Unices, escrito por Andrew Tridgell.[8] Isso significa que Linux pode atuar como cliente ou servidor de uma rede Windows©, Novell©, Unix, etc.

[8] NetBIOS é o protocolo no qual as aplicações como lanmanager e Windows para Workgroups são baseadas.

1.4.1 Diferentes Formas de Desenvolvimento

Neste meio tempo, Fred continuou o desenvolvimento no Net-2e, cujas funcionalidades foram profundamente revisadas na camada de rede. Uma das mais notáveis implementações é a incorporação do DDI, a *Interface de Controle de Dispositivos*[9], a qual oferece um acesso uniforme e métodos de configuração de todos os dispositivos de rede e protocolos.

Uma outra implementação de redes TCP/IP foi desenvolvida por Matthias Urlichs, o qual escreveu um programa de controle de ISDN para `Linux` e FreeBSD. Para este, ele integrou algum código de rede BSD ao núcleo do `Linux`.

Como uma previsão futura, diria que Net-3 parece ter chegado para ficar. Alan trabalha atualmente na implementação do protocolo AX.25 usado pelos rádio amadores. Sem dúvida este novo módulo certamente produzirá um novo impulso no uso de códigos de rede. Módulos permitem a adição de programas de controle de dispositivos ao kernel em tempo de execução.

Apesar destas diferentes implementações de protocolos de rede, todas provêem o mesmo tipo de serviço, diferenciando-se basicamente ao nível do kernel e programas de controle de dispositivos. De qualquer forma, não será possível utilizar um kernel com Net-2e e utilitários Net-2d ou Net-3, e vice e versa. Isso somente se aplica aos programas que lidam com o kernel mais intimamente; aplicações e comandos de rede como `rlogin` ou `telnet` podem ser executados em quaisquer versões TCP/IP instaladas. O kernel oficial liberado será sempre acompanhado de um conjunto de ferramentas de rede compatíveis com o seu código.

1.4.2 Onde conseguir os códigos fontes

A última versão dos fontes de rede `Linux` podem ser obtidos através de FTP anônimo a partir de diversos sites. O site oficial FTP para o Net-3 é espelhado em `metalab.unc.edu` no caminho `system/Network/sunacm`. A mais recente atualização do Net-2 e seus binários estão disponíveis em `ftp.aris.com`. O código de rede de Matthias Urlichs derivado de BSD pode ser obtido a partir de `ftp.ira.uka.de` no caminho `/pub/system/linux/netbsd`.

Os kernels mais recentes podem ser encontrados em `nic.funet.fi` no caminho `/pub/OS/Linux/PEOPLE/Linus`; `metalab` e `tsx-11.mit.edu` espelha este diretório.

[9] Device Driver Interface

1.5 Mantendo seu Sistema

Ao longo deste livro, lidaremos basicamente com temas relacionados com a instalação e configurações de itens de rede. Administração é, de qualquer forma, muito mais que isso, pois após instalar e configurar um serviço, será necessário mantê-lo. Para a maior parte destes serviços será necessária alguma pequena atenção, enquanto outros como correio eletrônico e notícias requerem a execução de rotinas diárias para mantê-los atualizados. Iremos discutir mais acuradamente estas tarefas posteriormente.

O mínimo absoluto na manutenção de qualquer serviço consiste em checar os arquivos de mensagens de cada aplicação, buscando condições de erro ou eventos não usuais. Comumente, pode-se fazer isto através de pequenos programas administrativos que são executados periodicamente pelo utilitário `cron`. A fonte de distribuição das principais aplicações, como `smail` ou C News, já contém tais aplicativos. Você somente terá que adequar estes programas às suas necessidades e preferências.

A saída de qualquer tarefa ativada pelo `cron` pode ser enviada por email para a conta do administrador. Por padrão, muitas aplicações enviarão mensagens de erro, estatísticas de uso ou resumos de arquivos de mensagens para a conta do superusuário[10]. Isso somente faz sentido caso a conta do superusuário seja usada com freqüência. Uma idéia melhor pode ser o redirecionamento das mensagens do superusuário para uma conta pessoal, criando-se um nome alternativo de email, conforme descrito no capítulo 14.

Ainda que o site tenha sido configurado cuidadosamente, a lei de Murphy garante que alguns problemas *irão* acontecer. De qualquer forma, manter um sistema significa ainda estar disponível para receber sugestões e reclamações. Normalmente as pessoas esperam que o administrador de sistemas possa ser no mínimo alcançado via correio eletrônico, através de uma mensagem enviada para o *superusuário*. Porém há outros endereços que são comumente utilizados para estas tarefas. Por exemplo, reclamações sobre o mal funcionamento de correio eletrônico são normalmente enviadas para `postmaster`, e problemas com o sistema de notícias devem ser reportados ao `newsmaster` ou `usenet`. Mensagens para o `hostmaster` devem ser redirecionadas para a pessoa encarregada dos serviços básicos de rede do servidor e do servidor de nomes DNS, caso se esteja executando um.

[10] root

1.5.1 Sistema de Segurança

Outro aspecto importante da administração do sistema é proteger o ambiente de rede contra usuários mal intencionados e intrusos. O gerenciamento descuidado do sistema pode oferecer muitos alvos para usuários sem escrúpulos: ataques podem variar da tentativa de descoberta de uma senha até a monitoração do tráfego da rede, e os danos causados podem produzir desde mensagens com remetentes falsos até perda de dados ou violação da privacidade dos usuários. Mencionaremos aqui alguns dos problemas mais comuns, a forma como eles ocorrem e forma de evitá-los.

Esta seção irá discutir alguns exemplos e técnicas básicas em lidar com o sistema de segurança. Obviamente, os tópicos aqui descritos não descrevem as situações de forma extremamente detalhada, porém servem como forma ilustrativa das situações com as quais o administrador poderá se defrontar. De qualquer forma, a leitura de um bom livro de segurança é absolutamente necessária, especialmente em um sistema de redes. O livro "Practical UNIX Security" de Simon Garfinkel's (veja [Spaf93]) é altamente recomendada.

O sistema de segurança começa com uma boa administração do sistema. Isso inclui a checagem do dono e das permissões de todos os arquivos e diretórios, a monitoração do uso de contas privilegiadas, etc. O programa COPS, por exemplo, irá checar o sistema de arquivos e os arquivos de configurações mais utilizados, procurando por permissões não usuais e outras anomalias. É aconselhável ainda utilizar um programa de aperfeiçoamento de senhas, tornando-as mais difíceis de serem descobertas. O utilitário de senhas sombra[11], por exemplo, requer que uma senha tenha no mínimo cinco letras e contenha tanto maiúsculas como minúsculas, além de números.

Ao criar um serviço acessível pela rede, deve-se estar seguro de dar-lhe o menor privilégio possível, significando que não será permitido executar atividades não enquadradas nos objetivos do programa. Por exemplo, pode-se desenvolver programas que utilizem `setuid` para o superusuário `root` ou alguma outra conta privilegiada, os quais dêem privilégios ao programa somente quando for necessário. Por exemplo, caso se deseje permitir que estações sem disco rígido sejam inicializadas a partir de sua máquina central, deve-se prover o serviço de TFTP (Serviço de Transferência Simples de Arquivos), permitindo que este possa receber os arquivos de configuração a partir do diretório `/boot`. De qualquer forma, ao ser usado de maneira irrestrita, o TFTP permitirá que qualquer pessoa no mundo possa receber qualquer arquivo a partir de seu sistema. Caso não seja isso que se queira, porque

[11] veja o Guia de Instalação do Conectiva Linux para informações sobre senhas sombra

não restringir o serviço TFTP ao diretório `/boot`?[12]

Seguindo a mesma linha de pensamento, pode-se querer restringir certos serviços para usuários de certos servidores, digamos que dentro da rede local. No capítulo 9, apresentamos o servidor `tcpd`, o qual executa esta tarefa para uma grande variedade de aplicações de rede.

Outro ponto importante é evitar softwares suspeitos ou perigosos. Claro que qualquer software pode ser potencialmente perigoso, uma vez que pode ter problemas que gente esperta pode utilizar para explorar um sistema e até mesmo ganhar acesso à máquina. Coisas como essa acontecem e não há proteção completa que garanta a infalibilidade do sistema. Estes problemas afetam softwares de livre distribuição, assim como também produtos comerciais.[13]

De qualquer forma, programas que requerem privilégios especiais são potencialmente mais perigosos que outros, porque qualquer falha poderá trazer conseqüências catastróficas. Caso se instale um programa que utilize o setuid para propósitos de configuração de redes, deve-se redobrar os cuidados para não se esquecer de nada que esteja na documentação, para não se criar acidentalmente um problema de segurança.

Não se deve nunca esquecer de que, por maiores que sejam as precauções, elas podem falhar, independente de quão cuidadoso se seja. Deve-se também ser capaz de detectar intrusos o mais cedo possível. Verificar os arquivos de mensagens é um bom ponto de partida, porém o intruso é provavelmente esperto o suficiente para apagar as pistas de sua presença nestes arquivos. De qualquer forma há ferramentas como `tripwire`[14], a qual permite uma checagem em arquivos vitais ao sistema, caso estes tenham tido o seu conteúdo ou permissões alterados. `tripwire` executa diversas checagens da integridade destes arquivos e armazena as informações em uma base de dados. Durante as execuções subseqüentes, os números de verificação serão recalculados e comparados com aqueles armazenados, fazendo com que eventuais modificações sejam detectadas.

---

[12] Nós veremos este tema mais profundamente no capítulo 9.

[13] Há alguns `Unices` comerciais pelos quais se pagam valores consideráveis que permitem que usuários recebam privilégios de superusuário com alguns truques simples.

[14] Escrito por Gene Kim e Gene Spafford.

1.6 Perspectiva dos Capítulos Seguintes

Os próximos capítulos lidam com a configuração do `Linux` para o uso do TCP/IP e execução de algumas das aplicações principais. Antes de "sujar as mãos" com a edição de arquivos, iremos examinar o protocolo IP um pouco mais detidamente no capítulo 2. Caso você já tenha um conhecimento razoável de como o roteamento IP funciona e como a resolução de endereços é executada, você pode passar diretamente para o outro capítulo.

O capítulo 3 lida com temas de baixo nível como construção do kernel e configuração de uma placa de rede Ethernet. A configuração de uma porta serial é coberta em um capítulo em separado (capítulo 4), uma vez que as discussões não se aplicam somente a redes TCP/IP, mas são também relevantes para o UUCP.

O capítulo 5 auxilia na configuração de uma máquina para o uso de redes TCP/IP. Ele contém dicas de instalação desde máquinas isoladas com somente um dispositivo local até equipamentos conectados a uma rede Ethernet. Irão ainda ser apresentadas algumas ferramentas úteis para testar e depurar a sua configuração. O próximo capítulo discute como configurar a resolução de nomes de máquinas e explica como configurar um servidor de nomes.

Este é seguido por dois capítulos que abordam a configuração e uso do SLIP e PPP respectivamente. O capítulo 7 explica como estabelecer conexões SLIP e fornece referências detalhadas do programa `dip`, uma ferramenta que permite a automação de muitos dos passos necessários. O capítulo 8 cobre o protocolo PPP e o servidor `pppd`, necessário para o seu funcionamento.

O capítulo 9 fornece uma breve descrição de algumas das mais importantes aplicações de rede, tais como `rlogin`, `rcp`, etc. Ele cobre ainda alguns serviços gerenciados pelo programa `inetd` e como restringir certos serviços relevantes à segurança na configuração de um conjunto de máquinas confiáveis.

Os próximos dois capítulos discutem o NIS, o Sistema de Informações em Rede e o NFS, o Sistema de Arquivos em Rede. NIS é uma ferramenta útil para distribuir informações administrativas em uma rede local, como por exemplo senhas de usuários. Já o NFS permite o compartilhamento de sistemas de arquivos entre diversas máquinas em uma rede local.

O capítulo 12 fornece uma extensa introdução à administração do UUCP Taylor, uma implementação de livre distribuição das ferramentas UUCP.

A revisão deste livro é levada a cabo no passeio detalhado pelos serviços de correio eletrônico e servidor de notícias Usenet. O Capítulo 13 introduz os conceitos de correio eletrônico, assim como o funcionamento do endereçamento de mensagens e a forma como o correio administra o sistema de obtenção de mensagens.

Os capítulos 14 e 15 cobrem, cada um, a configuração dos programas `smail` e `sendmail`, dois agentes transportadores de mensagens que podem ser utilizados no `Linux`. Este livro explica ambos, uma vez que `smail` é simples de ser instalado por iniciantes, enquanto `sendmail` é bem mais flexível, poderoso e complexo.

Os capítulos 16 e 17 explicam como os sistemas de notícias são gerenciados na Usenet e como instalar e usar o C news, um pacote popular destinado ao gerenciamento de notícias da Usenet. O capítulo 18 cobre as instruções sobre a configuração de um servidor NNTP para prover acesso às notícias em uma rede local. Finalmente o capítulo 19 mostra como configurar e manter diversos leitores de notícias.

Capítulo 2

Redes TCP/IP

Voltaremos agora aos detalhes com os quais se toma contato ao se conectar uma máquina `Linux` em uma rede TCP/IP, incluindo detalhes de endereços IP, nomes de servidor e algumas funções de roteamento. Este capítulo proporcionará os subsídios necessários para que se compreenda os ajustes necessários, enquanto os próximos cobrirão as ferramentas para a sua implementação.

2.1 Interfaces de Rede

Para esconder a diversidade de equipamentos que podem ser usados em ambientes de rede, o TCP/IP define uma *interface* abstrata através da qual o hardware é acessado. Esta interface oferece um conjunto de operações idênticas para todos os tipos de hardware e basicamente trabalha enviando e recebendo pacotes.

Para cada dispositivo periférico que se deseje usar na rede, uma interface correspondente deve estar presente no núcleo do sistema. Por exemplo, a interface Ethernet no `Linux` é chamada `eth0`, `eth1`, etc. e as interfaces SLIP são denominadas `sl0`, `sl1`, etc. Estes nomes de interfaces são usados na configuração, durante a definição de um dispositivo físico particular no kernel. Eles não têm nenhum outro significado além disto. Para que possa ser utilizada em redes TCP/IP, deve ser designado um endereço IP à interface, o qual serve como sua identificação ao comunicar-se com o resto do mundo. Este endereço é diferente do nome da interface mencionado acima. Por exemplo ao se comparar uma interface à uma porta de uma casa, o endereço é como a placa de número pendurada nesta.

Naturalmente há outros parâmetros do dispositivo que devem ser ajustados. Um destes é o tamanho máximo de datagramas que podem ser processados por um hardware específico, também chamado de *Unidade Máxima de Transferência*[1], ou MTU. Outros atributos serão apresentados mais tarde.

2.2 Endereços IP

Como mencionado no capítulo anterior, os endereços compreendidos pelo protocolo de rede IP são números formados por 32 bits. Para toda máquina deve ser designado um número único no ambiente de rede. Caso se esteja em uma rede local que não possui tráfego TCP/IP com outras redes, é possível designar estes números de acordo com as preferências pessoais do administrador. Porém para sites com conexões com a Internet, estes são designados por uma autoridade central, o Centro de Informações de Rede ou NIC.[2]

Para uma leitura mais simples, os endereços IP são divididos em números de 8 bits chamados *octetos*. Por exemplo, `xavante.conectiva.com.br` possui um endereço IP `0x954C0C04`, o qual pode ser representado como `149.76.12.4`. Este formato é freqüentemente chamado de *notação das quatro partes*. Outra razão para esta notação é que os endereços IP são divididos em duas partes: o número de *rede*, contido em um ou mais octetos e o número de *máquina*, o qual é a identificação da máquina na rede. Ao receber endereços IP, estes não serão fornecidos pelo órgão responsável na proporção de um para cada servidor que se planeje usar. Ao contrário, normalmente é fornecido somente um número de rede, e é permitido que todos os endereços IP válidos dentro desta faixa sejam utilizados para máquinas da rede de acordo com as preferências e necessidades do administrador. Dependendo do tamanho da rede, a parte do endereço que indica os servidores pode variar de tamanho. Para atender a diferentes necessidades, existem as chamadas *classes de rede*, definindo diferentes divisões em endereços IP entre a parte do endereço que indica a rede e a parte que indica a estação. As classes existentes são as seguintes:

Classe A Classe A compreende as redes de endereços `1.0.0.0` até `127.0.0.0`. O número de rede está contido no primeiro octeto. Isso possibilita que a parte

[1] Maximum Transfer Unit

[2] Freqüentemente, os endereços IP são designados pelo provedor do qual se adquire a conectividade. Nos EUA é possível dirigir-se diretamente ao NIC solicitando um endereço de IP enviando uma mensagem a `hostmaster@internic.net`. No Brasil a numeração é definida pelo provedor de recursos Internet: a EMBRATEL ou outros, como as companhias telefônicas locais.

do endereço reservada às máquinas tenha um tamanho de 24 bits, permitindo assim aproximadamente 16 milhões de máquinas em uma mesma rede.

Classe B Classe B compreende as redes de endereços `128.0.0.0` até `191.255.0.0`. Neste caso o número de rede está contido nos dois primeiros octetos. Isto permite 16.320 redes com até 65.024 máquinas cada.

Classe C Classe C compreende as redes de endereços `192.0.0.0` até `223.255.255.0`, com o número de rede contido nos primeiros três octetos. Isto permite aproximadamente 2 milhões de redes com até 254 máquinas cada.

Classes D, E e F Endereços que estão na faixa de `224.0.0.0` até `254.0.0.0` ou são ainda experimentais ou são reservadas para uso futuro e não especificam qualquer rede válida.

Retornando ao exemplo do capítulo anterior, percebemos que `149.76.12.4`, o endereço de `jacare`, refere-se à máquina `12.4` na rede de classe B `149.76.0.0`.

Ao analisar mais cuidadosamente os endereços acima, pode-se perceber que nem todos os valores possíveis foram permitidos para cada octeto na parte do endereço que indica a máquina. Isso se deve à uma convenção onde os octetos de máquina com valores iguais a `0` ou `255` são reservados para propósitos especiais. Um endereço de máquina igual a `zeros` referencia a rede, e um endereço onde todos os bits são iguais a 1 é denominado endereço de propagação (significa *todas as máquina da rede* simultaneamente). Por exemplo, o endereço `149.76.255.255` não pode ser atribuído à uma máquina da rede, porém faz referência a todas as máquinas da rede `149.76.0.0`.

Há ainda dois outros endereços de rede reservados: `0.0.0.0` e `127.0.0.0`. O primeiro é chamado de *rota padrão*, o último de *endereço local*. A rota padrão está relacionada com a forma como os datagramas IP são roteados, a qual será explicada adiante.

O endereço de rede `127.0.0.0` é reservado para o tráfego local da máquina. Normalmente o endereço `127.0.0.1` será definido para uma interface especial da máquina denominada *interface local*[3], a qual atua como um circuito fechado. Qualquer pacote IP enviado para esta interface a partir dos protocolos TCP ou UDP será retornado à própria máquina que o enviou como se estivesse chegando da rede.

[3]loopback interface.

Isso permite a aplicação de testes de redes, sem necessariamente se estar conectado a uma rede "real". Outra aplicação útil é a utilização de softwares de rede em máquinas isoladas[4]. Isso pode não ser tão raro quanto possa parecer à primeira vista. Por exemplo, muitos sites UUCP não têm na realidade conectividade IP, mas necessitam executar o sistema de notícias INN. Para uma operação adequada no `Linux`, INN necessitará de uma interface local adequada.

2.3 Resolução de Endereços

Agora que já vimos como endereços IP são formados, pode-se estar curioso em saber como eles são usados em uma rede Ethernet para referenciar diferentes equipamentos. Na verdade, o protocolo *Ethernet* identifica uma máquina através de um número de seis octetos que não tem nada em comum com o endereço IP.

Um mecanismo de mapeamento de endereços é então necessário, para que possamos relacionar endereços Ethernet com endereços IP. Este sistema é denominado *Protocolo de Resolução de Endereços*, ou ARP[5]. Na verdade ARP não está restrito a redes Ethernet, mas pode ser usado, por exemplo, em redes de rádio amadores. A idéia básica do ARP consiste no modelo usado por muitas pessoas que precisam encontrar Sr. Pedro Cabral em um conjunto de 150 pessoas: ele passeia entre elas, chamando pelo nome, seguramente se terá uma resposta caso ele esteja presente.

Quando ARP necessita descobrir o endereço Ethernet correspondente a um endereço IP fornecido, ele usa uma funcionalidade Ethernet conhecida como *broadcasting*, onde um datagrama é endereçado a todas as estações da rede simultaneamente. Ele contém um questionamento sobre o endereço IP. Cada máquina que receba o datagrama, compara este com o seu próprio endereço IP, e caso eles coincidam, a máquina retornará uma resposta ARP à estação de origem da pesquisa. Esta por sua vez pode agora extrair o endereço Ethernet da resposta.

Obviamente pode-se perguntar em como obter o endereço Ethernet de uma única máquina, entre "zilhões" de máquinas através de todo o mundo, que pode ainda sequer usar redes de tipo Ethernet. Estas questões envolvem um processo denominado *roteamento*, que tem a função de obter a localização física de uma máquina em uma rede. Este será o tema do próximo tópico.

Por hora, vamos tratar do ARP um pouco mais detidamente. Uma vez que o

[4]Não conectadas a nenhuma rede.
[5]Address Resolution Protocol

endereço Ethernet da máquina tenha sido descoberto, ele é armazenado no cache ARP, permitindo que o próximo acesso ao equipamento não tenha que sofrer o mesmo tipo de pesquisa de envio de datagramas a todas as máquinas da rede. Porém não seria muito inteligente manter esta informação indefinidamente. Por exemplo, a placa de rede Ethernet pode ser substituída por problemas técnicos, tornando o endereço ARP inválido. Para forçar uma nova pesquisa de endereços IP, as entradas no cache ARP são descartadas após algum tempo.

Algumas vezes é necessário ainda encontrar o endereço IP associado a um endereço Ethernet fornecido. Isso ocorre quando um equipamento sem disco rígido necessita inicializar o sistema operacional a partir de um servidor de rede, o que é uma situação comum em uma rede local. Um cliente sem disco rígido, praticamente não tem nenhuma informação sobre si mesmo, exceto talvez o seu endereço Ethernet. Então, basicamente o que ele faz, é divulgar uma mensagem contendo um pedido aos servidores de inicialização para que informem qual é o seu endereço IP. Há ainda o protocolo RARP - *(Protocolo de Resolução de Endereços Reversos).* Em conjunto com o protocolo BOOTP serve para definir um procedimento de configuração de inicialização de clientes sem discos em uma rede[6].

2.4 Roteamento IP

2.4.1 Redes IP

◇ Ao se escrever uma carta para alguém, deve ser colocado o endereço completo do destinatário no envelope, especificando-se o País, estado, CEP, etc.. Após isso ela é colocada em uma caixa de correio e os Correios a enviarão para o seu destino: a carta vai até o País indicado, onde o serviço de correio local a enviará para o estado indicado, para a cidade de destino, etc. A vantagem deste sistema hierárquico é óbvia: toda vez que uma carta for postada, o correio local saberá o endereço do destinatário, mas não tem que preocupar-se em como a carta irá viajar até chegar ao seu destino final.

Redes IP estão estruturadas de uma forma similar. Toda a Internet consiste em um número de redes próprias, denominadas *sistemas autônomos.* Cada sistema destes executa qualquer roteamento interno entre seus membros, porém a tarefa de entregar um datagrama resume-se em encontrar-se um caminho para a rede

[6] Veja o *Como Fazer - Estações Sem Disco Rígido* no Guia do Servidor Linux, para maiores detalhes sobre este tópico.

da máquina de destino. Isso significa que assim que o datagrama é enviado para *qualquer* máquina que esteja em uma rede em particular, processos adicionais são executados exclusivamente pela rede de destino (como no caso dos correios locais).

2.4.2 Sub-redes

Esta estrutura é produzida através da divisão de um endereço IP em uma parte destinada à identificação da rede e outra parte destinada à máquina. Por padrão a rede de destino é derivada da parte do endereço IP definida para redes. Obviamente máquinas com endereços IP de rede idênticos devem estar localizadas na mesma rede.[7]

Faz sentido disponibilizar um sistema similar do lado *interno* de uma rede, uma vez que ela pode consistir de uma coleção de centenas de pequenas redes, sendo as menores unidades as redes físicas, como por exemplo Ethernet. Ou seja, o protocolo IP permite a divisão de uma rede IP em diversas *sub-redes*.

Uma sub-rede assume a responsabilidade pela entrega de datagramas em uma determinada faixa de endereços IP de uma rede IP da qual ela faça parte. Assim como nas classes de rede A, B ou C, ela é identificada pela parte de rede do endereço IP. A parte de rede é porém expandida, incluindo alguns bits da parte de endereço de máquina. O número de bits que são interpretados como o número da sub-rede é definido pelo parâmetro definido como *máscara de sub-rede*, ou *netmask*. Esta é composta por um número de 32 bits, o qual especifica a parte de rede do endereço IP.

A rede do campus da Universidade do Pantanal é um exemplo de tal rede. Ela usa um endereço Classe B igual a `149.76.0.0` e sua máscara de rede é igual a `255.255.0.0`, e está conectada à Internet através de uma única máquina no centro de computação, fazendo com que todos os datagramas externos à rede passem por esta máquina.

Internamente, o campus da UP consiste de diversas redes menores, como redes locais dos diversos departamentos. A faixa de endereços IP está dividida em 254 sub-redes, de `149.76.1.0` até `149.76.254.0`. Por exemplo, o Departamento de Física recebeu o endereço `149.76.12.0`. A rede do campus tem o endereço `149.76.1.0`. Estas sub-redes compartilham o mesmo endereço de rede, sendo usado o terceiro octeto para se poder distinguir as sub-redes umas das outras. Para

[7] Sistemas autônomos são ligeiramente diferentes, pois podem conter mais de um endereço de rede.

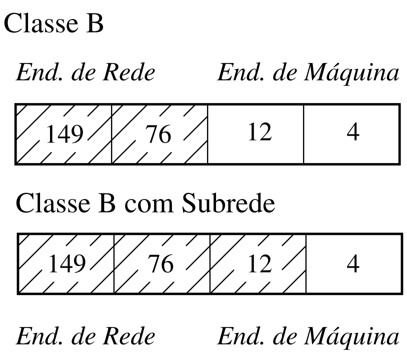

Figura 2.1: Criando sub-redes em uma rede classe B

tanto elas utilizam uma máscara de sub-rede igual a `255.255.255.0`. A figura 2.1 mostra como `149.76.12.4`, o endereço de `jacare`, é interpretado diferentemente quando o endereço é obtido de uma rede de Classe B normal e quando é utilizado o sistema de sub-redes.

É importante frisar que a definição de sub-redes é somente uma *divisão interna* da rede. sub-redes são geradas pelos administradores locais das redes. Freqüentemente, sub-redes são criadas para refletir limites existentes, sejam físicos (entre duas redes Ethernets), administrativos (entre dois departamentos) ou geográficos, sendo que a autoridade sobre essas sub-redes é delegada a alguma pessoa de contato. De qualquer forma, esta estrutura afeta somente o comportamento interno da rede e é completamente invisível para o mundo externo.

2.4.3 Ponto de Passagem

O sistema de sub-redes não tem somente benefícios organizacionais. É freqüentemente uma conseqüência de limites de equipamentos. A visão de uma máquina em uma determinada rede física, como em uma rede Ethernet, é muito limitada: os únicos equipamentos com os quais ele pode se comunicar diretamente são os que estão presentes na mesma rede. Todos os outros equipamentos fora da rede podem ser acessados através de máquinas conhecidas como *pontos de passagem*[8]. Um ponto de passagem é um equipamento que está conectado fisicamente a uma ou mais redes simultaneamente e está configurado para trocar pacotes entre elas.

[8] gateway

Para que o protocolo IP seja capaz de reconhecer facilmente se uma máquina está em uma rede local, diferentes redes físicas têm que possuir diferentes endereços IP. Por exemplo o número de rede `149.76.4.0` está reservado para a rede local do Departamento de Matemática. Ao enviar um datagrama para a máquina `jacare`, o software de rede em `jaburu` imediatamente conclui que o endereço IP, `149.76.12.4`, da máquina de destino está em uma rede física diferente, e que somente pode ser alcançado através de um ponto de passagem (`dourado` por padrão).

`dourado` está conectado a duas diferentes sub-redes: o Departamento de Matemática e a rede do campus. Ele acessa cada uma com diferentes interfaces, `eth0` e `fddi0`, respectivamente. Agora, qual o endereço de rede que deve ser definido para ele? Devemos definir o endereço de uma sub-rede `149.76.1.0` ou da sub-rede `149.76.4.0`?

A resposta é: *ambos*. Ao se comunicar com a rede local do Departamento de Matemática, a máquina `dourado` deve usar o endereço IP `149.76.4.1`, e ao comunicar-se com a rede do campus deve usar o endereço `149.76.1.4`[9].

Um ponto de passagem recebe um endereço IP para cada rede à qual esteja conectado. Estes endereços, junto com as máscaras de rede correspondentes, são definidas para a interface da sub-rede à qual ele esteja conectado. Por exemplo, o mapa de interfaces e endereços da máquina `dourado` terá o seguinte conteúdo:

Interface	**Endereço**	**Máscara**
`eth0`	`149.76.4.1`	`255.255.255.0`
`fddi0`	`149.76.1.4`	`255.255.255.0`
`lo`	`127.0.0.1`	`255.0.0.0`

A última entrada descreve uma interface local `lo`, a qual é descrita acima.

A Figura 2.2 mostra parte da topologia da rede da UP - Universidade do Pantanal. Máquinas que estão em duas sub-redes ao mesmo tempo mostram ambos os endereços.

Geralmente, pode-se ignorar a sutil diferença entre a definição de um endereço para uma máquina e sua interface. Para máquinas que estão em somente uma rede, como por exemplo `jacare`, pode-se referenciar a máquina como tendo um endereço IP, porém para tratar adequadamente o tema deveríamos dizer que a

[9] Veja no Guia do Servidor Linux sobre como configurar mais de uma placa de rede em um equipamento Linux.

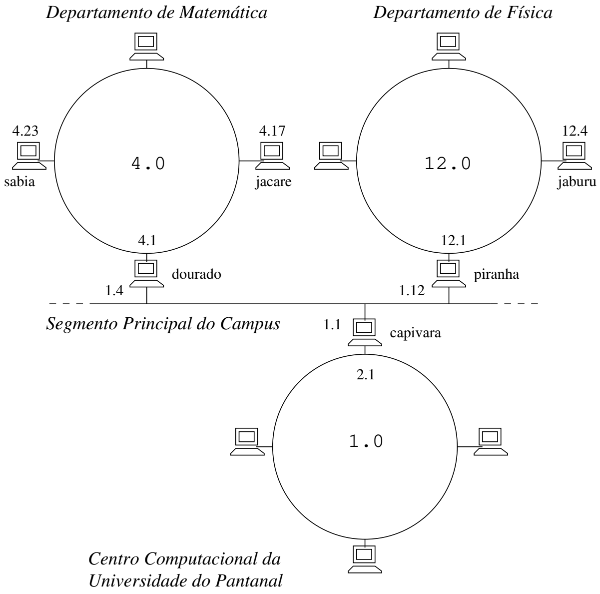

Figura 2.2: Parte da topologia de rede da Universidade do Pantanal

interface Ethernet tem um endereço IP. De qualquer forma a distinção somente é importante ao se referenciar um ponto de passagem.

Cabe acrescentar que uma sub-rede pode também ser ainda subdividida. Por exemplo, o Departamento de Matemática poderia ter duas redes Ethernets que estão conectadas por um único ponto de passagem que provê ainda conexão à rede FDDI do campus. Para executar o roteamento entre elas a rede `149.76.4.0` é subdividida em duas sub-redes de 126 endereços cada. A máscara de rede passa a ser `255.255.255.128`, e as máquinas em cada rede Ethernet passam a ter endereços nas faixas `149.76.4.1` até `149.76.4.127`, e na segunda sub-rede de `149.76.4.129` até `149.76.4.254`, respectivamente.

2.4.4 A Tabela de Roteamento

Vamos tratar agora sobre como o protocolo IP escolhe o ponto de passagem a ser usado ao enviar um datagrama para uma rede remota.

Já pudemos ver que a máquina `jacare`, ao receber um datagrama destinado a `jaburu`, verifica o endereço de destino e descobre que ele não está na rede local.

Ele então envia o datagrama para o ponto de passagem padrão, `dourado`, o qual enfrenta basicamente a mesma tarefa. `dourado` conclui que `jaburu` não está em nenhuma das redes às quais ele está conectado. Ele deve então encontrar um outro ponto de passagem para enviar o datagrama. A escolha correta deveria ser `piranha`, o ponto de passagem do Departamento de Física. `dourado` necessita ainda de informações que possam associar uma rede destino com o ponto de passagem adequado.

As informações utilizadas pelo protocolo IP para roteamento consistem basicamente em uma tabela relacionando redes e pontos de passagem utilizados para alcançá-las. Uma entrada genérica, aplicada a todos os endereços não localizados na tabela local também deve ser normalmente informada. Este é um ponto de passagem associado à rede `0.0.0.0`. Todos os pacotes destinados a uma rede desconhecida são enviados através desta rota padrão. No caso da máquina `dourado`, esta tabela deve assemelhar-se a algo como:

Rede	Ponto de Passagem	Interface
`149.76.1.0`	-	`fddi0`
`149.76.2.0`	`149.76.1.2`	`fddi0`
`149.76.3.0`	`149.76.1.3`	`fddi0`
`149.76.4.0`	-	`eth0`
`149.76.5.0`	`149.76.1.5`	`fddi0`
...	...	...
`0.0.0.0`	`149.76.1.2`	`fddi0`

Roteamento para a rede à qual `dourado` está diretamente conectado não requer um ponto de passagem. De qualquer forma foi definida como "-", significando a máquina local.

Tabelas de roteamento podem ser construídas de várias maneiras. Para pequenas redes locais é normalmente mais eficiente construí-las manualmente e mantê-las usando o comando `route` durante a inicialização do sistema (veja o capítulo 5). Para redes maiores, elas são construídas e ajustadas em tempo de execução pelos programas *servidores de roteamento*, normalmente executados em servidores da rede e que trocam informações de roteamento para definir os melhores "caminhos" ou rotas entre os membros da rede.

Dependendo do tamanho da rede, diferentes protocolos de roteamento podem ser usados. Para roteamento dentro de sistemas autônomos (como a Universidade do Pantanal), *protocolos de roteamento interno* são utilizados. O mais conhecido é o RIP, o Protocolo de Informações de Roteamento, o qual é implementado pelo servidor BSD `routed`. Para roteamento entre sistemas autônomos, *protocolos de roteamento externos* como EGP[10] (Protocolo de Ponto de Passagem Externo) ou BGP[11](Protocolo de Ponto de Passagem de Fronteira) devem ser usados. Estes (assim como o RIP) foram implementados no programa servidor `gated` da Universidade de Cornell.[12]

Normalmente, nenhum roteamento dinâmico será necessário a menos que a rede seja muito grande ou contenha um grande número de conexões. Por esta razão, somente tabelas de roteamento estáticas criadas durante a inicialização do sistema serão criadas.

2.4.5 Valores de Métrica

Roteamento dinâmico baseado em RIP escolhe a melhor rota de algumas máquinas ou redes de destino baseado no número de "hops", ou seja no número de pontos de passagem que devem ser utilizados até que o destino seja atingido. Quanto menor o caminho, melhor o RIP irá classificá-lo. Rotas muito longas, com 16 ou mais hops são definidas como inúteis e descartadas.

Para utilizar o RIP para gerenciar as informações de roteamento internas da rede de uma rede local, deve-se executar o programa `gated` em todas as máquinas. Durante a inicialização do sistema o programa `gated` verificará todas as interfaces de rede ativas. Caso haja mais de uma interface ativa (desconsiderando a interface local), ele assume que a máquina está trocando pacotes com outras redes, e irá ativamente trocar e divulgar informações de roteamento. De outra forma ele passivamente irá receber quaisquer atualizações da tabela de roteamento RIP.

Ao divulgar as informações de uma tabela de roteamento local, `gated` calcula o tamanho de uma rota através da *métrica de roteamento* associada com a entrada na tabela de roteamento. Este valor é definido pelo administrador do sistema ao configurar a rota e pode refletir o "custo" de utilizar-se este caminho. Assim a métrica de uma rota de uma sub-rede à qual a máquina esteja conectada será sempre igual a zero e uma rota que utilize dois pontos de passagem deve ter um valor igual a 2. Não se deve preocupar-se com estes valores caso não se esteja utilizando `RIP` ou `gated`.

[10]External Gateway Protocol

[11]Border Gateway Protocol

[12]`routed` é considerado um pouco problemático por muitos usuários. Uma vez que o programa `gated` suporta RIP também, é melhor utilizá-lo ao invés do `routed`.

2.5 O Protocolo de Controle de Mensagens Internet

O protocolo IP tem um protocolo companheiro, o qual ainda não foi comentado. Ele é denominado ICMP - *Protocolo de Controle de Mensagens Internet*[13] e é usado pelo código do núcleo do sistema de rede para enviar mensagens de erro para outras máquinas. Por exemplo, assumindo que você esteja utilizando a máquina `jacare` novamente e deseja executar o programa `telnet` na porta 12345 da máquina `jaburu`, porém não há nenhum processo recebendo mensagens naquela porta. Quando o primeiro pacote TCP para esta porta chega em `jaburu`, a camada de rede irá reconhecer o que ocorre e retornará uma mensagem ICMP para `jacare` com a mensagem "Porta Indisponível".

Há um número expressivo de mensagens que o ICMP compreende, muitas das quais lidam com condições de erro. De qualquer forma há uma em especial, muito interessante chamada de mensagem de redirecionamento. Ela é gerada pelo módulo de roteamento, ao detectar que outra máquina está usando este como um ponto de passagem, apesar de haver um caminho muito mais curto. Por exemplo, após a inicialização a tabela de roteamento de `dourado` pode estar incompleta, contendo as rotas para a rede do Departamento de Matemática e do campus, além da rota padrão, apontando para o ponto de passagem do Centro de Computação da Universidade do Pantanal (`capivara`). Desta forma, qualquer pacote para `jacare` será enviado para `capivara` ao invés de o ser para `piranha`, o ponto de passagem do Departamento de Física. Ao receber tal datagrama, `capivara` notará que esta é uma opção ruim de escolha de roteamento e irá repassar o pacote para `piranha`, ao mesmo tempo em que irá retornar uma mensagem de Redirecionamento ICMP para `dourado` avisando da melhor opção de roteamento.

Agora, este parece ser o meio mais inteligente de evitar a configuração manual, não somente desta, mas da maioria das rotas básicas. De qualquer forma é importante frisar que basear-se em sistemas de rotas dinâmicas, seja RIP ou redirecionamento ICMP não é sempre a melhor opção, pois não há praticamente forma alguma de verificar se as informações de roteamento são autênticas. Isso pode permitir que informações escusas possam prejudicar toda uma rede. Por esta razão, há algumas versões de código de rede `Linux` que tratam mensagens de redirecionamento que afetam roteamentos de rede como se elas fossem somente redirecionamentos de máquinas.

[13] Internet Control Message Protocol

2.6 O Sistema de Nomes de Domínios

2.6.1 Resolução de Nomes de Máquinas

◇ Conforme descrito anteriormente, endereçamento em uma rede TCP/IP envolve um número de 32 bits, que certamente será difícil de relembrar quando tratamos com diversas máquinas. De qualquer forma, máquinas podem ser conhecidas por um nome em especial, como `limeira` ou `campinas`. Desta forma é transferida para a aplicação a tarefa de encontrar o endereço IP correspondente ao nome informado. Este processo é chamado de *resolução de nomes de máquinas*. Uma aplicação que deseje encontrar um endereço IP de uma determinada máquina não necessita ter as suas próprias rotinas de pesquisa de máquinas e endereços IP. Ao invés disso ela pode utilizar diversas funções de bibliotecas que fazem isso de forma transparente, chamadas `gethostbyname(3)` e `gethostbyaddr(3)`. Tradicionalmente, estas e diversas outras funções estão agrupadas em uma biblioteca em separado denominada `resolver`. No `Linux`, elas fazem parte da `libc` padrão. Coloquialmente esta coleção de funções será referenciada como "resolvedor".

Em uma pequena rede Ethernet ou mesmo em um pequeno conjunto delas, não é muito difícil manter uma tabela de mapeamento de nomes de máquinas e seus endereços. Esta informação é normalmente mantida em um arquivo denominado /etc/hosts. Ao adicionar máquinas à rede ou removê-las, o arquivo `hosts` deverá ser atualizado em todas as máquinas da rede. Obviamente isso se tornará inviável em redes que contenham mais que algumas poucas máquinas.

Uma solução para o problema é a utilização do NIS, *Sistema de Informações de Rede*[14] desenvolvido pela Sun Microsystems, coloquialmente denominado YP ou *Páginas Amarelas*[15]. NIS armazena o arquivo `hosts` (e outras informações) em uma base de dados mestre em uma máquina servidora, a partir da qual os clientes podem recuperar as informações toda vez que seja necessário. De qualquer forma, esta abordagem somente pode ser utilizada por redes de tamanho médio, pois envolve a manutenção de um arquivo `hosts` centralmente e a sua distribuição através de todos os equipamentos da rede.

Na Internet, as informações foram inicialmente armazenadas em um único arquivo `HOSTS.TXT` também. O arquivo era mantido no Centro de Informações da Rede ou NIC e tinha que ser transferido e atualizado por todos os sites integrantes da rede. Quando esta cresceu, diversos problemas começaram a surgir. Além do trabalho

[14] Network Information System
[15] Yellow Pages

adicional na manutenção do arquivo e na sua instalação, a carga nos servidores que o distribuíam começou a ficar muito grande. E ainda mais grave foi o problema de que todos os nomes tinham que ser registrados no NIC para garantir que o mesmo nome não fosse utilizado mais de uma vez.

Devido a isso, em 1984, um novo sistema de resolução de nomes foi adotado, O *Sistema de Nomes de Domínio*[16]. DNS foi desenvolvido por Paul Mockapetris, e resolveu ambos os problemas simultaneamente.

2.6.2 Entradas DNS

O DNS organiza o nome das máquinas em uma hierarquia de domínios. Um domínio é uma coleção de sites que estão relacionados de alguma forma: formam uma rede formal (por exemplo, as máquinas de uma campus ou todas as máquinas da BITNET), pertencem a uma determinada organização (a rede do governo de um País), ou estão geograficamente próximas. Por exemplo, universidades brasileiras estão agrupadas no domínio edu.br, com cada uma usando um *subdomínio* em separado, o qual pode ser subdividido e sob o qual as suas máquinas estarão configuradas. A Universidade do Pantanal pode ter um domínio chamado por exemplo `pantanal.edu.br`, com a rede do Departamento de Matemática definida como `mat.pantanal.edu.br`. Máquinas em uma rede departamental terão o nome do domínio adicionado ao seu nome individual. Então `jacare` será conhecida como `jacare.mat.pantanal.edu.br`. Esta denominação é chamada de *nome de domínio totalmente qualificado*, ou FQDN, o qual identifica uma única máquina em todo o mundo.

A Figura 2.3 mostra uma seção de um espaço de nome de domínio. A entrada na raiz desta árvore, a qual é definida por um simples ponto, é apropriadamente chamada de *domínio raiz*, e engloba todos os demais domínios. Para indicar que um nome de uma máquina está no formato totalmente qualificado, ao invés de estar no formato de nome relativo de algum domínio local, ele será definido com um ponto ao final. Isso significa que o último componente do nome é o domínio raiz.

Dependendo de sua localização na hierarquia de nomes, um domínio pode ser denominado de nível primário, secundário ou terciário. Mais níveis podem ocorrer, porém são muito raros. Há alguns domínios de primeiro e segundo nível que serão vistos com alguma freqüência:

[16] Domain Name System

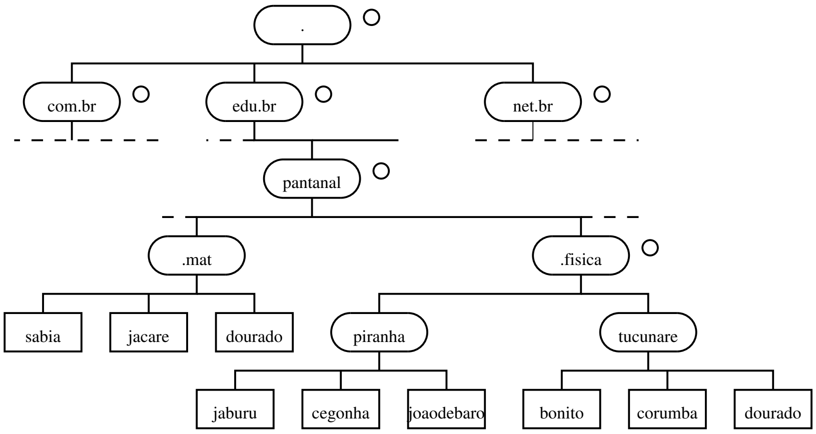

Figura 2.3: Parte do Espaço de Nome de Domínio

`.br` Indica os sites localizados no Brasil. Outros exemplos são: .es - Espanha, .ar - Argentina, .uk - Reino Unido, etc.. Note que os Estados Unidos são os únicos a não usarem um sufixo de primeiro nível.

`edu.br` Destinadas a instituições educacionais como universidades, etc.. Para outros países podemos ter .edu.es, .edu.ar, etc.. Nos Estados Unidos teremos somente o sufixo .edu.[17]

`com.br` Companhias, organizações comerciais, etc..

`org.br` Organizações não comerciais. Normalmente redes privadas UUCP estão neste domínio.

`net.br` Pontos de passagem e outras máquinas administrativas estão nesta rede.

`mil.br` Instituições militares brasileiras.

`gov.br` Instituições governamentais brasileiras.

No site `http://www.registro.fapesp.br`, que é o órgão mantenedor do DNS do domínio *.br*, podem ser encontrados diversos outros identificadores de domínios, inclusive para pessoas físicas.

[17] no Brasil as instituições de ensino não seguem à risca este padrão. Elas foram os primeiros órgãos a se conectar a Internet e utilizaram simplesmente o sufixo *.br*.

O código de País é baseado no seu nome, sendo utilizada a tabela ISO-3166 que atribui duas letras a cada País. A Finlândia, por exemplo, utiliza o domínio `fi`, `fr` é usado pela França, `de` pela Alemanha, ou `it` pela Itália etc. Sob esses domínios de primeiro nível, cada País tem a liberdade de organizar os nomes das máquinas da forma que quiser. A maioria dos Países tem um domínio de segundo nível similar ao utilizado nos EUA. Por exemplo, a Austrália tem um domínio de primeiro nível denominado .au e domínios de segundo nível denominados `com.au`, `edu.au`, e assim por diante. Alguns como a Alemanha não utilizam níveis extras, mas utilizam nomes mais longos que referenciam diretamente um domínio em particular. Por exemplo, não é incomum ver máquinas com nomes como `ftp.informatik.uni-erlangen.de`.

Evidentemente, esses domínios nacionais não implicam que a máquina esteja localizada na realidade naquele País. Ele somente indica que ela foi registrada no NIC daquele País. Uma empresa sueca pode ter uma filial na Austrália, e ainda assim ter todas as máquinas registradas no domínio primário `se`.

Organizando um espaço de nomes em uma hierarquia de domínios resolve de forma elegante o problema de nomes únicos. Com o DNS, um nome de máquina tem que ser único no domínio ao qual ela pertença, garantindo-se assim que ele seja único em todo o mundo. Além disso, nomes totalmente qualificados são mais simples de serem lembrados. Por si só, estas são razões muito boas para se dividir um grande domínio em diversos subdomínios.

O DNS faz ainda mais do que isso: permite delegar autoridade sobre subdomínios a seus administradores. Por exemplo, os mantenedores do centro de computação da Universidade do Pantanal podem criar um subdomínio para cada departamento, nós já encontramos alguns deles como por exemplo, `mat`.

Ao encontrar a rede do Departamento de Matemática muito grande e caótica de ser administrada de fora, pode-se simplesmente passar o controle do domínio `mat.pantanal.edu.br` para os administradores daquela rede. Eles terão toda a liberdade de utilizar os nomes que queiram e assinalar os endereços IP que desejem às maquinas de sua rede, sem qualquer interferência externa (dentro de seu domínio e de sua faixa de endereços).

O nome é dividido em *zonas*, cada uma roteando para um domínio: o *domínio* `pantanal.edu.br` engloba todas as máquinas da Universidade do Pantanal, enquanto a *zona* `pantanal.edu.br` inclui somente as máquinas que estão *diretamente* ligadas ao Centro de Computação. As máquinas do Departamento de Matemática pertencem à uma zona diferente, chamada `mat.pantanal.edu.br`. Na figura 2.3, o início da zona está marcada com um pequeno círculo à direita do nome do domínio.

2.6.3 Resolução de nomes com DNS

Num primeiro momento, todas estas informações sobre domínios e zonas podem parecer um pouco confusas. Afinal se nenhuma autoridade central controla os nomes que são definidos para as máquinas, como as aplicações poderão descobrir ou encontrar uma máquina em todo o planeta.

Aqui começa a parte realmente engenhosa sobre o DNS. Caso se deseje encontrar o endereço IP da máquina `jacare`, então, o DNS responderá: pergunte às pessoas que a gerenciam e elas responderão.

Na verdade, DNS é uma base de dados gigantesca que está distribuída. Ela é implementada através dos denominados servidores de nomes que fornecem informações sobre um determinado domínio ou conjunto de domínios. Para cada zona, há no mínimo dois servidores de nomes que detêm informações sobre as máquinas daquela zona. Para obter o endereço IP de `jacare`, tudo o que se deve fazer é contactar o servidor de nomes da zona `pantanal.edu.br`, o qual retornará os dados solicitados.

É mais fácil falar do que fazer, é o que se poderá imaginar num primeiro momento. Então como fazer para se comunicar com o servidor de nomes da Universidade do Pantanal? Caso o seu computador não esteja equipado com um oráculo capaz de resolver todos os nomes da Internet, o DNS resolverá esta questão. Caso a aplicação necessite, por exemplo, pesquisar as informações na máquina `jacare`, ele contatará inicialmente o servidor local de nomes, o qual efetuará a pesquisa interativamente. Ela é iniciada através do envio de uma solicitação para o servidor de nomes do domínio raiz, perguntando qual o endereço da máquina `jacare.mat.pantanal.edu.br`. O servidor de nomes raiz reconhece que este nome não pertence à sua zona de autoridade, mas que ela pertence ao domínio sob o nível `.br`. Adicionalmente indica que deve ser contatado o servidor de nomes da zona `.br`, o qual contém a lista de todos os servidores `.br` com os seus respectivos endereços. O servidor de nomes local então irá pesquisar um dos servidores de nomes raiz, por exemplo `amon.fapesp.br`. De uma forma similar o servidor de nomes raiz sabe que o domínio `pantanal.edu.br` é mantido pela própria Universidade e indica os seus servidores. O servidor de nomes local irá então enviar a pesquisa de endereço do servidor `jacare` para um dos servidores de nomes da Universidade, o qual finalmente reconhece o nome como pertencente à sua zona e retorna o endereço IP correspondente.

Desta forma, apesar de aparentemente ser gerado um tráfego intenso na pesquisa de endereços IP, ele é realmente minúsculo quando comparado com a quantidade de dados que teria que ser transferida através do método da transferência do arquivo `HOSTS.TXT`. Mas certamente há muito espaço para melhorias.

Costumeiramente a biblioteca que resolve nomes, ao invés de conduzir a pesquisa DNS por si própria, irá delegar esta tarefa ao servidor de nomes que esteja sendo executado na rede local. Este servidor irá executar as pesquisas DNS conforme descrito acima e retornará o resultado à estação solicitante.

Para melhorar o tempo de resposta de pesquisas futuras, o servidor de nomes irá armazenar as informações obtidas em um *cache* local. Da próxima vez que a máquina `pantanal.edu.br` for solicitada, o servidor de nomes local não terá que executar a mesma operação novamente, mas contatará o servidor de nomes `pantanal.edu.br` diretamente.[18]

Obviamente, o servidor de nomes não irá manter estas informações indefinidamente, e sim as descartará após algum tempo. Este intervalo de expiração é chamado de *tempo de vida* ou TTL. Cada intervalo na base de dados DNS é definida pelos administradores responsáveis pela zona.

2.6.4 Servidor de Nomes do Domínio

Servidores de nomes que contêm as informações dos equipamentos da zona são chamados *autoritativos* para a zona e algumas vezes referenciados como *servidores mestres de nomes*. Qualquer pesquisa por uma máquina na zona, irá finalizar em um destes servidores.

Para disponibilizar uma imagem coerente da zona, o servidor mestre de nomes deve ser sincronizado eficientemente. Isso pode ser obtido tornando-o o servidor *primário* e transformando os demais servidores em *secundários*, os quais recebem os dados da zona a partir do servidor primário em intervalos regulares.

Razões para se ter diversos servidores de nomes é a possibilidade de distribuição de carga e a necessidade de redundância. Quando um servidor de nomes falha de uma forma benigna, como problemas de hardware ou a perda de conexão com a rede, todas as pesquisas serão direcionadas para outros servidores. Evidentemente

[18] Caso isso não ocorresse o sistema DNS seria tão ruim como qualquer outro método, uma vez que cada pesquisa deveria envolver o servidor de nomes raiz.

este esquema não protege a rede de mal funcionamento de software por exemplo.

Evidentemente pode-se querer um servidor de nomes que não seja autoritativo para nenhum domínio.[19] Este tipo de servidor é útil ainda para conduzir pesquisas DNS para aplicações que são executadas na rede local, que são colocadas no cache do servidor. É também denominado como servidor *somente para cache*.

2.6.5 A Base de Dados DNS

Conforme descrito anteriormente, o DNS não lida somente com endereços IP de máquinas, mas trata também da troca de informações entre servidores de nomes. Há ainda todo um conjunto de diferentes tipos de entradas na base de dados DNS que pode ser utilizado.

Uma parte única da informação da base de dados DNS é chamada *registro de recurso*, ou RR em seu formato resumido. Cada registro tem um tipo associado, descrevendo o tipo de dado que ele representa e uma classe especificando o tipo de rede ao qual ele se aplica, destinada à resolução de necessidades posteriores de diferentes esquemas de endereçamento, como endereços IP (a classe IN), ou endereços em redes Hesiod (usadas no MIT), e algumas outras. O tipo típico de registro de recurso é o registro A que associa um domínio totalmente qualificado com um endereço IP.

Uma máquina pode ter mais de um nome. Um destes será identificado como oficial ou *nome canônico da máquina*, os demais são denominados nomes alternativos ao oficial[20]. A diferença do nome canônico é que ele possui um registro tipo A associado, enquanto os demais têm somente registros de tipo CNAME que apontam para o nome canônico do nome da máquina.

Não veremos aqui todos os tipos de registros, uma vez que pouparemos alguns para capítulos posteriores, porém vamos comentar alguns deles. A descrição a seguir mostra uma parte da base de dados de domínios que está carregada nos servidores de nomes da zona `fisica.pantanal.edu`.

```
;
; Informações Autoritativas em fisica.pantanal.edu.br
@                    IN    SOA         {
```

[19] Um servidor de nomes deve prover serviços de nome para pelo menos a `máquina local - localhost` e a interface local `127.0.0.1`.

[20] alias name

```
                      piranha.fisica.pantanal.edu.br.
                      hostmaster.piranha.fisica.pantanal.edu.br.
                      1034            ; número serial
                      360000          ; atualização
                      3600            ; tentativa
                      3600000         ; expiração
                      3600            ; ttl padrão
                    }
;
; Servidores de Nomes
                           IN    NS       piranha
                           IN    NS       sabia.mat.pantanal.edu.br.
sabia.mat.pantanal.edu.br. IN    A        149.76.4.23
;
; Física Teórica (sub-rede 12)
piranha                 IN    A        149.76.12.1
                        IN    A        149.76.1.12
nameserver              IN    CNAME    piranha
cegonha                 IN    A        149.76.12.2
jacare                  IN    A        149.76.12.4
corumba                 IN    A        149.76.12.5
dourado                 IN    A        149.76.12.6
...
; Laboratório (sub-rede 14).
paraguai              IN    A        149.76.14.1
parana                IN    A        149.76.14.7
lontra                IN    A        149.76.14.12
...
```

Além dos registros A e CNAME, pode-se ver um registro especial no início do arquivo, utilizando diversas linhas. Este é o registro de recursos SOA, sinalizando o *Início de Autoridade*, o qual contém informações gerais da zona na qual o servidor é autoritativo. Ele define por exemplo o tempo de vida padrão de todos os registros.

Note que todos os nomes no arquivo de exemplo que não finalizem com um ponto, devem ser interpretados como relativos ao domínio `pantanal.edu.br`. O nome especial "`@`" usado no registro SOA referencia-se ao domínio indicado por ele mesmo.

Conforme visto acima, o servidor de nomes do domínio `pantanal.edu.br` deve possuir informações sobre a zona `fisica` para apontar as pesquisas para o servidor de nomes `piranha`. Isso é geralmente obtido através de um par de registros: o

NS que fornece o FQDN do servidor e um registro de tipo A que associa um endereço ao nome. Uma vez que esses registros utilizam o espaço de nome em conjunto, eles são costumeiramente chamados *registros colados*. Eles são a única instância de registros onde uma zona superior mantém informações sobre as zonas subordinadas. Os registros colados do servidor de nomes para `fisica.pantanal.edu.br` são mostrados a seguir.

```
;
; Dados de zona pantanal.edu.br
@                   IN      SOA         {
                     linux12.gcc.pantanal.edu.br.
                     hostmaster.linux12.gcc.pantanal.edu.br.
                     233              ; número serial
                     360000           ; atualização
                     3600             ; tentativa
                     3600000          ; expiração
                     3600             ; ttl padrão
                   }
....
;
; registros colados  para a zona fisica.pantanal.edu.br
fisica              IN     NS       piranha.fisica.pantanal.edu.br.
                    IN     NS       sabia.mat.pantanal.edu.br.
piranha.fisica      IN     A        149.76.12.1
sabia.mat           IN     A        149.76.4.23
...
```

2.6.6 Resolução Reversa

Além da pesquisa do endereço IP pertencente à máquina, é desejável algumas vezes descobrir-se o nome canônico correspondente a um determinado endereço. Ele é chamado de *mapeamento reverso* e é usado de maneira geral pelo serviço de rede para verificar a identificação dos clientes. Ao usar um arquivo `hosts` simples, pesquisas reversas envolvem a busca por uma máquina que atenda pelo endereço IP em questão. Com o DNS, uma longa e exaustiva pesquisa por um espaço de nome está fora de questão. Ao invés disso um domínio especial, `in-addr.arpa`, foi criado com o conteúdo de todas as máquinas em uma notação de quatro campos. Por exemplo o endereço IP de `149.76.12.4` corresponde ao nome `4.12.76.149.in-addr.arpa`. O tipo de registro que liga estes nomes ao seu nome canônico é denominado PTR.

Criar uma zona de autoridade normalmente significa que seus administradores têm controle total sobre seus endereços e nomes. Uma vez que eles usualmente têm uma ou mais redes IP ou sub-redes em suas mãos, há um mapeamento de uma para várias entre zonas DNS e redes IP. O Departamento de Física, por exemplo, contém as sub-redes `149.76.8.0`, `149.76.12.0` e `149.76.14.0`. Como conseqüência, novas zonas no domínio `in-addr.arpa` devem ser criadas junto com a zona `fisica` e delegada aos administradores da rede do Departamento: `8.76.149.in-addr.arpa`, `12.76.149.in-addr.arpa` e `14.76.149.in-addr.arpa`. De outra forma, a instalação de uma nova máquina no laboratório exigiria um contato com seu domínio superior para a introdução de um novo endereço no arquivo de zona `in-addr.arpa`.

A base de dados de zona da sub-rede 12 é mostrada no arquivo abaixo. Logo após, são mostrados os registros colados correspondentes na base de dados de seu domínio superior, no extrato do arquivo `named.rev`.

```
;
; o domínio  12.76.149.in-addr.arpa.
@               IN     SOA   {
                piranha.fisica.pantanal.edu.br.
                hostmaster.piranha.fisica.pantanal.edu.br.
                233 360000 3600 3600000 3600
                  }
2               IN     PTR       cegonha.fisica.pantanal.edu.br.
4               IN     PTR       jaburu.fisica.pantanal.edu.br.
5               IN     PTR       joaodebarro.fisica.pantanal.edu.br.


Um extrato do arquivo named.rev para a rede 149.76.

;
; o domínio  76.149.in-addr.arpa.
@                  IN       SOA         {
                   linux12.gcc.pantanal.edu.br.
                   hostmaster.linux12.gcc.pantanal.edu.br.
                   233 360000 3600 3600000 3600
                  }
...
; sub-rede 4: Departamento de Matemática
1.4             IN     PTR      dourado.mat.pantanal.edu.br.
```

```
17.4            IN     PTR      jacare.mat.pantanal.edu.br.
23.4            IN     PTR      sabia.mat.pantanal.edu.br.
...
; sub-rede 12: Departamento de Física, zona separada
12              IN     NS       piranha.fisica.pantanal.edu.br.
                IN     NS       sabia.mat.pantanal.edu.br.
piranha.fisica.pantanal.edu.br.  IN  A 149.76.12.1
sabia.mat.pantanal.edu.br.       IN  A 149.76.4.23
...
```

Uma conseqüência importante destas zonas reside no fato delas permitirem somente a criação de subconjuntos de endereços IP válidos, e o mais importante, as máscaras de rede têm que estar rigorosamente dentro dos limites. Todas as sub-redes na Universidade do Pantanal têm uma máscara de rede igual a `255.255.255.0`, permitindo que uma zona `in-addr.arpa` possa ser criada para cada sub-rede. De qualquer forma se uma máscara de rede `255.255.255.128` fosse criada em seu lugar, a criação de zonas para a sub-rede `149.76.12.128` seria impossível, pois não há forma de dizer ao DNS que o domínio `12.76.149.in-addr.arpa` foi subdividido em duas zonas de autoridade, com nomes de máquinas variando de `1` até `127`, e `128` até `255`, respectivamente.

Capítulo 3

Configurando Hardware de Rede

3.1 Dispositivos, Programas de Controle e Outros

Até aqui falamos somente sobre interfaces e características gerais da rede TCP/IP, mas não exatamente sobre *o que* realmente acontece quando o "código de rede" no kernel acessa um componente de hardware. Para isso, temos que falar um pouco mais sobre o conceito de interfaces e programas de controle.

Primeiro, é claro, existe o hardware por si mesmo, como por exemplo uma placa Ethernet: esta é uma peça de Epoxy, desordenada em muitos e minúsculos chips com números sobre eles, colocados em um conector do PC. Isto é o que nós geralmente chamamos de um dispositivo.

Para que seja possível usar uma placa Ethernet, funções especiais têm que estar presentes no kernel do `Linux`, as quais compreendem de modo particular como se relacionar com este dispositivo. Estes são chamados programas de controle[1] de dispositivo. Por exemplo, o `Linux` possui seus programas de controle de dispositivo para vários tipos de placas Ethernet, os quais são muito similares na sua função. Eles são conhecidos como "programas de controle de dispositivos seriais de Becker", e tem este nome devido ao seu autor: Donald Becker. Um exemplo diferente é o programa de controle D-Link, o qual manipula uma placa de rede D-Link conectada

[1] drivers

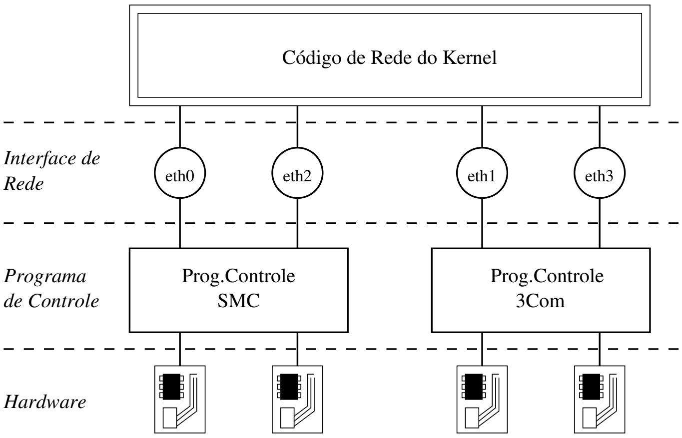

Figura 3.1: O relacionamento entre programas de controle, interfaces e o hardware

a uma porta paralela.

Mas o que significa quando dizemos que um programa de controle manipula um dispositivo? Vamos voltar para a placa Ethernet descrita acima. O programa de controle tem que ser capaz de se comunicar com o(s) programa(s) que estão na placa de algum modo: ele tem que enviar comandos e dados para a placa, enquanto a placa deve entregar todos os dados recebidos para o programa de controle.

Em PCs, esta comunicação ocorre em lugares da área de memória designadas como de Entrada e Saída, que são mapeados para os registradores que estão na placa. O kernel tem que enviar todos comandos e dados para a placa através destes registradores. A memória de E/S é geralmente descrita a partir de seu início ou *endereço base*. Geralmente os endereços base para as placas Ethernet são `0x300` ou `0x360`.

Usualmente, não é necessário preocupar-se com nenhuma informação de hardware, tal como endereço base, pois o kernel faz tentativas na hora da inicialização do sistema para detectar a localização da placa. Isto é chamado de teste automático, composto pela leitura realizada pelo kernel de várias localizações de memória e comparação dos dados lidos com o que deveria ser detectado caso uma certa placa Ethernet estivesse instalada. No entanto há placas Ethernet que não podem ser detectadas automaticamente. Isto às vezes é o que ocorre com placas baratas Ethernet que não são cópias completas de outras placas padrão. Por outro lado, o kernel tentará detectar somente um dispositivo Ethernet durante a inicialização.

Caso se esteja usando mais que uma placa, terá que ser indicada explicitamente ao kernel a presença de placas adicionais.

Um outro parâmetro de informação que pode ser passado para o kernel é o canal de pedido de interrupção - IRQ. Geralmente os componentes do hardware interrompem o kernel quando eles necessitam de sua atenção, por exemplo, quando dados chegam, ou quando ocorre alguma condição especial. Em um PC, interrupções podem ocorrer em um dos 15 canais de interrupção numerados 0, 1 e 3 até 15. O número da interrupção atribuído ao componente do hardware é chamado *número do pedido de interrupção* ou IRQ.[2]

Como descrevemos no capítulo 2, o kernel acessa um dispositivo através da interface. Interfaces oferecem um conjunto abstrato de funções que são idênticas para todos os tipos de hardware, tais como mandar ou receber um pacote de informações (datagrama) por um sistema de comunicação.

Interfaces são identificadas por meio de nomes. Estes nomes são definidos internamente no kernel, e não são iguais aos arquivos de dispositivos `/dev` do diretório de mesmo nome. Nomes típicos são `eth0`, `eth1`, etc., para interfaces Ethernet. A atribuição de interfaces para dispositivos usualmente depende da ordem na qual estes são configurados; por exemplo a primeira placa Ethernet instalada torna-se-á `eth0`, a próxima será `eth1`, e assim por diante. A única exceção para esta regra são as interfaces SLIP, que são atribuídas dinamicamente; isto é, sempre que uma conexão SLIP for estabelecida, uma interface diferente pode ser atribuída para uma porta serial.

O quadro dado na figura 3.1 procura mostrar o relacionamento entre o hardware, programas de controle de dispositivos e interfaces.

Quando iniciado, o kernel indica quais dispositivos ele detecta, e quais interfaces ele instala. A seguir está uma amostra de uma típica tela de inicialização:

```
.
.
This processor honours the WP bit even when in supervisor mode. Good.
Floppy drive(s): fd0 is 1.44M
Swansea University Computer Society NET3.010
IP Protocols: ICMP, UDP, TCP
PPP: version 0.2.1 (4 channels) OPTIMIZE_FLAGS
TCP compression code copyright 1989 Regents of the University of California
```

[2]IRQs 2 e 9 são idênticas porque o PC possui dois processadores com um sistema em cascata de interrupções com oito IRQ's cada; o processador secundário é conectado com o IRQ 2 do primeiro.

```
PPP line discipline registered.
SLIP: version 0.7.5 (4 channels)
CSLIP: code copyright 1989 Regents of the University of California
dl0: D-Link DE-600 pocket adapter, Ethernet Address: 00:80:C8:71:76:95
Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Linux version 1.1.11 (okir@monad) #3 Sat May 7 14:57:18 MET DST 1994
```

As mensagens indicam que o kernel foi compilado com TCP/IP habilitado e os programas de controle para SLIP, CSLIP e PPP foram incluídos. A terceira linha debaixo para cima indica que a placa de rede D-Link foi detectada, e instalada como interface `dl0`. Se você tem diferentes tipos de placas Ethernet, o kernel geralmente imprimirá uma linha de início com `eth0`, seguido pelo tipo de placa detectada. Se você tem uma placa Ethernet instalada, mas não visualiza nenhuma destas mensagens, isto significa que o kernel é incapaz de detectar sua placa corretamente. Isto será tratado em uma seção posterior.

3.2 Configuração do Kernel

Muitas distribuições do `Linux` vêm com discos de inicialização com suporte a todos os tipos comuns de hardware do PC. Isto significa que o kernel destes discos possui muitos tipos de programas de controle configurados, os quais podem não ser necessários, e que utilizariam memória preciosa caso fossem carregados indiscriminadamente. Conseqüentemente, cada máquina geralmente rodará seu próprio kernel, incluindo somente aqueles programas de controle de dispositivos necessários ao seu funcionamento e adequados à sua configuração, o que tornará o sistema mais eficiente e ágil no seu processamento.

Ao rodar um sistema `Linux`, é necessário familiarizar-se com a construção de um kernel. Os princípios desta atividade estão explicados no Guia de Matt Welsh "Instalando e Iniciando o Linux", que também é parte das séries do projeto de documentação do `Linux` e no Guia do Usuário do Conectiva Linux. Por conseqüência, nesta seção, discutiremos somente aquelas opções de configuração que afetem a rede.

Ao executar o programa `make config`, inicialmente aparecerão questões sobre as configurações gerais, como por exemplo sobre a necessidade de simulação de coprocessador matemático no kernel, etc.. Uma destas perguntará sobre a necessidade de suporte a redes TCP/IP. Deve-se responder com `y` para que o sistema seja inicializado com um kernel que contenha as funcionalidades de rede.

3.2.1 Opções do Kernel no Linux 1.0 e Acima

Após a conclusão da parte de opções gerais, a configuração irá perguntar por várias outras características, tais como drivers SCSI, etc.. A lista subseqüente de questões trata do suporte de rede. O conjunto exato das opções de configuração está em constante mudança devido ao processo em desenvolvimento. Uma lista de opções típica oferecida pela maioria das versões do kernel 2.0 se parece com o que segue:

```
*
* Network device support
*
Network device support? (CONFIG_ETHERCARDS) [y]
```

Apesar do nome da macro indicada nos parênteses, deve-se responder a esta pergunta com `y`, caso se queira usar *qualquer* tipo de dispositivo da rede, sobretudo se este é uma Ethernet, SLIP ou PPP. Quando respondida esta questão com `y`, o suporte para dispositivos do tipo Ethernet são automaticamente instalados. O suporte para outros tipos de programas de controle de rede devem ser autorizados separadamente:

```
SLIP (serial line) support? (CONFIG_SLIP) [y]
SLIP compressed headers (SL_COMPRESSED) [y]
PPP (point-to-point) support (CONFIG_PPP) [y]
PLIP (parallel port) support (CONFIG_PLIP) [n]
```

Estas questões interessam aos diversos protocolos da camada de ligação suportados pelo `Linux`. SLIP permite o transporte de datagramas IP através de linhas seriais. A opção cabeçalho comprimido que fornece o suporte para CSLIP, consiste em uma técnica de compressão da parte inicial dos datagramas TCP/IP, que podem chegar ao tamanho de 3 bytes. Note que esta opção do kernel não aciona o CSLIP automaticamente, ela simplesmente fornece as funções necessárias ao kernel para poder executá-lo.

PPP é outro protocolo utilizado para conduzir tráfego na rede através de linhas seriais. Ele é muito mais flexível que o SLIP, e não é limitado a IP, pois suporta também IPX.

PLIP fornece um modo de enviar datagramas IP através de conexões através de portas paralelas. Isto é geralmente usado para comunicação com PCs rodando em DOS e em ambientes que não contenham a estrutura de uma rede local disponível.

As questões a seguir tratam de placas Ethernet de vários fabricantes. Como muitos programas de controle estão sempre sendo desenvolvidos, novas perguntas sempre são adicionadas a esta seção. Se você quiser construir um kernel, poderá habilitar mais de um programa de controle de dispositivos, caso seja necessário.

```
NE2000/NE1000 support (CONFIG_NE2000) [y]
WD80*3 support (CONFIG_WD80x3) [n]
SMC Ultra support (CONFIG_ULTRA) [n]
3c501 support (CONFIG_EL1) [n]
3c503 support (CONFIG_EL2) [n]
3c509/3c579 support (CONFIG_EL3) [n]
HP PCLAN support (CONFIG_HPLAN) [n]
AT1500 and NE2100 (LANCE and PCnet-ISA) support (CONFIG_LANCE) [n]
AT1700 support (CONFIG_AT1700) [n]
DEPCA support (CONFIG_DEPCA) [n]
D-Link DE600 pocket adaptor support (CONFIG_DE600) [y]
AT-LAN-TEC/RealTek pocket adaptor support (CONFIG_ATP) [n]
*
* CD-ROM drivers
*
...
```

Finalmente, na seção do sistema de arquivos, o programa de configuração perguntará sobre o suporte para NFS, o sistema de arquivos em rede. O NFS permite que sistemas de arquivos sejam exportados para várias máquinas, as quais o tratam como se fossem arquivos locais ou um disco auxiliar do equipamento.

```
NFS filesystem support (CONFIG_NFS_FS) [y]
```

3.2.2 Opções do kernel no Linux 2.0 e Acima

No Linux 2.0.x, ao qual adicionou-se o suporte para IPX, o procedimento da configuração mudou ligeiramente. A seção das opções gerais agora pergunta sobre o desejo de suporte à rede de forma geral. Isto é imediatamente seguido por um par de perguntas com opções variadas de rede.

```
*
* Networking options
*
TCP/IP networking (CONFIG_INET) [y]
```

Para usar a rede TCP/IP, deve-se responder essa pergunta com `y`. Ao se responder com `n`, de qualquer forma, ainda será possível compilar o kernel com o suporte a IPX.

```
IP forwarding/gatewaying (CONFIG_IP_FORWARD) [n]
```

É necessário habilitar esta opção para que o sistema aja como uma conexão entre duas redes Ethernets, ou entre uma Ethernet e uma ligação SLIP, etc.. Embora ela não interfira caso seja habilitada como padrão, possivelmente será necessário desabilitá-la para configurar uma máquina como firewall.

Firewalls são clientes que estão conectados em duas ou mais redes, mas não permitem o livre tráfego entre elas. Eles são geralmente usados para fornecer aos usuários de uma rede de uma companhia o acesso a Internet com um risco mínimo para a rede interna. Aos usuários será permitido acesso interno ao firewall e o uso dos serviços da Internet, mas as máquinas da companhia serão protegidas dos ataques externos, pois nenhuma conexão de entrada deverá atravessar o firewall.

```
*
* (it is safe to leave these untouched)
*
PC/TCP compatibility mode (CONFIG_INET_PCTCP) [n]
```

Esta opção trabalha em torno de uma incompatibilidade com algumas versões do PC/TCP, uma implementação comercial do TCP/IP baseado em DOS para PCs. Ao habilitá-la, ainda será possível comunicar-se normalmente com máquinas `Unix`, mas a performance pode sofrer interferências com ligações muito lentas.

```
Reverse ARP (CONFIG_INET_RARP) [n]
```

Esta função habilita RARP, o Protocolo Reverso de Definição de Endereço. RARP é usado por clientes sem disco e terminais X na busca de seu endereço IP ao serem inicializados. Deve-se habilitar RARP somente quando se planeje utilizar este tipo de cliente. Este último pacote de utilidades de rede (`net-0.32d`) contém um pequeno utilitário chamado `rarp` que permite adicionar sistemas de cache ao RARP.

```
Assume subnets are local (CONFIG_INET_SNARL) [y]
```

Ao mandar dados via TCP, o kernel tem que quebrar a mensagem dentro de vários pacotes antes de liberá-lo ao IP. Para máquinas que podem ser alcançadas sobre uma rede local, tais como uma Ethernet, pacotes maiores devem ser usados, ao passo que em ligações de longa distância, como linhas discadas ou linhas de dados dedicadas, pacotes menores constituem a melhor opção a ser utilizada.[3] Caso seja habilitado o parâmetro `SNARL`, o kernel irá assumir que somente as redes que são locais terão a comunicação através de pacotes maiores. De qualquer modo, ao se analisar a rede classe B da Universidade do Pantanal, veremos que toda ela é local, mas a maioria das interface das máquinas aponta somente para uma ou duas sub-redes. Ao se habilitar `SNARL`, o kernel irá assumir que *todas* as sub-redes são locais e usar blocos de dados maiores também na comunicação com as demais redes do campus.

Caso se deseje usar blocos de dados com tamanhos menores a serem enviados a máquinas específicas (porque, por exemplo, o dado será enviado através de uma ligação SLIP), pode-se configurar o parâmetro `mtu` do `route`, o qual é descrito no capítulo 5.

```
Disable NAGLE algorithm (normally enabled) (CONFIG_TCP_NAGLE_OFF) [n]
```

Esta opção habilita o suporte a IPX, o qual transporta o protocolo usado por Redes Novell©. Um benefício direto desta funcionalidade é a possibilidade de trocar dados com utilitários IPX do DOS, e possibilitar o tráfego entre suas redes baseadas em Novell através de uma ligação PPP. O suporte para protocolos em um nível mais alto na rede Novell também já está disponível, inclusive uma versão do próprio Netware para a plataforma Linux. A partir do kernel 1.1.16, o `Linux` passou a suportar outro tipo de programa de controle de dispositivos, o falso dispositivo. A questão a seguir é apresentada para iniciar a seção de programa de controle de falso dispositivo.

```
Dummy net driver support (CONFIG_DUMMY) [y]
```

O falso dispositivo na verdade não realiza muitas tarefas, mas é totalmente útil em dispositivos independentes ou em máquinas SLIP. Ele é basicamente uma interface de teste local mascarada. A razão de existir este tipo de interface é que em máquinas que executam o SLIP, mas não possuem Ethernet, tem-se a necessidade de uma interface que sustente o endereço IP durante todo o tempo. Isto é discutido com mais detalhes na seção 5.7.7 do capítulo 5.

[3]Isto serve para evitar fragmentação em ligações que têm um pacote de tamanho máximo muito pequeno.

3.3 Introdução Sobre Programas de Controle de Dispositivos de Rede `Linux`

O kernel do `Linux` suporta um grande número de programas de controle de dispositivos de hardware para diversos tipos de equipamentos. Esta seção fornece uma pequena visão geral de famílias de programas disponíveis, e os nomes das interfaces utilizadas.

Existe uma série de nomes padrões para interfaces no `Linux`, as quais são listadas abaixo. Muitos programas de controle de dispositivos suportam mais que uma interface, sendo todas elas numeradas, como por exemplo em `eth0`, `eth1`, etc..

`lo` A interface local de testes[4] é usada com a finalidade de validar a interface de rede, da mesma forma que um dispositivo de testes de rede. Ela trabalha como um circuito fechado, onde qualquer datagrama escrito na rede será imediatamente retornado para a camada de rede da própria máquina. Automaticamente um dispositivo local de testes está presente no kernel, não fazendo muito sentido se ter mais de um.

`ethn` A placa Ethernet n-th. Este é o nome genérico da interface para a maioria placas Ethernet.

`dln` Estas interfaces acessam um adaptador de rede D-Link DE-600, outro dispositivo Ethernet. Ele é um pouco especial, onde o DE-600 é dirigido através de uma porta paralela. Kernels posteriores a 1.1.22 não utilizam mais uma família especial de nomes para estes dispositivos, incluindo a DE-600 na família eth.

`sln` A interface n-th SLIP. Interfaces SLIP estão associadas com linhas seriais na ordem em que são alocadas, isto é, a primeira linha configurada com SLIP torna-se `sl0`, etc.

`pppn` A interface n-th PPP. Assim como as interfaces SLIP, a interface PPP é associada à linha serial, uma vez que esta seja adequada ao modo PPP.

`plipn` A interface n-th PLIP. PLIP transporta datagramas IP sobre linhas paralelas. Elas são alocadas por um programa de controle de dispositivos PLIP no momento da inicialização do sistema e são mapeadas sobre portas paralelas.

[4] loopback

Para outras interfaces de programas de controle de dispositivos, como ISDN ou AX.25, outros nomes serão introduzidos. Controladores para IPX (protocolo para rede Novell©) e AX.25 (usado por rádio amadores) estão disponíveis. Durante as seções seguintes, discutiremos os detalhes do uso dos programas de controle de dispositivos descritos acima.

3.4 Instalação Ethernet

O código atual de rede `Linux` suporta vários tipos de placas Ethernet. Muitos programas de controle de dispositivos foram escritos por Donald Becker (`becker@cesdis.gsfc.nasa.gov`), que foi o autor de uma família de programas para placas baseadas no chip semicondutor National 8390. Estes têm-se tornado conhecidos como a *Série Programas de Controles de Dispositivos de Becker*. Existem também programas de controle para muitos produtos D-Link, entre eles a placa de rede D-Link que permite acesso a Ethernet através de uma porta paralela. Este programa foi escrito por Bjørn Ekwall (`bj0rn@blox.se`). O programa DEPCA foi escrito por David C. Davies (`davies@wanton.lkg.dec.com`).

3.4.1 Cabeamento Ethernet

Caso se esteja instalando uma rede Ethernet pela primeira vez, algumas poucas palavras sobre cabeamento podem ser úteis neste momento. Ethernet é muito seletivo com relação ao cabeamento apropriado. Cabeamentos do tipo thin ou thick já estão totalmente fora de uso, logo é fortemente sugerido o uso de par trançado em redes de até média demanda, estando disponíveis nas velocidades de 10 Mbps ou 100 Mbps. Estas redes exigirão, além das placas de rede em cada máquina obviamente, um hub, uma espécie de concentrador de conexões e cabos adequados.

3.4.2 Placas Suportadas

Uma lista completa de placas disponíveis está disponível no Como Fazer - Ethernet divulgado mensalmente em `comp.os.linux.announce` por Paul Gortmaker.[5]

Apresentamos a seguir uma lista dentre as muitas placas conhecidas pelo `Linux`. A

[5]Paul pode ser encontrado no `gpg109@rsphysse.anu.edu.au`.

lista atual no Como Fazer[6] é muitas vezes maior do que esta. No entanto, mesmo ao se encontrar uma placa nesta lista, deve ser verificado o Como Fazer primeiro; algumas vezes existem detalhes importantes sobre a operação destas placas. Um exemplo desta questão é o caso de algumas placas Ethernet baseadas em DMA que usam o mesmo canal da controladora Adaptec 1542 SCSI por padrão. A menos que se altere o DMA de qualquer um deles para um canal DMA diferente, se terá uma placa Ethernet escrevendo blocos de dados em localizações arbitrárias no seu disco rígido.

3Com EtherLink 3c503 e 3c503/16 são suportados, assim como 3c507 e 3c509. A placa 3c501 também é suportada.

Novell Eagle NE1000 e NE2000 e uma variedade de cópias. NE1500 E NE2100 também são suportadas.

Western Digital/SMC D8003 e WD8013 (algo como SMC Elite e SMC Elite Plus) são suportadas, assim como o SMC Elite 16 Ultra.

Hewlett Packard HP 27252, HP 27247B, e HP J2405A.

D-Link Placas DE-600, DE-100, DE-200 e DE-220-T. Existe também um kit de correção para o DE-650-T, que é uma placa PCMCIA.[7]

DEC DE200 (32K/64K), DE202, DE100, e DEPCA rev E.

Allied Teliesis AT1500 e AT1700.

Para utilizar alguma destas placas de rede com o `Linux`, pode-se usar uma versão pré-compilada do kernel a partir de uma das distribuições do `Linux`[8]. Estas geralmente contêm programas de controle de dispositivos para todas estas placas, previamente construídos. A longo prazo, de qualquer modo, é melhor rodar kernel individualizado e compilar somente os programas de controle realmente necessários.

---

[6] HOWTO

[7] Ela pode ser obtida, junto com outros materiais relacionados a computadores portáteis em `tsx-11.mit.edu` no caminho `packages/laptops`.

[8] Como por exemplo a Conectiva Linux.

3.4.3 Detecção automática da placa Ethernet

No momento da inicialização do sistema, o código da Ethernet tentará localizar a placa e determinar seu tipo. Elas são analisadas para os seguintes endereços e na seguinte ordem:

Placa	Endereços testados
WD/SMC	`0x300`, `0x280`, `0x380`, `0x240`
SMC 16 Ultra	`0x300`, `0x280`
3c501	`0x280`
3c503	`0x300`, `0x310`, `0x330`, `0x350`, `0x250`, `0x280`, `0x2a0`, `0x2e0`
NEx000	`0x300`, `0x280`, `0x320`, `0x340`, `0x360`
HP	`0x300`, `0x320`, `0x340`, `0x280`, `0x2C0`, `0x200`, `0x240`
DEPCA	`0x300`, `0x320`, `0x340`, `0x360`

Existem duas limitações para o código de teste automático de placas de rede. Primeiro, ele não pode reconhecer todas as placas corretamente. Isto é especialmente verdade para algumas cópias mais baratas de placas padrão, mas também para algumas placas WD80x3. O segundo problema é que o kernel não executa o teste automático para mais de uma placa ao mesmo tempo. Isto é na verdade uma funcionalidade, pois ele supõe que se quer ter controle sobre qual interface é atribuída à determinada placa.

Caso se esteja usando mais de uma placa, ou se o teste automático falhar na detecção da placa, há que explicitar para o kernel, o endereço base da placa e o seu nome.

Na Net-3, podem ser utilizados dois esquemas diferentes para realizar isto. Uma forma é mudar ou adicionar informações ao arquivo `drivers/net/Space.c` que contém o código fonte do kernel, o qual contém todas as informações necessárias sobre os programas de controle de dispositivos. Isto é recomendado somente quando se está familiarizado com o código de rede. Um modo muito mais indicado é fornecer ao kernel esta informação no momento da inicialização do sistema. Caso se esteja utilizando o utilitário `lilo` para iniciar o sistema, é possível passar parâmetros para o kernel, utilizando-se a opção `append` no arquivo `lilo.configuração`. Para passar as informações para o kernel sobre um dispositivo Ethernet, devem ser informados os seguintes parâmetros:

```
ether=irq,endereço_base,param1,param2,nome
```

Os primeiros quatro parâmetros são numéricos, enquanto o último é o nome do dispositivo. Todos os valores numéricos são opcionais. Caso eles sejam omitidos ou ajustados para zero, o kernel tentará detectar o valor através de testes automáticos ou utilizará um valor padrão.

O primeiro parâmetro configura o IRQ atribuído ao dispositivo. Por definição, o kernel tentará detectar automaticamente o canal IRQ. O controlador 3c503 tem um recurso especial que seleciona um IRQ livre da lista 5, 9, 3, 4 e configura a placa para o uso nesta linha.

O parâmetro *endereço_base* fornece o endereço base de entrada e saída da placa. Um valor zero indica ao kernel a necessidade de execução de testes para obtenção destes valores.

Os dois parâmetros restantes devem ser usados de modo diferente por diferentes programas de controle de dispositivos. Para placas com memória compartilhada tal como a WD80x3, eles especificam os endereços de início e fim da área da memória compartilhada. Outras placas geralmente usam *param1* para ajustar o nível de depuração de informação que está sendo indicado. Valores de 1 até 7 denotam aumentos nos níveis de apresentação de mensagens, enquanto que o valor 8 desliga-os completamente. O padrão é igual a 0 (zero). O controlador 3c503 usa *param2* para selecionar o transceptor interno (padrão) ou externo (de valor 1). O primeiro indica um conector de placa BNC, o último indica uma porta AUI.

Caso estejam presentes duas placas Ethernet, pode-se ter uma placa detectada automaticamente pelo `Linux` e passar os parâmetros da segunda placa com `lilo`. No entanto, é necessário certificar-se que o programa de controle de dispositivos não tenha encontrado acidentalmente a segunda placa ao invés da primeira, pois neste caso a segunda não será configurada. Pode-se fazer isso configurando a opção `lilo reserve`, a qual indica ao kernel claramente que evite testar o espaço de Entrada e Saída utilizado pela segunda placa. Por exemplo , para fazer o `Linux` instalar uma segunda placa Ethernet em `0x300` como `eth1`, deve-se informar os seguintes parâmetros para o kernel:

```
reserve=0x300,32 ether=0,0x300,eth1
```

A opção `reserve` garante que nenhum programa de controle de dispositivo utilize o espaço de entrada e saída da placa em testes de detecção automática de dispositivos. Pode-se também usar parâmetros do kernel para que não seja executado o teste automático para `eth0`:

```
reserve=0x340,32 ether=0,0x340,eth0
```

Para desabilitar o teste automático completamente, pode-se especificar um argumento *`endereço_base`* igual a -1:

```
ether=0,-1,eth0
```

3.5 O Programa de Controle PLIP

PLIP funciona em *linhas paralelas IP* e é um meio econômico para redes compostas por somente duas máquinas. Ele usa uma porta paralela e um cabo especial, alcançando velocidades de 10kBps a 20kBps.

PLIP foi originalmente desenvolvido por Crynwr, Inc. Seu projeto é bastante engenhoso (ou, se preferir, um grande trabalho de hacker): por um longo tempo, as portas paralelas nos PCs costumavam ser utilizadas somente com impressoras unidirecionais, ou seja, as oito linhas de dados podem ser usadas para enviar dados do PC para os dispositivos periféricos, mas não do periférico para o PC. PLIP resolve esta limitação através do uso da linha de status da porta cinco como forma de entrada de dados no PC, através da transferência de todos os dados no formato nibbles - pequenos pedaços (metade dos bytes). Este modo de operação é chamado de modo PLIP zero. Hoje, estas portas unidirecionais parecem não ser muito usadas. No entanto, existe também uma extensão chamada modo 1 que usa uma interface de 8 bits completos.

Atualmente, o `Linux` suporta somente o modo 0. Diferentemente das versões anteriores do código PLIP, ele agora tenta ser compatível com as implementações PLIP de Crynwr, assim como o programa de controle PLIP na NCSA `telnet`.[9] Para conectar duas máquinas usando PLIP, é necessário um cabo especial vendido em algumas lojas, conhecido como "Null Printer" ou "Turbo Laplink". É possível no entanto confeccioná-lo facilmente. O Apêndice A mostra como fazê-lo.

O controlador PLIP para o `Linux` é o resultado do trabalho de incontáveis pessoas. Ele é atualmente mantido por Niibe Yutaka. Se compilado no kernel, ele prepara

[9] NCSA `telnet` é um programa popular para DOS que roda TCP/IP sobre Ethernet ou PLIP, e suporta `telnet` e FTP.

uma interface de rede para cada porta de impressora possível, com `plip0` correspondendo à porta paralela `lp0`, `plip1` correspondendo à `lp1`, etc.. O mapeamento da interface para as portas tem o seguinte formato:

Interface	Porta E/S	IRQ
`plip0`	`0x3BC`	7
`plip1`	`0x378`	7
`plip2`	`0x278`	5

Caso se tenha configurado a porta de impressora de um modo diferente, deve-se então mudar estes valores no arquivo `drivers/net/Space.c` no fonte do kernel do `Linux`, e construir um novo kernel.

Este mapeamento não significa, no entanto, que não se possa utilizar estas portas paralelas da forma usual. Elas são acessadas por um controlador PLIP somente quando a interface correspondente é configurada como ativa.

3.6 Os Programa de Controle de Dispositivos SLIP e PPP

SLIP (linha serial IP) e PPP (Protocolo ponto a ponto) são protocolos extensamente usados no envio de blocos de dados IP sobre ligações seriais. Um número significativo de instituições oferecem acessos através de discagem SLIP e PPP para máquinas que estão na Internet, fornecendo assim conectividade IP para pessoas privadas.

Nenhuma modificação no hardware é necessária para se executar SLIP ou PPP. Pode-se usar qualquer porta serial, desde que a sua configuração não seja especificada na rede TCP/IP. Um capítulo em separado descreve como fazê-lo. Por favor, consulte o capítulo 4 para maiores informações.

Capítulo 4

Configurando o Hardware Serial

Existem rumores de que há algumas pessoas de fora que desembarcaram na rede, tendo somente um velho PC e sem dinheiro para gastar em uma conexão dedicada T1 Internet. Para receber sua dose diária de notícias e mensagens sem impedimentos, eles se baseiam em ligações PPP, SLIP, redes UUCP e sistemas de acesso remoto compartilhado (ISPs) que utilizam rede pública de telefonia. Será verdade?[1]

Este capítulo é destinado a ajudar todas aquelas pessoas que utilizam modems para manter suas conexões. Contudo, existem muitos detalhes nos quais não poderemos nos aprofundar neste capítulo, como por exemplo em como configurar um modem para discar. Todos esses tópicos são abrangidos no Como Fazer - Serial mantido por Greg Hankins,[2] o qual é enviado para `comp.os.linux.announce` em bases regulares. Ele pode ser encontrado também em `http://ldp-br.conectiva.com.br/documentos/comofazer/html/HOWTO-INDEX.html`.

---

[1] N.T.: Lembramos que este guia foi originalmente escrito em 1994, e apesar de termos procurado atualizar o maior número possível de informações, algumas notas foram mantidas, como neste caso, para que possamos dar-nos conta de quão rápido foi o crescimento da Internet.

[2] Encontrado em `gregh@cc.gatech.edu`.

4.1 Software de Comunicação para Ligações Via Modem

Existe um grande número de pacotes de comunicação disponíveis para o `Linux`. Muitos deles são *programas de emulação de terminal* que permitem a um usuário utilizar outro computador como se estivesse em frente do console deste. Um programa tradicional de emulação de terminal para `Unices` é o `kermit`. Porém ele nos parece pouco espartano. Existem programas disponíveis com maiores funcionalidades que, por exemplo, suportam dicionários de números de telefones, linguagens de programação para chamadas e acessos a sistemas remotos, etc. Um deles é o `minicom`, que está muito próximo a alguns programas comerciais de emulação de terminal baseados em DOS. Existem também os pacotes de comunicações baseados em interface gráfica, por exemplo `seyon`.

Há também pacotes BBS baseados em `Linux` disponíveis para aqueles que necessitem de um sistema de acesso remoto compartilhado. Alguns destes pacotes podem ser encontrados em `metalab.unc.edu` no caminho `/pub/Linux/system/Network`.

Ao lado dos programas de emulação de terminais, existe também um software que utiliza uma ligação serial não interativa e que transporta dados para ou de outro computador. A vantagem desta técnica é que ela leva menos tempo para realizar a transferência automática de poucos kilobytes do que o tempo necessário para a leitura de uma mensagem on-line em alguma caixa postal ou aquele que se leva para explorar em uma BBS os artigos interessantes. Por outro lado, requer mais espaço de armazenamento no disco, uma vez que geralmente pode-se receber mais informações do que as necessárias.

O nome deste software de comunicação é UUCP, que significa Cópia de Unix para Unix. Trata-se de um programa integrado que copia arquivos de uma máquina para outra, possibilita a execução de programas em uma máquina remota, entre outras utilidades. Ele é freqüentemente usado para transportar mensagens ou notícias em redes privadas. O pacote UUCP de Ian Taylor, que roda sobre o `Linux`, é descrito no capítulo seguinte. Outro software de comunicação não interativa é, por exemplo, usado na Fidonet. Portes de aplicações da Fidonet como `ifmail` também estão disponíveis.

PPP, o protocolo de comunicação para conexões assíncronas possibilita uma forma de comunicação intermediária, permitindo o uso tanto de interatividade como de comunicação assíncrona. Muitas pessoas utilizam o PPP para discarem para suas redes ou para algum outro tipo de servidor público PPP, como um Provedor de

Acesso Internet, a fim de executarem sessões FTP, telnet, etc.. PPP pode ser usado também sobre conexões permanentes ou semi-permanentes, como redes locais interligadas.

4.2 Introdução sobre Dispositivos Seriais

Os dispositivos do kernel do `Unix` que proporcionam acesso aos dispositivos seriais são geralmente chamados de *tty*. Uma abreviação para *Teletype*™, que era um dos principais fabricantes de terminais nos primeiros dias do Unix. O termo é usado hoje em dia para terminais de dados baseado em caracteres. Ao longo do capítulo, o termo será usado exclusivamente para se referir aos dispositivos do kernel.

O `Linux` distingue três classes de ttys: consoles (virtuais), pseudo terminais (parecidos com o conector de duas mãos, usado em aplicações como X11) e dispositivos seriais. Os últimos são contados também como tty, porque permitem sessões interativas sobre conexões seriais, sejam estas um terminal conectado fisicamente ou um computador remoto utilizando uma linha de telefone.

`ttys` possuem parâmetros de configuração que podem ser ativados através da chamada de sistema denominada `ioctl(2)`. Muitos desses referem-se somente a dispositivos seriais, visto que necessitam de uma maior flexibilidade para operarem vários tipos de conexões simultaneamente.

Entre o grande número de parâmetros de linha possíveis estão a paridade da linha e velocidade. Existem também indicadores para a configuração da conversão entre os caracteres maiúsculos e minúsculos, da tecla que comanda o avanço de linha, etc.. O dispositivo tty pode suportar também várias *linhas de parâmetros* que fazem o programa de controle de dispositivo comportar-se de forma totalmente diferente. Por exemplo, o programa SLIP para o `Linux` é implementado por meio de linhas de parâmetros especiais.

Existe um pouco de ambigüidade sobre a forma de medir a velocidade de uma conexão. O termo correto é *taxa de bits*, que está relacionado com a velocidade de transferência na linha medida em bits por segundo (ou bps na forma abreviada). Algumas vezes, é possível ouvir pessoas referindo-se a isto como a *taxa de transmissão*, o que não é totalmente correto. Estes dois termos, contudo não devem ser trocados. A taxa de transmissão refere-se à característica física de alguns dispositivos seriais, chamados de taxa de clock nos quais pulsos são transmitidos. A taxa de bits denota um estágio corrente de uma conexão serial existente entre dois pontos, para saber a média do número de bits transferidos por segundo. É importante salientar que estes dois valores geralmente são diferentes, já que a maioria dos dispositivos codificam mais que um bit por pulso elétrico.

4.3 Acessando Dispositivos Seriais

Como todos os dispositivos do sistema `Unix`, as portas seriais são acessadas através de arquivos de dispositivos especiais, localizados no diretório `/dev`. Existem duas variedades de arquivos de dispositivos relacionados a programas de controle de dispositivos seriais, e para cada porta existe um arquivo. Dependendo do arquivo que é acessado por ele, o dispositivo se comportará diferentemente.

A primeira variedade é usada sempre que a porta seja utilizada no recebimento de chamadas discadas; ela possui um número principal de 4, e os arquivos são chamados `ttyS0` `ttyS1`, etc.. A segunda variedade é usada quando a discagem é efetuada na máquina local para acesso externo através de uma porta. Os arquivos são chamados `cua0`, e possuem um número principal igual a 5.

Os números menores são idênticos para ambos os tipos. Caso o modem esteja em uma das portas que vão de `COM1` até `COM4`, seu número menor será o número da porta `COM` mais 63. Caso a configuração seja diferente destas, por exemplo ao se usar uma placa que suporte diversas linhas seriais, por favor consulte o Como Fazer - Serial.

Assumindo-se que o modem esteja na `COM2`, seu número menor será 65 e seu número principal será 5 para a execução de discagem de saída. Deve haver um dispositivo `cua1` que possua estes números. Para encontrá-lo deve-se listar os ttys seriais no diretório `/dev`. As colunas 5 e 6 devem mostrar os números principal (maior) e o menor, respectivamente:

```
$ ls -l /dev/cua*
crw-rw-rw-   1 root     root       5,  64 Nov 30 19:31 /dev/cua0
crw-rw-rw-   1 root     root       5,  65 Nov 30 22:08 /dev/cua1
crw-rw-rw-   1 root     root       5,  66 Oct 28 11:56 /dev/cua2
crw-rw-rw-   1 root     root       5,  67 Mar 19  1992 /dev/cua3
```

Se não existir tal dispositivo, você terá que criar um, utilizando o superusuário e digitando o seguinte:

```
# mknod -m 666 /dev/cua1 c 5 65
```

```
# chown root.root /dev/cua1
```

Algumas pessoas sugerem que seja feita do arquivo `/dev/modem` uma ligação simbólica para o arquivo de dispositivo de modem, de forma que usuários ocasionais não tenham que lembrar de algo não intuitivo como `cua1`. De qualquer modo, não se pode usar o nome `modem` em um programa e simultaneamente no nome real do arquivo de dispositivo. Isto porque estes programas usam os chamados *arquivos de reserva de recursos* para sinalizar que um dispositivo está em uso. Por convenção, o nome do arquivo de reserva de recursos para `cua1` é `LCK..cua1`. Usar arquivos de dispositivos diferentes para a mesma porta significa que o programa falhará ao reconhecer outros arquivos de reserva de recursos e usará ambos os dispositivos ao mesmo tempo. Como resultado, ambas as aplicações falharão.

4.4 Hardware Serial

O `Linux` suporta atualmente uma extensa variedade de placas seriais que usam o padrão RS-232. Atualmente RS-232 é o padrão mais comum para comunicações seriais para PC. Ele usa um número de circuitos para a transmissão de bits sozinhos assim como para o sincronismo das transmissões. Linhas adicionais podem ser usadas para sinalizar a presença de portadora e negociação da comunicação.

Embora a negociação da comunicação seja opcional, ela é muito útil. Permite que qualquer uma das duas estações possa sinalizar se está pronta para receber mais dados, ou se a outra estação deverá fazer uma pausa até que o processamento feito pelo receptor esteja concluído. As linhas usadas para isto são chamadas ”Livres para Enviar” (CTS) e ”Prontas para Enviar” (RTS), descrevendo o nome da negociação da comunicação por hardware chamada “RTS/CTS”.

Em PCs, a interface RS-232 é geralmente controlada por um chip UART derivado do chip semicondutor 16450, ou de uma versão mais nova: o NSC 16550A. [3]

Algumas marcas (muitos modems equipados internamente com o conjunto de chips Rockwell) também usam chips completamente diferentes, os quais foram programados para funcionarem como se fossem 16550’s.

A principal diferença entre os 16450s e os 16550s é que o último tem um buffer FIFO de 16 bytes, enquanto que o anterior tem um buffer de somente 1-Byte. Isto torna os 16450s convenientes para velocidades até 9600 bps, enquanto que

[3] Houve também o NSC 16550, mas a sua FIFO nunca funcionou realmente.

velocidades mais altas necessitam de um chip 16550 ou compatível. Ao lado destes chips, o `Linux` suporta também o chip 8250, o UART original para o PC-AT.

Na configuração padrão, o kernel checa as quatro portas seriais padrão, de `COM1` até `COM4`. Estas receberão números menores de dispositivos iguais a 64 até 67, conforme descrito anteriormente.

Caso se necessite configurar as portas seriais corretamente, deve-se instalar o comando `setserial` de Ted Tso junto com o programa `rc.serial`. Este programa deve ser chamado a partir do programa `/etc/rc` durante a inicialização do sistema. Ele usa o comando `setserial` para configurar os dispositivos seriais no kernel. Um programa típico `rc.serial` terá a seguinte aparência:

```
# /etc/rc.serial - programa de configuração da linha serial
#
# Executa detecção de interrupções
/sbin/setserial -W /dev/cua*

# Configura dispositivos seriais
/sbin/setserial /dev/cua0 auto_irq skip_test autoconfig
/sbin/setserial /dev/cua1 auto_irq skip_test autoconfig
/sbin/setserial /dev/cua2 auto_irq skip_test autoconfig
/sbin/setserial /dev/cua3 auto_irq skip_test autoconfig

# Apresenta a configuração dos dispositivos seriais
/sbin/setserial -bg /dev/cua*
```

Por favor consulte a documentação que acompanha o programa `setserial` para o detalhamento dos parâmetros.

Caso a porta serial não seja detectada, ou o comando `setserial -bg` mostre valores incorretos, será necessário forçar a configuração e explicitar os valores corretos. Os usuários com modems equipados com o conjunto de chips Rockwell são informados para analisar esta situação. Se, por exemplo, o chip UART é detectado como um NSC 16450, enquanto na verdade trata-se de um NSC 16550, sendo necessário alterar o comando de configuração:

```
/sbin/setserial /dev/cua1 auto_irq skip_test autoconfig uart 16550
```

Opções similares existem para forçar o valor da porta `COM`, do endereço base e da IRQ. Por favor consulte a página do manual do programa `setserial(8)` para maiores detalhes.

Caso o modem suporte a negociação através de hardware, deve-se estar seguro de que ele esteja habilitado. Por mais surpreendente que isto possa parecer, a maioria dos programas de comunicação não tenta habilitá-la automaticamente, havendo necessidade de ajustá-la manualmente. A melhor forma é através do programa de inicialização `rc.serial`, usando o comando `stty`:

```
$ stty crtscts < /dev/cua1
```

Para checar se a negociação de comunicação por hardware está de fato habilitada, deve-se utilizar:

```
$ stty -a < /dev/cua1
```

Este comando fornece a situação de todos os parâmetros para o dispositivo, onde um indicador precedido por um sinal de menos como em `-crtscts`, significa que ele não está ativo.

Capítulo 5

Configurando Redes TCP/IP

Neste capítulo, conheceremos todas as etapas necessárias para configurar os elementos de redes TCP/IP. Iniciando com as atribuições dos endereços IP, pausadamente caminharemos através da configuração das interfaces da rede TCP/IP e apresentaremos algumas ferramentas úteis nas soluções para problemas na instalação de redes.

A maior parte das tarefas incluídas neste capítulo será executada somente uma vez. A maioria dos arquivos de configuração somente será alterada posteriormente ao se adicionar novos protocolos, servidores, placas, etc. na sua rede, ou quando um sistema for reconfigurado inteiramente. Alguns dos comandos usados para configurar o TCP/IP, contudo, devem ser executados cada vez que o sistema é inicializado. Isto é geralmente feito invocando-se os programas do sistema denominados `/etc/rc`.

Comumente, os itens referentes à rede para este procedimento estão contidos em um programa chamado `rc.net` ou `rc.inet`. Algumas vezes, podem ser vistos dois outros chamados `rc.inet1` e `rc.inet2`, onde o primeiro inicializa a parte de rede do kernel, enquanto o último inicializa os serviços básicos e as aplicações da rede. Nos passos seguintes, iremos mostrar como estes arquivos são compostos.

A seguir, discutiremos as ações executadas por `rc.inet1`, enquanto que as aplicações serão discutidas em capítulos posteriores. Ao finalizar este capítulo, deve-se ter à disposição uma seqüência de comandos que configurem corretamente a rede TCP/IP em um computador. Deve-se então substituir os comandos de exemplos no arquivo `rc.inet1` pelos novos aqui descritos, certificar-se que `rc.inet1` é executado na hora da inicialização do sistema e reinicializar sua máquina. Os programas `rc` de rede que vêm com a sua distribuição favorita do `Linux`[1] devem propiciar um bom exemplo.

5.1 Configurando o Sistema de Arquivos `proc`

Algumas das ferramentas de configuração da versão da Net-2 baseiam-se no sistema de arquivos `proc` para se comunicarem com o kernel. Esta é uma interface que permite acessar as informações do kernel em tempo de execução, através de um mecanismo similar a um sistema de arquivos. Quando montado, é possível listar os arquivos disponíveis como em qualquer outro sistema de arquivos, ou ainda exibir seus conteúdos. Itens típicos do sistema de arquivos `proc` incluem por exemplo o arquivo `loadavg`, o qual contém a carga média do sistema, ou o arquivo `meminfo`, que mostra o núcleo de memória corrente e o uso da área de troca.

Os programas de rede adicionam o diretório `net`. Ele contém diversos arquivos que contêm informações como tabelas ARP do kernel, o estado das conexões TCP e as tabelas de roteamento. A maioria das ferramentas de administração de rede busca informações nestes arquivos.

O sistema de arquivos `proc` (ou `procfs` como é também conhecido) é geralmente montado no diretório `/proc` durante a inicialização do sistema. O melhor método de se fazer isso é acrescentar a seguinte linha ao arquivo `/etc/fstab`:

```
# ponto de montagem do sistema de arquivos proc:
none /proc proc defaults
```

e após executar o comando "`mount /proc`" a partir do programa `/etc/rc`.

O `procfs` é atualmente configurado automaticamente na maioria dos kernels. Se o `procfs` não estiver presente, será emitida uma mensagem no seguinte formato: "`mount: fs type procfs not supporated by kernel`".[2] Será necessário então recompilar o kernel e responder "yes" quando questionado pelo suporte do `procfs`.

[1] Como por exemplo o Conectiva Linux
[2] O sistema de arquivos do tipo proc não é suportado pelo kernel.

5.2 Instalando os Binários

Caso se esteja usando uma das distribuições do `Linux`, provavelmente ela disponibilizará a maioria das aplicações e utilitários de rede, assim como um conjunto coerente de arquivos de exemplo. O único caso onde será necessário obter e instalar novos utilitários, será quando for instalada uma nova versão do kernel. Como ocasionalmente elas envolvem mudanças na camada de rede do kernel, será preciso atualizar as ferramentas básicas de configuração. Isto envolve, no mínimo a recompilação, mas algumas vezes pode ser necessário obter um novo conjunto de binários. Estes são distribuídos geralmente junto com o kernel, contidos em um pacote chamado `net-XXX.tar.gz`, onde *XXX* é o número da versão. A distribuição relacionando ao `Linux` 1.0 é a 0.32b. A partir do kernel 1.1.27, o nome do arquivo foi alterado para `net-tools-XXX.tar.gz`, e o número de versão reflete o número da revisão do kernel ao qual eles se aplicam. `A versão atual, no momento da tradução deste guia, no Conectiva Linux é a net-tools-1.49.2cl.i386.rpm`.

Caso se deseje compilar e instalar as aplicações da rede TCP/IP padrão, é possível obter os fontes a partir de servidores de FTP `Linux`. Estas são versões consideravelmente alteradas de programas oriundo do Net-BSD ou de outras fontes. Outras aplicações, tais como `Xmosaic`, `xarchie`, ou o Gopher e o IRC devem ser obtidas separadamente. A maioria delas pode ser compilada tranqüilamente, ao se seguir as instruções. Elas podem ser obtidas também junto com as diversas distribuições Linux disponíveis.

O site FTP oficial para o Net-3 é denominado `sunacm.swan.ac.uk`, espelhado em `metalab.unc.edu` no caminho `system/Network/sunacm`. O conjunto de ferramentas Net-2 atualizado está disponível em `ftp.aris.com`. O código de rede de Matthias Urlichs derivado do BSD, pode ser obtido em `ftp.ira.uka.de` no caminho `/pub/system/linux/netbsd`.

5.3 Outro Exemplo

Para o restante deste livro, vamos introduzir um novo exemplo, menos complexo do que o da Universidade do Pantanal, e o qual pode estar mais próximo das tarefas que realmente encontramos em nosso dia a dia. Considere a Cervejaria Virtual, pequena companhia que fabrica, como o nome indica, Cerveja Virtual. Para administrar seu negócio mais eficientemente, o fabricante de cerveja virtual quer seus computadores conectados em rede, e que todos passem a ser PCs executando um novíssimo e brilhante `Linux` 2.x.

No mesmo piso, do outro lado da entrada, existe a Vinícola Virtual, que está muito próxima da fábrica de cerveja. Eles têm uma rede Ethernet própria. Naturalmente, as duas companhias querem conectar suas redes assim que elas estiverem operacionais. Como primeiro passo, é necessário configurar a máquina que servirá de caminho e repassará os datagramas entre as duas sub-redes. Mais tarde, será necessário ainda ter-se uma conexão UUCP com o mundo exterior, com o qual se trocarão mensagens e notícias. A longo prazo, será necessário também configurar uma conexão PPP para interligação com a Internet.

5.4 Configurando o Nome de Máquina

A maioria, se não todas as aplicações de rede, necessitam que o nome local da máquina tenha sido configurado para algum valor razoável. Isto é geralmente feito durante o procedimento de inicialização, executando-se o comando `hostname`. Para ajustar o nome da máquina para *`itaparica`*, por exemplo, deve-se executar o comando:

```
# hostname itaparica
```

É uma prática comum usar um nome de máquina não qualificado, sem nenhum nome de domínio adicional. Por exemplo, as máquinas da Cervejaria Virtual podem ser chamadas de `aracaju.cvirtual.com.br`, `jpessoa.cvirtual.com.br`, etc.. Estes são seus nomes de domínio oficiais e totalmente qualificados. Os nomes das máquinas locais correspondem na verdade ao primeiro componente do nome totalmente qualificado, como por exemplo `aracaju`. Contudo, como o nome local da máquina é freqüentemente usado para pesquisar o seu IP, tem-se que estar certo de que a biblioteca de resolução de nomes está capacitada para buscar os endereços IP das máquinas da rede. Isto geralmente significa que os nomes das máquinas devem ser informados no arquivo `/etc/hosts` (descrito mais adiante).

Algumas pessoas sugerem o uso do comando `domainname` para configurar o nome do domínio junto ao kernel, como elemento complementar do nome de máquina totalmente qualificado - FQDN. Desta forma é possível combinar a saída de `hostname` com `domainname` para obter-se o FQDN. Que esta seja a melhor forma de formar o FQDN é uma afirmativa parcialmente correta. O `domainname` é geralmente usado para configurar o nome do domínio NIS da máquina, que pode ser inteiramente diferente do domínio DNS, ao qual a máquina local pertence. O NIS será discutido no capítulo 10.

5.5 Definindo Endereços IP

Caso se esteja configurando o software de rede na máquina local para operações independentes ou seja autônomas e desvinculadas de uma rede (como por exemplo, ser capaz de rodar o software INN de notícias em rede), pode-se passar diretamente à outra seção com segurança, pois nestes casos será necessário somente o endereço IP para a interface de rede local, o qual é sempre `127.0.0.1`.

Os detalhes podem ser um pouco mais complexos com redes reais, como por exemplo uma Ethernet. Para conectar-se à uma rede existente é necessário solicitar aos seus administradores os endereços IP adequados e disponíveis para esta rede. Ao se configurar uma rede totalmente nova, os endereços IP deverão ser atribuídos conforme o descrito a seguir.

As máquinas dentro de uma rede local devem geralmente compartilhar endereços IP de uma mesma rede lógica. Para tanto deve-se definir uma faixa de endereços IP para a rede. Caso existam muitas redes físicas, deve-se atribuir a elas diferentes números, ou usar sub-redes para dividir seu endereço IP dentro de várias sub-redes.

Caso a rede não esteja conectada a Internet e não haja nenhuma previsão de que isto ocorra, pode-se escolher qualquer endereço de rede válido. Há que certificar-se somente que foi escolhida corretamente uma das classes A, B, ou C, caso contrário as coisas provavelmente não funcionarão muito corretamente. No entanto, caso haja alguma previsão de conexão a Internet em um futuro próximo, deve-se necessariamente obter um endereço IP oficial *imediatamente*. A melhor forma de fazer isso, é pedir ao provedor de acesso a Internet que lhe atenda para ajudá-lo, como por exemplo o seu provedor local ou a companhia de telecomunicações ou de telefonia local.

A Autoridade Internet de Definição de Numeração - IANA reservou diversos endereços que podem ser utilizados sem a necessidade de registro. Estes endereços devem ser utilizados somente em redes privadas e não podem ser roteados entre sites Internet[3]. As faixas são:

[3]N.T.: é fortemente recomendada a utilização de um dos endereços abaixo, independente de sua realidade em relação à conectividade com Internet.

Classe	Faixa de IPs
A	`10.0.0.0`
B	`172.16.0.0` até `172.31.0.0`
C	`192.168.0.0` até `192.168.255.0`

Note que o segundo e o terceiro blocos contêm 16 e 256 redes, respectivamente.

A escolha de um destes endereços de rede não é útil somente para redes totalmente desconectadas da Internet, pode-se ainda implementar maiores restrições e controles de acesso usando-se uma única máquina como caminho padrão de toda a rede. Para a rede local o caminho padrão é acessível através de endereços IP internos, enquanto o mundo exterior o reconhece através de um endereço oficialmente registrado.

No decorrer deste guia assumiremos que a rede da Cervejaria Virtual utilizará uma rede de classe B, com a numeração 191.72.0.0. Obviamente uma rede classe C seria mais que suficiente para atender às suas necessidades e às da rede da Vinícola Virtual. Utilizaremos esta classe somente para simplificar os exemplos de sub-redes na próxima seção.

Para operar várias redes Ethernets (ou outros tipos de redes, uma vez que o programa de controle de dispositivos esteja disponível), é necessário dividir a rede em pequenas redes menores: sub-redes. Note que a subdivisão de redes só é necessária se houver mais de uma *rede lógica*, onde conexões ponto a ponto com outras redes não contam, como por exemplo, em uma rede Ethernet com uma ou mais conexões PPP com o mundo exterior. Neste caso não é necessário subdividir sua rede. A razão disto será apresentada no capítulo 7.

No nosso exemplo, o administrador da rede da cervejaria adotou um número da rede classe B, igual a `191.72.0.0`. Para acomodar as duas redes, ele decide usar oito bits do endereço de máquina como bits adicionais da sub-rede e deixa outros oito bits para a própria máquina, permitindo 254 máquinas para cada uma das sub-redes. Ele então atribui o número de sub-rede 1 para a cervejaria e para a vinícola o número 2. Seus respectivos endereços de rede são `191.72.1.0` e `191.72.2.0`. A máscara da sub-rede é `255.255.255.0`.

A máquina `aracaju`, que é o caminho padrão entre as duas redes, recebe o número 1 em ambas, que é dado pelos endereços `191.72.1.1` e `191.72.2.1`, respectivamente. A figura 5.1 mostra as duas sub-redes e o caminho padrão.

Com o novo código de rede, as subdivisões não têm limitações nos bytes, então cada rede classe C pode ser dividida dentro de várias sub-redes. Por exemplo,

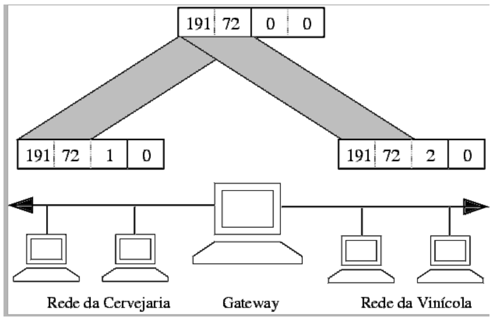

Figura 5.1: Cervejaria e Vinícola Virtuais – duas sub-redes

seria possível usar dois bits do endereço de máquina como máscara de rede, possibilitando 4 possíveis sub-redes com 64 máquinas em cada.[4]

5.6 Os Arquivos `hosts` e `networks`

Depois de se subdividir a rede, deve-se estar preparado para alguns tipos simples de resoluções de nomes de máquinas, através do uso do arquivo `/etc/hosts`. Caso não se utilize DNS ou NIS para resolução de endereços, deve-se então colocar todas as máquinas da rede no arquivo `hosts`.

Mesmo quando se utilizar DNS ou NIS, é indicado ter-se um conjunto com o nome de ao menos algumas máquinas em um arquivo `/etc/hosts`. Pode ser necessário, por exemplo algum tipo de resolução de nomes e endereços quando as interfaces de rede ainda não estejam ativas, por exemplo em tempo de inicialização do sistema. Isso não é somente uma questão de conveniência, pois também permite usar nomes simbólicos de máquinas nos programas `rc.inet`. Assim, caso algum endereço IP seja alterado, deve-se copiar um novo arquivo `hosts` atualizado para todas as máquinas e reiniciá-las, ao invés de editar-se um número grande de arquivos `rc` separadamente. Geralmente deve-se colocar todos os nomes locais das máquinas e seus respectivos endereços no arquivo `hosts`, incluindo quaisquer caminhos padrão e servidores NIS, caso estes sejam usados.[5]

[4] O último número em cada sub-rede é reservado como endereço de propagação para toda a rede, então na verdade são 63 máquinas por sub-rede.

[5] Os endereços de servidores NIS somente serão necessários, caso se utilize o NYS de Peter Eriksson. Outras implementações do NIS localizam seus servidores no momento de execução

Durante os testes iniciais deve-se estar seguro de que o resolvedor de nomes usa somente informações do arquivo `hosts`. Os softwares DNS ou NIS podem vir com arquivos de exemplo que podem produzir resultados estranhos ao serem usados. Para fazer com que todas as aplicações usem exclusivamente o arquivo `/etc/hosts` ao procurar o IP de uma máquina, é necessário editar o arquivo `/etc/host.conf`. Comente qualquer linha que inicie com a palavra `order` precedendo-as com um sinal de número (`#`) e insira a seguinte linha:

```
order hosts
```

A configuração da biblioteca de resolução será discutida em detalhes no capítulo 6.

O arquivo `hosts` contém uma entrada por linha, constituída de um endereço IP, um nome da máquina e uma lista opcional de nomes alternativos para o nome da máquina. Os campos são separados por espaços ou tabulações e o campo endereço deve iniciar necessariamente na coluna um. Qualquer coisa após um sinal `#` é considerado um comentário e será ignorado.

Nomes de servidores podem ser totalmente qualificados, ou relativos ao domínio local. Para a máquina `maceio`, normalmente seria informado o nome totalmente qualificado, igual a `maceio.cvritual.com.br`, ou simplesmente `maceio`, tornando-a conhecida tanto pelo seu nome oficial como pelo seu nome local mais curto.

Este é um exemplo de como o arquivo `hosts` da Cervejaria Virtual pode parecer. Dois nomes especiais são incluídos, o `aracaju-if1` e o `aracaju-if2`, que fornecem os endereços para ambas as interfaces usadas na máquina `aracaju`.

```
#
# Arquivo Hosts da Cervejaria Virtual e da Vinícola Virtual
#
# IP            nome local   nome de máquina totalmente qualificado
#
127.0.0.1       localhost
#
191.72.1.1      aracaju      aracaju.cvirtual.com.br
191.72.1.1      aracaju-if1
191.72.1.2      maceio       maceio.cvirtual.com.br
191.72.1.3      jpessoa      jpessoa.cvirtual.com.br
#
```

através do programa `ypbind`.

```
191.72.2.1      aracaju-if2
191.72.2.2      caxias        caxias.cvirtual.com.br
191.72.2.3      gramado       gramado.cvirtual.com.br
191.72.2.4      garibadi      garibaldi.cvirtual.com.br
```

Assim como em um endereço IP de uma máquina, algumas vezes pode ser necessário utilizar um nome simbólico também para as redes. Por isso, o arquivo `hosts` tem um companheiro chamado `/etc/networks` que mapeia nomes de redes para números e vice-versa. Na Cervejaria Virtual, podemos instalar um arquivo `networks` como este:[6]:

```
# /etc/networks para a Cervejaria Virtual
cerveja-net      191.72.1.0
vinho-net        191.72.2.0
```

5.7 Configuração de Interfaces

Depois de configurar o hardware como explicado no capítulo anterior, há que fazer com que esses dispositivos sejam conhecidos pela camada de rede do kernel. Muitos comandos são usados para configurar as interfaces da rede e inicializar a tabela de roteamento. Estas tarefas são executadas geralmente pelo programa `rc.inet1` toda vez que o sistema é inicializado. As ferramentas básicas para isto são chamadas `ifconfig`(onde "if" significa interface) e `route`.

O programa `ifconfig` é usado para construir uma interface que possa ser acessada pela camada de rede do kernel. Isto envolve a atribuição de um endereço IP e de outros parâmetros e a ativação da interface, também conhecida como "montagem". Ser ativo aqui significa que o kernel enviará e receberá datagramas IP através da interface. O modo mais simples para ativá-la é:

```
ifconfig interface endereço IP
```

o qual atribui o *endereço IP* para a *interface* e a ativa. Todos os outros parâmetros são ajustados para valores padrão. Por exemplo, a máscara da sub-rede padrão é derivada da classe do endereço IP da rede, tal como `255.255.0.0` para

[6]Note que os nomes em `networks` não devem coincidir com os nomes das máquinas no arquivo `hosts`, senão alguns programas podem produzir resultados estranhos.

um endereço de uma classe B. O programa `ifconfig` é descrito em detalhes no final deste capítulo.

O programa `route` permite adicionar ou remover caminhos da tabela de roteamento do kernel. Ele pode ser ativado da seguinte forma:

```
route [add|del] rede|endereço|máquina
```

onde os argumentos `add` e `del` determinam se o caminho para uma rede, máquina, endereço,... devem ser adicionados ou removidos, respectivamente.

5.7.1 A Interface Local de Rede

A primeira interface a ser ativada é a interface local de rede:[7]

```
# ifconfig lo 127.0.0.1
```

Ocasionalmente, será possível ver também o nome local da máquina denominado `localhost` sendo usado ao invés do endereço IP. O `ifconfig` buscará o nome no arquivo `hosts` onde uma entrada deve definir o nome da máquina para o endereço `127.0.0.1`:

```
# Exemplo de entrada de localhost no  arquivo /etc/hosts
localhost     127.0.0.1
```

Para visualizar a configuração de uma interface, execute o programa `ifconfig` fornecendo o nome da interface como argumento:

```
$ ifconfig lo
lo        Link encap Local Loopback
          inet addr 127.0.0.1  Bcast [NONE SET]  Mask 255.0.0.0
          UP BROADCAST LOOPBACK RUNNING  MTU 2000  Metric 1
          RX packets 0 errors 0 dropped 0 overrun 0
          TX packets 0 errors 0 dropped 0 overrun 0
```

Como se pode perceber, a interface local de rede (loopback) atribui uma máscara de rede `255.0.0.0`, uma vez que `127.0.0.1` é um endereço de classe A. Pode-se ver ainda que a interface não tem um conjunto de endereços de transmissão,

[7] Conhecida também como interface loopback.

que normalmente não são úteis para a interface local de rede. No entanto, ao se executar o servidor `rwhod` na máquina, é necessário ter o conjunto dos endereços de transmissão dos dispositivos de rede local em ordem para que o programa `rwho` funcione corretamente. Pode-se verificar como configurar endereços de propagação em rede na seção "Tudo sobre o `ifconfig`" a seguir.

Agora quase é possível iniciar a execução da sua pequena rede. O que ainda está faltando é uma entrada na tabela de roteamento que indique ao IP que ele pode usar esta interface como caminho para o destino `127.0.0.1`. Isto é obtido através do seguinte comando:

```
# route add 127.0.0.1
```

Novamente, pode-se usar `localhost` ao invés do endereço IP.

Agora, deve-se checar se tudo está funcionando, usando por exemplo o comando `ping`. `ping` é equivalente a um dispositivo de sonar[8] usado para verificar se um determinado endereço pode ser alcançado, e para medir o tempo decorrido entre o envio de um datagrama e o recebimento da resposta. O tempo necessário para isto é freqüentemente chamado de tempo de resposta.

```
# ping localhost
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=32 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=32 time=0 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=32 time=0 ms
^C

--- localhost ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0/0/1 ms
```

O exemplo acima mostra que a configuração da sua primeira interface de rede foi bem sucedida. Ao acionar o programa `ping` ele ficará ativo até que seja interrompido, como por exemplo através das teclas Control+C.

Se a saída do comando `ping` não se assemelha com a mostrada acima, então estamos com problemas. Verifique qualquer erro, como por exemplo se algum arquivo não foi instalado corretamente, verifique se os binários dos programas

[8] Alguém se lembra dos "Echoes" do Pink Floyd?

`ifconfig` e do `route` usados são compatíveis com a versão do kernel que possui e, acima de tudo, certifique-se de que o kernel foi compilado com a opção de rede habilitada (pode-se verificar isto através da presença do diretório `/proc/net`). Caso se obtenha uma mensagem de erro contendo a mensagem "Rede inatingível", então provavelmente o comando `route` foi utilizado de forma equivocada. Deve-se verificar se foi usado o mesmo endereço informado no `ifconfig`.

Os passos acima descritos são suficientes para que se possa utilizar as aplicações da rede em uma máquina isolada. Após adicionar as linhas acima ao `rc.inet1` e certificar-se que ambos os programas `rc.inet` são executados por `/etc/rc`, pode-se reinicializar a máquina e executar-se várias aplicações, como por exemplo, o programa "`telnet localhost`" deve estabelecer uma conexão `telnet` com a máquina local, fornecendo uma mensagem de acesso ao sistema.

No entanto, a interface de rede local não é útil somente em livros de rede, ou como um teste durante o desenvolvimento, mas é também usada por algumas aplicações durante sua operação normal.[9] De qualquer forma a sua configuração é obrigatória, independente da máquina estar conectada a uma rede ou não.

5.7.2 Interfaces Ethernet

A configuração de uma interface Ethernet é consideravelmente parecida com a de uma interface local de rede, ela necessita somente de alguns parâmetros adicionais, especialmente quando se utilizam sub-redes.

Na Cervejaria Virtual, dividimos a rede IP, que inicialmente foi uma rede classe B, e após, dentro dessa, criamos sub-redes classe C. Para que a interface reconheça esta situação, o comando `ifconfig` deverá ser similar a:

```
# ifconfig eth0 maceio netmask 255.255.255.0
```

Isto atribui à interface `eth0` o endereço IP da máquina `maceio`, igual a `191.72.1.2`. Caso tivéssemos omitido a máscara de rede, o programa `ifconfig` deduziria que a máscara da classe do IP da rede seria igual a `255.255.0.0`. Um rápido teste mostra:

```
# ifconfig eth0
```

[9]Por exemplo, todas as aplicações baseadas em RPC usam a interface local de rede para se registrarem junto ao servidor `portmapper` durante a inicialização.

```
eth0      Link encap 10Mps Ethernet HWaddr  00:00:C0:90:B3:42
          inet addr 191.72.1.2 Bcast 191.72.1.255 Mask 255.255.255.0
          UP BROADCAST RUNNING  MTU 1500  Metric 1
          RX packets 0 errors 0 dropped 0 overrun 0
          TX packets 0 errors 0 dropped 0 overrun 0
```

Pode-se ver que o `ifconfig` automaticamente ajusta o endereço de transmissão (o campo `Bcast` acima) para o valor usual, igual à parte de endereço de máquina com todos os bits configurados, e a mensagem da unidade de transferência - MTU - (o tamanho máximo de quadros Ethernet que o kernel irá gerar para esta interface) também ajustada para o valor máximo de 1500 bytes. Todos estes valores podem ser alterados através de opções especiais que serão descritas mais tarde.

Muito similar ao exemplo da interface de rede local, pode-se agora instalar uma entrada de roteamento que informe ao kernel sobre a rede que pode ser alcançada através da interface `eth0`. Para a Cervejaria Virtual, o programa `route` seria chamado da seguinte forma:

```
# route add -net 191.72.1.0
```

Primeiramente, isto parece um pouco de mágica, pois não fica realmente claro como o programa `route` detecta qual a rota para a interface. No entanto o truque é muito simples: o kernel testa todas as interfaces que foram configuradas anteriormente e compara o endereço de destino (no caso `191.72.1.0`) com o endereço da interface na rede (que atende à lógica booleana de "e" bit a bit entre o endereço da interface e o da máscara). A única interface que ele encontra é `eth0`.

Qual a finalidade da opção `-net`? Ela é usada neste caso, porque o programa `route` pode manipular tanto rotas para redes quanto rotas para servidores isolados (como foi visto acima com o `localhost`). Ao ser dado um endereço na notação do endereço IP, o programa tenta descobrir se o endereço é referente a uma rede ou a uma máquina em especial, analisando a parte do endereço reservado às máquinas. Se a parte do endereço referente à máquina é igual a zeros, o programa `route` assume que se trata de uma rede, caso contrário o toma como endereço de uma máquina. Porém o programa `route` pode concluir que host191.72.1.0 seja um endereço de servidor ao invés de um endereço de rede, pois ele não consegue descobrir se sub-redes estão sendo utilizadas. Portanto, há que explicitar que se trata de uma rede, através da opção `-net`.

É claro que o comando acima pode ser um pouco mais complexo, e está sujeito a erros de ortografia. Uma forma mais conveniente é usar os nomes de rede definidos no arquivo `/etc/networks`. Isto faz o comando muito mais legível, onde até mesmo o parâmetro `-net` pode ser omitido, pois agora o programa `route` sabe que o endereço `191.72.1.0` indica uma rede.

```
# route add cerveja-net
```

Agora que concluimos os passos básicos da configuração, queremos ter certeza de que a interface Ethernet está bem configurada. Deve se escolher uma máquina da rede Ethernet, por exemplo `jpessoa`, acessá-la no seu console e digitar:

```
# ping aracaju
PING aracaju: 64 byte packets
64 bytes from 191.72.1.1: icmp_seq=0. time=11. ms
64 bytes from 191.72.1.1: icmp_seq=1. time=7. ms
64 bytes from 191.72.1.1: icmp_seq=2. time=12. ms
64 bytes from 191.72.1.1: icmp_seq=3. time=3. ms
^C

----maceio.cvirtual.com.br PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 3/8/12
```

Caso não se veja uma saída parecida com esta, então obviamente algo está errado. Ao se encontrar uma taxa incomum de perda de blocos, isto pode ser um problema de hardware, como uma falha ou um problema físico, etc.. Caso não se receba nenhum bloco, deve-se checar a configuração da interface com o programa `netstat`. A estatística dos blocos mostrada pelo `ifconfig` informa se algum bloco foi enviado pela interface. Caso se tenha acesso a um servidor remoto, deve-se ir até aquela máquina e checar as estatísticas da interface também, pois deste modo, pode-se determinar exatamente os pacotes que foram perdidos. Além disso, é possível exibir ainda informações do roteamento através do programa `route` a fim de verificar se ambos os servidores possuem a entrada correta de roteamento. O programa `route` imprime a tabela de roteamento completa do kernel, quando chamado sem nenhum argumento (a opção `-n` apresenta endereços no formato numérico IP ao invés de usar o nome da máquina):

```
# route -n
Kernel routing table
Destination     Gateway         Genmask         Flags Metric Ref Use    Iface
127.0.0.1       *               255.255.255.255 UH    1      0    112      lo
191.72.1.0      *               255.255.255.0   U     1      0     10    eth0
```

O significado detalhado destes campos é explicado na seção Verificação Com o Comando `netstat`. A coluna `Flag` contém uma lista de sinais para cada interface, onde o `U` é indicado para interfaces ativas, e o `H` diz que o endereço de destino é uma máquina. Se o sinal `H` for configurado para uma rota que se acredite que seja uma rota de rede, há então que se especificar a opção `-net` com o comando `route`. Para testar se uma rota definida está sendo utilizada, deve-se verificar se o campo `Use`, o penúltimo, é incrementado entre duas execuções do programa `ping`.

5.7.3 Roteamento Através de um Caminho Padrão

Na seção anterior, explicamos como configurar uma máquina em uma única rede Ethernet, porém, muito freqüentemente, uma rede encontra-se conectada a outra por portas de entrada, que podem simplesmente ligar duas ou mais Ethernets, como podem também fornecer uma conexão com o outro lado do mundo. Para utilizar o serviço de uma destas portas, tem-se que fornecer informações adicionais de roteamento para a camada da rede.

Por exemplo, as Ethernets da Cervejaria e Vinícola Virtuais estão conectadas através de um caminho padrão, denominado `aracaju`. Assumindo que a máquina `aracaju` já foi configurada, temos somente que adicionar outra entrada na tabela de roteamento de `maceio`, a qual indica para o kernel que ele pode alcançar todas as máquinas da rede da Vinícola através de `aracaju`. A "magia" apropriada do `route` é mostrada abaixo: a palavra chave `gw` indica que o próximo argumento é um caminho padrão.

```
# route add vinho-net gw aracaju
```

É claro que qualquer máquina da rede da Vinícola com a qual se deseje falar deve ter uma entrada de roteamento correspondente para a rede da Cervejaria, caso contrário somente seria possível enviar dados de `aracaju` para a máquina `caxias` por exemplo, porém sem nenhuma resposta de retorno.

Este exemplo descreve somente um caminho padrão que troca pacotes entre duas Ethernets isoladas. Agora assumiremos que `aracaju` tem também uma conexão com a Internet (através de uma conexão PPP adicional). Pode-se então enviar datagramas de *qualquer* outro destino da rede para serem entregues a `aracaju`. Isto pode ser realizado tornando `aracaju` o caminho padrão:

```
# route add default gw aracaju
```

O nome de rede `default` é uma abreviatura para `0.0.0.0`, que significa a rota padrão. Não é necessário adicionar-se este nome ao arquivo `/etc/networks`, pois ele já está previamente construído no programa `route`.

Quando houver perdas elevadas de pacotes apresentadas pelo programa `ping` de uma máquina atrás de um ou mais caminhos padrão, isso pode sugerir uma rede muito congestionada. A perda de pacotes pode indicar uma carga excessiva temporária e não necessariamente deficiências técnicas. Estes congestionamentos podem ocasionar atrasos ou até mesmo abandono de datagramas recebidos.

5.7.4 Configurando um roteador

Configurar uma máquina para trocar pacotes entre duas Ethernets é algo muito objetivo. Assumindo-se que voltamos à máquina `aracaju`, que está equipada com duas placas Ethernet, cada uma sendo conectada a uma das duas redes. Tudo que se tem que fazer é configurar ambas as interfaces separadamente, dando a elas, seus respectivos endereços IP e isso é tudo.

É aconselhável acrescentar informaçoões das duas interfaces ao arquivo `hosts` da forma mostrada a seguir:

```
191.72.1.1      aracaju      aracaju.cvirtual.com.br
191.72.1.1      aracaju-if1
191.72.2.1      aracaju-if2
```

A seqüência de comandos para configurar as duas interfaces é:

```
# ifconfig eth0 aracaju-if1
# ifconfig eth1 aracaju-if2
# route add cerveja-net
# route add vinho-net
```

5.7.5 A interface PLIP

Ao se usar o protocolo PLIP para conectar duas máquinas, as coisas serão um pouco diferentes do que usar uma rede Ethernet. Elas são chamadas conexões *ponto a ponto*, porque envolvem somente dois servidores ("pontos"), em oposição às redes que têm várias máquinas comunicando-se entre si. Como um exemplo, consideramos o portátil de algum funcionário da Cervejaria Virtual que está conectado com a máquina `aracaju` via PLIP. O portátil é chamado de `itatiaia`, e tem somente uma porta paralela. Na hora da inicialização esta porta será registrada como `plip1`. Para ativar a conexão, é necessário configurar a interface `plip1` através dos seguintes comandos:[10].

```
# ifconfig plip1 itatiaia pointopoint aracaju
# route add default gw aracaju
```

O primeiro comando configura a interface, dizendo ao kernel que esta é uma conexão ponto a ponto, com a ponta remota possuindo o endereço de `aracaju`. O segundo instala a rota padrão, usando a máquina `aracaju` como caminho padrão. Na máquina `aracaju`, um comando parecido com o `ifconfig` será necessário para ativar a conexão (não é necessária a execução do programa `route`):

```
# ifconfig plip1 aracaju pointopoint itatiaia
```

Um ponto interessante é que a interface `plip1` na máquina `aracaju` não tem que ter um endereço IP separado, mas pode ter o endereço `191.72.1.1`.[11].

Agora resta configurar o roteamento do portátil para a rede da Cervejaria, faltando ainda uma forma de rotear a partir de qualquer uma das máquinas da Cervejaria até a máquina `itatiaia`. Um jeito particularmente trabalhoso de fazer isso é adicionar uma rota específica à tabela de roteamento de todas as máquinas definindo `aracaju` como o caminho padrão para `itatiaia` da seguinte forma:

```
# route add itatiaia gw aracaju
```

Uma opção mais adequada para se trabalhar com rotas temporárias é o uso de roteamento dinâmico. Um modo de se fazer isso é usar o programa `gated`, um servidor de roteamento, que pode ser instalado em cada máquina na rede para distribuir as informações de roteamento dinamicamente. O modo mais fácil, contudo, é usar o *proxy* ARP, onde a máquina `aracaju` responderá à qualquer consulta de ARP sobre a máquina `itatiaia` enviando o seu próprio endereço Ethernet. O

[10] Note que `pointopoint` não é uma abreviatura. Ele é realmente escrito desta forma.

[11] Apenas como precaução, deve-se configurar uma conexão PLIP ou SLIP ou PPP somente após se ter configurado totalmente as entradas da tabela de roteamento das Ethernets. Com alguns kernels mais antigos, a rota de rede pode apontar para o outro lado da conexão ponto-a-ponto.

resultado final é que todos os pacotes destinados a `itatiaia` serão enviados primeiro até `aracaju`, que então os reenvia para o portátil. Retornaremos a tratar do proxy ARP na seção Verificando as Tabelas ARP abaixo.

5.7.6 A Interface SLIP e PPP

Embora as conexões SLIP e PPP sejam usadas somente em ligações ponto a ponto como as conexões PLIP, existe muita coisa mais a ser dita sobre elas. Geralmente, estabelecer uma conexão SLIP envolve a discagem para um site remoto através de um modem e o ajuste da linha serial para o modo SLIP. O PPP é utilizado de forma similar. As ferramentas necessárias para se configurar uma conexão SLIP ou PPP serão descritas nos capítulos 7 e 8 respectivamente.

5.7.7 A Interface Fantasma

A interface fantasma é realmente um pouco exótica, contudo bastante útil. Sua principal aplicação se dá em máquinas isoladas ou naquelas onde a conexão IP de rede somente é realizada através de uma conexão discada. Na verdade, estas últimas também são máquinas isoladas na maior parte do tempo.

A questão com máquinas isoladas é que estas possuem um único dispositivo ativo de rede, o dispositivo local de rede, que é geralmente definido como endereço `127.0.0.1`. Em algumas ocasiões, no entanto, pode ser necessário enviar dados através do endereço IP "oficial" da máquina. Por exemplo, considerando o portátil `itatiaia`, que foi desconectado da rede neste exemplo e supondo-se que uma aplicação da máquina `itatiaia` queira agora enviar algum dado para uma aplicação no servidor. A pesquisa da máquina `itatiaia` no arquivo `/etc/hosts` resulta no endereço IP `191.72.1.65`, fazendo com que a aplicação tente enviar dados para este endereço. Como a interface local de rede está geralmente ativa somente ao nível da própria máquina, o kernel não tem idéia de que este endereço se refere na verdade a ELE mesmo! Como conseqüência o kernel ignora o datagrama e retorna um erro para a aplicação.

Este é o lugar onde o dispositivo fantasma atua. Ele resolve o dilema simplesmente servindo como "alter ego" da interface local de rede. No caso da máquina `itatiaia`, pode-se simplesmente indicar para a interface o endereço `191.72.1.65` e acrescentar um apontador de rota de máquina para ele. Todo datagrama para `191.72.1.65` será então entregue localmente. A forma correta de configurar esta situação é:

```
# ifconfig dummy itatiaia
# route add itatiaia
```

5.8 Tudo Sobre o `ifconfig`

Existem muito mais parâmetros para o `ifconfig` do que os anteriormente descritos. Seu formato usual é:

`ifconfig` *`interface`* `[[-net|-host]` *`endereço`* `[`*`parâmetros`*`]]`

A *`interface`* é o nome da interface e *`endereço`* é o endereço IP a ser atribuído para a interface. Este pode ser ou um endereço IP na notação numérica IP, ou um nome que o `ifconfig` pode encontrar nos arquivos `/etc/hosts` ou `/etc/networks`. As opções `-net` e `-host` forçam o `ifconfig` a tratar o endereço como uma rede ou um endereço da máquina, respectivamente.

Se o `ifconfig` é chamado somente pelo nome da interface, ele exibe a configuração daquela interface. Quando chamado sem nenhum parâmetro, ele mostra todas as interfaces configuradas e uma opção `-a` força-o a mostrar também as interfaces inativas. Um exemplo de chamada para a interface Ethernet `eth0` pode ter o seguinte formato:

```
# ifconfig eth0
eth0      Link encap 10Mbps Ethernet  HWaddr 00:00:C0:90:B3:42
          inet addr 191.72.1.2 Bcast 191.72.1.255 Mask 255.255.255.0
          UP BROADCAST RUNNING  MTU 1500  Metric 0
          RX packets 3136 errors 217 dropped 7 overrun 26
          TX packets 1752 errors 25 dropped 0 overrun 0
```

Os campos `MTU` e `Metric` mostram a unidade máxima de transferência e o valor métrico para a interface. O valor métrico é tradicionalmente usado por alguns sistemas operacionais para calcular o custo de uma rota. O `Linux` ainda não usa este valor, porém define-o para manter a compatibilidade com outros sistemas.

As linhas RX e TX mostram quantos pacotes foram recebidos ou transmitidos livres de erros, quantos erros ocorreram, quantos pacotes foram perdidos, provavelmente devido à falta de memória e quantos foram perdidos por falta de sincronismo no seu envio. Estas situações ocorrem geralmente quando pacotes chegam em uma velocidade maior do que a capacidade do kernel de atender à última interrupção. Os valores dos parâmetros utilizados pelo `ifconfig` correspondem mais ou menos aos nomes das opções na linha de comandos; eles serão explicados abaixo.

A seguir apresentamos uma lista de parâmetros identificados pelo `ifconfig` com detalhes sobre o uso de cada um. As opções que simplesmente ativam alguma funcionalidade, também permitem desligá-las através da adição de um hífen (-) precedendo o nome da opção.

`up` Define uma interface como ativa (up), ou seja, acessível para a camada do IP. Esta opção é completada quando também é fornecido um *`endereço`* na linha de comando e pode ser usada também para reativar uma interface que foi temporariamente desativada através da opção `down` (esta opção corresponde aos termos `UP` (ativo) e em execução apresentados na saída do comando).

`down` Define uma interface como inativa (down), isto é, inacessível ao nível do IP. Qualquer tráfego IP é efetivamente desabilitado através desta interface. Note que isto não elimina as entradas de roteamento que usam esta interface automaticamente. Para eliminar a interface permanentemente, devem ser retiradas todas as entradas de roteamento e fornecidas rotas alternativas se possível.

`netmask mask` Define uma máscara de sub-rede a ser usada pela interface. Ela pode ser fornecida como um número hexadecimal de 32 bits precedido de 0x, ou através de números decimais na notação do IP.

`pointopoint address` Esta opção é usada para conexões de IP ponto a ponto que envolvem somente duas máquinas, e é necessária para configurar, por exemplo, interfaces SLIP ou PLIP

(caso esta opção tenha sido ativada, a expressão `POINTOPOINT` será apresentada na saída do comando).

`broadcast address` Um endereço de propagação na rede é geralmente composto pela parte do endereço IP correspondente à rede com os bits da parte da máquina ajustados para o endereço destinado a todas as máquinas da rede. Algumas implementações do IP (em sistemas derivados do BSD) usam um esquema diferente onde a parte de endereço de máquina é totalmente igual a zeros. Esta opção está disponível para atender a ambientes pouco ortodoxos.

(caso esta opção tenha sido ativada, a expressão `BROADCAST` será apresentada na saída do comando).

`metric number` Esta opção pode ser usada para atribuir um valor métrico à entrada na tabela de roteamento criada para a interface. Esta métrica é usada pelo Protocolo de Informações de Roteamento (RIP) para construir tabelas de roteamento para a rede.[12]

A métrica padrão usada pelo `ifconfig` é igual a zero. Caso um servidor RIP não seja executado, esta opção não será necessária. Caso um servidor RIP esteja ativo, então raramente será necessário mudar o valor métrico.

`mtu bytes` Isto ajusta a Unidade Máxima de Transferência, que é o número máximo de octetos que a interface é capaz de manipular em uma transação. Para Ethernets, o padrão da MTU é de 1500, para as interfaces SLIP, o padrão é 296.

`arp` Esta é uma opção específica para propagação em redes, tais como protocolos Ethernets ou transmissões por rádio e ela habilita o uso do ARP, o Protocolo de Resolução de Endereços, para detectar os endereços físicos das máquinas ligadas a uma rede. Para propagação em redes esta opção é padrão.

(caso o ARP não esteja habilitado, o `ifconfig` exibirá a expressão `NOARP`).

`-arp` Desabilita o uso do ARP nesta interface.

`promisc` Coloca a interface no modo não padronizado. Em uma rede, faz com que a interface receba todos os pacotes, não importando se eles foram destinados a outra máquina ou não. Isto permite uma análise do tráfego da rede usando filtros de pacotes, também chamada *"Ethernet curiosa"*. Em geral esta é uma boa técnica de se encontrar problemas na rede.

Por outro lado, isto permite a qualquer um verificar o tráfego de senhas da rede e executar outras coisas desagradáveis. Uma proteção contra este tipo de ataque é não permitir que alguém possa simplesmente conectar o seu computador na sua Ethernet. Outra opção é usar os protocolos seguros de autenticação, como o Kerberos, ou o pacote de acesso do SRA.[13]

(esta opção corresponde ao `PROMISC` na saída do comando `ifconfig`).

`-promisc` Desliga o modo não padronizado.

[12] O RIP escolhe a rota otimizada para um certo servidor baseada no "tamanho" do caminho, que é calculado pela soma dos valores métricos individuais de cada conexão entre máquinas. Por padrão, um salto tem comprimento 1 (sair de uma máquina e chegar à máquina seguinte), mas pode ser qualquer valor positivo menor que 16. (Uma rota de comprimento 16 é igual ao infinito. Tais rotas são consideradas inutilizáveis). O parâmetro `metric` ajusta o custo deste salto, que é então transmitido pelo servidor de roteamento.

[13] O SRA pode ser obtido em `ftp.tamu.edu` no caminho `/pub/sec/TAMU`.

`allmulti` Os endereços multicast são alguns tipos de endereços de propagação na rede para um grupo de máquinas que não precisam ser necessariamente da mesma sub-rede. Esta opção ainda não é suportada pelo kernel.

(esta opção corresponde ao `ALLMULTI` na saída do comando `ifconfig`).

`-allmulti` Desliga os endereços multicast.

5.9 Verificação Com o Comando `netstat`

Abordaremos agora uma ferramenta muito útil para checar a configuração e a atividade de uma rede. Ele é o programa `netstat` e é na verdade um pouco de várias ferramentas. Discutiremos cada uma de suas funções nas seções seguintes.

5.9.1 Mostrando a Tabela de Roteamento

Quando o programa `netstat` é acionado com a opção `-r`, ele exibe a tabela de roteamento do kernel, da forma como fizemos com o programa `route` descrito acima. Na máquina `maceio`, ele produz o seguinte resultado:

```
# netstat -nr
Tabela de Roteamento IP do Kernel
Destino         Roteador        MáscaraGen.     Opções   MSS Janela  irtt Iface
192.168.255.0   0.0.0.0         255.255.255.0    U      1500 0          0 eth0
127.0.0.0       0.0.0.0         255.0.0.0        U      3584 0          0 lo
0.0.0.0         192.168.255.220 0.0.0.0          UG     1500 0          0 eth0
```

A opção `-n` faz com que o programa `netstat` imprima os endereços como números IP ao invés dos nomes da rede e simbólicos da máquina. Isto é útil especialmente quando se quer evitar pesquisas de endereço através da rede (como por exemplo em um servidor DNS ou NIS).

A segunda coluna da saída do programa `netstat` mostra onde o caminho padrão está configurado. Se o caminho padrão é usado, um asterisco é impresso. A coluna três mostra a "generalidade" da rota. Quando é dado um endereço IP para encontrar uma rota apropriada, o kernel examina todas as entradas da tabela de roteamento, fazendo um "e" booleano bit a bit do endereço e da máscara geradora comparando-o com o destino da rota.

A quarta coluna exibe vários indicadores que descrevem a rota:

G A rota usa um caminho padrão.

U A interface a ser usada está ativa.

H Somente uma única máquina pode ser alcançada através desta rota. Por exemplo, para uma entrada local de rede com endereço `127.0.0.1`.

D Configurado se a entrada da tabela foi gerada por uma mensagem de redirecionamento ICMP (ver a seção 2.5).

M Presente se a entrada da tabela foi modificada por uma mensagem de redirecionamento ICMP.

As outras colunas significam:

`MSS` Tamanho máximo de segmento padrão para conexões TCP através desta rota (em bytes).

`Janela` Tamanho da janela padrão para conexões TCP através desta rota.

`irtt` Tempo de ida e volta (RTT) inicial.

`Iface` Interface através da qual os pacotes IP serão enviados.

5.9.2 Mostrando as Estatísticas de Interface

Quando acionado com o parâmetro `-i`, o programa `netstat` exibirá as estatísticas de uso das interfaces de rede configuradas. Se, além disso, for informada a opção `-a`, ele imprimirá *todas* as interfaces presentes no kernel, e não só aquelas atualmente configuradas. Na máquina `maceio`, a saída do programa `netstat` terá a seguinte aparência:

```
$ netstat -i
Kernel Interface table
Iface   MTU Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flags
lo        0   0   3185      0      0      0   3185      0      0      0 BLRU
eth0   1500   0 972633     17     20    120 628711    217      0      0 BRU
```

Os campos MTU e Met mostram a MTU e o valor métrico correntes para aquela interface e as colunas RX e TX mostram quantos pacotes foram recebidos ou transmitidos: livres de erros (RX-OK/TX-OK), danificados (RX-ERR/TX-ERR), quantos foram abandonados (RX-DRP/TX-DRP) e quantos foram perdidos na transmissão (RX-OVR/TX-OVR).

A última coluna mostra os parâmetros que foram ajustados para esta interface, eles são versões compostas por um caractere dos nomes dos parâmetros longos que são impressos quando é exibida a configuração da interface com o comando `ifconfig`.

B Um endereço de propagação na rede foi definido.

L Esta interface é um dispositivo local de rede.

M Todos os pacotes são recebidos (modo não padronizado).

N Finais de arquivos são evitados.

O O protocolo ARP está inibido para esta interface.

P Esta é uma conexão ponto a ponto.

R A interface está em uso.

U A interface está ativa.

5.9.3 Mostrando Conexões

O programa `netstat` suporta um conjunto de opções para apresentação dos canais ativos ou passivos. As opções `-t`, `-u`, `-w`, e `-x` mostram os canais de conexão TCP, UDP, RAW ou UNIX que estão ativos. Ao se fornecer o sinal `-a`, os canais que estão esperando por uma conexão (isto é, ouvindo) também serão exibidos. Isso fornecerá uma lista de todos os servidores que estão sendo executados corretamente no sistema.

O comando `netstat -ta` na máquina `aracaju` produz o seguinte resultado:

```
$ netstat -ta
Conexões Internet Ativas (com os servidores)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto     Estado
tcp        0      0 *:6000                  *:*               OUVINDO
tcp        0      0 *:netbios-ssn           *:*               OUVINDO
tcp        0      0 *:www                   *:*               OUVINDO
tcp        0      0 aracaju:smtp     maceio:1040              ESTABELECIDA
tcp        0      0 *:smtp                  *:*               OUVINDO
tcp        0      0 localhost:1046   caxias:telnet            ESTABELECIDA
tcp        0      0 *:2049                  *:*               OUVINDO
```

```
tcp        0      0 *:635                   *:*                 OUVINDO
tcp        0      0 *:printer               *:*                 OUVINDO
tcp        0      0 *:ftp                   *:*                 OUVINDO
tcp        0      0 *:sunrpc                *:*                 OUVINDO
```

A saída mostra muitos dos servidores simplesmente esperando por uma conexão de entrada, contudo, a quarta linha mostra uma conexão de entrada SMTP oriunda da máquina `maceio` e a sexta linha informa que existe uma conexão `telnet` de saída para `caxias`.[14]

Usando-se o parâmetro `-a` serão mostrados todos os canais de todas as famílias.

5.10 Verificando as Tabelas ARP

Em algumas ocasiões será útil visualizar ou até mesmo alterar o conteúdo das tabelas ARP do kernel, por exemplo quando se suspeite que um endereço da Internet duplicado é a causa de algum problema intermitente na rede. A ferramenta denominada `arp` foi desenvolvida para tarefas como esta. Suas opções na linha de comando são:

```
arp [-v] [-t tipo_do_hardware] -a [nome_da_máquina]
arp [-v] [-t tipo_do_hardware] -s nome_da_máquina endereço_hardware
arp [-v] -d nome_da_máquina [nome_da_máquina...]
```

Todos os argumentos em *`nome_da_máquina`* podem ser definidos como nomes simbólicos da máquinas ou como endereços IP.

A primeira chamada exibe a entrada ARP para o endereço IP ou um servidor específico, ou todas as máquinas conhecidas caso o *`nome_da_máquina`* não seja fornecido. Por exemplo, acionando o programa `arp` para a máquina `aracaju` podemos ter algo como:

```
# arp -a
IP address      HW type                 HW address
191.72.1.3      10Mbps Ethernet         00:00:C0:5A:42:C1
```

[14] Você pode dizer se uma conexão é de saída ou não pelo número de portas envolvidas. O número da porta mostrada para o servidor de *chamada* será sempre um inteiro simples (no exemplo é igual a 1040), enquanto que nos casos em que a máquina local estiver sendo chamada, uma porta bem conhecida estará em uso, para a qual o `netstat` usará o nome simbólico encontrado no arquivo `/etc/services` (no caso ele é igual a telnet).

```
191.72.1.2      10Mbps Ethernet        00:00:C0:90:B3:42
191.72.2.4      10Mbps Ethernet        00:00:C0:04:69:AA
```

que mostra os endereços Ethernet das máquinas `aracaju`, `maceio` e `jpessoa`.

Usando a opção `-t` pode-se limitar a exibição em um tipo específico de hardware. Ele pode ser igual a `ether`, `ax25` ou `pronet`, os quais significam 10/100 Mbps Ethernet, AMPR AX.25 ou um equipamento token ring IEEE 802.5, respectivamente.

A opção `-s` é usada para adicionar permanentemente o endereço Ethernet do *`nome_da_máquina`* informado nas tabelas ARP e o argumento `endereço_ _hardware` especifica o endereço do hardware, que por padrão espera-se ser um endereço Ethernet, especificado como seis bytes hexadecimais separados por dois pontos. Pode-se ajustar ainda o endereço do hardware para outros tipos de dispositivos, usando-se a opção `-t`.

Uma situação na qual pode ser preciso adicionar manualmente um endereço IP à tabela ARP é quando, por alguma razão, as consultas ARP para servidores remotos falham, quando por exemplo o controlador ARP tem defeitos ou existe alguma máquina remota na rede que erroneamente identifica-se com um endereço IP do servidor. Manter manualmente endereços em uma tabela ARP é uma forma (drástica) de proteção contra máquinas que não estejam bem configuradas.

Acionando-se o programa `arp` com a opção `-d` faz com que todas as entradas ARP relativas à máquina informada sejam removidas. Isto pode ser usado para forçar a interface a tentar novamente obter o endereço Ethernet para o endereço IP em questão e é muito útil quando um sistema mal configurado tem uma informação ARP errada (é claro, há que se reconfigurar a máquina problemática antes).

A opção `-s` pode também ser usada para implementar o *proxy* ARP. Esta é uma técnica especial, onde uma máquina, chamada digamos `curumin`, age como um caminho padrão para outra máquina chamada `atlantida`, fingindo que ambos os endereços referem-se à uma mesma máquina, chamada `curumin`. Isso é possível através de uma entrada ARP para a máquina `atlantida` que aponta para sua própria interface Ethernet. Agora, quando uma máquina envia uma consulta ARP sobre a máquina `atlantida`, `curumin` retornará então uma resposta contendo o seu próprio endereço Ethernet. A máquina remetente da consulta enviará então todos os datagramas para `curumin`, que deverá reenviá-los para `atlantida`.

Estes estratagemas podem ser necessários, por exemplo, quando se quiser acessar `atlantida` a partir de uma máquina DOS com uma implementação TCP incompleta, que não compreenda o sistema de roteamento muito bem. Ao se usar o proxy ARP, ela parecerá para a máquina DOS como se estivesse em uma sub-rede local, não sendo necessário assim que ela conheça a rota através de `atlantida`, o caminho padrão.

Outra aplicação muito útil do proxy ARP se dá quando uma das máquinas de uma rede age temporariamente como um caminho padrão para alguma outra máquina, por exemplo, quando esta estiver conectada através de uma conexão discada. No exemplo da Cervejaria Virtual, já tínhamos encontrado o portátil `itatiaia` ligando-se à máquina `aracaju` através de uma conexão PLIP de tempos em tempos. É claro, isto funcionará somente se o endereço da máquina a qual se deseja fornecer o serviço proxy esteja na mesma sub-rede IP atuando como um caminho padrão. Por exemplo, a máquina `maceio` poderá atuar como o proxy ARP de qualquer máquina da sub-rede da Cervejaria (`191.72.1.0`), mas nunca para uma máquina da sub-rede da Vinícola (`191.72.2.0`).

A chamada correta para fornecer o proxy ARP para a máquina `atlantida` é apresentado a seguir. Obviamente o endereço Ethernet dado deve ser o da máquina `curumin`.

```
# arp -s atlantida 00:00:c0:a1:42:e0 pub
```

A entrada do proxy ARP pode ser removida através do comando:

```
# arp -d atlantida
```

5.11 O Futuro

As funcionalidades de rede `Linux` estão em contínuo desenvolvimento. As mudanças principais na camada do kernel trarão um esquema de configuração mais flexível que permitirá a configuração dos dispositivos da rede durante a execução. Por exemplo, o comando `ifconfig` irá usar argumentos que ajustam a linha IRQ e o canal DMA.

Outra mudança que logo estará disponível é o parâmetro adicional `mtu` para o comando `route`, que ajusta a Unidade Máxima de Transferência para uma rota em particular. Pode-se usar esta opção tipicamente para rotas entre um caminho padrão, onde a conexão entre o caminho padrão e o servidor de destino necessitam de uma MTU muito baixa. Por exemplo, assumindo-se que a máquina `natal` esteja conectada com a máquina `aracaju` através de uma conexão SLIP. Ao se enviar dados de `maceio` para `natal`, a camada da rede em `natal` usaria pacotes de 1500 bytes, pois eles são enviados através da Ethernet. A conexão SLIP, por outro lado, é operada com uma MTU de 296, deste modo a camada de rede em `aracaju` teria que quebrar os pacotes IP em pacotes menores de 296 bytes. Porém se fosse possível configurar uma rota em `maceio` para usar uma MTU de 296 desde o início, esta relativamente cara fragmentação poderia ser evitada da seguinte forma:

```
# route add natal gw aracaju mtu 296
```

Note que a opção `mtu` também permite seletivamente desfazer os efeitos da Política 'sub-redes são Locais' (SNARL). Esta política é uma opção de configuração do kernel e é descrita no capítulo 3.

Capítulo 6

Servidor de Nomes e Resolvedor de Endereços

Como foi discutido no capítulo 2, uma rede TCP/IP pode contar com esquemas diferentes para a conversão de nomes em endereços. A maneira mais simples, que tira vantagem dos intervalos de nomes divididos em zonas é uma tabela das máquinas armazenada em um arquivo denominado `/etc/hosts`. Isto é útil somente para pequenas LANs que são mantidas por um único administrador, e quando esta não possui tráfego IP com o mundo exterior. O formato do arquivo `hosts` já foi descrito no capítulo 5.

Alternativamente, pode-se usar o BIND – O Servidor de Nomes DNS de Berkeley – para realizar o mapeamento dos nomes dos servidores para endereços IP. Configurar o BIND pode ser uma tarefa complexa, mas uma vez feita, qualquer mudança na topologia da rede será realizada facilmente. No `Linux`, como em muitos outros sistemas `Unix`, o servidor de nomes é fornecido através de um programa chamado `named`. Na inicialização, ele carrega um conjunto de arquivos mestres dentro de seu cache e fica aguardando consultas de processos de usuários locais ou remotos. Existem diferentes jeitos de configurar o BIND, e nem todos exigem que se execute um servidor de nomes em todas as máquinas.

Este capítulo pode ser um pouco mais do que apenas um esboço de como operar um servidor de nomes. Caso se planeje usar o BIND em um ambiente maior do que uma pequena rede local e provavelmente uma conexão com a Internet, é sugerida uma leitura mais aprofundada sobre o tema, como por exemplo em um livro específico sobre BIND, como por exemplo o "DNS e BIND" de Cricket Liu (see [AlbitzLiu92]). Para uma informação atualizada, pode-se também checar as notas de distribuição nos fontes do BIND. Existe também um grupo de notícias dedicado a questões sobre DNS denominado `comp.protocols.tcp-ip.domains`. Outras informações podem ser encontradas na tradução do Como Fazer - DNS, incluído nesta publicação.

6.1 A Biblioteca Resolver

Quando falamos do "resolver", não nos referimos a nenhuma aplicação em especial, mas preferencialmente nos referimos à *biblioteca resolver*, uma coleção de funções que podem ser encontradas na biblioteca padrão do C. As rotinas principais são denominadas `gethostbyname(2)` e `gethostbyaddr(2)`, as quais buscam todos os endereços IP que pertencem a um servidor e vice-versa. Eles podem ser configurados para procurar a informação somente no arquivo `hosts`, o qual contém um conjunto de nomes de máquinas, ou usar a base de dados do NIS (Serviço de Informações em Rede) do arquivo `hosts`. Outras aplicações, como o `smail`, podem incluir controladores diferentes para qualquer uma destas funções e necessitam de configurações especiais.

6.1.1 O arquivo `host.conf`

O arquivo central que controla a configuração do resolvedor é denominado `host.conf`. Ele reside no caminho `/etc` e diz ao resolvedor quais serviços utilizar e em qual ordem.

As opções no arquivo `host.conf` devem ser definidas em linhas separadas. Os campos devem estar separados por espaço em branco (espaços ou tabs). O sinal (`#`) introduz um comentário.

As seguintes opções estão disponíveis:

`order` Determina a ordem em que os serviços são analisados. As opções válidas são `bind` para a consulta do nome no servidor, o arquivo `hosts` para pesquisas no arquivo `/etc/hosts` e `nis` para pesquisas no NIS. Alguns ou todos eles podem ser especificados e a ordem na qual eles aparecem na linha determina a ordem na qual os respectivos serviços serão analisados.

`multi` Ativa ou desativa através das palavras chaves `on` ou `off` a permissão de uma máquina do arquivo `/etc/hosts` ter vários endereços IP, conhecido como "endereços múltiplos". Este parâmetro não tem efeito para consultas ao DNS ou ao NIS.

`nospoof` Como explicado no capítulo anterior, o DNS permite encontrar o nome de uma máquina pertencente a um endereço IP usando o domínio definido em `in-addr.arpa`. As tentativas feitas pelos servidores de nome para suprir um nome falso de uma máquina são denominadas "*spoofing*". Para proteger-se desta tentativa de fraude, o resolvedor pode ser configurado para verificar se o endereço IP original está de fato associado com o nome de máquina obtido, caso não esteja, o nome é rejeitado e um erro é retornado. Este procedimento é acionado pelo parâmetro `nospoof on`.

`alert` Esta opção utiliza os parâmetros `on` ou `off` como argumentos. Caso ele esteja ativo (on), qualquer tentativa de "spoof" (veja acima) fará com que o resolvedor registre uma mensagem no histórico do sistema - `syslog`.

`trim` Esta opção recebe um nome de domínio como argumento, o qual será removido dos nomes de máquinas antes de uma busca. Isto é especialmente útil para `máquinas` onde se queira especificar somente os nomes, sem o domínio local. Uma busca por uma máquina com o nome do domínio local especificado terá o domínio removido, permitindo assim que a busca no arquivo `/etc/hosts` tenha sucesso.

As opções `trim` são cumulativas, tornando possível considerar uma máquina como sendo "local" para diversos domínios.

Um arquivo de exemplo para a máquina `aracaju` é mostrado a seguir:

```
# /etc/host.conf
# named está em execução, mas sem o  NIS
order   bind hosts
# permite múltiplos endereços
multi   on
# proteção contra tentativas de spoof
nospoof on
# elimina o domínio local (não é realmente necessário).
trim    cvirtual.com.br.
```

6.1.2 Variáveis de Ambiente do Resolvedor

As configurações no arquivo `host.conf` podem ser substituídas usando-se diversas variáveis de ambiente, como por exemplo:

`RESOLV_HOST_CONF` Especifica um arquivo a ser lido no local do arquivo `/etc/host.conf`.

`RESOLV_SERV_ORDER` Cancela a opção `order` informada no arquivo `host.conf`. Os serviços são informados como `hosts`, `bind` e `nis`, separados por espaço, vírgula, dois pontos, ou ponto e vírgula.

`RESOLV_SPOOF_CHECK` Determina as medidas tomadas contra spoofing. Ele é totalmente desabilitado por `off`. Os valores `warn` e `warn off` habilitam a checagem, acionando e desabilitando os registros em arquivo, respectivamente. Um valor `*` aciona a checagem, mas abandona facilidades de registro conforme definido em `host.conf`.

`RESOLV_MULTI` Um valor `on` ou `off` pode ser usado para cancelar as opções `multi` do arquivo `host.conf`.

`RESOLV_OVERRIDE_TRIM_DOMAINS` Esta variável de ambiente especifica uma lista de domínios que substituem aqueles informados no arquivo `host.conf`, na linha `trim`.

`RESOLV_ADD_TRIM_DOMAINS` Esta variável especifica uma lista de domínios que são adicionados àqueles informados no arquivo `host.conf`, na linha `trim`.

6.1.3 Pesquisas no Servidor de Nomes — `resolv.conf`

Ao configurar a biblioteca de resolução de nomes para usar o BIND para pesquisas de máquinas, deve-se informar ainda quais servidores devem ser usados. Existe um arquivo específico para isto, denominado `resolv.conf`. Se este arquivo não existir ou estiver vazio, o resolvedor assume que o servidor de nomes está na máquina local.

Ao se executar um servidor de nomes na máquina local, deve-se configurá-lo separadamente, como será explicado na seção seguinte. Se a máquina já está em uma rede local é sempre preferível usar um servidor de nomes já existente, caso seja possível.

A opção mais importante no `resolv.conf` é denominada `nameserver`, que fornece o endereço IP de um servidor de nomes para uso. Eles são consultados na ordem fornecida, caso sejam especificados vários servidores de nome com a opção `nameserver`, por isso, deve-se colocar o servidor mais confiável em primeiro lugar. Atualmente, até três servidores de nomes são suportados. Se a opção `nameserver` não for fornecida, o resolvedor tentará conectar-se ao servidor de nomes na máquina local.

Duas outras opções: `domain` e `search` permitem a utilização de atalhos para os nomes das máquinas no domínio local. Normalmente, ao se executar o comando `telnet` para outra máquina, não há necessidade de informar o nome totalmente qualificado, mas simplesmente um nome como *aracaju* ou *natal* na linha de comando, e o resolvedor anexará a este nome o sufixo *cvirtual.com.br*, por exemplo.

É exatamente isso que a opção `domain` faz. Permite especificar um nome de domínio padrão a ser anexado quando o DNS falhar na pesquisa de um nome de máquina. Por exemplo, ao se fornecer o nome *jacare* e o resolvedor falhar na busca de seu nome no DNS, porque não há domínio de primeiro nível, ao se informar *mat.pantanal.edu.br* como domínio padrão, ele repetirá a pesquisa por *jacare* agora com o domínio padrão anexado.

Isso parece ser bastante interessante, até o momento em que se tenha que contactar uma máquina localizada fora do Departamento de Matemática, pois estaremos de volta novamente com nomes totalmente qualificados. Obviamente seria desejável ter-se atalhos, como por exemplo *jaburu.fisica* para uma máquina de nome *jaburu* localizada no domínio *fisica.pantanal.edu.br*.

É neste momento que a opção `search` entra em ação. Uma lista a ser pesquisada pode ser fornecida usando a opção `search`, a qual é uma generalização do comando `domain`. Enquanto o último fornece um único domínio padrão, o segundo fornece uma relação destes, testados na ordem fornecida, até que um deles seja bem sucedido. A lista deve ser separada por brancos ou tabulações.

As opções `search` e `domain` são mutuamente exclusivas e não podem aparecer em conjunto na mesma configuração. Se nenhuma opção for informada, uma lista padrão de procura é construída a partir do nome de domínio local usando o nome de domínio da máquina local, mais todos os domínios paternos até o domínio raiz. O nome do domínio local pode ser dado através da opção `domain`. Caso ela não seja informada, o resolvedor irá obtê-la através da chamada do sistema `getdomainname(2)`.

É aconselhável limitar e conferir atentamente os domínios a serem pesquisados, pois uma definição mal feita pode gerar problemas colaterais estranhos no uso da rede.

Caso domínios padrões pareçam confusos, deve-se considerar o seguinte arquivo `resolv.conf` de exemplo para a Cervejaria Virtual:

```
# /etc/resolv.conf
# Nosso domínio
domain          cvirtual.com.br
#
# usaremos aracaju como o servidor de nomes central:
nameserver      191.72.1.1
```

ao encontrar o nome `maceio`, o resolvedor buscará a máquina `maceio`, e se este falhar, buscará a máquina `maceio.cvirtual.com.br`, `maceio.com.br` e `maceio.br`.

6.1.4 A Robustez do Resolvedor

Caso se esteja em uma rede local dentro de uma rede maior, definitivamente deve-se usar os servidores centrais de nomes, caso estejam disponíveis. A vantagem disto é que estes desenvolverão caches ricos em informações, já que todas as consultas passam por eles.

Este esquema no entanto, tem uma desvantagem: quando um incêndio recente destruiu o cabo da estrutura principal de nossa universidade, todas as estações da rede local ficaram inativas, pois o resolvedor não conseguia mais alcançar nenhum dos servidores de nome, não havia mais conexão com os terminais X, impressoras, etc..

Embora isto não seja comum para a estrutura principal de um campus, é aconselhável tomar determinadas precauções em relação a casos como estes.

Uma opção é configurar um servidor de nomes local o qual determina os nomes de máquinas do domínio local, e reenvia todas as consultas sobre máquinas externas à rede local para o servidor principal. Obviamente isto é aplicável somente se você está administrando seu próprio domínio.

Alternativamente, pode-se manter uma cópia de segurança da tabela de máquinas do servidor do domínio ou rede local no arquivo `/etc/hosts`. No arquivo `/etc/host.conf` é possível incluir então a opção "`order bind hosts`" para que a resolução de nomes seja efetuada pelo servidor de nomes local, caso o servidor principal apresente alguma falha.

6.2 Executando o `named`

O programa que fornece o serviço de domínio de nomes na maioria das máquinas `Unix` é geralmente chamado de `named` (pronunciado *name-di*). Este é um programa servidor desenvolvido originalmente para que sistemas BSD forneçam o serviço de nomes aos clientes e possivelmente para outros servidores de nomes. A versão corrente usada na maioria das instalações do `Linux` é denominada BIND-8-1-2-6cl.i386.rpm.

Esta seção requer alguma compreensão do modo como o Sistema de Domínio de Nomes funciona. Se a discussão parecer um pouco de grego com japonês, então releia o capítulo 2, no qual estão disponíveis mais informações básicas sobre o DNS.

O programa `named` é geralmente acionado na hora da inicialização do sistema e é executado até que a máquina seja desligada. Ele busca as informações em um arquivo de configuração chamado `/etc/named.boot` e em vários outros arquivos que contêm o mapeamento dos nomes de domínio para endereços. Os últimos são chamados de *arquivos de zona.* O formato e a semântica desses arquivos serão explicados na seção seguinte.

Para executar o `named` basta informar o seguinte comando:

```
# /usr/sbin/named
```

na linha de comando. O programa `named` é acionado, lê o arquivo `named.boot` e todos os arquivos de zona especificados. Ele grava a identificação do processo (`pid`) no arquivo `/var/run/named.pid` em formato ASCII, carrega todos os arquivos de zona primários, se necessário, e inicia a escuta na porta 53 para consultas DNS.[1]

[1] Existem vários binários do `named` em sites FTP `Linux`, cada um configurado diferentemente. Alguns possuem seu arquivo de pid em `/etc`, outros o armazenam em `/tmp` ou ainda em `/var/tmp`.

6.2.1 O arquivo `named.boot`

O arquivo `named.boot` é geralmente muito pequeno, não contendo nada além de alguns ponteiros para arquivos mestres contendo informações de zona e ponteiros para outros servidores de nome. Os comentários no arquivo de inicialização iniciam com um ponto e vírgula e se estendem até o final da linha. Antes de discutirmos o formato do arquivo `named.boot` com mais detalhes, daremos uma olhada no arquivo exemplo para a máquina `aracaju` apresentado a seguir.[2]

```
O arquivo named.boot para a máquina aracaju:

;
; arquivo /etc/named.boot para a máquina aracaju.cvirtual.com.br
;
directory     /var/named
;
;             domínio                  arquivo
;---------------------------------------------------
cache         .                        named.ca
primary       cvirtual.com.br          named.hosts
primary       0.0.127.in-addr.arpa     named.local
primary       72.191.in-addr.arpa      named.rev
```

Os comandos `cache` e `primary` mostrados neste exemplo carregam informação para o `named`. Esta informação é tomada dos arquivos mestres especificados no segundo argumento. Eles contêm representações textuais dos registros de recursos DNS, que serão vistas a seguir.

Neste exemplo, configuramos o `named` como o servidor de nomes primário para três domínios, como indicado pelas declarações `primary` no fim do arquivo. A primeira destas linhas, por exemplo, instrui o `named` para agir como um servidor primário para o domínio `cvirtual.com.br`, buscando os dados da zona no arquivo `named.hosts`. O parâmetro `directory` indica que todos os arquivos de zona estão localizadas em `/var/named`.

O parâmetro `cache` é muito especial e deve estar presente em praticamente todas as máquinas que executam um servidor de nomes. Sua função é dupla: ele instrui o `named` para habilitar seu cache e carrega o *acertos do servidor de nomes raiz* a

[2]Note que os nomes de domínio neste exemplo são dados *sem* o ponto ao final. Versões anteriores do `named` parecem tratar estes pontos como um erro no `named.boot` e silenciosamente descartam a linha.

partir do arquivo de cache especificado (`named.ca` no nosso exemplo). Voltaremos aos acertos do servidor de nomes a seguir.

Aqui está uma lista das opções mais importantes que podem ser usadas no arquivo `named.boot`:

`directory` Especifica um diretório onde os arquivos de zona residem. Os nomes de arquivos podem ser especificados relativos a este diretório. Vários diretórios podem ser especificados repetidamente usando-se o parâmetro `directory`. De acordo com o sistema de arquivos padrão do `Linux`, este deve ser igual a `/var/named`.

`primary` Recebe um *`nome de domínio`* e um *`nome de arquivo`* como argumentos, declarando a autoridade do servidor local para o domínio nomeado. Como um servidor primário, o `named` carrega a informação de zona do arquivo mestre dado.

Geralmente, sempre existirá pelo menos uma entrada `primary` em cada arquivo de inicialização, nomeado para o mapeamento reverso da rede `127.0.0.0`, a qual é a interface local de rede.

`secondary` Esta declaração recebe um *`nome de domínio`*, uma **lista de endereços** e um *`nome de arquivo`* como argumentos. Ele declara o servidor local como um servidor mestre secundário para o domínio especificado.

Um servidor secundário também detém dados autoritativos sobre o domínio, mas não mantém os arquivos localmente, e sim tenta transmití-los a partir do servidor primário. O endereço IP de pelo menos um servidor primário deve ser informado para o programa `named` na lista de endereços. O servidor local contactará cada um deles até que se consiga transferir com sucesso a base de dados da zona, a qual então será armazenada no arquivo fornecido como terceiro argumento. Se nenhum dos servidores primários responder, os dados de zona serão recuperados a partir deste arquivo.

O `named` tentará então atualizar os dados de zona em intervalos regulares. Isto é explicado abaixo durante a conexão com os registros de recursos do tipo SOA.

`cache` Recebe um *`domínio`* e um *`nome de arquivo`* como argumentos. Este arquivo contém a lista de acertos do servidor raiz, que é uma lista de registros que apontam para os servidores de nome raiz. Somente os registros NS e A serão reconhecidos. O argumento *`domínio`* é geralmente o nome do domínio raiz, um “`.`”.

Esta informação é absolutamente crucial para o programa `named`: caso o parâmetro `cache` não esteja presente no arquivo de inicialização, o programa `named` não criará uma memória cache local. Isso degradará seriamente a performance do sistema e aumentará a carga na rede, toda vez que o endereço a ser consultado não esteja presente na rede local. Mais ainda, o programa `named` não será capaz de resolver qualquer endereço, exceto aqueles sobre os quais ele tenha autoridade. Uma exceção a esta regra é uso de servidores retransmissores (vide a opção `forwarders` abaixo).

`forwarders` Esta opção recebe uma *`lista de endereços`* como argumento. Os endereços IP na lista especificam uma lista de servidores de nome, que o programa `named` pode consultar, caso ele falhe em resolver uma consulta ao seu cache local. Eles são testados em ordem até que um deles responda à consulta.

`slave` Esta declaração transforma o servidor de nomes em um servidor *escravo*, isto é, ele nunca executará consultas recursivas por si só, mas somente as reenvia para servidores especificados através da declaração `forwarders`.

Existem algumas opções que são raramente usadas e não serão descritas aqui, `sortliste` `domain`. Adicionalmente existem duas diretivas que podem ser usadas dentro destas bases de dados, que são `$INCLUDE` e `$ORIGIN`.

6.2.2 Os Arquivos da Base de Dados do DNS

Os arquivos mestre incluídos pelo programa `named`, como por exemplo o arquivo `named.hosts`, sempre possuem um domínio associado a eles, o qual é chamado de *origin*. Este é o nome de domínio especificado pelos comandos `cache` e `primary`. Dentro de um arquivo mestre, é permitido especificar o domínio e os nomes das máquinas relativas a esse domínio. Um nome dado em um arquivo de configuração é considerado *absoluto*, caso ele finalize com um único ponto, caso contrário ele é considerado relativo à origem, a qual pode ser referenciada através do símbolo "`@`".

Todos os dados contidos no arquivo mestre são divididos em *registros de recursos*, ou abreviadamente RR. Eles compõem a menor unidade de informação disponível no DNS. Cada registro de recurso possui um tipo. Um registro, por exemplo, de tipo A mapeia uma máquina para um endereço IP e um registro CNAME associa um apelido de uma máquina com o seu nome oficial.

As representações dos registros de recursos nos arquivos mestres compartilham um formato comum, o qual é igual a:

```
[domínio] [ttl] [classe] tipo dados
```

Os campos são separados por espaços ou tabulações. Uma entrada pode utilizar várias linhas caso estejam entre um abre parêntese na primeira linha e um fecha parêntese na última linha. Qualquer coisa entre um ponto e vírgula e uma linha nova é ignorada.

`domínio` Este é o nome do domínio ao qual as informações se aplicam. Caso nenhum domínio seja informado, assume-se que o RR aplica-se ao domínio do último RR informado.

`ttl` A fim de forçar os resolvedores a descartar informações após um certo tempo, a cada RR é associado um "*tempo de vida*", ou abreviadamente *ttl*. O campo *`ttl`* especifica o tempo em segundos, nos quais a informação será válida após a sua recuperação do servidor. Deve ser informado em formato decimal, com no máximo oito dígitos.

Caso nenhum ttl seja informado, é assumido o valor do campo ***`minimum`*** que precede o registro SOA.

`classe` Indica a classe do endereço, IN para endereços IP, HS para objetos da classe Hesiod. Caso nenhuma classe seja informada, a classe do último registro RR será utilizada.

`tipo` Descreve o tipo do registro RR. Os tipos mais comuns são A, SOA, PTR e NS. As seções seguintes descrevem os vários tipos de RRs.

`dados` Contém os dados associados ao registro RR. O formato deste campo depende do tipo do registro RR, os quais serão descritos separadamente.

A lista a seguir é uma descrição parcial dos RR usados nos arquivos master DNS. Há alguns outros não descritos aqui por estarem em fase experimental ou pela pouca utilização.

SOA Descreve a zona de autoridade (SOA significa "Início da Autoridade"[3]). Indica que os registros que sucedem o RR SOA contêm informações autoritativas

[3]Start of Authority

sobre o domínio. Cada arquivo mestre incluído pelo parâmetro `primary` deve conter um registro SOA para esta zona. Os dados contidos neste RR são os seguintes:

`origem` O nome canônico da máquina do servidor de nomes primário deste domínio. Normalmente é informado como um nome absoluto.

`contato` O endereço de email da pessoa responsável pela manutenção do domínio, com o caractere '`@`' substituído por um ponto. Por exemplo, caso a pessoa responsável pela Cervejaria Virtual tenha o nome de `janete`, o campo conterá o seguinte formato: `janete.cvirtual.com.br.`

`serial` É o número de versão do arquivo de zona, expressado como um número decimal simples. Toda vez que o arquivo de zona for alterado, este número deverá ser incrementado.

O número serial é usado pelos servidores secundários de nomes para detectarem se as informações sobre a zona foram alteradas. Para estar atualizado, o servidor secundário solicita o registro SOA ao servidor primário a intervalos regulares, e compara o número serial com o registro SOA recebido anteriormente. Caso o número tenha sido alterado, o servidor secundário transfere todo o banco de dados a partir do servidor primário.

`atualização` Especifica o intervalo mínimo em segundos que o servidor secundário deve esperar entre as verificações dos registros SOA no servidor primário. Novamente trata-se de um número decimal com no máximo oito dígitos.

Geralmente a topologia de rede não é alterada com freqüência, então este número deve indicar no mínimo um dia para grandes redes e ainda mais para redes menores.

`tentativas` Define o intervalo em segundos, no qual o servidor secundário deve tentar contato com o servidor primário, caso uma solicitação de atualização de zona tenha falhado. Não deve ser muito pequeno para que um problema temporário no servidor ou uma falha de rede possa ocasionar o uso exagerado dos recursos de redes. Um hora, talvez meia hora, podem ser boas opções.

`expiração` Especifica o tempo em segundos após o qual o servidor deve descartar todas as informações de zona, caso um contato com o servidor primário para a renovação das informações não tenha sido possível. É normalmente bastante dilatado. Craig Hunt ([Hunt92]) recomenda algo como 42 dias.

`mínimo` Este é o valor de validade padrão (ttl) dos registros de recursos que não tenham um especificado individualmente. Ele especifica o tempo máximo que outros servidores de nomes podem manter as informações de RR em seu cache. Isso aplica-se somente a pesquisas normais, e não tem relação com o tempo que um servidor secundário aguarda até tentar atualizar as informações sobre a zona do domínio.

`mínimo` deve ser um valor dilatado, especialmente para redes locais onde a topologia quase nunca muda. Um valor em torno de uma semana ou um mês é provavelmente uma boa opção. Nos casos onde estes RR possam mudar mais freqüentemente, pode-se definir diferentes ttl.

A Associa um endereço IP com um nome de máquina. O campo dados de recurso contém um endereço no formato decimal do endereço IP.

Cada máquina deve possuir somente um registro de tipo A. O nome de máquina usado neste registro é considerado o oficial ou nome de máquina *canônico*.

Todos os demais nomes de máquinas são apelidos e devem ser mapeados para o nome canônico da máquina através do registro CNAME.

NS Indica o servidor mestre de nomes de uma zona subordinada. Para uma melhor explicação sobre os registros NS, veja a seção 2.6. Os campos de dados contêm o nome da máquina do servidor de nomes. Para resolver um nome de máquina, um registro adicional de tipo A é necessário, o denominado *registro cola* o qual fornece o endereço IP do servidor de nomes.

CNAME Associa um apelido para uma máquina com o seu *nome de máquina canônico*. O nome canônico é fornecido pelo registro de tipo A, enquanto que os apelidos são somente ligações para ele, mas não possuem quaisquer outros tipos de registros associados presentes.

PTR Este tipo de registro é usado para associar nomes no domínio `in-addr.arpa` com os nomes de máquinas. É usado para mapeamento reverso de endereços IP para nomes de máquinas. Note que o nome de máquina informado deve ser igual ao nome canônico da máquina.

MX Este Registro de Recursos - RR indica o *servidor de correio eletrônico* para o domínio. As razões para se ter um servidor de correio são discutidas na seção Roteamento de Mensagens Na Internet no capítulo 13. A sintaxe de um registro MX é

```
[domínio] [ttl] [classe] MX preferência máquina
```

máquina define o servidor de correio para o *domínio*. Todo servidor de correio tem um número inteiro definindo a *preferência* associada a ele. Um agente de transporte de mensagens que deseje entregar uma mensagem de correio para o *domínio* tentará todas as máquinas que têm um registro MX para o domínio, até que seja bem sucedido. Aquele que tiver o menor valor no campo preferência será utilizado em primeiro lugar, seguindo-se os demais em ordem crescente até que a mensagem possa ser entregue.

HINFO Este registro provê informações sobre o software e o hardware do sistema. Sua sintaxe é:

```
[domínio] [ttl] [classe] HINFO hardware software
```

O campo *hardware* identifica o hardware utilizado pela máquina. Há convenções especiais para especificá-lo. Uma lista dos nomes válidos é fornecida pela RFC 1340 - Definindo Números. Caso o campo contenha somente espaços em branco, deverá estar entre aspas. O campo *software* contém o nome do sistema operacional usado pelo sistema. Novamente, um nome válido da RFC 1340 deve ser utilizado.

6.2.3 Criando Arquivos Master

As descrições a seguir fornecem arquivos de exemplos para o servidor de nomes da Cervejaria Virtual, localizados na máquina `aracaju`. Devido ao pequeno tamanho da nossa rede de exemplo, os arquivos tornam-se muito objetivos. No caso de necessidades mais complexas e caso o programa `named` não consiga ser executado, considere a leitura do livro "DNS and BIND" de Cricket Liu e Paul Albitz ([AlbitzLiu92]).

O arquivo de cache `named.ca` mostra exemplos de registros localizados em um servidor de nomes raiz. Um típico arquivo de cache normalmente descreve uma dúzia de servidores de nomes, ou mais. Pode-se obter a lista atual de nomes de servidores para um domínio raiz através do comando `nslookup` descrito no final deste capítulo.[4]

[4] Note que não é possível pesquisar no servidor de nomes local, um outro servidor de domínio raiz indicado. Para resolver este dilema, pode-se ou indicar para o comando `nslookup` um nome diferente de servidor ou utilizar um arquivo simples (por exemplo, `named.cache`) como ponto de partida e então obter uma lista completa de servidores válidos.

O arquivo named.ca.

```
;
; /var/named/named.ca
; Arquivo de cache da cervejaria Virtual.
; Não estamos na Internet, logo não necessitaremos
; de servidores de domínios raiz. Para ativar estes
; registros, basta remover os ponto e vírgula.
;
; .                99999999   IN    NS  NS.NIC.DDN.MIL
; NS.NIC.DDN.MIL   99999999   IN    A   26.3.0.103
; .                99999999   IN    NS  NS.NASA.GOV
; NS.NASA.GOV      99999999   IN    A   128.102.16.10
```

O arquivo named.hosts.

```
;
; /var/named/named.hosts     Máquinas locais da Cervejaria Virtual
;                            Origem é cvirtual.com.br
;
@                 IN  SOA   aracaju.cvirtual.com.br. janete.cvirtual.com.br.
                        (   16         ; número da versão do arquivo
                            86400      ; atualização: uma vez ao dia
                            3600       ; tentativas:   uma hora
                            3600000    ; expiração:  42 dias
                            604800     ; mínimo: 1 semana
                            )
                  IN  NS    aracaju.cvirtual.com.br.
;
; mensagens são distribuídas localmente pela máquina aracaju
                  IN  MX    10 aracaju

; endereço da interface de rede local
localhost.        IN  A     127.0.0.1
; rede Ethernet da Cervejaria Virtual
aracaju           IN  A     191.72.1.1
aracaju-if1       IN  CNAME aracaju
; aracaju é também um servidor de notícias
news              IN  CNAME aracaju
maceio            IN  A     191.72.1.2
jpessoa           IN  A     191.72.1.3
; rede Ethernet da Vinícola Virtual
aracaju-if2       IN  A     191.72.2.1
caxias            IN  A     191.72.2.2
```

```
gramado           IN  A     191.72.2.3
garibaldi         IN  A     191.72.2.4
```

```
O arquivo named.local.

;
; /var/named/named.local       Mapeamento reverso de  127.0.0
;                              Origin é 0.0.127.in-addr.arpa.
;
@                 IN SOA aracaju.cvirtual.com.br. janete.cvirtual.com.br.
                        (   2         ; número da versão do arquivo
                            360000    ; atualização: a cada 100 horas
                            3600      ; tentativas:   uma hora
                            3600000   ; expiração:  42 dias
                            360000    ; mínimo: 100 horas
                            )
                  IN NS  aracaju.cvirtual.com.br.
1                  IN  PTR   localhost.
```

```
O arquivo named.rev.

;
; /var/named/named.rev         mapeamento reverso dos endereços IP
;                              Origin é 72.191.in-addr.arpa.
;
@                 IN SOA aracaju.cvirtual.com.br. janete.cvirtual.com.br.
                        (   16        ; número da versão do arquivo
                            86400      ; atualização: uma vez ao dia
                            3600       ; tentativas:   uma hora
                            3600000    ; expiração:  42 dias
                            604800     ; mínimo: 1 semana
                            )
                  IN NS  aracaju.cvirtual.com.br.
; Cervejaria Virtual
1.1               IN PTR aracaju.cvirtual.com.br.
2.1               IN PTR maceio.cvirtual.com.br.
3.1               IN PTR jpessoa.cvirtual.com.br.
; Vinícola
1.2               IN PTR aracaju-if1.cvirtual.com.br.
2.2               IN PTR caxias.cvirtual.com.br.
3.2               IN PTR gramado.cvirtual.com.br.
4.2               IN PTR garibaldi.cvirtual.com.br.
```

6.2.4 Verificando a Configuração do Servidor de Nome

Há uma excelente ferramenta para testar o funcionamento de um servidor de nomes e a sua configuração denominada `nslookup` e pode ser usada tanto interativamente como através da linha de comando. Neste caso pode ser acionada digitando-se:

```
$ nslookup nome_da_máquina
```

o qual irá acionar o servidor de nomes especificado no arquivo `resolv.conf` para *`nome_da_máquina`* (caso haja mais de um servidor de nomes especificado, o programa `nslookup` escolherá um aleatoriamente).

O modo interativo porém é muito mais interessante, apesar de pesquisar somente máquinas individualmente, podendo ser solicitado qualquer tipo de registro DNS, assim como a transferência de todas as informações de zona de um domínio.

Quando acionado sem argumentos, o programa `nslookup` irá listar o nome e o endereço IP do servidor utilizado e iniciar o modo interativo. Na linha de comandos indicada pelo caractere '>' pode-se digitar qualquer nome de domínio que se deseje pesquisar. Por padrão, as pesquisas são baseadas em registros tipo A, os quais contêm o endereço IP relativo ao nome do domínio.

Pode-se alterar o tipo de registro através do comando "`set type=`*tipo*", onde *`tipo`* é um dos nomes de registros de recursos descritos na seção 6.2, ou ANY (qualquer).

Por exemplo, pode-se ter o seguinte diálogo com o comando:

```
$ nslookup
Default Server:  aracaju.cvirtual.com.br
Address:  172.16.1.1

> metalab.unc.edu
Server:  aracaju.cvirtual.com.br
Address:  172.16.1.1

Non-authoritative answer:
Name:    metalab.unc.edu
Address:  152.19.254.81
```

Caso se tente uma pesquisa por um nome do qual não se tenha nenhum endereço IP associado, mas outros registros forem encontrados na base de dados DNS, o `nslookup` retornará uma mensagem de erro dizendo "`Nenhum registro tipo A foi encontrado`". De qualquer forma, é possível pesquisar por registros diferentes do tipo A, através do comando "`set type`". Por exemplo para obter o registro SOA de `unc.edu`, pode-se utilizar:

```
> unc.edu
*** No address (A) records available for unc.edu
Server:  aracaju.cvirtual.com.br
Address:  172.16.1.1

> set type=SOA
> unc.edu
Server:  aracaju.cvirtual.com.br
Address:  172.16.1.1

Non-authoritative answer:
unc.edu
        origin = ns.unc.edu
        mail addr = host-reg.ns.unc.edu
        serial = 1998122401
        refresh = 14400 (4 horas)
        retry   = 3600 (1 hour)
        expire  = 1209600 (14 days)
        minimum ttl = 86400 (1 day)

Authoritative answers can be found from:
UNC.EDU nameserver = NS.UNC.EDU
NS.UNC.EDU       internet address = 152.2.21.1
```

De uma forma similar pode-se pesquisar os registros MX, etc.. Usando-se o tipo ANY fará com que todos os registros associados a um determinado nome sejam apresentados.

```
> set type=MX
> unc.edu
Non-authoritative answer:
unc.edu preference = 10, mail exchanger = imsety.oit.unc.edu
imsety.oit.unc.edu     internet address = 152.2.21.99

Authoritative answers can be found from:
UNC.EDU nameserver = NS.UNC.EDU
```

```
NS.UNC.EDU       internet address = 152.2.21.1
```

Uma aplicação prática do programa **nslookup**, além de servir à depuração da configuração do DNS, é obter a lista de servidores de domínios raiz para o arquivo **named.ca**. Pode-se obtê-la através da pesquisa de todos os tipos de registros NS associados com o domínio raiz:

```
> set typ=NS
> .
Server:  aracaju.cvirtual.com.br
Address:  172.16.1.1

Non-authoritative answer:
(root)  nameserver = M.ROOT-SERVERS.NET
(root)  nameserver = A.ROOT-SERVERS.NET
(root)  nameserver = H.ROOT-SERVERS.NET
(root)  nameserver = B.ROOT-SERVERS.NET
(root)  nameserver = C.ROOT-SERVERS.NET
(root)  nameserver = D.ROOT-SERVERS.NET
(root)  nameserver = E.ROOT-SERVERS.NET
(root)  nameserver = I.ROOT-SERVERS.NET

M.ROOT-SERVERS.NET    internet address = 202.12.27.33
A.ROOT-SERVERS.NET    internet address = 198.41.0.4
H.ROOT-SERVERS.NET    internet address = 128.63.2.53
B.ROOT-SERVERS.NET    internet address = 128.9.0.107
C.ROOT-SERVERS.NET    internet address = 192.33.4.12
D.ROOT-SERVERS.NET    internet address = 128.8.10.90
E.ROOT-SERVERS.NET    internet address = 192.203.230.10
I.ROOT-SERVERS.NET    internet address = 192.36.148.17
```

A lista completa do conjunto de comandos disponíveis com o programa **nslookup** pode ser obtida através do comando **help**, a partir da linha de comando do programa, ou através da página de manual on-line (man nslookup)[5] ou no anexo de comandos relacionados a redes no final deste Guia.

[5] em português no Conectiva Linux.

6.2.5 Outras Ferramentas Úteis

Há algumas outras ferramentas que podem auxiliar as tarefas de administração do DNS/BIND. Descreveremos rapidamente algumas delas. Por favor consulte a documentação que acompanha estes programas para maiores informações sobre o seu uso.

O programa `hostcvt` é uma ferramenta que ajuda com a configuração inicial do BIND, convertendo o arquivo `/etc/hosts` em arquivos de configuração do servidor de nomes `named`. Ele gera tanto os registros tipo A como os de mapeamento reverso PTR, cuidando ainda dos nomes alternativos das máquinas. Obviamente ele não faz todo o trabalho do administrador, pois pode ser necessário atualizar os valores do registro SOA, por exemplo, ou adicionar registros MMX. Porém ele pode ajudá-lo a economizar algumas aspirinas. Ele integra os fontes do BIND e pode ser encontrado como um pacote à parte em diversos servidores FTP `Linux`.

Após configurar o servidor de nomes, é aconselhável testar a sua configuração. A ferramenta ideal (segundo o autor) é o programa `dnswalk`, uma aplicação desenvolvida em `perl` e que possibilita uma "caminhada" pelas bases de dados DNS, procurando pelos erros mais comuns e verificando eventuais inconsistências. O programa foi liberado recentemente e pode ser encontrado em diversos sites FTP `Linux`.

Capítulo 7

IP em Linha Serial

Os protocolos de linha serial SLIP e PPP fornecem conectividade com a Internet para aqueles que não podem manter linhas dedicadas, apesar de também poderem ser utilizados também neste tipo de linhas. Além de um modem e de uma placa serial equipada com um buffer com suporte a filas FIFO, nenhum outro hardware é necessário. Seu uso não é muito mais complicado do que uma caixa de correio e um crescente número de organizações particulares oferecem conexões IP discadas e dedicadas a custos aceitáveis para todo o tipo de usuário.

Existem tanto programas de controle de dispositivos SLIP como PPP disponíveis para `Linux`. O SLIP vem sendo usado há bastante tempo, e funciona de uma maneira razoavelmente confiável. Um programa de controle para PPP foi recentemente desenvolvido por Michael Callahan e Al Longyear. Ele será descrito no próximo capítulo.

7.1 Requisitos Gerais

Para usar SLIP ou PPP, é necessário configurar algumas funcionalidades básicas de rede, conforme descrito nos capítulos anteriores. No mínimo, deve-se configurar a interface local de rede e fornecer um serviço de resolução de nomes. Ao conectar-se a Internet, certamente será necessário usar o DNS. A opção mais simples é colocar o endereço de algum servidor de nomes no arquivo `resolv.conf` local. Este servidor será requisitado tão logo a conexão SLIP ou PPP esteja ativada. Quanto mais próximo este servidor de nomes estiver do ponto de onde se estiver discando, melhor.

De qualquer forma esta ainda não é a solução ideal, pois todas as buscas por nomes ainda serão feitas através da conexão SLIP/PPP. Caso a largura de banda que este procedimento consuma seja um problema (o que é improvável), pode-se ainda configurar um servidor de nomes *somente para cache*. Ele não configura realmente um domínio, mas somente age como um ponto de resposta para todas as requisições ao DNS produzidas na máquina local. A vantagem deste esquema é que ele cria uma cache, fazendo com que muitas requisições sejam enviadas pela linha serial apenas uma vez. O arquivo `named.boot` para um servidor somente de cache assemelha-se a:

```
; arquivo named.boot para um servidor somente de cache
directory                          /var/named

primary       0.0.127.in-addr.arpa   db.127.0.0 ; rede da interface local
cache         .                      db.cache   ; servidores raiz
```

Em adição ao arquivo `name.boot`, será necessário ainda configurar o arquivo `db.cache` com uma lista válida de servidores de domínios raiz. Isso é descrito ao final do capítulo Configuração de Servidor de Nomes e Resolvedor de Endereços.

7.2 Operação do SLIP

Servidores IP de acesso discado freqüentemente oferecem serviço SLIP através de contas especiais para os usuários. Após conectar-se com tal conta, o usuário ainda não tem acesso a um ambiente usual de um interpretador de comandos. Ao invés disso, um programa é executado para habilitar o programa do protocolo SLIP do servidor para a linha serial e configurar a interface de rede apropriada. O mesmo deve ser feito no lado local da conexão.

Em alguns sistemas operacionais, o protocolo SLIP é um programa de usuário, no `Linux`, ele é parte integrante do kernel, o que o torna bem mais rápido. Porém isto exige que a linha serial seja convertida para o modo SLIP de forma explícita. Isto é feito por meios de regras de funcionamento especiais para a linha tty, denominadas SLIPDISC. Enquanto um terminal normal (tty) utilizar regras de funcionamento usuais (DISC0), ele irá trocar dados apenas com processos de usuários, usando as chamadas normais `read(2)` e `write(2)`. Porém o programa de controle do SLIP é incapaz de escrever em um terminal tty ou ler diretamente de um tty, já que todos os dados oriundos da porta serial serão passados diretamente para o programa de controle do SLIP.

O programa de controle do SLIP entende um grande número de variações no protocolo SLIP. Além do SLIP normal, ele compreende ainda o CSLIP, o qual executa a chamada compressão de cabeçalho de Van Jacobson para compressão de pacotes IP enviados.[1] Isso aumenta notavelmente a performance em sessões interativas. Adicionalmente, há versões de seis bits para cada um desses protocolos.

Uma maneira simples de converter uma linha serial para o modo SLIP é usando a ferramenta `slattach`. Considerando-se que se tenha um modem em `/dev/cua3` e se tenha conectado a um servidor SLIP com sucesso. Deve-se então executar:

```
# slattach /dev/cua3 &
```

Isto irá trocar o modo de funcionamento da linha de `cua3` para SLIPDISC, e irá conectá-la a uma das interfaces de rede SLIP. Caso esta seja a primeira conexão SLIP ativa, a linha será conectada a `sl0`; o segundo será anexado a `sl1`, e assim sucessivamente.

O encapsulamento padrão escolhido por `slattach` é o CSLIP. É possível escolher qualquer outro modo usando a opção `-p`. Para usar o protocolo SLIP (sem compressão), deve-se usar:

```
# slattach -p slip /dev/cua3 &
```

Outras opções são `cslip`, `slip6`, `cslip6` (para a versão de seis bits do SLIP) e a opção `adaptive` para SLIP adaptativo. As versões mais atuais deixam que o kernel descubra que tipo de encapsulamento SLIP o usuário remoto está utilizando.

Note que é necessário utilizar o mesmo encapsulamento que a outra ponta da ligação. Por exemplo, se a máquina `atibaia` usa CSLIP, deve-se também usá-lo. Os sintomas de um conflito podem ser detectados através de um comando `ping` para uma máquina remota que nunca retorna uma resposta. Se a outra máquina responder, ainda assim podem ser recebidas mensagens como “`Não foi possível construir o cabeçalho ICMP`”. Uma maneira de evitar essas dificuldades é usar SLIP adaptativo.

[1] A compressão de cabeçalhos de Van Jacobson está descrita na RFC 1441.

De fato, `slattach` não apenas permite habilitar o SLIP, mas também outros protocolos que usam a linha serial da mesma maneira, como por exemplo PPP ou KISS (outro protocolo utilizado por usuários de rádio amador). Para maiores detalhes, por favor refira-se à página do manual `slattach(8)`.

Após transferir a linha para o programa de controle do SLIP, é preciso configurar a interface de rede. Novamente, faremos isso usando os comandos `ifconfig` e `route`. Assumindo-se que a partir de `aracaju`, discamos para um servidor chamado `aquitemslip`. Pode-se então executar:

```
# ifconfig sl0 aracaju pointopoint aquitemslip
# route add aquitemslip
# route add default gw aquitemslip
```

O primeiro comando configura a interface como uma conexão ponto a ponto para `aquitemslip`, enquanto que o segundo e terceiro adicionam a rota para `aquitemslip` e a rota padrão usando `aquitemslip` como um caminho padrão.

Para desconectar uma conexão SLIP, deve-se primeiro remover todas as rotas relacionadas com a máquina `aquitemslip`, usando os programas `route` com a opção `del`, após deve-se desativar a interface e enviar para o programa `slattach` o sinal de desconexão. Em outras palavras, é necessário desconectar o modem usando o programa de terminal novamente:

```
# route del default
# route del aquitemslip
# ifconfig sl0 down
# kill -HUP 516
```

7.3 Usando `dip`

Bem, isso parece ser um pouco mais simples. Entretanto, pode-se ainda automatizar os passos acima para que se tenha apenas que ativar um simples comando que realize todos estes procedimentos. Isso é o que o programa `dip` faz.[2] A versão atual, quando escrevemos este documento, é a versão 3.3.7, sendo que ele foi bastante alterado por um grande número de pessoas, então não se pode mais falar sobre *o programa* `dip`.

[2] `dip` significa *IP discado*. Ele foi escrito por Fred van Kempen.

O programa `dip` fornece um interpretador para uma linguagem de programação simples que permite a operação de modem, conversão da linha para o modo SLIP e configuração de interfaces. Isto é um pouco primitivo e restritivo, mas suficiente para a maioria dos casos.

Para ser capaz de configurar a interface SLIP, o programa `dip` requer privilégios de superusuário. Isso faz com que `dip` seja executado com identificação do usuário `root`, permitindo a todos os usuários discarem para um servidor SLIP, sem que eles necessitem de privilégios de superusuário. Isto pode ser perigoso, pois a configuração de interfaces adulteradas e rotas padrão com o programa `dip` pode interromper drasticamente todo o roteamento na rede. Pior ainda, isso dará aos seus usuários poderes para conectarem-se a *qualquer* servidor SLIP e desferir ataques perigosos à rede. Então, caso se deva permitir que os usuários executem uma conexão SLIP, devem ser escritos pequenos programas para cada provável servidor SLIP, onde `dip` será executado em um programa específico para cada conexão. Estes programas podem então seguramente fazerem uso do comando suid `root`.[3]

7.3.1 Um Programa Exemplo

```
# Exemplo de um programa dip para conexão com a máquina ``aquitemslip''.
# Define os nomes locais e remotos e os endereços

get $local aracaju
get $remote aquitemslip

port cua3                # escolhe uma porta serial
speed 38400              # configura a velocidade para o máximo
modem HAYES              # define o tipo de modem
reset                    # reinicializa o modem
flush                    # descarrega as respostas do modem

# Preparando a discagem
send ATQ0V1E1X1\r
wait OK 2
if $errlvl != 0 goto error
dial 41988
```

[3] `diplogin` pode (e deve) ser executado também com suid `root`. Veja a sessão no fim deste capítulo.

```
 if $errlvl != 0 goto error
 wait CONNECT 60
 if $errlvl != 0 goto error

 # Ok, agora estamos conectados
 sleep 3
 send \r\n\r\n
 wait ogin: 10
 if $errlvl != 0 goto error
 send Saracaju\n
 wait ssword: 5
 if $errlvl != 0 goto error
 send euqueroslip\n
 wait running 30
 if $errlvl != 0 goto error

 # Estamos conectados e o lado remoto está iniciando o protocolo SLIP

 print Connected to $remote with address $rmtip
 default                 # Torna esta conexão como a rota padrão
 mode SLIP               # Iniciamos o protocolo SLIP local.

 # mensagem em caso de erro
error:
 print SLIP to $remote failed.
```

Este programa pode ser usado na conexão com a máquina `aquitemslip` através do utilitário `dip` com o nome do programa como argumento:

```
# dip aquitemslip.dip
DIP: Dialup IP Protocol Driver version 3.3.7 (12/13/93)
Written by Fred N. van Kempen, MicroWalt Corporation.

connected to aquitemslip.pantanal.edu.br with addr 193.174.7.129
#
```

Após a conexão à máquina `aquitemslip` e habilitar-se o protocolo SLIP, o programa `dip` irá liberar o terminal e passará para o modo de execução em segundo plano. A partir daí pode-se iniciar o uso dos serviços de rede através da conexão SLIP. Para finalizar a conexão, basta acionar o programa `dip` com a opção `-k`.

Isso envia um sinal de desconexão para o processo `dip`, usando a identificação de processo (pid) registrada no arquivo `/etc/dip.pid`: [4]

```
# kill -k
```

Na linguagem de programação do programa `dip`, palavras chaves iniciadas com um cifrão ($) significam nomes de variáveis. O programa `dip` tem um conjunto de variáveis predefinidas listadas abaixo. `$remote` e `$local` por exemplo, contêm o nome das máquinas local e remotas envolvidas na conexão SLIP.

Os primeiros dois argumentos no programa de exemplo são os comandos `get`, o qual é a forma do programa `dip` inicializar uma variável. No nosso exemplo, as variáveis `$remote` e `$local` são configuradas com o conteúdo `aquitemslip` e `aracaju`, respectivamente.

Os próximos cinco comandos configuram a linha do terminal e o modem. A palavra reservada `reset` envia um conjunto de caracteres de reinicialização para o modem. No caso de modems compatíveis com o padrão Hayes ele é igual a `ATZ`. O próximo comando inibe as respostas do modem, permitindo que a conversação de acesso possa funcionar adequadamente. Esta conversação é bastante objetiva: simplesmente disca 41988, o número do telefone de `aquitemslip`, acessa o sistema utilizando a conta `aracaju` usando a senha `euqueroslip`. A palavra reservada `wait` faz com que `dip` aguarde por um conjunto de caracteres informado em seu primeiro argumento, o número fornecido como segundo argumento indica o tempo de espera em segundos, caso os caracteres não sejam recebidos. A palavra chave `if` intercalada nos procedimentos de acesso verifica se os comandos estão sendo executados corretamente.

Os comandos finais, executados após a obtenção de acesso são o `default`, o qual faz com que o SLIP use a conexão como caminho padrão para todas as máquinas, e `mode`, que habilita o modo SLIP na linha e configura a interface e a tabela de roteamento.

7.3.2 Referências dip

Apesar de ser largamente utilizado, o programa `dip` não está muito bem documentado. Nesta seção apresentaremos uma referência à maioria dos comandos `dip`. É

[4] Veja o grupo de notícias `alt.tla` para maiores informações bem humoradas sobre acrônimos de três letras.

possível ainda ter-se uma visão geral de todos os comandos através da execução do programa `dip` em modo de teste, ao informar-se o comando `help`. Para descobrir a sintaxe de um comando, basta informá-lo sem argumentos.

```
$ dip -t
DIP: Dialup IP Protocol Driver version 3.3.7 (12/13/93)
Written by Fred N. van Kempen, MicroWalt Corporation.

DIP> help
DIP knows about the following commands:

databits default  dial     echo     flush
get      goto     help     if       init
mode     modem    parity   print    port
reset    send     sleep    speed    stopbits
term     wait

DIP> echo
Usage: echo on|off
DIP> _
```

Através das seções seguintes, exemplos apresentados com a referência `DIP>` mostram como executar um comando no modo de teste e qual o resultado que ele produz. Exemplos sem este indicativo devem ser entendidos como partes de um programa `dip`.

Os comandos de Modem

Há diversos comandos que o programa `dip` disponibiliza para a configuração da linha serial e do modem. Algumas destas são óbvias, como `port`, o qual seleciona a porta serial a ser usada, `speed`, `databits`, `stopbits` e `parity`, os quais configuram os parâmetros comuns de linha.

O comando `modem` seleciona o tipo de modem. Atualmente somente há um tipo suportado, que é `HAYES` (necessariamente em maiúsculas). Deve-se informar o tipo de modem para o programa `dip`, caso contrário ele recusará os comandos `dial` e `reset`. Este comando envia um conjunto de caracteres de reinicialização do modem. Os caracteres dependem do tipo de modem. Para modems HAYES e compatíveis o comando é keywordATZ.

O comando `flush` é usado para limpar todas as respostas enviadas pelo modem até aquele momento. De outra forma, um programa de conversação seguido do comando `reset` pode ser um tanto confuso, uma vez que seria possível receber por exemplo uma resposta `OK` de um comando antigo enviado para o modem.

O comando `init` seleciona um conjunto de caracteres de inicialização a serem enviados para o modem antes da discagem. O padrão para modems hayes e compatíveis é "`ATE0 Q0 V1 X1`", o qual habilita o eco de comandos e códigos de resultado longos e seleciona o método de discagem sem necessidade de checagem de tom de linha.

O comando `dial` envia um conjunto de caracteres de inicialização para o modem e disca para o sistema remoto. O comando de discagem padrão para os modems Hayes e compatíveis é `ATD`.

`echo` e `term`

O comando `echo` serve como uma ferramenta de auxílio à depuração, uma vez que o uso do parâmetro `echo on` faz com que o programa `dip` ecoe no terminal tudo o que for enviado pelo dispositivo serial. Essa funcionalidade pode ser desligada através do comando `echo off`.

`dip` permite que o programa seja temporariamente suspenso e se possa entrar em modo terminal. Desta forma, pode-se usar o programa `dip` como um programa comum de emulação de terminal, escrevendo e lendo na linha serial. Para finalizar o modo terminal, basta teclar `Ctrl-]`.

O comando `get`

O comando `get` é a forma utilizada pelo programa `dip` de inicializar uma variável. É a forma mais simples de configurar uma variável como uma constante, confopme visto no exemplo anterior. Pode-se ainda solicitar informações ao usuário do programa em tempo de execução, através do uso do comando `ask`, no lugar de definir-se um valor para a variável:

```
DIP> get $local ask
Enter the value for $local: _
```

Um terceiro método de obter-se informações é solicitá-las à máquina remota. Apesar de parecer estranho num primeiro momento, isso pode ser muito útil em alguns casos: alguns servidores SLIP não permitem que o usuário utilize o seu próprio endereço IP em uma conexão, mas irão fornecer um dentro de uma faixa predefinida, assim que a conexão for estabelecida, listando-o em uma mensagem impressa no terminal. A mensagem terá uma aparência similar a "`Your address: 193.174.7.202`", sendo possível então ao programa `dip` usá-la para definir o endereço IP da estação:

```
... conversação ....
wait address: 10
get $locip remote
```

O comando `print`

Este comando é utilizado para apresentar um texto na console em que o programa `dip` foi inicializado. Quaisquer variáveis podem ser utilizadas com este comando, como por exemplo:

```
DIP> print Usando a  porta $port à velocidade de  $speed bps
Usando a porta cua3 à velocidade de 38400 bps
```

Nomes de Variáveis

O programa `dip` entende somente um conjunto predefinido de variáveis. Um nome de variável sempre inicia com o símbolo de cifrão ($) e deve ser escrito em minúsculas.

As variáveis `$local` e `$locip` contêm o nome da máquina local e o endereço IP, respectivamente. Configurando o nome da máquina faz com que `dip` armazene o nome canônico em `$local`, ao mesmo tempo que inicializa a variável `$locip` com o endereço IP correspondente. Um processo similar ocorre ao se configurar a variável `$locip`.

As variáveis `$remote` e `$rmtip` fazem o mesmo com o nome da máquina remota e com seu endereço IP. `$mtu` contém o valor da Unidade Máxima de Transferência MTU da conexão.

Estas cinco variáveis são as únicas que podem ter os seus valores definidos diretamente pelo comando `get`. Outras variáveis podem ser iniciadas somente através de seus comandos correspondentes, porém poderão ser usadas com o comando `print`. São elas as variáveis `$modem`, `$port` e `$speed`.

`$errlvl` é a variável que contém o resultado do último comando executado. Um nível de erro igual a zero indica sucesso, enquanto que valores diferentes de zero significam erro.

Os Comandos `if` e `goto`

O comando `if` aciona uma condição a ser testada. Sua sintaxe é:

`if` *`var op número`* `goto` *`localização`*

onde a expressão deve ser uma simples comparação entre uma das variáveis `$errlvl`, `$locip` ou `$rmtip`. O segundo parâmetro deve ser um número inteiro, o parâmetro *`op`* deve ser um dos seguintes `==`, `!=`, `<`, `>`, `<=`, e `>=`.

O comando `goto` faz com que a execução do programa continue na linha que contém o endereço indicado. Um endereço estará definido sempre no início da linha e deve ser seguido por dois pontos.

`send`, `wait` e `sleep`

Estes comandos auxiliam na implementação de programas de conversação no utilitário `dip`. `send` envia os seus argumentos pela linha serial, não suporta variáveis, mas entende todas as seqüências de caracteres no estilo C, como por exemplo `\n` e `\b`. O caractere til é usado como abreviatura de retorno de carro e nova linha.

`wait` recebe uma palavra como argumento e pesquisa a entrada da linha serial até que consiga reconhecê-la. A palavra não pode conter espaços em branco. Opcionalmente, pode-se fornecer um tempo máximo de espera através do parâmetro `wait`. Se a palavra indicada não for recebida nos n segundos especificados, o comando retornará o valor 1 na variável `$errlvl`.

O comando `sleep` pode ser usado para que se aguarde um determinado espaço de tempo. Pode ser usado, por exemplo, na espera de uma seqüência de acesso a ser completada. Novamente o intervalo é especificado em segundos.

`mode` e `default`

Estes comandos são usados para acionar o protocolo SLIP e configurar a interface.

O comando `mode` é o último comando executado pelo programa `dip` antes de entrar no modo servidor. A menos que um erro ocorra, o comando não gerará nenhuma resposta.

`mode` recebe o nome do protocolo como argumento. `dip` atualmente reconhece como válidos `SLIP` e `CSLIP`. A versão atual não reconhece SLIP adaptativo por exemplo.

Após habilitar o modo SLIP na linha serial, `dip` executa o comando `ifconfig` para configurar a interface como uma conexão ponto a ponto e executa o comando `route` para configurar a rota para a máquina remota.

Caso seja executado o comando `default` antes do comando `mode`, `dip` irá tornar a conexão SLIP como a rota padrão da máquina local.

7.4 Executando no Modo Servidor

Configurar um cliente SLIP foi a parte mais difícil. Fazer o contrário, tornando uma máquina um servidor SLIP é muito mais simples.

Uma forma de fazer isso é utilizar o programa `dip` no modo servidor, o que pode ser feito através de sua execução como `diplogin`. O arquivo de configuração é denominado `/etc/diphosts`, o qual associa nomes de acesso com endereços definidos para a máquina. Alternativamente, pode-se usar ainda o `sliplogin`, uma ferramenta derivada do BSD, com um conjunto de funcionalidades mais flexíveis, que permitem a execução de programas estando as máquinas conectadas ou não.

Ambos os programas requerem a criação de uma conta de acesso para cada cliente SLIP. Por exemplo, para criar uma conta de acesso SLIP para um cliente denominado Roberto Azevedo na máquina `canoas.cvirtual.com.br`, deve-se criar um usuário chamado `azevedo` através da adição da seguinte linha ao arquivo `/etc/passwd`:

```
azevedo:*:501:60:conta SLIP de Roberto Azevedo:/tmp:/usr/sbin/diplogin
```

Após isso deve-se definir a senha do usuário `azevedo` através do utilitário `passwd`.

Agora, quando o usuário `azevedo` acessar o sistema, `dip` será iniciado no modo servidor. Para verificar se o usuário tem permissão de uso do SLIP, ele irá procurar pelo seu nome no arquivo `/etc/diphosts`. Este contém detalhes dos direitos de acesso e parâmetros de conexão para cada usuário SLIP. Uma entrada simples neste arquivo para o usuário `azevedo` pode ter o seguinte formato:

```
azevedo::canoas.cvirtual.com.br:Roberto Azevedo:SLIP,296
```

O primeiro campo separado por dois pontos é o nome da conta de acesso do usuário, a qual deve ser usada durante as conexões. O segundo campo pode conter uma senha adicional (veja abaixo), o terceiro é o nome ou o endereço IP da máquina do usuário. A seguir um campo de uso livre e o último campo contendo os parâmetros de conexão. Separados por vírgulas, temos ainda o protocolo a ser usado na conexão e a seguir o seu MTU.

Quando o usuário `azevedo` acessa o sistema, o programa `diplogin` extrai informações sobre ele no arquivo `diphosts` e, se o segundo campo não estiver vazio, solicita uma "senha externa de segurança". A informação digitada pelo usuário será comparada com a senha não cifrada do arquivo `diphosts`. Caso elas não coincidam, a tentativa de acesso será rejeitada.

Caso contrário, `diplogin` prosseguirá, acionando o modo SLIP na linha serial e configurará a interface e a rota. Esta conexão permanecerá estabelecida até que o usuário saia do sistema ou a ligação telefônica seja finalizada. Nestes casos `diplogin` retornará a linha ao seu modo normal e será encerrado.

O programa `diplogin` necessita de privilégios de superusuário para ser executado. Caso não se tenha um programa `dip` sendo executado via setuid `root`, pode-se criar o `diplogin` como uma cópia separada do programa `dip` ao invés de uma simples ligação simbólica. `diplogin` poderá então ser configurado através do setuid com segurança, sem afetar o status do programa `dip`.

Capítulo 8

O Protocolo Ponto a Ponto

8.1 Desvendado os P

Assim como o SLIP, o PPP é um protocolo destinado ao envio de datagramas através de conexões seriais, porém ele supre algumas deficiências do protocolo SLIP. Permite que as pontas da conexão negociem opções como endereço IP e o tamanho máximo do datagrama durante a sua inicialização e provê uma forma de autorização de acesso para clientes. Para cada uma destas capacidades, PPP tem um protocolo em separado. A seguir discutiremos os fundamentos do protocolo PPP. Esta discussão está longe de ser completa sendo que é sugerida a leitura da RFC 1661, assim como de uma dúzia de outras relacionadas para uma visão completa do protocolo.[1]

Na camada mais básica do PPP está o protocolo HDLC - *Controle de Conexões de Dados de Alto Nível*[2][3][4] o qual define os limites das unidades de transmissão na camada de conexão de dados[5] e provê uma verificação de integridade de mensagens com chaves de 16 bits. Opostamente ao SLIP, o qual é mais primitivo, é possível ainda o encapsulamento de outros protocolos, ou seja um pacote PPP pode conter dentro dele informações de protocolos como IP, IPX da Novell ou Appletalk. Isso

[1] Os RFCs relevantes estão descritos no segmento Bibliografia ao final deste Guia.

[2] High-Level Data Link Control

[3] Na verdade HDLC é um protocolo muito mais genérico, definido pela Organização Mundial de Padrões - ISO

[4] International Standards Organization (ISO).

[5] Segundo a RFC 1661.

é possível através da adição de um campo destinado à definição do protocolo na unidade básica HDLC que identifica o tipo de pacote que está sendo transmitido.

LCP, o Protocolo de Controle de Conexão[6] é usado sobre o HDLC para negociar opções referentes à conexão de dados, como por exemplo a Unidade Máxima de Transferência - MTU, que define o tamanho máximo do datagrama que uma das pontas da conexão aceita receber. LCP é também responsável pelo monitoramento da qualidade da conexão e pela detecção de linhas em autoteste[7].

Um importante passo no estágio de configuração de uma conexão PPP é a autorização de acesso do cliente. Ainda que não seja obrigatória, é realmente uma necessidade para acessos via linhas discadas. Normalmente a máquina para a qual se disca (o servidor) solicita ao cliente uma comprovação de identidade através da informação da senha de acesso. Caso o cliente falhe em informar uma senha correta, a conexão é finalizada. Com o PPP a autorização funciona em ambos os sentidos, ou seja o cliente pode solicitar que o servidor também se identifique. Estes procedimentos de identificação são totalmente independentes. Há dois protocolos distintos encarregados destas tarefas, os quais são discutidos a seguir, denominados PAP - Protocolo de Autenticação de Senhas[8] e CHAP - Protocolo de Autenticação de Apresentação[9].

Cada protocolo de rede que é roteado através de uma conexão de dados, como por exemplo IP, Appletalk, etc., é configurado dinamicamente utilizado-se o Protocolo de Controle de Redes (NCP) correspondente. Por exemplo, para enviar datagramas IP através de uma conexão, ambas as pontas da conexão PPP devem inicialmente negociar qual o endereço IP que cada uma utilizará. O protocolo de controle usado para isso é denominado IPCP - Protocolo de Controle do Protocolo Internet[10].

Além de enviar datagramas IP através de uma conexão serial, o PPP também suporta a compressão de cabeçalhos de datagramas IP de Van Jacobson. Esta técnica comprime os cabeçalhos dos pacotes TCP em tamanhos de até três bytes. É também usado no CSLIP e é mais coloquialmente conhecido como compressão de cabeçalho VJ. O uso desta compressão pode ser negociada em tempo de inicialização através do protocolo IPCP.

[6] Link Control Protocol

[7] Uma linha serial encontra-se em autoteste quando recebe de volta tudo aquilo que é gravado nela.

[8] Password Authentication Protocol

[9] Challenge Handshake Authentication Protocol

[10] Internet Protocol Control Protocol

8.2 PPP no Linux

No Linux, as funcionalidades PPP estão divididas em duas partes, um programa de controle HDLC localizado no kernel e um programa servidor denominado pppd que administra os vários protocolos de controle. A versão atual do PPP para Linux é ppp-2.3.5, que contém o módulo do kernel para PPP, o servidor pppd e um programa chamado chat utilizado para discar para sistemas remotos.

O programa de controle do PPP foi escrito por Michael Callahan. pppd foi derivado de uma livre implementação do PPP para máquinas Sun e 386BSD, escrita por Drew Perkins e outros e mantida por Paul Mackerras. Ela foi portada para o Linux por Al Longyear.[11] O programa chat foi escrito por Karl Fox.[12]

Assim como no SLIP, o PPP está implementado sobre uma forma especial de funcionamento da linha do terminal. Para usar uma linha serial em uma conexão PPP, é necessário inicialmente estabelecer uma conexão com um modem da forma usual e subseqüentemente converter a linha para o modo de operação do protocolo PPP. Neste modo, todo o tráfego será tratado pelo programa de controle do PPP, o qual verifica as unidades de transmissão HDLC (cada unidade contém um número de validação de 16 bits), desempacota os dados e os despacha. Atualmente ele pode manusear datagramas IP, opcionalmente com cabeçalhos comprimidos VJ, assim como pode lidar também com pacotes IPX.

O programa de controle do kernel é auxiliado pelo pppd, o servidor PPP, o qual executa toda a fase de inicialização e autenticação antes do tráfego de rede ser estabelecido na conexão serial. O comportamento do ppp pode ser ajustado através de diversas opções. Como o PPP é relativamente complexo, é praticamente impossível explicar todas os aspectos do pppd. Porém procuraremos oferecer uma introdução para a maioria deles. Para maiores informações, veja as páginas de manual e os arquivos READMEs que acompanham os fontes do pppd, os quais devem ajudar a elucidar eventuais dúvidas. Caso o problema persista, mesmo após a leitura da documentação, pode-se inscrever em um grupo de notícias como o comp.protocols.ppp em busca de auxílio, o qual é o lugar onde se podem encontrar a maioria das pessoas envolvidas no desenvolvimento do programa pppd.

[11] Ambos os autores declaram que estarão ocupados por um bom tempo. Caso você tenha alguma questão sobre PPP em geral, o melhor caminho serão as listas de discussão (como por exemplo a linux-br) ou os canais IRC de Linux disponíveis na Internet.

[12] karl@morningstar.com.

8.3 Executando o pppd

Quando for necessário interligar-se a Internet através de uma conexão PPP, há que se configurar algumas funcionalidades básicas de rede tais como o dispositivo local de rede e o resolvedor de nomes. Ambos estão descritos em capítulos anteriores. Há ainda alguns aspectos sobre o uso de DNS sobre uma ligação serial que foram abordadas no capítulo anterior, dedicado ao protocolo SLIP.

Como um exemplo introdutório de como estabelecer uma conexão PPP com o servidor `pppd`, vamos assumir que estamos trabalhando com a máquina `aracaju` novamente. Já houve alguma vez um acesso discado com o servidor PPP denominado `itaim` e foi utilizada uma conta de acesso denominada `ppp`. O servidor `itaim` também já foi configurado com um programa de controle PPP. Após finalizar o programa de comunicação usado para discagem deve-se executar o seguinte comando:

```
# pppd /dev/cua3 38400 crtscts defaultroute
```

Isso irá converter a linha serial `cua3` para o modo PPP e estabelecer uma conexão IP com `itaim`. A velocidade de transferência usada na porta serial será de 38400 bps. A opção `crtscts` indica que a negociação de parâmetros será feita na porta, a qual deve necessariamente trabalhar acima de 9600 bps.

A primeira ação que o `pppd` toma após a sua inicialização é negociar as diversas características da conexão com o usuário remoto usando LCP. Normalmente o conjunto padrão de opções que o `pppd` apresenta funciona prontamente. Retornaremos ao LCP mais adiante nesta seção.

Por hora, assumiremos que `itaim` não solicita qualquer tipo de autenticação, portanto a fase de configuração foi completada com sucesso.

O programa `pppd` irá então negociar os parâmetros com a outra ponta da conexão usando o IPCP, o protocolo de controle IP. Uma vez que não especificamos qualquer endereço IP em particular para o servidor `pppd`, ele tentará obter o endereço da máquina local através da pesquisa junto ao resolvedor de nomes local. Ambos então anunciarão o seu endereço, um para o outro.

Normalmente não há nada errado com estes padrões. Mesmo se a máquina está em uma rede Ethernet, pode-se usar o mesmo endereço IP para ambas as interfaces: PPP e Ethernet. De qualquer forma, o `pppd` permite o uso de endereços distintos, ou mesmo perguntar sobre algum endereço específico a ser utilizado. Estas opções serão discutidas em uma seção posterior.

Após a fase de configuração através do IPCP, o `pppd` irá preparar a camada de rede da máquina para uso da conexão PPP. Inicialmente ele configura a interface de rede PPP como uma conexão ponto a ponto, usando `ppp0` para a primeira conexão PPP ativa, `ppp1` para a segunda e assim por diante. Após, ele irá configurar uma entrada na tabela de roteamento que apontará para a máquina na outra ponta da conexão. No exemplo mostrado acima, `pppd` irá apontar a rota padrão de rede para a máquina `itaim`, uma vez que a opção `defaultroute` foi fornecida.[13] Isso faz com que todos os datagramas para as máquinas que não estejam na rede local, sejam enviados através de `itaim`. Há outras formas de roteamento que são suportadas pelo `pppd`, as quais apresentaremos mais detalhadamente a seguir.

8.4 Usando Arquivos de Opções

Antes do `pppd` receber os seus argumentos de linha de comando, ele pesquisa diversos arquivos na busca de opções padrão. Estes arquivos podem conter qualquer argumento válido que seja utilizado na linha de comando, localizados em um número de linhas arbitrário. Comentários são definidos através do sinal #.

O primeiro arquivo de opções é denominado `/etc/ppp/options`, o qual é sempre pesquisado quando o `pppd` é inicializado. Deve ser usado para a configuração de alguns valores padrão globais, uma vez que eles permitem inibir que os usuários executem ações que possam comprometer a segurança. Por exemplo, para fazer com que o programa `pppd` solicite algum tipo de autenticação (ou PAP ou CHAP), deve-se adicionar a opção `auth` ao arquivo. Esta opção não poderá ser substituída pelo usuário, tornando-se impossível estabelecer uma conexão PPP com qualquer sistema que não esteja definido na base de dados de autenticação.

O outro arquivo de opções que é lido após o `/etc/ppp/options` é denominado `.ppprc` e está localizado no diretório pessoal do usuário. Ele permite que cada usuário tenha as suas próprias opções padrão.

Um arquivo exemplo de opções, `/etc/ppp/options`, terá uma aparência similar a:

```
# opções globais para o pppd em execução na máquina aracaju.cvirtual.com.br
```

[13] A rota padrão de rede é instalada somente se nenhuma outra estiver presente.

```
auth                   # requer autenticação
usehostname            # usa o nome de máquina local para CHAP
lock                   # reserva de dispositivos em estilo similar ao UUCP
domain cvirtual.com.br # nome do domínio local
```

As primeiras duas linhas do arquivo aplicam-se à autenticação e serão explicadas abaixo. A palavra chave `lock` faz com que o `pppd` torne-se compatível com o método padrão de reserva de dispositivos do UUCP. Com esta convenção cada processo que acesse um dispositivo serial, digamos por exemplo o `/dev/cua3`, criará um arquivo de reserva deste recurso denominado `LCK..cua3` no diretório de tarefas do UUCP, sinalizando a todos os demais processos que desejem utilizar este recurso, que ele está em uso. Isso é necessário para prevenir que qualquer outro programa, como por exemplo o `minicom` ou o `uucico` abra a linha serial enquanto o PPP está sendo usado.

A razão para prover estas opções no arquivo de configuração global, conforme descrito anteriormente, é a impossibilidade do usuário substituir ou alterar estas opções, provendo assim um razoável nível de segurança. Cabe salientar que algumas opções poderão ser alteradas posteriormente, dentre elas o parâmetro `connect`.

8.5 Discando Com o Programa `chat`

Uma das possíveis desvantagens do processo previamente descrito é a necessidade de estabelecimento manual de conexão antes da ativação do servidor `pppd`. Diferentemente do programa `dip`, `pppd` não tem uma linguagem de programação própria para discagem para sistemas remotos e acesso, mas pode ser usado em programas externos como programas de interpretadores de comando. O comando a ser executado pode ser fornecido a `pppd` com a opção de linha de comando `connect`. `pppd` irá redirecionar a entrada e a saída padrão do comando para a linha serial. Um programa útil para isto é denominado `expect`, escrito por Don Libes. Tem uma poderosa linguagem baseada em Tcl, e foi desenhado exatamente para este tipo de aplicação.

Junto com o pacote `pppd` há um programa similar denominado `chat`, o qual permite a criação de um programa de conexão similar a um programa UUCP. Basicamente, um programa de conversação consiste em uma seqüência de alternativas que esperamos receber de um sistema remoto e as respostas que devem ser enviadas. Denominaremos estas como expressões esperadas e enviadas respectivamente. A seguir apresentamos parte de um típico programa de conversação:

```
ogin: medeiros ssword: s3kr3t
```

Isso diz ao programa `chat` para aguardar que o sistema remoto envie o indicador de acesso ao sistema (login), retorne o nome de acesso `medeiros`, aguarde o indicativo de solicitação de senha (password), e envie a senha `s3kret`. Inicialmente basta esperar pela expressão `ogin:`, não importando se o indicativo de acesso inicia com um l maiúsculo ou minúsculo (já que ele foi suprimido) ou se ele chegou com problemas.

Este é, basicamente, todo o programa de conversação necessário. Um programa completo de discagem para um servidor PPP deveria obviamente incluir os comandos de modem apropriados. Assumindo que o modem entende o conjunto de comandos Hayes e o número de telefone seja igual a 318714. O programa completo em `chat` para estabelecer a conexão com `itaim` seria algo como:

```
$ chat -v '' ATZ OK ATDT318714 CONNECT '' ogin: ppp word: GaGariN
```

Por definição, o primeiro conjunto de caracteres deve ser uma expressão esperada, mas como o modem não responde absolutamente nada, foi criada uma expressão vazia para "enganar" o programa `chat`. Seguimos enviando a expressão `ATZ`, o comando de reinicialização para modems compatíveis com o padrão Hayes e aguardando a sua resposta, no caso (`OK`). A próxima expressão a ser enviada é o número do telefone e então o programa `chat` irá aguardar a expressão `CONNECT` como resposta. Isso é seguido por uma nova expressão vazia, já que não queremos enviar nenhum dado para a máquina remota, mas sim esperar pela expressão que indica o acesso ao sistema. O restante da conversação funciona exatamente da mesma forma que a descrita anteriormente.

A opção `-v` faz com que o programa `chat` registre todas as suas atividades, indicando-as através da palavra chave `local2` no servidor `syslog`.[14]

Especificar o programa de conversação na linha de comando pode apresentar certos riscos, uma vez que os usuários poderão visualizar toda a linha através do utilitário `ps`. Isso pode ser evitado, colocando-se o programa em um arquivo, chamado

[14] Ao se editar o arquivo `syslog.conf` para redirecionar estas mensagens para um arquivo, esteja certo que este arquivo pode ser acessado por todos os demais usuários, assim o programa `chat` sempre registrará todo o programa de conversação por padrão, incluindo senhas e tudo o mais.

digamos `conecta-itaim`. Após indica-se ao programa `chat` que o leia, através da opção `-f`, seguida do nome do arquivo. A linha de comando completa ao se invocar o arquivo terá um aspecto similar a:

```
# pppd connect "chat -f conecta-itaim" /dev/cua3 38400 -detach \
        crtscts modem defaultroute
```

Além do parâmetro `connect` que especifica um programa de conexão, adicionamos duas novas opções à linha de comando: `-detach`, que indica ao `pppd` para não entrar no modo de execução em segundo plano após a conexão e o parâmetro `modem` que aciona a execução de algumas tarefas específicas de modems no dispositivo serial, tais como desconectar a linha antes e depois de uma chamada. Caso este parâmetro não seja usado, `pppd` não monitorará as portas da linha DCD e não detectará finais não esperados da ligação remota.

Os exemplos apresentados acima foram bastante simples, porém o programa `chat` permite a utilização de programas muito mais complexos. Uma funcionalidade bastante útil é a possibilidade de especificar expressões que finalizem o programa com erro. Típicas finalizações desta ordem são mensagens como `BUSY`, ou `NO CARRIER`, que o modem gera quando o número indicado está ocupado e quando não há tom de linha, respectivamente. Para que o programa `chat` reconheça estas situações imediatamente, o que é melhor que uma saída por ter-se atingido o tempo máximo de espera, pode-se especificar estas expressões no início do programa, utilizando-se o parâmetro `ABORT`:

```
$ chat -v ABORT BUSY ABORT 'NO CARRIER' '' ATZ OK ...
```

De uma forma similar, pode-se alterar o tempo de espera para partes específicas do programa, inserindo-se o parâmetro `TIMEOUT`.

Para maiores detalhes, por favor verifique a página de manual do programa `chat(8)`.

Algumas vezes, é necessário inserir condições na execução de partes do programa de conversação. Por exemplo, quando não se recebe a linha de indicação para acesso ao sistema remoto (login:), pode-se enviar uma mensagem de conteúdo igual a BREAK, ou um comando de retorno de cursor ao início da linha. Isso pode ser obtido através da inserção de um subprograma à uma expressão esperada. Ele consiste em uma seqüência de expressões a serem enviadas e recebidas, assim como todo o programa, porém separados por hífens. O subprograma é executado toda vez que a expressão esperada não é recebida dentro do tempo previsto. No exemplo abaixo podemos modificar o programa para o seguinte formato:

```
ogin:-BREAK-ogin: ppp ssword: GaGariN
```

Agora, quando o programa `chat` não receber o indicativo de acesso ao sistema remoto, o subprograma será executado inicialmente enviando um BREAK, e após aguardará pelo indicativo de acesso novamente. Caso este seja recebido, o programa continuará da forma usual, caso contrário ele será terminado com erro.

8.6 Depurando a Configuração do PPP

Por padrão, o programa `pppd` irá registrar qualquer mensagem de erro através das funcionalidades disponíveis no servidor `syslogd`. Pode-se editar o arquivo de configuração, denominado `syslog.conf` para redirecionar as mensagens para um arquivo, ou mesmo para a console, pois de outra forma o programa `syslog` simplesmente descartará estas mensagens. A seguinte configuração envia todas as mensagens para o arquivo `/var/log/ppp-hist`:

```
daemon.*                /var/log/ppp-hist
```

Caso a configuração PPP não funcione logo à primeira tentativa, verificar o conteúdo deste arquivo pode fornecer indicativos muito úteis sobre o que pode estar acontecendo de errado. Caso isso não ajude, pode-se ainda adicionar informações extras ao conteúdo daquele arquivo através da opção `debug`. Isso faz com que o programa `pppd` registre o conteúdo de todos os pacotes de controle enviados e recebidos através do `syslog`.

Finalmente, a forma mais drástica de depuração é habilitar o histórico ao nível de kernel do sistema, através da execução do programa `pppd` com a opção `kdebug`. Ela é seguida por um argumento numérico que é a soma de bits dos seguintes valores: 1 para mensagens gerais de depuração, 2 para a impressão do conteúdo de todos os pacotes HDLC recebidos e 4 para que o programa de controle imprima todos os pacotes HDLC enviados. Para capturar as mensagens de depuração do kernel, deve-se executar o servidor de mensagens do sistema, chamado `syslogd`, o qual lê o conteúdo do arquivo `/proc/kmsg` ou o servidor `klogd`. Qualquer um deles direciona as mensagens de depuração do kernel para o programa `syslog`.

8.7 Opções de Configuração IP

O protocolo IPCP é usado para negociar alguns parâmetros do IP durante a configuração da conexão. Normalmente cada ponto da conexão envia um pacote de Requisição de Configuração IPCP, indicando os parâmetros padrão que devem ser alterados e qual o seu valor. Após recebê-lo, o sistema remoto inspeciona cada opção e responde aceitando-as ou rejeitando-as.

O programa `pppd` possibilita uma série de opções IPCP que poderão ser negociadas. Isso pode ser configurado através das opções de linha de comando que são discutidas a seguir.

8.7.1 Escolhendo Um Endereço IP

No exemplo anterior, o programa `pppd` discou para a máquina `itaim` e estabeleceu uma conexão IP. Nenhuma providência foi tomada no sentido de escolher um endereço IP em particular em nenhuma das pontas da conexão. Ao invés disso, utilizamos o endereço da máquina `aracaju` como o endereço IP local e deixamos `itaim` providenciar o seu. Algumas vezes, porém, pode ser útil ter-se o controle sobre quais endereços são usados em uma ou em ambas as pontas da conexão. O programa `pppd` suporta diversas variações deste tema.

Para solicitar um endereço em particular, deve-se geralmente fornecer ao programa `pppd` a seguinte opção:

```
end_local:end_remoto
```

onde *`end_local`* e *`end_remoto`* devem ser especificados na notação decimal IP ou como nomes de máquinas.[15] Com isso, o programa `pppd` tenta usar o primeiro endereço como o seu próprio e o segundo na máquina remota. Caso a outra ponta de linha rejeite algum deles durante a negociação IPCP, nenhuma conexão IP será estabelecida.[16] Caso se deseje configurar somente o endereço local e aceitar qualquer endereço que a máquina remota utilize, basta simplesmente não informar o parâmetro *`remote_addr`*. Por exemplo, para fazer com que a máquina `aracaju`

[15] Usar nomes de máquinas nesta opção traz conseqüências no uso da autenticação CHAP. Veja na seção específica a seguir maiores detalhes.

[16] Pode-se permitir que o sistema remoto substitua as configurações de endereços IP, fornecendo ao programa `pppd` a opção `ipcp-accept-local` e/ou `ipcp-accept-remote`. Por favor verifique a página de manual para maiores informações.

utilize o endereço IP `200.255.203.92` ao invés de seu próprio, basta informá-lo na linha de comando, no seguinte formato `200.255.203.92:`. De forma similar, para configurar somente o endereço remoto, basta deixar o campo *`end_local`* em branco. Por padrão, `pppd` usará o endereço associado ao nome da máquina.

Alguns servidores PPP que lidem com muitos clientes (como por exemplo um Provedor de Acesso a Internet) devem definir o endereço do sistema remoto de forma dinâmica: os endereços são definidos para o cliente somente enquanto este estiver conectado e serão liberados imediatamente após a desconexão. Ao discar para um servidor deste tipo, deve-se estar seguro que o programa `pppd` não requer qualquer endereço IP em particular, mas sim, que está apto a aceitar o fornecido pelo servidor. Isso significa que a opção *`end_local`* não pode ser especificada. Adicionalmente, deve-se usar a opção `noipdefault`, a qual faz com que o programa `pppd` espere pelo endereço IP fornecido pelo sistema remoto ao invés de utilizar o IP da máquina local.

Sob certas circunstâncias, pode ser desejável desligar a negociação de endereços IP totalmente e basear-se somente nos endereços especificados pela linha de comando. Isso pode ser atingido fornecendo-se a opção `-ip` para o programa `pppd`.

8.7.2 Roteamento Através de Uma Conexão PPP

Após configurar uma interface de rede, `pppd` irá configurar uma rota para a máquina que está na outra ponta da conexão. Caso a máquina remota esteja conectada a uma rede (a Internet por exemplo), certamente será desejável conectar-se às máquinas que estão "atrás" da máquina remota. Por isso, uma rota de rede deve ser configurada.

Vimos anteriormente que o programa `pppd` pode configurar uma rota padrão usando a opção `defaultroute`. Esta opção é muito útil caso o servidor PPP para o qual se discou seja um caminho para a Internet.

No caso inverso, onde o sistema local age como um caminho para uma única máquina o caso é relativamente simples. Por exemplo, digamos que algum funcionário da Cervejaria Virtual tem uma máquina em sua casa chamada `angra`. Ao se conectar à máquina `aracaju` através do PPP, ele usa um endereço da sub-rede da Cervejaria Virtual. Em `aracaju`, podemos informar a opção `proxyarp` ao programa `pppd`, o qual instalará uma entrada em seu proxy ARP para `angra`. Isso tornará `angra` acessível a qualquer outra máquina da Cervejaria Virtual ou da Vinícola Virtual.

De qualquer forma, as coisas nem sempre são tão simples como parecem, como por exemplo ao se conectar duas redes locais. Isso normalmente requer a adição de rotas de redes específicas, uma vez que estas redes normalmente já têm as suas rotas padrão. Além disso, utilizando ambas as pontas da conexão PPP como rota padrão pode gerar um círculo interminável de idas e vindas, onde os pacotes que não conhecem o seu destino ficarão na conexão até que o seu tempo de validade expire.

Como exemplo, suponhamos que a Cervejaria Virtual abriu uma filial em Goiânia. A subsidiária tem uma rede Ethernet própria usando o endereçamento IP de rede `191.72.3.0`, o qual é a sub-rede 3 da rede classe B da Cervejaria Virtual. Eles desejam conectar-se à rede Ethernet principal da Cervejaria através de uma ligação PPP para atualização de bases de dados de clientes, etc.. Novamente, `aracaju` agirá como caminho padrão e a máquina remota será chamada `pirenopolis` com um endereço IP igual a `191.72.3.1`.

Quando `pirenopolis` se conecta a `aracaju`, ele definirá uma rota padrão apontando para `aracaju` da maneira usual. Na máquina `aracaju` porém, teremos que instalar uma rota de rede para a sub-rede 3, passando por `pirenopolis`. Para tanto, utilizaremos uma funcionalidade do programa `pppd` não discutida até aqui, chamada `ip-up`. Este é um programa localizado em `/etc/ppp` que é executado após a interface PPP ser configurada. É acionado da seguinte forma e com os seguintes parâmetros:

```
ip-up interface dispostivo velocidade end_local end_remoto
```

onde ***`interface`*** é o nome da interface de rede usada (p.ex.ppp0), ***`dispositivo`*** é o caminho do arquivo do dispositivo serial usado (`/dev/tty` caso stdin/stdout seja usado) e ***`velocidade`*** é a velocidade do dispositivo. *`end_local`* e *`end_remoto`* fornecem o endereço IP usado em ambas as pontas da conexão no formato decimal do endereço IP. No nosso caso, o programa `ip-up` pode conter os seguintes parâmetros:

```
#!/bin/sh
case $5 in
191.72.3.1)           # esta é a máquina pirenopolis
        route add -net 191.72.3.0 gw 191.72.3.1;;
...
esac
exit 0
```

De uma forma similar, o programa `/etc/ppp/ip-down` é usado para desfazer todas as ações do programa `ip-up` após a conexão IP ter sido desfeita.

De qualquer forma, o sistema de roteamento ainda não está completo. Devemos configurar as entradas de roteamento nas tabela de ambas as máquinas PPP, porque até o momento as demais máquinas de ambas as redes não sabem da existência umas das outras. Este não é um grande problema se todas as máquinas da subsidiária têm a rota padrão apontando para `pirenopolis` e todas as máquinas da Cervejaria Virtual apontam da mesma forma para `aracaju`. Caso este não seja o caso, a única opção será utilizar um servidor de roteamento como `gated`. Após criar a rota de rede em `aracaju`, o servidor de roteamento irá propagar a nova rota para todas as máquinas das sub-redes.

8.8 Opções de Controle de Conexão

Vimos anteriormente o LCP, o Protocolo de Controle de Conexão, o qual é usado para negociar as características da conexão e testar o seu funcionamento.

As duas mais importantes opções que podem ser negociadas pelo LCP são a unidade máxima de recepção e o Mapa de Caracteres de Controle Assíncrono. Há um número razoável de opções de configuração do LCP, mas não estão dentro do escopo deste guia. De qualquer forma a RFC 1548 traz uma descrição detalhada do LCP e de suas opções.

O Mapa de Caracteres de Controle Assíncrono, coloquialmente chamada de "mapa assíncrono", é usado em conexões assíncronas, como por exemplo em linhas discadas, para identificar o conjunto de caracteres que devem ser substituídos (por uma seqüência específica de dois outros caracteres) e interpretados como caracteres normais de texto e não como uma seqüência de comando, indicando alguma ação a ser tomada. Por exemplo, deve-se evitar o uso dos caracteres XON e XOFF, usado por softwares de negociação, porque algum modem mal configurado pode desconectar-se após receber um XOFF. Outros possíveis candidatos incluem `Ctrl-]` (o caractere de fuga do programa `telnet`). PPP permite substituir qualquer caractere com código ASCII entre 0 e 31 através da utilização do "mapa assíncrono".

O "mapa assíncrono" é um mapa de 32 bits de tamanho, com o bit menos significativo indicando o caractere ASCII nulo, e o mais significativo correspondendo ao caractere de código ASCII 31. Caso um bit esteja configurado, isso significa que o caractere correspondente àquela posição deve ser substituído antes de ser enviado através da conexão. Inicialmente, o mapa assíncrono está configurado como `0xffffffff`, ou seja com todos os bits configurados, indicando que todos os caracteres devem ser substituídos.

Para informar à máquina remota que ela não deve substituir todos os caracteres de controle, mas somente alguns deles, pode-se especificar um novo mapa assíncrono para o programa `pppd` através da opção `asyncmap`. Por exemplo, caso somente os caracteres `^S` e `^Q` (ASCII 17 e 19), comumente usados como XON e XOFF devam ser substituídos, deve ser usada a seguinte opção:

```
asyncmap 0x000A0000
```

A Unidade Máxima de Recepção, ou MRU, indica para a máquina remota qual o tamanho máximo da unidade de transferência HDLC que se deseja receber. Apesar de lembrar o conceito de MTU - Unidade Máxima de Transferência, há pouca coisa em comum entre eles. A MTU é um parâmetro do kernel para dispositivos de rede e descreve o tamanho máximo de pacotes que a interface pode manusear. Já a MRU é como um aviso à máquina remota para não gerar pacotes com tamanho maiores que o definidos, a interface nunca será capaz de receber pacotes maiores que 1500 bytes.

Escolher uma MRU não é somente uma questão da capacidade de transferência, mas sim da definição de qual opção propicia a melhor performance. Caso se pretenda utilizar aplicações interativas sobre a conexão, configurar uma MRU ao redor de 296 pode ser uma boa idéia, uma vez que o uso de pacotes maiores (digamos de uma sessão FTP) fará o cursor pular. Para informar ao programa `pppd` que este deve solicitar uma MRU de, por exemplo 296, basta indicar a opção `mru 296`. MRUs muito pequenas somente fazem sentido se não se estiver utilizando a compressão de cabeçalho VJ (a qual é habilitada por padrão).

O programa `pppd` entende algumas opções do protocolo LCP que configuram o comportamento geral do processo de negociação, como por exemplo o número máximo de solicitações de configuração que podem ser trocadas antes que a conexão seja interrompida. A menos que se saiba exatamente o que se está fazendo, deve-se usar os parâmetros padrão.

Finalmente há duas opções que se aplicam às mensagens de eco do LCP. PPP as define como Solicitação de Eco e Resposta de Eco. O programa `pppd` usa estas funcionalidades para verificar se a conexão ainda está ativa. Pode-se habilitar isto usando a opção `lcp-echo-interval` junto com o tempo desejado expresso em segundos. Caso nenhum pacote seja recebido da máquina remota no intervalo definido, o programa `pppd` gerará uma Solicitação de Eco e esperará receber da máquina remota uma Resposta de Eco. Caso a resposta não seja recebida, a conexão será interrompida após o envio de um certo número de requisições. Este número pode ser configurado através da opção `lcp-echo-failure`. Por padrão estas funcionalidades não são habilitadas.

8.9 Considerações Gerais de Segurança

Um servidor PPP mal configurado pode tornar-se um problema devastador na segurança. Ele pode permitir que qualquer um conecte-se a uma máquina e entre na rede Ethernet. Nesta seção, discutiremos algumas medidas que podem dar maior segurança à configuração do PPP.

Um problema com o `pppd` é que para configurar os serviços de rede e a tabela de roteamento, ele requer privilégios de `superusuário`. Isso pode ser resolvido através da execução do comando setuid `root`. De qualquer forma o programa `pppd` permite a configuração de diversas opções de segurança importantes. Para proteger-se de ataques é aconselhável que estas opções estejam devidamente configuradas, de acordo com os padrões sugeridos tanto para o arquivo global `/etc/ppp/options`, quanto para aqueles apresentados no exemplo da seção Usando Arquivos de Opções. Alguns deles, como as opções de autenticação, não podem ser substituídas pelo usuário, criando assim uma proteção razoável contra intrusos.

Evidentemente deve-se proteger o sistema local contra os sistemas com os quais se estabelecem conexões PPP também. Para evitar que as máquinas conectem-se à rede da forma como bem entendam, deve-se habilitar algum tipo de autenticação para as máquinas remotas. Adicionalmente, não se deve permitir que máquinas externas usem o endereço IP que elas escolham, mas sim estes devem ser restritos a somente alguns predefinidos. A seção seguinte lida com estes tópicos.

8.10 Autenticação Com PPP

8.10.1 CHAP versus PAP

Com o PPP, cada sistema pode solicitar que a outra ponta da conexão se autentique utilizando um dos dois protocolos de verificação, conhecidos como Protocolo de

Autenticação de Senha - PAP[17] e Protocolo de Autenticação de Apresentação[18]. Quando uma conexão é estabelecida cada ponta requisita que a outra se autentique, independente de quem está discando para quem. A seguir usaremos os termos cliente ou servidor para fazer distinção entre o sistema que envia o pedido de autenticação e o sistema autenticador respectivamente. Um servidor PPP pode solicitar a identificação da máquina remota através do envio de uma requisição de configuração de identificação LCP solicitando que o sistema se autentique de acordo com o protocolo especificado (PAP ou CHAP).

O protocolo PAP trabalha basicamente da mesma forma que um procedimento de acesso normal. O cliente se autentica enviando um nome de usuário e uma senha, que opcionalmente pode estar encriptada, a qual é comparada com a base de senhas secretas do servidor. Esta técnica é vulnerável contra intrusos que tenham condições de verificar todo o tráfego corrente na linha serial, e consigam capturar um usuário válido e a sua senha, ou contra tentativas de "adivinhação" de senhas através do método de tentativas e erros.

CHAP por sua vez não possui essas deficiências. Com o CHAP, o servidor envia para o cliente, uma expressão aleatória, contendo um "desafio", em conjunto com o seu nome de máquina. O cliente utiliza o nome de máquina para buscar a chave da solução, combina com a expressão aleatória e encripta o resultado usando uma função numérica que não pode ser revertida. O resultado é enviado para o servidor que executa a mesma tarefa e compara os resultados. Caso sejam idênticos o cliente é considerado autêntico.

Outra funcionalidade do CHAP reside no fato dele não requerer que o cliente seja autenticado somente no momento do estabelecimento da conexão, mas envia a intervalos regulares novos "desafios" para estar seguro que o cliente não foi substituído por um intruso neste meio tempo, somente trocando as linhas telefônicas.

O programa `pppd` mantém as chaves de solução para PAP e CHAP em diferentes arquivos denominados `/etc/ppp/chap-secrets` e `pap-secrets`, respectivamente. Ao configurar uma máquina remota em um ou outro arquivo, passa-se a ter um controle estrito do CHAP e PAP durante a comunicação, garantindo-se tanto a autenticidade do servidor como a do cliente.

Por padrão, o programa `pppd` não solicita autenticação da máquina remota, mas concordará em se autenticar caso solicitado. Como o CHAP é muito mais robusto que o PAP, o programa `pppd` tenta usá-lo sempre que possível. Caso ele não

[17] Password Authentication Protocol
[18] Challenge Handshake Authentication Protocol

consiga encontrar a chave de solução do "desafio" no seu arquivo `chap-secrets`, ele tentará usar as chaves PAP. Caso esta também não esteja presente ele irá recusar a autenticação e a conexão será interrompida.

Este comportamento pode ser modificado de diversas formas. Por exemplo, quando for fornecida a opção `auth`, o programa `pppd` solicitará a autorização da outra ponta da ligação. O programa `pppd` suportará o uso do CHAP ou PAP para isso, assim como ele utilizará o arquivo de soluções para uso com CHAP ou PAP, respectivamente. Há outras opções para ativar ou desativar um protocolo de autenticação, mas estas não serão descritas aqui. Por favor verifique a página de manual do programa `pppd(8)` para maiores detalhes.

Caso todos os sistemas com os quais haja conexões PPP concordem com os sistemas de autenticação utilizados pela rede local, então será possível utilizar o parâmetro `auth` no arquivo de opções globais `/etc/ppp/options` e definir senhas para cada sistema no arquivo `chap-secrets`. Se um sistema não suporta CHAP, deve ser adicionada uma entrada para ele no arquivo `pap-secrets`. Desta forma, pode-se estar seguro de que nenhum sistema não autorizado poderá estabelecer uma conexão com a sua rede.

As próximas duas seções discutem os dois arquivos de soluções secretas usadas pelo PPP, denominados `pap-secrets` e `chap-secrets`. Ambos estão localizados em `/etc/ppp` e contêm conjuntos de clientes, servidores e senhas, opcionalmente seguidos de endereços IP. A interpretação dos campos cliente e servidor é diferente para os protocolos PAP e CHAP, e depende se a autenticação é feita pela própria máquina no servidor remoto, ou, ao contrário, se o servidor remoto se autenticará na máquina local.

8.10.2 O Arquivo de Segredos do CHAP

Quando não é possível autenticar a si mesmo em algum servidor CHAP, o programa `pppd` pesquisa o arquivo `chap-secrets` buscando uma entrada com o campo cliente igual ao nome da máquina local e com o campo servidor igual ao nome da máquina remota igual à que tenha enviado o "desafio". Ao requerer da outra ponta que se autentique, as regras são simplesmente invertidas: o programa `pppd` procurará por uma entrada aonde o cliente seja igual ao nome da máquina remota (enviado na resposta do cliente CHAP) e onde o campo servidor seja igual ao nome de máquina local.

A seguir apresentamos um exemplo de um arquivo `chap-secrets` para a máquina aracaju:[19]

```
# Soluções secretas CHAP para aracaju.cvirtual.com.br
#
# cliente                  servidor             segredo              endereço
#-----------------------------------------------------------------------------
aracaju.cvirtual.com.br itaim.engenho.com.br  "ubatuba"  aracaju.cvirtual.com.br
itaim.engenho.com.br    aracaju.cvirtual.com.br "caragua"   itaim.engenho.com.br
*                       aracaju.cvirtual.com.br "guarapuava" jau.cvirtual.com.br
```

Ao estabelecer uma conexão com `itaim`, este solicitará a `aracaju` que se autentique usando CHAP e enviando um "desafio". O programa `pppd` pesquisará o arquivo `chap-secrets` por uma entrada onde o campo cliente seja igual a `aracaju.cvirtual.com.br` e o servidor seja igual a `itaim.engenho.com.br`[20], selecionando então a primeira linha do arquivo acima. É então produzida a resposta CHAP contendo o desafio e a resposta, que será então enviada para `itaim`.

Ao mesmo tempo, o programa `pppd` compõe um "desafio" CHAP para `itaim`, contendo uma expressão e o nome totalmente qualificado: `aracaju.cvirtual.com.br`. `itaim` constrói a resposta CHAP na maneira discutida anteriormente e a retorna para `aracaju`. O programa `pppd` extrai o nome da máquina cliente (`itaim.engenho.com.br`) a partir da resposta, pesquisa o arquivo `chap-secrets` por uma linha onde `itaim` seja o cliente e `aracaju` o servidor. A segunda linha é então selecionada, o programa `pppd` combina o "desafio" CHAP com a senha "caragua", encriptando-os e compara o resultado com a resposta CHAP da máquina `itaim`.

O quarto campo opcional lista os endereços IP que são aceitos para os clientes nomeados na primeira coluna. Os endereços podem ser fornecidos no formato decimal, separado por pontos ou como nomes de máquinas, as quais são pesquisadas pelo resolvedor de nomes. Por exemplo, se `itaim` solicita um endereço IP durante uma negociação IPCP que não esteja na lista, o pedido será rejeitado, e IPCP será finalizado. No arquivo de exemplo apresentado acima, `itaim` está limitado a usar o seu próprio endereço IP. Caso o campo de endereço esteja vazio, qualquer endereço IP será aceito.

A terceira linha do arquivo de exemplo `chap-secrets` permite que qualquer máquina estabeleça uma conexão PPP com `aracaju`, uma vez que o campo cliente contém um `*`, que significa "qualquer nome". O único requisito é que ele conheça o segredo (guarapuava) e use o endereço de `jau.cvirtual.com.br`. O caractere de generalização (`*`) pode aparecer em qualquer ponto do arquivo de segredos, uma

[19] As aspas duplas não são parte do campo senha, elas simplesmente servem para proteger espaços em branco no campo senha.

[20] Este nome de máquina é retirado do "desafio" CHAP.

vez que o programa pppd irá sempre utilizar a informação mais específica que se aplicar a um par cliente/servidor.

Há ainda algumas informações que precisam ser ditas sobre a forma como pppd pesquisa o arquivo de segredos. Como explicado anteriormente, o nome da máquina remota é sempre enviado pela outra ponta da conexão. O nome de máquina local será derivado de uma chamada à função gethostname(2). Caso o sistema local esteja configurado com um nome de máquina não qualificado, então o programa pppd deverá ser informado do nome do domínio local através da opção domain:

```
# pppd ...domain cvirtual.com.br
```

Esta medida irá anexar o nome de domínio da Cervejaria Virtual ao nome da máquina aracaju para todas as atividades relacionadas com autenticações. Outras opções podem modificar o conceito do nome da máquina local para o programa pppd, como por exemplo as opções usehostname e name. Ao se fornecer o endereço IP na linha de comando utilizando-se as opções "*local*:varremoto" e *local* é um nome ao invés de uma notação decimal do endereço IP, pppd irá usar este nome ao invés do nome da máquina local. Para maiores detalhes, por favor consulta a página de manual do programa pppd(8).

8.10.3 O Arquivo de Segredos do PAP

O arquivo de segredos do PAP é muito similares ao usado pelo CHAP. Os primeiros dois campos contêm um nome de usuário e o nome do servidor, o terceiro contém o segredo PAP. Quando uma máquina remota solicita autenticação, o programa pppd utiliza uma entrada do arquivo que contenha o nome de servidor igual ao nome da máquina local e um campo de usuário igual ao nome enviado na requisição. Ao solicitar uma autenticação em um servidor remoto, o programa pppd escolhe um "segredo" a ser enviado, a partir da linha onde o campo usuário contém o nome da máquina local e o campo servidor tenha o nome da outra ponta da conexão.

A seguir apresentamos um arquivo de exemplo:

```
# /etc/ppp/pap-secrets
#
# usuário      servidor   segredo       endereços
aracaju-pap    itaim      caragua       aracaju.cvirtual.com.br
itaim          aracaju    guarapuava    itaim.engenho.com.br
```

A primeira linha é usada para autenticar a máquina local ao conectar-se à `itaim`. A segunda linha descreve como o usuário de nome `itaim` deve autenticar-se na máquina local.

O nome `aracaju-pap` na primeira coluna é o nome de usuário enviado para a máquina `itaim`. Por padrão, `pppd` irá escolher o nome da máquina local como nome do usuário, mas é possível especificar um nome diferente através da opção `user` seguida pelo nome desejado.

Ao escolher uma entrada no arquivo `pap-secrets` para autenticação com a máquina remota, o programa `pppd` deve conhecer o seu nome. Como não há forma dele descobrir, é necessário especificar na linha de comando, utilizando a opção `remotename` seguida do nome de conexão da máquina local. Por exemplo, para utilizar a definição acima para autenticação na máquina `itaim`, temos que adicionar a seguinte opção à linha de comando:

```
# pppd ... remotename itaim user aracaju-pap
```

No quarto campo (e todos os campos seguintes), é possível especificar quais os endereços IP que são permitidos para aquela máquina em particular, da mesma forma que nos arquivos de segredos CHAP. A máquina remota somente poderá solicitar um endereço daquela lista. No arquivo de exemplo, solicitamos que `itaim` seja conhecida pelo seu endereço IP real.

Note que PAP é um método de autenticação muito menos eficiente, sendo o CHAP o mais indicado. Desta forma não o apresentaremos detalhadamente neste Guia. Caso você deseje maiores informações, verifique a página de manual do programa `pppd(8)`.

8.11 Configurando um Servidor PPP

Disponibilizar um servidor `pppd` é somente uma questão de adicionar as opções adequadas à linha de comando. A forma ideal é criar uma conta especial, chamada digamos `ppp` e configurar um programa especial que acione o servidor `pppd` com determinadas opções, o qual é executado automaticamente toda vez que o usuário se conecte ao sistema. Pode-se por exemplo adicionar a seguinte linha ao arquivo `/etc/passwd`:

```
ppp:*:500:200:Conta PPP pública:/tmp:/etc/ppp/ppplogin
```

Obviamente pode-se usar diferentes identificações de usuário e grupo (uid e gid) das mostradas acima. Pode-se ainda configurar o campo senha para a conta ppp utilizando o comando `passwd`.

O programa `ppplogin` poderá ter o seguinte formato:

```
#!/bin/sh
# ppplogin - programa usado para acionar o pppd no acesso do cliente
mesg n
stty -echo
exec pppd -detach silent modem crtscts
```

O comando `mesg` não permite que outros usuários escrevam no terminal tty, usando por exemplo o comando `write`. O comando `stty` desabilita o eco de caracteres. Esta opção é necessária pois de outra forma tudo o que for enviado através da conexão será ecoado de volta. A opção mais importante do programa `pppd` é a `-detach`, porque evita que o programa seja colocado no modo de execução de segundo plano. A opção `silent` faz com que `pppd` aguarde até receber um pacote do sistema cliente antes que inicie a transmissão de dados. Isso evita a ultrapassagem dos tempos de espera definidos quando o sistema cliente tornar-se muito lento ao iniciar o seu PPP. A opção `modem` faz com que `pppd` administre o controle do modem na linha serial, verificando se a conexão não foi desfeita. Esta opção deve ser sempre utilizada ao se usar o `pppd` através de linhas discadas. Finalmente a opção `crtscts` habilita a negociação de conexão por hardware.

Além dessas opções, pode-se ainda forçar a autenticação dos clientes, por exemplo ao se especificar a opção `auth` na linha de comando do programa `pppd` ou no arquivo de opções globais do sistema. A página de manual discute alguns aspectos mais específicos de ativação e inibição dos protocolos de autenticação.

Capítulo 9

Importantes Funcionalidades de Rede

Após configurar com sucesso o endereço IP e o resolvedor de nomes, deve-se configurar os serviços que serão disponibilizados através da rede. Este capítulo cobre a configuração de algumas aplicações importantes de rede, inclusive o servidor `inetd` e os programas da família `rlogin`. A interface RPC - Chamada de Procedimentos Remotos[1], na qual serviços como o Sistema de Arquivos em Rede - NFS e Sistemas de Informações em Rede - NIS estão baseados, será também descrita. A configuração do NFS e NIS será descrita em capítulo à parte, assim como aplicações como correio eletrônico e sistemas de notícias.

Obviamente, não podemos cobrir todas as aplicações de rede neste Guia. Caso se instale algumas das aplicações não discutidas aqui, como `talk`, `gopher`, ou `Xmosaic`, por favor verifique atentamente as instruções que acompanham esses programas.

9.1 O Superservidor `inetd`

Freqüentemente, serviços são executados por programas denominados *servidores*. Um servidor é um programa que abre uma determinada porta e fica aguardando por solicitações de conexão. Quando uma solicitação é recebida, ele cria um

[1] Remote Procedure Call

processo filho, o qual trata aquela conexão específica, enquanto o processo pai continua a escutar na porta aguardando novas solicitações. Este conceito sintetiza, na sua essência e de maneira simplificada, a forma como os serviços são oferecidos em uma máquina Linux, ou seja um servidor escutando em uma porta, aguardando pedidos de conexão, o que geralmente pode significar a perda de alguns recursos de sistema, como por exemplo a área de troca.

Porém, praticamente todo sistema `Unix` executa uma espécie de superservidor, o qual é capaz de criar conectores para uma série de serviços e ouvir todas as portas simultaneamente, utilizando para tanto uma chamada ao sistema denominada `select(2)`. Quando uma máquina remota solicita algum de seus serviços, o superservidor percebe o fato e aciona o servidor específico da porta envolvida.

O superservidor normalmente utilizado é conhecido por `inetd`, o Servidor Internet[2]. Ele inicia a sua execução quando o sistema é inicializado, recebendo a lista de serviços a serem monitorados a partir de uma arquivo denominado `/etc/inetd.conf`. Além dos serviços que envolvem outros servidores, existe uma série de serviços simples que são executados pelo próprio `inetd` denominados *serviços internos*. Eles incluem a função `chargen`, a qual simplesmente gera uma cadeia de caracteres e a função `daytime`, a qual retorna o conceito do sistema da hora do dia.

Um registro neste arquivo consiste de uma linha simples composta pelos seguintes campos:

```
serviço tipo protocolo espera usuário servidor linha_de_comando
```

O significado dos campos é o seguinte:

`serviço` Fornece o nome do serviço a ser disponibilizado. Ele deve ser traduzido em um número de porta através de uma pesquisa no arquivo `/etc/services`. Este arquivo será descrito na seção Os Arquivos `services` e `protocols`, a seguir.

`tipo` Especifica o tipo de conexão que será utilizada, `stream` (para conexões orientadas a protocolo) ou `dgram` (para protocolos que utilizem datagramas). Serviços baseados em TCP devem sempre ser especificados como `stream`, enquanto que serviços baseados em UDP devem sempre ser definidos como `dgram`.

[2]Internet Daemon

`protocolo` Especifica o nome do protocolo usado pelo serviço. Deve ser um nome válido que possa ser encontrado no arquivo `protocols`, também explicado a seguir.

`espera` Esta opção aplica-se somente a conexões por datagramas. Ela pode ser igual a `wait` ou `nowait`. Caso `wait` seja especificado, `inetd` irá executar somente um servidor por vez para a porta especificada. De outra forma, ele imediatamente voltará a ouvir a porta após atender a uma requisição. Isso é útil para servidores "mono-executáveis" que necessitam ler todos os datagramas até que mais nenhum seja enviado, e então finaliza. Muitos servidores RPC são deste tipo e devem ter esta configuração. O tipo oposto, os servidores "multi-executáveis" permitem um número ilimitado de instâncias do programa sendo executadas concorrentemente. Estes são utilizados mais raramente. Estes servidores devem receber o parâmetro `nowait`.

Conexões `stream` devem sempre usar o parâmetro `nowait`.

`usuário` Esta é a identificação de acesso do usuário sob o qual o processo será executado. Como freqüência ele será igual ao superusuário `root`, porém alguns serviços podem utilizar contas diferentes. É aconselhável aplicar os princípios de uso do usuário menos privilegiado, o que significa que não se deve executar um comando com uma conta com privilégios que não sejam realmente necessários na sua execução. Por exemplo, o servidor de notícias NNTP será executado com o usuário `news`, enquanto serviços que podem produzir riscos de segurança, como por exemplo o `tftp` ou `finger`, são normalmente executados com o usuário `nobody`.

`servidor` Fornece o caminho completo do programa servidor a ser utilizado. Serviços internos terão um valor igual a `internal` neste campo.

`linha_de_comando` Esta é a linha de comando a ser enviada para o servidor. Isso inclui o argumento 0, que é o nome do comando. Normalmente conterá o nome do programa servidor, a menos que o programa comporte-se diferentemente quando acionado com um nome diferente. Este campo não deverá conter nenhuma informação para serviços internos.

```
 Um exemplo do arquivo  \file{/etc/inetd.conf}.

#
# serviços inetd
ftp       stream tcp nowait root   /usr/sbin/ftpd    in.ftpd -l
```

```
telnet    stream tcp nowait root   /usr/sbin/telnetd in.telnetd -b/etc/issue
#finger   stream tcp nowait bin    /usr/sbin/fingerd in.fingerd
#tftp     dgram  udp wait   nobody /usr/sbin/tftpd   in.tftpd
#tftp     dgram  udp wait   nobody /usr/sbin/tftpd   in.tftpd /boot/diskless
login     stream tcp nowait root   /usr/sbin/rlogind in.rlogind
shell     stream tcp nowait root   /usr/sbin/rshd    in.rshd
exec      stream tcp nowait root   /usr/sbin/rexecd  in.rexecd
#
# serviços internos inetd
#
daytime   stream tcp nowait root internal
daytime   dgram  udp nowait root internal
time      stream tcp nowait root internal
time      dgram  udp nowait root internal
echo      stream tcp nowait root internal
echo      dgram  udp nowait root internal
discard   stream tcp nowait root internal
discard   dgram  udp nowait root internal
chargen   stream tcp nowait root internal
chargen   dgram  udp nowait root internal
```

Um exemplo de um arquivo `inetd.conf` foi mostrado anteriormente. O serviço `finger` está comentado (contém um caractere # no início da linha), portanto não está disponível. Isso é feito com freqüência, por motivos de segurança, uma vez que ele pode ser usado por intrusos para obter nomes válidos de usuários do sistema local.

O programa `tftp` também é mostrado de forma comentada. `tftp` implementa o Protocolo Primitivo de Transferência de Arquivos[3], o qual permite transferir qualquer arquivo legível do sistema local sem a necessidade de verificação de senhas. Esse serviço é especialmente perigoso quando utilizado com o arquivo `/etc/passwd`, o qual pode conter as senhas de todos os usuários do sistema, mesmo quando se esteja utilizando o sistema de senhas sombra.

TFTP é comumente utilizado em estações sem disco rígido ou terminais X para transferência de seu código a partir de um servidor de inicialização. Caso se necessite do serviço `tftpd` para esta finalidade, deve-se estar seguro de limitar o seu escopo para os diretórios onde os clientes têm seus arquivos, através da adição de seu nome à linha de comando de acionamento do servidor `tftpd`. Isso é mostrado na segunda linha do exemplo anterior.

[3] Primitive File Transfer Protocol

9.2 A Funcionalidade tcpd de Controle de Acesso

Abrir um computador para acesso pela rede envolve muitos riscos de segurança. Algumas aplicações foram escritas para proteger o sistema contra diversos tipos de ataques. Alguns desses porém podem ser bastante frágeis, ou não conseguem distinguir entre máquinas seguras a partir das quais a requisição de um determinado serviço será aceito e máquinas inseguras cujas solicitações serão rejeitadas. Já comentamos rapidamente os serviços `finger` e `tftp` acima. Deve-se limitar o acesso a estes serviços somente a máquinas confiáveis, o que é impossível com a configuração usual, pois o programa `inetd` disponibiliza um serviço a todos os clientes ou a nenhum.

Uma ferramenta útil nestes casos é o servidor `tcpd`,[4] um servidor de observação. Para os serviços TCP que se deseje monitorar ou proteger, ele deverá ser acionado ao invés do programa servidor. O programa `tcpd` registra todas as requisições através do servidor de mensagens do sistema denominado `syslog`, verifica se a máquina remota tem permissão de usar este serviço e somente se a resposta for positiva executará o real servidor do serviço. Cabe ressaltar que esta funcionalidade não está disponível para serviços baseados em UDP.

Por exemplo, para observar o serviço `finger`, deve-se alterar a linha correspondente no arquivo `inetd.conf` para:

```
# servidor de observação do serviço finger
finger  stream  tcp    nowait  root   /usr/sbin/tcpd   in.fingerd
```

Sem a adição de qualquer controle de acesso, o cliente não perceberá qualquer diferença de um serviço `finger` usual, exceto pelo fato de que todas as requisições serão registradas pelo `syslog`.

O controle de acesso é implementado através de dois arquivos denominados `/etc/hosts.allow` e `/etc/hosts.deny`. Eles contêm informações que permitem e negam o acesso, respectivamente. Quando o servidor `tcpd` manuseia uma requisição para um serviço, como por exemplo uma chamada ao `finger` a partir de um cliente chamado `itabaiana.cvirtual.com.br`, ele pesquisa nos arquivos `hosts.allow` e `hosts.deny` (nesta ordem) buscando uma entrada que coincida com o serviço e com a máquina cliente. Caso uma entrada seja encontrada no arquivo `hosts.allow`, o acesso será permitido, independente de qualquer referência no arquivo `hosts.deny`. Caso alguma entrada seja encontrada no arquivo `hosts.deny`, a requisição será

[4] Escrito por Wietse Venema, `wietse@wzv.win.tue.nl`.

rejeitada e a conexão encerrada. Caso nenhuma entrada seja encontrada, a requisição será aceita.

Registros nos arquivos de acesso têm a seguinte aparência:

```
lista_de_serviços: lista_de_máquinas [:comando]
```

`lista_de_serviços` é uma relação de nomes de serviços existentes no arquivo `/etc/services` ou a palavra chave `ALL`. Para definir todos os serviços exceto `finger` e `tftp`, use "`ALL EXCEPT finger, tftp`".

`lista_de_máquinas` é uma lista de nomes de máquinas ou endereços IP, ou as palavras chave `ALL`, `LOCAL`, ou `UNKNOWN`. `ALL` significa qualquer máquina, enquanto `LOCAL` é utilizado para qualquer máquina cujo nome não contenha um ponto.[5] `UNKNOWN` significa qualquer máquina cujo nome ou endereço não seja localizado. Um nome começando com um ponto significa todas as máquinas de um determinado domínio. Por exemplo, `.cvirtual.com.br` faz com que a máquina `itabaiana.cvirtual.com.br` tenha acesso ao serviço. Há ainda o uso de endereços IP para redes e sub-redes. Por favor consulte a página de manual `hosts_access(5)` para maiores detalhes.

Para evitar o acesso aos serviços `finger` e `tftp` para todas as máquinas, exceto as máquinas locais, deve ser criado o arquivo `/etc/hosts.deny` com o seguinte conteúdo, deixando o arquivo `/etc/hosts.allow` vazio:

```
in.tftpd, in.fingerd:  ALL EXCEPT LOCAL, .seu.dominio
```

O campo opcional `comando` pode conter um comando que pode ser acionado quando a entrada coincidir. Isso é útil para configurar armadilhas que exponham intrusos em potencial:

```
in.ftpd: ALL EXCEPT LOCAL, .cvirtual.com.br :
      echo "origem da solicitação em %d@%h" >> /var/log/finger.log;
      if [ %h != "aracaju.cvirtual.com.br" ]; then
          finger -l @%h >> /var/log/finger.log
      fi
```

Os argumentos `%h` e `%d` são expandidos para o nome da máquina cliente e o nome do serviço pelo programa `tcpd`, respectivamente. Por favor consulte a página de manual do `hosts_access(5)` para maiores detalhes.

[5] Normalmente somente nomes de máquinas obtidos de pesquisas no arquivo `/etc/hosts` não contêm pontos.

9.3 Os Arquivos services e protocols

O número das portas de certos serviços padrão são definidos na RFC denominada Definindo Números. Para viabilizar que programas servidores ou clientes convertam os nomes de serviços para estes números, no mínimo parte desta lista deve ser mantida em cada máquina. Estas definições são armazenadas em um arquivo chamado `/etc/services`. Uma entrada neste arquivo tem o seguinte formato:

`serviços porta/protocolo [apelidos]`

O parâmetro *serviço* especifica o nome do serviço, *porta* define a porta onde o serviço é oferecido e *protocolo* define qual o protocolo de transporte a ser usado. Comumente, ele será `udp` ou `tcp`. É possível que um serviço seja oferecido por mais de um protocolo, assim como serviços diferentes podem ser oferecidos na mesma porta. O campo *apelidos* permite a especificação de nomes alternativos para o mesmo serviço.

Usualmente, não se deve alterar o arquivo de serviços que vem com o software de rede de seu sistema Linux. De qualquer forma, segue uma demonstração do conteúdo do arquivo.

```
# Arquivo de serviços:
#
# serviços bem conhecidos
echo          7/tcp                 # Eco
echo          7/udp                 #
discard       9/tcp  sink null      # Descartar
discard       9/udp  sink null      #
daytime      13/tcp                 # Hora do Dia
daytime      13/udp                 #
chargen      19/tcp  ttytst source  # Gerador de Caracteres
chargen      19/udp  ttytst source  #
ftp-data     20/tcp                 # Prot.de Transf. de Arquivos(Dados)
ftp          21/tcp                 # Prot.de Transf. de Arquivos(Controle)
telnet       23/tcp                 # Protocolo de Terminais Virtuais
smtp         25/tcp                 # Prot. Simples de Transf. de Mensagens
nntp        119/tcp  readnews       # Prot. de Transf. de Rede de Notícias
#
# serviços UNIX
exec        512/tcp                 # rexecd BSD
biff        512/udp  comsat         # notificação de correio
login       513/tcp                 # acesso remoto
who         513/udp  whod           # comandos remotos  who e uptime
shell       514/tcp  cmd            # comando remoto,sem senha
syslog      514/udp                 # sistema remoto de registro
printer     515/tcp  spooler        # sistema de impressão remota
route       520/udp  router routed  # protocolo de informações de roteamento
```

Note que o serviço `echo`, por exemplo, é oferecido na porta 7 por ambos os protocolos TCP e UDP, e que a porta 512 é usada para dois serviços denominados servidor COMSAT (o qual notifica o usuário da chegada de mensagens; veja por exemplo o programa `xbiff(1x)`), sobre UDP, e é usada para a execução de comandos remotos (`rexec(1)`), usando TCP.

Similar ao arquivo de serviços, a biblioteca de rede necessita traduzir nomes de protocolos, por exemplo aqueles usados no arquivo serviços, em números conhecidos pela camada IP de outras máquinas. Isso é feito através de uma pesquisa no arquivo `/etc/protocols`. Ele contém uma entrada por linha, constituída pelo nome do protocolo e o número associado. É muito improvável que este arquivo deva ser alterado. Um exemplo é apresentado a seguir:

```
#
# Protocolos Internet (IP)
#
ip      0       IP      # protocolo internet, número do pseudo-protocolo
icmp    1       ICMP    # protocolo de controle de mensagens internet
igmp    2       IGMP    # protocolo de propagação de grupos internet
tcp     6       TCP     # protocolo de controle de transmissão
udp     17      UDP     # protocolo de datagrama de usuário
raw     255     RAW     # interface de IP não tratado
```

9.4 RPC - Chamada de Procedimento Remoto

Um mecanismo muito genérico de aplicações cliente-servidor é disponibilizado pelo pacote RPC *Chamada de Procedimento Remoto*[6]. Ele foi desenvolvido pela Sun Microsystems e é uma coleção de ferramentas e bibliotecas. Aplicações importantes construídas sobre a RPC são NFS - Sistemas de Arquivos em Rede e NIS - Sistemas de Informações em Rede, ambos os quais serão melhor descritos em capítulos posteriores.

Um servidor RPC consiste de uma coleção de procedimentos que um cliente utiliza enviando uma solicitação RPC ao servidor, junto com os parâmetros do procedimento. O servidor irá acionar o procedimento indicado pelo cliente, retornando para aquele os valores obtidos, caso haja algum que tenha sido retornado pelo programa acionado. Para tornar este serviço independente de plataforma, todos os dados trocados entre o cliente e o servidor são convertidos para o formato XDR - *Representação Externa de Dados* pelo emissor e convertido para a representação da plataforma local pelo receptor.

[6] Remote Procedure Call

Algumas vezes, melhorias em uma aplicação RPC podem introduzir incompatibilidades na interface RPC. Obviamente, uma simples mudança do servidor pode trazer problemas para todas as aplicações que esperam o comportamento original. De qualquer forma, programas RPC têm números de versão definidos, normalmente iniciados por 1, e onde cada nova versão da interface incrementa a versão como em um contador. Freqüentemente, um servidor é oferecido em diversas versões simultaneamente, permitindo que os clientes possam então indicar o número da versão desejada nas suas requisições.

A comunicação de rede entre servidores RPC e cliente é bastante peculiar. Um servidor RPC oferece uma ou mais coleções de procedimentos, onde cada conjunto é chamado por um *programa* e é identificado por um *número de programa*. Uma lista mapeando o nome do serviço para o número de programa é normalmente mantida no arquivo `/etc/rpc`, o qual é parcialmente apresentado a seguir.

```
  Um exemplo do arquivo /etc/rpc.

#
# /etc/rpc - diversos serviços baseados em RPC
#
portmapper      100000  portmap sunrpc
rstatd          100001  rstat rstat_svc rup perfmeter
rusersd         100002  rusers
nfs             100003  nfsprog
ypserv          100004  ypprog
mountd          100005  mount showmount
ypbind          100007
walld           100008  rwall shutdown
yppasswdd       100009  yppasswd
bootparam       100026
ypupdated       100028  ypupdate
```

Em redes TCP/IP, os autores do RPC confrontaram-se com o problema de mapeamento de números de programas com serviços genéricos de rede. A solução foi definida de forma que cada servidor atende aos protocolos TCP e UDP para uma versão específica de um determinado programa. Geralmente aplicações RPC utilizarão UDP para o envio de dados e utilizarão TCP quando os dados a serem transferidos não caibam em um único datagrama UDP.

Obviamente, programas clientes têm que encontrar uma forma de saber em qual porta o programa está mapeado. Usar um arquivo de configuração para isso poderia ser um tanto inflexível, uma vez que aplicações RPC não utilizam portas reservadas. Não haveria nenhuma garantia de que a porta originalmente definida pela base de dados seria a utilizada pelo processo. Ou seja, aplicações RPC utilizam qualquer porta que possam e as registram no denominado *servidor de mapeamento de portas*[7]. Este age como um negociador de serviços para todos os programas RPC em execução na máquina: um cliente que deseje contactar um serviço com um determinado número de programa, irá inicialmente contactar o servidor de mapeamento de portas do servidor, o qual retornará os números das portas TCP e UDP de serviço desejado.

Este método tem uma deficiência particular, pois introduz um ponto de falha, similar ao mesmo introduzido pelo servidor `inetd` de serviços de Berkeley. De qualquer forma ele é um pouco pior pois o programa portmapper pode terminar anormalmente e as informações das portas RPC serão perdidas. Isso significa que os servidores RPC deverão ser reinicializados manualmente ou ainda todo o sistema deverá ser reinicializado.

No `Linux`, o portmapper é chamado `rpc.portmap` e reside no diretório `/usr/sbin`. Além de estar certo de que ele é inicializado pelo programa `rc.inet2`, o portmapper não requer qualquer esforço de configuração.

9.5 Configurando os Comandos `r`

Alguns comandos podem ser executados remotamente. São eles `rlogin`, `rsh`, `rcp` e `rcmd`. Eles criam um ambiente de trabalho na máquina remota e permitem que o usuário execute o comando. Obviamente, o usuário necessita ter uma conta na máquina onde os comandos são executados. Mais ainda, todos estes comandos executam um procedimento de autorização. Normalmente o cliente deverá informar o nome de acesso à máquina remota, a qual solicitará uma senha que será validada da forma usual.

Algumas vezes, poderá ser necessário não exigir certas verificações de segurança para determinados usuários. Por exemplo, caso se tenha que acessar outras máquinas freqüentemente em uma mesma rede local, pode-se admitir que o usuário não tenha que indicar sua senha a todo momento. Se algumas vezes isso pode ser uma questão de conveniência, por vezes, em outros casos pode ser uma questão de

[7] portmapper daemon

segurança não enviar a senha por uma rede insegura.

Desabilitar os procedimentos de segurança é aconselhável somente com um pequeno número de máquinas, cujas bases de dados de senhas sejam sincronizadas ou para um pequeno número de usuários privilegiados que necessitem acessar muitas máquinas por razões administrativas. Toda vez que se deseje permitir que pessoas acessem a uma máquina sem um usuário específico ou sem senha, deve-se estar seguro de que não se deve fornecer este tipo de acesso para todo e qualquer usuário.

Há duas formas de desabilitar a verificação de acesso para os comandos `r`. Uma é o superusuário permitir que alguns ou todos os usuários acessem algumas ou todas as máquinas da rede (a última opção é definitivamente desaconselhável!) sem a necessidade de senhas. Este acesso é controlado por uma arquivo denominado `/etc/hosts.equiv`. Ele contém uma lista de máquinas e nomes de usuários que são considerados equivalentes aos usuários da máquina local. Um opção alternativa seria o usuário manter contas em todas as máquinas. Isso deve estar especificado no arquivo `.rhosts` no diretório pessoal do usuário. Por razões de segurança, este arquivo deve pertencer ao próprio usuário ou ao superusuário e não deve ser uma ligação simbólica, pois desta forma será ignorado.[8] Quando um cliente solicita um serviço `r`, a sua máquina e o nome do usuário são pesquisados no arquivo `/etc/hosts.equiv` e no arquivo `.rhosts` do usuário, caso ele deseje acessar a máquina remota com a mesma conta. Por exemplo, assumindo-se que a usuária `janete` esteja trabalhando na máquina `roraima` e deseje conectar-se com a conta `catia` na máquina `minas`. Através dos comandos seguintes, nós nos referiremos a Janete como usuário *cliente* e a Cátia como o usuário *local*. Agora quando Janete digitar

```
$ rlogin -l catia minas
```

na máquina `roraima`, o servidor irá inicialmente verificar no arquivo `hosts.equiv`[9] se Janete tem autorização para livre acesso e caso isso falhe irá tentar verificar o arquivo `.rhosts` no diretório pessoal do usuário `catia`.

O arquivo `hosts.equiv` na máquina `minas` tem o seguinte conteúdo:

```
roraima
```

[8] Em um ambiente NFS, pode ser necessário definir as autorizações de acesso ao arquivo iguais a 444, porque o superusuário normalmente não tem autoridade para acessar arquivos residentes em um disco montado via NFS.

[9] Note que o arquivo `hosts.equiv` *não* é pesquisado quando alguém tenta conectar-se como `superusuário`.

```
minas
-publica
amapa.fisica.pantanal.edu.br      gustavo
```

cada entrada consiste de um nome de máquina, opcionalmente seguido de um nome de usuário. Caso um nome de máquina apareça de forma solitária, todos os usuários daquela máquina serão admitidos como usuários com contas locais sem qualquer verificação adicional. No exemplo acima, Janete tem permissão de acesso à máquina `minas`, utilizando a sua própria conta `janete` quando a origem do acesso for `roraima` e o mesmo se aplica a todo e qualquer usuário de `roraima`, exceto para o `superusuário`. De qualquer forma, caso Janete planeje utilizar o usuário `catia`, será necessário informar uma senha como de costume.

Caso um nome de máquina seja seguido por um nome de usuário, como por exemplo, na última linha do arquivo acima, este usuário estará dispensado de utilizar senha para todas as contas exceto a de `superusuário`.

O nome da máquina pode ainda ser precedido pelo sinal de menos (-), como na entrada da máquina "`-publica`". Ele requer autorização para todas as contas na máquina `publica`, a menos que o usuário tenha seus direitos de acesso expressos no arquivo `.rhosts`.

O formato do arquivo `.rhosts` é idêntico ao do arquivo `hosts.equiv`, mas tem um significado ligeiramente diferentemente. Considerando o arquivo `.rhosts` da usuário Cátia na máquina `minas`:

```
tocantins.pantanal.edu.br
roraima      janete
```

A primeira entrada fornece ao usuário `catia` livre acesso ao acessar o sistema a partir da máquina `tocantins.pantanal.edu.br`, mas não afeta os direitos de qualquer outra conta em `minas` ou `tocantins`. A segunda entrada é uma pequena variação, fornecendo a `janete` livre acesso à conta de `catia` ao acessar a máquina `minas`.

Note que o nome da máquina do cliente é obtido a partir do mapeamento reverso do endereço da máquina de origem da requisição, fazendo que com esta funcionalidade falhe caso a máquina seja desconhecida para o resolvedor de nomes. O nome da máquina cliente é considerada válida em um dos seguintes casos:

- O nome canônico da máquina (não um nome alternativo) confere literalmente com o nome do arquivo.

- Caso o nome da máquina seja um nome totalmente qualificado (como um que retorne do resolvedor de nomes quando o DNS está sendo executado) e ele não confere com nenhum nome do arquivo, ele é comparado com o nome da máquina expandida com o nome do domínio local.

Capítulo 10

O NIS - Sistema de Informações em Rede

Hoje em dia, NIS está disponível virtualmente para todos os `Unices` e existem diversas implementações gratuitas dele. Uma delas é o Net-2 da distribuição BSD que foi derivada de uma implementação de domínio público doada pela Sun. O código da biblioteca cliente desta versão está na GNU `libc` há um bom tempo, enquanto os programas administrativos foram recentemente portados para o `Linux` por Swen Thummler.[1] Um servidor NIS está faltando nas referências de implementações: Tobias Reber escreveu outro pacote NIS incluindo todas as ferramentas e um servidor, chamados `yps`.[2]

Atualmente, uma nova versão do código NIS chamada NYS está sendo desenvolvida por Peter Eriksson,[3] a qual suporta tanto o NIS puro quanto a revisão da Sun conhecida como NIS+. NYS não provê somente um conjunto de ferramentas NIS e um servidor, mas também adiciona um novo conjunto de funções em uma biblioteca, as quais provavelmente serão convertidas em um padrão da `libc` padrão. Ele inclui um novo esquema de configuração para a resolução de nome de máquina que substitui o sistema atual de uso do `host.conf`. As facilidades destas funções serão discutidas a seguir.

[1] Ele pode ser encontrado em `swen@uni-paderborn.de`. Os clientes NIS estão disponíveis em `yp-linux.tar.gz` no domínio `metalab.unc.edu` em `system/Network`.

[2] A versão atual é denominada `yp-tools` e pode ser obtida em `ftp.lysator.liu.se` no diretório `/pub/NYS`.

[3] Ele pode ser encontrado em `pen@lysator.liu.se`.

Este capítulo irá focar o NYS ao invés de observar os outros pacotes, os quais serão referenciados como código NIS tradicional. Caso se deseje executar quaisquer destes pacotes, as instruções deste capítulo talvez sejam suficientes. Para obter informações adicionais, por favor obtenha um livro padrão para NIS, como o "*NFS and NIS*" de Hal Stern (ver [Stern92]), ou veja o Como Fazer – NIS, traduzido pela Conectiva Informática; veja na Bibliografia, na página 459 os documentos que podem ser pesquisados.

Quando este livro foi escrito originalmente, o NYS ainda estava sob desenvolvimento, e diversos utilitários de rede do `Linux` como os programas `login` ainda não estavam cientes do esquema de configuração do NYS. Até que ele seja absorvido pela `libc` principal, ainda será necessário recompilar os binários, caso se deseje que eles utilizem o NYS. Em qualquer destas aplicações os arquivos `Makefiles`, devem especificar `-lnsl` como última opção antes da `libc` para a ligação do programa. Assim pode-se utilizar as funções relevantes a partir da `libnsl`, a biblioteca do NYS, ao invés da biblioteca C padrão.

10.1 Conhecendo o NIS

NIS mantém uma base de dados de informações denominadas *mapas*, que contém pares de chaves. Mapas são armazenados em uma máquina central que executa o servidor NIS e a partir da qual, os clientes recuperam as informações através de diversas chamadas RPC. Muito freqüentemente, mapas são armazenados no formato DBM.[4]

Os mapas em si são gerados a partir de arquivos textos mestres, como por exemplo os arquivos `/etc/hosts` ou `/etc/passwd`. Para alguns arquivos, diversos mapas são criados, um para cada tipo de chave. Por exemplo, pode-se pesquisar o arquivo `hosts` na busca por um nome de máquina ou por um endereço IP. Neste caso, dois mapas são gerados a partir deste arquivo, os mapas `hosts.byname` e `hosts.byaddr`, respectivamente. A tabela 10.1 lista os mapas mais comuns e os seus arquivos de origem.

Há outros arquivos e mapas que podem encontrar suporte em um ou outro pacote NIS e que podem conter informações para aplicações não discutidas neste livro, como o mapa `bootparams` que pode ser usado por alguns servidores BOOTP,

[4]DBM é uma biblioteca de gerenciamento de bases de dados que usa técnicas numéricas para acelerar as operações de pesquisa. Esta implementação livre do DBM no projeto GNU é chamada de `gdbm`, a qual é parte de muitas distribuições `Linux`.

Arquivo Mestre	Mapa(s)	Mapa(s)
/etc/hosts	hosts.byname	hosts.byaddr
/etc/networks	networks.byname	networks.byaddr
/etc/passwd	passwd.byname	passwd.byuid
/etc/group	group.byname	group.bygid
/etc/services	services.byname	services.bynumber
/etc/rpc	rpc.byname	rpc.bynumber
/etc/protocols	protocols.byname	protocols.bynumber
/usr/lib/aliases	mail.aliases	

Tabela 10.1: Alguns mapas padrões do NIS e seus arquivos correspondentes

ou mapas que não têm atualmente qualquer função no `Linux` (como os mapas `ethers.byname` e `ethers.byaddr`).

Para alguns mapas, as pessoas comumente utilizam *nomes curtos*, os quais são mais simples de serem memorizados e digitados. Para se obter uma lista completa dos nomes curtos conhecidos pelas ferramentas NIS, deve-se executar o seguinte comando:

```
$ ypcat -x
NIS map nickname translation table:
        "passwd" -> "passwd.byname"
        "group" -> "group.byname"
        "networks" -> "networks.byaddr"
        "hosts" -> "hosts.byname"
        "protocols" -> "protocols.bynumber"
        "services" -> "services.byname"
        "aliases" -> "mail.aliases"
        "ethers" -> "ethers.byname"
        "rpc" -> "rpc.bynumber"
        "netmasks" -> "netmasks.byaddr"
        "publickey" -> "publickey.byname"
        "netid" -> "netid.byname"
        "passwd.adjunct" -> "passwd.adjunct.byname"
        "group.adjunct" -> "group.adjunct.byname"
        "timezone" -> "timezone.byname"
```

O servidor NIS é tradicionalmente chamado de `ypserv`. Para uma rede média, um único servidor será suficiente, porém redes grandes podem executar diversos servidores em máquinas diferentes e em segmentos diferentes da rede permitindo maior segurança e balanceamento entre servidores e roteadores. Estes servidores são sincronizados definindo-se um como *servidor mestre* e os demais como *servidores escravos*. Mapas serão criados somente na máquina onde for executado o servidor mestre. A partir deste, eles serão distribuídos para todos os escravos.

O leitor mais atento pode ter percebido que o termo rede foi colocado de forma muito vaga até aqui e isso se deve ao fato de NIS ter um conceito distinto para se referir a uma rede, ou seja, o conjunto de todas as máquinas que compartilham parte de suas informações e dados de configuração do sistema através do NIS, é o chamado: *domínio NIS*. Infelizmente, domínios NIS não têm absolutamente nada em comum com os domínios encontrados no DNS. A fim de evitar esta ambigüidade ao longo deste capítulo, sempre especificaremos o tipo de domínio a que se está referindo.

Os domínios NIS têm exclusivamente uma função administrativa. São invisíveis para a maioria dos usuários, exceto para aqueles que compartilham senhas entre todas as máquinas do domínio. Desta forma, o nome dado a um domínio NIS é relevante somente para administradores. Normalmente, qualquer nome servirá, desde que ele seja diferente de qualquer outro nome de domínio NIS existente na rede local. Por exemplo, caso o administrador da rede da Cervejaria Virtual resolva criar dois domínios NIS, um para a Cervejaria e outro para a Vinícola, eles podem chamar-se, por exemplo `cervejaria` e `vinicola`, respectivamente. Outro esquema comumente utilizado é o de simplificar o nome do domínio NIS, chamando-o somente de NIS. Para configurar e mostrar o nome do domínio NIS de uma máquina, deve-se utilizar o comando `domainname`. Ao ser acionado sem argumentos, ele imprime o nome do domínio NIS. Para configurar o nome do domínio deve-se executá-lo como superusuário e digitar-se:

```
# domainname cervejaria
```

Domínios NIS determinam que servidor NIS deverá ser pesquisado pela aplicação. Por exemplo, o programa `login` em uma máquina da Vinícola pode, obviamente, somente pesquisar o servidor NIS da Vinícola (ou um deles, caso haja mais de um) para descobrir a informação de senha de um usuário, enquanto uma aplicação em uma máquina da Cervejaria deve pesquisar o servidor NIS da Cervejaria.

Um mistério porém permanece sem solução: como os clientes descobrem a qual servidor eles devem se conectar. A abordagem mais simples poderia ser um arquivo de configuração que define o nome da máquina onde o servidor é executado. De qualquer forma, esta abordagem é pouco flexível, porque não permite que os clientes usem diferentes servidores (para o mesmo domínio), dependendo de sua disponibilidade. De qualquer forma, implementações tradicionais do NIS baseiam-se em um servidor especial chamado `ypbind` para detectar um servidor NIS adequado para o seu domínio NIS. Antes de estar apta a executar quaisquer pesquisas NIS, uma aplicação deve encontrar qual servidor `ypbind` pode ser usado.

O servidor `ypbind` testa os servidores através da propagação na rede local. O primeiro servidor que responder é assumido como sendo o potencialmente mais rápido e será usado nas pesquisas NIS subseqüentes. Após um certo intervalo ou se o servidor se tornar indisponível, `ypbind` irá repetir o teste para servidores ativos novamente.

Agora o ponto de discórdia sobre a definição dinâmica do servidor NIS reside no fato de ela ser raramente utilizada e que introduz um problema sério de segurança: `ypbind` cegamente acredita em qualquer máquina que responda, o qual pode ser um perfeito servidor NIS, assim como um intruso mal intencionado. Desnecessário dizer que isto se torna especialmente problemático quando se administram bases de dados de senhas sobre NIS. Para proteger-se disso, NIS *não* usa `ypbind` por padrão, mas escolhe o nome da máquina servidora a partir de um arquivo de configuração.

10.2 NIS versus NIS+

NIS e NIS+ compartilham pouco mais que seus nomes e um objetivo comum. NIS+ é estruturado de uma forma totalmente diferente. Ao invés de um simples espaço com nomes de domínios NIS, ele usa um nome hierárquico similar ao DNS. Ao invés de mapas, são usadas *tabelas* que são constituídas de linhas e colunas, onde cada linha representa um objeto na base de dados do NIS+, enquanto as colunas representam as propriedades dos objetos que o NIS+ conhece e se relaciona. Cada tabela para um domínio NIS+ mantém as características das tabelas dos domínios pais. Adicionalmente, uma entrada em uma tabela pode conter uma ligação para outra tabela. Estas funcionalidades tornam possível estruturar as informações de diversas formas.

O NIS tradicional tem uma versão RPC igual a 2, enquanto que NIS+ tem a versão igual a 3.

NIS+ não parece ser muito usado ainda, portanto não entraremos em maiores detalhes sobre ele. Para maiores informações recomendamos uma leitura do manual do administrador NIS+ da Sun ([NISPlus]).

10.3 O Cliente NIS

Caso se esteja familiarizado no desenvolvimento ou porte de aplicações de rede, pode-se notar que os mapas NIS listados acima correspondem a funções de uma biblioteca C. Por exemplo, para obter informações a partir do arquivo `passwd`, geralmente são utilizadas as funções `getpwnam(3)` e `getpwuid(3)`, as quais retornam informações sobre a conta associada a determinado nome de usuário ou a uma identificação numérica, respectivamente. Sob condições normais, estas funções executam uma pesquisa no arquivo padrão, no caso o `/etc/passwd`.

Uma aplicação NIS que necessite destas funções, irá modificar seu comportamento, e colocar uma chamada RPC para fazer com que o servidor NIS execute as pesquisas de nomes de usuários ou identificações. Isso ocorre de maneira totalmente transparente para a aplicação. A função pode ou anexar o mapa NIS ou substituir o arquivo original por ele. Obviamente, as alterações não são realizadas diretamente no arquivo original, porém para a aplicação elas *aparentam* terem sido.

Para implementações NIS tradicionais, há certas convenções a serem usadas, assim como na substituição de mapas ou sobre aqueles que foram anexados à informação original. Alguns mapas, como o `passwd`, requerem modificações no arquivo `passwd`, as quais se forem realizadas de forma errônea, podem gerar problemas de segurança. Para evitar estes problemas, NYS usa um sistema geral de configuração, que determina se um determinado conjunto de funções cliente usa os arquivos originais, mapas NIS ou NIS+ e em qual ordem. Ele será descrito em maiores detalhes em uma seção posterior.

10.4 Servidor NIS

Após tanta teoria tecno-babel, é tempo de "sujar as mãos" com algum trabalho de configuração. Neste seção, cobriremos a configuração de um servidor NIS.

◇ Caso se estejam executando somente testes com o servidor, esteja certo de não configurar um nome de domínio NIS que já esteja em uso na rede. Isso pode corromper todos os serviços de rede e provocar tristeza e ira em diversas pessoas.

Há atualmente dois servidores NIS de livre distribuição disponíveis para Linux, um deles no pacote ypserv de Peter Eriksson. Não importa qual será executado, independente de se usar NYS ou o NIS padrão cujo código cliente utiliza a biblioteca libc. Quando este livro foi escrito, o código para o gerenciamento do servidores escravos NIS parecia ser mais completo em yps. Então, caso se tenha a possibilidade de lidar com diversos servidores escravos, yps poderá ser uma sábia escolha.

Após instalar o programa servidor (ypserv) no diretório /usr/sbin, deve-se criar o diretório que conterá os arquivos de mapas que serão distribuídos. Ao configurar o domínio NIS para o domínio cervejaria, os mapas irão para o diretório /var/yp/cervejaria. O servidor determinará se está servindo a um domínio NIS em particular ao checar se o diretório de mapas está presente. Caso o serviço não esteja habilitado para algum domínio NIS, deve-se estar seguro de que o diretório foi removido.

Mapas são atualmente armazenados em arquivos DBM para agilizar as pesquisas. Eles são criados a partir de arquivos mestres usando um programa chamado makedbm (no servidor de Tobias) ou dbmload (no servidor de Peter). Eles não podem ser intercambiados. Transformar um arquivo mestre em um formato utilizável pelo programa dbmload normalmente requer alguma mágica dos utilitários awk ou sed, o qual tende a ser um pouco tedioso para digitar e difícil de relembrar. De qualquer forma, o pacote ypserv de Peter Eriksson contém um arquivo Makefile (chamado ypMakefile) que executará todas as tarefas necessárias. Deve-se instalar o Makefile no mapa de diretórios e editá-lo para refletir os mapas que se deseja distribuir. A partir do topo do arquivo, pode se encontrar o parâmetro all que lista os serviços que o servidor ypserv oferece. Por padrão, a linha terá a seguinte aparência:

```
all: ethers hosts networks protocols rpc services passwd group netid
```

Caso não se deseje produzir os mapas ethers.byname e ethers.byaddr, por exemplo, basta remover os pré-requisitos ethers para esta regra. Para testar a configuração, deve-se iniciar com somente um ou dois mapas, como por exemplo com os mapas services.*.

Após editar o `Makefile`, enquanto ele já esteja no diretório de mapas, basta digitar "`make`". Este procedimento irá gerar automaticamente os mapas e instalá-los. Deve-se estar seguro de atualizar os mapas toda vez que os mestres forem alterados, caso contrário eles permanecerão invisíveis para o restante da rede.

A próxima seção explica como configurar o cliente NIS. Caso a configuração não funcione, deve-se tentar descobrir se alguma requisição foi recebida do servidor ou não. Caso se especifique o indicador `-D` ou `-debug` na linha de comando do servidor NYS, ele apresentará uma série de mensagens informativas na console sobre todas as pesquisas NIS recebidas e o resultado retornado. Certamente isso será de extrema utilidade na busca da causa de um problema. O servidor de Tobias não tem tal opção disponível.

10.5 Segurança em Um Servidor NIS

O uso do NIS tem aspectos de segurança bastante delicados: ele pode deixar o arquivo de senhas da rede acessível virtualmente a todos os usuários da rede local e a outras que possam estar interconectadas, o que pode facilitar o acesso a um grande número de intrusos. Assim que um intruso souber o nome do domínio NIS e o endereço do servidor, ele pode simplesmente enviar uma requisição do mapa `passwd.byname` e instantaneamente receber todas as senhas do sistema. Com um programa de quebra de senhas ágil e um bom dicionário, descobrir a senha de alguns usuários do sistema não será problema.

Devido a isso, foram criadas as opções conhecidas como `securenets`. Elas restringem o acesso ao servidor NIS a somente algumas máquinas, baseado no seu endereço IP ou nos seus números de rede. A última versão do `ypserv` implementa esta funcionalidade de uma forma bastante simples, utilizando os arquivos `etc/hosts.allow` e `etc/hosts.deny`, descrito no capítulo 9. Por exemplo, para restringir o acesso ao servidor NIS somente às máquinas da rede da Cervejaria Virtual, o administrador de redes deverá informar o seguinte no arquivo `hosts.allow`:

```
ypserv: 172.16.2
```

Isso faz com que todas as máquinas na rede IP **172.16.2.0** tenham acesso ao servidor NIS. Para evitar que qualquer outra máquina acesse o servidor NIS deve-se incluir a seguinte linha no arquivo `hosts.deny`:

```
ypserv: all
```

Números IP não são a única forma de especificar máquinas ou redes nos arquivos `hosts.allow` e `hosts.deny`. Verifique na página de manual de `hosts.access(5)` para maiores detalhes. De qualquer forma é importante saber que não é possível usar nomes de máquinas ou domínios como uma entrada de `ypserv`. Caso seja especificado um nome de máquina, o servidor tentará resolver o endereço IP, mas o resolvedor chama o servidor NYS, provocando um círculo vicioso sem fim.

Pode-se ainda utilizar um programa `portmaster` seguro ao invés das opções `securenets`. O `portmap-3.0`[5] também utiliza o arquivo `hosts.allow`, disponibilizando-o para todos os servidores RPC e não somente para `ypserv`. Não se deve utilizar os dois sistemas de segurança ao mesmo tempo, pois este procedimento causaria uma sobrecarga nos processos de autorização.

10.6 Configurando um Cliente NIS com NYS

O primeiro passo para a configuração de um cliente NYS é informar qual o servidor NIS que deve ser utilizado para os serviços NIS, configurando o arquivo de configuração `/etc/yp.conf`. Uma entrada muito simples para a máquina da rede da Vinícola teria a seguinte aparência:

```
# yp.conf - configuração YP para a biblioteca NYS.  #
domainname vinicola
server garibaldi
```

O primeiro comando indica a todos os clientes NIS que eles pertencem ao domínio NIS chamado `vinicola`. Caso esta linha seja omitida, NYS irá utilizar o nome de domínio definido no sistema através do comando `domainname`. O comando `server` no arquivo acima define o nome do servidor NIS a ser utilizado. Obviamente, o endereço IP de `garibaldi` deve estar configurado no arquivo `hosts`. Alternativamente pode-se usar o endereço IP no comando `server`.

Na forma mostrada acima, o comando `server` indica ao NYS qual o nome do servidor a ser usado, independente do conteúdo do campo nome de domínio. Caso

[5] Disponível via FTP anônimo em metalab.unc.edu sob o diretório `Linux/systems/Network`

haja mudanças freqüentes entre diferentes domínios NIS para a mesma máquina pode-se desejar manter a informação de diversos domínios no arquivo `yp.conf`. Para tanto é necessário adicionar o nome do domínio ao comando `server`. Por exemplo, pode-se alterar o exemplo acima para um equipamento portátil, que teria o seguinte conteúdo:

```
# yp.conf - YP configuração para biblioteca NYS
#
server garibaldi  vinicola
server aracaju    cervejaria
```

Isto permite que o portátil possa ser conectado a qualquer um dos domínios NIS, simplesmente alterando-se o nome do domínio através do comando `domainname` em tempo de inicialização do sistema.

Após a criação de arquivos de configuração básica e assegurando-se de que eles estejam acessíveis a todos os usuários deve-se executar os primeiros testes, a fim de verificar se a máquina consegue conectar-se com o servidor NIS. Deve-se escolher qualquer mapa que seja realmente distribuído pelo servidor, como por exemplo o `hosts.byname` e tentar recuperá-lo através do uso do utilitário `ypcat`, que assim como todas as outras ferramentas administrativas NIS, deverá estar no diretório `/usr/sbin`.

```
# ypcat hosts.byname
191.72.2.2      caxias      caxias.vinicola.com.br
191.72.2.3      gramado     gramado.vinicola.com.br
191.72.1.1      aracaju     aracaju.cvirtual.com.br
191.72.2.1      aracaju     aracaju.vinicola.com.br
191.72.1.2      maceio      maceio.cvirtual.com.br
191.72.1.3      jpessoa     jpessoa.cvirtual.com.br
191.72.2.4      garibaldi   garibaldi.cvirtual.com.br
```

A saída obtida deverá ter uma aparência similar à lista acima apresentada. Caso se obtenha uma mensagem de erro como por exemplo “” ou algo similar, as causas podem ser: ou o nome do servidor NIS do domínio definido em `yp.conf` não existe, ou o servidor não pode ser alcançado por alguma razão. Neste caso, deve ser executado um comando `ping` para as máquinas envolvidas, a fim de verificar as condições de conectividade com o servidor NIS e caso a conexão esteja funcionando corretamente deve-se checar se o servidor NIS está ativo. Isto pode ser feito através do comando `rpcinfo`, o qual deve produzir a seguinte saída:

```
# rpcinfo -u servidor ypserv
programa 100004 versão 2 pronto e aguardando
```

10.7 Escolhendo os Mapas Corretos

Estando-se seguro de que a comunicação com o servidor NIS está em perfeitas condições, deve-se decidir quais arquivos de configuração serão substituídos ou incrementados com os mapas NIS. Comumente, utilizam-se os arquivos NIS para funções de pesquisas de máquinas e senhas. O primeiro é especialmente útil caso não se execute o BIND. O último permite que todos os usuários acessem qualquer máquina do domínio utilizando a mesma conta e senha. Isto normalmente exige que o diretório pessoal do usuário esteja localizado de forma central e seja compartilhado por todas as máquinas via NFS. Isso é explicado de forma detalhada na seção 10.8 a seguir. Outros mapas, como `services.byname`, não provocam um ganho tão visível, mas economizam a edição de alguns arquivos, caso se tenha instalado qualquer aplicação de rede que use o nome do serviço que não esteja no arquivo padrão `services`.

Geralmente, é desejável ter-se alguma liberdade de escolha quando uma função de pesquisa deve utilizar arquivos locais e quando deve utilizar o servidor NIS. NYS permite a configuração da ordem em que uma função acessa estes serviços. Isso é controlado através do arquivo `/etc/nsswitch.conf`, que significa *Troca de Serviços de Nomes*[6], o qual obviamente não está limitado somente a serviços de nome. Para qualquer uma das funções de pesquisa suportadas pelo NYS, ele conterá uma linha denominando os serviços a serem utilizados.

A ordem correta dos serviços depende do tipo de dados. Porém não é como se as entradas contidas no mapa `services.byname` contenham entradas diferentes do arquivo `services` local, na verdade ele somente contém mais dados. Então uma boa opção pode residir em selecionar os arquivos locais inicialmente e checar os mapas NIS somente se o nome do serviço não for encontrado. As informações de nomes de máquinas, por outro lado, podem mudar freqüentemente; por outro lado DNS ou o servidor NIS deve ter sempre as informações mais atualizadas, enquanto o arquivo local `hosts` é mantido somente como uma cópia de segurança caso DNS e NIS falhem. Neste caso, os arquivos locais devem ser checados por último.

O exemplo abaixo mostra como configurar as funções `gethostbyname(2)`,

[6] Name Service Switch

`gethostbyaddr(2)` e `getservbyname(2)` conforme descrito acima. Elas tentarão os serviços listados na ordem apresentada, onde caso uma pesquisa seja bem sucedida, o resultado será informado, caso contrário o próximo serviço será testado.

```
# pequeno exemplo do arquivo  /etc/nsswitch.conf
#
hosts:     nis dns files
services:  files nis
```

A lista completa de serviços pode ser definida como uma entrada no arquivo `nsswitch.conf` mostrado. Os mapas correntes, arquivos, servidores e objetos podem ser pesquisados dependendo do nome da entrada.

`nisplus` **ou** `nis+` Usa o servidor NIS+ para este domínio. A localização do servidor é obtida no arquivo `/etc/nis.conf`.

`nis` Usa o servidor NIS atual deste domínio. A localização do servidor pesquisado é configurado no arquivo `yp.conf`, conforme apresentado na seção anterior. Para as entradas em `hosts`, os mapas `hosts.byname` e `hosts.byaddr` serão pesquisados.

`dns` Usa o servidor de nomes DNS. Este tipo de serviço é somente útil com uma entrada no arquivo `hosts`. As pesquisas de servidores de nomes são ainda determinadas pelo arquivo padrão `resolv.conf`.

`files` Usa o arquivo local, como o arquivo `/etc/hosts` com entrada de `hosts`.

`dbm` Pesquisa os arquivos DBM localizados em `/var/dbm`. O nome usado pelo arquivo é o correspondente no mapa NIS.

Atualmente, NYS suporta as seguintes entradas no arquivo `nsswitch.conf`: `hosts`, `networks`, `passwd`, `group`, `shadow`, `gshadow`, `services`, `protocols`, `rpc` e `ethers`. Outras entradas similares podem ser adicionadas.

A descrição abaixo apresenta um exemplo completo, que introduz uma outra funcionalidade no arquivo `nsswitch.conf`: o parâmetro `[NOTFOUND=return]` no parâmetro `hosts` informa ao NYS para retornar caso o item desejado não seja encontrado na base de dados do NIS ou DNS. Ou seja, NYS irá continuar a pesquisa nos arquivos locais *somente* se o acionamento do NIS ou do servidor DNS falhar por algum motivo. Neste caso os arquivos locais podem ser usados em tempo de inicialização e como uma cópia de segurança quando o servidor NIS estiver inativo.

```
Exemplo do arquivo nsswitch.conf:


# /etc/nsswitch.conf
#
hosts:      nis dns [NOTFOUND=return] files
networks:   nis [NOTFOUND=return] files

services:   files nis
protocols:  files nis
rpc:        files nis
```

10.8 Usando os Mapas `passwd` e `group`

Uma das maiores aplicações do NIS é a sincronização de usuários e informações de contas em todas as máquinas de um domínio NIS. Portanto, normalmente mantêm-se somente um pequeno arquivo `/etc/passwd` local, no qual as informações de todo o domínio são anexadas a partir dos mapas NIS. De qualquer forma, a simples habilitação de pesquisas NIS para este serviço no arquivo `nsswitch.conf` pode não ser suficiente.

Ao basear-se nas informações de senhas distribuídas pelo NIS, deve se estar seguro de que a identificação numérica de um usuário de qualquer conta do arquivo `passwd` local não coincida com a identificação numérica dos usuários do servidor NIS. Pode-se utilizar estas facilidades para outros propósitos, como a montagem de volumes NFS de outras máquinas na rede local.

Caso as identificações numéricas nos arquivos `/etc/passwd` ou `/etc/group` tenha alguma inconsistência com os mapas NIS, isso provocará a necessidade de ajustes de propriedade de todos os arquivos de usuários com identificação numérica duplamente referenciada em ambos os arquivos. Inicialmente deve-se alterar as identificações numéricas de usuários e grupos nos arquivos `passwd` e `group` para novos valores, em segundo lugar deve-se localizar todos os arquivos pertencentes ao usuário recém-alterado, e finalmente alterar a propriedade para a nova identificação. Assumindo que o usuário `news` tenha uma identificação numérica igual a 9 e o usuário `sandro` tenha uma identificação igual a 103, a qual foi mudada para outro valor, devem então ser empregados os seguintes comandos:

```
# find / -uid   9 -print >/tmp/uid.9
# find / -uid 103 -print >/tmp/uid.103
# cat /tmp/uid.9   | xargs chown news
# cat /tmp/uid.103 | xargs chown sandro
```

É importante que estes comandos sejam executados com o *novo* arquivo `passwd` instalado e que sejam identificados todos os nomes de arquivos antes de se alterar a propriedade de qualquer um deles. Para atualizar os grupos proprietários dos arquivos deve-se usar um comando similar.

Após este procedimento, a identificação numérica dos usuários e grupos do sistema deverá estar em acordo com os demais sistemas do domínio NIS. O próximo passo a ser executado, será a adição das linhas de configuração do arquivo `nsswitch.conf` que habilita pesquisas NIS para informações de usuários e grupos:

```
# /etc/nsswitch.conf - tratamento de passwd e group
passwd: nis files
group:  nis files
```

Isso habilita a pesquisa em primeiro lugar nos mapas NIS dos comandos `login` e todos os seus amigos, ao tentar acessar o sistema e caso a pesquisa falhe, indica a utilização dos arquivos locais. Normalmente praticamente todos as usuários serão removidos dos arquivos locais e somente as entradas para o superusuário `root` e contas genéricas como `mail` estarão presentes. Isso se deve ao fato de que tarefas vitais do sistema podem requerer o mapeamento de identificações numéricas para nomes de usuários e vice-versa. Por exemplo, tarefas administrativas do programa `cron` devem ser executadas pelo comando `su` para temporariamente tornar-se o usuário `news`, ou o subsistema UUCP pode gerar um relatório de condição. Caso os usuários `news` e `uucp` não tenham entradas no arquivo local `passwd`, estas tarefas irão falhar lamentavelmente caso o NIS não esteja ativo.

Há dois importantes aspectos de tudo o que foi apresentado até aqui: por um lado, a configuração conforme descrita neste guia funciona somente para conjuntos de programas que não utilizem senhas sombra, como aqueles incluídos no pacote `util-linux`. Os detalhes de se utilizar senhas sombra com NIS serão mostrados a seguir. Por outro lado, os comandos de acesso não são os únicos a utilizarem o arquivo `passwd` –por exemplo o comando `ls` utilizado constantemente por um grande número de usuários. Sempre que seja construída uma longa lista, `ls` irá mostrar os nomes simbólicos dos usuários e grupos proprietários de arquivos, os quais são na verdade o que o comando encontrou para cada identificação numérica de usuário e grupo no servidor NIS na primeira pesquisa efetuada. Isso irá tornar o processamento terrivelmente mais lento caso a rede local esteja congestionada, ou ainda pior, quando o servidor NIS não estiver na mesma rede física, fazendo com que os datagramas tenham que passar através de um roteador.

Bem, esta ainda não é a história completa. Imaginemos o que acontece quando um usuário muda a sua senha. Normalmente, ele aciona o comando `passwd`, o qual receberá a nova senha e atualizará o arquivo `passwd` local. Isso é impossível com o NIS, uma vez que na verdade ele não está disponível localmente, porém fazer com que os usuários acessem diretamente o servidor NIS pode não ser uma opção aceitável. Desta forma, NIS provê um substituto ao programa `passwd` chamado `yppasswd`, o qual faz um trabalho análogo na presença do NIS. Para alterar a senha na máquina servidora, deve-se contactar o servidor `yppasswdd` via RPC e prover as informações da senha alterada. Normalmente deve-se instalar o programa `yppasswd` sobre o programa `passwd` normal da seguinte forma:

```
# cd /bin
# mv passwd passwd.old
# ln yppasswd passwd
```

Simultaneamente deve-se instalar o programa `rpc.yppasswdd` no servidor e iniciá-lo a partir do `rc.inet2`. Isto irá efetivamente esconder todos os detalhes de tratamento de senhas e usuários pelo NIS.

10.9 Utilizando NIS com Suporte a Senhas Sombra

Ainda não há suporte a senhas sombra no NIS. John F. Haugh, o autor do conjunto de softwares de senhas sombra, liberou uma versão da biblioteca de funções sombra através da licença GPL no grupo `comp.sources.misc`. Há algum suporte a NIS, mas ainda está incompleto e os arquivos não podem ser adicionados à biblioteca padrão C. Por outro lado, a publicação de informações do arquivo `/etc/shadow` via NIS é uma forma de deficiência aos propósitos das ferramentas de senhas sombra.

Apesar das funções de pesquisas de senhas NYS não utilizarem o mapa `shadow.byname` ou qualquer outro similar, NYS suporta esta sistemática usando um arquivo `/etc/shadow` de forma transparente. Quando a implementação NYS de `getpwnam` é acionada para pesquisar as informações especificadas em um nome de acesso fornecido, as facilidades especificadas pela entrada `passwd` no arquivo `nsswitch.conf` são utilizadas.

O serviço `nis` irá simplesmente pesquisar o nome no mapa `passwd.byname` no servidor NIS. O serviço `files` irá verificar se o arquivo `/etc/shadow` está presente e em caso positivo, tentará abrí-lo. Caso não esteja, ou caso não tenha privilégios de `root`, usará o comportamento tradicional de pesquisar somente o arquivo `/etc/passwd`. De qualquer forma, caso o arquivo `shadow` exista e possa ser aberto, NYS irá extrair a senha de usuário do arquivo `shadow`. A função `getpwuid` é implementada de forma similar. Desta forma, os binários compilados com NYS irão lidar com um conjunto de ferramentas de senhas sombra de forma transparente.

10.10 Utilizando o Tradicional Código NIS

Caso se esteja utilizando o código tradicional que pode ser encontrado na `libc`, a configuração de um cliente NIS pode ser um pouco diferente. Por um lado, pode-se usar um servidor `ypbind` para propagar os servidores ativos, ao invés de divulgar isso a partir de um arquivo de configuração. Deve-se estar seguro de que o programa `ypbind` será acionado na inicialização do sistema. Ele deve ser acionado após o domínio NIS ter sido configurado e o `portmapper RPC` ter sido inicializado. Acionando-se o `ypcat` para testar o servidor pode funcionar conforme o mostrado anteriormente.

Recentemente, tem havido algumas informações de que o NIS pode falhar com a seguinte mensagem de erro "`clntudp_create: RPC: falha de portmapper - RPC: incapaz de receber`". Isso se deve a uma incompatibilidade na forma como `ypbind` comunica-se com as informações de construção das funções da biblioteca. Neste caso deve-se utilizar uma versão atualizada do NIS e recompilá-la para que o problema seja solucionado.[7]

Adicionalmente cabe citar que a forma como o NIS decide se deve mesclar as informações NIS com os arquivos locais é distinta daquela utilizada pelo NYS. Por exemplo, para utilizar os mapas de senhas NIS, deve-se incluir a seguinte linha no mapa `/etc/passwd`:

```
+:*:0:0:::
```

Isto assinala o local onde as funções de pesquisa de senhas "inserem" os mapas NIS. Deve-se inserir uma linha similar (menos as últimas duas colunas) em um

[7] O fonte do `yp-linux` pode ser obtido em `ftp.uni-paderborn.de` no diretório `/pub/Linux/LOCAL`.

arquivo `/etc/group` para se obter o mesmo resultado. Para usar os mapas `hosts.*` distribuídos pelo NIS, deve-se mudar a linha `order` no arquivo `host.conf`. Por exemplo, caso se deseje usar NIS, DNS e o arquivo `/etc/hosts` (nesta ordem) será necessária a seguinte alteração na linha para

```
order yp bind hosts
```

A implementação tradicional do NIS não suporta outros mapas no momento.

Capítulo 11

O Sistema de Arquivos de Rede

NFS, o sistema de arquivos de redes, é provavelmente o serviço de rede mais importante que utiliza o RPC. Ele permite acessar arquivos em máquinas remotas exatamente da mesma maneira como um usuário acessa qualquer arquivo localmente. Isto é possível graças à mistura de funcionalidades do kernel no lado do cliente (que usa o sistema de arquivos remoto) e um servidor NFS no lado do servidor (que provê os arquivos de dados). Este acesso ao arquivo é completamente transparente ao cliente e funciona com uma grande variedade de servidores e arquiteturas de máquinas.

NFS oferece diversas vantagens:

- Os dados acessados por todos os usuários podem ser mantidos numa máquina central, com os clientes montando seus diretórios no momento da inicialização. Por exemplo, pode-se manter todas as contas de usuários em uma única máquina e ter-se todas as máquinas da rede montando os diretórios pessoais `/home` a partir daquela máquina (um típico servidor de arquivos). Se for instalado junto com NIS, os usuários podem acessar qualquer sistema e ainda trabalhar em um único conjunto de arquivos.
- Dados que consomem muito espaço em disco podem ser mantidos em uma única máquina. Por exemplo, todos os arquivos e programas relacionados com o LaTeX e METAFONT podem ser mantidos em um único servidor.

- Dados Administrativos podem ser mantidos em uma única máquina, não sendo necessário mais utilizar o comando de cópias remotas `rcp` para instalar o mesmo arquivo em 20 diferentes máquinas.

Não é difícil executar a configuração básica de operações NFS, tanto no lado cliente quanto servidor e este capítulo tem a função de ensinar como isso é feito.

`Linux` NFS é principalmente um trabalho de Rick Sladkey,[1] que escreveu o código correspondente ao núcleo do NFS e muitas partes do servidor NFS. Este último é derivado do programa de usuário `unfsd` do servidor NFS originalmente escrito por Mark Shand e do `hnfs` (Harris NFS) servidor NFS, escrito por Donald Becker.

Vamos dar uma olhada agora no funcionamento do NFS. Um cliente solicita a montagem de um diretório de um servidor remoto em um diretório local, da mesma maneira que ele monta um dispositivo físico local. No entanto a sintaxe não é exatamente a mesma. Por exemplo, para montar o arquivo `/home` da máquina `aracaju` em `users` na máquina `blumenau`, o administrador poderá executar o seguinte comando na máquina `blumenau`:[2]

```
# mount -t nfs aracaju:/home /users
```

`mount` irá tentar conectar-se com o programa servidor remoto chamado `mountd` na máquina `aracaju` via RPC. O servidor irá verificar se a máquina `blumenau` tem permissão para montar o diretório em questão, e caso tenha, retorna a ela um descritor de arquivos. Este descritor de arquivos será usado em todas as requisições posteriores aos arquivos sob o `users`.

Quando alguém acessa um arquivo sob NFS, o kernel manda uma chamada RPC para o servidor `nfsd` na máquina servidora. Esta chamada leva como parâmetro o descritor de arquivos, o nome do arquivo a ser acessado, o identificador de usuário e de grupo. Eles são usados para controlar os direitos de acesso sobre um determinado arquivo. Para poder evitar que usuários não autorizados leiam ou modifiquem os arquivos, identificações de usuários e grupos devem ser iguais em ambos os servidores.

Em várias implementações de `Unix`, as funcionalidades de clientes e servidores NFS são implementadas em programas a nível de kernel, que são ativados no espaço de

[1] Rick pode ser encontrado no `jrs@world.std.com`.

[2] Note que se pode omitir a opção `-t nfs`, porque o programa `mount` sabe que o caractere dois pontos (:) indica um volume NFS.

usuário durante a inicialização do sistema. Eles são: o programas NFS (`nfsd`) no servidor e o programa *Servidor de Blocos de Entrada e Saída* , (`biod`) que é executado no cliente. Para aumentar a performance, o programa `biod` realiza entradas e saídas assíncronas usando "leituras adiantadas" e "gravações atrasadas", assim como diversas instâncias do programa `nfsd` são executadas concorrentemente.

A implementação de NFS no `Linux` é um pouco diferente: o código do cliente está integrado firmemente nas camadas do sistema de arquivo virtual (VFS) do kernel e não requerem controle adicional do programa `biod`. Por outro lado, o código do servidor é executado totalmente no espaço de usuário, tornando praticamente impossível executarem-se várias cópias do programa servidor simultaneamente, devido às questões de sincronismo que isto pode envolver.

O maior problema com o código NFS `Linux` é que o kernel do `Linux` versão 1.0 não pode alocar pedaços de memória com mais de 4kb; como conseqüência, o código de rede não pode suportar datagramas maiores que 3500 bytes após terem sido retirados o tamanho do cabeçalho, etc.. Isto significa que as transferências com servidores NFS rodando em sistemas que utilizam como padrão tamanhos grandes de datagramas UDP (por exemplo: 8k no SunOS) precisam ser reduzidos artificialmente. Isto produz perda de performance em algumas circunstâncias.[3] Esta limitação desapareceu nos kernels posteriores ao kernel `Linux`-1.1 e o código do cliente vem sendo modificado para se ter melhorias.

11.1 Preparando o NFS

Antes de se usar o NFS, tanto o servidor como o cliente, deve-se estar seguro de que o kernel tenha suporte a NFS. Os kernels modernos têm uma interface simples para esta verificação que reside no sistema de arquivos `/proc`: o arquivo `/proc/filesystems`, que você pode ser visualizado através do programa `cat`:

```
$ cat /proc/filesystems
minix
ext2
msdos
nodev proc
```

[3]Como me explicou Alan Cox: a especificação NFS requer que o servidor descarregue cada bloco de dados gravados em disco antes de retornar um OK. Como no kernel do BSD, é possível somente escrever páginas com tamanho de 4 Kb, gravar quatro pedaços de 1 Kb em um servidor baseado em BSD resulta em quatro operações de gravação com 4 Kb cada.

```
nodev nfs
```

Se a palavra `nfs` estiver faltando na lista, então será necessário compilar o kernel habilitando o suporte a NFS. Veja na seção "Configuração do kernel " no capítulo 3, como configurar as opções do kernel .

Para versões do kernel anteriores a `Linux` 1.1, a maneira mais simples de descobrir se o kernel tem suporte a NFS habilitado é tentar montar um sistema de arquivo NFS. Para isto, deve-se criar um diretório sob `/tmp`, e tentar montar um diretório local nele, como por exemplo:

```
# mkdir /tmp/teste
# mount localhost:/etc /tmp/teste
```

Se o comando `mount` falhar e apresentar a seguinte mensagem "`tipo de sistema de arquivos nfs não suportado pelo kernel`", deverá ser compilado um novo kernel com o NFS habilitado. Outras mensagens de erros não representarão problemas, como por exemplo o fato de ainda não ter sido configurado o servidor NFS na máquina local.

11.2 Montando um Volume NFS

Os volumes NFS[4] são montados de uma maneira muito similar à forma como os sistemas de arquivos normais são montados.

Pode-se acionar o comando `mount` usando a seguinte sintaxe:

```
# mount -t nfs volume_nfs diretório_local opções
```

O parâmetro *`volume_nfs`* deve ser especificado com a seguinte sintaxe: *`máquina_remota`*:*`diretório_remoto`*. Dado que esta notação é própria do sistema de arquivo NFS, não será necessário utilizar a opção `-t nfs`.

Existem opções adicionais que podem ser especificadas com o comando `mount`, sobre a montagem de um volume NFS. Elas podem ser informadas após a opção `-o` na linha de comando, ou no campo de opções do arquivo `/etc/fstab`. Em ambos

[4]Nos referimos a volumes e não a sistemas de arquivos, porque eles não são realmente sistemas de arquivos.

os casos, múltiplas opções devem ser separadas por vírgulas. As opções especificadas na linha de comando têm preferência sobre as dadas no arquivo `fstab`.

Segue um exemplo de entrada no arquivo `/etc/fstab` :

```
# volume              ponto de montagem   tipo  opções
news:/usr/spool/news  /usr/spool/news     nfs   timeo=14,intr
```

Este volume pode ser montado através do comando:

```
# mount news:/usr/spool/news
```

Na ausência de uma entrada no arquivo `fstab`, as chamadas NFS ao programa `mount` podem parecer complexas. Supondo-se que se queira montar o diretório home de uma máquina chamada `lua`, que usa tamanho de bloco de 4k para operações de leitura/escrita. Será necessário então diminuir o tamanho de bloco em 2k para adaptar-se ao tamanho do datagrama `Linux`, utilizando-se o comando:

```
# mount lua:/home /home -o rsize=2048,wsize=2048
```

A lista de todas as opções válidas está descrita completamente nas páginas de manual do `nfs(5)`, que vêm com os utilitários do NFS de Rick Sladkey (os quais podem ser encontradas no pacote `util-linux`). As opções mais importantes são:

`rsize=n` **and** `wsize=n` Esta opção especifica o tamanho do datagrama usado pelos clientes NFS nas requisições de leitura e escrita, respectivamente. Elas têm como padrão o tamanho de 1024 bytes, devido ao limite do datagrama do UDP descrito abaixo.

`timeo=n` Esta opção configura o tempo (em décimos de segundo) que o cliente NFS irá esperar por uma requisição completar. O valor padrão é de 0.7 segundos.

`hard` Marca o volume como uma montagem direta. É um valor padrão.

`soft` A montagem do volume é lógica (oposta à opção hard).

`intr` Permite que sinais do núcleo interrompam uma chamada NFS. Útil quando se quer interromper uma opção, ou seja quando o servidor não estiver respondendo por algum motivo, como por exemplo queda da rede.

Exceto para as opções `rsize` e `wsize`, todas as demais aplicam-se ao comportamento do cliente, caso o servidor fique inacessível temporariamente. Elas atuam em conjunto na seguinte situação: se o cliente envia um requisição ao servidor NFS, ele espera que a operação termine após um certo intervalo (especificado na opção `timeout`. Caso não seja recebida qualquer confirmação dentro do tempo predeterminado, ocorrerá a chamada *ultrapassagem de tempo menor* e a operação será repetida dentro do intervalo de tempo de espera definido. Ao se atingir 60 segundos sem resposta, ocorrerá uma *ultrapassagem de tempo maior*.

Por padrão, a ultrapassagem de tempo maior fará com que seja impressa uma mensagem na tela do cliente e todo o processo seja reiniciado, com um tempo de espera igual ao dobro do tempo anterior. Teoricamente isto poderá perpetuar-se eternamente. Volumes que ficam tentando uma operação até que o servidor se torne disponível são conhecidos como *montagem direta*[5]. O oposto, os volumes *montados pelo método simbólico*[6] geram um erro de E/S para o processo cliente quando ocorrer a "ultrapassagem do tempo de espera". Devido ao processo de "gravação atrasada" introduzido no buffer de cache de E/S, esta condição de erro não é informada ao processo cliente antes dele chamar a próxima função de gravação[7], fazendo com que o programa não possa garantir que uma operação de escrita em um volume montado simbolicamente foi concluída com sucesso.

O volume estar montado direta ou simbolicamente não é uma simples questão de gosto, mas tem muito a ver com o tipo de operação que se deseje efetuar neste volume. Por exemplo, caso se queira montar programas X via NFS, certamente não se gostaria que uma sessão X parasse, somente porque alguém interrompeu a rede, ou porque alguém tirou o cabo da placa Ehternet por um momento. Através da montagem direta de um volume, pode-se garantir que a estação irá esperar até que se restabeleça o contato com o servidor NFS. Por outro lado, dados não críticos como arquivos FTP, podem ser montados de forma simbólica, fazendo com que a sessão local não seja interrompida nos casos em que a máquina remota está inoperante ou temporariamente inacessível. Caso a conexão com o servidor seja de má qualidade ou utiliza um roteador sobrecarregado, pode-se incrementar o tempo de espera inicial usando a opção `timeo`, ou montar o volume de forma direta, porém permitindo interrupções por sinais às chamadas NFS, fazendo com que qualquer espera excessiva no acesso a um arquivo possa ser interrompida.

Normalmente, o servidor `mountd` acompanhará de alguma forma quais os diretórios

[5] hard-mount
[6] soft-mount
[7] write(2)

que estão montados e em quais máquinas. Esta informação pode ser apresentada através do programa `showmount`, incluído no pacote de aplicações NFS. No entanto o programa `mountd` do `Linux` pode também informar os volumes disponíveis via NFS através do comando `mount`, explicitado sem parâmetros.

11.3 Os Servidores NFS

Caso se queira prover serviços NFS para outras máquinas, ou seja tornar-se um servidor NFS, deve ser executado o programa `nfsd` e o servidor `mountd` na máquina local. Como acontece em programas baseados em RPC, eles não são gerenciados pelo servidor `inetd`, mas sim acionados durante a inicialização e auto-registrados no servidor portmapper. Entretanto tem-se que garantir que ele só será executado após o início da execução do programa `rpc.portmap`. Normalmente são incluídas as seguintes linhas no arquivo de inicialização `rc.inet2`:

```
if [ -x /usr/sbin/rpc.mountd ]; then
        /usr/sbin/rpc.mountd; echo -n " mountd"
fi
if [ -x /usr/sbin/rpc.nfsd ]; then
        /usr/sbin/rpc.nfsd; echo -n " nfsd"
fi
```

As informações referentes à propriedade dos arquivos que um servidor NFS proporciona aos clientes são compostas somente pelo número de identificação do usuário (*uid*) e do grupo (*gid*). Se tanto o cliente como o servidor associarem o mesmo nome de usuário e grupo a estes números de identificação, eles compartilham do mesmo espaço de usuários e grupos. Por exemplo, este é o caso quando se utiliza o NIS para distribuir as informações do arquivo `passwd` para todas as máquinas da rede.

No entanto, em algumas ocasiões os números não coincidem. Melhor do que ficar atualizando as identificações de clientes e grupos para coincidirem com os do servidor, pode-se utilizar o servidor de mapas `ugidd` para a execução desta tarefa. Utilizando-se a opção `map_daemon` explicada abaixo, pode-se indicar ao programa `nfsd` que estabeleça uma correspondência das identificações (uid/gid) do servidor com as dos equipamentos clientes, com a ajuda do programa `ugidd` executado na máquina cliente.

`ugidd` é um servidor baseado em RPC e é iniciado em `rc.inet2`, da mesma forma que `nfsd` e `mountd`.

```
if [ -x /usr/sbin/rpc.ugidd ]; then
        /usr/sbin/rpc.ugidd; echo -n " ugidd"
fi
```

11.4 O Arquivo `exports`

Enquanto as opções acima são aplicadas na configuração do cliente NFS, existe um diferente conjunto de opções que se aplicam ao servidor e que afetam o seu relacionamento com cada possível cliente. Estas opções são incluídas no arquivo `/etc/exports`.

Por padrão, o programa `mountd` não permite a ninguém que monte qualquer diretório de sua máquina, o que é uma atitude correta. Para permitir uma ou mais máquinas montarem um diretório via NFS, ele deve ser *exportado*, ou seja deve estar especificado no arquivo `exports`. Um exemplo deste arquivo é o seguinte:

```
# arquivo exports para aracaju
/home            blumenau(rw) maceio(rw) caxias(rw)
/usr/X386        blumenau(ro) maceio(ro) caxias(ro)
/usr/TeX         blumenau(ro) maceio(ro) caxias(ro)
/                blumenau(rw,no_root_squash)
/home/ftp        (ro)
```

Cada linha define um diretório e uma lista de máquinas que podem montá-lo. Um nome de máquina é normalmente indicado pelo nome totalmente qualificado, mas podem ser adicionadas as máscaras `*` e `?`, com ação idêntica a que elas têm no intepretador de comandos Bourne . Por exemplo: `lab*.pantanal.edu.br` coincide com `lab01.pantanal.edu.br` assim como também com `laber.pantanal.edu.br`. Se não for informado nenhum nome de máquina, como no exemplo acima do diretório `/home/ftp`, qualquer máquina tem permissão para montar este diretório.

Quando é feita a checagem de uma máquina cliente no arquivo `exports`, o programa `mountd` procurará o nome de máquina cliente usando a chamada `gethostbyaddr(2)`. Com o DNS, esta chamada retorna o nome canônico de máquina do cliente, criando assim a proibição da utilização de nomes alternativos de máquinas no arquivo `exports`. Sem o uso de DNS, o nome retornado é o primeiro nome de máquina encontrado no arquivo `hosts` que coincida com o endereço do cliente.

O nome da máquina pode ser seguido por uma lista de opções, entre parênteses e separadas por vírgulas. Estas opções têm os seguintes valores:

`insecure` Permite o acesso não autenticado a partir desta máquina.

`unix-rpc` Requer autenticação RPC (domínio UNIX) para esta máquina. Isto é requerido somente para as requisições originadas a partir de uma porta reservada Internet (isto é, portas com números menor que 1024). Esta opção está ativa por padrão.

`secure-rpc` Requer autenticação segura RPC para esta máquina. Isto ainda não foi implementado. Veja a documentação da Sun em "Secure RPC".

`kerberos` Requer autenticação Kerberos para acesso desta máquina. Isto ainda não está implementado. Veja a documentação do MIT sobre sistemas de autenticação Kerberos.

`root_squash` Esta é uma característica de segurança que proíbe que o superusuário dos servidores especificados tenha qualquer direito de acesso especial a partir de sua identificação igual a 0 no cliente, que será alterada no servidor para 65534 (-2). Esta identificação deve ser associada ao usuário `nobody`.

`no_root_squash` Não mapeia requisições do usuário com identificação 0. Esta opção é ativada por padrão.

`ro` Monta hierarquicamente os arquivos, somente para leitura. Esta opção é usada por padrão.

`rw` Monta hierarquicamente os arquivos, com autorizações para leitura e gravação.

`link_relative` Converte ligações simbólicas absolutas (onde a ligação começa com uma barra) em ligações relativas colocando os prefixos `../` que sejam necessários para obter a rota do diretório que contém a ligação para a raiz no servidor. Esta opção somente faz sentido quanto é montado um sistema de arquivos completo de uma máquina, onde algumas ligações podem apontar para arquivos inválidos, ou pior, para arquivos que nunca deveriam ser apontados. Esta opção é usada por padrão.

`link_absolute` Deixa todas as ligações simbólicas inalteradas (é a opção normal dos servidores NFS da Sun).

`map_identity` A opção `map_identity` indica ao servidor para assumir que o cliente usa as mesmas identificações de usuário e grupos que o servidor. Esta opção é usada por padrão.

`map_daemon` Esta opção avisa o servidor NFS para assumir que o cliente e o servidor não compartilham a mesma identificação de usuários e grupos. O servidor `nfsd` irá então construir uma lista da identificação de mapas entre cliente e servidor, através da chamada ao servidor `ugidd` na máquina cliente.

Durante a inicialização dos programas `nfsd` ou `mountd`, qualquer erro de análise do arquivo `exports` será relatado ao `servidor syslogd` com o nível de `aviso`.

Note que o nome de máquina é obtido a partir do endereçamento IP do cliente através do mapeamento reverso, devendo-se configurar corretamente o resolvedor de nomes. Caso se utilize o BIND e a segurança seja um item fundamental, deve-se habilitar a checagem de nomes falsos ("spoof") no arquivo `host.conf`.

11.5 O AutoMontador `Linux`

Às vezes pode ser contra indicado montar todos os volumes NFS que os usuários possivelmente queiram acessar. Devido ao grande número de volumes a serem montados, ou devido ao tempo que será utilizado na inicialização do sistema. Uma alternativa viável para isto é o utilitário *automounter*. Trata-se de um servidor que automática e transparentemente monta qualquer volume NFS sempre que for necessário e o desmonta quando eles não forem usados por um determinado período de tempo. Uma das coisas inteligentes do automontador é que ele pode montar alguns volumes a partir de locais alternativos. Por exemplo caso se queira manter cópias de programas X em duas ou três máquinas, pode-se especificar todas elas para serem montadas no diretório `usr/X386`; fazendo com que o automontador tente montar alguma delas até que consiga obter sucesso com alguma.

O programa automounter normalmente usado no `Linux` é conhecido como `amd`. Ele foi originalmente escrito por Jan-Simon Pendry e foi portado para o `Linux` por Rick Sladkey. A versão atual é a `amd-6.0b1`.

Explicar o programa `amd` está além do escopo deste capítulo. Para uma boa leitura sobre o tema, por favor veja nos fontes; eles contêm um arquivo texinfo com diversas informações detalhadas.

Capítulo 12

Gerenciando o Taylor UUCP

12.1 História

O UUCP foi desenvolvido no final dos anos setenta por Mike Lesk nos Laboratórios AT&T Bell para efetuar uma ligação de rede e notícias Usenet sobre uma linha discada. Uma vez que a maioria das pessoas deseja ter à sua disposição o correio eletrônico e notícias Usenet em máquinas domésticas conectadas através de modems, o UUCP ainda se mantém bastante popular. Apesar de haver muitas implementações sendo executadas em uma grande variedade de plataformas de hardware e sistemas operacionais, elas têm um alto nível de compatibilidade.

De qualquer forma, como muitos softwares que de alguma forma se tornaram um "padrão" através dos anos, não há uma versão que possa ser chamada de *o* programa UUCP. Na verdade há um incessante processo de evolução desde a primeira versão implementada em 1976. Atualmente, há duas linhas que diferem basicamente no seu suporte a hardware e na sua forma de configuração. A partir destas há várias derivações que diferem minimamente.

Uma das variações conhecida como "Versão 2 UUCP", a qual data de 1977, é uma implementação de Mike Lesk, David A. Novitz, e Greg Chesson. Apesar de já ser um pouco antiga, ainda é usada com bastante freqüência. Recentes implementações da Versão 2 disponibilizam muitas das facilidades das variações mais recentes do UUCP.

A segunda variação é mais recente, tendo sido desenvolvida em 1983 e é denominada BNU (Utilidades Básicas de Rede), HoneyDanBer UUCP, ou HDB, de forma abreviada. O nome é derivado dos nomes dos autores, P. Honeyman, D. A. Novitz e B. E. Redman. HDB foi concebido para eliminar algumas das deficiências da Versão 2 do UUCP. Por exemplo, novos protocolos de transferência foram adicionados e o diretório de trabalhos transitórios foi dividido, existindo agora um diretório para cada um dos diferentes sites com os quais se tenha tráfego UUCP.

A implementação atual do UUCP distribuída com o `Linux` é denominada Taylor UUCP 1.06,[1] na qual este capítulo está baseado. A versão Taylor UUCP 1.04 foi liberada em fevereiro de 1993. Além dos arquivos de configuração usuais, Taylor UUCP pode ainda ser compilado para utilizar os novos estilos de arquivos de configuração, conhecidos como "Taylor".

A versão 1.05 foi liberada recentemente e em breve estará disponível na maioria das distribuições. As diferenças entre versões normalmente afetam funcionalidades raramente utilizadas. Portanto é muito provável que se consiga configurar as novas versões com as informações existentes neste Guia.

O Taylor UUCP é normalmente compilado com compatibilidade BNU na maioria das distribuições Linux, no esquema de formato Taylor ou em ambos os formatos. Como este último é muito mais flexível e provavelmente mais simples de ser entendido do que os obscuros arquivos de configuração BNU, descreveremos o formato Taylor.

O propósito deste capítulo não é o de fornecer uma descrição exaustiva de quais opções podem ser usadas na linha de comandos do UUCP e o que elas fazem, mas sim apresentar uma descrição clara sobre a configuração de um nó funcional do UUCP. A primeira seção fornece uma introdução sobre a execução de implementações remotas do UUCP e transferências de arquivos. Caso você não seja um usuário iniciante com o UUCP, poderá passar diretamente para a seção Arquivos de Configuração UUCP, a qual explica os vários arquivos usados na configuração do UUCP.

De qualquer forma assumiremos que o leitor está familiarizado com o conjunto de programas de usuário UUCP. Há os programas `uucp` e `uux`. Para uma descrição completa, por favor referencie-se às páginas de manual destes comandos.

Apesar destes programas estarem amplamente disponíveis ao público em geral, o conjunto de programas de usuários `uux` e `uucp` contém uma série de comandos destinados somente ao uso administrativo. Eles são usados para monitorar o tráfego

[1] Desenvolvido e registrado por Ian Taylor, 1993.

UUCP através do nó, removendo antigos arquivos de histórico ou compilando estatísticas. Nenhum destes será aqui descrito, uma vez que são comandos periféricos às tarefas principais do UUCP. Além do mais, eles devem estar bem documentados, de uma forma relativamente simples de entendimento. Mais, há os programas que executam a maior parte do trabalho do UUCP. Eles são denominados `uucico` (onde cico significa copiar-para e copiar-de)[2] e `uuxqt`, o qual executará tarefas enviadas por sistemas remotos.

12.1.1 Maiores Informações Sobre o UUCP

Aqueles que não encontrarem neste capítulo tudo o que necessitem, devem ler adicionalmente a documentação que acompanha o programa: normalmente um conjunto de arquivos no formato texinfo que descrevem a configuração usando o esquema de Taylor[3]. Texinfo pode ser convertido para DVI e para arquivo info GNU usando os programas `tex` e `makeinfo`, respectivamente.

Caso se deseje usar o BNU ou mesmo os arquivos de configuração da Versão 2, há um livro muito bom sobre o tema chamado "Gerenciando UUCP e Usenet" ([OReilly89]). Particularmente eu o acho muito útil. Outra boa fonte de informações sobre o UUCP para Linux é o Como Fazer-UUCP de Vince Shakan, o qual é postado regularmente em `comp.os.linux.announce`.

Há ainda um grupo de notícias para discussão do UUCP, chamado `comp.mail.uucp`. Caso você tenha questões específicas sobre o Taylor UUCP, o local mais indicado para postar as suas perguntas é o grupo de discussão `comp.os.linux`.

12.2 Introdução

12.2.1 Transportadores UUCP e Execução Remota

Vital para o entendimento do UUCP é o conceito de *tarefa*. Cada transferência que um usuário inicia com o programa `uucp` ou `uux` é chamada de tarefa. Ela é constituída por um *comando* a ser executado no sistema remoto e uma coleção de *arquivos* a ser transferida entre sites. Uma das partes pode ser omitida, caso necessário.

[2]copy-in copy-out

[3]Através do comando `info` na linha de comandos.

Como exemplo, assumiremos que foi processado o seguinte comando na máquina local, o qual faz uma cópia UUCP do arquivo `guia.ps` para a máquina `sergipe` e executa o comando `lpr` para imprimir o arquivo.

```
$ uux -r sergipe!lpr !guia.ps
```

O UUCP geralmente não aciona o sistema remoto imediatamente para executar uma tarefa (caso assim fosse seria possível executar estas tarefas através do comando `kermit`). Ao invés disso ele temporariamente armazena a descrição da tarefa. Isso é chamado *spooling*. A árvore de diretórios com as tarefas são armazenadas em uma estrutura chamada *diretório de tarefas temporárias*[4] e está normalmente localizado em `/var/spool/uucp`. No nosso exemplo, a descrição da tarefa pode conter informações sobre o comando remoto a ser executado, denominado (`lpr`), o usuário que solicitou a sua execução e alguns outros itens. Adicionalmente à descrição da tarefa, o UUCP tem que armazenar o arquivo de entrada, neste caso o arquivo denominado chamado `guia.ps`.

A exata localização e nome dos arquivos temporários podem variar e dependem das opções em tempo de compilação. O UUCP compatível com HDB geralmente armazena os arquivos temporários em um diretório chamado `/var/spool/uucp/nome_do_site`, onde *`nome_do_site`* é o nome do site remoto. Quando compilado com a configuração Taylor, o UUCP irá criar subdiretórios sob um diretório de tarefas temporárias específico para o site, com diferentes tipos de arquivos temporários.

Em intervalos regulares, o UUCP disca para o sistema remoto. Quando a conexão com a máquina remota é estabelecida, ele transfere os arquivos descrevendo as tarefas a serem executadas, mais os arquivos de entrada. As tarefas destinadas à máquina local não serão executadas imediatamente, mas somente após o final da conexão. Isso é feito pelo programa `uuxqt`, o qual se encarrega também do reenvio de quaisquer tarefas designadas para outros sites.

Para distinguir entre tarefas mais e menos importantes, o UUCP associa um *índice* a cada uma delas. Ele é composto por uma letra, variando entre 0 a 9, A até Z, e a até z, em ordem decrescente. Tarefas com índice maior são transferidos prioritariamente. Mensagens costumam ser caracterizadas com índices B ou C, enquanto notícias recebem o índice N. Tarefas com índices maiores serão transferidas antes. Estes índices podem serem alterados através do indicador `-g` durante a execução dos comandos `uucp` ou `uux`.

[4] `spool`

Pode-se ainda desabilitar as transferências de tarefas abaixo de um determinado índice por certos períodos. Isso também é chamado de *índice máximo de arquivos temporários* permitidos durante uma conversação, cujo padrão é z. Cabe salientar que nestes casos os arquivos somente serão transferidos se tiverem um índice *igual ou maior* que o índice máximo para transferência.

12.2.2 O Trabalho Interno do `uucico`

◇ Para entender porque o `uucico` necessita conhecer certas coisas, uma rápida descrição de como ele se conecta a um sistema remoto será apresentada a seguir.

Ao se executar um comando `uucico -s` *`sistema`* a partir da linha de comandos, inicialmente deve-se ter os sistemas conectados fisicamente. As ações executadas dependem do tipo de conexão a ser estabelecida – por exemplo, ao se usar a linha telefônica, ele deve encontrar um modem e discar. Sobre TCP, deve executar a função `gethostbyname(3)` para converter o nome em um endereço de rede, descobrir qual porta deve ser aberta e associar o endereço com a conexão correspondente.

Após a conexão ter sido estabelecida, um procedimento de autorização deve ser executado. Ele geralmente consiste de um sistema remoto solicitando por um nome de acesso e possivelmente uma senha. Isso é comumente denominado *conversação de acesso.* O procedimento de autorização é executado pelo conjunto de utilitários `getty`/`login`, ou – em conexões TCP – pelo programa `uucico`. Caso a autorização seja bem sucedida, a máquina remota inicia o programa `uucico`. A cópia local do `uucico` a qual inicia a conexão é referenciada como *master*, a cópia remota é denominada *escrava.*

A seguir temos a *fase de negociação*: a master envia o nome da máquina e uma série de indicadores. A máquina escrava verifica se a máquina está autorizada a acessar o sistema local, enviar e receber arquivos, etc.. Os indicadores descrevem (entre outras coisas) o índice máximo de transferência de arquivos. Caso este esteja habilitado, um contador de conversação ou uma checagem do *número de seqüência de chamadas* assume o seu lugar. Com esta funcionalidade, ambos os sites mantêm um contador de conexões bem sucedidas, as quais são comparadas. Caso elas não coincidam, a negociação falhará. Isso é útil para proteger o sistema contra impostores.

Finalmente, os dois `uucico` tentam entrar em um acordo sobre o *protocolo de transferência.* Este protocolo gerenciará o modo de transferência dos dados, retransmitindo as informações em caso de erro. Há necessidade de suporte a diferentes protocolos, porque diferentes tipos de conexões são suportadas. Por exemplo, linhas telefônicas requerem um protocolo seguro, o qual tem uma postura "pessimista" sobre erros, enquanto uma transmissão TCP é naturalmente mais confiável e pode ser um protocolo mais eficiente com menor rigor na checagem de erros adicionais.

Após a finalização da negociação, a fase real de transmissão começa. Ambas as pontas ativam o programa de controle do protocolo selecionado. O programa possivelmente realiza uma seqüência específica de inicialização do protocolo.

Inicialmente, a máquina mestre envia todos os arquivos cujos índices podem ser transmitidos para o sistema remoto. Ao finalizar, ele informa à escrava que a transferência foi concluída, e que a escrava pode finalizar a ligação. Neste ponto há uma mudança de papéis: a remota torna-se a mestre e a máquina local se torna a escrava. A nova máquina mestre passa então a enviar seus arquivos. Ao finalizar, ambos os programas `uucico` trocam mensagens de finalização de transmissão e encerram a conexão.

Não entraremos em maiores detalhes: por favor verifique ou os fontes ou qualquer outro bom livro sobre UUCP. Há ainda um artigo muito interessante circulando pela Net, escrito por David A. Novitz, o qual fornece uma descrição detalhada do protocolo UUCP. O FAQ do Taylor UUCP discute ainda alguns detalhes de como o UUCP é implementado. Ele é postado regularmente em `comp.mail.uucp`.

12.2.3 Opções de Linha de Comando do `uucico`

Esta seção descreve as opções mais importantes da linha de comandos para o programa `uucico`. Para uma lista completa, por favor consulte a página de manual do comando `uucico(1)`.

`-s sistema` Disca para o *`sistema`* a menos que haja alguma restrição de horário de chamadas.

`-S sistema` Disca para o *`sistema`* informado, incondicionalmente.

`-r1` Inicia o programa `uucico` no modo mestre. Esta é a opção padrão quando as opções `-s` ou `-S` são informadas. A opção `-r1` faz com que o programa `uucico` tente acionar todos os sistemas descritos em `sys`, a menos que haja alguma restrição de horário de conexão.

`-r0` Inicia o programa `uucico` no modo escravo. Este é o padrão quando as opções `-s` ou `-S` *não* são informadas. No modo escravo, o sistema assume que tanto a entrada como a saída padrão estão conectadas à porta serial, ou à porta TCP especificada pela opção `-p` caso esta seja utilizada.

`-x tipo`, `-X tipo` Ativa o modo de depuração de uma determinada forma. Diversos tipos podem ser fornecidos simultaneamente através de uma lista, com os tipos separados por vírgulas. Os tipos válidos são os seguintes: `abnormal`, `chat`, `handshake`, `uucp-proto`, `proto`, `port`, `config`, `spooldir`, `execute`, `incoming`, `outgoing`. O tipo `all` provoca o acionamento de todas as opções. Por questões de compatibilidade com outras implementações do UUCP, o tipo pode ser especificado em um formato numérico, o qual ativa o tipo de depuração *n* na ordem da lista apresentada acima.

A depuração de mensagens será registrada no arquivo `Debug` no diretório `/var/spool/uucp`.

12.3 Arquivos de Configuração UUCP

Ao contrário dos programas de simples transferência de arquivos, o UUCP foi desenvolvido para ser capaz de gerenciar todas as transferências automaticamente. Uma vez que ele esteja devidamente configurado, uma ação diária do administrador não será necessária. As informações necessárias serão mantidas em alguns *arquivos de configuração* que residem no diretório `/usr/lib/uucp`. Muitos destes arquivos serão usados somente no momento da discagem.

12.3.1 Introdução ao Taylor UUCP

Afirmar que a configuração do UUCP é difícil pode ser uma questão de falta de entendimento. Este é um tema polêmico, porém é inegável que alguns formatos complexos de arquivos de configuração não tornam as coisas mais simples (apesar dos formatos de arquivos Taylor serem simples se comparados aos antigos formatos HDB ou Versão 2).

Para fornecer um sentimento de como estes arquivos interagem, apresentaremos os mais importantes e alguns exemplos. Não explicaremos detalhadamente cada um deles, onde uma descrição mais aprofundada será apresentada nas seções posteriores a seguir. Caso se deseje configurar uma máquina com UUCP, o melhor caminho será começar pelos arquivos de exemplos, adaptando-os gradualmente. Pode-se escolher entre os apresentados abaixo e aqueles incluídos na sua distribuição Linux.

Todos os arquivos descritos nesta seção são mantidos em /usr/lib/uucp ou em um subdiretório específico. Algumas distribuições específicas contêm os binários UUCP que suportam tanto os arquivos de configuração HBD como o formato Taylor, e usam diferentes subdiretórios para cada conjunto de arquivos de configuração. Normalmente haverá um arquivo README no diretório /usr/lib/uucp.

Para que o UUCP funcione adequadamente, estes arquivos devem ter como dono o usuário uucp. Alguns deles contêm senhas e números de telefones e adicionalmente têm as permissões configuradas em 600.[5]

O arquivo principal de configuração é denominado /usr/lib/uucp/config e é usado para a configuração dos parâmetros gerais. O mais importante deles (e por enquanto o único) é o nome da máquina UUCP local. Como na Cervejaria Virtual, eles usam aracaju como o caminho padrão do UUCP teremos o seguinte:

```
# /usr/lib/uucp/config - arquivo principal de configuração UUCP
hostname         aracaju
```

O próximo arquivo de configuração é denominado sys. Ele contém todas as informações específicas dos sites com os quais as conexões serão estabelecidas. Isso inclui o nome do site, informações da conexão, como número do telefone ao se usar uma conexão via modem, etc.. Uma entrada típica para uma conexão com o site chamado parintins tem o seguinte formato:

```
# /usr/lib/uucp/sys - nome dos vizinhos UUCP
# sistema: parintins
system          parintins
time            Any
phone           123-456
port            serial1
speed           38400
chat            ogin: aracaju ssword: pororoca
```

A palavra chave port indica a porta que será usada e time especifica o horário em

[5]Note que muitos comandos UUCP devem ter o setuid configurado para uucp, pois de outra forma os usuários serão capazes de acessar as senhas de terceiros, mesmo com as permissões configuradas para 600.

que a conexão pode ser estabelecida. `chat` descreve os programas de conversação de acesso – a seqüência de caracteres que deve ser trocada para permitir que o `uucico` acesse a máquina `parintins`. Retornaremos ao programa de conversação posteriormente. O comando `port` não contém o nome de um arquivo especial de dispositivo como `/dev/cua1`, mas sim o nome de uma entrada do arquivo `port`. Pode-se definir os nomes das portas da forma que se queira, desde que eles estejam presentes no arquivo `port`.

O arquivo `port` mantém informações específicas sobre a conexão. Para ligações via modem, ele descreve o arquivo especial de dispositivos a ser usado, a faixa de velocidade suportada e o tipo de discagem do equipamento conectado à porta. A entrada abaixo descreve a porta `/dev/cua1` (também conhecida como COM 2), na qual um modem US Robotics está conectado e é capaz de chegar a velocidades de até 38400bps. O nome da entrada foi escolhido para coincidir com o nome da porta definida no arquivo `sys`.

```
# /usr/lib/uucp/port - portas UUCP
# /dev/cua1 (COM2)
port            serial1
type            modem
device          /dev/cua1
speed           38400
dialer          usrobotics
```

As informações pertinentes às discagens são mantidas em outro arquivo, chamado – `dial`. Para cada tipo de discagem ele contém basicamente a seqüência dos comandos necessários, exceto o número do telefone. Novamente isso é especificado como um programa de conversação. Por exemplo uma entrada para o arquivo acima poderia ter o seguinte conteúdo:

```
# /usr/lib/uucp/dial - informações de discagem
# modems usrobotics
dialer          usrobotics
chat            "" ATZ OK ATDT\T CONNECT
```

A linha começando com a palavra `chat` especifica uma conversação com o modem, a qual na verdade é uma seqüência de comandos enviados e recebidos do modem, para inicializá-lo e fazer com que ele disque para o número indicado. A expressão "`\T`" será substituída pelo número do telefone pelo programa `uucico`.

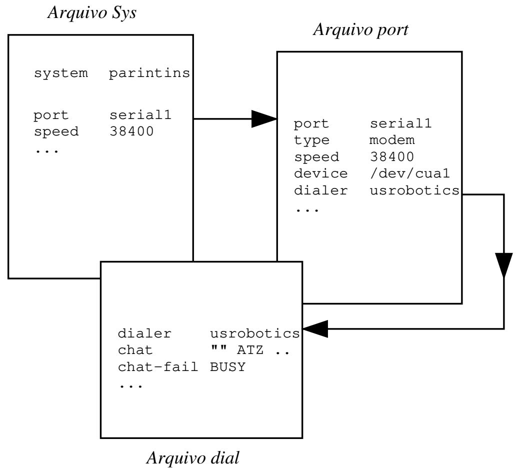

Figura 12.1: Interação dos Arquivos de Configuração do Taylor UUCP

Para ter-se uma idéia de como o programa `uucico` lida com estes arquivos de configuração, vamos assumir que foi executado o seguinte comando:

```
$ uucico -s parintins
```

na linha de comando. A primeira ação de `uucico` será procurar a máquina `parintins` no arquivo `sys`. A partir de uma entrada para `parintins` neste arquivo, ele verificará que deverá utilizar a porta de nome `serial1` para estabelecer a conexão. Este nome por sua vez indica que esta é uma porta que tem um modem UsRobotics conectado a ela.

`uucico` pesquisará agora o arquivo `dial` por uma entrada descrevendo o modem usrobotics e após encontrá-la, abrirá o arquivo da porta serial `/dev/cua1` e executará o programa de conversação. Este por sua vez enviará a expressão "`ATZ`", aguarda "`OK`" como resposta, etc.. Ao encontrar a expressão "`\T`", a substituirá pelo número de telefone (123-456) extraído do arquivo de configuração `sys`.

Após o modem retornar a expressão `CONNECT`, a conexão estará estabelecida e o programa de conversação do modem estará concluído. `uucico` agora retorna ao programa `sys` e executa a conversação de acesso ao sistema. No nosso exemplo, ele aguardará o indicativo "`login:`", enviando a seguir o nome de usuário (`aracaju`), aguardará pela expressão "`password:`", e enviará a senha, "`pororoca`".

Após completar a autorização, o ponto remoto inicia a execução de seu próprio programa `uucico`. Ambos irão entrar na fase de negociação descrita na seção anterior.

A forma como os arquivos de configuração dependem uns dos outros é descrito na figura 12.1.

12.3.2 O que o UUCP Necessita Saber

Antes de se iniciar a gravação dos arquivos de configuração do UUCP, há algumas informações que se fazem necessárias.

Primeiro, deve-se compreender como as interfaces seriais do modem estão conectadas. Normalmente as portas (DOS) COM1 a COM4 estão mapeadas para arquivos especiais de dispositivos denominados `/dev/cua0` até `/dev/cua3`. Muitas distribuições criam uma ligação simbólica de `/dev/modem` para o arquivo de dispositivos adequado e configuram programas como `kermit`, `seyon`, etc, para que usem este arquivo genérico. Neste caso, pode-se usar o arquivo `/dev/modem` na configuração do UUCP também.

A razão para o uso de uma ligação simbólica se deve à existência de *arquivos de reserva de recursos* gerados por programas que executam discagens, indicando que a porta serial está em uso. O nome desses arquivos é gerado a partir de uma concatenação da expressão `LCK..` com o nome do arquivo do dispositivo, por exemplo `LCK..cua1`. Caso algum programa use um nome diferente para o mesmo dispositivo, eles falharão no reconhecimento dos arquivos de reservas. Como conseqüência, eles irão corromper as sessões uns dos outros quando inicializadas ao mesmo tempo. Isso pode não ser tão incomum quando se configuram as chamadas UUCP usando-se uma entrada no arquivo `crontab`.

Para maiores detalhes sobre a configuração de portas seriais, por favor verifique o capítulo 4.

A seguir, deve-se descobrir a velocidade do modem e o `Linux` finalmente irá restabelecer a comunicação. Deve-se configurar este parâmetro para a taxa de transferência máxima que se puder obter. Ela pode ser muito maior do que a taxa nominal descrita pelo modem. Por exemplo, muitos modems enviam e recebem informações a taxas de 28,8Kbps (bits por segundo). Usando-se protocolos de compressão como V.42bis, a taxa de transferência pode chegar a valores bem superiores (3, 4, 5 vezes maior, dependendo das condições).

Obviamente, para que o UUCP possa funcionar é necessário ter-se o número da máquina remota a ser contactada, assim como um usuário de acesso válido e possivelmente uma senha.[6]

É preciso saber ainda *exatamente* como acessar o sistema remoto. Por exemplo, é necessário pressionar a tecla BREAK antes de que o indicativo de acesso ao sistema seja apresentado? É apresentada a expressão `login:` ou `user:`? Estas informações são necessárias para a composição do *programa de conversação*, o qual indica ao `uucico` como acessar o sistema remoto. Caso você não saiba, ou o programa tradicional não funcione, tente discar para o sistema remoto com um programa de emulação de terminal como o `kermit` ou `minicom` e anote exatamente o comportamento do sistema remoto.

12.3.3 Nomeando Sites

Assim como em uma rede baseada em TCP/IP, a máquina deve ter um nome na rede UUCP. Para simplesmente transferir arquivos para ou de uma máquina diretamente utilizando o UUCP, o nome da máquina não necessita obedecer a nenhum padrão.[7]

Porém ao se usar o UUCP para a transferência de mensagens ou notícias, deve-se analisar a possibilidade de registrar o nome da máquina junto ao Projeto de Mapeamento UUCP. Ele é descrito no capítulo 13. Mesmo ao se participar de um domínio, deve-se considerar a obtenção de um nome oficial UUCP para o site.

Freqüentemente pessoas escolhem o nome UUCP de uma máquina para que coincida com o primeiro componente do nome totalmente qualificado. Supondo-se que uma máquina tenha o nome qualificado de `santos.spaulo.com.br`, seu nome UUCP de máquina poderia ser `santos`. Obviamente, pode-se usar um nome UUCP sem qualquer relação com o nome qualificado.

Esteja seguro de não utilizar um nome não qualificado em endereços de mensagens, a menos que o registre como o seu nome UUCP oficial.[8]

[6] Caso se esteja somente testando o UUCP, pode-se utilizar sites públicos, que fornecem usuários e senhas de conhecimento geral. Na maioria dos casos eles se chamam `uucp/uucp` ou `nuucp/uucp`.

[7] A única limitação refere-se ao tamanho do nome da máquina, que não deve exceder 7 caracteres, para não ser confundido em sistemas de arquivos que impõem nomes menores em arquivos ou máquinas.

[8] O Projeto de Mapeamento UUCP registra todos os nomes de máquinas UUCP em todo o mundo, garantindo que eles sejam únicos. Para registrar um nome UUCP, solicite aos mantene-

Ao se enviar uma mensagem para uma máquina não registrada, este desaparecerá em algum buraco negro ou lata de lixo da rede. Caso o nome já esteja em uso por algum outro site, ele será roteado para o que estiver registrado, provocando um bocado de trabalho ao administrador da rede.

Por padrão, o conjunto de ferramentas UUCP utiliza o nome definido em `hostname` como seu nome UUCP. Este nome é comumente configurado pelo programa `/etc/rc.local`. Caso o nome UUCP da máquina seja diferente daquele, deve-se usar a opção `hostname` no arquivo `config` para indicar ao `uucico` o seu nome UUCP, conforme descrito a seguir.

12.3.4 Arquivos de Configuração de Taylor

Retornaremos agora aos arquivos de configuração. O Taylor UUCP obtém as suas informações a partir dos seguintes arquivos:

`config` Mantém a configuração principal do UUCP. Pode-se definir o nome do site UUCP aqui.

`sys` Este arquivo descreve todos os sites conhecidos pela máquina local. Para cada site, ele especificará um nome, a que horas deve ser estabelecida uma conexão, qual o número de discagem (caso necessário), o tipo de dispositivo que deve ser usado e quanto tempo pode-se estar conectado.

`port` Contém entradas descrevendo cada porta disponível, junto com a velocidade de linha suportada e o arquivo de discagem a ser utilizado.

`dial` Descreve os parâmetros de discagem a serem utilizados em uma conexão telefônica.

`dialcode` Contém expansões de códigos de discagem simbólicos.

`call` Contém o nome de acesso e a senha a serem usados ao se discar para um sistema. Raramente é utilizado.

`passwd` Contém o nome de acesso e a senha que os sistemas remotos podem utilizar para acessar a máquina local. Este arquivo é usado somente quando o programa `uucico` executa a sua própria verificação de senhas.

dores do site que administra as suas mensagens, eles serão capazes de auxiliá-lo.

Os arquivos de configuração de Taylor são geralmente constituídos de linhas contendo pares de palavras chave. Um símbolo de número (#) indica a existência de um comentário até o final da linha. Caso seja necessário utilizar este sinal com outro significado, ele deverá ser precedido por uma barra.

Há algumas opções que podem ajustar estes arquivos de configuração. Não descreveremos todos os parâmetros aqui, mas abordaremos somente os mais importantes. Ele serão capazes de configurar uma ligação baseada em modem. Seções adicionais descreverão modificações que se fizerem necessárias para se usar UUCP sobre TCP/IP ou sobre uma linha serial direta. Uma referência completa pode ser encontrada nos documentos Texinfo que acompanham os fontes do UUCP.

Quando se supor que o sistema UUCP esteja completamente configurado, pode-se checar a configuração utilizando-se a ferramenta `uuchk` localizada em `/usr/lib/uucp`. Este programa lê os arquivos de configuração e imprime um relatório detalhado dos valores de configuração usados em cada sistema.

12.3.5 O Arquivo `config` - Opções Gerais de Configuração

Normalmente este arquivo não é utilizado para descrever muito mais que o nome da máquina. Por padrão, UUCP utilizará o nome definido pelo comando `hostname`, mas geralmente é uma boa idéia configurar o nome explicitamente no UUCP. Um exemplo do arquivo pode ser visto a seguir:

```
# /usr/lib/uucp/config - arquivo de configuração principal do UUCP
hostname        aracaju
```

Há ainda um número de parâmetros diversos que podem ser aqui configurados, como o nome do diretório de tarefas temporárias ou os direitos de acesso para UUCP anônimo. Este último será discutido em uma seção posterior.

12.3.6 O Arquivo `sys` - Como Dizer ao UUCP Sobre os Outros Sistemas

O arquivo `sys` descreve os sistemas remotos que a máquina deve conhecer. Uma entrada é definida através da palavra chave `system`, as linhas subseqüentes até a próxima diretiva `system` detalham os parâmetros específicos do site. Comumente, uma entrada de sistema irá definir parâmetros como número de telefone e conversação de acesso.

Parâmetros antes da definição do primeiro sistema configuram valores padrões válidos para todos os sistemas definidos no arquivo. Normalmente, os parâmetros de protocolo são aqui definidos.

Abaixo, os parâmetros mais importantes são discutidos em detalhes:

Nome do Sistema

O comando `system` define o sistema remoto. Deve-se especificar o nome corrente do sistema remoto, não um apelido, porque o programa `uucico` irá checar este parâmetro.[9] Cada nome de sistema pode aparecer somente uma vez no arquivo. Caso deseje-se usar diferentes conjuntos de configurações para o mesmo sistema (como por exemplo números diferentes de telefone que o programa `uucico` deve tentar), pode-se especificá-los no parâmetro *alternate*, que é descrito a seguir.

Número de Telefone

Caso o sistema remoto seja alcançado através de uma linha telefônica, o campo `phone` especifica o número de telefone que deve ser usado. Ele pode conter diversos caracteres interpretados pelo procedimento de discagem do `uucico`. Um sinal de igual significa que deve ser esperado um tom de discagem secundário e um traço gera uma pausa de um segundo. Por exemplo, algumas instalações não conseguirão efetivar as ligações caso não haja uma pausa entre o código DDD e o número do telefone. Existem os casos ainda onde é necessária a discagem de 0 ou 9 para que centrais telefônicas disponibilizem uma linha externa.

Qualquer expressão alfabética pode ser usada para esconder informações dependentes do site, como por exemplo o DDD. Qualquer expressão é traduzida para um código de discagem através do arquivo `dialcode`. Suponha que temos o seguinte arquivo `dialcode`:

```
# /usr/lib/uucp/dialcode - tradução de códigos de discagem
PortoAlegre         0XX51382
Curitiba            0XX41332
```

Onde XX representa o número da prestadora (que deve ser colocado). Com estas traduções, pode-se usar um número de telefone no arquivo `sys`, como por exemplo

[9] Versões mais antigas do UUCP Versão 2 não propagam os seus nomes quando são chamados, ainda que outras implementações o façam, como por exemplo o Taylor UUCP.

PortoAlegre7732, o qual pode tornar as coisas um pouco mais legíveis.

Porta e Velocidade

As opções port e speed são usadas para selecionar o dispositivo usado para acessar o sistema remoto, e a velocidade máxima que deve ser configurada.[10] Uma entrada de system pode usar somente uma ou ambas as opções. Ao procurar por um dispositivo adequado no arquivo port, somente serão selecionadas as portas que têm um nome coincidente e/ou uma velocidade similar.

Geralmente, o uso da opção speed deve ser suficiente. Caso se tenha somente um dispositivo serial definido em port, uucico irá sempre escolher o correto. Logo faz-se necessário somente configurar a velocidade desejada. Caso existam vários modems conectados ao sistema, freqüentemente não será necessário definir uma porta em particular, porque o programa uucico encontrará aquelas que se adeqüem aos parâmetros definidos e testará cada dispositivo para encontrar algum que esteja disponível.

A Conversação de Acesso

Conforme descrito anteriormente o programa de conversação de acesso descreve como o programa uucico deve acessar o sistema remoto. Ele consiste em uma lista de convenções especificando expressões esperadas e enviadas pelo processo local uucico. O objetivo é fazer com que o programa uucico aguarde até que a máquina remota envie o indicativo de acesso ao sistema, e então envie o nome para acesso, aguarde que o sistema remoto envie o indicativo de senha, enviando então a senha de acesso. Aguardar e enviar dados são passos alternados que fazem parte do programa uucico. Este automaticamente anexa um caractere de retorno de carro (\r) a qualquer caractere enviado. Um programa de conversação de acesso simples terá o seguinte aspecto:

```
ogin:  aracaju ssword:  pirapora324
```

Note que os campos esperados não contêm os indicativos completos. Este procedimento dá maior segurança ao procedimento, uma vez que este independe se o sistema remoto utiliza o indicador de acesso ao sistemas com maiúsculas (por ex. Login:) ou com minúsculas(por ex. login:).

[10] A taxa de transmissão do tty deve ser no mínimo igual à velocidade máxima de transferência.

O programa `uucico` permite a construção de execuções condicionais. Por exemplo caso a máquina remota `limeira` necessite ser inicializada antes de enviar o indicativo de acesso, pode-se adicionar um subprograma a uma expressão esperada, normalmente através de um traço. O subprograma é executado somente se o principal falhar, como por exemplo pela ultrapassagem do tempo de espera definido. Uma das finalidades, voltando ao nosso exemplo, é o envio de um comando BREAK para um sistema remoto que não apresente o indicativo de acesso. O seguinte exemplo fornece uma programa de conversação que soluciona o problema, caso seja necessário pressionar por exemplo `return` para que o indicativo seja disponibilizado. `""` diz a UUCP para não esperar por nada e que continue para a próxima expressão imediatamente.

```
"" \n\r\d\r\n\c ogin:-BREAK-ogin: aracaju ssword: pirapora
```

Há ainda alguns caracteres especiais e caracteres de fuga que podem ocorrer em um programa de conversação. A seguir apresentamos uma lista parcial dos caracteres aceitos em expressões esperadas:

`""` Significa vazio, ausência de qualquer caractere. Indica ao programa `uucico` para não esperar por nenhuma informação, e sim que deve proceder o próximo envio de expressão imediatamente.

`\t` Caractere Tab.

`\r` Caractere de retorno de carro.

`\s` Caractere de espaços. Deve ser usado para anexar espaços à uma expressão de conversação.

`\n` Caractere de nova linha.

`\\` Caractere de barra.

No envio de expressões, os seguintes caracteres de fuga e expressão são aceitos em adição aos acima descritos:

`EOT` Caractere de fim de transmissão (`^D`).

`BREAK` Caractere de reinicialização.

`\c` Suprime o envio de retorno de linha ao final da expressão.

`\d` Produz uma pausa de um segundo no envio.

`\E` Habilita a verificação de eco. Requer que o programa `uucico` aguarde o eco de tudo o que for escrito na linha, ou seja, ele primeiramente recebe de volta as expressões enviadas antes de continuar. É bastante útil quando usado em conversações de modems (a qual encontramos a seguir). Não estará ativo por padrão.

`\e` Desabilita a verificação de eco.

`\K` O mesmo que `BREAK`.

`\p` Pausa de uma fração de segundo.

Alternativas

Algumas vezes é desejável ter-se múltiplas entradas para um único sistema, por exemplo para se discar para diferentes números de conexão com um determinado sistema remoto. Com o Taylor UUCP isso pode ser realizado através da opção *alternate*.

Uma entrada alternativa retém todos os padrões de configurações da entrada principal e especifica somente aquelas que devem ser alteradas ou adicionadas. Uma alternativa é iniciada para o sistema nas linhas seguintes à especificação da opção.

Para utilizar dois números de telefones para acessar a máquina `parintins`, pode-se modificar a entrada no arquivo `sys` para o seguinte:

```
system       parintins
phone        123-456
... demais entradas ...
alternate
phone        123-334
```

Ao discar para `parintins`, `uucico` inicialmente irá discar 123-456 e caso este falhe irá utilizar o número de telefone alternativo. Esta entrada retém todas as configurações realizadas anteriormente e somente sobrepõem o número do telefone.

Restringindo os Horários de Conexões

O Taylor UUCP provê diversas formas de restringir os horários de realização de chamadas para um sistema remoto. Pode-se usar este procedimento devido a limitações de uso do sistema remoto em horários comerciais ou fora deles, ou simplesmente para evitar tarifas telefônicas mais caras, por exemplo. Note que sempre é possível alterar as restrições de horário de chamadas através das opções `-S` ou `-f` do programa `uucico`.

Por padrão, o Taylor UUCP desabilitará conexões a qualquer tempo, fazendo com que seja *necessário* algum tipo de especificação de horário no arquivo `sys`. Caso não haja nenhuma restrição de horário, deve-se especificar a opção `time` com um valor igual a `Any` no arquivo `sys`.

A forma mais simples de se restringir o horário de acesso é através da opção `time`, o qual é seguido por uma expressão composta pelos campos dia e horário. Dia pode ser igual a `Mo, Tu, We, Th, Fr, Sa, Su`[11] ou composto por uma combinação destes, ou `Any`, `Never`, ou `Wk` para dias de semana. O tempo consiste de dois relógios marcando de 0 a 24 horas, separados por um hífen. A combinação destas convenções deve ser especificada sem espaços entre elas, podendo ser especificada qualquer quantidade de entradas, devendo estar separadas por vírgulas. O exemplo a seguir

```
time            MoWe0300-0730,Fr1805-2000
```

permite chamadas às Segundas e Quartas, das 3 da manhã até as 7.30 horas e nas Sextas entre 18.05 e 20.00 horas. Quando um campo de horário ultrapassa a meia noite, digamos `Mo1830-0600`, ele na realidade significa Segunda-feira, entre meia-noite e 6 da manhã e 18.30 até a meia-noite.

As expressões especiais de horário `Any` e `Never` significam que as conexões podem ser efetuadas em qualquer horário ou em nenhum, respectivamente.

O comando `time` pode ser especificado com um segundo argumento que descreve o intervalo de tentativas em minutos. Quando uma tentativa de estabelecimento de conexão falha, `uucico` não irá tentar uma nova conexão com o sistema remoto durante um determinado intervalo. Por padrão, `uucico` usa um esquema exponencial para definir o intervalo a ser usado, aonde o intervalo cresce após cada falha

[11] Representam Segunda-feira, Terça-feira, Quarta-feira, Quinta-feira, Sexta-feira, Sábado e Domingo, respectivamente.

ocorrida. Por exemplo, ao se especificar um intervalo de 5 minutos, `uucico` irá se recusar a tentar uma nova conexão em um intervalo menor que 5 minutos após a última falha.

A opção `timegrade` permite a definição das tarefas que serão executadas em determinado horário. Por exemplo, imaginemos que temos um sistema com os seguintes comandos `timegrade` em uma entrada `system`:

```
timegrade          N Wk1900-0700,SaSu
timegrade          C Any
```

Essa definição permite que as tarefas com índices C ou maiores (normalmente as mensagens enfileiradas de correio utilizam índices B ou C) sejam transferidas em qualquer conexão que seja realizada, enquanto que as notícias (normalmente com o índice N) serão transferidas somente à noite ou nos finais de semana.

Assim como em `time`, o comando `timegrade` tem um terceiro parâmetro opcional que define o intervalo de tentativas em minutos.

Existem porém algumas deficiências sobre os índices de ordem: primeiro, a opção `timegrade` aplica-se somente ao que o sistema *local* envia, o sistema remoto ainda pode transferir qualquer coisa que ele queira. Neste caso deve-se usar a opção `call-timegrade` para explicitamente definir as tarefas que podem ser recebidas, porém não há garantias de que a máquina remota irá obedecer.[12]

Similarmente o campo `timegrade` não é checado pelo sistema remoto, quando ele inicia uma conexão, sendo que qualquer tarefa enfileirada para a máquina será executada. De qualquer forma o sistema remoto pode explicitamente requisitar que o programa `uucico` local restrinja as transferências dentro de certos índices.

12.3.7 O Arquivo `port` - O Que São Dispositivos Seriais

O arquivo `port` indica ao programa `uucico` as portas disponíveis. Elas podem ser modems, porém outros tipos de linhas seriais e conexões TCP são também suportadas.

Assim como o arquivo `sys`, o arquivo `port` consiste de entradas separadas começando com a palavra chave `port`, seguida pelo nome da porta. Este nome deve ser usado pelo comando `port` do arquivo `sys`. O nome não precisa ser único, caso

[12] Caso o sistema remoto execute o Taylor UUCP ele funcionará.

haja diversas portas com mesmo nome, `uucico` testará cada uma delas até que uma se adeqüe à sua necessidade e esteja liberada.

O comando `port` deve ser imediatamente seguido pela opção `type`, a qual indica o tipo de porta a ser descrita. Tipos válidos são `modem`, `direct` para conexões diretas e `tcp` para conexões TCP. Caso o comando `port` não esteja presente, será assumido o tipo padrão: modem.

Nesta seção cobriremos somente portas com modem. Portas TCP e diretas serão discutidas posteriormente.

Para modem e portas diretas, deve ser especificado o dispositivo para chamadas usando-se a diretiva `device`. Normalmente, este é o nome de um dispositivo especial no diretório `/dev`, como por exemplo `/dev/cua1`.[13]

No caso de modems, a entrada deve determinar também o tipo de modem que está conectado à porta. Diferentes tipos de modem devem ser configurados diferentemente. Mesmo modems que se dizem compatíveis com o padrão Hayes podem não ser necessariamente compatíveis uns com os outros. De qualquer forma há que se indicar para o programa `uucico` como inicializar o modem e como discar para o número desejado. O Taylor UUCP mantém as descrições de todas as discagens em um arquivo denominado `dial`. Para usar qualquer um deles, deve-se especificar o nome da discagem que será usada através do comando `dialer`.

Algumas vezes, pode-se desejar usar um modem de forma diferente, dependendo de qual sistema deseje-se acessar. Por exemplo, alguns modems antigos não conseguem entender a tentativa de um modem de alta velocidade de conectar-se a 56 Kbps; eles simplesmente desconectam a linha ao invés de negociar a velocidade de conexão. Ao se conhecer esta situação, deve-se ter uma configuração diferente ao se conectar com este site. Para tanto, deve-se definir uma entrada adicional para a porta no arquivo `port` que especifique uma discagem diferenciada. Agora pode-se fornecer à porta um nome diferente, como por exemplo `serial1-28800` e usar-se a diretiva `port` para a entrada da máquina mais lenta, que podemos chamar, por exemplo, de `28800` no arquivo `sys`.

Uma forma mais adequada é distinguir as portas através das velocidades que elas suportam. Por exemplo, as duas portas de entrada para a situação acima podem ter a seguinte aparência:

```
# UsRobotics; conexão a alta velocidade
```

[13] Algumas pessoas usam os dispositivos `ttyS*`, quando se pretende utilizar somente recepção de ligações de acesso discado.

```
port            serial1         # nome da porta
type            modem           # porta do tipo modem
device          /dev/cua1       # COM2
speed           56000           # velocidade suportada
dialer          usrobotics      # discagem normal

# UsRobotics; conexão à baixa velocidade
port            serial1         # nome da porta
type            modem           # porta do tipo modem
device          /dev/cua1       # COM2
speed           28800           # velocidade suportada
dialer          usr-28800       # não tenta conexões mais rápidas
```

A entrada no sistema para o site 28800 deve usar o nome de porta `serial1`, conectando-se à velocidade de 28800 bps somente. O `uucico` irá automaticamente usar a segunda porta. Todos os demais sites que utilizem a velocidade 56000 bps no sistema utilizarão a entrada que será acionada através da primeira entrada da porta serial1.

12.3.8 O Arquivo `dial` - Como Discar

O arquivo `dial` descreve como diversas formas de discagem são usadas. Tradicionalmente, o UUCP conversa com discadores e não com modems, uma vez que nos primórdios de seu desenvolvimento eram comuns sites terem serviços de discagem automática servindo um banco de modems. Hoje muitos desses modems têm suporte interno à recepção de chamadas, tornando a distinção um pouco difusa.

De qualquer forma, discadores ou modems diferentes podem requerer diferentes configurações. Pode-se descrever cada um deles no arquivo `dial`. Entradas no arquivo `dial` começam com o comando `dialer` que fornece o nome da discagem.

A mais importante entrada além desta é a conversação de modem, especificada pelo comando `chat`. Similar à conversação de acesso, ela consiste de uma seqüência de expressões enviadas pelo programa `uucico` para o discador e as respostas que ele espera receber de retorno. É comumente usado para inicializar o modem com alguns status conhecidos e para discar o número desejado e na definição das respostas que serão recebidas em retorno. O exemplo a seguir mostra uma típica configuração de conversação com um modem compatível com o padrão Hayes:

```
# modem USR; conexão de alta velocidade
```

```
dialer          usrobotics          # nome da discagem
chat            "" ATZ OK\r ATH1E0Q0 OK\r ATDT\T CONNECT
chat-fail       BUSY
chat-fail       ERROR
chat-fail       NO\sCARRIER
dtr-toggle      true
```

A conversação com o modem começa com `""`, a expressão de espera vazia. `uucico` irá então enviar o primeiro (`ATZ`) logo em seguida. `ATZ` é o comando Hayes para reinicializar o modem. Ele então aguardará o modem enviar a expressão `OK` e logo após enviará o próximo comando o qual desliga o eco local. Após o modem retornar a resposta `OK` novamente, `uucico` enviará o comando de diálogo (`ATDT`). A seqüência de fuga `\T` nesta expressão é substituída pelo número de telefone recebido da entrada de sistema do arquivo `sys`. `uucico` aguardará pela expressão `CONNECT`, a qual sinaliza que a conexão com a máquina remota foi estabelecida com sucesso.

Freqüentemente o modem falha ao tentar conectar-se a um sistema remoto, por exemplo caso o sistema já esteja conectado a outro sistema e a linha esteja ocupada. Neste caso o modem retornará alguma mensagem de erro indicando a razão. Conversações com modems não são capazes de detectar tais mensagens. `uucico` ficará aguardando até que uma resposta esperada seja recebida ou seja ultrapassado o tempo de espera. O arquivo de registro do UUCP irá somente mostrar uma mensagem "tempo estourado no arquivo de conversação" ao invés da verdadeira razão.

De qualquer forma, o Taylor UUCP permite comunicar ao `uucico` as mensagens de erro usando-se o comando `chat-fail` descrito acima. Quando `uucico` detecta uma expressão de falha de conversação durante a execução de um programa de conversação ele finaliza a chamada e registra a mensagem de erro no arquivo de registros de ocorrências do UUCP.

O último comando mostrado acima avisa ao UUCP para alternar a linha para o modo DTR antes de iniciar a conversação com o modem. Muitos comandos podem ser configurados para permanecerem conectados ao detectarem uma mudança na linha DTR e entrarem em modo de comando.[14]

[14] Pode-se ainda configurar alguns modems para reinicializarem-se ao detectar a transição para o modo DTR. Alguns deles, de qualquer forma parecem não executar esta operação e ocasionalmente desligam a conexão.

12.3.9 UUCP Sobre TCP

Por mais absurda que possa parecer à primeira vista, utilizar UUCP para transferir dados sobre TCP não é uma má idéia, especialmente ao se transferir grandes quantidades de notícias Usenet. Em ligações baseadas em TCP, notícias são geralmente transferidas utilizando-se o protocolo NNTP, onde os artigos são solicitados e transferidos individualmente, sem compressão ou outras otimizações. Apesar de adequada para grandes sites com diversas conexões concorrentes de recepção de notícias, esta técnica não é indicada para pequenos sites que recebam notícias sobre uma conexão de baixa velocidade como ISDN. Estes sites usualmente procuram combinar as qualidades do TCP com as vantagens de envio de notícias em grandes lotes, os quais podem ser comprimidos com um trabalho adicional muito pequeno. Uma forma padrão de transferir estes lotes sobre TCP é utilizar UUCP.

No arquivo `sys`, deve-se especificar um sistema que será chamado via TCP da seguinte forma:

```
system          pantanal
address         news.pantanal.edu.br
time            Any
port            conexão-tcp
chat            ogin: aracaju word: dourados
```

O comando `address` fornece o endereço IP da máquina ou o seu nome totalmente qualificado. Uma entrada no arquivo `port` teria a seguinte configuração:

```
port            conexão-tcp
type            tcp
service         540
```

A entrada especifica os padrões que a conexão TCP deverá usar quando uma entrada no arquivo `sys` se referenciar a `conexão-tcp`, definindo que o programa `uucico` deverá tentar conectar-se à porta de rede TCP de número 540 na máquina remota. Esta é a porta padrão para serviços UUCP. Ao invés do número da porta, pode-se fornecer ainda um nome de porta simbólico para o comando `service`. O número de porta correspondente será pesquisado no arquivo `/etc/services`. O nome comum para o serviço UUCP é `uucpd`.

12.3.10 Usando Uma Conexão Direta

Assumindo-se que haja uma conexão direta entre a máquina local denominada `aracaju` e a máquina `arcoverde`. Assim como no caso da ligação via modem, deve-se escrever uma entrada no arquivo `sys`. O comando `port` identifica a porta serial à qual `arcoverde` está conectada.

```
system          arcoverde
time            Any
port            direta1
speed           38400
chat            ogin: vitalino word: ano2000
```

No arquivo `port`, deve haver uma descrição da porta serial para a conexão direta. Uma entrada em `dialer` não será necessária, uma vez que não será efetuada nenhuma discagem.

```
port            direta1
type            direct
speed           19200
```

12.4 O Que Fazer E Não Fazer No UUCP – Ajustando Permissões

12.4.1 Execução de Comandos

As tarefas básicas do UUCP são copiar arquivos de um sistema para outro e requisitar a execução de certos comandos em máquinas remotas. Obviamente, um administrador deseja controlar os direitos definidos para outros sistemas, afinal de contas permitir que eles executem qualquer comando no sistema local pode não ser uma boa idéia.

Os únicos comandos que podem ser executados por sistemas remotos na máquina local, segundo o padrão do Taylor UUCP, são `rmail` e `rnews`, os quais são comumente usados para troca de mensagem e notícias Usenet sobre UUCP. O padrão do caminho de pesquisa usado pelo programa `uuxqt` é definido em tempo de compilação, mas usualmente contém os diretórios `/bin`, `/usr/bin` e `/usr/local/bin`. Para mudar o conjunto de comandos para um sistema em particular, pode-se usar o parâmetro `commands` no arquivo `sys`. Similarmente, para mudar o caminho de pesquisa a ser usado pode-se usar o comando `command-path`. Por exemplo, pode-se desejar que o sistema `parintins` execute o comando `rsmtp` além dos comandos `rmail` e `rnews`:[15]

```
system          parintins
...
commands        rmail rnews rsmtp
```

12.4.2 Transferência de Arquivos

O Taylor UUCP permite uma sintonia fina, ricamente detalhada na funcionalidade de transferência de arquivos. Pode-se, por exemplo, desabilitar a transferência de e para um sistema em particular. Basta configurar o parâmetro `request` para `no` e o sistema remoto não será capaz de receber ou enviar arquivos para o sistema local. Similarmente, pode-se proibir os usuários de transferirem arquivos de ou para um sistema, configurando-se o parâmetro `transfer` para `no`. Por padrão, usuários dos sistemas local ou remoto têm permissão para receber ou enviar arquivos.

Adicionalmente pode-se configurar os diretórios para e de onde os arquivos podem ser copiados. Normalmente, deseja-se restringir os acessos de sistemas remotos a uma única árvore de diretórios e permitir que os usuários possam enviar arquivos a partir de seus diretórios pessoais. Comumente, usuários remotos terão permissão para receber arquivos somente dos diretórios públicos do UUCP, denominado `publicspool`. Este é o local tradicional para tornar arquivos públicos, muito similar ao servidores FTP da Internet. É comumente referenciado através do uso do caractere til.

O Taylor UUCP provê quatro comandos diferentes para configuração dos diretórios de envio e recepção de arquivos. Eles são definidos através dos parâmetros `local-send`, que especifica a lista de diretórios que os usuários podem usar para o envio de arquivos, `local-receive` que define a lista de diretórios usados para a recepção de arquivos, `remote-send` e `remote-receive` que possuem funções similares em sistemas remotos. Considerando-se o seguinte exemplo:

```
system          parintins
...
```

[15] `rsmtp` é usado para entrega de mensagens em lotes SMTP. Isso é descrito no capítulo Correio Eletrônico.

```
local-send      /home ~
local-receive   /home ~/recebidos
remote-send     ~ !~/incoming !~/recebidos
remote-receive  ~/incoming
```

O comando `localsend` permite aos usuários do sistema local o envio de qualquer arquivo contido no diretório `home` ou no diretório público UUCP para o sistema `parintins`. O comando `localreceive` permite que os arquivos recebidos sejam armazenados nos diretórios pessoais sob `home` e no diretório `recebidos` sob o diretório público, normalmente denominado `uucppublic`. A diretiva `remotesend` permite que os usuários de `parintins` requisitem arquivos a partir do diretório `/var/spool/uucppublic`, exceto dos diretórios `incoming` e `receive`. Isso é sinalizado ao programa `uucico` através de um ponto de exclamação (!) antes do nome do diretório. Finalmente a última linha permite que os usuários de `parintins` transfiram quaisquer arquivos para o diretório `recebidos`.

Um dos maiores problemas com a transferência de arquivos através do UUCP reside na necessidade de permissão de gravação para qualquer usuário nos diretórios de destino para que a transferência possa ser efetuada. Isso pode permitir que alguns usuários montem "armadilhas" para os demais, etc.. De qualquer forma, não há forma de contornar este problema, sem desabilitar todas as transferências de arquivos via UUCP.

12.4.3 Reenvio

O UUCP provê um mecanismo que permite que outros sistemas executem as transferências de arquivos ao invés do sistema local. Por exemplo, isso permite que o sistema `blumenau` recupere um arquivo de `chapeco` e o envie para o sistema local. Para que isso ocorra deve ser enviado o seguinte comando:

```
$ uucp -r blumenau!chapeco!~/find-ls.gz ~/chapeco.arqs.gz
```

Esta técnica de passar tarefas através de diversos sistemas é denominada *reenvio*. No exemplo acima, a razão para o uso de reenvio pode ser a disponibilidade de acesso a sessões UUCP na máquina `chapeco` para a máquina `blumenau`, mas não para a máquina local. De qualquer forma, caso a máquina local execute um sistema UUCP, pode-se querer limitar os serviços de reenvio para somente algumas máquinas conhecidas a fim de evitar problemas com a conta telefônica quando alguém quiser baixar a última versão do X11R6.

Por padrão o Taylor UUCP desabilita o reenvio para toda e qualquer máquina. Para habilitar o reenvio para um sistema em particular, pode-se usar o comando `forward`. Este comando especifica uma lista de sites para os quais o sistema pode reenviar ou receber tarefas a serem executadas. Por exemplo, o administrador UUCP de `blumenau` terá que adicionar as seguintes linhas ao arquivo `sys` para permitir que `parintins` requisite arquivos da máquina `chapeco`:

```
####################
# parintins
system          parintins
...
forward         chapeco
####################
# chapeco
system          chapeco
...
forward-to      parintins
```

O parâmetro `forward-to` para `chapeco` é necessário para que qualquer arquivo retornado por ele seja realmente passado para `parintins`. De outra forma o UUCP poderia simplesmente esquecê-los. Esta entrada usa uma variação do comando `forward` que permite que `chapeco` somente envie arquivos para `parintins` através de `blumenau` e não através de outros caminhos.

Para permitir o reenvio para todo e qualquer sistema, basta usar a palavra especial `ANY` (necessariamente em maiúsculas).

12.5 Configurando O Sistema Para o Recebimento de Ligações

Caso deseje-se configurar um site para o recebimento de ligações discadas, deve-se permitir o acesso através da porta serial e customizar-se alguns arquivos do sistema para proverem contas UUCP. Este é o tema desta seção.

12.5.1 Configurando `getty`

Caso deseje-se utilizar uma linha serial como uma porta de recepção de conexões discadas, deve-se habilitar um processo `getty` nesta porta. Eventualmente as implementações do programa `getty` podem não ser totalmente adequadas para estas tarefas, pois pode ser necessário utilizar a porta serial para discar e receber chamadas. Deve-se ainda garantir que o programa `getty` está configurado para compartilhar a linha com outros programas como `uucico`, ou `minicom`. Um programa que pode executar todas estas tarefas é o denominado `uugetty` do pacote `getty_ps`. Muitas distribuições `Linux` trazem o programa. Para certificar-se disso verifique no diretório `/sbin` se ele está disponível. Outro programa que pode ser utilizado é o `mgetty` de autoria de Gert Doering, o qual pode também suportar conexões com fax. Pode-se obter as versões mais recentes em `metalab.unc.edu` nos formatos binário ou através dos fontes.

Explicar as diferenças na forma como `uugetty` e `mgetty` manuseiam os acessos está além do escopo desta pequena seção. Para maiores informações por favor consulte o Como Fazer Serial de Grag Hankins, assim como a documentação que acompanha os programas `getty_ps` e `mgetty`.

12.5.2 Provendo Contas UUCP

O próximo passo será a configuração de contas de usuários para permitir que usuários remotos estabeleçam uma conexão remota com a máquina local. Geralmente, será providenciado um nome de acesso distinto para cada sistema com o qual se estabeleçam conexões. Ao se configurar uma conta para o sistema `parintins`, provavelmente será criada a conta `Uparintins`.

Para sistemas que discam através de uma porta serial, normalmente se deve adicionar uma dessas contas ao arquivo de usuários do sistema denominado `/etc/passwd`. Uma prática recomendada é colocar todos os acessos UUCP em um grupo especial, como por exemplo `uuguest`. O ambiente de trabalho deve ser o programa `uucico`, e o diretório pessoal das contas pode ser definido como sendo o diretório público de tarefas temporárias do UUCP:`/var/spool/uucppublic`.

Caso se tenha as facilidades de senhas sombra instaladas, deve-se utilizar o programa `useradd` ou a interface gráfica de administração do Conectiva Linux denominada Linuxconf:

```
# useradd -d /var/spool/publicspool -G uuguest -s /usr/lib/uucp/uucico
Uparintins
```

Caso as facilidades de senhas sombra não estejam instaladas, pode-se provavelmente editar o arquivo `/etc/passwd` manualmente, adicionando uma linha conforme a apresentada a seguir, onde 5000 e 150 são as identificações numéricas de usuário e grupo definidas para o usuário `Uparintins` e para o grupo `uuguest`, respectivamente.

```
Uparintins:x:5000:150:Usuário UUCP:/var/spool/uucppublic:/usr/lib/uucp/uucico
```

Após a criação da conta, deve-se ativar a senha do usuário através do comando `passwd`.

Para atender a sistemas UUCP que se conectam ao sistema local através de conexões TCP, deve-se configurar o servidor `inetd` para gerenciar conexões à porta `uucp`. Isso pode ser feito através da adição da seguinte linha ao arquivo `/etc/inetd.conf`:[16]

```
uucp   stream  tcp nowait root /usr/sbin/tcpd /usr/lib/uucp/uucico -l
```

A opção `-l` faz com que `uucico` ative o seu próprio processo de autorização. Ele solicitará um nome de acesso e uma senha, da mesma forma que o programa `login`, mas irá basear-se na sua própria base de senhas, ao invés de utilizar o arquivo `/etc/passwd`. Este arquivo é denominado `/usr/lib/uucp/passwd` e contém pares de nomes de acesso e senhas:

```
Uparintins  PororOca
Ublumenau   CapoeiRa
```

Este arquivo deve ter como dono o usuário `uucp` e ter como permissões o valor 600.

Caso essa base de dados pareça uma boa idéia e que deseje-se implementá-la para o controle de acessos em qualquer terminal serial, isso somente será possível com uma série de pequenas configurações. Inicialmente, deve-se ter pelo menos o Taylor UUCP 1.05, porque ele permite que o programa `getty` passe o nome do usuário ao programa `uucico` através da opção `-u`.[17] Após, deve-se configurar o programa `getty` para acionar o programa `uucico` ao invés do tradicional `/bin/login`. No arquivo de configuração `getty_ps`, pode-se configurar a opção `LOGIN`. Veja que todos os acessos interativos ao sistema serão desabilitados. `mgetty` por outro lado

[16] Note que normalmente o programa `tcpd` tem permissões iguais a 700, portanto somente o superusuário, e não o usuário `uucp` deverá acioná-lo.

[17] A opção `-u` está presente também na versão 1.04, mas não está operacional.

tem uma funcionalidade que permite utilizar diferentes comandos baseado no nome do usuário. Por exemplo, pode-se dizer ao `mgetty` para usar `uucico` para todos os usuários cujo nome comece com um U maiúsculo, enquanto que todos os demais devem utilizar o comando padrão `login`.

Para proteger os usuários UUCP de usuários que utilizem suas identificações e bisbilhotem seu correio pessoal, deve-se adicionar o comando `called-login` para cada sistema descrito no arquivo `sys`. Isso é descrito na seção Protegendo-se Contra Invasores abaixo.

12.5.3 Protegendo-se Contra Invasores

Um dos maiores problemas em relação ao UUCP é o fato dele permitir que o sistema de origem possa "mentir" sobre o seu nome, anunciando o seu nome após acessar o sistema de destino e o sistema de destino não tem meios de verificar essa situação. Ou ainda um invasor pode acessar o sistema usando a conta UUCP de outrem e bisbilhotar a sua correspondência. Isso é particularmente problemático se o sistema oferece ainda acesso a UUCP anônimo, onde um usuário e senha de acesso são tornados públicos.

A menos que se possa garantir que todos os sistemas que acessem o sistema local não sofram riscos de ataques desta natureza (alguém pode?), a solução será exigir que cada sistema utilize um nome de acesso particular especificado no parâmetro `called-login` no arquivo `sys`. Um exemplo de entrada neste arquivo se parece com:

```
system          parintins
... opções usuais ...
called-login    Uparintins
```

O parâmetro acima faz com que um sistema que queira acessar o sistema local como `parintins`, tenha que utilizar o usuário `Uparintins`, o qual será verificado pelo programa `uucico`. Caso o usuário definido não seja utilizado, a conexão será desfeita. A utilização da sistemática de adicionar o comando `called-login` para todo sistema que acesse a máquina local deve ser um hábito do bom administrador de sites UUCP. É importante utilizá-lo para *todos* os sistemas, independente se eles ligarão para o sistema local ou não. Para aqueles que nunca ligarão para o sistema local podem ser utilizados usuários com nomes sem sentido como `nuncaacessa`.

12.5.4 Verificação de Seqüência de Chamadas - Seja Paranóico

Outra forma de evitar que impostores acessem o sistema local é o uso de verificações de seqüências de chamadas. Esta sistemática evita que intrusos que de alguma forma obtiveram a senha de acesso ao sistema UUCP local possam efetivamente acessar o sistema.

Ao utilizar a verificação de seqüência de chamadas, ambas as máquinas mantêm o controle do número de conexões estabelecidas até então. Ele é incrementado a cada nova conexão. Após o acesso, o sistema de origem envia o número de seqüência de chamadas e o sistema de origem checará este valor contra o seu controle local. Caso estes valores não coincidam, a tentativa de conexão será rejeitada. O número inicial é escolhido aleatoriamente, fazendo com que invasores tenham um trabalho imenso para descobrirem a seqüência de chamada válida.

Porém, a verificação de seqüência de chamadas pode fazer mais que isso: mesmo que alguma pessoa esperta possa detectar a seqüência de chamada, assim como a senha do usuário, isso pode ser descoberto. Quando um intruso fizer uma conexão e roubar as mensagens disponíveis, isso irá incrementar a seqüência de chamada de transferência de mensagens em um. A próxima vez que o *usuário real* tentar transferir suas mensagens e acessar o sistema, o sistema remoto `uucico` irá recusar a conexão porque a seqüência não confere.

Caso a verificação da seqüência de chamadas seja ativada, deve-se verificar os arquivos de históricos periodicamente buscando mensagens de erro referentes a possíveis ataques. Caso o sistema rejeite um número de seqüência de chamada de um sistema remoto, `uucico` irá colocar uma mensagem em um arquivo de histórico com uma mensagem do tipo "Chamada Fora de Seqüência Rejeitada". Caso uma chamada seja rejeitada pelo sistema alimentador pela falta de sincronismo no número de transferências de mensagens, será então gerada a mensagem "Falha na Negociação (RBADSEQ)".

Para habilitar a verificação da seqüência de chamadas, deve-se adicionar o seguinte comando à entrada do sistema:

```
# habilita a verificação de seqüência de chamadas
sequence        true
```

Além disto, pode-se criar um arquivo contendo um número de seqüência. O Taylor UUCP mantém este valor em um arquivo chamado `.Sequence` no diretório de tarefas temporárias do sistema remoto. Ele *deve* pertencer ao usuário `uucp` e suas permissões devem ser iguais a 600 (ou seja pode ser lido e gravado pelo usuário `uucp`). É recomendável inicializar este arquivo com um valor aleatório, predefinido por ambas as partes. De outra forma, um ataque em potencial poderá tentar adivinhar o número, tentando por exemplo todos os números menores que 60, por exemplo.

A seguir apresentamos a seqüência de criação do arquivo. Na primeira linha, muda-se a localização para o diretório de tarefas temporárias do sistema chamado `parintins`. A seguir ele é inicializado com um número aleatório (no caso 94.316). Após, as permissões são alteradas para leitura e gravação pelo dono do arquivo e finalmente o dono e o grupo do dono do arquivo são alterados para `UUCP`. Maiores informações sobre os comandos `chmod` e `chown` podem ser encontradas nas páginas de manual[18].

```
# cd /var/spool/uucp/parintins
# echo 94316 > .Sequence
# chmod 600 .Sequence
# chown uucp.uucp .Sequence
```

Obviamente o sistema remoto tem também que habilitar a verificação de seqüência de chamadas e iniciar a sua numeração da mesma forma que o sistema local.

12.5.5 UUCP Anônimo

Caso deseje-se prover acesso UUCP anônimo ao sistema local, inicialmente deve-se configurar uma conta especial conforme descrito anteriormente. Uma prática comum é fornecer uma conta de usuário de acesso e uma senha iguais a `uucp`.

Adicionalmente deve-se configurar algumas opções de segurança para sistemas desconhecidos. Por exemplo, deve-se proibi-los de executarem qualquer comando no sistema local. De qualquer forma, não será possível configurar estes parâmetros como uma entrada no arquivo `sys`, uma vez que será exigido o comando `system`, o qual requer um nome de sistema que não está disponível. O Taylor UUCP resolve este dilema através do comando `unknown`. Este pode ser usado no arquivo `config` para especificar qualquer comando que pode normalmente aparecer em uma entrada do sistema:

[18] No Conectiva Linux estarão em português.

```
unknown        remote-receive ~/recebidos
unknown        remote-send ~/pub
unknown        max-remote-debug none
unknown        command-path /usr/lib/uucp/anon-bin
unknown        commands rmail
```

Esta configuração restringe sistemas desconhecidos a receberem arquivos a partir do diretório `pub` e a enviarem arquivos para o diretório `recebidos` sob `/var/spool/uucppublic`. A próxima linha faz com que `uucico` ignore qualquer tentativa do sistema remoto ativar o modo de depuração local. As últimas duas linhas permitem que sistemas remotos desconhecidos executem o comando `rmail`, mas o caminho do comando é especificado, fazendo com que o programa `uucico` procure pelo comando somente em um diretório privativo chamado `anon-bin`. Isso permite que sejam implementadas algumas funcionalidades especiais do programa `rmail`, como por exemplo o reenvio de todas as mensagens para verificação pelo superusuário. Isso permite que os usuários do sistema remoto consigam comunicar-se com o administrador do sistema local, mas evita que eles possam enviar mensagens para outros sites, por exemplo.

Para habilitar o UUCP anônimo, deve-se especificar pelo menos um parâmetro `unknown` no arquivo `config`. Caso contrário `uucico` rejeitará todos os sistemas desconhecidos.

12.6 Protocolos UUCP de Transferência

Para negociar o controle de uma sessão e a transferência de arquivos com um sistema remoto, `uucico` usa um conjunto de mensagens padronizadas. Isso é freqüentemente referenciado como protocolos de alto nível. Durante a fase de inicialização e de desconexão ocorre o simples envio de expressões entre as pontas da conexão. Durante a fase real de transferência de dados, um protocolo adicional de baixo nível é utilizado praticamente de forma transparente para os níveis mais altos. Isso torna possível a checagem de possíveis erros de transmissão ao se utilizar linhas não confiáveis, por exemplo.

12.6.1 Visão Geral do Protocolo

Como o UUCP é usado sobre diferentes tipos de conexão, como linhas seriais ou TCP ou mesmo X.25, protocolos específicos de baixo nível serão necessários. Adicionalmente diversas implementações do UUCP introduziram diferentes protocolos que a grosso modo fazem a mesma coisa.

Protocolos podem ser divididos em duas categorias: protocolos orientados por pacotes e por fluxo de informações. Estes últimos transferem um arquivo como um todo, um fluxo de dados, possivelmente com um número de verificação ao final. É praticamente livre de qualquer informação adicional de protocolo que incremente o tráfego na linha, porém requer uma conexão confiável, uma vez que erros podem provocar a retransmissão de todo o arquivo. Estes protocolos são comumente usados com conexões TCP, mas não são muito adequados para conexões via linha discada. Ainda que os modems mais modernos façam um bom trabalho de correção de erros, eles não são perfeitos, nem conseguem detectar erros entre o modem e o computador.

Por outro lado, protocolos de pacotes dividem o arquivo em diversos pedaços de mesmo tamanho, enviando e recebendo cada pacote separadamente, gerando um número de verificação e um conhecimento de recebimento que é retornado ao remetente. Para tornar isso mais eficiente, protocolos não lineares foram inventados, os quais permitem o envio de um determinado número de pacotes sem a necessidade de recebimento do conhecimento pelo receptor. Isso reduz enormemente o tempo que o programa `uucico` tem que esperar durante a transmissão. O volume de tráfego de controle gerado quando comparado a um protocolo por fluxo torna o protocolo por pacotes ineficiente para uso sobre TCP.

O tamanho dos dados também faz muita diferença. Algumas vezes enviar caracteres de 8 bits sobre uma conexão serial pode ser impossível, por exemplo, para uma conexão efetuada por um servidor de terminais burros. Neste caso, os caracteres com o oitavo bit com conteúdo igual a 1 devem ser tratados antes da transmissão, ou seja eles serão divididos em dois bytes, dobrando o volume de dados transmitidos, apesar de alguma eventual compressão por hardware minorar este problema. Linhas que podem transmitir caracteres de oito bits arbitrariamente são denominadas oito bits livres. Este é o caso de todas as conexões TCP, assim como para a maioria das conexões por modem.

Os seguintes protocolos estão disponíveis com o Taylor UUCP 1.04:

g Este é o protocolo mais comum e é entendido por praticamente todos os programas `uucico`. Efetua verificação de erros e tem sua aplicação indicada para linhas telefônicas ruidosas. *g* requer um conexão oito bits livre. É um protocolo orientado a pacotes com técnicas de verificação não lineares.

i Este é um protocolo bidirecional orientado a pacotes, o qual pode ser usado para o envio e a recepção de arquivos simultaneamente. Requer uma conexão full-duplex e um caminho de dados do tipo oito bits livre. É suportado somente pelo Taylor UUCP.

t Este protocolo foi desenhado para uso sobre conexões TCP, ou outras conexões de rede confiáveis. Usa pacotes de 1.024 bytes e requer uma conexão tipo oito bits livre.

e É basicamente igual a *t*. A principal diferença é que este protocolo é orientado a fluxo de dados.

f Este protocolo foi desenvolvido para uso sobre conexões X.25 confiáveis. É um protocolo orientado por fluxo e trabalha com dados no formato de sete bits. Caracteres de oito bits serão divididos em dois bytes, tornando este protoclo muito ineficiente nestes casos.

G Este é a versão do Unix©System V Release 4 do procolo *g*. É conhecido também por outras versões do UUCP.

a Este protocolo é similar ao ZMODEM. Requer uma conexão de oito bits, mas divide certos caracteres de controle como XON e XOFF.

12.6.2 Ajustando o Protocolo de Transmissão

Todos os protocolos permitem alguma variação do tamanho de pacotes, tempos de espera e outros parâmetros. Normalmente os padrões disponibilizados funcionarão bem sob condições normais, mas podem não ser o ideal para a sua situação. O protocolo *g* por exemplo, usa checagens de transmissão entre 1 a 7 pacotes, tamanhos de pacotes em potências de 2, variando de 64 a 4096 bytes.[19] Caso a linha de telefone seja normalmente tão ruidosa que perca mais de 5 por cento dos pacotes, provavelmente será necessário diminuir o tamanho de pacote e do padrão de verificação. Por outro lado, em uma ligação de boa qualidade, o reconhecimento de recepção pode significar um excesso de tráfego desnecessário a cada 128 bytes, podendo-se alterá-lo para 512 ou mesmo 1024 bytes.

O Taylor UUCP provê um mecanismo para atender as necessidades de ajustes de parâmetros através do parâmetro `protocol-parameter` no arquivo `sys`. Por

[19] Muitos binários incluídos nas distribuições `Linux` têm um padrão de verificação de erros a cada 7 pacotes e tamanhos de pacotes de 128 bytes.

exemplo, para configurar os pacotes do protocolo *g* para um tamanho de 512 bytes durante uma conexão com a máquina `parintins`, deve-se adicionar os seguintes parâmetros:

```
system          parintins
...
protocol-parameter g  packet-size  512
```

Os parâmetros configuráveis e seus nomes variam de protocolo para protocolo. Para uma lista completa por favor verifique a documentação que acompanha o software Taylor UUCP.

12.6.3 Selecionando Protocolos Específicos

Nem toda a implementação de `uucico` pode falar e entender cada protocolo, então durante a fase da negociação ambas as pontas devem concordar sobre um protocolo em comum a ser utilizado. O mestre `uucico` oferece uma lista dos protocolos suportados ao enviar o parâmetro ***`Pprotlist`***, a partir da qual o escravo pode escolher uma opção.

Baseada no tipo de porta (modem, TCP ou direta), `uucico` irá compor uma lista de protocolos padrões. Para modems e conexões diretas esta lista normalmente compreende os protocolos *i*, *a*, *g*, *G* e *j*. Para conexões TCP, a lista normalmente é igual a *t*, *e*, *i*, *a*, *g*, *G*, *j* e *f*. Pode-se substituir esta lista padrão através do comando `protocols`, o qual pode ser especificado na entrada do sistema assim como na entrada da porta. Por exemplo, pode-se editar o arquivo `port` na referência à porta de modem para algo similar a:

```
port            serial1
...
protocols       igG
```

Isso fará com que qualquer conexão de entrada ou de saída através desta porta, use os protocolos *i*, *g*, ou *G*. Caso o sistema remoto não suporte nenhum deles, a conexão irá falhar.

12.7 Problemas & Soluções

Esta seção descreve o que pode sair errado com uma conexão UUCP e apresenta sugestões para sua correção. De qualquer forma, as questões foram compiladas de memória e baseadas na minha experiência. Há muito mais coisas que podem dar errado.

Em qualquer caso, habilite a depuração através da opção `-xall` e verifique o conteúdo do arquivo `Debug` no diretório de tarefas temporárias. Este procedimento pode ajudar a reconhecer rapidamente onde o problema reside. É interessante ainda habilitar o alto falante do modem quando a conexão não ocorre. Em modems compatíveis com o padrão Hayes, isso pode ser obtido adicionando-se "`ATL1M1 OK`" à conversação com o modem no arquivo `dial`.

A primeira verificação deve ser realizada nas permissões de todos os arquivos de configuração. `uucico` deve ser setuid `uucp` e todos os arquivos `/usr/lib/uucp`, `/var/spool/uucp` e `/var/spool/uucppublic` devem ter como dono o usuário `uucp`. Há ainda alguns arquivos escondidos que devem ser checados[20] no diretório de tarefas temporárias e que também devem ter como dono o usuário `uucp`.

`uucico` continua dizendo "Hora Errada Para Discar": Isso provavelmente significa que há uma entrada no arquivo `sys` para o sistema, com o comando `time` não configurado devidamente, ou no momento se está dentro de um horário não permitido para acesso. Caso nenhum parâmetro seja fornecido ao comando, `uucico` assume que o sistema não pode ser acionando nunca.

`uucico` reclama que o site está em uso: Significa que o programa `uucico` detectou um arquivo de reserva de recursos para o sistema remoto em `/var/spool/uucp`. O arquivo pode ter sido gerado por uma conexão anterior que teve um fim anormal. De qualquer forma é como se houvesse outro programa `uucico` tentando discar para o programa remoto e que ficou preso em um programa de conversação, etc. Caso o processo `uucico` não seja bem sucedido na conexão com o sistema remoto, ele pode ser finalizado através do comando `kill` com um sinal de saída e devem ser removidos os arquivos de reservas de recursos criados.

Posso conectar com o site remoto, mas o programa de conversação falha: Verifique o texto recebido do site remoto. Caso ele esteja ilegível, este é provavelmente um problema relacionado com a velocidade. De outra forma, confirme se ela realmente confere com o esperado pelo programa de conversação. Lembre

[20] São arquivos, cujos nomes começam com um ponto. Tais arquivos não são normalmente listados pelo programa `ls`.

que o programa inicia com a espera do recebimento de uma expressão. Caso seja recebido o indicativo de acesso ao sistema, insira algum tempo de espera antes de enviar o nome de usuário, ou mesmo entre as letras. O envio pode estar sendo muito rápido para o modem disponível.

Meu modem não disca: Caso o modem não indique que a linha DTR foi acionada pelo programa `uucico` quando este tenta fazer uma ligação, possivelmente o dispositivo correto não está configurado para o programa `uucico`. Caso seu modem reconheça DTR, verifique através de um programa de terminal se é possível escrever nele. Em caso afirmativo, ative o eco utilizando o comando `\E` no início da conversação com o modem, verifique se a velocidade da linha não está acima da capacidade do modem. Caso se possa visualizar o eco, verifique se as respostas do modem estão habilitadas ou se não estão no formato numérico. Cheque o próprio programa de conversação com o modem. Lembre-se que é necessário definir duas barras para que uma possa ser enviada para o modem.

Meu modem tenta discar, mas não consegue conexão: Insira uma espera no número do telefone. Isso é especialmente útil ao se discar a partir de uma central telefônica interna à uma companhia. Para pessoas na Europa, Brasil e outros países, que usualmente utilizam pulso para discagem, deve ser verificada se a linha telefônica não está configurada para tom. Em alguns países esse tipo de mudança foi feita recentemente.

Meu arquivo de ocorrências diz que há um excesso de perdas de pacotes: Possivelmente há um problema de velocidade. Talvez a conexão entre o computador e o modem seja muito lenta (lembre-se de utilizar a mais alta velocidade de transmissão possível entre estes pontos), ou o hardware é muito lento para prover os serviços de interrupção adequados. De qualquer forma, sem FIFOS (como chips 16450), 9600 bps será o limite. Esteja seguro de que a negociação por hardware esteja habilitada em linhas seriais.

Outra causa possível pode residir na incapacidade da negociação por hardware na porta especificada. O Taylor UUCP 1.04 não habilita a negociação RTS/CTS automaticamente. Deve-se definí-la explicitamente em `rc.serial` usando-se o seguinte comando:

```
$ stty crtscts < /dev/cua3
```

Eu consigo conexão mas a negociação falha: Existem uma série de possíveis causas para este tipo de problema. A saída do arquivo de histórico pode dar indicativos fortes do que está ocorrendo. Verifique os protocolos que o site remoto oferece (ele deve enviar a cadeia P*protlist* durante a negociação). Talvez eles não tenham nada em comum (foi selecionado algum protocolo nos arquivos `sys` ou `port`?).

Caso o sistema remoto envie a expressão `RLCK`, há um arquivo de reserva de recursos para o sistema informado no sistema remoto. Caso ele não seja originado de uma conexão ao mesmo sistema através de uma linha diferente, solicite a sua remoção.

Caso o sistema remoto envie a expressão `RBADSEQ`, o outro site tem a verificação de seqüência de acesso habilitada, mas os números não conferem. Caso ele envie a expressão `RLOGIN`, o usuário informado não tem permissão de acesso ao sistema remoto.

12.8 Arquivos de Históricos

Ao se compilar o conjunto de ferramentas UUCP para usar o acesso ao estilo Taylor UUCP, existem três arquivos globais de históricos, todos residindo no diretório de tarefas temporárias. O principal arquivo é denominado `Log` e contém informações sobre conexões estabelecidas e arquivos transferidos. Um típico extrato deste arquivo tem a aparência a seguir (após pequenos ajustes para que coubesse nesta página):

```
uucico parintins - (1999-05-28 17:15:01.66 539) Calling system parintins (port cua3)
uucico parintins - (1999-05-28 17:15:39.25 539) Login successful
uucico parintins - (1999-05-28 17:15:39.90 539) Handshake successful
               (protocol 'g' packet size 1024 window 7)
uucico parintins postmaster (1999-05-28 17:15:43.65 539) Receiving D.parintinsB04aj
uucico parintins postmaster (1999-05-28 17:15:46.51 539) Receiving X.parintinsX04ai
uucico parintins postmaster (1999-05-28 17:15:48.91 539) Receiving D.parintinsB04at
uucico parintins postmaster (1999-05-28 17:15:51.52 539) Receiving X.parintinsX04as
uucico parintins postmaster (1999-05-28 17:15:54.01 539) Receiving D.parintinsB04c2
uucico parintins postmaster (1999-05-28 17:15:57.17 539) Receiving X.parintinsX04c1
uucico parintins - (1999-05-28 17:15:59.05 539) Protocol 'g' packets: sent 15,
                resent 0, received 32
uucico parintins - (1999-05-28 17:16:02.50 539) Call complete (26 seconds)
uuxqt parintins postmaster (1999-05-28 17:16:11.41 546) Executing X.parintinsX04ai
               (rmail okir)
uuxqt parintins postmaster (1999-05-28 17:16:13.30 546) Executing X.parintinsX04as
               (rmail okir)
uuxqt parintins postmaster (1999-05-28 17:16:13.51 546) Executing X.parintinsX04c1
               (rmail okir)
```

O próximo arquivo importante de histórico é denominado `Stats`, o qual lista as estatísticas de transferências de arquivos. Um seção do arquivo `Stats` referente às transferências acima tem o seguinte aspecto:

```
postmaster parintins (1999-05-28 17:15:44.78)
                received 1714 bytes in 1.802 seconds (951 bytes/sec)
postmaster parintins (1999-05-28 17:15:46.66)
                received 57 bytes in 0.634 seconds (89 bytes/sec)
postmaster parintins (1999-05-28 17:15:49.91)
                received 1898 bytes in 1.599 seconds (1186 bytes/sec)
postmaster parintins (1999-05-28 17:15:51.67)
                received 65 bytes in 0.555 seconds (117 bytes/sec)
postmaster parintins (1999-05-28 17:15:55.71)
                received 3217 bytes in 2.254 seconds (1427 bytes/sec)
postmaster parintins (1999-05-28 17:15:57.31)
                received 65 bytes in 0.590 seconds (110 bytes/sec)
```

Novamente as linhas foram divididas para caberem nesta página.

O terceiro arquivo é denominado `Debug`. Este é o local onde os arquivos de depuração são escritos. Caso se esteja utilizando a depuração, deve-se assegurar que o arquivo tem permissões de valor igual a 600. Dependendo do tipo de depuração que se configure, ele poderá conter identificações de usuários e senhas usadas na conexão a sistemas remotos.

Alguns binários UUCP incluídos nas distribuições `Linux` devem ser compilados para utilizarem o estilo de arquivos de histórico no formato HDB. Este utiliza uma série de arquivos de históricos sob o diretório `/var/spool/uucp` com a extensão `/.Log`. Este diretório contém três ou mais subdiretórios, chamados `uucico`, `uuxqt` e `uux`, que contêm a saída do acesso gerado por cada um dos comandos, ordenados em diferentes arquivos para cada site; ou seja, a saída do programa `uucico` ao acessar o site `parintins` será registrada no arquivo `.Log/uucico/parintins`, enquanto que um programa `uuxqt` subseqüente irá gerar o arquivo `.Log/uuxqt/parintins`. Os registros gerados nestes arquivos têm o mesmo formato dos arquivos de histórico de Taylor mostrados anteriormente.

Ao habilitar a saída de depuração no estilo HDB UUCP através da compilação do UUCP, será utilizado o diretório `.Admin` sob `/var/spool/uucp`. Durante a execução de chamadas de saída, as informações de depuração serão registradas no arquivo `.Admin/audit.local`, enquanto que as saídas do comando `uucico` referentes às chamadas de entrada serão registradas no arquivo `.Admin/audit`.

Capítulo 13

Correio Eletrônico

Um dos mais destacados usos das redes desde que as primeiras instalações foram feitas, tem sido o correio eletrônico. Ele começou como um serviço simples que copiava um arquivo de uma máquina para outra e o anexava ao arquivo *mailbox*[1] do destinatário. Este é basicamente o que o correio eletrônico ainda faz, apesar do crescimento da rede e das complexas rotinas de roteamento requeridas, assim como do incremento do número de mensagens que provocaram a criação de esquemas mais elaborados.

Vários padrões de trocas de mensagens foram definidos. Sites na Internet aderiram à recomendação descrita na RFC 822, argumentando que algumas outras recomendações eram dependentes da forma como a máquina transferia caracteres especiais e outros aspectos relacionado à arquitetura. Muito esforço foi feito recentemente para criar "mensagens multimídia", as quais lidam com imagens e sons em mensagens de correio. Outro padrão, denominado X.400, foi definido pelo CCITT.

Um número razoável de programas de transporte de mensagens foi implementado para os sistemas `Unix`. Um dos mais conhecidos é o programa `sendmail` da Universidade de Berkeley, o qual é usado em diversas plataformas. O autor original é Eric Allman, o qual está trabalhando ativamente no grupo de desenvolvimento do programa. Há dois portes disponíveis para `Linux` do `sendmail-5.56c`, um dos quais será descrito no capítulo 15. A versão atual do programa `sendmail` disponível é a 8.9.1a.

[1] caixa de correio

O agente de mensagens mais usado com `Linux` é o programa `smail-3.1.28`, escrito e registrado por Curt Landon Noll e Ronald S. Karr, o qual está incluído na maioria das distribuições `Linux`. A seguir faremos referência simplesmente ao programa `smail`, apesar de existirem versões inteiramente diferentes e que não serão descritas aqui.

Comparado com o programa `sendmail`, `smail` é bastante jovem. Ao manusear mensagens para um pequeno site, sem a necessidade de roteamentos complexos, suas funcionalidades são muito próximas. Para sites maiores porém, `sendmail` sempre será mais eficiente, devido ao seu sistema de configuração mais flexível.

Ambos os programas, `smail` e `sendmail`, suportam um conjunto de arquivos de configuração que devem ser personalizados. Além das informações necessárias para que o subsistema de mensagens funcione (como por exemplo o nome da máquina local), há diversos parâmetros que devem ser adequados ao site local. O arquivo principal de configuração do programa `sendmail` pode parecer bastante complexo à primeira vista. Parece como se um gato tivesse tirado uma soneca em seu teclado com a tecla Shift pressionada. Os arquivos de configuração do programa `smail` são mais estruturados e simples de serem compreendidos do que os do programa `sendmail`, mas não fornece ao usuário muitas opções sobre o comportamento do correio eletrônico. Porém para pequenos sites UUCP ou Internet, o trabalho de configuração necessário será basicamente o mesmo.

Neste capítulo, abordaremos o correio eletrônico e os aspectos ligados à sua administração. Os capítulos 14 e 15 fornecerão maiores informações sobre a configuração inicial dos programas. As informações aqui disponibilizadas devem ser suficientes para poder-se operacionalizar pequenos sites, porém há muitas outras opções e pode-se despender horas em frente do computador configurando-se funcionalidades exóticas.

Ao final desde capítulo cobriremos rapidamente o programa `elm`, um agente de usuário muito comum em muitos sistemas `Unix`, inclusive no `Linux`.

Para maiores informações sobre o tema correio eletrônico em `Linux`, por favor consulte o Como Fazer - Correio Eletrônico de Vince Skahan, o qual é postado no `comp.os.linux.announce` regularmente. Os fontes da distribuição dos programas `elm`, `smail` e `sendmail` contêm também uma extensiva documentação, que poderá responder à maioria das dúvidas sobre a sua configuração. Caso se esteja buscando informações sobre correio eletrônico em geral, há um número razoável de RFCs que tratam do tema. Elas estão listadas na bibliografia ao final deste livro.

13.1 O Que É Uma Mensagem de Correio Eletrônico?

Uma mensagem de correio eletrônico consiste de um corpo de mensagem, o qual é o texto enviado pelo remetente e um conjunto de dados especiais que indicam conteúdo, meio de transporte, etc., similar ao que se vê em um envelope de correio comum.

Os dados administrativos dividem-se em duas categorias: a primeira trata do meio de transporte, como o endereço do remetente e o conteúdo é conhecido como *envelope*. Pode ser transformada pelo programa de transporte à medida que a mensagem é transferida de um site para outro. A segunda categoria contém os dados necessários para o manuseio da mensagem, que não sejam proprietários de nenhum mecanismo particular de transporte, como a linha do assunto da mensagem, uma lista dos destinatários e a data em que a mensagem foi enviada. Em muitas redes, tem se tornado um padrão anexar no início da mensagem estes dados, formando um conjunto de dados chamados *cabeçalho da mensagem*. Ele é separado do corpo da mensagem por uma linha em branco.[2]

Muitos softwares de transporte de mensagens no mundo `Unix` usam o formato de cabeçalho delineado na RFC 822. O seu propósito original foi especificar um padrão para uso na ARPANET, mas uma vez que ele foi desenvolvido para ser independente de qualquer ambiente, foi facilmente adaptado para outras redes, inclusive aquelas baseadas em UUCP.

A RFC 822 de qualquer forma é somente um grande denominador comum. Os padrões mais recentes foram desenvolvidos para cobrir o crescimento de necessidades, como por exemplo, criptografia de dados, suporte a conjuntos de caracteres internacionais e extensões multimídia para mensagens (MIME).

Em todos estes padrões, o cabeçalho consiste de diversas linhas, separadas por caracteres de nova linha. Uma linha consiste de um nome de campo, começando na coluna um, o campo em si separado por dois pontos e um espaço. O formato e a semântica de cada campo varia dependendo de seu nome. Um campo de cabeçalho pode ter informações que utilizem mais de uma linha. A linha seguinte deverá começar com o caractere de tabulação. Os campos podem aparecer em qualquer ordem.

[2] É comum ainda estar anexo à mensagem um arquivo de *assinatura* ou `.sig`, normalmente contendo informações sobre o autor, junto com alguma mensagem adicional. Ele é separado do corpo da mensagem por uma linha contendo "`-`" seguidos de um espaço.

Um cabeçalho de mensagem típico tem o seguinte aspecto:

```
From conectiva.com.br!crhl.com.br!andyo Wed Apr 13 00:17:03 1999
Return-Path: < conectiva.com.br!crhl.com.br!andyo>
Received: from conectiva.com.br by mail.conectiva.com.br with uucp
        (Smail3.1.28.1 #6) id m0pqqlT-00023aB; Wed, 13 Apr 99 00:17 MET DST
Received: from crhl.com.br (cascao.crhl.com.br) by conectiva.com.br with smtp
        (Smail3.1.28.1 #28.6) id <m0pqoQr-0008qhC>; Tue, 12 Apr 99 21:47 MEST
Received: by crhl.com.br (8.6.8/8.6.4) id RAA26438; Tue, 12 Apr 99 15:56 -040
Date: Tue, 12 Apr 1999 15:56:49 -0400
Message-Id: <199904121956.PAA07787@ruby>
From: andy@crhl.com.br (Andy Oram)
To: linux@conectiva.com.br
Subject: Re: Seção Correio Eletrônico
```

Normalmente todos os campos de cabeçalho necessários são gerados pela interface de correio utilizada pelo usuário, como por exemplo os programas `elm`, `pine`, `mush` ou `mailx`. Há porém alguns campos opcionais que podem ser adicionados pelo usuário. O programa `elm`, por exemplo, permite que possa ser editada parte do cabeçalho da mensagem. Outros campos são adicionados pelo software de transporte de mensagens. Uma lista dos campos mais comuns nos cabeçalhos de mensagens é apresentado a seguir:

`From:` Contém o endereço de correio eletrônico do remetente e possivelmente o seu "nome completo". Os mais diversos formatos podem ser utilizados.

`To:` Contém o endereço do destinatário da mensagem.

`Subject:` É o assunto da mensagem, caracterizando o seu conteúdo em poucas palavras.

`Date:` A data em que a mensagem foi enviada.

`Reply-To:` Especifica o endereço em que o remetente deseja receber a resposta da mensagem. Isso pode ser útil caso se utilizem diversos endereços distintos de correio, mas se deseje receber as respostas somente naquele usado mais freqüentemente. Este campo é opcional.

`Organization:` A organização à qual a máquina pertence e na qual a mensagem foi originada. Caso a máquina seja de um particular, esse campo pode ser deixado em branco ou pode se inserir a expressão "privado" ou outro texto qualquer. Este campo é opcional.

`Message-ID:` Uma expressão gerada pelo programa de transporte no sistema de origem. É único para cada mensagem.

`Received:` Toda a máquina que receber e processar esta mensagem (incluindo as máquinas do remetente e do destinatário) inserem este campo no cabeçalho, fornecendo o nome do site, a identificação da mensagem, a hora e a data que a mensagem foi recebida, o site de origem e o software de transporte utilizado. É possível assim conhecer o caminho que a mensagem utilizou e encontrar o responsável caso algum problema tenha ocorrido.

`X-`*anything*: Nenhum programa relacionado a mensagens eletrônicas pode rejeitar qualquer mensagem que tenha um cabeçalho que comece com `X-`. Ele é usado para implementar funcionalidades adicionais que não estão definidas por uma RFC ou nunca estarão. Isso é usado, por exemplo pela lista de discussão dos Ativistas Linux, onde o canal é selecionado através do campo de cabeçalho `X-Mn-Key:`.

A única exceção para esta estrutura é a primeira linha. Ela necessariamente começa com a palavra chave `From` a qual é seguida por um espaço em branco ao invés de dois pontos. Para distingui-lo de um campo comum `From:`, ela é freqüentemente referenciada como `From_`. Contém a rota da mensagem no formato de caminho UUCP (explicado a seguir), a hora e a data em que ela foi recebida pela última máquina que a processou e uma parte opcional que especifica a máquina de origem da última transmissão efetuada. Uma vez que este campo é gerado novamente por todo o sistema que processa a mensagem, ele é algumas vezes resumido sob os dados de envelope.

O campo `From_` é na verdade um artifício para se manter a compatibilidade com alguns agentes de mensagens antigos, mas que não é muito usado atualmente, exceto pelas interfaces de correio eletrônico para usuários. Para evitar possíveis problemas com linhas nas mensagens que também comecem com a expressão "From", tornou-se um procedimento padrão fazer com que estas ocorrências comecem com o caractere ">".

13.2 Como Uma Mensagem É Enviada?

Geralmente, uma mensagem será composta através de uma interface de correio como o programa `mail` ou `mailx`, ou outras mais sofisticadas como o programa `elm`, `mush` ou `pine`. Estes programas são chamados *agentes de mensagens de usuários*[3], ou MUAs em formato resumido. Ao se enviar uma mensagem, o programa de interface irá em muitos casos acionar outro programa para a entrega. Este é denominado *agente de transporte de mensagens*[4], ou MTA. Em alguns sistemas, há diferentes agentes de transporte de mensagens para entrega local ou remota, em outros há somente um. O comando para entrega remota de mensagens normalmente utilizado é o `rmail`, para entregas locais normalmente é utilizado o comando `lmail` (caso a distinção esteja implementada).

Entregas locais de mensagens, envolvem mais aspecto que somente anexar a mensagem que chega à caixa postal do destinatário. Normalmente o MTA local irá executar tarefas relacionadas com o uso de nomes alternativos ou apelidos (configurando os endereços dos destinatários locais que apontam para os nomes alternativos) e o reenvio de mensagens (redirecionando uma mensagem de um usuário para outra destinação). Há ainda mensagens que não podem ser entregues e que serão *devolvidas*, ou seja retornadas para o remetente com alguma mensagem de erro (da mesma forma que uma carta comum é devolvida).

Para entregas remotas, o programa de transporte usado depende da natureza da conexão. Caso as mensagens devam ser entregues através de uma rede usando TCP/IP, o protocolo mais comumente utilizado é o SMTP, que significa Protocolo Simples de Transferência de Mensagens[5] e é definido pelas RFC 788 E RFC 821. SMTP normalmente se conecta diretamente com a máquina do destinatário, negociando a transferência da mensagem com o servidor SMTP da máquina remota.

Em redes UUCP, mensagens não serão normalmente entregues diretamente, mas sim reenviadas para a máquina de destino através de um conjunto de sistemas intermediários. Para enviar uma mensagem através de uma conexão UUCP, o MTA remetente normalmente executará o programa `rmail` no sistema que fará o reenvio, usando o programa `uux` e escrevendo a mensagem na entrada padrão do sistema remoto.

Uma vez que isto é feito para cada mensagem separadamente, pode-se produzir uma considerável demanda nos principais pontos de reenvio de mensagens, assim como seria possível congestionar as filas de tarefas temporárias do UUCP com milhões de mensagens utilizando uma quantidade de disco descomunal.[6] Alguns

---

[3]mail user agents

[4]mail transport agent

[5]Simple Mail Transfer Protocol

[6]Isso se deve ao fato do espaço em disco ser alocado em blocos de 1024 bytes. Em assim sendo, mesmo mensagens de 400 bytes utilizarão um espaço de 1 Kb.

MTAs entretanto permitem que diversas mensagens para um sistema remoto sejam colecionadas em um único arquivo, chamado arquivo de lote. Este arquivo contém os comandos SMTP que seriam normalmente utilizados pelo sistema local ao enviar diretamente a mensagem durante uma conexão. Isso é denominado BSMTP, ou SMTP *em lotes*[7]. O lote é então enviado para o programa `rsmtp` ou `bsmtp` no sistema remoto, o qual irá processar a entrada como se uma transferência SMTP normal tivesse ocorrido.

13.3 Endereço de Correio Eletrônico

Para o uso do correio eletrônico, um endereço deve ser definido, com no mínimo o nome de uma máquina que administre as mensagens do destinatário e a identificação pela qual o usuário é reconhecido neste sistema. Esse pode ser o nome de acesso do destinatário, mas pode ter outros formatos. Outro sistema de endereçamento, como por exemplo o C.400, usa uma configuração mais genérica de atributos, que são utilizados para pesquisar um destinatário em um servidor de diretórios X.500.

A forma como o nome da máquina é interpretado, ou seja para qual site a mensagem deve ser enviada e como combinar este nome com a identificação do destinatário depende grandemente do tipo de rede na qual se esteja conectado.

Sites Internet aderem ao padrão definido na RFC 822, a qual exige uma notação do tipo `usuário@máquina.domínio`, onde `máquina.domínio` é o nome da máquina de destino totalmente qualificado. O caractere que divide os campos é denominado arroba (ou "na", com o significado de usuário X na máquina Z do domínio Y). Uma vez que esta notação não envolve uma rota para uma máquina de destino, mas fornece somente um nome de máquina (único), ele é chamado de endereço *absoluto*.

No ambiente UUCP, a forma predominante é `caminho!máquina!usuário`, onde `caminho` é descrito como uma seqüência de máquinas pelas quais a mensagem tem que passar antes de alcançar a `máquina` de destino. Esta construção é chamada de *caminho bang*, porque um ponto de exclamação é comumente chamado de "bang" (em inglês) e é utilizado para separar as máquinas utilizadas. Hoje muitas máquinas UUCP adotaram a RFC 822 e entenderão este tipo de endereço.

Bem, estes dois formatos de endereços não se misturam muito bem. Assumindo

[7] batched

um endereço igual a `máquinaA!usuário@máquinaB`, não fica claro se o sinal '`@`' tem precedência sobre o caminho ou vice-versa: a mensagem tem que ser enviada para a `máquina B`, a qual a reenviará para `máquinaA!usuário`, ou deve ser enviada para `máquinaA`, a qual a reenviará para o `usuário@máquinaB`?

Endereços que têm diferentes operadores são chamados de *endereços híbridos*. O mais notório é o exemplo acima. Ele é normalmente resolvido dando-se precedência ao caractere '`@`' sobre o caminho. No exemplo acima, isso significa que a mensagem será inicialmente enviada para a `máquinaB`.

De qualquer forma há uma maneira de especificar rotas nos formatos descritos na RFC 822: <`@máquinaA,@máquinaB:usuário@máquinaC`> descreve o endereço de `usuário` na `máquinaC`, onde `máquinaC` pode ser alcançada através das máquina `máquinaA` e `máquinaB` (nesta ordem). Este tipo de endereço é freqüentemente chamado de *endereço route-addr*.

Há ainda o operador de endereços '`%`': `usuário%máquinaB@máquinaA`, o qual faz com que a mensagem seja enviada inicialmente para a `máquinaA`, a qual expande o sinal `%` mais à direita (neste caso o único) para um sinal '`@`'. O endereço será agora `usuário@máquinaB` e o MTA irá reenviar esta mensagem tranqüilamente para a máquina `máquinaB` a qual a entregará a `usuário`. Este tipo de mensagem é algumas vezes referenciada como "Coisas Antigas da ARPANET" e seu uso não é aconselhado. De qualquer forma, muitos agentes de transporte de mensagens geram este tipo de endereço.

Outras redes têm ainda diferentes tipos de endereçamento. Redes baseadas em DECnet, por exemplo, utilizam dois sinais de dois pontos como separador de endereços, formando um endereço de tipo *`máquina::usuário`*. [8] Mais recentemente o padrão X.400 descreve um usuário de forma totalmente diferentes através de um conjunto de pares de atributos, como por exemplo país e organização.

Na FidoNet, cada usuário é identificado por um código como `2:320/204.9`, consistindo de quatro números significando a zona (2 para a Europa), rede (320 significando Paris e Banlieue), o nó (nó local que centraliza as mensagens) e o ponto (o usuário da máquina). Endereços Fidonet podem ser mapeados para a RFC 822; o endereço acima poderia ser descrito da seguinte forma:

```
Thomas.Aquino@p9.f204.n320.z2.fidonet.org
```

[8] Ao tentar alcançar um endereço em uma rede DECnet a partir de um ambiente compatível com a RFC 822, deve-se usar o formato email"*`máquina::usuário`*"@*`retransmissor`*, onde *`retransmissor`* é o nome de retransmissor Internet-DECnet conhecido.

Agora não diga que eu afirmei que é endereço simples de ser lembrado. :-) Há algumas implicações em se usar estes diferentes tipos de endereçamento, as quais serão descritas nas seções seguintes. Em um ambiente compatível com a RFC 822 raramente será utilizado um formato diferente do endereço absoluto como os endereços `usuário@máquina.domínio`.

13.4 Como Funciona o Roteamento de Mensagens?

O processo de direcionar uma mensagem para a máquina do destinatário é denominado *roteamento*. Além de encontrar um caminho do site de origem até a máquina de destino, ele envolve a verificação de erros, assim como otimizações de velocidade e custos.

Há uma grande diferença na forma como um site UUCP administra o roteamento e a forma como isso é feito na Internet. Na Internet, a tarefa principal é direcionar os dados para a máquina do destinatário (uma vez que se tenha o seu endereço IP), a qual é executada pela camada de rede IP, enquanto que no UUCP o roteamento é fornecido pelo usuário ou gerado pelo agente de transferência de mensagens.

13.4.1 Roteamento de Mensagens Na Internet

Na Internet, as tarefas de roteamento estão baseadas inteiramente na máquina de destino. O padrão é entregar a mensagem diretamente à máquina de destino através da pesquisa do seu endereço IP e deixar a rotina de roteamento dos dados a cargo da camada de transporte IP.

Muitos sites desejarão direcionar todas as mensagens destinadas a eles para um servidor de alta disponibilidade, capaz de administrar um tráfego intenso e posteriormente distribuir as mensagens localmente. Para divulgar este serviço, o site publica um registro de recursos denominado MX para o domínio local na base de dados DNS. MX significa *Negociador de Mensagens*[9] e basicamente indica que o servidor disponibiliza serviços de reenvio para todas as máquinas do domínio. Registros MX podem ser usados para manusear o tráfego entre máquinas que não estejam diretamente conectadas a Internet, como por exemplo redes UUCP ou redes corporativas com máquinas que contêm informações confidenciais.

Registros MX indicam ainda uma *preferência* associada a eles, indicada por um

[9] Mail Exchanger

número positivo inteiro. Caso existam diversos registros MX para uma máquina, o agente de transporte de mensagens irá tentar transferir a mensagem através do servidor de menor valor e caso essa tentativa não seja bem sucedida tentará através da máquina de maior valor seguinte. Caso a máquina local seja ela própria o servidor de correio para o endereço de destino indicado, ele não poderá reenviar mensagens para qualquer máquina MX de valor maior, a fim de se evitar a criação de uma rotina circular sem saída.

Supondo-se que uma organização chamada Cnclinux Ltda., deseje que todos as suas mensagens sejam administradas por uma máquina chamada `correio`. A base de dados DNS conterá então o seguinte registro:

```
teresina.cnclinux.com.br    IN   MX     5  correio.cnclinux.com.br
```

Isso anuncia que `correio.cnclinux.com.br` é o servidor de correio para a máquina `teresina.cnclinux.com.br` com um valor de preferência igual a 5. Uma máquina que queira entregar uma mensagem para `di@cnclinux.com.br` irá verificar a base de dados DNS para `cnclinux.com.br` e encontrará um registro MX apontando para `correio`. Caso não haja um outro registro MX com um indicador de preferência menor que 5, a mensagem será entregue então para `correio`, a qual a despachará para `teresina`.

Esta descrição é na verdade um resumo simplificado de como os registros MX funcionam. Para maiores informações sobre roteamento de mensagens, por favor verifique a RFC 974.

13.4.2 Roteamento de Mensagens no Mundo UUCP

O roteamento de mensagens em redes UUCP é muito mais complexo que na Internet, uma vez que o software de transporte não executa qualquer rotina desta natureza. No início, todas as mensagens deviam ser endereçadas usando-se caminhos bang. Estes especificam uma lista de máquinas separadas por bangs (pontos de exclamação), seguidos pelo nome do usuário. Para endereçar uma mensagem para Paulo Renato na máquina chamada `cianorte`, deve-se usar, por exemplo, o caminho `pgrossa!guarapuava!cianorte!pr`. Esta mensagem será então enviada para `pgrossa`, a partir daí para `guarapuava` e finalmente para `cianorte`.

O ponto crítico desta técnica reside na necessidade de se conhecer toda a topologia da rede, caminhos mais eficientes, etc. Muito pior que isso, mudanças provocadas na topologia da rede, tais como conexões que são removidas ou máquinas que sejam desativadas, podem causar falhas na entrega das mensagens simplesmente porque o remetente não sabia das mudanças. E finalmente, no caso de mudanças para uma nova base, todos as rotas terão que ser reaprendidas.

Uma das causas que criou a necessidade do uso de roteamento na origem foi a presença de nomes ambíguos: por exemplo, assumindo-se que há duas máquinas chamadas `cianorte`, uma no Brasil e outra na Argentina. Para qual site o endereço `cianorte!pr` aponta? Isso pode se tornar claro à medida que se defina qual o caminho que deve ser usado para se chegar a `cianorte`.

O primeiro passo para eliminar eventuais ambigüidades foi a criação do *Projeto de Mapeamento UUCP*[10]. Ele está localizado na Universidade Rutgers e registra todos os nomes oficiais de máquinas UUCP e a localização geográfica de seus vizinhos mais próximos, assegurando que o nome da máquina não será utilizado novamente. As informações mantidas pelo Projeto de Mapeamento são publicadas regularmente com o nome de *Mapas Usenet*, através da Usenet.[11] Uma entrada típica no Mapa (após a remoção dos comentários) tem o seguinte aspecto:

```
cianorte
        guarapuava(DAILY/2),
        londrina(WEEKLY)
```

Esta entrada indica que a máquina `cianorte` tem uma ligação com `guarapuava`, duas vezes ao dia e com a máquina `londrina` semanalmente. Retornaremos ao formato do arquivo de Mapa em maiores detalhes adiante.

Usando as informações de conectividade disponibilizadas pelos mapas, pode-se automaticamente gerar caminhos completos a partir de uma máquina local para qualquer site de destino. Esta informação é normalmente armazenada no arquivo `paths` também chamado de arquivo de *base de dados de caminhos alternativos*. Assumindo que os Mapas indiquem que se pode acessar `marilia` através de `bauru`, então um caminho alternativo para a máquina `marilia` gerado pelo Map acima teria o seguinte formato:

```
marilia         bauru!limeira!marilia!%s
```

[10] The UUCP Mapping Project

[11] Mapas de sites registrados no Projeto de Mapeamento UUCP são distribuídos através do grupo de notícias `comp.mail.maps`. Outras organizações podem publicar mapas separadamente para suas redes.

Caso o endereço de destino seja `alice@marilia.uucp`, o MTA irá escolher a rota acima e enviará a mensagem para `bauru` com um envelope de endereçamento igual a `limeira!marilia!alice`.

Construir um arquivo `path` com todos os mapas da Usenet não é uma idéia muito interessante. As informações podem estar distorcidas e ocasionalmente desatualizadas. Assim sendo, somente um pequeno número de máquinas principais usam o mapa mundial completo do UUCP para construir seus arquivos `path`. Muitos sites mantêm somente informações de roteamento para os sites próximos e enviam as mensagens destinadas a sites desconhecidos para uma máquina que tenha um mapeamento mais completo. Este sistema é chamado *roteamento de máquinas otimizado*. Máquinas que têm somente uma ligação de correio UUCP (também chamadas *sites folha*) não executam nenhuma ação de roteamento por si só, baseando-se inteiramente no roteamento otimizado.

13.4.3 Misturando-se UUCP e RFC 822

A melhor cura contra todos os problemas de roteamento em redes UUCP encontrada até aqui foi a adoção do sistema de nomes de domínios para redes UUCP. Obviamente que não se pode fazer uma pesquisa de um servidor de nomes sobre o UUCP. Na verdade muitos sites UUCP formaram pequenos domínios que coordenam o roteamento internamente. Nos Mapas, estes domínios anunciam uma ou duas máquinas como o caminho padrão para o envio de mensagens, não necessitando assim de uma entrada no Mapa para cada máquina do domínio. As máquinas configuradas como caminhos padrão gerenciam todas as mensagens que chegam e saem do domínio. O sistema de roteamento interno do domínio é completamente invisível para o mundo externo.

Isso funciona muito bem com o sistema otimizado por máquina descrito acima. Informações de roteamento global são mantidas somente nas máquinas que servem como caminho padrão, enquanto que as máquinas menores dentro do domínio terão somente um pequeno arquivo `paths` que lista as rotas dentro do domínio e o servidor que atua como caminho padrão de mensagens. Mesmo estes não terão as informações de todas as máquinas UUCP dos Mapas. Além das informações de roteamento das máquinas do domínio ao qual elas servem, elas necessitam somente ter rotas para os domínios principais. Por exemplo, um caminho alternativo para a entrada abaixo, irá rotear todas as mensagens para os sites dos domínios `edu.br` para a máquina `macunaima`:

```
.edu.br        parintins!macunaima!%s
```

Qualquer mensagem endereçada para `drumond@ufsj.edu.br` será enviada para a máquina `parintins` com um endereço de envelope igual a `macunaima!ufsj!drumond`.

A organização hierárquica de um nome de domínio permite que servidores de correio misture rotas mais específicas com outras menos detalhadas. Por exemplo, um sistema na França pode ter rotas específicas para os subdomínios `fr`, mas roteará qualquer mensagem para o domínio `br` para uma máquina localizada no Brasil. Desta forma, o roteamento baseado em domínios (como esta técnica é denominada) reduz enormemente o tamanho das bases de dados de roteamento assim como as tarefas administrativas adicionais necessárias à sua manutenção.

O principal benefício do uso de nomes de domínio em um ambiente UUCP, reside na conformidade com o padrão RFC 822, que permite a troca simples de mensagens entre redes UUCP e a Internet. Muitos domínios UUCP nestes dias têm uma conexão com a Internet e um caminho padrão que age na tarefa de roteamento otimizado por máquina. O envio de mensagens através da Internet é mais rápido e as informações de roteamento são muito mais confiáveis uma vez que os servidores Internet utilizam DNS ao invés de Mapas Usenet.

Para se poder alcançar a partir da Internet um domínio baseado em UUCP, normalmente o seu caminho padrão na Internet anunciará um registro MX para ele. Por exemplo, assumindo que a máquina `macunaima` pertença à rede do domínio `ufsj.edu.br`. `dourados.pantanal.edu.br` age como seu caminho padrão para a Internet. `macunaima` poderá então utilizar `dourados` como a sua máquina de otimização de roteamento, enviando assim todas as mensagens para domínios externos através de `dourados`. Por outro lado, `dourados` poderá anunciar um registro MX para o domínio `ufsj.edu.br` e entregar todas as mensagens que chegue para sites deste domínio para a máquina `macunaima`.

O único problema reside no fato dos programas de transporte UUCP não lidarem com nomes de domínios totalmente qualificados. Muitos conjuntos de ferramentas UUCP foram desenhadas para utilizarem nomes de sites com até oito caracteres, alguns sete, ou até menos, e o uso de caracteres alfanuméricos como pontos é completamente fora de questão para muitos deles.

De qualquer forma o mapeamento entre nomes padrão RFC 822 e nomes de máquinas UUCP é necessário. A forma de fazê-lo é completamente dependente da máquina. Uma forma comum de mapeamento FQDN para UUCP é o uso do arquivo de caminhos alternativos:

```
macunaima.ufsj.edu.br  guarapuava!dourados!macunaima!%s
```

Esta configuração irá produzir um caminho bang no estilo UUCP a partir de um endereço no formato de nome de domínio totalmente qualificado. Alguns programas de correio disponibilizam alguns arquivos especiais para isto, como por exemplo o `sendmail`, que usa o arquivo `uucpxtable`.

A transformação reversa (coloquialmente denominada "dominização") é algumas vezes necessária quando se envia uma mensagem a partir de uma rede UUCP para a Internet. Desde que o remetente utilize o nome de domínio totalmente qualificado no endereço de destino, este problema pode ser evitado através da não remoção do nome de domínio do envelope de endereço ao se reenviar a mensagem para uma máquina de otimização de roteamento. De qualquer forma, há ainda os sites UUCP que não formam parte de nenhum domínio. Eles são normalmente "dominizados" através da utilização do pseudo-domínio `uucp`.

13.5 Formato dos Arquivos Caminhos Alternativos e Mapas

A base de dados de caminhos alternativos provê informações de roteamento em redes baseadas em UUCP. Um entrada típica se parece com o seguinte (nomes de sites e caminhos são separados por tabulações):

```
macunaima.ufsj.edu.br guarapuava!dourados!macunaima!%s
macunaima             guarapuava!dourados!macunaima!%s
```

Isso faz com que qualquer mensagem para `macunaima` seja entregue através de `guarapuava` e `dourados`. Ambas as identificações, como o nome totalmente qualificado de `macunaima` e seu nome UUCP, devem ser fornecidas ao programa de correio, caso ele não tenha condições de separar a forma de mapeamento entre esses nomes.

Caso se deseje direcionar todas as mensagens para máquinas dentro de um domínio para seus retransmissores de mensagens, deve-se especificar um caminho na base de dados de caminhos alternativos, fornecendo o nome de domínio como alvo, precedido por um ponto. Por exemplo, se todas as máquinas no domínio `edu.br` podem ser alcançadas através de `parintins!macunaima`, a entrada no arquivo de caminhos alternativos terá o seguinte aspecto:

```
.edu.br         parintins!macunaima!%s
```

Escrever uma arquivo de caminhos alternativos somente é aceitável quando se está administrando um site que não possui muitos roteamentos. Caso seja necessário definir um número muito grande de máquinas, a melhor maneira será usar o comando `pathalias` para criar o arquivo a partir dos arquivos mapas. Mapas podem ser mantidos muito mais facilmente devido à sua simplicidade em se adicionar ou remover um sistema, editando-se a entrada no mapa e recriando-se o arquivo. Apesar dos arquivos publicados no Projeto de Mapeamento Usenet não serem mais usados para rotinas de roteamento, pequenas redes UUCP podem prover informações de roteamento através de seus próprios mapas.

Um arquivo de mapa consiste fundamentalmente de uma lista de máquinas, contendo os sites que o sistema acessa e as máquinas que acessam o sistema local. O nome do sistema começa na coluna um e é seguido por uma lista das conexões separadas por vírgula. Caso a lista necessite de mais de uma linha, a linha seguinte deve necessariamente começar com um caractere de tabulação. Cada conexão consiste de um nome de um site, seguido por um indicador entre parênteses. Este indicador é uma expressão aritmética, formada por números e custos simbólicos. Linhas iniciadas com # deve ser ignoradas, pois são somente comentários.

Como um exemplo, consideremos que `macunaima` conecta-se com `cebolinha.cnclinux.com.br` duas vezes ao dia e com `cveloso.tropicalia.com.br` uma vez por semana. A conexão com `cveloso` é de somente 9600 bps. `macunaima` poderia então publicar um mapa com o seguinte formato

```
macunaima.ufsj.edu.br
        cebolinha.cnclinux.com.br (DAILY/2),
         cveloso.tropicalia.com.br (WEEKLY+LOW)

macunaima.ufsj.edu.br = macunaima
```

A última linha torna a máquina conhecida também pelo seu nome UUCP. Note que deve-se usar `DAILY/2`, porque chamar duas vezes ao dia na verdade divide o custo por dois, caso fosse realizada somente uma conexão.

Usando a informação de tais arquivos de mapas, o comando `pathalias` é capaz de calcular a melhor rota para qualquer site de destino que esteja descrito em mapas e de produzir uma base de dados de caminhos alternativos que podem ser usados no roteamento para estes sites.

O comando `pathalias` disponibiliza algumas funcionalidades tais como esconder sites (por exemplo tornando-os acessíveis somente através de caminho padrão), etc. Veja a página de manual para maiores detalhes, assim como para uma lista completa de indicadores disponíveis.

Comentários nos arquivos de mapas geralmente trazem alguma informação adicional sobre os sites ali descritos. Há um formato rígido para especificá-los, permitindo que eles sejam recuperados. Por exemplo um programa chamado `uuwho` usa uma base de dados criada a partir dos arquivos de mapas para apresentar estas informações de uma forma mais legível para o usuário.

Ao registrar um site em uma organização que distribui arquivos de mapas aos seus membros, deve-se geralmente preencher uma entrada de mapa similar ao exemplo abaixo (na verdade este é o registro do site do autor):

```
#N      monad, monad.swb.de, monad.swb.sub.org
#S      AT 486DX50; Linux 0.99
#O      private
#C      Olaf Kirch
#E      okir@monad.swb.de
#P      Kattreinstr. 38, D-64295 Darmstadt, FRG
#L      49 52 03 N / 08 38 40 E
#U      brewhq
#W      okir@monad.swb.de (Olaf Kirch); Sun Jul 25 16:59:32 MET DST 1993
#
monad   brewhq(DAILY/2)
# Domains
monad = monad.swb.de
monad = monad.swb.sub.org
```

O espaço em branco após os primeiros dois caracteres é uma tabulação. O significado de muitos campos é bastante óbvio, porém normalmente se recebe uma descrição detalhada de qualquer domínio no qual se registre. O campo `L` é o mais interessante de se descobrir: ele fornece a posição geográfica do site no formato latitude/longitude e é usado para gerar mapas em formato postscript que mostram todos os sites em cada País assim como no mundo.[12]

[12] Eles são postados regularmente em `news.lists.ps-maps`. Cuidado: eles são ENORMES.

13.6 Configurando o elm

elm significa "mensagem eletrônica" e é uma das mais interessantes ferramentas Unix. Ela disponibiliza uma interface que ocupa toda a tela com um sistema de ajuda bastante bom. Não discutiremos aqui como usar o programa elm, mas somente algumas de suas opções de configuração.

Teoricamente, pode-se executar o programas elm sem qualquer configuração e tudo funcionará perfeitamente, com um pouco de sorte. Porém há poucas opções que devem ser obrigatoriamente configuradas, apesar de requeridas em somente algumas ocasiões.

Ao iniciar, o programa elm lê um conjunto de variáveis de configuração a partir do arquivo elm.rc em /usr/lib/elm. Então, ele tentará ler o arquivo .elm/elmrc no diretório pessoal do usuário. Este arquivo não deve ser criado pelo próprio usuário. Ele é criado ao se escolher "save options" a partir do menu de opções do programa elm.

O conjunto de opções para um arquivo pessoal do tipo elmrc também está disponível no arquivo global elm.rc. Muitas configurações no arquivo pessoal substituem as definições do arquivo global.

13.6.1 Opções Globais do Programa elm

No arquivo de opções globais (elm.rc do programa elm) devem ser especificadas as definições que pertencem a todo o sistema local. Por exemplo, na Cervejaria Virtual, o arquivo para a máquina aracaju poderia conter o seguinte:

```
#
# Nome da máquina local
hostname = aracaju
#
# Nome do Domínio
hostdomain = .cvirtual.com.br
#
# Nome de domínio totalmente qualificado
hostfullname = aracaju.cvirtual.com.br
```

Estas opções dão ao programa elm uma idéia do nome da máquina local. Apesar desta informação ser raramente usada, deve-se obrigatoriamente configurá-la.

Estas informações em um arquivo pessoal de configuração serão ignoradas.

13.6.2 Conjunto de Caracteres Nacionais

Recentemente foi proposta uma emenda ao padrão da RFC 822 para o suporte a vários tipos de mensagens, tais como texto puro, dados binários, arquivos Postscript, etc. O conjunto de padrões e RFCs que cobrem estes aspectos são comumente referenciados como MIME ou Extensões de Mensagens Internet de Múltiplos Propósitos[13]. Entre outras coisas, isso permite que o destinatário saiba se um conjunto de caracteres diferente de ASCII foi usado ao se escrever uma mensagem recebida, por exemplo usando acentos em Francês, Português ou caracteres do Alemão. Estas funcionalidades são suportadas pelo programa `elm` para algumas extensões.

O conjunto de caracteres usado internamente pelo `Linux` para representação de dados é denominado ISO-8859-1, o qual é o nome do padrão que o descreve. É também conhecido como Latin-1. Quaisquer mensagens usando caracteres deste conjunto terão a seguinte linha no seu cabeçalho:

```
Content-Type:  text/plain; charset=iso-8859-1
```

Um sistema receptor deverá reconhecer este campo e adotar as medidas necessárias para apresentar a mensagem. O padrão para mensagens de texto puro[14] é um valor para `charset` igual a `us-ascii`.

Para poder apresentar as mensagens com um conjunto de caracteres diferentes de ASCII, `elm` deve saber como imprimir estes caracteres. Por padrão, quando o `elm` recebe uma mensagem com um valor em `charset` diferente de `us-ascii` (ou um conteúdo diferente de `text/plain`), ele tenta listar a mensagem usando um comando chamado `metamail`. Mensagens que requerem `metamail` para serem listadas são mostradas com o indicador "`M`" na primeira coluna da tela de visão geral.

Uma vez que o conjunto de caracteres nativos do `Linux` é o ISO-8859-1, não será necessário executar o programa `metamail` para mostrar mensagens neste formato. Caso indicado ao programa `elm` que o sistema suporta ISO-8859-1, ele não utilizará o `metamail`. Isso pode ser feito através da configuração da seguinte opção no arquivo global `elm.rc`:

[13] Multipurpose Internet Mail Extensions
[14] text/plain

```
displaycharset = iso-8859-1
```

Note que deve-se configurar estas opções mesmo quando não se vá enviar ou receber mensagens que contenham caracteres diferentes de ASCII. Isso se deve ao fato de que outros usuários poderão enviar mensagens com o campo `Content-Type:` no cabeçalho da mensagem, independentemente de estarem enviando mensagens somente em formato ASCII.

De qualquer forma, configurar esta opção no arquivo `elm.rc` não é suficiente. O problema é que ao apresentar esta mensagem com um paginador interno, `elm` chama uma função de biblioteca para cada caractere para determinar se ele pode ser apresentado ou não. Por padrão, esta função somente reconhecerá caracteres ASCII e listará todos os demais como "`^?`". Pode-se contornar isso através da configuração da variável de ambiente `LC_CTYPE` para `ISO-8859-1`, o que indica que os caracteres de Latin-1 são considerados como possíveis de serem apresentados. Suporte a esta facilidade está disponível desde a biblioteca `libc-4.5.8`.

Ao enviar mensagens que contêm caracteres especiais do ISO-8859-1, deve-se estar seguro de que as seguintes variáveis no arquivo `elm.rc` foram configuradas:

```
charset = iso-8859-1
textencoding = 8bit
```

Isso faz com que o programa `elm` indique o conjunto de caracteres no cabeçalho da mensagem como ISO-8859-1 e os envie no formato 8 bits (o padrão é dividí-los usando o padrão 7 bits).

Obviamente qualquer uma destas opções pode ser configurada no arquivo pessoal `elmrc` ao invés de se utilizar o arquivo global.

Capítulo 14

Configurando e Executando o `smail`

Este capítulo dará uma breve introdução para a configuração do programa `smail` e uma visão geral das funcionalidades disponíveis. Apesar do `smail` ser bastante compatível com o `sendmail` no seu comportamento, seus arquivos de configuração são completamente diferentes.

O principal arquivo de configuração é denominado `/usr/lib/smail/config`. Deve-se sempre editar este arquivo para refletir os valores específicos do site. Caso se esteja executando somente um site folha UUCP, haverá relativamente pouca coisa para se fazer. Outros arquivos que configuram o roteamento e as opções de transporte também podem ser usadas. Nós lidaremos rapidamente com esses arquivos também.

Por padrão, o programa `smail` processa e entrega todas as mensagens imediatamente após a sua chegada. Caso o tráfego seja relativamente alto, pode-se utilizar o programa `smail` para coletar as mensagens em uma *fila* e processá-los somente em intervalos regulares.

Ao lidar com mensagens em uma rede TCP/IP, `smail` é freqüentemente executado no modo servidor: durante a inicialização do sistema ele é normalmente acionado pelo programa `rc.inet2`[1] em modo de execução de segundo plano onde ele fica aguardando por conexões TCP endereçadas à porta SMTP local (normalmen-

[1] exemplo utilizando o slackware

te a porta 25). Isso é interessante à medida que se espere ter uma quantidade significativa de tráfego, uma vez que o programa `smail` não será inicializado separadamente para cada conexão solicitada. Uma alternativa seria ter-se o programa `inetd` gerenciando a porta SMTP e acionando `smail` toda vez que houvesse uma conexão nesta porta.

O `smail` tem uma série de indicadores que controlam o seu comportamento, porém descrevê-los em detalhes pode não ajudar muito. Felizmente ele suporta um grande número de modos padrão de operação que são habilitados quando se aciona o programa através de um nome especial na linha de comando, como por exemplo `rmail` ou `smtpd`. Normalmente estes nomes alternativos são ligações simbólicas para o próprio binário do smail. Encontraremos muitos deles na discussão das várias funcionalidades do `smail`.

Há duas ligações simbólicas que devem estar configuradas sob quaisquer circunstâncias; chamadas `/usr/bin/rmail` e `/usr/sbin/sendmail`.[2] Ao se compor e enviar uma mensagem com um agente de usuário como o programa `elm`, a mensagem será enviada para o programa `rmail` para entrega, com a lista de destinatários informada na linha de comando. O mesmo acontece com a chegada de mensagens via UUCP. Algumas versões do `elm` acionam o programa `/usr/sbin/sendmail` ao invés de `rmail`, tornando necessários ambos. Por exemplo, caso se mantenha o `smail` em `/usr/local/bin`, deve-se utilizar o seguinte na linha do interpretador de comandos:

```
# ln -s /usr/local/bin/smail /usr/bin/rmail
# ln -s /usr/local/bin/smail /usr/sbin/sendmail
```

Para maiores informações sobre a configuração do programa `smail`, por favor verifique as páginas de manual `smail(1)` e `smail(5)`. Caso elas não estejam inclusas na sua distribuição `Linux`, pode-se obter estas informações a partir dos fontes do `smail`.

14.1 Configuração UUCP

Para usar o programa `smail` em um ambiente exclusivamente UUCP, a instalação básica é bastante simples. Primeiro, esteja certo de que as ligações simbólicas

[2] Este é o novo padrão de localização do programa `sendmail` de acordo com o Padrão de Sistemas de Arquivos `Linux`. Outra localização comum é em `/usr/lib`.

descritas acima estão perfeitamente configuradas. Caso se espere receber lotes de mensagens SMTP de outros sites, deve-se criar uma ligação também para o programa `rsmtp`.

Na distribuição de Vince Skahan do programa `smail` será possível encontrar um arquivo exemplo de configuração. Ele é denominado `config.sample` e está localizado em `/usr/lib/smail`. Basta copiá-lo com o nome de `config` e editá-lo para que contenha os valores específicos do site local.

Assumindo que o site tenha o nome de `lindoia.cnclinux.com.br` e esteja registrado nos mapas UUCP como `lindoia`, tendo como máquina de otimização de roteamento a máquina `iracema`, então o arquivo `config` terá o seguinte conteúdo:

```
#
# Nome dos domínios
visible_domain=cnc.linux:uucp
#
# Nome de envio de mensagens
visible_name=lindoia.cnclinux.com.br
#
# Nome UUCP
uucp_name=lindoia.cnclinux.com.br
#
# Máquina de otimização de roteamento
smart_host=iracema
```

O primeiro comando indica ao `smail` os nomes dos domínios aos quais a máquina pertence. Os nomes podem ser inseridos aqui, separados por dois pontos. Caso o site tenha um nome registrado nos Mapas UUCP, deve-se adicionar a palavra `uucp`. Ao lidar com uma mensagem, `smail` determina o nome da máquina local através da chamada de sistema `hostname(2)` e verifica se os endereços dos destinatários não contêm este nome. Caso o endereço coincida ou o endereço do destinatário não esteja qualificado, este será considerado local e o programa `smail` tentará entregar a mensagem para o usuário ou apelido na máquina local. De outra forma o destinatário é considerado remoto e será tentado o processo de entrega para a máquina de destino.

O parâmetro `visible_name` deve conter o nome totalmente qualificado do site, a ser usado nas mensagens enviadas. Este nome é utilizado na geração do endereço do remetente nas mensagens enviadas. Deve-se estar certo de que o programa `smail` reconhece o endereço como sendo da máquina local (por exemplo o nome de máquina de um dos domínios listados no parâmetro `visible_domain`). De outra forma as respostas às mensagens serão devolvidas ao remetente.

O último comando configura o caminho a ser usado para o roteamento via máquina de otimização (descrita na seção 13.4). Com este arquivo de exemplo de configuração, `smail` irá reenviar toda mensagem destinada a um endereço remoto para a máquina de otimização de roteamento. O caminho especificado em `smart_path` será usado como rota. Uma vez que as mensagens serão entregues via UUCP, este atributo deve especificar um sistema conhecido pelo programa UUCP. Por favor consulte o capítulo 12 para saber como fazer para que o UUCP possa reconhecer uma máquina.

Há uma opção no arquivo acima ainda não explicada: `uucp_name`. A razão para o uso desta opção baseia-se no fato do `smail`, por padrão, utilizar um valor retornado de `hostname(2)` para assuntos específicos do UUCP, como o caminho de retorno dado pela linha de cabeçalho `From_`. Caso o nome da máquina *não* esteja registrado no projeto de mapeamento UUCP, deve-se indicar para o `smail` a obrigatoriedade de utilização do nome totalmente qualificado.[3] Isso pode ser feito através da adição da opção `uucp_name` ao arquivo `config`.

Há outro arquivo em `/usr/lib/smail`, chamado `paths.sample`. Ele é um exemplo de como o arquivo `paths` pode parecer. De qualquer forma, ele não será necessário a menos que se tenha conexões para troca de mensagens com mais de um site. Caso este seja o caso, o arquivo deverá ser escrito por conta própria ou através da geração de um arquivo a partir dos Mapas da Usenet. O arquivo `paths` será descrito posteriormente neste capítulo.

14.2 Configuração em Uma Rede Local

Caso você esteja configurando um site com duas ou mais máquinas conectadas em uma rede local, uma das máquinas deve ser designada como a responsável pela conexão UUCP com o mundo exterior. Entre as máquinas na rede local, as mensagens serão provavelmente trocadas através de uma conexão SMTP sobre TCP/IP. Retornando ao exemplo da Cervejaria Virtual e considerando a máquina

[3] Assumindo que o nome da máquina seja `curitiba`, mas não esteja registrada nos mapas, de qualquer forma pode haver um site já registrado utilizando este nome, fazendo com que toda mensagem destinada a `parolin!root`, mesmo quando enviada de uma máquina vizinha, seja enviada para a máquina registrada: `curitiba`. Isso pode ser um problema para todos os envolvidos.

`aracaju` como o caminho padrão UUCP.

Em um ambiente de rede, a melhor forma de manter as caixas postais é tê-las em um único sistema de arquivos, o qual pode ser montado via NFS em todas as máquinas da rede. Isso permite aos usuários moverem-se de uma máquina para outra, sem ter que mover as suas caixas postais (ou ainda pior, verificar em três ou quatro máquinas diferentes eventuais mensagens que tenham chegado). Pode-se ainda querer utilizar endereços de remetentes independentes da máquina onde as mensagens foram geradas. É uma prática comum usar somente o nome do domínio como parte do endereço do remetente sem a inclusão do nome da máquina. Por exemplo, Machado de Assis, por exemplo, poderia usar o endereço `machado@cvirtual.com.br` ao invés de `machado@maceio.cvirtual.com.br`. Explicaremos a seguir como fazer para que um servidor reconheça o nome de domínio como nome válido para a máquina.

Uma forma diferente de manter todas as caixas postais em uma máquina central é o uso do POP ou IMAP. POP significa *Protocolo de Agência de Correio*[4] e permite que os usuários acessem suas caixas postais sobre uma simples conexão TCP/IP. IMAP, o *Protocolo Interativo de Acesso a Mensagens*[5], é similar ao POP, porém mais genérico. Tanto os clientes como os servidores para IMAP e POP foram portados para o `Linux` e estão disponíveis em `metalab.unc.edu` sob o caminho `/pub/Linux/system/Network`.

14.2.1 Gravando os Arquivos de Configuração

A configuração para a Cervejaria Virtual funciona da seguinte forma: todas as máquinas exceto o servidor de correio `aracaju` roteiam todas as mensagens a serem enviadas para o servidor, utilizando a sistemática de roteamento otimizado de máquina. `aracaju` enviará todas as mensagens de saída para a máquina que realmente faz o roteamento otimizado de todas as mensagens da Cervejaria. Esta máquina é chamada `santos`.

O arquivo padrão de configuração `config` para todas as máquinas da rede exceto `aracaju` terá o seguinte aspecto:

```
#
# Domínio local:
visible_domain=cvirtual.com.br
```

[4] Post Office Protocol

[5] Interactive Mail Access Protocol

```
#
# Denominação local do domínio:
visible_name=cvirtual.com.br
#
# Roteamento otimizado: via SMTP para aracaju
smart_path=aracaju
smart_transport=smtp
```

Esta configuração é muito similar à utilizada para sites no estilo UUCP. A principal diferença é que o transporte usado para o envio de mensagens para a máquina de roteamento otimizado é, obviamente, o SMTP. O atributo `visible_domain` faz com que o programa `smail` use o nome do domínio ao invés do nome da máquina local no envio de mensagens.

Em uma máquina UUCP que atue como caminho padrão de uma rede, como por exemplo em `aracaju`, o arquivo `config` tem a seguinte aparência:

```
#
# Nomes da máquina local:
hostnames=cvirtual.com.br:aracaju.cvirtual.com.br:aracaju
#
# Nome utilizado pela máquina local:
visible_name=cvirtual.com.br
#
# Nome utilizado para conexões UUCP
uucp_name=cvirtual.com.br
#
# Transporte otimizado via UUCP até a máquina santos
smart_path=santos
smart_transport=uux
#
# Domínios pelos quais a máquina local responde autoritativamente
auth_domain=cvirtual.com.br
```

Este arquivo `config` usa um esquema diferente para dizer ao `smail` qual é o nome da máquina local. Ao invés de fornecer uma lista de domínios e deixar que ele descubra como encontrar o nome da máquina através de uma chamada ao sistema, ele especifica a lista explicitamente. A lista acima contém tanto o nome totalmente qualificado, quanto o nome curto, além do domínio isoladamente. Isso faz com que `smail` reconheça `machado@cvirtual.com.br` como um endereço local e entregue a mensagem para o usuário `machado`.

A variável `auth_domains` define os domínios sob os quais a máquina `aracaju` é considerada o servidor autoritativo. Isso é, se o `smail` receber qualquer mensagem endereçada a *nome_da_máquina*`.cvirtual.com.br` onde *nome_da_máquina* não se refere à uma máquina existente, ele rejeitará a mensagem e a devolverá ao remetente. Caso esta opção não esteja configurada, qualquer mensagem será enviada para a máquina responsável pelo roteamento otimizado, a qual a retornará para `aracaju` e assim por diante, até que ela seja descartada por exceder o número máximo de transmissões.

14.2.2 Executando smail

É necessário definir se o programa `smail` será executado como um servidor em separado, ou se o servidor `inetd` administrará a porta SMTP e acionará `smail` sempre que uma conexão SMTP for solicitada por algum cliente. Normalmente, será preferível utilizar o programa como um servidor de correio em separado, uma vez que a carga de máquina será muito menor do que a ativação de uma instância de `smail` para cada simples conexão. Como o servidor de mensagens busca entregar as mensagens diretamente na caixa postal do destinatário, provavelmente o uso do `inetd` será o mais indicado para as outras máquinas.

Independente do modo de operação escolhido para cada máquina individualmente, deve-se assegurar que a seguinte entrada está configurada no arquivo `/etc/services`:

```
smtp            25/tcp         # Protocolo de Simples Transferência de Mensagens
```

Isto define o número da porta TCP que o `smail` deverá usar para conversações SMTP. O padrão é 25 na RFC Definindo Números.

Ao ser executado em modo servidor, `smail` será executado automaticamente em segundo plano, aguardando conexões na porta SMTP. Quando a conexão ocorrer, ele executa uma nova cópia do programa, a qual conduzirá a conversação SMTP. O servidor `smail` é normalmente inicializado através do programa `rc.inet2`[6] utilizando-se o seguinte comando:

```
/usr/local/bin/smail -bd -q15m
```

A opção `-bd` ativa o modo servidor e a opção `-q15m` provoca o processamento a cada 15 minutos das mensagens que estejam na fila.

[6] arquivo de configuração do Slackware

Caso deseje-se utilizar o programa `inetd`, o arquivo `/etc/inetd.conf` deverá conter uma linha similar ao seguinte:

```
smtp    stream  tcp nowait  root  /usr/sbin/smtpd smtpd
```

`smtpd` deverá ser uma ligação simbólica para o binário do programa `smail`. Lembre-se de fazer com que o programa `inetd` leia novamente o arquivo `inetd.conf`. Para tanto basta enviar o sinal `HUP` após a efetivação das mudanças.

O modo servidor e o uso do `inetd` são mutuamente exclusivos. Caso se execute o programa `smail` em formato servidor, deve-se estar seguro de que qualquer linha no arquivo `inetd.conf` contendo o serviço `smtp` esteja comentada. O contrário é verdadeiro, ou seja quando se quer utilizar o servidor `inetd` deve-se estar seguro de que o programa `rc.inet2` não inicializa o servidor `smail`.

14.3 Quando as Coisas Não Funcionam...

Caso algum problema ocorra com a instalação, existem algumas ferramentas que podem auxiliar a encontrar a raiz do problema. O primeiro local que deve ser examinado são os arquivos de histórico do `smail`. Eles são mantidos no diretório `/var/spool/smail/log` e são denominados `logfile` e `paniclog`, respectivamente. O primeiro contém registros de todas as transações e o segundo traz somente as mensagens de erro relacionadas com a configuração e assuntos similares.

Uma entrada típica no arquivo `logfile` tem o seguinte formato:

```
04/24/94 07:12:04: [m0puwU8-00023UB] received
|            from: root
|         program: sendmail
|            size: 1468 bytes
04/24/94 07:12:04: [m0puwU8-00023UB] delivered
|             via: aracaju.cvirtual.com.br
|              to: root@aracaju.cvirtual.com.br
|         orig-to: root@aracaju.cvirtual.com.br
|          router: smart_host
|       transport: smtp
```

Indica que uma mensagem enviada pelo superusuário `root` para `root@aracaju.cvirtual.com.br` foi adequadamente recebida pela máquina `aracaju` utilizando SMTP.

Mensagens que não puderam ser entregues pelo `smail`, geram uma mensagem de erro no local dos dados de entrega:

```
04/24/94 07:12:04: [m0puwU8-00023UB] received
|           from: root
|        program: sendmail
|           size: 1468 bytes
04/24/94 07:12:04: [m0puwU8-00023UB] root@aracaju.cvirtual.com.br ... deferred
 (ERR_148) transport smtp: connect: Connection refused
```

O erro acima é um erro típico, que ocorre quando o programa `smail` reconhece adequadamente que deve enviar a mensagem para `aracaju`, porém este não tem os serviços SMTP adequadamente configurados. Esse problema é causado possivelmente por problemas de configuração ou por falta de suporte TCP nos binários do `smail`.

Este é um problema que pode ocorrer com certa freqüência. Há binários pré-compilados do `smail`, inclusive em distribuições `Linux`, sem suporte a redes TCP/IP. Caso este seja o caso, será necessário recompilar o programa por conta própria ou buscar uma versão na Internet. Após instalar o programa `smail`, pode-se verificar se o suporte a TCP está ativo através da execução do comando `telnet` descrito a seguir na porta SMTP de sua máquina. Uma conexão bem sucedida com o servidor SMTP é mostrada a seguir:

```
$ telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 aracaju.cvirtual.com.br Smail3.1.28.1 #6 Sun, 23 Jan 99 19:26
MET
QUIT
221 aracaju.cvirtual.com.br closing connection
```

Caso o teste não produza o aviso SMTP (a linha começando com o número 220), esteja seguro de que a sua configuração está *realmente* correta antes de prosseguir na compilação do `smail`, conforme descrito a seguir.

Ao se encontrar um problema com o `smail` que não seja possível localizar a partir das mensagens de erro geradas, pode-se ativar o modo de depuração do programa. Isso pode ser feito através do indicador `-d`, o qual opcionalmente pode ser seguido por um número que identifica o nível das mensagens que serão apresentadas (não pode haver espaços entre a opção e o nível). `smail` irá gerar um relatório da operação na tela, o que certamente fornecerá mais informações sobre o que está ocorrendo.

[Não Sei,... Mas Talvez as Pessoas Achem Isso Engraçado:] Caso nada mais ajude, pode-se acionar o programa `smail` no modo Rogue, através da opção `-bR` na linha de comando. A página de manual traz as seguintes informações sobre esta opção "Informe o domínio hostil do gigante servidor de mensagens e o padrão RFC irá correr. Tente desativá-lo no protocolo nível 26 e retorne". Apesar desta opção não resolver os seus problemas, ela pode proporcionar algum consolo e conforto.[7]

14.3.1 Compilando o smail

Caso se esteja certo de que o programa `smail` está sem suporte a rede TCP, então deve-se obter os fontes, que provavelmente estão no CDROM da distribuição, ou então transferi-los a partir da Internet.[8]

Ao compilar o programa `smail` é preferível começar com os arquivos de configuração da distribuição `newspak` de Vince Skahan. Para compilar o dispositivo de controle de redes TCP, deve-se configurar a macro `DRIVER_CONFIGURATION` no arquivo `conf/EDITME` para `bsd-network` ou `arpa-network`. O primeiro é requerido por instalações de redes locais, mas a Internet requer `arpa-network`. A diferença entre estas duas reside no fato da última reconhecer os serviços BIND e o registro MX.

14.4 Modos de Entrega de Mensagens

Conforme descrito anteriormente, o programa `smail` é capaz de entregar mensagens imediatamente ou criar uma fila para entrega posterior. Caso se escolha o método de fila de mensagens, `smail` irá armazenar todas as mensagens no diretório `messages` sob `/var/spool/smail`. Elas não serão processadas até que isso seja explicitamente indicado (este método é também conhecido como "executando a fila").

[7] Não utilize esta opção se você estiver de mau humor.

[8] Caso se tenha adquirido o CD de uma distribuição `Linux` de um vendedor, então tem-se direito aos fontes do programa por uma taxa mínima de acordo com as condições de cópias e distribuição do `smail`.

Pode-se selecionar um de três métodos de entrega através da configuração do parâmetro `delivery_mode` no arquivo `config`, o qual poderá ser igual a `foreground`, `background`, ou `queued`. O método `foreground` faz com que a mensagem seja entregue imediatamente após o seu recebimento, o método `background` faz com que seja gerada uma instância do programa para o processamento da mensagem e o método `queued` gera as filas de mensagens. Mensagens recebidas serão sempre colocadas em fila independente desta opção caso a variável `queue_only` esteja configurada no arquivo `config`.

Caso o método de fila seja ativado, esteja seguro de que as filas sejam verificadas periodicamente, provavelmente a cada 10 ou 15 minutos. Caso se esteja executando `smail` no modo servidor, deve-se adicionar a opção `-q10m` à linha de comando, para que a fila seja processada a cada 10 minutos. Alternativamente, pode-se executar `runq` a partir do utilitário `cron` nestes intervalos. `runq` deverá ser uma ligação simbólica para `smail`.

Pode-se listar a fila de mensagens atual acionando-se o programa `smail` com a opção `-bp`. Um processo equivalente pode ser feito através da definição de uma ligação simbólica de `mailq` para `smail`, e após deve ser chamado `mailq`:

```
$ mailq -v
m0pvB1r-00023UB From: root  (in /var/spool/smail/input)
                Date: Sun, 24 Apr 9907:12 MET DST
                Args: -oem -oMP sendmail root@aracaju.cvirtual.com.br
Log of transactions:
 Xdefer: <root@aracaju.cvirtual.com.br> reason: (ERR_148) transport smtp:
 connect: Connection refused
```

Este comando mostra uma única mensagem na fila. O registro da transação (a qual somente é mostrada caso se informe a opção `-v` para o programa `mailq`) pode fornecer informações adicionais sobre a causa da espera para a sua entrega. Caso nenhuma tentativa de entrega tenha sido feita até o momento, nenhum registro da transação será apresentado.

Mesmo caso o método de filas não esteja sendo utilizado, o `smail` irá ocasionalmente colocar mensagens em fila ao encontrar alguma falha na sua transferência. Para conexões SMTP, a razão pode estar em uma máquina que não pode ser alcançada, porém existem casos em que a causa pode residir na ausência de espaço no sistema de arquivos[9]. Portanto deve-se processar a fila no máximo a cada hora (utilizando-se o programa `runq`), senão uma transferência adiada poderá ficar

[9] N.A. - muito comum, quando tudo estava bem e deixou de funcionar repentinamente...

presa na fila eternamente.

14.5 Opções Diversas do Arquivo config

Há um número razoável de opções que podem ser utilizadas no arquivo `config`, algumas, apesar de úteis, não são essenciais para a execução do programa `smail`, portanto não serão discutidas aqui. Ao invés disso, somente mencionaremos algumas das mais significativas:

`error_copy_postmaster` Caso esta variável booleana esteja configurada, qualquer erro irá gerar uma mensagem para o administrador do correio, o portmaster. Normalmente, isto somente é feito para erros de configuração. A variável pode ser ativada através de um sinal de mais (+) antes da variável no arquivo `config`.

`max_hop_count` Caso o número de transmissões de uma mensagem (ou seja o número de máquinas pelas quais ela passou) exceda ao número de transmissões definidas, a tentativa de entregar a mensagem em um sistema remoto provocará uma mensagem de erro e a devolução da mensagem ao remetente. Isso é usado para prevenir a geração de círculos de roteamento, onde as mensagens ficam indo e vindo indefinidamente. O contador de transmissões é geralmente computado a partir do campo numérico na linha `Received:` do cabeçalho da mensagem, mas pode ser configurado manualmente através da opção `-h` na linha de comando. O padrão desta variável é 20.

`postmaster` Define o endereço do administrador do correio - postmaster. Caso o endereço `postmaster` não seja localizado, então este será utilizado. O padrão é enviar as mensagens para `root`.

14.6 Roteamento de Mensagens e Entrega

O `smail` divide a entrega de mensagens em três tarefas diferentes: roteador, diretor e módulo de transporte.

O módulo roteador resolve os endereços remotos, determinando qual a máquina que deve receber a mensagem e qual o transporte que deve ser usado. Dependendo da natureza da conexão, diferentes transportes como UUCP ou SMTP podem ser usados.

Endereços locais são entregues ao módulo diretor que responde pelas tarefas de reenvio e nomes alternativos. Por exemplo, um endereço pode ser um nome alternativo ou uma lista de mensagens, ou o usuário pode querer que as suas mensagens sejam reenviadas para outro endereço. Caso o endereço resultante seja remoto, ele então será resolvido pelo roteador, caso contrário será entregue ao módulo de transporte para entrega local. Na maioria absoluta dos casos a tarefa será a de entregar a mensagem em alguma caixa postal, porém mensagens podem estar em um conector de comandos[10] ou anexadas a algum arquivo qualquer.

O módulo de transporte é responsável pela entrega da mensagem, qualquer que seja o método escolhido. Em caso de falha na entrega da mensagem ele gerará uma mensagem de devolução ou adiará a entrega para um momento posterior.

Com o programa `smail`, há muita liberdade para se configurar estas tarefas. Para cada uma delas, um número de programas de controle está disponível, a partir dos quais se pode escolher o mais adequado às suas necessidades. Pode-se defini-los para o programa `smail` através de alguns arquivos chamados `routers`, `directors` e `transports`, localizados em `/usr/lib/smail`. Caso estes arquivos não existam, padrões razoáveis são assumidos e devem ser adequados à maioria dos sites que utilizem SMTP ou UUCP como transporte. Caso deseje-se modificar a política de roteamento ou modificar o transporte do programa `smail`, deve-se obter um exemplo destes arquivos a partir dos fontes da distribuição,[11] copiar os arquivos de exemplo para `/usr/lib/smail` e modificá-los de acordo com as necessidades. Exemplos de arquivos de configuração podem ser encontrados no Apêndice B.

14.7 Roteando Mensagens

Ao receber uma mensagem, `smail` inicialmente verifica se ela é destinada à rede local ou a um sistema remoto. Caso o destino seja uma das máquinas configuradas no arquivo `config`, esta será enviada para o módulo de transporte. Caso contrário `smail` utiliza um dos programas de controle de roteamento para descobrir qual a máquina que deve receber a mensagem. Isso pode ser descrito no arquivo `routers`, e caso este arquivo não exista, um conjunto de roteadores padrão será utilizado.

A máquina de destino é enviada para todos os roteadores e aquele que encontrar a rota mais específica é selecionado. Considerando-se uma mensagem endereçada

[10] pipe

[11] Os arquivos padrões de configuração podem ser encontrados em `samples/generic` sob o diretório source.

para `ccolombo@ssmaria.caravelas.com.es` e que um roteador conheça a rota padrão para todas as máquinas no domínio `caravelas.com.es`, enquanto outro tem a informação da rota direta para a máquina `ssmaria.caravelas.com.es`. Como o segundo tem informações mais específicas, este será escolhido em detrimento do primeiro. Caso haja dois roteadores que forneçam os melhores resultados de forma idêntica, o que tiver sido encontrado em primeiro lugar no arquivo `routers` será escolhido.

A seguir, o roteador especifica o transporte que será utilizado, por exemplo UUCP e gera o novo endereço de destino. O novo endereço é passado para o transporte juntamente com a máquina para a qual a mensagem deve ser enviada. No exemplo acima, `smail` poderá descobrir que o caminho para `ssmaria.caravelas.com.es` pode ser atingido via UUCP através do caminho `recife!lisboa`. Ele então irá gerar uma destinação igual a `lisboa!smaria.caravelas.com.es!ccolombo` e indicará para o UUCP que use este endereço de envelope a ser passado para a máquina `recife`.

Ao se utilizar a configuração padrão, os seguintes roteadores estarão disponíveis:

- Caso o endereço da máquina de destino possa ser resolvido através das chamadas `gethostbyname(3)` ou `gethostbyaddr(3)`, a mensagem será enviada via SMTP. A única exceção ocorrerá quando o endereço encontrado se referir à máquina local, a qual é administrada pelo módulo diretor também.

  `smail` reconhece também endereços IP em formato decimal e separado por pontos como endereços válidos, desde que eles possam ser resolvidos pela função `gethostbyaddr(3)`. Por exemplo, `amelia@[149.76.12.4]` poderia ser um endereço válido, apesar de pouco usual, para o usuário `amelia` em `quark.fisica.pantanal.edu.br`.

  Caso a máquina esteja conectada a Internet, estes roteadores não atenderão às necessidades, uma vez que não suportam os registros MX. Veja a seguir o que fazer nestes casos.

- Caso `/usr/lib/smail/paths`, o arquivo de caminhos alternativos exista, `smail` irá tentar procurar a máquina de destino neste arquivo (exceto aquelas que contenham o sufixo `.uucp`). Mensagens para um endereço que coincida com este roteador serão entregues via UUCP, usando o caminho encontrado no arquivo.

- O endereço de destino (menos aqueles com sufixo igual a `.uucp`) serão comparados com a saída do comando `uuname` a fim de verificar se a máquina de destino é realmente um vizinho UUCP. Neste caso a mensagem será entregue usando-se o transporte UUCP.

- Caso o endereço não coincida com qualquer um dos roteadores anteriores, a mensagem será entregue para a máquina responsável pela otimização do roteamento. O caminho, assim como o transporte a ser usado estarão definidos no arquivo `config`.

Estes padrões funcionam para as configurações mais simples, mas podem falhar quando os requisitos de roteamento forem mais complexos. Caso este seja o seu caso, será necessária a instalação de roteadores próprios em substituição aos padrões. Um exemplo do arquivo `routers` pode ser encontrado no apêndice B. Algumas distribuições `Linux` também vêm com alguns arquivos de configuração que podem ser um bom ponto de partida nestes casos.

Provavelmente os piores problemas surgem quando uma máquina está em dois universos simultaneamente com discagens IP e conexões UUCP. Deve-se ter os nomes das máquinas no arquivo `hosts`, o qual será acionado ocasionalmente nas conexões SLIP e o programa `smail` irá tentar entregar as mensagens para qualquer uma dessas máquinas via SMTP. Isso não é o que usualmente se quer, uma vez que a conexão SLIP não é ativada regularmente e SMTP é muito mais lento do que o envio via UUCP. Com a configuração padrão, não há como escapar do `smail`.

Pode-se evitar este problema fazendo com que `smail` verifique o arquivo `paths` antes de questionar o resolvedor e colocando todas as máquinas neste arquivo para forçar o envio via UUCP. Caso não se deseje que nenhuma mensagem SMTP seja enviada para aquelas máquinas, pode-se comentá-las nos roteadores.

Outro problema na configuração padrão é que ela não provê roteamento para mensagens Internet, uma vez que o roteador não avalia os registros MX. Para habilitar suporte completo a roteamento de mensagens Internet comente os roteadores definidos e utilize um que use o BIND. De qualquer forma, há binários `smail` incluídos em algumas distribuições `Linux` que não tem suporte a BIND compilado. Caso se habilite o BIND, mas se obtenha uma mensagem no arquivo `paniclog` dizendo "`roteador inet_hosts: programa de controle do bind não localizado`", então será necessário obter os fontes do `smail` e recompilá-los (veja a seção 14.2 acima).

Finalmente, é importante frisar que não é uma boa idéia usar o programa de controle `uuname`. Primeiro porque ele irá gerar mensagens de erro quando o UUCP não estiver instalado, uma vez que o comando `uuname` não será localizado. Em segundo lugar, podem existir mais sites listados no arquivo `Systems` do UUCP, dos que os que realmente executam a troca de mensagens com a máquina atual, como por exemplo, troca exclusivamente de notícias, ou sites com o qual se fazem conexões anônimas ocasionais.

Para prevenir o primeiro problema, pode-se substituir o comando `uuname` por um pequeno programa que simplesmente execute `exit 0`. A solução mais genérica e definitiva, é editar o arquivo `routers` e remover a sua indicação.

14.7.1 A Base de Dados paths

`smail` espera encontrar uma base de dados de caminho alternativos no arquivo `paths` sob `/usr/lib/smail`. Este arquivo é opcional, então caso não se queira executar qualquer roteamento via caminhos alternativos, basta simplesmente remover o arquivo `paths`.

`paths` deve ser um arquivo PATH com entradas que indicam nomes de sites de destinos como caminho UUCP. O arquivo tem que estar ordenado, pois `smail` usa uma pesquisa binária para encontrar um site. Comentários não são permitidos neste arquivo e o nome do site deve estar separado do caminho através de tabulações. Bases de dados de caminhos são discutidas detalhadamente no capítulo 13.

Caso este arquivo seja gerado manualmente, esteja seguro de incluir todos os nomes conhecidos para um determinado site. Por exemplo, se um site é conhecido tanto pelo seu nome UUCP como pelo nome de domínio totalmente qualificado, deve-se adicionar um nome para cada um deles. Este arquivo pode ser ordenado com o auxílio do comando `sort(1)` e conectores de comandos.

Caso o site seja do tipo "folha", então o arquivo `paths` não será necessário, bastando configurar a máquina responsável pela otimização do roteamento no arquivo `config` e deixar a tarefa de roteamento para o alimentador de mensagens.

14.8 Entregando Mensagens para Endereços Locais

Comumente, um endereço local é somente o nome do usuário e nestes casos a mensagem deverá ser entregue na sua caixa postal em `/var/spool/mail/`*`usuário`*. Outros casos incluem apelidos e listas de mensagens além de reenvio de mensagens.

Nestes casos, o endereço local é expandido para um outro endereço ou até mesmo para uma lista de endereços, os quais podem ser locais ou remotos.

Além destes endereços "normais", o programa `smail` pode manusear outros tipos de destinações de mensagens locais, tais como nomes de arquivos e conectores de comandos. Eles não são endereços, não se podendo portanto enviar uma mensagem para, digamos, `/etc/passwd@cvirtual.com.br`. Eles são válidos somente se forem gerados a partir de tarefas de reenvio ou apelidos.

Um *nome de arquivo* deverá começar com uma barra (/) ou um til (~). Este último se refere ao diretório pessoal e pode ser utilizado somente se o nome de arquivo foi retirado do arquivo `.forward` ou foi gerado através do reenvio automático (veja acima). Ao entregar uma mensagem para um arquivo, `smail` anexa a mensagem ao arquivo caso ele já exista, ou executa a sua criação se necessário.

Um *comando com conector* pode ser qualquer comando `Unix` precedido pelo símbolo de conexão (`|`[12]). Isso faz com que o programa `smail` manuseie o comando em um interpretador com seus argumentos, porém sem o conector '`|`'. A mensagem em si será enviada ao comando através da entrada padrão.

Por exemplo, para conectar uma lista de mensagens com um grupo de notícias local, pode-se usar um programa interpretado chamado `gateit` e configurar um apelido local que entregue todas as mensagens da lista de mensagens para o programa usando a expressão "`|gateit`".

Caso a chamada ao programa contenha espaços, eles devem estar entre aspas duplas. Devido a questões de segurança envolvidas, deve-se ter cuidado e não executar o programa, caso o endereço tenha sido obtido de maneira dúbia (por exemplo, caso o arquivo de nomes alternativos da qual o endereço foi obtido possa ser gravado por qualquer usuário).

14.8.1 Usuários Locais

O caso mais comum para um endereço local é que ele signifique a caixa postal de um usuário. Esta caixa postal está localizada em `/var/spool/mail` e tem o nome do usuário, que é também o seu dono, com grupo igual a `mail` e modo igual a 660. Caso ela não exista é criada automaticamente pelo programa `smail`.

Note que `/var/spool/mail` é o local padrão para se colocar os arquivos de caixas postais, porém alguns arquivos podem ter diferentes caminhos pré-compilados,

[12] Conhecido como pipe.

como por exemplo `/usr/spool/mail`. Caso a entrega para os usuários de sua máquina falhe constantemente, deve-se tentar utilizar uma ligação simbólica para `/var/spool/mail`.

Há dois endereços que o programa `smail` exige que existam: `MAILER-DAEMON` e `Postmaster`. Ao devolver uma mensagem para um endereço inatingível, uma cópia é enviada para a conta `postmaster` para exame (no caso em que haja problemas de configuração). A conta `MAILER-DAEMON` é usada como remetente das mensagens devolvidas.

Caso estes endereços não contenham uma conta válida no sistema local, `smail` mapeará `MAILER-DAEMON` para `postmaster`, e `postmaster` para `root`, respectivamente. É indicado que isso seja alterado através da criação de um nome alternativo igual a `postmaster` para o responsável pela manutenção do programa de mensagens.

14.8.2 Reenvio

Um usuário pode redirecionar suas mensagens através de um endereço alternativo usando um dos dois métodos suportados pelo programa `smail`. Um opção é colocar a expressão

```
Forward to destinatário,...
```

na primeira linha da caixa postal do usuário. Isso fará com que todas as mensagens enviadas para o usuário com o reenvio ativado sejam automaticamente enviadas para o(s) destinatário(s) descritos na caixa postal. Alternativamente pode-se criar um arquivo `.forward` no diretório pessoal do usuário, o qual deve conter uma lista de destinatários separados por vírgulas. Neste caso, todas as linhas serão lidas e interpretadas.

Qualquer tipo de endereçamento pode ser usado. Um exemplo prático de um arquivo `.forward` para um período de férias é apresentado a seguir:

```
iuri, "|ferias"
```

O primeiro endereço faz com que as mensagens recebidas sejam enviadas para a caixa postal de `iuri`, enquanto que o comando `ferias` retorna uma breve notificação para o remetente.

14.8.3 Aliases de Arquivos

O `smail` pode lidar com arquivos de apelidos ou nomes alternativos compatíveis com o formato do `sendmail` de Berkeley. Entradas no arquivo de apelidos têm o seguinte aspecto:

apelido: *destinatários*

destinatários é uma lista de endereços separada por vírgulas, que serão substituídas pelo apelido. A lista de destinatários pode continuar na linha seguinte, desde que esta comece com um caractere de tabulação.

Há uma funcionalidade especial que permite ao programa `smail` manusear listas de mensagens a partir de um arquivo de nomes alternativos: caso se especifique "`:include:`*nome_do_arquivo*" como destinatário, `smail` irá ler o arquivo especificado e utilizar o seu conteúdo como uma lista de destinatários.

O arquivo de nomes alternativos é denominado `/usr/lib/aliases`[13]. Caso este arquivo possa ser gravado por todo e qualquer usuário, `smail` não irá entregar mensagens para o interpretador de comandos. Um exemplo deste arquivo é apresentado a seguir:

```
# cvirtual.com.br arquivo /usr/lib/aliases
hostmaster: machado
postmaster: machado
usenet: alencar
# A lista de mensagens do desenvolvimento.
desenvolvimento: linus, cava, acme, aurelio,
        /var/mail/log/desenvolvimento
owner-desenvolvimento: god
# Anúncios de interesse geral são enviados para todo o staff
announce: :include: /usr/lib/smail/staff,
                    /var/mail/log/announce
owner-announce: root
# interligando a lista de mensagens ppp para um grupo de notícias
ppp-list: "|/usr/local/lib/gateit local.lists.ppp"
```

Caso um erro ocorra durante a entrega para um endereço gerado a partir do arquivo `aliases`, `smail` tentará enviar uma cópia da mensagem de erro para o

[13] /etc/aliases no Conectiva Linux

"dono do nome alternativo". Por exemplo, caso o envio para `cava` falhe na entrega da mensagem para a lista de mensagens `desenvolvimento`, uma cópia de erro da mensagem será enviada para o remetente, assim como para o `postmaster` e `owner-desenvolvimento`. Caso o endereço do dono não exista, não será gerada mensagem de erro adicional.

Quando entrega mensagens em arquivo ou ao acionar programas no arquivo `aliases`, `smail` se torna o usuário `nobody` a fim de evitar quaisquer problemas de segurança. Especialmente ao gerar arquivos, isso pode não fazer sentido. No arquivo acima fornecido, por exemplo, os arquivos de históricos devem pertencer e serem gravados por `nobody`, ou o processo de registro falhará.

14.8.4 Listas de Mensagens

Ao invés de usar o arquivo aliases, listas de mensagens podem ser administradas através de arquivos residentes no diretório `/usr/lib/smail/lists`. Uma lista de mensagens conhecida como `gar-bugs` é descrita pelo arquivo `lists/gar-bugs`, o qual pode conter os endereços dos membros separados por vírgulas. A lista pode ser formada por múltiplas linhas, com comentários sendo introduzidos através um sinal numérico (#) no início da linha.

Para listas de mensagens, um usuário (ou apelido) denominado `owner-nome_da_lista` deve existir, quaisquer erros que ocorram na resolução de endereços são comunicados a este usuário. Este endereço é também usado como endereço de remetente de todas as mensagens enviadas no campo `Sender:` do cabeçalho da mensagem.

14.9 Transportes Baseados em UUCP

Há um grande número de módulos de transporte compilados para `smail` que utilizam o conjunto de ferramentas UUCP. Neste ambiente, mensagens são normalmente enviadas acionando-se o programa `rmail`, fornecendo-se a mensagem na entrada padrão e criando-se um endereço na linha de comando. Na máquina local, `rmail` poderá ser uma ligação simbólica com o comando `smail`.

Ao manusear uma mensagem em um transporte UUCP, `smail` converte o endereço de destino para o formato UUCP. Por exemplo, `usuário@máquina` irá ser transformado em `máquina!usuário`. Qualquer ocorrência de um operador de endereços '`%`'

é preservada, então o endereço `usuário%máquina@caminho_padrão` irá se tornar `caminho_padrão!usuário%máquina`. De qualquer forma, `smail` nunca gerará este tipo de endereço por si só.

Alternativamente, `smail` pode enviar ou receber lotes de mensagens BSMTP, via UUCP. Desta forma, uma ou mais mensagens são "embaladas" em um único lote que contém os comandos para que o servidor de correio local possa tratá-las como se uma conexão real SMTP houvesse sido estabelecida. BSMTP é freqüentemente usada em redes que utilizem a sistemática "armazenamento e reenvio" objetivando a economia de espaço em disco, assim como possibilita que máquinas que não tenham uma conexão permanente possam receber suas mensagens, quando uma ligação for realizada. O arquivo `transports` de exemplo apresentado no apêndice B contém um transporte `bsmtp` que gera lotes parciais BSMTP em um diretório de filas. Eles serão combinados em lotes finais posteriormente, utilizando-se um programa interpretado que adiciona os comandos `HELO` e `QUIT` adequados.

Para habilitar o transporte `bsmtp` para conexões UUCP, deve-se ter o arquivo denominado *method* (por favor verifique a página de manual do `smail(5)` para maiores detalhes). Caso tenha-se somente uma conexão UUCP e se utilize a máquina de otimização de roteamento, pode-se habilitar os lotes SMTP através da configuração do parâmetro `smart_transport` para `bsmtp` ao invés de `uux`.

Para receber lotes SMTP sobre UUCP, deve-se estar seguro de se ter o comando de "desempacotamento" de mensagens capaz de abrir o lote enviado pela máquina remota. Caso o sistema remoto utilize o programa `smail`, será necessário criar uma ligação simbólica de `rsmtp` para `smail`. Caso o sistema remoto execute o `sendmail`, deve-se adicionalmente instalar um programa interpretado chamado `/usr/bin/bsmtp` que simplesmente executa o comando "`exec rsmtp`" (uma ligação simbólica não irá funcionar neste caso).

14.10 Transportes Baseados em SMTP

O `smail` atualmente suporta um programa de controle SMTP para entrega de mensagens sobre conexões TCP.[14] Ele é capaz de entregar uma mensagem para qualquer número de endereços em uma única máquina, com o nome da máquina sendo especificado ou como um nome totalmente qualificado que pode ser resolvido por uma programa de rede, ou no formato de endereços IP com notação decimal,

[14] Os autores denominam este suporte como "simples". Uma nova versão do programa `smail`, contém um suporte completo que administra este tipo de conexão mais eficientemente.

separada por pontos e mantida entre colchetes. Geralmente, endereços resolvidos por programas de controle de roteamento baseados em BIND, `gethostbyname(3)` ou `gethostbyaddr(3)` serão entregues através de um transporte.

Os programas de controle do SMTP tentarão conectar-se à máquina remota através da porta `smtp` listada no arquivo `/etc/services`. Caso isso não possa ser feito ou a conexão seja desfeita por excesso de tempo de espera, a entrega será novamente tentada posteriormente.

A entrega de mensagens na Internet requer que o roteamento para a máquina de destino seja especificado no formato *route-addr* descrito no capítulo 13, diferentemente do formato UUCP.[15] `smail` irá transformar o endereço `usuário\%máquina@caminho_padrão`, onde `caminho_padrão` é alcançado através da rota `máquina1!máquina2!máquina3` em um endereço com o seguinte formato:

```
<@máquina2,@máquina3:usuário\%máquina@caminho\_padrão>
```

o qual enviará a mensagem com este envelope para `máquina1`. Para habilitar esta transformação (juntamente com o programa de controle BIND), deve-se editar a entrada para o programa de controle do `smtp` no arquivo `transports`. Um arquivo de exemplo `transports` é fornecido no Apêndice B.

14.11 Definição de Nome de Máquina

Algumas vezes é desejável tratar endereços não qualificados (ou seja aqueles que não têm o nome de domínio) especificado no remetente ou no destinatário, por exemplo quando se conecta duas redes, onde uma delas requer um nome totalmente qualificado. Caso haja uma conexão UUCP-Internet, nomes de máquinas podem ser mapeadas para o domínio `uucp` por padrão.

O arquivo `/usr/lib/smail/qualify` indica ao `smail` quais nomes de domínios são mantidos em quais máquinas. Entradas no arquivo `qualify` consistem de um nome de máquina começando na primeira coluna, seguido por um nome de domínio. Linhas contendo um sinal numérico (#) na primeira posição diferente de espaços são comentários. As entradas são pesquisadas na ordem em que aparecem no arquivo.

Caso um arquivo `qualify` não exista, nenhuma qualificação do nome da máquina será executada.

[15] Não é aconselhável o uso de rotas na Internet e sim o nome qualificado de domínio.

Um nome de máquina especial igual a * gera uma coincidência com qualquer nome de máquina, fazendo com que se possa mapear todas as máquinas em um domínio padrão. Ele deve ser usado somente como a última entrada do arquivo.

Na Cervejaria Virtual, todas as máquinas foram configuradas para usarem um nome de domínio totalmente qualificado no endereço do remetente. Destinatários com endereços não qualificados são considerados como pertencentes ao domínio UUCP, sendo necessária somente uma entrada no arquivo `qualify`.

```
# /usr/lib/smail/qualify, última mudança Feb 12, 1999 por rodrigo
#
*           uucp
```

Capítulo 15

Sendmail+IDA

15.1 Introdução ao Sendmail + IDA

Tem sido dito que não se é um *real* administrador de sistemas `Unix` até que se tenha editado um arquivo `sendmail.cf`. Também é dito que há que ser um pouco insano para fazê-lo duas vezes `:-)`.

O sendmail é um programa incrivelmente poderoso. E também difícil de aprender e entender segundo muitas pessoas. Qualquer programa cuja referência definitiva pode ser encontrada em um livro chamado *Sendmail*, (publicado por O'Reilly and Associates) com 792 páginas deve ser justificadamente assustador para qualquer pessoa.

Sendmail+IDA é diferente. Ele elimina a necessidade de editar sempre o arquivo crítico `sendmail.cf` e permite ao administrador definir roteamentos específicos para determinados sites e configurações de endereçamento através de arquivos de suporte mais legíveis chamados *tabelas*. Utilizar sendmail+IDA pode economizar diversas horas de trabalho e evitar muito stress.

Comparado com outros dos principais agentes de transportes de mensagens, provavelmente não há nada mais rápido e simples que o sendmail+IDA. Tarefas típicas que são necessárias para administrar o UUCP ou um site Internet podem se tornar bastante simples. Configurações que normalmente são extremamente difíceis, tornam-se simples de serem configuradas e mantidas.

No momento em que este guia está sendo traduzido a versão atual é `sendmail8.9.1a` e está disponível para FTP anônimo em `vixen.cso.uiuc.edu`. Pode ser compilado sem pacotes adicionais em qualquer sistema `Linux`.

Todos os arquivos de configuração requeridos para se compilar sendmail+IDA, instalá-los e executá-los sob `Linux` estão incluídos no pacote `newspak-2.2.tar.gz`, o qual também está disponível via FTP anônimo em `metalab.unc.edu` no diretório `/pub/Linux/system/Mail`.

15.2 Visão Geral do Arquivo de Configuração

O sendmail tradicional é configurado através de uma arquivo normalmente denominado `/etc/sendmail.cf` (ou `/usr/lib/sendmail.cf`), que não se assemelha à nenhuma linguagem que você tenha visto antes. Editar um arquivo `sendmail.cf` para que se tenha um determinado comportamento pode ser uma experiência complexa.

O sendmail+IDA torna este problema coisa do passado ao tornar as opções de configuração orientadas ao formato de tabelas com um fácil entendimento de sua sintaxe. Estas opções são configuradas através da execução do programa `m4` (um processador de macros) ou pelo programa `dbm` (um processador de bases de dados) em diversos arquivos de dados via Makefiles fornecidos com os fontes.

Um arquivo `sendmail.cf` define somente o comportamento padrão do sistema. Virtualmente todas as customizações especiais são feitas através de tabelas opcionais, fora do arquivo `sendmail.cf`. As tabelas de arquivos de suporte do `sendmail` são as seguintes:

`mailertable` Define comportamentos especiais para máquinas remotas ou domínios.

`uucpxtable` Força a entrega de mensagens UUCP para máquinas que estão no formato DNS.

`pathtable` Define o estilo de endereçamento UUCP para máquinas remotas ou domínios.

`uucprelays` Define a localização dos caminhos alternativos para as máquinas remotas bem conhecidas.

`genericfrom` Converte endereços internos em endereços genéricos visíveis ao mundo exterior.

`xaliases` Converte endereços genéricos para/de endereços válidos.

`decnetxtable` Converte endereços RFC 822 para endereços no estilo DECnet.

15.3 O Arquivo `sendmail.cf`

Com sendmail+IDA o arquivo `sendmail.cf` não é editado diretamente, mas é gerado a partir de um programa de configuração disponibilizado pelo administrador do sistema para o programa `m4`. Nas seções seguintes iremos nos referir a este arquivo como `sendmail.m4`.

Este arquivo contém algumas definições e alguns apontadores para as tabelas onde a real configuração acontece. Em geral, é necessário especificar somente:

- Os caminhos e nomes de arquivos usados no sistema local.
- O(s) nome(s) pelos quais o site é conhecido para o propósito de envio de mensagens.
- O servidor de correio padrão (e talvez uma máquina de retransmissão).

Há uma grande variedade de parâmetros que podem ser definidos para estabelecer o comportamento do site local ou para redefinir os itens de configuração pré-compilados. Estas opções de configuração são identificadas no arquivo `ida/cf/OPTIONS` no diretório fonte.

Um arquivo `sendmail.m4` com uma configuração mínima (UUCP ou SMTP com todas as mensagens locais sendo enviadas diretamente para a máquina responsável pelo roteamento) terá um tamanho entre 10 e 15 linhas excluindo-se os comentários.

15.3.1 Um Exemplo do Arquivo `sendmail.m4`

Um arquivo `sendmail.m4` da máquina `aracaju` para a Cervejaria Virtual é apresentado a seguir. `aracaju` usa o SMTP para conectar-se com todas as máquinas da rede local da Cervejaria e envia todas as mensagens endereçadas para outras destinos através da máquina `parintins`, a máquina retransmissora Internet, via UUCP.

```
dnl #------------------ ARQUIVO DE EXEMPLO SENDMAIL.M4  ------------------
dnl # (a expressão 'dnl' em m4 é equivalente à criação de um comentário)
dnl #  geralmente não é necessário substituir LIBDIR
dnl #define(LIBDIR,/usr/local/lib/mail)dnl    # caminho dos arquivos de suporte
define(LOCAL_MAILER_DEF, mailers.linux)dnl    # servidor de entrega local
define(POSTMASTERBOUNCE)dnl                   # end. de envio das devoluções
define(PSEUDODOMAINS, BITNET UUCP)dnl         # evita o uso do DNS
dnl #-------------------------------------------------------------
dnl #
define(PSEUDONYMS, aracaju.cvirtual.com.br  aracaju.UUCP cvirtual.com.br)
dnl                                    # nomes pelos quais a máquina é conhecida
define(DEFAULT_HOST, aracaju.cvirtual.com.br)
dnl                                    # nome primário para envio de mensagens
define(UUCPNAME, aracaju)dnl           # nome UUCP
dnl #
dnl #-------------------------------------------------------------
dnl #
define(UUCPNODES, |uuname|sort|uniq)dnl       # nossos vizinhos UUCP
define(BANGIMPLIESUUCP)dnl                    # garante o uso do formato UUCP
define(BANGONLYUUCP)dnl                       # tratamento de mensagens
define(RELAY_HOST, parintins)dnl              # máquina de roteamento otimizado
define(RELAY_MAILER, UUCP-A)dnl               # como alcançar parintins via uucp
dnl #
dnl #--------------------------------------------------------------------
dnl #
dnl # tabelas
dnl #
define(ALIASES, LIBDIR/aliases)dnl            # apelidos do sistema
define(DOMAINTABLE, LIBDIR/domaintable)dnl    # máquinas do domínio
define(PATHTABLE, LIBDIR/pathtable)dnl        # caminhos das bases de dados
define(GENERICFROM, LIBDIR/generics)dnl       # endereços genéricos
define(MAILERTABLE, LIBDIR/mailertable)dnl    # serv. correio por máq. ou domínio
define(UUCPXTABLE, LIBDIR/uucpxtable)dnl      # caminhos para máquinas servidas
define(UUCPRELAYS, LIBDIR/uucprelays)dnl      # atalhos dos caminhos de envio
dnl #
dnl #--------------------------------------------------------------------
dnl #
dnl # inclui o código "real" que faz com tudo funcione
dnl # (provido com o código fonte)
dnl #
include(Sendmail.mc)dnl                         # INFORMAÇÃO OBRIGATÓRIA !!!
dnl #
dnl #------------ FIM DO ARQUIVO DE EXEMPLO SENDMAIL.M4 -------
```

15.3.2 Parâmetros Tipicamente Usados no Arquivo `sendmail.m4`

Alguns poucos itens do arquivo `sendmail.m4` são sempre necessários, outros podem ser ignorados caso se utilizem os padrões. As seções seguintes descrevem cada um dos itens do arquivo de exemplo `sendmail.m4` em maiores detalhes.

Itens Que Definem Caminhos

```
dnl #define(LIBDIR,/usr/local/lib/mail)dnl  # local dos arquivos de suporte
```

LIBDIR define o diretório onde o sendmail+IDA espera encontrar os arquivos de configuração, as diversas tabelas de bases de dados e as definições das localizações especiais. Em uma distribuição típica, isso é compilado junto com os binários do sendmail e não necessita ser explicitado no arquivo sendmail.m4.

O exemplo acima tem a expressão dnl no início da linha, o que significa que esta linha é um comentário somente para fins informativos.

Para mudar a localização dos arquivos de suporte para um local diferente, basta remover o parâmetro dnl inicial, configurar o caminho para a localização esperada, reconstruir e reinstalar o arquivo sendmail.cf.

Definindo o Servidor de Correio Local

```
define(LOCAL_MAILER_DEF, mailers.linux)dnl  # servidor de entrega local
```

Muitos sistemas operacionais disponibilizam programas destinados à entrega de mensagens. Programas típicos para a maioria das variantes do Unix já estão disponíveis no sendmail.

No Linux, é necessário explicitar uma instrução "define", indicando o servidor de mensagens apropriado, caso um programa de entrega local não esteja presente na distribuição instalada. Isso é feito através da especificação do parâmetro LOCAL_MAILER_DEF no arquivo sendmail.m4.

Por exemplo, para se usar o programa deliver[1], deve-se configurar a variável LOCAL_MAILER_DEF para mailers.linux.

O seguinte arquivo deve ser então criado e nomeado como mailers.linux no diretório apontado em LIBDIR. Ele explicitamente define o programa deliver na palavra chave Mlocal com os parâmetros adequados fazendo com que o sendmail entregue corretamente as mensagens para o sistema local. A menos que você seja um expert em sendmail, a mudança destes parâmetros será indesejável. Provavelmente não se desejará mudar estes parâmetros.

[1] deliver foi escrito por Chip Salzenberg (chip%tct@ateng.com). Integra diversas distribuições Linux e pode ser encontrado nos sites FTP usuais, tais como ftp.uu.net.

```
# -- /usr/local/lib/mail/mailers.linux --
#     (servidores de correio local para uso no Linux )
Mlocal, P=/usr/bin/deliver, F=SlsmFDMP, S=10, R=25/10, A=deliver $u
Mprog,  P=/bin/sh,       F=lsDFMeuP,   S=10, R=10, A=sh -c $u
```

Há ainda um padrão pré-construído para o parâmetro `deliver` no arquivo `Sendmail.mc` que é incluído no arquivo `sendmail.cf`. Para alterá-lo, não se deve usar o arquivo mailers.linux e sim a seguinte definição no arquivo `sendmail.m4`:

```
dnl --- (em sendmail.m4) ---
define(LOCAL_MAILER_DEF, DELIVER)dnl    # servidor de entrega local
```

Infelizmente, o arquivo `Sendmail.mc` assume que o programa `deliver` está instalado no `/bin`, o que não é o caso, por exemplo, do Slackware 1.1.1 (o qual instala o programa em `/usr/bin`). Neste caso é necessário definir uma ligação simbólica ou reconstruir o software a partir dos fontes, alterando a sua localização para `/bin`.

Lidando Com Mensagens Devolvidas

```
define(POSTMASTERBOUNCE)dnl     # postmaster recebe as devoluções
```

Muitos administradores crêem ser importante garantir que uma mensagem seja enviada e recebida com uma taxa próxima a 100.

Ao se definir `POSTMASTERBOUNCE`, uma cópia de cada mensagem devolvida será enviada para a pessoa definida como `Postmaster` do sistema.

Infelizmente, a configuração deste parâmetro resulta no envio do *contéudo* da mensagem para o Postmaster, o que potencialmente reduz os aspectos de privacidade das pessoas que utilizam o serviço de correio.

Administradores de correio devem em geral tentar disciplinar-se (ou utilizar algumas ferramentas como programas interpretadores que retirem a parte de texto das mensagens) a fim de evitarem a leitura de mensagens não endereçadas para eles.

Itens Relacionados Com o Servidor de Nomes de Domínio

```
define(PSEUDODOMAINS, BITNET UUCP)dnl      # não tente DNS nesta opção
```

Há diversas redes bem conhecidas que são comumente referenciadas nos endereços das mensagens por razões históricas, mas que não são válidas para os propósitos de DNS. Ao se definir `PSEUDODOMAINS` previne-se que uma pesquisa DNS seja tentada, o que, diga-se de passagem, falhará sempre.

Definindo Nomes Para o Sistema Local

```
define(PSEUDONYMS, aracaju.cvirtual.com.br aracaju.UUCP cvirtual.com.br)
dnl                           # nomes do sistema local
define(DEFAULT_HOST, aracaju.cvirtual.com.br)
dnl                           # nome primário para uso em mensagens
```

Freqüentemente alguns sistemas desejam omitir a sua real identidade, servindo como um caminho padrão de mensagens ou processando mensagens endereçadas para nomes "antigos" pelos quais eles eram normalmente conhecidos.

`PSEUDONYMS` especifica a lista de todos os nomes de máquinas para os quais ele aceitará mensagens.

`DEFAULT_HOST` especifica o nome da máquina que irá aparecer nas mensagens originadas no sistema local. É importante que este parâmetro esteja configurado para uma valor válido ou todas as mensagens de retorno não poderão ser entregues.

Itens Relacionados Com UUCP

```
define(UUCPNAME, aracaju)dnl                # nome UUCP
define(UUCPNODES, |uuname|sort|uniq)dnl     # vizinhos UUCP
define(BANGIMPLIESUUCP)dnl                  # define o formato UUCP
define(BANGONLYUUCP)dnl                     # msgs são tratadas corretamente
```

Freqüentemente, sistemas são conhecidos por um nome para uso com o DNS e por outro nome para uso com UUCP. `UUCPNAME` permite que se defina um nome de máquina diferente daquele que aparece nos cabeçalhos em mensagens a serem enviadas via UUCP.

`UUCPNODES` define o comando que fornece a lista de nomes de máquinas que estão conectadas diretamente ao sistema local via UUCP.

`BANGIMPLIESUUCP` e `BANGONLYUUCP` garantem que o endereço de email com a sintaxe no formato UUCP (com separadores bang (!)) seja tratado de acordo com o comportamento UUCP ao invés do comportamento do Serviço de Nomes de Domínios usado na Internet.

Sistemas Retransmissores e Servidores de Mensagens

```
define(RELAY_HOST, parintins)dnl          # nossa máquina de retransmissão
define(RELAY_MAILER, UUCP-A)dnl           # atinge-se parintins via UUCP
```

Muitos administradores de sistemas não desejam preocupar-se com o trabalho necessário para garantir que o sistema local seja capaz de atingir todas as redes (e por conseqüência os sistemas) do mundo. Ao invés de fazer isso, pode ser preferível retransmitir todas as mensagens para outro sistema que é conhecido como "inteligente".

`RELAY_HOST` define o nome UUCP da máquina "inteligente" vizinha.

`RELAY_MAILER` define o servidor de mensagens usado como retransmissor de mensagens.

É importante notar que a configuração destes parâmetros resultará no reenvio das mensagens de saída para este sistema remoto, o que afetará a carga de processamento naquela máquina. Esteja certo de ter a concordância do Portmaster antes de configurar o sistema local para o uso de outro sistema com o propósito geral de retransmissão de mensagens.

As Várias Tabelas de Configuração

```
define(ALIASES, LIBDIR/aliases)dnl          # nomes alternativos
define(DOMAINTABLE, LIBDIR/domaintable)dnl  # máquinas do domínio
define(PATHTABLE, LIBDIR/pathtable)dnl      # base de dados de caminhos
define(GENERICFROM, LIBDIR/generics)dnl     # aspectos genéricos
define(MAILERTABLE, LIBDIR/mailertable)dnl  # serv.msg por máquina ou domínio
define(UUCPXTABLE, LIBDIR/uucpxtable)dnl    # caminhos p/as máquinas
define(UUCPRELAYS, LIBDIR/uucprelays)dnl    # caminhos otimizados
```

Através destas macros, pode-se alterar o local onde o sendmail+IDA procura as diversas tabelas do sistema. É recomendado deixá-las sob o caminho definido em `LIBDIR`.

O Arquivo `Sendmail.mc` Mestre

```
include(Sendmail.mc)dnl                     # ENTRADA OBRIGATÓRIA
```

Os autores do sendmail+IDA disponibilizam um arquivo `Sendmail.mc`, o qual contém os itens necessários à criação do arquivo sendmail.cf. Periodicamente, novas versões são liberadas para a correção de problemas e adição de novas funcionalidades sem requerer a instalação de uma nova versão completa ou da recompilação do sendmail a partir dos fontes. Este arquivo *não* deve ser editado.

Quais Entradas São Realmente Necessárias?

Ao não se utilizar qualquer arquivo de tabela opcional, o sendmail+IDA entregará as mensagens via `DEFAULT_MAILER` (e possivelmente `RELAY_HOST` e `RELAY_MAILER`) definidos no arquivo `sendmail.m4` usado para gerar o arquivo `sendmail.cf`. É possível alterar este comportamento facilmente através de entradas no arquivos `domaintable` ou `uucpxtable`.

Um site genérico que está sob a Internet e utilize DNS, ou um que utilize somente UUCP e reenvie todas as suas mensagens através de uma máquina de roteamento otimizado definida em `RELAY_HOST`, provavelmente não necessitará que todas as entradas da tabela sejam especificadas.

Praticamente todos os sistemas devem possuir configuradas as macros `DEFAULT_HOST` e `PSEUDONYMS`, as quais definem o nome canônico do site e nomes alternativos pelos quais ele é conhecido, e `DEFAULT_MAILER`. Caso tudo o que se tenha seja uma máquina de reenvio e um servidor de reenvio de mensagens, não será necessário sequer configurar estes valores uma vez que ele funcionarão "automagicamente".

Máquinas UUCP provavelmente necessitarão ainda configurar as variáveis `UUCPNAME` para seu nome UUCP oficial, `RELAY_MAILER` e `RELAY_HOST`, as quais habilitam o uso da máquina de roteamento otimizado para o reenvio de mensagens. O meio de transporte de mensagens a ser usado é definido em `RELAY_MAILER` e pode ser usualmente encontrado em `UUCP-A` para sites UUCP.

Caso o site utilize somente SMTP e use DNS, deve-se alterar `DEFAULT_MAILER` para `TCP-A` e provavelmente remover as linhas `RELAY_MAILER` e `RELAY_HOST`.

15.4 Um Tour Pelas Tabelas Sendmail+IDA

Sendmail+IDA disponibilizam um conjunto de tabelas que permitem alterar o comportamento padrão do sendmail (definidos no arquivo `sendmail.m4`) e definir comportamentos especiais em situações específicas, sistemas remotos e redes. Estas tabelas são pós-processadas com o programa `dbm` usando o utilitário Makefile disponibilizado com a distribuição.

Muitos sites necessitarão de poucas ou somente alguma destas tabelas. Caso o site não as requeira, a maneira mais simples é torná-las arquivos de tamanho zero (com o comando `touch`) e usar o utilitário Makefile padrão, localizado em `LIBDIR` ao invés de tentar editar o Makefile por conta própria.

15.4.1 `mailertable`

O arquivo `mailertable` define os tratamentos especiais para máquinas ou domínios específicos, baseados no nome de rede ou de máquina remota. É freqüentemente usado em sites Internet para selecionar um servidor de reenvio de mensagens intermediário ou caminho padrão para se alcançar um site remoto e para se especificar um protocolo em particular (UUCP ou SMTP) a ser usado. Sites UUCP geralmente não necessitam deste arquivo.

A ordem é importante. Sendmail lê o arquivo de cima para baixo e processa as mensagens de acordo com a primeira regra que atenda às especificações da mensagens. Desta forma, é aconselhável colocar as regras mais explícitas antes daquelas mais genéricas.

Suponhamos que se queira reenviar as mensagens do Departamento de Computação da Universidade do Pantanal via UUCP para uma máquina de reenvio denominada `tuiuiu`. Para tanto, deve-se ter uma entrada no arquivo `mailertable` com a seguinte aparência:

```
# (em mailertable)
#
# enviar todas as mensagens do domínio .dc.pantanal.edu.br via UUCP para tuiuiu
UUCP-A,tuiuiu         .dc.pantanal.edu.br
```

Supondo-se que queiramos que todo o domínio `pantanal.edu.br` envie suas mensagens através de uma máquina chamada `jacare`, a qual fará a resolução de endereços e a entrega. O arquivo mailertable expandido terá então o seguinte formato:

```
# (em mailertable)
#
# enviar todas as mensagens do domínio .dc.pantanal.edu.br via UUCP para tuiuiu
UUCP-A,tuiuiu      .dc.pantanal.edu.br
#
# enviar todas as mensagens do domínio pantanal.edu.br via UUCP para jacare
UUCP-A,jacare      .pantanal.edu.br
```

Como mencionado anteriormente a ordem é importante. Caso se altere a ordem acima por exemplo, isso fará com que todas as mensagens enviadas para `.dc.pantanal.edu.br` sigam através da máquina `jacare` ao invés de `tuiuiu` que é o realmente desejado.

```
# (em mailertable)
#
# enviar todas as mensagens do domínio pantanal.edu.br via UUCP para jacare
UUCP-A,jacare      .pantanal.edu.br
#
# (é impossível utilizar estas linhas porque a regra acima atende também a
#  todos os endereços da regra abaixo)
UUCP-A,tuiuiu       .dc.pantanal.edu.br
#
```

Nos exemplos acima do arquivo mailertable, o parâmetro `UUCP-A` faz com que o `sendmail` utilize a entrega UUCP com cabeçalhos contendo as informações de domínio.

A vírgula entre o meio de envio e o nome do sistema remoto indica que as mensagens devem ser reenviadas para `tuiuiu` para a resolução de endereços e entrega.

Entradas em Mailertable têm o seguinte formato:

meio_de_entrega delimitador máquina_de_reenvio máquina_ou_domínio

Há diversos meios de entrega possíveis. As diferenças residem basicamente em como os endereços são tratados. Tipicamente têm o valor igual a `TCP-A` (TCP/IP com endereço no estilo Internet), `TCP-U` (TCP/IP com endereço no estilo UUCP) e `UUCP-A` (UUCP com endereço no estilo Internet).

O caractere que separa o meio de entrega do nome da máquina de reenvio no lado esquerdo da linha define como os endereços são modificados por mailertable. Deve-se atentar que somente o envelope é reescrito (para se enviar a mensagem para o sistema remoto). Reescrever qualquer outro dado que não o envelope é geralmente contra indicado, pois pode trazer problemas de configuração da mensagem.

! Um ponto de exclamação retira o nome de máquina do destinatário antes de reenviar a mensagem. Isso pode ser usado quando se deseja forçar o envio de mensagens para um site com problemas de configuração.

, Uma vírgula não faz qualquer mudança no endereço. A mensagem será somente reenviada através do meio especificado para o servidor de mensagens indicado.

: Dois pontos removem o nome da máquina de destino somente se houver máquinas intermediárias entre a origem e o destino. Por exemplo, `aracaju!maceio!sonia` terá `aracaju` removida, enquanto `juizdefora!clarissa` permanecerá inalterada.

15.4.2 `uucpxtable`

Normalmente mensagens para máquinas com nomes de domínios totalmente qualificados são remetidas através do estilo Internet (SMTP) utilizando-se DNS, ou via uma máquina de reenvio ou distribuição de mensagens. O arquivo `uucpxtable` força a entrega via roteamento UUCP convertendo o nome em formato de domínio para o estilo UUCP de nome de máquinas remotas.

É freqüentemente usado quando a máquina local executa tarefas de reenvio de mensagens para um site ou domínio ou quando se deseja enviar mensagens através de uma conexão UUCP confiável, ao invés de se utilizar múltiplas máquinas através do modo de entrega padrão ou quaisquer sistemas ou redes intermediárias.

Neste caso, sites UUCP necessitam informar aos seus vizinhos UUCP que usam cabeçalhos de mensagens com nomes de domínios, que eles podem utilizar este arquivo para forçar a entrega através de uma conexão ponto a ponto UUCP ao invés de usarem a rota `RELAY_MAILER` e `RELAY_HOST` ou através do `DEFAULT_MAILER`.

Sites Internet que não utilizam UUCP provavelmente não necessitarão do `uucpxtable`.

Supondo-se que uma máquina disponibilize serviços de reenvio de mensagens para um sistema chamado `campinas.com.br` com DNS e denominado `campinas` nos mapas UUCP. Será necessária então a seguinte entrada no arquivo `uucpxtable` para forçar que as mensagens sejam enviadas através de conexão direta UUCP.

```
#============= /usr/local/lib/mail/uucpxtable ===========
# Mensagens enviadas para delga@campinas.com.br serão reescritas
# como campinas!delga e após serão enviados via UUCP
#
campinas    campinas.com.br
#
#-----------------------------------------------------------
```

15.4.3 pathtable

O arquivo `pathtable` é usado para definir explicitamente o roteamento para máquinas remotas ou redes. O arquivo `pathtable` tem uma sintaxe de estilo "definição de caminhos", ordenados alfabeticamente. Os dois campos em cada linha devem ser separados por um caractere de tabulação, ou o programa `dbm` emitirá uma mensagem de erro.

Muitos sistemas não necessitarão de qualquer entrada no arquivo `pathtable`.

```
#=============== /usr/local/lib/mail/pathtable ================
#
# este é uma arquivo no estilo de nomes alternativos de caminhos,
# para entrega rápida de mensagens de vizinhos UUCP por um caminho
# direto UUCP.
#
# é obrigatório o uso de tabulações entre os campos
#
# roteamento de mensagens através de um ou mais sites intermediários para
# um sistema remoto utilizando o endereçamento no formato UUCP
#
campinas!vinhedo!%s            vinhedo
#
# reenviando para um sistema que é um vizinho UUCP de um site Internet que
# pode ser alcançado.
#
piranha!%s@dc.pantanal.edu.br    piranha
#
# As linhas seguintes enviam todas as mensagens para duas redes através de
# diferentes caminhos (veja o '.' inicial).
# Neste exemplo, "portauucp" e "portacorreio" são sistemas que servem
# especificamente com pontos de passagem para os pseudo domínios .UUCP e .BITNET
#
%s@portauucp.pantanal.edu.br              .UUCP
portacorreio!%s@correio.bitnet.com.br      .BITNET
#
#================== final do pathtable ======================
```

15.4.4 domaintable

O arquivo `domaintable` é usado geralmente para forçar certos comportamentos após uma pesquisa DNS. Ele permite que o administrador torne certos nomes resumidos disponíveis como uma referência a sistemas ou domínios comumente acessados. Ele pode ser usado ainda para corrigir nomes de máquinas e domínios mal informados.

Muitos sites não necessitarão de entradas no arquivo `domaintable`.

Os seguintes exemplos mostram como substituir um endereço incorreto por um endereço mais adequado:

```
#============= /usr/local/lib/mail/domaintable =================
#
#
maquina1.dominio.correto         maquina1.dominio.errado
#
#
#================== end of domaintable =======================
```

15.4.5 aliases

Nomes alternativos permitem a implementação de uma série de funcionalidades:

- Disponibilizar um nome curto ou atalho ou um nome mais simples para o destinatário de mensagens, permitindo que estas possam ser enviadas para uma ou mais pessoas.
- Acionar um programa com a mensagem como entrada para o processamento.
- Gravar a mensagem em um arquivo.

Todos os sistemas requerem nomes alternativos para os usuários `Postmaster` e `MAILER-DAEMON` para se tornarem compatíveis com as RFCs.

Esteja sempre atento à segurança ao se definir nomes alternativos que acionem programas ou gerem entradas para estes, uma vez que o sendmail geralmente é executado com identificação de superusuário[2].

Mudanças no arquivo de `aliases` não têm efeito até que o comando

```
# /usr/lib/sendmail -bi
```

seja executado para a construção das tabelas dbm. Isso pode ser feito através da execução do comando `newaliases`, normalmente a partir do cron.

Detalhes sobre nomes alternativos podem ser encontrados na página de manual do comando `aliases(5)`.

[2] setuid root

```
#-------------------- /usr/local/lib/mail/aliases ------------------
#
# demonstra tipos comuns de nomes alternativos
#
usenet:         janete                    # apelido para uma pessoa
adm:            japa,gafa                 # apelido para diversas pessoas
lista-funci:    :include:/usr/lib/listas/funcionarios
                                          # lê os destinatários de um arquivo
formata_msg:    | /usr/local/lib/form     # apelido que aciona um programa
pedidos:        /var/log/pedidos          # apelido que grava a msg. em arquivo
#
# Os dois apelidos a seguir estão presentes para compatibilidade com RFC
# É importante tê-los direcionados para alguém que utilize o correio
# rotineiramente
#
postmaster:     root                      # informação obrigatória
MAILER-DAEMON:  postmaster                # informação obrigatória
#
#-------------------------------------------------------------------
```

15.4.6 Tabelas Raramente Utilizadas

As seguintes tabelas estão disponíveis, mas são raramente utilizadas. Para maiores detalhes, por favor consulte a documentação que acompanha o sendmail+IDA.

`uucprelays` O arquivo `uucprelays` é usado como atalho para caminhos UUCP bem conhecidos ao invés de se utilizar diversos pontos de reenvio não confiáveis gerados pelo processamento de mapas UUCP através do programa `pathalias`.

`genericfrom` e `xaliases` O arquivo `genericfrom` esconde o nome de usuários locais e endereços para usuários externos através da conversão automática de nomes de usuários locais para endereços genéricos que não coincidem com nomes internos de usuários.

O utilitário `xalparse` automatiza a geração dos arquivos genericfrom e aliases, de forma que ambas as conversões de entrada e saída de mensagens ocorram a partir de um arquivo mestre xaliases.

`decnetxtable` O arquivo `decnetxtable` reescreve endereços no estilo de nome com domínios no formato DECnet, de forma similar a que a tabela domaintable é utilizada na conversão de nomes para o estilo de domínios SMTP de endereços.

15.5 Instalando o `sendmail`

Nessa seção, daremos uma visão geral de como instalar uma típica distribuição em formato binário do sendmail+IDA e verificaremos os aspectos necessários para torná-la funcional.

A versão atual da distribuição do sendmail+IDA para `Linux` pode ser obtida em `metalab.unc.edu` no caminho `/pub/Linux/system/Mail`. Mesmo que já se tenha uma versão mais antiga do `sendmail` recomendamos fortemente que seja usada uma versão atualizada, uma vez que as atualizações são padronizadas para o Linux e diversas questões de segurança e novas funcionalidades são disponibilizadas a cada versão.

Caso se esteja construindo o `sendmail` a partir dos fontes, deve-se seguir as instruções contidas nos arquivos `README` incluídos na distribuição. A versão atual de sendmail+IDA pode ser encontrada em `vixen.cso.uiuc.edu`. Para construir-se o sendmail+IDA no `Linux`, são necessários ainda os arquivos de configuração específicos do `Linux` a partir do pacote `newspak-2.2.tar.gz`, o qual está disponível em `metalab.unc.edu` no caminho `/pub/Linux/system/Mail`.

Caso se tenha instalado anteriormente uma versão do `smail` ou outro agente de transporte de mensagens, provavelmente será necessário remover ou renomear todos os arquivos do smail como medida de segurança.

15.5.1 Extraindo a Distribuição Binária

Inicialmente deve-se desempacotar o arquivo maior que contém todos os arquivos necessários em alguma localização segura:

```
$ gunzip -c sendmail5.65b+IDA1.5+mailx5.3b.tgz | tar xvf -
```

Caso se tenha uma versão atualizada do programa `tar`, por exemplo, uma gerada a partir de uma versão recente do Conectiva Linux, pode-se simplesmente executar o comando `tar -zxvf` *`filename`*`.tgz` e se obterá o mesmo resultado.

Ao desempacotar o arquivo é criado o diretório chamado `sendmail5.65b+IDA1.5+mailx5.3b`. Neste diretório, pode-se encontrar uma instalação completa de sendmail+IDA mais um binário do agente de usuário `mailx`. Toda a árvore sob este diretório reflete as localizações onde os arquivos devem ser instalados, sendo indicado utilizar o comando `tar` para movê-los para a sua localização.

```
# cd sendmail5.65b+IDA1.5+mailx5.3b
# tar cf - . | (cd /; tar xvvpoof -)
```

15.5.2 Construindo sendmail.cf

Para construir um arquivo `sendmail.cf` customizado para o site local, deve-se atualizar o arquivo `sendmail.m4` e processá-lo com o comando `m4`. No diretório `/usr/local/lib/mail/CF`, pode-se encontrar um arquivo de exemplo chamado `sample.m4`. Pode-se copiá-lo para *nome_da_máquina*`.m4` e editá-lo para refletir a situação do site local.

O arquivo de exemplo é configurado para uso por um site UUCP com cabeçalho no estilo nome de domínio e se comunica com uma máquina que faz o roteamento de mensagens. Sites como este necessitam somente que sejam alterados poucos itens.

Na versão atual, forneceremos apenas uma breve visão geral das macros que devem ser alteradas. Para uma descrição completa da forma que deve ser utilizada, por favor referencie-se ao aqui descrito sobre o arquivo `sendmail.m4`.

`LOCAL_MAILER_DEF` Define o arquivo que indica as formas de entrega do correio local. Veja a seção "Definindo o Servidor de Correio Local" para verificar a configuração deste parâmetro.

`PSEUDONYMS` Define todos os nomes pelos quais o sistema local é conhecido.

`DEFAULT_HOST` Coloca o nome totalmente qualificado da máquina em todas as mensagens enviadas a partir do sistema local.

`UUCPNAME` Inclui o nome não qualificado da máquina.

`RELAY_HOST` e `RELAY_MAILER` Caso se use UUCP via uma máquina com roteamento otimizado, `RELAY_HOST` deve ser configurado para o nome UUCP daquela máquina. Deve-se usar UUCP-A caso se deseje cabeçalhos no estilo nomes de domínios.

`DEFAULT_MAILER` Caso se esteja na Internet e se deseje utilizar DNS, deve-se configurar este parâmetro para `TCP-A`. Isso indica ao sendmail para usar o modo de transporte `TCP-A`, o qual entrega mensagens via SMTP utilizando o estilo de endereçamento compatível com as RFCs. Sites Internet provavelmente não necessitarão da definição das variáveis `RELAY_HOST` ou `RELAY_MAILER`.

Para criar um arquivo `sendmail.cf`, deve-se executar o seguinte comando:

```
# make nome_da_máquina.cf
```

Isto processa o arquivo *nome_da_máquina*`.m4` e cria um arquivo *nome_da_máquina*`.cf` a partir dele.

Após, deve-se testar se o arquivo de configuração criado atende às necessidades. Isso é explicado nas próximas seções.

Uma vez que se esteja satisfeito com seu comportamento, ele pode ser copiado com o comando:

```
# cp  nome_da_máquina.cf /etc/sendmail.cf
```

Neste ponto, o sistema sendmail está pronto para execução. Deve-se colocar a seguinte linha no arquivo de inicialização adequado (geralmente `/etc/rc.inet2` para o Slackware[3]). Pode-se ainda executá-lo manualmente para se iniciar o processo imediatamente.

```
# /usr/lib/sendmail -bd -q1h
```

15.5.3 Testando o Arquivo `sendmail.cf`

Para se colocar o sendmail no modo de teste, deve-se acioná-lo com a opção `-bt`. O arquivo de configuração padrão é o arquivo sendmail.cf instalado no sistema. Pode-se testar arquivos alternativos usando-se a opção `-C`*`filename`*.

Nos exemplos seguintes, testamos `aracaju.cf`, o arquivo de configuração gerado a partir do arquivo `aracaju.m4` que foi mostrado no extrato do arquivo `sendmail.m4`.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
>
```

[3]no Conectiva Linux esta inicialização dá-se no arquivo `/etc/rc.d/init.d/sendmail`

Os seguintes testes asseguram que o programa `sendmail` é capaz de entregar mensagens para os usuários do sistema local. Em todos os casos o resultado dos testes deve ser o mesmo, apontando para o nome do sistema local com o meio de entrega `LOCAL`.

Inicialmente deve-se testar como uma mensagem para um usuário local é entregue.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 me
rewrite: ruleset  3   input: me
rewrite: ruleset  7   input: me
rewrite: ruleset  9   input: me
rewrite: ruleset  9 returns: < me >
rewrite: ruleset  7 returns: < > , me
rewrite: ruleset  3 returns: < > , me
rewrite: ruleset  0   input: < > , me
rewrite: ruleset  8   input: < > , me
rewrite: ruleset 20   input: < > , me
rewrite: ruleset 20 returns: < > , @ aracaju . cvirtual . com . br , me
rewrite: ruleset  8 returns: < > , @ aracaju . cvirtual . com . br , me
rewrite: ruleset 26   input: < > , @ aracaju . cvirtual . com . br , me
rewrite: ruleset 26 returns: $# LOCAL $@ aracaju . cvirtual . com . br $: me
rewrite: ruleset  0 returns: $# LOCAL $@ aracaju . cvirtual . com . br $: me
```

A saída acima mostra como o programa `sendmail` processa endereços internamente. Ele é administrado através de diversas regras que o analisam, invocam outras regras e o dividem em diversos componentes.

No nosso exemplo, enviamos o endereço `me` para as regras 3 e 0 (este é o significado do termo `3,0` informados antes do endereço). A última linha mostra o endereço retornado pela regra 0, contendo onde o meio de transporte deverá entregar a mensagem, além do nome da máquina e do nome do usuário de destino.

A seguir, deve-se testar uma mensagem no sistema com a sintaxe UUCP.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 aracaju!me
rewrite: ruleset  3   input: aracaju ! me
[...]
rewrite: ruleset  0 returns: $# LOCAL $@ aracaju . cvirtual . com . br  $: me
>
```

A seguir, deve-se testar um endereço de um usuário com a sintaxe Internet destinado a um nome de máquina totalmente qualificada.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 me@aracaju.cvirtual.com.br
rewrite: ruleset  3   input: me @ aracaju . cvirtual . com . br
[...]
rewrite: ruleset  0 returns: $# LOCAL $@ aracaju . cvirtual . com . br $: me
>
```

Deve-se repetir os dois testes acima para cada um dos nomes especificados nos parâmetros `PSEUDONYMS` e `DEFAULT_NAME` no arquivo `sendmail.m4` local.

Finalmente deve-se testar o envio de mensagens através da máquina de roteamento e reenvio.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 fred@parintins.com.br
rewrite: ruleset  3   input: fred @ parintins.com.br
rewrite: ruleset  7   input: fred @ parintins.com.br
rewrite: ruleset  9   input: fred @ parintins.com.br
rewrite: ruleset  9 returns: < fred > @ parintins.com.br
rewrite: ruleset  7 returns: < @ parintins.com.br>,fred
rewrite: ruleset  3 returns: < @ parintins.com.br>,fred
rewrite: ruleset  0   input: < @ parintins.com.br>,fred
rewrite: ruleset  8   input: < @ parintins.com.br>,fred
rewrite: ruleset  8 returns: < @ parintins.com.br>,fred
rewrite: ruleset 29   input: < @ parintins.com.br>,fred
rewrite: ruleset 29 returns: < @ parintins.com.br>,fred
rewrite: ruleset 26   input: < @ parintins.com.br>,fred
rewrite: ruleset 25   input: < @ parintins.com.br>,fred
rewrite: ruleset 25 returns: < @ parintins.com.br>,fred
rewrite: ruleset  4   input: < @ parintins.com.br>,fred
rewrite: ruleset  4 returns: fred @ parintins.com.br
rewrite: ruleset 26 returns: < @ parintins.com.br>,fred
```

```
rewrite: ruleset  0 returns: $# UUCP-A $@ parintins $:\
                             < @ parintins.com.br>,fred
>
```

15.5.4 Integrando Todos os Componentes - Testando o Arquivo sendmail.cf e as Tabelas

Neste ponto, já foi possível verificar que as mensagens têm o comportamento padrão desejado e que será possível enviar e receber mensagens com endereços válidos. Para completar a instalação, deve-se criar as tabelas dbm apropriadas para se obter os resultados finais desejados.

Após a criação das tabelas requeridas pelo sistema, deve-se processá-las através do programa `dbm` acionando-se `make` no diretório onde elas estejam localizadas.

Caso o site utilize somente UUCP, *não* será necessário criar qualquer uma das tabelas mencionadas no arquivo `README.linux`. Deve-se somente utilizar o comando touch nos arquivo que trabalham com o Makefile.

Caso o site utilize somente UUCP, porém estabelece conexões com outros sites além da máquina de roteamento otimizado, será necessário incluir as entradas no arquivo `uucpxtable` para cada um deles (ou as mensagens a eles endereçadas serão transferidas via máquina de roteamento otimizado) e deve-se executar ainda o programa `dbm`.

Inicialmente, deve-se estar seguro de que as mensagens através do `RELAY_HOST` são enviadas através do `RELAY_MAILER`.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 fred@campinas.com.br
rewrite: ruleset  3   input: fred @ campinas.com.br
rewrite: ruleset  7   input: fred @ campinas.com.br
rewrite: ruleset  9   input: fred @ campinas.com.br
rewrite: ruleset  9 returns: < fred > @ campinas.com.br
rewrite: ruleset  7 returns: < @ campinas.com.br>,fred
rewrite: ruleset  3 returns: < @ campinas.com.br>,fred
```

```
rewrite: ruleset  0   input: < @ campinas.com.br>,fred
rewrite: ruleset  8   input: < @ campinas.com.br>,fred
rewrite: ruleset  8 returns: < @ campinas.com.br>,fred
rewrite: ruleset 29   input: < @ campinas.com.br>,fred
rewrite: ruleset 29 returns: < @ campinas.com.br>,fred
rewrite: ruleset 26   input: < @ campinas.com.br>,fred
rewrite: ruleset 25   input: < @ campinas.com.br>,fred
rewrite: ruleset 25 returns: < @ campinas.com.br>,fred
rewrite: ruleset  4   input: < @ campinas.com.br>,fred
rewrite: ruleset  4 returns: fred @ campinas.com.br
rewrite: ruleset 26 returns: < @ campinas.com.br>,fred
rewrite: ruleset  0 returns: $# UUCP-A $@ parintins $:\
                              < @ campinas.com.br>,fred
>
```

Caso se tenham outros vizinhos UUCP além do `RELAY_HOST`, deve-se assegurar que as mensagens para eles têm um comportamento adequado. Mensagens endereçadas em uma sintaxe de estilo UUCP devem ser remetidas diretamente para estes (a menos que isso seja explicitamente evitado através de uma entrada no arquivo `domaintable`). Assumindo-se que a máquina `vinhedo` é um vizinho direto da máquina local. Ao enviar uma mensagem para `vinhedo!fred`, esta deve produzir o seguinte efeito:

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 vinhedo!fred
rewrite: ruleset  3   input: vinhedo ! fred
[...linhas omitidas...]
rewrite: ruleset  0 returns: $# UUCP $@ vinhedo $: < > , fred
>
```

Caso tenham sido definidas entradas no arquivo `uucpxtable` que forcem a entrega para vizinhos no formato UUCP que por sua vez enviam suas mensagens com cabeçalhos no formato Internet, isso necessariamente deve ser testado.

```
# /usr/lib/sendmail -bt -Caracaju.cf
ADDRESS TEST MODE
```

```
Enter <ruleset> <address>
[Note: No initial ruleset 3 call]
> 3,0 dude@vinhedo.2birds.com.br
rewrite: ruleset  3   input: dude @ vinhedo . campinas . com . br
[...lines omitted...]
rewrite: ruleset  0 returns: $# UUCP $@ vinhedo . campinas $: < > , dude
>
```

15.6 Dicas de Administração e Outros Detalhes

Agora que discutimos a configuração, instalação e testamos o sistema sendmail+IDA, dedicaremos alguns momentos para observar coisas que *ocorrem* rotineiramente na vida de um administrador de correio.

Sistemas remotos algumas vezes ficam inoperantes, modems e linhas falham, definições DNS podem ser mal feitas devido a erros humanos, redes param de funcionar repentinamente, etc. Nestes casos, administradores de correio necessitam saber como agir corretamente, de forma rápida e segura, para que o fluxo de mensagens possa ser mantido corretamente, através de rotas alternativas até os sistemas remotos ou no restabelecimento dos serviços de envio de mensagens.

O restante deste capítulo busca prover as soluções para os problemas mais comuns encontrados em "emergências com correio eletrônico".

15.6.1 Reenviando Mensagens Para Um Servidor

Para enviar mensagens destinadas a uma máquina em particular ou a um domínio específico através de um sistema de reenvio predeterminado, geralmente é usado o arquivo `mailertable`.

Por exemplo, para reenviar uma mensagem para `amazonia.org.br` que tem um caminho padrão UUCP chamado `amazonas`, deve-se incluir a seguinte entrada no arquivo `mailertable`:

```
UUCP-A,amazonas   amazonia.org.br
```

15.6.2 Forçando Mensagens em um Site Mal Configurado

Freqüentemente, máquinas Internet terão problemas obtendo mensagens em sistemas remotos mal configurados. Há diversas variações deste problema, mas o sintoma usual é o retorno de mensagens pelo sistema remoto ou o fato delas nunca chegarem ao destino.

Estes problemas podem colocar o administrador do sistema local em uma posição difícil, uma vez que os usuários normalmente não se importam com o fato do administrador local não intervir em todos os sistemas de mensagens do mundo (ou não entendam como não se fez com que o administrador remoto corrigisse o problema). Eles apenas sabem que suas mensagens não chegaram ao destino desejado e o administrador de correio é a pessoa indicada para receber as reclamações.

Uma configuração de um site remoto é problema do administrador, não deles. Em todos os casos, esteja seguro de *não* gerar problemas de configuração ao comunicar-se com um site remoto mal configurado. Caso não se possa entrar em contato com o Postmaster no sistema remoto para que ele resolva o problema em um tempo razoável, tem-se duas opções.

- Geralmente é possível forçar o recebimento das mensagens por um sistema remoto, mesmo que aquele sistema esteja mal configurado, porém as respostas na ponta remota podem não funcionar... mas isso é problema do administrador do sistema remoto.

  Pode-se corrigir cabeçalhos incorretos no envelope de mensagens enviadas utilizando uma entrada no arquivo `domaintable` para as máquinas ou domínios que estejam apresentando problemas, com o seguinte formato:

  ```
  maquina1.dominio.correto.com.br        maquina1.dominio.incorreto.com.br
  ```

- Freqüentemente, sites mal configurados devolvem a mensagem enviada para o remetente e efetivamente informam "esta mensagem não é para este site", uma vez que não tenham o arquivo `PSEUDONYMNS` ou equivalente adequadamente configurado. Porém é possível retirar completamente o nome da máquina ou as informações de domínio do cabeçalho das mensagens enviadas.

  O caractere `!` no arquivo `mailertable` faz com que as mensagens sejam entregues em um site remoto como se elas estivessem sendo geradas no sistema local. Note que estas mudanças ocorrem somente no cabeçalho das mensagens, fazendo com que o endereço de retorno seja apresentado corretamente.

```
TCP!maquina1.dominio.correto.com.br    maquina1.dominio.incorreto.com.br
```

Mesmo com este artifício, não há garantias que a mensagem será entregue pelo sistema remoto (lembre-se que eles podem estar com problemas), mas então seus usuários estarão pressionando seus administradores ....

15.6.3 Forçando a Transferência de Mensagens Via UUCP

Em um mundo ideal (sob a perspectiva da Internet), todas as máquinas têm registros dos Serviços de Nomes de Domínios (DNS) e enviarão mensagens com nomes de domínios totalmente qualificados.

Caso se utilize uma conexão UUCP com um site, pode-se forçar que a mensagem seja entregue através desta conexão ao invés de utilizar o caminho padrão de entrega de mensagens, basicamente retirando as informações de domínios do nome da máquina através de uma entrada no arquivo `uucpxtable`.

Para forçar uma entrega UUCP para a máquina `campinas.com.br`, pode-se colocar a seguinte entrada no arquivo `uucpxtable`:

```
# retira as informações do domínio campinas.com.br para forçar a entrega UUCP
campinas    campinas.com.br
```

Isso faz com que o sendmail determine (via `UUCPNODES` no arquivo `sendmail.m4`) que se está diretamente conectado ao sistema remoto e que a mensagem será enfileirada para entrega via UUCP.

15.6.4 Evitando Que Mensagens Sejam Enviadas Via UUCP

As condições opostas também podem ocorrer. Freqüentemente, sistemas podem ter algumas conexões diretas UUCP que são usadas eventualmente ou que não estejam sempre disponíveis para serem utilizadas como caminho padrão ou o servidor de retransmissão de mensagens.

Por exemplo, na área de Seattle existem alguns sites que trocam distribuições `Linux` via UUCP anônimo quando estas são liberadas. Estes sistemas comunicam-se via UUCP somente quando necessário, tornando mais rápido e confiável enviar mensagens através de servidores retransmissores normalmente utilizados (e sempre disponíveis).

É muito simples evitar a entrega de uma mensagem via UUCP para uma máquina à qual se esteja conectado diretamente. Caso o sistema remoto tenha um nome de domínio totalmente qualificado, pode-se adicionar a seguinte entrada no arquivo `domaintable`:

```
# evita a entrega de uma mensagem para um vizinho UUCP
caruaru.com.br        caruaru
```

Isso substituirá qualquer ocorrência de um nome UUCP pelo nome FQDN e evitará que o nome coincida com um nome definido em `UUCPNODES` no arquivo `sendmail.m4`. Desta forma as mensagens serão enviadas através do `RELAY_MAILER` e `RELAY_HOST` (ou `DEFAULT_MAILER`).

15.6.5 Filas de Mensagens Por Demanda

Para processar uma fila de mensagens imediatamente, basta digitar `/usr/lib/runq`[4]. Este comando aciona o sendmail com as opções adequadas para que ele verifique as filas de mensagens e as envie imediatamente ao invés de esperar a próxima execução agendada.

15.6.6 Relatórios de Estatísticas de Mensagens

Muitos administradores de sites (e as pessoas para as quais eles trabalham) estão interessadas no volume de mensagens que são tratadas, originadas ou recebidas pelo site local. Há algumas formas de verificar o tráfego de mensagens.

- Sendmail vem com um utilitário chamado `mailstats` o qual lê um arquivo chamado `/usr/local/lib/mail/sendmail.st` e reporta o número de mensagens e de bytes transferidos por cada uma das formas de transporte definidas no arquivo `sendmail.cf`. Este arquivo deve ser criado manualmente pelo administrador do sistema local para que o sendmail registre as mensagens. Os dados serão eliminados através da remoção e da recriação do arquivo `sendmail.st`. Uma forma de se fazer isto é a seguinte:

  ```
  # cp /dev/null /usr/lib/local/mail/sendmail.st
  ```

[4] no Conectiva Linux `/usr/sbin/sendmail -q`

- Provavelmente a melhor forma de criar relatórios de qualidade descrevendo quem utiliza o correio e qual o volume de mensagens para, de e através do sistema local é ligar o sistema de depuração com o programa `syslogd(8)`. Geralmente isso significa que o servidor `/etc/syslogd` deve ser acionado durante a inicialização do sistema (o que normalmente é feito de qualquer forma), adicionando-se uma linha ao arquivo `/etc/syslog.conf(5)` com a seguinte aparência:

  ```
  mail.debug                              /var/log/syslog.mail
  ```

  Caso se utilize o `mail.debug` e se tenha um volume de mensagens entre médio e alto, a saída do syslog poderá gerar arquivos muito grandes. Estes arquivos normalmente necessitam ser rotacionados ou eliminados através de uma rotina baseada no `crond(8)`.

  Há alguns utilitários que podem ser comumente utilizados e que podem criar informações resumidas a partir da saída do syslogd. Um dos mais conhecidos é o programa `syslog-stat.pl`, escrito em `perl`, o qual é distribuído com os fontes do sendmail+IDA.

15.7 Misturando Distribuições

Não há uma configuração realmente padrão dos agentes de entrega e transporte de correio eletrônico, assim como não há uma definição estrita da estrutura de diretórios.

Desta forma, é necessário garantir que todas as diversas peças do sistema (notícias USENET, correio, TCP/IP) precisam conhecer a localização do programa local de entrega de mensagens (`lmail`, `deliver`, etc.), programas remotos de entregas de mensagens (`rmail`) e o programa de transporte de mensagens (`sendmail` ou `smail`). Tais localizações são muitas vezes "assumidas" pelos programas e podem estar não muito bem documentadas, apesar do uso do comando `strings` poder auxiliar na determinação de quais arquivos e diretórios são esperados. A seguir apresentamos alguns problemas vistos no passado com algumas das distribuições binárias e fontes:

- Algumas versões da distribuição do NET-2 do TCP/IP têm serviços definidos para um programa chamado `umail` ao invés de `sendmail`.

- Há vários portes do programa `elm` e `mailx` que procuram por um agente de entrega chamado `/usr/bin/smail` ao invés de sendmail.
- Sendmail+IDA tem como agente de entrega predefinido o `deliver`, mas espera que ele esteja localizado no diretório `/bin` ao invés de `/usr/bin`, que é a localização mais habitual no Linux.

Ao invés de construir todos os clientes de mensagens a partir dos fontes, geralmente é mais aconselhável criar ligações simbólicas adequadas...

15.8 Onde Obter Mais Informações

Há muitos locais onde podem ser encontradas informações sobre o `sendmail`. Para uma lista veja o Como Fazer `Linux` MAIL postada regularmente em `comp.answers`; está também disponível para FTP anônimo em `rtfm.mit.edu`. De qualquer forma, o local definitivo são os fontes do sendmail+IDA. Verifique no diretório `ida/cf` e procure pelos arquivos `DBM-GUIDE`, `OPTIONS`e `Sendmail.mc`.

Capítulo 16

Notícias na Internet

16.1 História da Usenet

A idéia de uma rede de notícias nasceu em 1979 quando dois estudantes graduados, Tom Truscott e Jim Ellis, pensaram em utilizar UUCP para conectar diversas máquinas, com o propósito de trocarem informações entre usuários `Unix`. Eles configuraram uma pequena rede de três máquinas na Carolina do Norte.

Inicialmente, o tráfego foi administrado por alguns programas interpretados (posteriormente reescritos em C), mas nunca foram liberados para o uso público. Eles foram rapidamente substituídos pelo "A" news, a primeira versão pública de um programa de notícias.

"A" news foi desenvolvido para lidar com alguns poucos artigos por grupo por dia. Quando o volume continuou a crescer, ele foi reescrito por Mark Horton e Matt Glickman, que o chamaram de versão "B" (também conhecido como Bnews). A primeira versão pública do Bnews foi a 2.1 datada de 1982. Ela foi atualizada permanentemente desde então, com a adição de uma série de novas funcionalidades. A versão mais atual na época em que este guia foi escrito era a Bnews 2.11. Ela vem se tornando obsoleta, tendo o seu mantenedor oficial mudado para o programa INN.

Outra versão foi desenvolvida e liberada em 1987 por Geoff Collyer e Henry Spencer, chamada de "C" ou C News. Atualmente diversas atualizações foram realizadas no C News, a mais proeminente é conhecida como a Versão de Performance

C News. Em sites que contêm um grande número de grupos, a sobrecarga envolvida em acionamentos freqüentes do programa `relaynews`, o qual é responsável pelo despacho de artigos recebidos para outras máquinas, é significativa. A Versão de Performance adiciona uma opção ao `relaynews` que permite que ele seja executado em *modo servidor*, no qual o programa se auto coloca em segundo plano.

A Versão de Performance no C News está atualmente disponível na maioria das distribuições `Linux`.

Todas as notícias liberadas por "C" são destinadas primeiramente a redes UUCP, apesar de poderem ser utilizadas em outras redes. Transferências eficientes de notícias sobre redes TCP/IP, DECNet ou similares requerem um novo esquema. Esta foi a razão que levou à criação em 1986 do *Protocolo de Transferência de Notícias em Rede*[1], NNTP. Ele é baseado em conexões de rede e especifica alguns comandos de transferência interativa e recuperação de artigos.

Há algumas aplicações baseadas em NNTP disponíveis na Internet. Uma das quais é o pacote `nntpd` de Brian Barber e Phil Lapsley, o qual pode ser usado, entre outras coisas, para prover o serviço de leitura de notícias para uma série de máquinas dentro de uma rede local. `nntpd` foi desenhado para complementar pacotes de notícias como Bnews ou C News, proporcionando-lhes funcionalidades NNTP.

Um pacote diferente baseado em NNTP é o INN, ou *Notícias Internet*[2]. Não se trata somente de uma interface para um sistema de notícias, mas um sistema completo. Ele abrange um sofisticado servidor de transmissão de notícias, capaz de manter diversas conexões concorrentes NNTP de forma eficiente e é o servidor de notícias escolhido por muitos sites Internet.

16.2 O Que é Usenet?

Um dos mais interessantes fatos sobre Usenet é que ela não faz parte de qualquer organização, ou de qualquer gerenciamento centralizado. Na verdade, faz parte da filosofia da Usenet que exceto pela sua descrição técnica, não se possa definir o *que* ela seja, mas sim somente o que ela não é. Caso se tenha à disposição o livro de Brendan Kehoe chamado "Zen and the Art of the Internet", pode-se descobrir uma série de não propriedades da Usenet.

---

[1] Network News Transfer Protocol
[2] Internet News

Com o risco de soar um pouco óbvio, podemos definir a Usenet como a colaboração entre diversos sites que trocam notícias Usenet. Para ser um site Usenet, tudo o que se deve fazer é encontrar um outro site Usenet e fazer um acordo com seus administradores para a troca de mensagens com o site local. Disponibilizar notícias para outro site é denominada *alimentação*, criando um novo ditado utilizado na filosofia Usenet: "Obtenha alimento e você estará dentro".

A unidade básica das notícias Usenet é o artigo. Esta é a mensagem que um usuário escreve e posta para a Internet. Para habilitar o sistema de notícias para lidar com eles, eles devem conter algumas informações administrativas adicionais, conhecidas como cabeçalho do artigo. São muito similares ao formato de cabeçalho de mensagens do correio eletrônico descritas no padrão Internet RFC 822, que consiste de diversas linhas de texto, cada uma começando com um nome de campo terminado por dois pontos, seguidos pelo valor do campo.[3]

Artigos são submetidos a um ou mais *grupos de notícias*[4]. Pode-se considerar um grupo de notícias como um fórum para artigos relacionado a um tópico comum. Todos os grupos de notícias são organizados em uma estrutura hierárquica, na qual cada nome de grupo indica o seu lugar na hierarquia. Isso freqüentemente torna mais simples a visualização da temática do grupo. Por exemplo, qualquer um pode perceber que o grupo de notícias chamado `comp.os.linux.announce`[5] é usado para anúncios relacionados com um sistema operacional chamado `Linux`.

Os artigos são trocados entre todos os sites Usenet que desejam receber notícias sobre o grupo. Quando dois sites concordam em trocar notícias, eles estão livres para trocar informações de qualquer grupo de notícias que desejam e podem ainda ter a sua hierarquia própria de notícias. Por exemplo `pantanal.edu.br` pode ter uma conexão de notícias com `amazonas.edu.br`, a qual é um alimentador de notícias maior e tem diversas conexões com sites menores. Então a Universidade do Amazonas pode receber todos os grupos da Usenet, enquanto a Universidade do Pantanal irá receber somente algumas hierarquias como `sci`, `comp`, `rec`, etc.. Alguns dos sites menores, como por exemplo um site chamado `corumba`, desejará receber ainda menos grupos, uma vez que não dispõem dos recursos de rede ou hardware necessários. Por outro lado, `corumba` pode desejar receber os grupos de notícias da hierarquia `fj`, a qual a Universidade do Pantanal não recebe. Porém ele mantém outra conexão com a máquina `uberaba.com.br`, a qual recebe os

[3] O formato das mensagens de notícias Usenet está especificado na RFC 1036, "Padrão de Troca de Notícias da USENET".

[4] newsgroups

[5] comp.so.anuncios.linux

grupos do `fj` e alimenta ainda a máquina `corumba`. As notícias fluem conforme a figura 16.1.

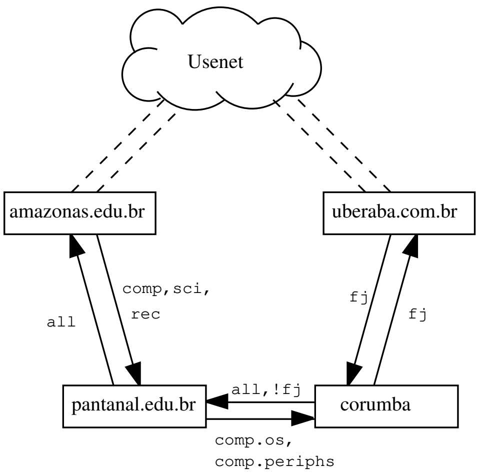

Figura 16.1: Fluxo de noticias através da Universidade do Pantanal

Os nomes e as setas originadas a partir de `corumba` podem necessitar de algumas explicações. Por padrão, ele deseja enviar todas as notícias recebidas localmente para `pantanal.edu.br`. As notícias enviadas para `corumba` a partir de `pantanal.edu.br` estão identificadas como `all,!fj`, indicando que todos os grupos, exceto aqueles sob `fj` são enviados.

16.3 Como a Usenet Lida com Notícias?

Atualmente a Usenet cresceu em proporções enormes. Sites que carregam todas as notícias normalmente transferem algo como centenas de megabytes por dia. Obviamente isso requer muito mais que transferir alguns arquivos. Vamos então verificar como a maioria dos sistemas `Unix` lida com notícias da Usenet.

Notícias são distribuídas através da Internet por vários transportes. O meio historicamente mais comum usado é o UUCP, porém atualmente o tráfego principal é realizado através dos sites Internet. O algoritmo usado é chamado *alimentador*: cada site contém um número de conexões (*alimentadoras de notícias*) para outros sites. Qualquer artigo gerado ou recebido pelo sistema de notícias local é reenviado para eles, a menos que o artigo já tenha sido visto naquele site, sendo descartado nestes casos. Um site pode reconhecer outros sites onde o artigo já tenha sido visto através do campo de cabeçalho Path:. Este cabeçalho contém uma lista de todos os sistemas pelos quais o artigo foi reenviado através de uma notação no formato "bang" (separados por !).

Para distinguir artigos, reconhecendo aqueles que estejam duplicados, os artigos Usenet têm uma identificação de mensagem (especificada no campo de cabeçalho denominado `Message-Id:`, a qual combina o nome do site de postagem e um número serial no formato "<*serial@site*>". Para cada artigo processado, o sistema de notícias registra o fato em um arquivo de *histórico*, o qual é utilizado na verificação de novas mensagens que chegam ao site.

O fluxo entre quaisquer dois sites pode ser limitado por dois critérios: no primeiro o artigo é assinalado como pertencente à uma distribuição (através do campo de cabeçalho `Distribution:`, o qual pode ser usado para confinar a mensagem a um certo grupo de sites. No segundo caso os grupos de notícias trocados podem ser limitados pelos sistemas remetentes, destinatários ou ambos. Uma definição do conjunto de grupos de notícias e distribuições permitidas para transmissão para um determinado site é normalmente mantida no arquivo `sys`.

Um número crescente de notícias normalmente requer que melhorias sejam implementadas neste esquema. Em redes UUCP, a coisa natural a se fazer é coletar notícias em um determinado período de tempo e combiná-las em um único arquivo, o qual é comprimido e enviado para um site remoto. Isso é chamado *loteamento*[6].

Uma técnica alternativa é o protocolo *ihave/sendme*[7] que evita a duplicação de artigos que estão sendo transferidos, além de economizar banda de comunicação. Ao invés de colocar todos os artigos em arquivos de lote e enviá-los em conjunto, somente as identificações dos artigos são combinadas em um imenso arquivo "eutenho" e enviadas para o site remoto. Este lê as mensagens, compara com seu arquivo de histórico e retorna uma lista de artigos que ele deseja em um arquivo do tipo "meenvie". Isso faz com que somente estes artigos sejam enviados.

Obviamente, eutenho/meenvie somente faz sentido ao envolver dois grandes sites que recebem notícias de diversos alimentadores independentes e que monitoram um ao outro com freqüência suficiente para que exista um fluxo eficiente de notícias.

Sites que estão na Internet geralmente baseiam-se em programas que utilizam TCP/IP e que utilizem o Protocolo de Transferência de Notícias de Rede - NNTP.[8]

[6] batching

[7] eutenho/meenvie

[8] Descrito na RFC 977.

Ele transfere notícias entre alimentadores e provê acesso Usenet para usuários individuais ou sistemas remotos.

NNTP conhece três maneiras diferentes de transferir notícias. Uma na versão em tempo real do eutenho/meenvie, também conhecida como *enviando* notícias. A segunda técnica é chamada *recebendo* notícias, na qual o cliente solicita uma lista de artigos de um determinado grupo de notícias ou hierarquia que tenha sido recebida pelo servidor após uma data especificada e escolhe entre os não encontrados no arquivo de histórico. A terceira forma permite a leitura de notícias interativamente e permite que um leitor de notícias recupere artigos a partir de grupos de notícias especificados, assim como a postagem de artigos com informações incompletas de cabeçalho.

Em cada site, notícias são mantidas em uma hierarquia de diretórios sob o `/var/spool/news`, cada artigo em um arquivo separado e cada grupo de notícias em um diretório em separado. O nome do diretório é constituído pelo nome do grupo de notícias, com os componentes fazendo parte do nome do caminho. Por exemplo, artigos `comp.os.linux.misc` são mantidos no diretório `/var/spool/comp/os/linux/misc`. Os artigos em um grupo de notícias recebem números na ordem em que chegam ao site. Este número é utilizado como nome do arquivo. A faixa de números de artigos que estejam on-line no momento é mantida em um arquivo chamado `active`, o qual serve simultaneamente como uma lista dos grupos de mensagens conhecidas no site local.

Uma vez que o espaço em disco é um recurso limitado[9], é necessário eliminar artigos após um determinado tempo. Isso é chamado de *expiração*. Normalmente artigos de certos grupos e hierarquias expiram após um determinado número de dias transcorridos da sua chegada. Isso pode ser redefinido pelo remetente do artigo, o qual pode especificar uma data de expiração no campo `Expires:` do cabeçalho do artigo.

[9] Algumas pessoas acusam a Usenet como uma conspiração dos vendedores de discos rígidos e modems.

Capítulo 17

C News

Um dos mais populares pacotes de softwares para notícias na rede é denominado C News. Foi desenvolvido para sites que enviam notícias através de conexões UUCP. Este capítulo irá discutir os principais conceitos do C News, o processo de instalação e as tarefas de manutenção.

C News armazena os seus arquivos de configuração em `/usr/lib/news` e muitos de seus binários no diretório `/usr/lib/news/bin`. Os artigos são mantidos sob o diretório `/var/spool/news`. Deve-se estar seguro de que praticamente todos os arquivos nestes diretórios sejam de propriedade do usuário `news` e do grupo `news`. Muitos problemas começam quando arquivos tornam-se inacessíveis para C News. Deve ser uma regra de trabalho utilizar-se sempre o usuário `news` através do comando `su` antes que qualquer atividade seja executada. A única exceção é o programa `setnewsids`, o qual é usado para configurar a identidade real dos usuários para alguns programas de notícias. Ele deve pertencer ao superusuário `root` e deve ter o bit de configuração setuid habilitado.

A seguir, descreveremos todos os arquivos de configuração de C News em detalhes, e mostraremos como manter um site em funcionamento.

17.1 Entregando Notícias

Artigos podem ser entregues ao C News de diversas formas. Quando um usuário local posta um artigo, o leitor de notícias normalmente envia o artigo para o comando `inews`, o qual complementa as informações do cabeçalho. Notícias oriundas de sites remotos, seja um único artigo ou um lote completo, serão enviadas para o comando `rnews`, o qual as armazena no diretório `/var/spool/news/in.coming`, de onde elas serão recuperadas posteriormente pelo programa `newsrun`. Qualquer que seja a forma utilizada, os artigos serão tratados posteriormente pelo comando `relaynews`.

Para cada artigo, o programa `relaynews` inicialmente verifica se ele já foi visto no site local, através da pesquisa da identificação do artigo no arquivo `history`. Artigos duplicados serão ignorados. A seguir, o programa `relaynews` pesquisará a linha de cabeçalho identificada pela expressão `Newsgroups:`, a fim de identificar se o site local recebe algum artigo do grupo indicado. Em caso positivo e o grupo de notícias estando listado no arquivo `active`, o programa `relaynews` tenta armazenar o artigo no diretório de arquivos temporários de notícias. Caso o diretório não exista, ele será criado. A identificação do artigo será então incluída no arquivo `history`. Caso ele já conste do arquivo de histórico, o programa `relaynews` ignorará o artigo.

Caso o programa `relaynews` falhe em armazenar um artigo recebido devido ao fato do grupo não estar listado no arquivo `active` local, o artigo será movido para o grupo `junk`.[1] O programa `relaynews` irá ainda verificar os artigos com datas inadequadas e os rejeitará. Lotes que sejam rejeitados por qualquer motivo serão movidos para o arquivo `/var/spool/news/in.coming/bad` e uma mensagem de erro será registrada.

Após isso, o artigo será retransmitido para todos os demais sites que requisitem as notícias pertencentes a esses grupos, usando o transporte especificado para cada site em particular. Para assegurar-se que o artigo não seja enviado para um site que já o tenha recebido, cada site de destino é comparado novamente com o campo de cabeçalho `Path:` do artigo, o qual contém uma lista dos nomes dos sites de destino pelos quais o artigo já tenha passado, escrito em um estilo UUCP bang, ou seja nomes separados por pontos de exclamação. Somente se o site de destino não aparecer na lista, o artigo será enviado.

C News é comumente usado para reenviar notícias entre sites UUCP, ainda que seja possível usá-lo também em um ambiente NNTP. Para entregar mensagens em

[1] Há uma diferença entre os grupos que existem no site local e os que você deseja receber. Por exemplo, a subscrição da lista `comp.all`, significa que serão recebidos todos os artigos sob a hierarquia `comp`, mas se por exemplo somente uma parte dos grupos sob `comp` estiver listada no arquivo `active`, somente os artigos destinados aqueles grupos serão tratados nos respectivos diretórios, sendo que os direcionados aos demais serão enviados para `junk`.

um site remoto UUCP — tanto um artigo quanto lotes inteiros — é utilizado o programa `uux` para executar o comando `rnews` no site remoto e entregar o artigo ou o lote na entrada padrão.

Quando o loteamento de mensagens em um determinado site é habilitado, C News não envia imediatamente qualquer artigo recebido, mas sim adiciona o nome do caminho a um arquivo, normalmente `out.going/`*`site`*`/togo`. Normalmente, um programa loteador é executado a partir de uma entrada no utilitário crontab.[2] Este colocará os artigos em um ou mais arquivos, opcionalmente os compacta e os envia para o programa `rnews` em um site remoto.

A figura 17.1 mostra o fluxo através do programa `relaynews`. Artigos devem ser transmitidos do site local (caracterizados pelo indicativo `ME`), para um site chamado `poxoreo` via email, e para um site chamado `coxim`, no qual o loteamento está habilitado.

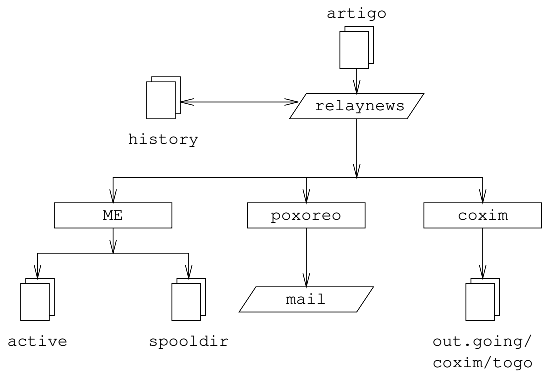

Figura 17.1: Fluxo de Notícias Através do `relaynews`

17.2 Instalação

Para instalar C News, deve-se descompactar os arquivos nos locais apropriados, caso isso ainda não tenha sido feito, e devem ser editados os arquivos de configuração abaixo listados. Eles estão localizados no diretório `/usr/lib/news`. Seus formatos serão discutidos nas seções seguintes.

[2]Note que a entrada deve estar na tabela de execuções do crontab do `news`, para que as permissões não criem dificuldades.

`sys` Provavelmente deve-se modificar a linha `ME` que descreve o sistema local, sendo que um parâmetro `all/all` pode ser um palpite confiável. Deve-se ainda adicionar uma linha para o site para o qual se deseja reenviar notícias. Caso o sistema local seja um site "folha" (vide UUCP), então o arquivo `sys` terá um conteúdo similar ao seguinte:

```
ME:all/all::
coxim/coxim.pantanal.edu.br:all/all,!local:f:
```

`organization` Este arquivo contém o nome da organização. Por exemplo, "*Cervejaria Virtual Ltda.*". Em uma máquina pessoal pode-se usar a expressão "site privado", ou algo similar. Muitas pessoas não poderão acionar o site local adequadamente, caso este arquivo não esteja devidamente customizado.

`newsgroups` Contém uma lista de todos os grupos de notícias disponíveis, com uma descrição on-line da finalidade de cada um deles. Estas descrições são freqüentemente usadas pelo leitor de notícias ao listar os grupos disponíveis na opção de subscrição.

`mailname` O nome do site usado no envio de mensagens de correio eletrônico, por exemplo `cvirtual.com.br`.

`whoami` O nome do site a ser usado no envio de notícias. Freqüentemente é usado o nome do site no UUCP, como por exemplo `cvirtual`.

`explist` Deve-se editar este site para configurar as datas de expiração de alguns grupos de notícias especiais. Espaço em disco pode ser um fator determinante nesta definição.

Para criar uma hierarquia inicial dos grupos e notícias, deve-se obter um arquivo `active` e um arquivo `newsgroups` a partir do site que alimentará o sistema local. Estes devem ser instalados no diretório `/usr/lib/news`, com o dono igual a news e modo de permissão igual a 644. Devem ser removidos todos os grupos `to.*` do arquivo de grupos ativos e adicionada a linha `to.`*`site_local`* e `to.`*`sites_de_destino`*, assim como `junk` e `control`. Os grupos `to.*` são normalmente usados para troca de mensagens do tipo eutenho/meenvie, mas devem ser criados independentemente do uso desta modalidade. A seguir, deve-se substituir os números dos artigos, atualizando-se o segundo e terceiro campos do arquivo `active` usando-se o seguinte comando:

```
# cp active active.old
```

```
# sed 's/ [0-9]* [0-9]* / 0000000000 00001 /' active.old > active
# rm active.old
```

O segundo comando aciona o comando `sed(1)`, um dos meus comandos `Unix` favoritos. O seu objetivo neste caso é substituir duas expressões compostas por dígitos por uma contendo zeros e outra contendo `000001`, respectivamente.

Finalmente, deve-se criar um novo diretório de tarefas temporárias e os subdiretórios usados para mensagens recebidas e notícias a enviar:

```
# cd /var/spool
# mkdir news news/in.coming news/out.going
# chown -R news.news news
# chmod -R 755 news
```

Caso se esteja usando versões mais antigas do C News, deve-se ainda criar o diretório `out.master` sob o diretório de tarefas temporárias.

Caso se esteja utilizando leitores de notícias diferentes dos distribuídos com o C News, é possível que alguns deles estejam esperando que as notícias sejam armazenadas em `/usr/spool/news` ao invés de `/var/spool/news`. Caso o leitor aparente não estar conseguindo localizar as notícias, deve-se criar uma ligação simbólica de `/usr/spool/news` para `/var/spool/news`.

Agora, o site local está pronto para receber as notícias. Note que não se deve criar quaisquer diretórios além dos acima descritos, já que cada vez que o programa C News recebe um artigo de um grupo cujo diretório ainda não tenha sido criado, ele fará a criação automaticamente.

Isso ocorrerá para *todos* os grupos para os quais um artigo tenha sido enviado. Então, após algum tempo, pode-se verificar que o diretório de tarefas temporárias de notícias estará abarrotado de diretórios de grupos de notícias que nunca foram solicitados, como por exemplo `alt.lang.teco`. Para evitar que isto ocorra pode-se simplesmente remover todos os grupos indesejados do arquivo `active`, ou regularmente executar um programa que esvazie os diretórios sob `/var/spool/news` (exceto os arquivos `out.going` e `in.coming`).

O C News necessita de um usuário para enviar mensagens de erro e também relatórios da sua situação. Por padrão, ele é o usuário `usenet`. Caso se utilize o padrão, pode-se configurar um nome alternativo para ele, o que provocará o reenvio das mensagens para uma ou mais pessoas responsáveis pela sua administração

(Capítulos 14 e 15 descrevem como fazer isso utilizando os programas `smail` e `sendmail`). Pode-se ainda alterar o seu comportamento através da configuração da variável de ambiente `NEWSMASTER` para um nome adequado. Ela deve ser atualizada ainda no arquivo de acionamentos programados das tarefas de `notícias`, assim como toda vez que seja acionada uma tarefa administrativa manualmente. Certamente o uso de um nome alternativo é mais simples.

Quando se estiver acessando o arquivo `/etc/passwd`, deve-se estar seguro de que todos os usuários têm seu nome real no campo `pw_gecos` devidamente configurado (o quarto campo de cada linha). Esta é uma questão de convivência na Internet, que faz com que o nome real do remetente apareça no campo `From:` do artigo. Obviamente, pode-se desejar fazer isso quando se utilize somente o envio de mensagens via correio.

17.3 O Arquivo sys

O arquivo `sys`, localizado em `/usr/lib/news`, controla a hierarquia de recepção e envio de notícias para outros sites. Apesar de haver uma ferramenta de manutenção chamada `addfeed` e `delfeed`, é recomendada a sua edição manualmente.

O arquivo `sys` contém entradas para cada site para o qual notícias são enviadas, assim como uma descrição dos grupos que eles aceitarão. Uma entrada neste arquivo tem o seguinte formato:

```
site[/exclusões]:lista_de_grupos[/lista_de_distribuição][:indicadores[:comandos]]
```

As entradas podem utilizar mais de uma linha. Para tanto deve ser indicado o caractere de barra invertida (\). Um sinal (#) significa um comentário.

`site` Este é o nome do site ao qual as entradas se aplicam. Normalmente é utilizada a identificação UUCP. Deve haver também uma entrada para o site local neste arquivo, ou nenhum artigo será recebido.

O nome especial de site `ME` significa o site local. Na entrada `ME` são definidos todos os grupos que se deseja receber e armazenar localmente. Artigos que não coincidam com `ME` serão destinado ao grupo `junk`.

Uma vez que o C News verifica o nome do *`site`* com os nomes dos sites no campo de cabeçalho da mensagem `Path:`, deve-se estar seguro de que eles realmente coincidem. Alguns sites usam seu nome totalmente qualificado neste campo, ou um nome alternativo, como por exemplo *news.site.domínio*. Para evitar que artigos sejam devolvidos para estes sites, deve-se adicioná-los à lista de exclusões, separados por vírgulas.

Para uma entrada aplicada ao site `coxim`, por exemplo, o campo site poderá conter o seguinte `coxim/coxim.pantanal.edu.br`.

`lista_de_grupos` Esta é uma lista, separada por vírgulas, de subscrição de grupos e hierarquias de um site em particular. Uma hierarquia pode ser especificada fornecendo-se o prefixo da hierarquia, (tais como `comp.os`) o que especificará todos os grupos cujo nome comece com este prefixo. Opcionalmente ele é seguido pela palavra chave `all` (por exemplo `comp.os.all`).

Uma hierarquia ou grupo é excluída do reenvio quando forem precedidos por um ponto de exclamação. Caso o grupo de noticias seja verificado contra uma lista, a coincidência mais longa será aplicada. Por exemplo, caso *lista_de_grupos* contenha:

```
!comp,comp.os.linux,comp.folklore.computers
```

nenhum grupo da hierarquia `comp` exceto `comp.folklore.computers` e todos os grupos sob `comp.os.linux` serão recebidos por este site.

Caso se deseje receber todas as notícias de todos os grupos, pode-se simplesmente utilizar os parâmetros `all` como **lista_de_grupos**.

`lista_de_distribuição` Este valor é uma forma complementar do parâmetro **lista_de_grupos** separada por uma barra e contém uma lista das distribuições a serem reenviadas. Novamente pode-se excluir certas distribuições precedendo seu nome com um ponto de exclamação. Todas as distribuições podem ser definidas através do parâmetro `all`. A omissão do parâmetro **lista_de_distribuição** implica em uma lista de `todas` as distribuições.

Por exemplo, pode-se usar uma lista de distribuição com o formato `all,!local` para evitar que as notícias de uso somente local sejam enviadas para sites remotos.

Há pelo menos duas distribuições previstas: `world`, a qual é freqüentemente a distribuição padrão usada, quando não há definição deste parâmetro pelo usuário e `local`. Podem haver outras distribuições como regiões, estados, países, etc.. Finalmente há duas distribuições usadas somente pelo C News: `sendme` e `ihave`, as quais são usadas pelo protocolo meenvie/eutenho.

O uso de distribuições é objeto de discussões. Por um lado, alguns leitores de notícias criam distribuições ineficientes, contendo somente o nível mais elevado da hierarquia, como por exemplo `comp` ao postar para `comp.os.linux`. Distribuições aplicadas a regiões também são bastante questionadas, uma vez que as notícias freqüentemente são enviadas além de seus limites quando transitam pela Internet.[3] Distribuições aplicadas à uma organização porém têm uma aplicação bastante prática, por exemplo para prevenir que informações internas sejam enviadas para o mundo exterior. Este objetivo porém pode ser melhor atendido através da criação de grupos de notícias e hierarquias em separado.

`indicadores` Descrevem certos parâmetros usados no envio de mensagens. Pode estar vazio ou conter uma combinação dos seguintes:

`F` Habilita o loteamento de mensagens.

`f` Quase idêntico ao indicador `F`, porém permite que o programa C News calcule o tamanho das mensagens de saída com maior precisão.

`I` Faz com que C News produza uma lista de artigos nos padrões do protocolo eutenho/meenvie. Modificações adicionais nos arquivos `sys` e `batchparms` são necessárias para habilitar este protocolo.

`n` Cria arquivos de lote para transferências via clientes NNTP, como por exemplo `nntpxmit` (veja o capítulo 18). Os arquivos de lote contêm os nomes dos artigos com a identificação da mensagem.

`L` Indica ao C News que transmita somente os artigos postados no site local. Este indicador pode ser seguido por um número decimal *n*, o qual faz com que o C News somente transfira artigos postados com até *n* pontos de passagem a partir do site local. C News determina o número de pontos a partir do campo `Path:` no cabeçalho da mensagem.

`u` Indica ao C News para lotear artigos somente de grupos não moderados.

`m` Indica ao C News para lotear somente artigos de grupos moderados.

Os seguintes parâmetros podem ser usados somente um por vez: `F`, `f`, `I` ou `n`.

`comandos` Este campo contém um comando a ser executado para cada artigo, a menos que o loteamento tenha sido habilitado. O artigo será enviado para o

[3] Não é incomum um artigo destinado digamos à região de Hamburgo, ir para Frankfurt via `reston.ans.net` na Holanda, ou mesmo via algum site nos EUA.

comando através da entrada padrão. Isso deve ser usado somente em grupos pequenos pois pode produzir uma carga muito expressiva para o site.

O comando padrão é o seguinte:

```
uux - -r -z sistema!rnews
```

o qual aciona o comando `rnews` no sistema remoto e envia o artigo para a sua entrada padrão.

O caminho padrão de pesquisa para os comandos fornecidos neste campo é `/bin:/usr/bin:\newsbin/batch`. Este diretório contém uma série de programas interpretados cujos nomes começam com `via`, os quais são rapidamente descritos mais adiante neste capítulo.

Caso o loteamento de mensagens esteja habilitado usando-se os parâmetros `F` ou `f`, `I` ou `n`, C News esperará encontrar um nome de arquivo neste campo, ao invés de um comando. Caso o nome do arquivo não comece com uma barra (/), ele assume o seu endereço relativo a partir do diretório `/var/spool/news/out.going`. Caso o campo esteja vazio, o padrão será *sistema*`/togo`.

Ao configurar o C News, deve-se provavelmente criar um arquivo `sys` próprio. Para facilitar esta tarefa, apresentamos a seguir um exemplo simples para o domínio `cvirtual.com.br`:

```
# o site local recebe tudo o que for transmitido para ele
ME:all/all::

# Tudo o que é recebido é enviado para coxim, exceto os artigos locais e
# relacionados com cervejarias. Utiliza-se loteamento de mensagens.
coxim/coxim.pantanal.edu.br:all,!to,to.coxim/all,!local,!brewery:f:

# O grupo comp.risks é enviado para sherry@guadalajara.com.es
guadalajara:comp.risks/all::rmail sherry@guadalajara.com.es

# Caruaru recebe alguns grupos
caruaru/caruaru.pantanal.edu.br:comp.os.linux,rec.humor.oracle/all,!local:f:

# Registrar mapas de artigos para processamento posterior
usenet-maps:comp.mail.maps/all:F:/var/spool/uumaps/work/batch
```

17.4 O Arquivo active

O arquivo `active` está localizado em `/usr/lib/news` e lista todos os grupos conhecidos pelo site local e os artigos atualmente ativos. Raramente ele terá que ser editado, porém apresentaremos uma descrição de seu conteúdo. As suas entradas têm o seguinte formato:

```
grupo_de_notícias maior menor perm
```

grupo_de_notícias é, obviamente, o nome do grupo. *maior* e *menor* são os maiores e menores números dos artigos atualmente disponíveis. Caso nenhuma notícia esteja disponível, *menor* é igual a *maior*+1.

O número da menor mensagem disponível, isto é o que o campo *menor* significa. Porém muitos leitores de notícias não o utilizam. Por exemplo, `trn` checa este campo para verificar se deve excluir algum artigo de sua base de dados. Para atualizar o campo *menor* deve-se executar o comando `updatemin` regularmente (ou em versões mais atualizadas do C News, o programa `upact`).

perm é um parâmetro que define as permissões de acesso de usuários ao grupo. Ele pode ter um dos seguintes valores:

`y` Usuários podem postar artigos para este grupo.

`n` Usuários não podem postar artigos para este grupo, porém podem ler os artigos existentes.

`x` Este grupo será localmente desabilitado. Isso acontece quando determinados grupos contêm material não indicado para o site local. Artigos recebidos destes grupos não serão armazenados localmente, apesar de poderem ser enviados para outros sites que os solicitem.

`m` Indica um grupo moderado. Quando o usuário tentar postar algum artigo para este grupo, um leitor de notícias inteligente irá notificar o usuário sobre o assunto e enviará o artigo ao moderador do grupo. O seu endereço é obtido a partir do arquivo `moderators` em `/usr/lib/news`.

`=nome_real_grupo` Indica que o *grupo_de_notícias* é tratado localmente como um nome alternativo, para um grupo chamado *nome_real_grupo*. Todos os artigos postados para o *grupo_de_notícias* serão redirecionados para o nome real.

No C News, normalmente não se tem acesso direto a este arquivo. Grupos podem ser adicionados ou suprimidos através do comando `addgroup` e `delgroup` (veja a seção 17.10). Quando grupos são adicionados ou removidos de toda a Usenet, isso normalmente é feito através do envio da mensagem de controle `newgroup` ou `rmgroup`, respectivamente. *Nunca enviem uma mensagem deste tipo!* Para maiores instruções sobre como criar um grupo de notícias deve-se ler as mensagens do grupo `news.announce.newusers`.

Um arquivo relacionado com `active` é o `active.times`. Toda vez que um grupo for criado, C News registra uma ocorrência neste arquivo, contendo o nome do grupo, a data da criação, se ele foi criado localmente ou através de uma mensagem de `newgroup` e quem o fez. Isso é interessante para os leitores de notícias que desejam avisar ao usuário sobre a criação recente de novos grupos. Este arquivo é usado também pelo comando `NEWGROUPS` do NNTP.

17.5 Loteando Artigos

Lotes de notícias seguem um formato particular que é idêntico ao utilizado por Bnews, C News e INN. Cada artigo é precedido pela seguinte linha:

```
#! rnews número
```

onde *número* é o total de bytes do artigo. Quando a compressão do lote é realizada, o arquivo resultante é comprimido como um todo e precedido por uma outra linha, indicando o comando a ser usado na descompressão do arquivo. A ferramenta padrão utilizada é o comando `compress`, o qual é acionado por

```
#! cunbatch
```

Algumas vezes, ao se enviar lotes através de programas de mensagens que removem o oitavo bit de todos os dados, um lote comprimido pode ser protegido utilizando-se a chamada c7-codificação, sendo que estes lotes serão gerados pelo programa `c7unbatch`.

Quando um lote é enviado para um comando `rnews` em um site remoto, ele verifica se estas marcas estão presentes e executa o processamento de forma adequada. Alguns sites utilizam ainda ferramentas de compressão tais como `gzip` e precedem os arquivos compactados com o programa `zunbatch`. C News não reconhece cabeçalhos não padronizados.

Em C News, o loteamento de notícias é executado pelo programa `/usr/lib/news/bin/batch/sendbatches`, o qual recebe uma lista de artigos a partir do arquivo `site/togo`, e os coloca em uma série de arquivos de lotes de notícias. Ele deve ser executado a cada hora, ou mesmo mais freqüentemente, dependendo do volume do tráfego.

Sua operação é controlada pelo arquivo `batchparms` localizado em `/usr/lib/news`. Este arquivo descreve o tamanho máximo de lote permitido para cada site, o programa de loteamento e compressão opcional a ser usado e o meio de transporte a ser utilizado na entrega para o site remoto. Pode-se especificar parâmetros de loteamento a nível de site de destino, assim como definir um conjunto de parâmetros padrões para os sites não definidos explicitamente.

Para executar o loteamento para um site específico, pode-se executar o seguinte:

```
# su news -c "/usr/lib/news/bin/batch/sendbatches site"
```

Ao ser acionado sem argumentos, o programa `sendbatches` tratará todas as filas de lotes. A interpretação de "todas" depende da presença de uma entrada no arquivo `batchparms`. Caso uma seja encontrada, todos os diretórios em `/var/spool/news/out.going` serão verificados, caso contrário ele executará um ciclo para todas as entradas existentes em `batchparms`. Note que `sendbatches`, ao pesquisar o diretório `out.going`, trata somente aqueles diretórios que não contenham um ponto ou o sinal (`@`) no nome dos sites.

Ao se instalar o C News, é disponibilizado um arquivo `batchparms`, o qual contém um exemplo bastante razoável. Cada entrada no arquivo é composta por uma linha composta por seis campos, separados por espaços ou tabulações:

```
site tamanho max loteador programa_de_compressão transporte
```

Estes campos têm o seguinte significado:

`site` é o nome do site ao qual os parâmetros se aplicam. O arquivo `togo` deste site deve residir em `out.going/togo` sob o diretório de tarefas temporárias de notícias. Um site chamado `/default/` indica a entrada padrão.

`tamanho` é o tamanho máximo dos artigos loteados (antes da compressão). Para artigos individuais maiores que este valor, C News faz uma exceção e os coloca em lotes individuais.

`max` é o número máximo de lotes criados e programados para transferência antes de uma interrupção na transferência para este site em particular. Isto é útil

especialmente no caso de sites remotos que estejam indisponíveis por um longo período de tempo, uma vez que evita que C News superlote os diretórios de arquivo temporários UUCP com milhões de lotes de notícias.

C News determina o número de lotes em fila usando o programa interpretado chamado `queulen` em `/usr/lib/news/bin`. A versão de Vince Skahan chamada `newspak` contém um programa compatível com UUCP BNU. Caso se use um conjunto diferente de diretórios de tarefas temporárias, como por exemplo, com o Taylor UUCP, deve-se utilizar um programa próprio.[4]

O campo *`loteador`* contém o comando usado para produzir um lote a partir da lista de artigos contida no arquivo `togo`. Para envios normais, ele normalmente será o programa `batcher`. Para outros propósitos, loteadores alternativos poderão ser disponibilizados. Por exemplo, o protocolo eutenho/meenvie requer uma lista a ser adequada ao controle de mensagens disponíveis ou solicitadas, as quais são postadas para o grupo de notícias `to.`*`site`*. Isso é executado pelos programas `batchih` e `batchsm`.

O campo *`programa_de_compressão`* especifica o comando usado para a compressão do lote. Normalmente ele é denominado `compcun`, um programa interpretado que produz um lote comprimido.[5] Alternativamente, pode-se indicar um compressor que use o `gzip`, chamado, digamos, `gzipcun` (porém ele terá que ser desenvolvido pelo administrador). Neste caso deve-se estar seguro de que o programa `uncompress` no site remoto tem capacidade de reconhecer arquivos comprimidos com `gzip`.

Caso o site remoto não tenha um comando `uncompress` disponível, pode-se especificar o parâmetro como `nocomp`, o que provocará que nenhuma compressão seja executada.

O último campo, *`transporte`*, descreve a forma de transporte a ser utilizada. Alguns comandos para diferentes padrões estão disponíveis com seus nome prefixados como a expressão `via`. `sendbatches` envia para o programa o nome do site de destino diretamente na linha de comando. Caso a entrada em `batchparms` não seja igual a `/default/`, ele derivará o nome do site a partir do campo *`site`*,

[4] Caso o número de arquivos temporários não seja importante (porque há somente uma pessoa usando o sistema, por exemplo, e não se está escrevendo artigos com gigabytes), pode-se substituir o programa por um simples comando `exit 0`.

[5] Uma vez que ele é distribuído com C News, `compcun` usa `compress` com a opção 12 bits, já que este é o menor denominador comum para a maioria dos sites. Pode-se produzir um cópia deste programa, digamos `compcun16` e usar-se por exemplo uma compressão de 16 bits. A melhoria é impressionante.

desmembrando qualquer valor existente após um ponto ou barra. Caso a entrada seja igual a `/default/`, o diretório `out.going` será utilizado.

Há dois comandos que usam `uux` para executar `rnews` no sistema remoto: `viauux` e `viauuxz`. Este último configura o parâmetro `-z` para (antigas versões) o programa `uux` evitar que as mensagens de sucesso sejam enviadas para cada artigo entregue. Outro comando, `viamail`, envia lotes de artigos para o usuário `rnews` no sistema remoto via correio eletrônico. Obviamente, isso requer que o sistema remoto envie as mensagens destinadas a `rnews` para o seu sistema local de notícias. Para um lista completa dos transportes disponíveis, veja a página de manual do comando `newsbatch(8)`.

Todos os comandos destes últimos três campos devem estar localizados ou no diretório `out.going/`*`site`* ou em `/usr/lib/news/bin/batch`. Muitos deles são programas interpretados, tornando simples a sua adequação para necessidades específicas do site local. Eles são acionados através de conectores de comandos[6]. A lista de artigos a serem enviados é disponibilizada para o loteador na entrada padrão, o qual produz o lote na saída padrão do sistema. Isso é conectado com o compressor e assim por diante. A seguir apresentamos um arquivo de exemplo:

```
# arquivo batchparms para a cervejaria
# site       |tamanho   |máximo    |loteador  |compressor |transporte
#------------+----------+----------+----------+-----------+----------
/default/     100000     22         batcher    compcun     viauux
caruaru       10000      10         batcher    nocomp      viauux
```

17.6 Expiração de Notícias

Em Bnews, a expiração de artigos é executada por um programa chamado `expire`, o qual recebe uma lista de grupos de notícias como argumentos, em conjunto com uma especificação de número de dias após os quais os artigos devem ser considerados expirados. Para ter-se prazos de expiração diferentes para diferentes hierarquias, deve-se configurar um programa interpretado que acione `expire` para cada um deles separadamente. Já o programa C News oferece uma solução mais conveniente: disponibiliza um arquivo chamado `explist`, onde podem ser especificados os grupos de notícias e os intervalos de expiração. Um comando chamado `doexpire` é executado normalmente uma vez ao dia a partir do `cron` e processa todos os grupos de acordo com esta lista.

[6] pipes

Ocasionalmente pode-se desejar manter certos artigos de certos grupos, mesmo após o seu prazo de expiração, como por exemplo os artigos postados em `comp.sources.unix`. Este processo é chamado de *arquivamento*. O arquivo `explist` permite que se indique grupos de notícias para arquivamento.

Uma entrada no arquivo `explist` tem o seguinte formato:

```
lista_de_grupos permissões prazo arquivar
```

`lista_de_grupos` é uma lista dos grupos de notícias, separada por vírgulas, aos quais a entrada se aplica. Hierarquias podem ser especificadas fornecendo-se ao grupo o nome do prefixo, seguida opcionalmente da palavra `all`. Por exemplo, uma entrada para todos os grupos sob `comp.os` pode ser representada por `comp.os` ou `comp.os.all` neste campo.

Na expiração de notícias de um grupo, o nome é verificado nas entradas disponíveis em `explist` na ordem informada. Por exemplo, para eliminar a maioria dos artigos de `comp` após quatro dias, exceto os disponíveis em `comp.os.linux.announce`, que devem ser mantidos por uma semana, basta especificar uma entrada para o último com um prazo igual a 7 e a seguir uma entrada para `comp`, com a especificação de 4 dias.

O campo *`permissões`* define se a entrada se aplica a grupos moderados, não moderados ou a todos, os quais são representados pelos valores `m`, `u` ou `x`, respectivamente.

O terceiro campo, *`prazo`*, normalmente contém um único número, significando o número de dias, após o qual os artigos estarão expirados, caso eles não tenham recebido uma data de expiração própria no campo do cabeçalho definido como `Expires:`. Deve-se atentar que o número de dias refere-se à chegada do artigo no site local e não à data da postagem.

O campo *`prazo`* pode, no entanto, ser um pouco mais complexo. Ele pode ser a combinação de até três números, separados por um hífen. Neste caso, o primeiro indica o número de dias para que o artigo se torne um candidato à expiração. Raramente se utiliza um valor diferente de zeros. O segundo subcampo indica o número de dias padrão após o qual os artigos serão considerados expirados. O terceiro indica o número de dias de expiração incondicional, independente do conteúdo do campo `Expires:`. Caso somente o segundo subcampo seja informado, os outros dois assumirão seus valores padrão. Eles podem ser definidos através da entrada especial `/bounds/`, definida a seguir.

O quarto campo, *arquivar*, define se o grupo de notícias será arquivado e onde. Caso nenhum arquivamento seja definido, um traço deve ser usado para especificar esta situação. Caso contrário deve-se especificar o nome completo do caminho (apontando para um diretório) ou para o caractere (`@`), o qual significa que o diretório padrão de arquivamento deve ser fornecido ao programa `doexpire` através da opção `-a` na linha de comando. Um diretório de arquivamento deve pertencer ao usuário `news`. Quando o programa `doexpire` arquiva um artigo digamos de `comp.sources.unix`, ele o armazena em `comp/sources/unix` sob o diretório de arquivamento, criando-o caso ele não exista. O diretório de arquivamento porém não será criado.

Há duas entradas especiais no arquivo `explist`, nas quais o programa `doexpire` se baseia. Ao invés de uma lista de grupos de notícias, pode-se informar as palavras chaves `/bounds/` e `/expired/`. A entrada `/bounds/` contém os valores padrão para os três valores do campo *prazo* acima descrito.

A expressão `/expired/` determina por quanto tempo C News irá manter as linhas no arquivo `history`. Isso é necessário porque C News não irá remover a linha do arquivo de histórico, uma vez que o artigo correspondente tenha expirado, a fim de evitar que o artigo seja recebido novamente em duplicidade. Caso os arquivos sejam recebidos a partir de um único site, pode-se manter valores pequenos neste campo. De outra forma, algumas semanas pode ser uma valor razoável para sites UUCP, dependendo da experiência do administrador com os prazos médios de recebimento de notícias. Caso o site utilize NNTP e esteja conectado a Internet um prazo médio de uma semana deverá ser mais que suficiente para atender a esta particularidade.

Um exemplo de um arquivo `explist` com alguns prazos de expiração é apresentado a seguir:

```
# Mantém as linhas de histórico por uma semana. Nenhum artigo é mantido por mais
# de 30 dias
/expired/                      x       7       -
/bounds/                       x       0-1-30  -

# grupos que se deseja manter por um tempo maior
comp.os.linux.announce         m       10      -
comp.os.linux                  x       5       -
alt.folklore.computers         u       10      -
rec.humor.oracle               m       10      -
soc.feminism                   m       10      -

# Arquivar os grupos *.sources
comp.sources,alt.sources       x       5       @
```

```
# aplicar os padrões para os grupos técnicos
comp,sci                     x      7      -

# suficientes para um longo final de semana
misc,talk                    x      4      -

# elimina o lixo rapidamente
junk                         x      1      -

# mensagens de controle também não são muito interessantes
control                      x      1      -

# todos os demais grupos são tratados por esta entrada
all                          x      2      -
```

A expiração em C News pode trazer uma série de situações complexas. Uma delas, reside no fato do leitor de notícias utilizar o terceiro campo do arquivo active, o qual contém o número do menor artigo on-line. Na expiração de artigos, C News não atualiza este campo. Caso seja necessário que ele represente a real situação, deve-se executar o programa chamado `updatemin` após cada execução do programa `doexpire`.[7]

Segundo, C News não executa o processo de expiração pesquisando os diretórios de grupos de notícias e sim o arquivo de `histórico`, na verificação dos prazos de expiração.[8] Caso o arquivo de histórico de alguma forma saia de sincronismo, artigos podem ser esquecidos no disco para sempre, já que C News passa simplesmente a ignorar a sua existência.[9] Pode-se corrigir esta situação usando o programa `addmissing` no diretório `/usr/lib/news/bin/maint`, o qual irá adicionar qualquer artigo faltante ao arquivo `history`, ou através do programa `mkhistory`, o qual reconstrói totalmente o arquivo `history`. Não se deve esquecer que na execução destes programas deve-se ser o usuário `news`, ou o arquivo `history` não poderá ser lido pelo programa C News.

17.7 Arquivos Diversos

Há diversos arquivos que controlam o comportamento do C News, mas não são essenciais para o seu funcionamento. Todos eles residem no diretório `/usr/lib/news`. A seguir apresentamos uma descrição sucinta:

[7] Em versões mais antigas do C News, isso era realizado por um programa chamado `upact`.

[8] A data de chegada de um artigo é mantida no campo do meio da linha do histórico e é fornecida em segundos transcorridos deste 1 de janeiro de 1970.

[9] Não sei *porque* isso ocorre, mas eventualmente isso realmente acontece.

`newsgroups` Este é o arquivo que acompanha o arquivo `active`, o qual contém uma lista dos nomes dos grupos de notícias, em conjunto com uma descrição de uma linha sobre o seu tema principal. Este arquivo é automaticamente atualizado toda vez que o C News recebe uma mensagem de controle `checknews` (veja a seção 17.8).

`localgroups` Se você tem um conjunto de grupos locais onde não quer que o C News apresente alguma mensagem de erro toda vez que for recebida uma mensagem `checknews`, coloque seus nomes e descrições neste arquivo, exatamente da mesma forma que eles aparecem no `grupos de notícias`.

`mailpaths` Este arquivo contém os endereços dos moderadores de cada grupo moderado. Cada linha contém o nome do grupo, seguido pelo endereço email do moderador (separado por uma tabulação).

Duas entradas especiais são disponibilizadas por padrão. Elas são `backbone` e `internet`. Ambas disponibilizam — na notação UUCP — o caminho mais próximo para um site que esteja no canal principal de comunicação e do site que compreende os endereços na notação definida na RFC 822 (`user@host`). As entradas padrão são:

```
internet      backbone
```

Não se deve alterar o parâmetro `internet` caso se esteja utilizando os programas `smail` ou `sendmail`, uma vez que eles compreendem a notação de endereçamento definida pela RFC 822.

O parâmetro `backbone` é utilizado toda vez que um usuário posta um artigo para um grupo moderado, cujo moderador não esteja listado explicitamente. Por exemplo, caso o nome do grupo de notícias seja `alt.costura` e o parâmetro `backbone` contenha *`caminho`*`!%s`, C News irá enviar o artigo para *`caminho`*`!alt-costura`, esperando que o site que esteja no canal principal de comunicação consiga descobrir o caminho e reenviar o artigo. Para descobrir qual o caminho a ser usado, deve-se contactar os administradores de notícias do site que envia notícias para o sistema local. Como última alternativa pode-se usar `uunet.uu.net!%s`.

`distributions` Este arquivo não é realmente um arquivo do C News, mas é utilizado pelos leitores de notícias e pelo `nntpd`. Ele contém uma lista das distribuições reconhecidas pelo site local e uma descrição de seus efeitos. Por exemplo, na Cervejaria Virtual temos o seguinte arquivo:

```
world        o mundo inteiro
local        somente este site
br           somente o Brasil
mugnet       somente a MUGNET
fr           somente a França
de           somente a Alemanha
cvirtual     somente a Cervejaria Virtual
```

`log` Este arquivo contém o registros de todas as atividades do C News. Ele é atualizado diariamente pela execução do programa `newsdaily`, que copia os arquivos de registros antigos para `log.o`, `log.oo`, etc..

`errlog` Este é um arquivo de registros das mensagens de erro geradas pelo C News. Ele não inclui artigos ignorados por pertencerem a grupos errados, etc.. Este arquivo é enviado automaticamente para o administrador do sistema de notícias (`usenet` por padrão) caso não esteja vazio.

`errlog` é esvaziado diariamente pelo programa `newsdaily`. Cópias antigas são mantidas em `errlog.o` e companhia.

`batchlog` Registra todas as execuções do `sendbatches`. Sua função é de acompanhamento e é tratado também pelo programa `newsdaily`.

`watchtime` Este é um arquivo vazio criado cada vez que o programa `newswatch` é executado.

17.8 Mensagens de Controle

O protocolo de notícias Usenet reconhece uma categoria especial de artigos, os quais acionam determinadas ações ou provocam determinadas respostas no sistema de notícias. Elas são chamadas de *mensagens de controle*. São reconhecidas pela presença do campo `Control:` no cabeçalho do artigo, o qual contém o nome da ação a ser executada. Há diversos tipos, todas administradas por programas interpretados localizados em `/usr/lib/news/ctl`.

Muitos destes executarão as ações automaticamente, no momento em que o artigo for processado por C News, sem notificar o administrador. Por padrão, somente mensagens de `checkgroups` serão tratadas pelo administrador, porém isso pode ser alterado[10] através da edição dos programas.

[10] Há uma curiosa entrada na RFC 1036 (p.12) que diz: "Implementadores e administradores

17.8.1 A Mensagem de `cancelamento`

A mais conhecida mensagem de controle é a denominada `cancel`, através da qual um usuário pode cancelar um artigo enviado por ele anteriormente. Isso efetivamente remove o artigo dos diretórios de tarefas temporárias, caso ele exista. A mensagem `cancel` é reenviada para todos os sites que recebem notícias dos grupos envolvidos, independentemente se o artigo já tenha sido enviado ou não. Alguns sistemas de notícias permitem que usuários cancelem mensagens de outras pessoas, porém isso é fortemente contra indicado.

17.8.2 `newgroup` e `rmgroup`

Duas mensagens que lidam com a criação e remoção de grupos de notícias são as mensagens `newgroup` e `rmgroup`. Grupos de notícias sob as hierarquias usuais podem ser criados somente após uma discussão e votação entre os usuários da Usenet. As regras aplicadas à hierarquia `alt` permitem algo próximo à anarquia. Para maiores informações veja as postagens regulares em `news.announce.newusers` e `news.announce.newgroups`. Nunca envie uma mensagem `newgroup` ou `rmgroup`, a menos que se esteja completamente seguro do que se está fazendo.

17.8.3 A Mensagem `checkgroups`

Mensagens `checkgroups` são enviadas pelos administradores de sistemas de notícias para fazer com que todos os sites na sua rede estejam sincronizados com o arquivo `active` local e com a realidade da Usenet. Por exemplo, provedores de acesso a Internet podem enviar tal mensagem para os sites de seus clientes. Uma vez ao mês a mensagem "oficial" de `checkgroups` para as hierarquias maiores é postada para o `comp.announce.newgroups` pelo seu moderador. Porém ela é postada como um artigo comum e não como uma mensagem de controle. Para executar a operação de `checkgroups`, o artigo deve ser salvo em um arquivo, digamos `/tmp/check`, deve ser removido o início da mensagem de controle e enviada para o programa `checkgroups` usando-se o seguinte comando:

```
# su news -c "/usr/lib/news/bin/ctl/checkgroups« /tmp/check
```

podem escolher entre a execução automática de mensagens de controle ou o seu enfileiramento para processamento anual."

Este procedimento irá atualizar o arquivo `newsgroups`, adicionando os grupos listados em `localgroups`. O arquivo antigo `newsgroups` será movido para `newsgroups.bac`. Note que a postagem desta mensagem localmente raramente funcionará, porque `inews` não aceita este tipo de mensagem como artigo.

Caso o programa C News encontre inconsistências entre a lista de `checkgroups` e o arquivo `active`, ele produzirá uma lista de comandos que atualizarão o sistema local e enviará uma mensagem para o administrador. A saída deste comando tem tipicamente a seguinte aparência:

```
From news Sun Jan 30 16:18:11 1999
Date: Sun, 30 Jan 99 16:18 MET
From: news (News Subsystem)
To: usenet
Subject: Problemas com o arquivo active

Os seguintes grupos de notícias não são válidos e devem ser removidos.
        alt.ascii-art
        bionet.molbio.gene-org
        comp.windows.x.intrisics
        de.answers

Pode-se fazer isso através dos seguintes comandos:
         /usr/lib/news/bin/maint/delgroup alt.ascii-art
         /usr/lib/news/bin/maint/delgroup bionet.molbio.gene-org
         /usr/lib/news/bin/maint/delgroup comp.windows.x.intrisics
         /usr/lib/news/bin/maint/delgroup de.answers

Os seguintes grupos de notícias não foram localizados:
        comp.binaries.cbm
        comp.databases.rdb
        comp.os.geos
        comp.os.qnx
        comp.unix.user-friendly
        misc.legal.moderated
        news.newsites
        soc.culture.scientists
        talk.politics.crypto
        talk.politics.tibet
```

Ao se receber uma mensagem como esta a partir do sistema de notícias, não se deve crer em tudo o que está descrito à primeira vista. Dependendo de quem enviou a mensagem `checkgroups`, pode-se perder alguns grupos ou mesmo hierarquias inteiras. Caso se entenda que os grupos listados como não localizados devem estar presentes no site local, pode-se usar o programa `addgroup`. Salve a lista de grupos não localizados e inicialize-os com o seguinte programa:

```
#!/bin/sh
cd /usr/lib/news

while read group; do
    if grep -si "^$group[[:space:]].*moderated" newsgroup; then
        mod=m
    else
        mod=y
    fi
    /usr/lib/news/bin/maint/addgroup $group $mod
done
```

17.8.4 `sendsys`, `version` e `senduuname`

Finalmente há três mensagens que podem ser usadas para se descobrir a topologia da rede. Elas são denominadas `sendsys`, `version` e `senduuname`. Elas fazem com que o C News indique ao remetente o arquivo `sys`, a versão do programa e a saída do comando `uuname(1)`, respectivamente. C News é bastante lacônico sobre a mensagem de controle `version`, retornando somente um "C".

Novamente, não se deve *nunca* enviar tal mensagem, a menos que se esteja seguro que ela não deixará a rede local. Respostas a uma mensagem `sendsys` podem rapidamente desativar uma rede UUCP.[11]

17.9 C News em Um Ambiente NFS

Uma forma simples de distribuir notícias em uma rede local é mantê-las todas em um servidor central e exportar os diretórios relevantes via NFS, possibilitando que os leitores de notícias possam pesquisar os arquivos de forma direta. A vantagem deste método sobre o NNTP é que a sobrecarga gerada pela transmissão do alto

[11] Também não deve ser tentada na Internet.

volume de notícias é reduzida significativamente. NNTP, por outro lado, tem vantagem quando se utiliza uma rede heterogênea onde os equipamentos sejam muito diferentes ou onde os usuários não tenham contas equivalentes na máquina servidora.

Ao se utilizar NFS, artigos postados em uma máquina local devem ser reenviados para o servidor, uma vez que o acesso a arquivos administrativos pode por outro lado expor o sistema a condições que gerem inconsistências. Adicionalmente, deve-se proteger a área de tarefas temporárias de notícias, exportando-a somente com permissões de leitura, a qual exigirá também uma sistemática de envio de notícias para o servidor.

O C News faz esta administração de forma transparente. Ao se postar um arquivo, o leitor de notícias normalmente aciona o programa `inews` para introduzir o artigo no sistema de notícias. Este comando executa diversas verificações no artigo, complementa o cabeçalho e verifica o arquivo `server` em `/usr/lib/news`. Caso este arquivo exista e contenha um nome de máquina diferente do sistema local, `inews` será acionado naquele servidor via `rsh`. Uma vez que o programa `inews` utiliza diversos arquivos binários e arquivos de suporte de C News, deve-se ter o programa C News instalado localmente ou montado via NFS a partir do servidor.

Para que o acionamento do comando `rsh` funcione adequadamente, cada usuário deve ter uma conta equivalente no sistema servidor, ou seja uma conta que possa acessar o servidor sem uso de senha.

Deve-se estar seguro de que o nome de máquina fornecido no arquivo `server` coincida literalmente com a saída do comando `hostname(1)` no equipamento servidor, ou o programa C News entrará em um círculo eterno de execução para poder entregar o artigo.

17.10 Tarefas e Ferramentas de Manutenção

Apesar da complexidade do C News, a vida de um administrador de notícias pode ser muito facilitada, já que o programa disponibiliza uma série de ferramentas de manutenção. Algumas destas ferramentas devem ser executadas regularmente através do `cron`, como por exemplo `newsdaily`. O uso destes programas reduz enormemente os cuidados diários e as demandas de recebimento e envio de mensagens.

A menos que tenha sido indicado de maneira diferente, estes comandos estão localizados no diretório `/usr/lib/news/bin/maint`. Note que é necessário tornar-se o usuário `news` antes da execução destes comandos. Executá-los como superusuário tornará os arquivos inacessíveis ao programa C News.

`newsdaily` Deve ser executado uma vez ao dia. É um programa importante, que ajuda a manter pequenos os arquivos de registros, mantendo cópias destes arquivos referentes às três últimas execuções. Ele ainda tenta monitorar algumas anomalias, como lotes perdidos em diretórios de envio ou remessa, postagens para grupos desconhecidos ou moderados, etc. As mensagens de erro resultantes são enviadas para o administrador.

`newswatch` Este programa deve ser executado regularmente para verificar possíveis anomalias no sistema de notícias, uma vez a cada hora ou mais. Ele visa detectar problemas que terão efeito imediato na operacionalização do sistema de notícias e envia mensagens dos problemas encontrados para o administrador do sistema. Entre os itens verificados estão arquivos de reserva de recursos não removidos, lotes de entrada sem tratamento, espaço em disco, etc.

`addgroup` Adiciona um grupo ao site local. A forma correta de acioná-lo é:

```
addgroup nome_do_grupo y|n|m|=nome_real
```

O segundo argumento tem o mesmo significado que o campo do arquivo `active`, definindo que qualquer usuário pode postar artigos para o grupo (`y`), que ninguém pode fazê-lo (`n`), ou que se trata de um arquivo moderado (`m`) e que este é na verdade um nome alternativo para o grupo (`=nome_real`).

Pode-se usar ainda o programa `addgroup` ao se receber artigos para um grupo, que cheguem antes do que a mensagem de controle `newgroup`, que tem a função de criação.

`delgroup` Permite a exclusão local de um grupo. Deve ser acionado da seguinte forma:

```
delgroup nome_do_grupo
```

Deve-se ainda excluir os artigos remanescentes do diretório de tarefas temporárias de notícias. Alternativamente, pode-se deixar que isso ocorra naturalmente, através da execução do programa `expire`.

`addmissing` Adiciona artigos faltantes ao arquivo `history`. Este programa deve ser executado quando artigos parecem ser mantidos eternamente.

`newsboot` Este programa deve ser executado durante a inicialização do sistema. Ele remove todos os arquivos de reserva de recursos deixados pelos processos encerrados em tempo de finalização do sistema, assim como encerra qualquer execução de lotes não concluída pelas conexões NNTP.

`newsrunning` Este programa reside em `/usr/lib/news/bin/input` e pode ser usado para desabilitar a abertura dos lotes recebidos, por exemplo durante o horário comercial. Pode-se desligar o desfazimento de lotes através do comando

```
/usr/lib/news/bin/input/newsrunning off
```

O restabelecimento da abertura dos lotes recebidos pode ser feito através da execução do mesmo programa, porém substituindo-se o parâmetro `off` por `on`.

Capítulo 18

Descrição do NNTP

18.1 Introdução

Dada às diferenças dos transportes de rede usados, NNTP provê uma abordagem bastante diferente para troca de mensagens em relação ao C news. NNTP significa *Protocolo de Transferência de Notícias em Rede*[1], e não se trata de um programa em particular, mas sim um padrão Internet.[2] Ele é baseado em uma conexão orientada a fluxo de dados – normalmente sobre TCP – entre um cliente em qualquer ponto da rede e um servidor em uma máquina que mantém as notícias armazenadas em disco. A conexão orientada por fluxo de dados permite que o cliente e o servidor negociem interativamente artigos e estes sejam transferidos praticamente sem tempo de espera, além de manter o nível de artigos repetidos em patamares bastante baixos, melhorando em muito o transporte de notícias e suplantando em muito as redes UUCP. Enquanto há alguns anos atrás não era incomum um artigo levar duas semanas ou mais para chegar ao último recanto da Usenet, hoje ele não tarda mais de dois dias e na Internet pode levar em média alguns minutos.

Diversos comandos permitem aos clientes recuperar, enviar e postar artigos. A diferença entre enviar e postar é que o último envolve artigos com informações incompletas de cabeçalho.[3] Uma recuperação de um artigo pode ser efetuada por

[1] Network News Transfer Protocol

[2] Formalmente especificado em RFC 977.

[3] Ao postar um artigo sobre NNTP, o servidor sempre adicionará ao menos um campo ao cabeçalho, denominado `Nntp-Posting-Host:`. Ele contém o nome da máquina cliente.

programas clientes de transferência de notícias, assim como por leitores de notícias. Isso torna o NNTP uma excelente ferramenta para prover acesso a notícias para diversos clientes em uma rede local sem os caminhos tortuosos às vezes necessários para se usar o NFS.

NNTP também provê formas ativa e passiva de transferência de notícias, coloquialmente chamadas de "enviando" e "recebendo". Enviando é basicamente um processo idêntico ao protocolo eutenho/meenvie do C news. O cliente oferece um artigo ao servidor através do comando "`IHAVE <varmsgid>`" e o servidor retorna um código indicando se ele já tem o artigo ou se deseja recebê-lo. Neste caso, o cliente envia o artigo, terminando a mensagem com um único ponto em uma linha à parte.

"Enviando notícias" tem a única desvantagem de gerar uma grande carga no sistema servidor, uma vez que ele deve pesquisar a base de históricos para cada artigo a ser recebido.

A técnica oposta é chamada de "Recebendo Notícias", na qual o cliente solicita uma lista de todos os artigos disponíveis de um grupo e que tenham chegado após uma determinada data. A pesquisa é executada pelo comando `NEWNEWS`. A partir da identificação das mensagens presente na lista enviada pelo servidor, o cliente seleciona os artigos que ainda não possua e usando o comando `ARTICLE` para cada um deles pode recuperá-los.

O problema com "recebendo notícias" é que se faz necessário um controle estreito pelo servidor sobre quais grupos e distribuições podem ser solicitados pelo cliente. Por exemplo, deve-se estar seguro de que material confidencial de grupos de notícias locais não seja enviado para o site de clientes não autorizados.

Existem ainda alguns comandos de auxílio para leitores de notícias que permitem a recuperação de partes separadas de artigos: o cabeçalho ou o corpo, ou mesmo somente algumas linhas do cabeçalho. Isso permite manter todas as notícias em uma máquina central, a qual pode ser utilizada por todos os usuários da rede que utilizem programas clientes baseados em NNTP para leitura e postagem. Isso funciona como uma alternativa à exportação de diretórios de notícias via NFS a qual é descrita no capítulo 17.

Um problema comum do NNTP reside no fato dele permitir que sites conhecidos insiram artigos no novo fluxo de transferência com uma especificação falsa do remetente. Isso é chamado de *falsificação de notícias*.[4] Uma extensão do NNTP

[4] O mesmo ocorre com o SMTP, Protocolo de Transferência Simples de Mensagens.

permite que seja exigida uma autenticação do usuário para uso de certos comandos.

Há diversos pacotes NNTP disponíveis. Um dos mais largamente conhecidos é o servidor NNTP, também conhecido como *implementação de referência*. Originalmente ele foi escrito por Stan Barber e Phil Lapsley para ilustrar os detalhes da RFC 977. A versão denominada `nntpd-1.5.11` será descrita a seguir. Pode-se ainda obter os fontes e compilá-los ou se utilizar o pacote binário `net-std` que contém o `nntpd` de Fred van Kempen. Nenhuma versão pronta para execução do `nntpd` é disponibilizada, uma vez que diversos parâmetros específicos dos sites devem ser compilados.

O pacote `nntpd` consiste de um servidor e dois clientes para receber e enviar notícias, respectivamente, assim como um substituto do `inews`. Eles subexistem em um ambiente Bnews, mas com alguns ajustes podem também ser executados em um ambiente C news. De qualquer forma, caso se deseje utilizar o NNTP para outras atividades diferentes da oferta de acesso de leitores de notícias ao servidor local de notícias, a implementação de referência não é realmente uma opção. Iremos discutir somente o servidor NNTP contido no pacote `nntpd` e não abordaremos os programas clientes.

Há ainda um pacote chamado "Notícias Internet"[5], ou INN em formato resumido, o qual foi escrito por Rich Salz. Ele provê tanto NNTP quanto transporte de notícias baseado em NNTP e é mais adequado a sites com grandes volumes de notícias. Quando se trata de transporte de notícias sobre NNTP, ele é definitivamente melhor que `nntpd`. INN está atualmente na versão `inn-2.2`. Há um kit para construção do INN para `Linux` desenvolvido por Arjan de Vet disponível em `metalab.unc.edu` no diretório `system/Mail`. Caso se deseje configurar o INN, por favor verifique a documentação que está disponível com os fontes, assim como o FAQ INN postado regularmente em `news.software.b`.

18.2 Instalando O Servidor NNTP

O servidor NNTP é chamado de `nntpd` e pode ser compilado de duas formas, dependendo da expectativa de carga de notícias no sistema. Não há nenhuma versão compilada disponível, uma vez que alguns padrões específicos do site são definidos no código do executável. Todas as configurações são feitas através de macros definidas em `common/conf.h`.

[5] Internet News

nntpd pode ser configurado ou como um servidor isolado que é acionado quando o sistema é inicializado a partir do arquivo rc.inet2, ou como um servidor administrado pelo inetd. Neste caso deve-se ter a seguinte entrada no arquivo /etc/inetd.conf:

```
nntp    stream  tcp nowait      news    /usr/etc/in.nntpd    nntpd
```

Caso se configure o programa nntpd como um servidor isolado, esteja certo que esta linha do arquivo inetd.conf esteja comentada. Neste caso, deve-se ter a seguinte linha no arquivo /etc/services:

```
nntp    119/tcp   readnews untp # Protocolo de Transferência de Notícias em Rede
```

Para temporariamente armazenar quaisquer artigos recebidos, o nntpd necessita ainda de um dire‘tório .tmp no diretório de tarefas temporárias do servidor de notícias. Ele pode ser criado da seguinte forma:

```
# mkdir /var/spool/news/.tmp
# chown news.news /var/spool/news/.tmp
```

18.3 Restringindo o Acesso ao NNTP

O acesso aos recursos do NNTP são definidos através do arquivo nntp_access no diretório /usr/lib/news. Linhas neste arquivo descrevem os direitos de acesso definidos para máquinas externas. Cada linha tem o seguinte formato:

site read|xfer|both|no post|no [!*exceptgroups*]

Caso um cliente conecte-se à uma porta NNTP, nntpd tenta obter o nome totalmente qualificado do domínio a partir de uma pesquisa reversa do endereço IP. O nome da máquina cliente e o endereço IP são checados contra o campo *site* na mesma ordem em que as entradas aparecem neste arquivo. Caso uma entrada coincida exatamente suas regras são aplicadas, caso seja parcial a coincidência, ela somente será aplicada se não houver outra entrada que sirva mais adiante no arquivo. *site* pode ser especificado de uma das seguintes formas:

`nome_da_máquina` Este é o nome totalmente qualificado da máquina. Caso este coincida literalmente com o nome canônico da máquina, a entrada será válida e todas as entradas seguintes serão ignoradas.

`Endereço IP` Este é o endereço IP no formato decimal. Caso o endereço IP do cliente coincida com este, a entrada será válida e todas as entradas seguintes serão ignoradas.

`nome de domínio` Este é o nome do domínio, especificado como `*.`*`domínio`*. Se o nome da máquina do cliente coincide com o nome de domínio, a entrada será considerada válida.

`nome_da_rede` Este é o nome da rede, conforme especificado no arquivo `/etc/networks`. Caso o número IP do cliente coincida com o endereço IP associado ao nome da rede, a entrada será considerada válida.

`default` A palavra chave `default` faz com que qualquer cliente seja considerado como válido.

Entradas com especificações de sites mais genéricas devem ser colocadas ao final, uma vez que entradas associadas e mais específicas seriam sobrepostas pelas primeiras.

O segundo e o terceiro campos descrevem os direitos de acessos fornecidos ao cliente. O segundo detalha as permissões para recuperação de notícias, permitindo a recepção de notícias através do parâmetro (`read`), e a transmissão de notícias através da especificação do parâmetro (`xfer`). Um valor igual a `both` habilita tanto a recepção como a transmissão, `no` inibe qualquer tipo de acesso. O terceiro campo fornece ao cliente o direito de postar artigos, ou seja, enviar artigos com o cabeçalho incompleto, o qual será completado pelo software de notícias. Caso o segundo campo contenha a expressão `no`, ele será ignorado.

O quarto campo é opcional, e contém uma lista, separada por vírgulas, dos grupos que o cliente não têm permissão de acesso.

Um exemplo do arquivo `nntp_access` é mostrado a seguir:

```
#
# por padrão, qqr. um pode transferir notícias, mas ninguém pode ler ou postar
default                 xfer            no
#
# noticias.cvirtual.com.br oferece acesso via modem, permitindo a leitura e
# postagem para todos os grupos menos local.
noticias.cvirtual.com.br        read            post    !local
```

```
#
# todas as outras máquinas da cervejaria podem ler e postar
*.cvirtual.com.br            read           post
```

18.4 Autorização NNTP

Ao colocar-se em letras maiúsculas os indicadores de acesso como `xfer` ou `read` no arquivo `nntp_acces`, `nntpd` requer uma autorização do cliente para executar a respectiva operação. Por exemplo, ao especificar a permissão de `Xfer` ou `XFER`, `nntpd` não permitirá que o cliente execute a transferência de artigos para o site local até que a autorização seja informada.

O procedimento de autorização é implementado através de um novo comando NNTP chamado `AUTHINFO`. Ao usar este comando, o cliente transmite um nome de usuário e uma senha para o servidor NNTP. `nntpd` irá validar os dados, confrontando-os com o arquivo `/etc/passwd` e verificando se o usuário pertence ao grupo `nntp`.

A implementação atual de autorização do NNTP é somente experimental e não foi implementada de uma forma muito portável. O resultado disso é o seu funcionamento somente através de arquivos de senhas em formato texto, sendo que senhas sombras não são suportadas.

18.5 `nntpd` Interação com C News

Ao receber um artigo, `nntpd` tem que entregá-lo ao subsistema de notícias. Dependendo se ele foi recebido como resultado de um comando `IHAVE` ou `POST`, o artigo é administrado por `rnews` ou `inews`, respectivamente. Ao invés de acionar o programa `rnews`, pode-se configurá-lo (em tempo de compilação) para lotear os artigos recebidos e mover o lote resultante para o arquivo `/var/spool/news/in.coming`, onde serão deixados para que o programa `relaynews` o coloque na próxima fila de execução.

Para executar adequadamente o protocolo `eutenho/meenvie`, o `nntpd` deve ter acesso ao arquivo `history`. Em tempo de compilação deve-se estar seguro de que o caminho esteja corretamente configurado. Deve-se observar ainda que C news e `nntpd` estão tratando o mesmo formato de arquivo de histórico. C news usa as funções de bancos de dados `dbm` para acessá-lo. De qualquer forma, há algumas implementações incompatíveis da biblioteca `dbm`. Caso C news seja construída com uma biblioteca diferente da definida como padrão para a `libc`, deve-se construir `nntpd` com esta biblioteca também.

Um sintoma típico de desacordo no formato da base de dados entre o `nntpd` e C news são as mensagens de erro no sistema de registros do `nntpd` comunicando que não se pode abrir o arquivo adequadamente, ou artigos repetidos sendo recebidos via NNTP. Um bom teste consiste em escolher um artigo a partir da área de tarefas temporárias, executar um comando `telnet` para a porta do `nntp` e oferecer o artigo para o programa `nntpd`, conforme mostrado a seguir (as entradas do usuário estão assinaladas *desta forma*). Obviamente, deve-se substituir a *<msg@id>* com a identificação da mensagem do artigo que se deseja enviar novamente para o `nntpd`.

```
$ telnet localhost nntp
Trying 127.0.0.1...
Connected to localhost
Escape characters is '^]'.
201 aracaju NNTP[auth] server version 1.5.11t (16 November 1999) ready at
Sun Feb 6 16:02:32 1199 (no posting)
IHAVE <msg@id>
435 Got it.
QUIT
```

esta conversação mostra a reação adequada do programa `nntpd` onde a mensagem "`Got it`" indica que ele já tem este artigo. Caso fosse recebida a mensagem "`335 Ok`", a pesquisa no arquivo de histórico teria falhado por alguma razão. Para finalizar a conversação basta digitar Ctrl-D. Pode-se verificar possíveis causas de erros através do sistema de registros do sistema, onde `nntpd` registra todos os tipos de mensagens através do servidor `syslog`. Uma biblioteca `dbm` incompatível normalmente se manifesta em uma mensagem de falha da função `dbminit`.

Capítulo 19

Configuração do Leitor de Notícias

Leitores de notícias buscam oferecer ao usuário funcionalidades que permitem um acesso facilitado ao sistema de notícias, possibilitando a postagem de arquivos ou a navegação pelos conteúdos dos grupos de notícias de uma forma confortável. A qualidade desta interface é objeto de intermináveis discussões.

Há alguns leitores de notícias disponíveis que foram portados para o `Linux`. A seguir descreveremos as funções básicas de configuração dos três mais populares, denominados `tin`, `trn` e `nn`.

Um dos mais efetivos leitores de notícias é:

```
$ find /var/spool/news -name '[0-9]*' -exec cat {} \; | more
```

Esta é a forma mais difícil de ler notícias no `Unix`.

A maioria dos leitores de notícias, no entanto, é muito mais sofisticada. Eles normalmente oferecem uma interface de tela cheia com diferentes níveis de apresentação dos grupos que o usuário subscreveu, além de mostrar uma visão geral dos artigos de um grupo ou de artigos individuais.

No nível de grupos de notícias, muitos leitores de notícias apresentam uma lista dos artigos disponíveis, mostrando o assunto e o autor. Em grupos grandes, é impossível o usuário manter controle dos artigos relacionados uns com os outros, apesar de ser possível identificar as respostas de artigos anteriores.

Uma resposta normalmente mantém o tema original do artigo, incorporando o campo "`Re: `". Adicionalmente a identificação do artigo é uma referência direta para o acompanhamento e pode ser fornecida no campo `References:` da linha de cabeçalho. Ordenar artigos por este critério normalmente gera pequenos grupos (na verdade árvores) de artigos, os quais são chamados de *temas*. Uma das principais tarefas ao se escrever um leitor de notícias é visualizar e implementar um esquema eficiente de busca e agrupamento por temas, uma vez que o tempo necessário para que isso ocorra é proporcional ao quadrado do número de artigos.

Não nos aprofundaremos muito mais neste tema de como a interface de usuário é construída. Todos os leitores de notícias do `Linux` têm boas funções de ajuda, para que o usuário não se sinta sozinho.

A seguir, lidaremos somente com tarefas administrativas. Muitas destas relacionadas com a criação de bancos de dados de temas e contabilização.

19.1 Configuração do Programa `tin`

O mais versátil leitor de notícias, no que diz respeito a temas, é o programa `tin`. Ele foi escrito por Iain Lea e foi modelado a partir de um antigo programa de leitura de notícias chamado `tass`.[1] Ele faz o ordenamento por tema assim que o usuário informa o grupo de notícias e é muito rápido, a menos que se esteja acessando os grupos via NNTP.

Em um 486DX50 ele leva cerca de 30 segundos para organizar 1000 artigos, quando eles são acessados diretamente do disco. Sobre uma conexão NNTP em um servidor de notícias com alto tráfego, isso pode levar cerca de 5 minutos.[2] Pode-se melhorar este tempo através da atualização regular do arquivo de índices com o uso da opção `-u`, ou acionando-se o programa `tin` com a opção `-U`.

Normalmente, `tin` registra as bases de dados de temas no diretório pessoal do usuário com o nomes de `.tin/index`. Isso pode de qualquer forma ter um custo em termos de recursos, caso não se queira manter uma única cópia em uma localização central. Isso pode ser obtido através do setuid para usuário `news` na execução do

[1] Escrito por Rich Skrenta.

[2] Este tempo pode ser melhorado drasticamente se o servidor NNTP não tiver que ordenar os artigos, deixando que o cliente recupere as bases de dados de temas. INN 1.4, por exemplo, funciona desta forma.

programa `tin`, por exemplo, ou para qualquer outro usuário sem privilégios.[3] `tin` irá manter todas as bases de dados de temas sob `/var/spool/news/.index`. Para o acesso a outros arquivos ou interpretadores de comando, ele irá retornar à efetiva identificação do usuário que acionou o programa.[4]

Uma solução mais adequada é instalar o servidor de indexação `tind` que é executado como um programa servidor e que regularmente atualiza os arquivos de índices. Este programa porém não é incluído em qualquer distribuição do `Linux`, devendo ser compilado pelo próprio usuário. Caso se esteja trabalhando em uma rede local com um servidor central de notícias, pode-se executar o programa `tind` no servidor e deixar que os clientes recuperem os arquivos de índices via NNTP. Atualizações para o `nntpd` que implementam estas extensões estão incluídas com os fontes do `tin`.

A versão do `tin` incluída em algumas distribuições `Linux` não tem suporte a NNTP pré compiladas, porém muitas já foram atualizadas. Ao se invocar o programa `rtin` ou usando-se a opção `-r`, `tin` tenta conectar-se a um servidor NNTP especificado no arquivo `/etc/nntpserver` ou na variável de ambiente **NNTPSERVER**. O arquivo `nntpserver` contém simplesmente o nome do servidor em uma única linha.

19.2 Configuração do Programa `trn`

`trn` também é o sucessor de um antigo leitor de notícias, chamado `rn` (que significa *leitor de notícias*[5]). O "t" no seu nome significa "temas"[6]. Foi escrito por Wayne Davidson.

Diferentemente de `tin`, `trn` não permite a criação de bases de dados de temas em tempo de execução. Ao invés disso, ele usa bases criadas pelo programa chamado `mthreads` que deve ser acionado regularmente pelo `cron` para atualizar os arquivos de índices.

Não executar `mthreads` não quer dizer que não se possa acessar novos artigos, significa somente que aqueles artigos "Intel investe no Linux" estarão dispersos no menu de seleção de artigos, ao invés de agrupados em um único tema que poderia

[3] De qualquer forma, *não* use `nobody` para isto. Como uma regra, nenhum arquivo ou comando pode ser associado com este usuário.

[4] Isso é causado pelo fato de se obter mensagens de erro ao acionar o programa como superusuário. Bem, de qualquer forma não se deve trabalhar como `root` todo o tempo.

[5] read news

[6] thread

ser mais facilmente tratado.

Para acionar o agrupamento por temas para um grupo de notícias em particular, `mthreads` é acionado com uma lista destes grupos na linha de comando. A lista deve ser feita exatamente da mesma forma que uma entrada do arquivo `sys`:

```
mthreads comp,rec,!rec.games.go
```

Isso irá habilitar o agrupamento para todos os grupos de `comp` e `rec`, exceto para `rec.games.go` (pessoas que jogam Go não necessitam de assuntos organizados). Após este procedimento basta acioná-lo sem qualquer opção para que ele organize todos os novos artigos. A organização de todos os grupos encontrados no arquivo `active` pode ser definida ao se acionar `mthreads` com a lista de grupos igual a `all`.

Caso se esteja recebendo notícias durante a noite, pode-se customizar `mthreads` para que seja executado uma vez pela manhã, por exemplo. Pode-se porém executá-lo tantas vezes quantas sejam necessárias. Sites que têm um tráfego muito intenso podem executar `mthreads` no modo servidor. Quando ele é acionado na inicialização do sistema, usando-se a opção `-d`, o programa entra em modo de execução em segundo plano e é automaticamente acionado a cada 10 minutos para checar se há novos artigos que devem ser organizados por tema. Para executar `mthreads` no modo servidor, basta colocar a seguinte linha no programa `rc.news`:

```
/usr/local/bin/rn/mthreads -deav
```

A opção `-a` faz com que `mthread` automaticamente acione a organização por tema de novos grupos, assim que eles sejam criados, a opção `-v` habilita o registro de mensagens de `mthreads` em seu arquivo de ocorrências denominado `mt.log` localizado no diretório onde `trn` foi instalado.

Artigos antigos não mais disponíveis devem ser removidos dos arquivos de índices regularmente. Por padrão, somente artigos cujos números estejam abaixo do menor número de identificação de artigos disponíveis serão removidos.[7] Artigos acima deste número que tenham de qualquer forma expirado (porque continham um valor já expirado no campo do cabeçalho `Expires:`), poderão ser removidos pelo programa `mthreads` através da opção `-e`, a qual força a execução de uma

[7] Note que o programa C news não atualiza este valor automaticamente. Deve-se executar o programa `updatemin` para fazê-lo. Por favor, verifique o capítulo 17.

verificação avançada de artigos expirados. Quando o programa `mthreads` é executado em modo servidor, a opção `-e` faz com que seja executada esta verificação avançada uma vez ao dia, logo após a meia-noite.

19.3 Configuração do Programa nn

`nn` foi escrito por Kim F.Storm e clama ser o leitor de notícias cujo principal objetivo não é disponibilizar artigos. Seu nome significa "Sem Notícias"[8] e sua chamada diz "Sem notícias é uma boa notícia. `nn` é melhor."

Para atingir este ambicioso objetivo, `nn` é distribuído com um conjunto de ferramentas de manutenção que permitem não somente a organização por temas, mas também checagens extensivas da consistência destas bases de dados, contabilidade, estatísticas de uso e restrições de acesso. Há também um programa de administração chamado `nnadmin`, o qual permite que sejam executadas diversas tarefas interativamente. É bastante intuitivo, então não entraremos em muitos detalhes sobre estes aspectos e focalizaremos a geração dos arquivos de índices.

O gerenciador de banco de dados de temas do programa `nn` é chamado `nnmaster`. Normalmente é executado como um programa servidor, inicializado pelos programas `rc.news` ou `rc.inet2`, sendo acionado da seguinte forma:

```
/usr/local/lib/nn/nnmaster -l -r -C
```

Isso habilita a organização por temas de todos os grupos de notícias presentes no arquivo `active`.

De forma equivalente, pode-se acionar o programa `nnmaster` periodicamente a partir do `cron`, fornecendo-se uma lista de grupos. Esta lista é muito similar à lista de subscrição do arquivo `sys`, exceto pelo fato de neste caso, os grupos serem separados por espaços em branco no lufar de vírgulas. Ao invés de um nome falso de grupo como `all`, um argumento vazio de `""` deve ser usado significando todos os grupos. Uma execução simples pode ser acionada da seguinte forma:

```
# /usr/local/lib/nn/nnmaster !rec.games.go rec comp
```

Note que a ordem é significativa nestes casos: o grupo mais à esquerda especifica os parâmetros que prevalecem sobre os demais. Ou seja se colocássemos o parâ-

[8] no news

metro `!rec.games.go` após `rec`, todos os artigos para este grupo seriam tratados, independentemente da definição posterior.

O programa `nn` oferece diversos métodos de remoção de artigos expirados das suas bases de dados. O primeiro é a atualização da base de dados através da pesquisa dos diretórios de grupos de notícias, descartando as entradas que correspondam a artigos não mais disponíveis. Esta é a operação padrão ao se acionar o programa `nnmaster` com a opção `-E`. Ela á relativamente rápida, a menos que esteja sendo realizada via NNTP.

O método 2 comporta-se exatamente como o padrão da execução da expiração do programa `mthreads`, na qual ele remove as entradas que se referem a artigos cujo número seja inferior ao artigo mais antigo ainda não expirado do arquivo `active`. Ele pode ser habilitado através da opção `-e`.

Finalmente, o método 3 consiste em descartar toda a base de dados e reconstruí-la totalmente. Isso pode ser feito através da opção `-E3` do programa `nnmaster`.

A lista de grupos a serem verificados pode ser fornecida através da opção `-F` no mesmo formato acima. De qualquer forma, caso se tenha o programa `nnmaster` sendo executado como servidor, deve-se finalizá-lo através da opção `-k`, antes da opção de verificação de expiração poder atuar e posteriormente ele deve ser reinicializado com as opções originais.

Este é o comando adequado para acionar a opção de verificação de expiração para todos os grupos usando-se o método 1:

```
# nnmaster -kF ""
# nnmaster -lrC
```

Há diversos outros indicadores que podem ser utilizados para refinar o comportamento do programa `nn`. Caso se deseje remover artigos ou resumos de artigos, sugerimos a leitura da página de manual do programa `nnmaster`.

O programa `nnmaster` baseia-se em um arquivo chamado `GROUPS`, o qual está localizado em uma área denominada `/usr/local/lib/nn`. Caso ele não exista inicialmente, então será criado. Para cada grupo de notícias, ele contém uma linha que começa com o nome do grupo, seguido opcionalmente por um indicador de data e hora e indicadores, os quais podem ser editados para se obter um certo comportamento personalizado para o grupo em questão. Não se pode alterar a ordem em que os grupos aparecem neste arquivo.[9] Os indicadores permitidos e seus

[9] Isso se deve ao fato de sua ordem ser idêntica às entradas do arquivo (binário) `MASTER`.

efeitos são detalhados também nas páginas de manual do programa `nnmaster`.

Apêndice A

Um Exemplo de Cabo Nulo de Impressora Para PLIP

Para se fazer um cabo nulo para uso em uma conexão PLIP, é necessário ter dois conectores de 25 pinos (também conhecidos como DB-25) e um cabo condutor tipo 11. O cabo pode ter no máximo 15 metros de comprimento.

Ao se olhar o conector pode-se verificar que ele tem pequenos números na base de cada pino, a partir do 1 no alto à esquerda (caso se esteja mantendo o lado maior para cima) até o 25 no canto inferior direito. Para montar um cabo nulo de impressora, deve-se conectar os seguintes pinos uns aos outros:

`D0`	2—15	`ERROR`
`D1`	3—13	`SLCT`
`D2`	4—12	`PAPOUT`
`D3`	5—10	`ACK`
`D4`	6—11	`BUSY`
`GROUND`	25—25	`GROUND`
`ERROR`	15— 2	`D0`
`SLCT`	13— 3	`D1`
`PAPOUT`	12— 4	`D2`
`ACK`	10— 5	`D3`
`BUSY`	11— 6	`D4`

Todos os demais pinos devem permanecer desconectados. Caso o cabo seja protegido, a cobertura deverá estar conectada à carcaça metálica do conector DB-25 e a nenhum outro ponto.

Apêndice B

Exemplo de Arquivos de Configuração do `smail`

Esta seção apresenta diversos arquivos de configuração para um site folha UUCP em uma rede local. Eles são baseados nos arquivos de exemplos incluídos na distribuição do programa `smail-3.1.28`. Apesar de todo o conteúdo do capítulo referente ao smail, é fortemente sugerida a leitura das páginas de manual do programa `smail(8)`, que descrevem estes arquivos detalhadamente (uma vez que se tenha entendido a idéia básica por trás dos arquivos de configuração `smail`, será uma leitura valiosa).

O primeiro arquivo a seguir mostra o conteúdo de `routers`, o qual descreve um conjunto de roteadores para `smail`. Quando `smail` tem que entregar uma mensagem para um determinado endereço, ele envia o endereço para todos os roteadores, até que um indique que possa lidar com ele, ou seja que pode encontrar uma máquina de destino em sua base de dados, seja no arquivo `paths`, no `/etc/hosts`, ou qualquer mecanismo de roteamento com o qual o roteador interaja.

Entradas nos arquivos de configuração do `smail` sempre começam com um único nome identificando o roteador, transporte ou diretor. Eles são seguidos por uma lista de atributos que definem o seu comportamento. Esta lista consiste da configuração de uma série de atributos globais, como por exemplo o *programa de controle de dispositivo* usado e de atributos privados que serão somente entendidos por programas específicos. Atributos são separados por vírgulas, enquanto os valores dos atributos globais e privados são separados por ponto e vírgula.

Para tornar estas distinções claras, assumiremos que se deseje manter dois arquivos separados de nomes alternativos de caminhos, um contendo as informações de roteamento do domínio e o segundo contendo informações de roteamento global, provavelmente geradas a partir de mapas UUCP. No programa `smail`, pode-se especificar dois roteadores no arquivo `routers`, ambos usando o programa de controle de dispositivos especificado no arquivo de caminhos alternativos. Este procurará os nomes das máquinas em uma base de dados. O nome do arquivo é esperado como sendo um atributo privado.

```
#
# base de dados de caminhos alternativos para roteamento intradomínio
domain_paths:
        driver=pathalias,         # procura a máquina no arquivo de caminhos
        transport=uux;            # caso coincida, utilizará UUCP
        file=paths/domain,        # arquivo é o  /usr/lib/smail/paths/domain
        proto=lsearch,            # arquivo fora de ordem (pesquisa linear)
        optional,                 # ignora caso o arquivo não exista
        required=cvirtual.com.br, # pesquisa som. as máquinas *.cvirtual.com.br

#
# base de dados de caminhos alternativos para roteamento extradomínio
world_paths:
        driver=pathalias,         # procura a máquina no arquivo de caminhos
        transport=uux;            # caso coincida, utilizará UUCP
        file=paths/world,         # arquivo é o /usr/lib/smail/paths/world
        proto=bsearch,            # arquivo está ordenado por sort(1)
        optional,                 # ignora caso o arquivo não exista
        -required,                # nenhum domínio será requerido
        domain=uucp,              # retira a expressão ".uucp" antes do envio
```

O segundo atributo global fornecido para cada um dos dois `roteadores` acima define o transporte que deve ser usado, caso o roteador encontre o endereço. Neste caso, a mensagem será entregue utilizando-se o transporte `uux`. Transportes são definidos no arquivo `transports`, o qual é explicado a seguir.

Pode-se ainda refinar a definição de qual transporte a mensagem utilizará ao se especificar um arquivo de método ao invés de um atributo de `transporte`. Arquivos de métodos contêm um mapa de máquinas de destinos e transportes associados.

Os arquivos `routers` a seguir, definem roteadores para uma rede local que efetuam pesquisas em uma biblioteca resolvedora. Em uma máquina Internet, por outro lado, pode-se utilizar um roteador que lide com registros MX. Deve-se de qualquer forma retirar-se o comentário do roteador alternativo `inet_bind` que usa o driver pré-construído em `smail`.

Em um ambiente que misture UUCP e TCP/IP, pode-se ter alguns problemas em se ter máquinas listadas no arquivo `/etc/hosts` que tem conexões SLIP ou PPP de forma ocasional. Normalmente será desejável enviar mensagens via UUCP. Para prevenir que o programa `inet_hosts` associe estas máquinas à uma conexão permanente, elas devem ser informadas no arquivo `paths/force`. Trata-se de outro arquivo no mesmo estilo do arquivo de caminhos alternativos e é consultado antes que `smail` consulte o resolvedor.

```
# Um exemplo do arquivo  /usr/lib/smail/routers
#
# force - força o uso do UUCP para certas máquinas, mesmo que elas estejam
# listadas no arquivo /etc/hosts
force:
        driver=pathalias,         # procura a máquina no arquivo de caminhos
        transport=uux;            # caso coincida, utilizará UUCP
        file=paths/force,         # o arquivo é  /usr/lib/smail/paths/force
        optional,                 # ignora caso o arquivo não exista
        proto=lsearch,            # arquivo não está ordenado(pesquisa linear)
        -required,                # domínios não são requeridos
        domain=uucp,              # retira a expressão ".uucp" antes do envio


# inet_addrs - compara literais como nomes contendo endereços IP, como por
# exemplo janete@[191.72.2.1]
inet_addrs:
        driver=gethostbyaddr,     # programa para comparar IPs e literais
        transport=smtp;           # utilizar SMTP sobre TCP/IP
        fail_if_error,            # falhar caso o endereço esteja mal formado
        check_for_local,          # entregar diretamente se a máquina de destino
                                  # for local

# inet_hosts - compara nomes de máquinas usando gethostbyname(3N)
# deve ser comentado caso se deseje usar a versão BIND.
inet_hosts:
        driver=gethostbyname,     # compara nomes com a função de biblioteca
        transport=smtp;           # usa SMTP padrão
        -required,                # nenhum domínio será requerido
        -domain,                  # sem sufixo de nome de domínio
        -only_local_domain,       # não se restringe aos domínios definidos

# inet_hosts - versão alternativa usando BIND para acesso ao DNS
#inet_hosts:
#       driver=bind,              # usa o programa BIND pré construído
#       transport=smtp;           # usa SMTP para entrega
#
#       defnames,                 # usa pesquisa padrão de domínios
#       defer_no_connect,         # tenta novamente se o servidos de nomes
#                                 # estiver inativo
#       -local_mx_okay,           # falhar (não passar através) de um MX para a
#                                 # máquina local.

#
# base de dados de caminhos para roteamento intradomínio.
domain_paths:
        driver=pathalias,         # pesquisa a máquina em um arquivo de caminhos
```

```
        transport=uux;            # caso coincida, utiliza o UUCP
        file=paths/domain,        # arquivo é /usr/lib/smail/paths/domain
        proto=lsearch,            # arquivo está desordenado (pesquisa linear)
        optional,                 # ignorar se o arquivo não existir
        required=cvirtual.com.br, # pesquisar somente máquinas *.cvirtual.com.br
#
# bases de dados de caminhos para roteamentos extradomínio
world_paths:
        driver=pathalias,         # pesquisa a máquina em um arquivo de caminhos
        transport=uux;            # caso coincida, utilizar o UUCP
        file=paths/world,         # arquivo é /usr/lib/smail/paths/world
        proto=bsearch,            # arquivo está ordenado com sort(1)
        optional,                 # ignorar se o arquivo não existir
        -required,                # domínios não são requeridos
        domain=uucp,              # retira a expressão ".uucp" antes do envio


# smart_host - uma especificação parcial da máquina de roteamento otimizado.
# Caso o atributo smart_path não esteja definido em /usr/lib/smail/config, este
# roteador será ignorado. O atributo transporte será sobreposto pela variável
# global smart_transport

smart_host:
        driver=smarthost,         # caso especial de roteamento
        transport=uux;            # por padrão utiliza UUCP
        -path,                    # usa a variável do arquivo de configuração
                                  # smart_path
```

Para lidar com mensagens para endereços locais é configurado o arquivo `directors`. Ele é constituído da mesma forma que o arquivo `routers`, com uma lista de entradas que definem uma forma de pesquisa cada uma delas. Estas entradas *não* definem a entrega da mensagem, mas sim o redirecionamento possível, por exemplo através de nomes alternativos, servidores de reenvio e assim por diante.

Ao entregar uma mensagem para um endereço local, como `didi`, `smail` passa o nome do usuário para todos os diretores, um a cada vez. Caso o nome seja localizado, o transporte especificado para o diretor será utilizado e a mensagem deverá ser entregue (por exemplo, no arquivo da caixa postal do usuário), ou será gerado um novo endereço (por exemplo, após o exame de um nome alternativo).

Devido aos aspectos de segurança envolvidos, diretores normalmente fazem uma série de checagens para verificarem se os arquivos utilizados podem estar com algum problema. Endereços obtidos de alguma forma dúbia (como a partir de um arquivo `aliases` que pode ser gravado por qualquer usuário) são assinalados como inseguros. Alguns programas de transporte poderão não utilizar estes endereços, por exemplo para entregar uma mensagem para um arquivo.

Além disso, `smail` também *associa um usuário* a cada endereço. Toda a operação de gravação e leitura é executada como se fosse realizada pelo usuário. Para entrega, por exemplo, na caixa postal de `janete`, o endereço é obviamente associado com o usuário `janete`. Outros endereços como aqueles obtidos a partir do arquivo `aliases`, tem outros usuários associados a eles, como por exemplo o usuário `nobody`.

Para maiores detalhes sobre estas funcionalidades, por favor visite a página de manual do programa `smail(8)`.

```
# Um exemplo do arquivo /usr/lib/smail/directors

# aliasinclude - expande endereços produzidos pela opção
#               ":include:nomedoarquivo" de arquivos de nomes alternativos
aliasinclude:
        driver=aliasinclude,      # caso especial de roteamento
        nobody;                   # acessa o arquivo como usuário nobody se for
                                  # inseguro
        copysecure,               # obtém as permissões do apelido diretor
        copyowners,               # obtém o dono do apelido diretor

# forwardinclude - expande endereços produzidos pela opção
#       ":include:nomedoarquivo" de arquivos produzidos por arquivos de reenvio
forwardinclude:
        driver=forwardinclude,    # caso especial de roteamento
        nobody;                   # acessa o arquivo como usuário nobody se for
                                  # inseguro
        checkpath,                # verifica se o caminho pode ser acessado
        copysecure,               # obtém as permissões do reenvio diretor
        copyowners,               # obtém o dono do reenvio diretor

# aliases - pesquisa por expansões armazenadas em um arquivo de nomes
#           alternativos
        driver=aliasfile,         # diretor de nomes alternativos de propósito
                                  # geral
        -nobody,                  # todos os endereços são associados ao usuário
                                  # nobody por padrão
        sender_okay,              # não remove o remetente durante a expansão
        owner=owner-$user;        # problemas vão para o dono do endereço
        file=/usr/lib/aliases,    # padrão: compatível com o sendmail
        modemask=002,             # não pode ser gravado por qualquer usuário
        optional,                 # ignora se o arquivo não existe
        proto=lsearch,            # arquivo ASCII não ordenado

# dotforward - expande arquivo .forward nos diretórios pessoais do usuários
dotforward:
        driver=aliasfile,         # diretor de reenvio de propósito geral
        owner=owner-$user;        # problemas vão para o dono do endereço
        nobody,                   # usa o usuário nobody, caso seja inseguro
        sender_okay,              # não remove o remetente durante a expansão
        file=~/.forward,          # verifica arquivo .forward no diretório
                                  # pessoal
        checkowner,               # o usuário pode ser dono deste arquivo
        owners=root,              # e o superusuário também
        modemask=002,             # não pode ser gravado por qualquer usuário
```

```
        caution=0-10:uucp:daemon, # não executar como root ou servidores

# deve-se ser muito cuidadoso com diretórios pessoais com possibilidade de
# acesso remoto
# inseguros:"~ftp:~uucp:~nuucp:/tmp:/usr/tmp",

# forwardto - expande o campo "Forward to " no arquivo de mensagem
#forwardto:
        driver=forwardfile,
        owner=Postmaster,         # erros irão para o Postmaster
        nobody,                   # usar o usuário nobody, caso seja inseguro
        sender_okay,              # não remove o remetente durante a expansão
        file=/var/spool/mail/${lc:user}, # localização da cxa. postal do usuário
        forwardto,                # habilita checagem do "Forward to "
        checkowner,               # o usuário pode ser dono deste arquivo
        owners=root,              # e o superusuário também
        modemask=002,             # não pode ser gravado por qualquer usuário
        caution=0-10:uucp:daemon, # não executar como root ou servidores

# user - verifica usuários na máquina local com entrega nas suas caixas postais
user:
        driver=user;              # programa para verificação de nomes de
                                  # usuários
        transport=local,          # usa transporte local

# real_user - verifica nomes de usuários quando prefixados pela expressão
#             "real-"
real_user:
        driver=user;              # programa para verificação de nomes de
                                  # usuários
        transport=local,          # usa transporte local
        prefix="real-",           # por exemplo, coincide real-janete

# lists - expande listas de destinatários sob /usr/lib/smail/lists
lists:  driver=forwardfile,
        caution,                  # indica todos os endereços com cuidado
        nobody,                   # e então associa ao usuário noboody
        sender_okay,              # não remove o remetente
        owner=owner-$user;        # o dono da lista
        file=lists/${lc:user},    # mapeia o nome da lista de destinatários para
                                  # letras minúsculas
```

Após rotear com sucesso ou direcionar uma mensagem, `smail` envia a mensagem para o transporte especificado pelo roteador ou diretor que coincidiu com as características de destino da mensagem. Estes transportes são definidos no arquivo `transports`. Novamente, um transporte é definido por opções globais ou privadas.

A mais importante definição de cada entrada é o programa que administra o transporte, por exemplo um programa `conector de comandos`, o qual aciona o comando específico através do atributo `cmd`. Além disso, há uma série de comandos que o transporte pode usar, que executam diversas transformações no cabeçalho da mensagem. O atributo `return_path`, por exemplo, faz com que o transporte insira uma campo de `caminho_retorno` na mensagem. O atributo `unix_from_hack` faz com que toda ocorrência da palavra `From` no início de uma linha seja precedido pelo caractere >.

```
# Um exemplo de arquivo /usr/lib/smail/transports

# local - entrega de mensagens para usuários locais
local:
        driver=appendfile,        # anexa a mensagem a um arquivo
        return_path,              # inclui o campo Return-Path:
        from,                     # fornece uma linha From_  ao envelope
        unix_from_hack,           # insere > antes de  From
        local;                    # usa as formas locais de entrega

        file=/var/spool/mail/${lc:user}, # localização dos arquivos de caixas
                                         # postais
        group=mail,               # grupo dono do arquivo para System V
        mode=0660,                # grupo mail pode acessar
        suffix="\n",              # adiciona uma nova linha

# pipe - entrega de mensagens para comandos
pipe:
        driver=pipe,              # envia a mensagem para outro programa
        return_path,              # inclui o campo Return-Path:
        from,                     # fornece uma linha From_  ao envelope
        unix_from_hack,           # insere > antes de  From
        local;                    # usa as formas locais de entrega

        cmd="/bin/sh -c $user",   # envia endereço para o  Bourne Shell
        parent_env,               # informação de ambiente do endereço pai
        pipe_as_user,             # usa identificação de usuário associada com o
                                  # endereço
        ignore_status,            # ignora um código de finalização diferente de
                                  # zero
        ignore_write_errors,      # ignora erros de gravação
        umask=0022,               # umask para processo filhos
        -log_output,              # não gera mensagens na saída padrão

# file - entrega mensagens para um arquivo
file:
        driver=appendfile,
        return_path,              # inclui o campo Return-Path:
        from,                     # fornece uma linha From_  ao envelope
        unix_from_hack,           # insere > antes de  From
        local;                    # usa as formas locais de entrega

        file=$user,               # arquivo é obtido de um endereço
        append_as_user,           # usa identificação de usuário associada com o
                                  # endereço
        expand_user,              # expande ~ e $ nos endereços
        suffix="\n",              # adiciona uma linha extra final
        mode=0600,                # configura permissões para 600

# uux - entrega mensagens via programa rmail em um site UUCP remoto
uux:    driver=pipe,
```

```
        uucp,                     # usa endereços no formato UUCP
        from,                     # suprime a linha de envelope From_
        max_addrs=5,              # no máximo 5 endereços a cada acionamento
        max_chars=200;            # no máximo 200 caracteres por endereço
        cmd="/usr/bin/uux - -r -a$sender -g$grade $host!rmail $(($user)$)",
        pipe_as_sender,           # mantêm o remetente nos registros UUCP
        log_output,               # salva as mensagens de erro de mensagens
                                  # devolvidas
#       defer_child_errors,       # tenta novamente se uux retorna algum erro

# demand - entrega para um programa rmail remoto, pesquisando imediatamente
        uucp,                     # usa endereços no formato UUCP
        from,                     # suprime a linha de envelope From_
        max_addrs=5,              # no máximo 5 endereços a cada acionamento
        max_chars=200;            # no máximo 200 caracteres por endereço
        cmd="/usr/bin/uux - -a$sender -g$grade $host!rmail $(($user)$)",
        pipe_as_sender,           # mantêm o remetente nos registros UUCP
        log_output,               # salva as mensagens de erro de mensagens
                                  # devolvidas
#       defer_child_errors,       # tenta novamente se uux retorna algum erro

# hbsmtp - usa lotes BSMTP. O arquivo de saída deve ser processado regularmente
#          e enviado via UUCP
hbsmtp:
        driver=appendfile,
        inet,                     # usa endereçamento RFC-822
        hbsmtp,                   # SMTP em lotes sem HELO e QUIT
        -max_addrs, -max_chars;   # sem limite no número de endereços

        file="/var/spool/smail/hbsmtp/$host",
        user=root,                # arquivo pertence ao superusuário
        mode=0600,                # somente pode ser lido ou gravado pelo
                                  # superusuário

# smtp - entrega utilizando SMTP sobre TCP/IP
smtp:
        driver=tcpsmtp,
        inet,
        -max_addrs, -max_chars;         # sem limite de número de endereços
        short_timeout=5m,               # tempo de espera para operações curtas
        long_timeout=2h,                # tempo de espera para operações longas
        service=smtp,                   # conectar a esta porta
# Para uso na Internet retirar os comentários das linhas abaixo
#       use_bind,                       # resolver MX e múltiplos registros A
#       defnames,                       # usar pesquisa padrão de nomes de
#                                       # domínios
#       defer_no_connect,               # tentar novamente se o servidor de
#                                       # nomes não estiver respondendo
#       -local_mx_okay,                 # falhar no MX para a máquina local
```

Apêndice C

COMO FAZER - DNS

C.1 Preâmbulo

Palavras chaves: DNS, bind, bind-4, bind-8, servidor de nomes, discagem, ppp, slip, isdn, Internet, domínio, nome de máquina, máquinas, resolução, named.

C.1.1 Aspectos Legais

C.1.2 Créditos e Pedidos de Ajuda

Gostaria de agradecer a Arnt Gulbrandsen, que leu o rascunho deste trabalho inúmeras vezes e forneceu inúmeras sugestões úteis. Gostaria de agradecer ainda às pessoas que têm mandado sugestões e notas via email.

Este nunca será um documento acabado dado à multiplicidade de detalhes do assunto, então por favor envie-me mensagens descrevendo seus problemas e seus sucessos e isto pode tornar este COMO FAZER melhor. Então por favor, ao enviar dinheiro, comentários e/ou perguntas, escreva para `janl@math.uio.no`. Ao enviar uma mensagem e, caso espere uma resposta, por favor seja cortês *certificando-se* que o endereço de retorno está correto e funcional. Também, **por favor** leia a seção C.8 de Perguntas e Respostas antes de enviar uma mensagem.

C.1.3 Dedicatória

Este COMO FAZER é dedicado a Anne Line Norheim Langfeldt. Embora ela provavelmente nunca o lerá, pois afinal ela não é este tipo de garota.

C.2 Introdução.

O que é e o que não é

Para iniciantes, DNS é o Servidor de Nomes do Domínio. O DNS converte os nomes das máquinas para números IP, que são os endereços das máquinas, mapeando de nome para endereço e de endereço para nome. Este COMO FAZER documenta como definir tais mapeamentos usando o sistema Linux. Um mapeamento é simplesmente uma associação entre duas informações, neste caso um nome de máquina, como `ftp.linux.org`, e o número IP da máquina, como por exemplo 199.249.150.4.

DNS é, para os não iniciados (você), uma das áreas mais opacas da administração de rede. Este COMO FAZER tentará clarificar alguns conceitos e aspectos sobre este tema. Ele descreve como configurar um nome do servidor DNS *simples*. Começando com um único servidor de cache e seguindo até a configuração de um servidor primário DNS para um domínio. Para configurações mais complexas pode-se checar a seção C.8 (QnA) de Perguntas e Respostas deste documento. Caso não esteja lá descrito, pode ser necessário *ler* a documentação que acompanha os fontes. Esclareceremos em que consiste esta documentação no C.9 (última seção).

Antes de começar, há que se configurar uma máquina para que ela possa se conectar interna e externamente e assim permitir as conexões à rede. Deve ser possível executar o comando `telnet 127.0.0.1` e ter acesso à máquina local (teste agora!). É necessário ainda ter-se arquivos de exemplo `/etc/nsswitch.conf` (ou `/etc/host.conf`), `/etc/resolv.conf` e`/etc/hosts` como ponto de partida, uma vez que não explicaremos aqui a sua função. Caso ainda não se tenha tudo isso configurado e operando, o documento NET-3 ou o COMO FAZER PPP explicam como configurá-los.

Ao nos referirmos a "máquina local", estamos referenciando à máquina na qual se está tentando configurar o DNS e não a qualquer outra máquina que se possa ter à disposição e que esteja conectada à rede.

Presumimos que esta máquina não está atrás de algum firewall que bloqueie as pesquisas de nomes. Caso seja necessária alguma configuração especial, por favor veja a seção C.8 (Perguntas e Respostas).

O serviço de nomes no Unix é feito por um programa servidor denominado `named`. Ele é integrante do pacote de bind que é coordenado por Paul Vixie para o Consórcio de Programas Para a Internet. O `Servidor de nomes` está incluído na maioria das distribuições Linux e é usualmente instalado como `/usr/sbin/named`. Caso se tenha um named à disposição pode-se usá-lo; caso contrário é possível obter-se um binário a partir de um site ftp Linux, ou conseguir os fontes mais recentes em `ftp.isc.org:/isc/bind/src/cur/bind-8/`. Este COMO FAZER trata sobre o bind em sua versão 8. A versão antiga do COMO FAZER, que tratava sobre o bind 4, ainda está ainda disponível em `http://www.math.uio.no/~janl/DNS/` no caso de necessitar utilizar o bind 4. Caso a página do manual sobre servidor de nomes fale sobre `named.conf` então tem-se disponível o bind 8, caso mencione o `named.boot` então trata-se do bind 4. Caso se tenha o 4 e se esteja com problemas de segurança, deve-se atualizar para a versão 8 mais recente.

O DNS é um banco de dados distribuído por toda a rede. É necessário ter-se extremo cuidado com tudo o que for colocado nele. Ao se colocar dados sem significado, outros utilizarão estes dados e certamente tudo ficará um pouco “estranho”. O DNS deve estar sempre atualizado e arrumado, evitando-se assim problemas desagradáveis. Deve-se aprender a usá-lo, administrá-lo, depurá-lo para tornar-se bom administrador da rede, evitando sobrecargas geradas por problemas de administração.

Neste documento é afirmado categoricamente algumas coisas que não são completamente verdadeiras (sendo então pelo menos meias verdades). Tudo em nome da simplificação. As coisas (provavelmente!) funcionarão quando o leitor acreditar no que está dito!

Dica: Devem ser feitas cópias de segurança de todos os arquivos. É aconselhável, ainda, que elas sejam alteradas de tempos em tempos. Assim se depois de todas as tentativas, nada funcionar, pode-se retornar à situação anterior.

C.3 Um Servidor de Nomes Somente Para Cache

Uma primeira aproximação à configuração do DNS, que pode ser muito útil para usuários que utilizam linhas discadas.

Um servidor de nomes somente para cache deve ser capaz de encontrar as respostas às pesquisas de nomes e endereços e deve ainda guardar as respostas, para a próxima em que sejam necessárias. Isto diminuirá o tempo de espera significativamente, especialmente quando se tem uma conexão lenta.

Inicialmente é necessário ter-se um arquivo `/etc/named.conf`, o qual será lido quando o servidor de nomes for inicializado. Por enquanto ele pode conter simplesmente:

```
// Configuração  do arquivo para um servidor de nomes
// somente para cache


opções {
        directory "/var/named";

        // Não comentar isto pode ajudar caso se tenha um firewall presente
        // e as coisas não estejam funcionando:

        // endereço de pesquisa: porta 53;
};

zone "." {
        type hint;
        file "rott.hints ";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

A linha "`directory`" indica onde os arquivos devem estar localizados. Todos os arquivos subseqüentes serão relativos a este. Assim `pz` é um diretório sob `/var/named`, ou seja estará localizado em `/var/named/pz`. `/var/named` é o diretório definido pelo *Padrão de Sistemas de Arquivos Linux*.

O arquivo denominado /var/named/root.hints deve conter:

```
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

Este arquivo descreve o nome dos servidores raiz no mundo. Este conteúdo pode mudar com o passar do tempo e *tem que* ser atualizado permanentemente. Veja a C.6 (seção de manutenção) para saber como mantê-lo atualizado.

A próxima seção em named.conf é a zona. Explicaremos o seu uso num capítulo adiante. Por hora somente façamos deste um arquivo chamado 127.0.0 no subdiretório pz:

```
@               IN      SOA     ns.linux.bogus.hostmaster.linux.bogus. (
```

```
                                1       ; Serial
                                8H        ; Atualização
                                2H      ; Tentativas
                                1W        ; Expiração
                                1D)       ; TTL mínimo
                NS      ns.linux.bogus.
1               PTR     localhost
```

Em seguida, será necessário um arquivo /etc/resolv.conf com o seguinte conteúdo:

```
search subdomínio.seu\_domínio.edu.br seu\_domínio.edu.br
nome\_do\_servidor 127.0.0.1
```

A linha "search" especifica que o domínio deve ser pesquisado para qualquer nome de máquina com a qual se queira conectar. A linha "nameserver" especifica o endereço do servidor de nomes. Neste caso, a própria máquina, uma vez que é nela que o programa named é executado (já que 127.0.0.1 foi informado, não importando se a máquina tem também outro endereço). Caso se queira indicar vários servidores de nomes, deve-se criar uma linha "nameserver" para cada um deles. (Nota: O programa named nunca lê este arquivo, e sim o resolvedor que utilizar o named).

Vamos ilustrar um pouco mais a função deste arquivo: caso um cliente tente procurar por `itamaraca`, então `itamaraca.subdomínio.seu_domínio.edu.br` será a primeira tentativa, então será tentado `itamaraca.seu_domínio.edu.br` e finalmente somente `itamaraca`. Se um cliente tentar procurar `metalab.unc.edu`, `metalab.unc.edu.subdomínio.seu_domínio.edu.br` será tentado inicialmente (sim, não faz muito sentido, mas é o jeito que ele funciona), então `metalab.unc.edu.seu_domínio.edu.br`, e finalmente `metalab.unc.edu`. Caso se queira colocar muitos domínios na linha search, isso pode provocar uma sobrecarga nos tempos de pesquisa.

O exemplo presume que a máquina pertence ao domínio `subdomínio.seu_domínio.edu.br`, sendo provavelmente o servidor de nomes `nome_da_máquina.subdomínio.seu_domínio.edu.br`. A linha de busca não deve conter o TLD (Domínio Raiz "edu.br" neste caso). Caso seja necessário conectar-se com freqüência a máquinas de outros domínios, deve-se acrescentar aqueles domínios à linha de busca, como por exemplo:

```
search subdomínio.seu_domínio.edu.br seu_domínio.edu.br outro_domínio.com.br
```

e assim por diante. Obviamente deve-se utilizar nomes reais de domínios. Os aqui colocados servem somente como exemplos. Por favor atente para a falta de pontos no final dos nomes dos domínios.

A seguir, dependendo da versão da biblioteca libc, tanto pode ser necessário atualizar o `/etc/nsswitch.conf` ou o `/etc/host.conf`. Caso se tenha o `nsswitch.conf` este será utilizado, caso contrário, atualizaremos o `host.conf`.

/etc/nsswitch.conf

Este é um arquivo longo que especifica onde podem ser obtidos diferentes tipos de dados, de que arquivos e de qual base de dados. Usualmente contém comentários úteis no topo, que podem ser lidos agora. Depois disso, deve ser encontrada uma linha que comece com "hosts:", onde se pode ler:

```
hosts:      files dns
```

Ccaso não haja nenhuma linha iniciada com "hosts:" então deve ser incluída a linha acima. Ela indica que os programas devem primeiramente pesquisar o arquivo `/etc/hosts`, e após então verificar o DNS de acordo com o configurado no arquivo `resolv.conf`.

/etc/host.conf

Provavelmente contém várias linhas, uma delas deve começar com `order` e deve ter o seguinte aspecto:

```
order hosts, bind
```

caso não haja nenhuma linha "order" , uma deve ser criada. Ela indica que a resolução de nomes de máquinas deve pesquisar inicialmente no arquivo `/etc/hosts`, e após pesquisar junto ao servidor de nomes (definido em `resolv.conf` como 127.0.0.1). Estes dois últimos arquivos estão documentados na página de manual on-line do utilitário resolver(8) (para acessá-la execute "man resolv") na maioria das distribuições Linux. Aquela página do manual é clara e em nossa opinião, todos, especialmente os administradores de DNS, devem lê-la. Faça-o agora, caso você seja daqueles que diz para si mesmo "Eu vou ler mais tarde" e nunca o faz.

C.3.1 Iniciando o named

Após tudo isto é hora de iniciar o servidor de nomes. Caso se esteja usando uma conexão discada, primeiro deve-se estabelecer a conexão. Deve-se digitar então "ndc start", sem opções. Caso isto não funcione, pode-se tentar "/usr/sbin/ndc start". Caso isto não funcione, deve-se verificar a seção C.8 (Perguntas e Respostas). Agora é possível testar a configuração. Ao se visualizar o arquivo de mensagens syslog (usualmente chamado `/var/adm/messages`; podem ser examinados também o diretório `/var/log` e o arquivo `syslog`) ao se iniciar o servidor de nomes (executando-se "tail -f/var/log/messages") deve-se obter algo como:

(linhas terminadas em \ continuam na linha seguinte)

```
Feb 15 01:26:17 roke named[6091]: starting.  named 8.1.1 Sat Feb 14 \
  00:18:20 MET 1998 ^Ijanl@roke.uio.no:/var/tmp/bind-8.1.1/src/bin/named
Feb 15 01:26:17 roke named[6091]: cache zone "" (IN) loaded (serial 0)
Feb 15 01:26:17 roke named[6091]: master zone "0.0.127.in-addr.arpa" \
  (IN) loaded (serial 1)
Feb 15 01:26:17 roke named[6091]: listening [127.0.0.1].53 (lo)
Feb 15 01:26:17 roke named[6091]: listening [129.240.230.92].53 (ippp0)
Feb 15 01:26:17 roke named[6091]: Forwarding source address  \
                                  is [0.0.0.0].1040
Feb 15 01:26:17 roke named[6092]: Ready to answer queries.
```

Se houver alguma mensagem de erro, ela deve ser examinada. O named indicará o arquivo onde o problema se encontra (ou `named.conf.` ou `root.hints`, esperamos `:-)`). O servidor de nomes deve ser finalizado e os arquivos devem ser corrigidos.

Agora é hora de iniciar o nslookup para examinar o trabalho realizado até aqui.

```
$ nslookup
default Server:  localhost
Address:  127.0.0.1

>
```

Caso este seja o resultado obtido, parabéns, está funcionando. Esperamos que sim. Caso se obtenha um resultado diferente, deve-se retornar e verificar todos os passos. Cada vez que se altere o arquivo `named.conf` será necessário reiniciar o servidor de nomes usando o comando `ndc restart`.

Agora podemos fazer pesquisas no sistema. Podemos procurar por alguma máquina próxima; A `pat.uio.no` está próxima a mim na Universidade de Oslo:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Name:    pat.uio.no
Address:  129.240.130.16
```

nslookup agora perguntou ao seu servidor de nomes para procurar a máquina `pat.uio.no`. Este contatou uma dos servidores de nomes listados no arquivo `root.hints`, e perguntou a um deles qual o caminho para a máquina desejada. Pode levar bem pouco tempo antes de se obter o resultado, enquanto named procura todos os domínios definidos em `/etc/resolv.conf`.

Ao se pesquisar novamente, tem-se:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address:  129.240.2.50
```

Note a linha "Non-authoritative answer:" que obtivemos desta vez. Isto indica que o servidor de nomes não saiu pela rede para perguntar sobre a máquina desejada. Ao invés disto procurou em seu cache e encontrou-o lá. Mas a informação do cache *pode* estar desatualizada (antiga). Então se está informado deste perigo (muito pequeno) quando o sistema informa "resposta Não autorizada:". Quando `nslookup` disser isto pela segunda vez para a mesma máquina, pode-se estar certo de que o cache está funcionando e fornecendo a informação certa. Pode sair-se do comando `nslookup` digitando-se "exit".

Agora que sabemos como configurar um servidor de nomes de cache, aproveite para tomar uma cerveja, leite, ou qualquer coisa que se queira para comemorar este fato memorável.

C.4 Um Domínio *Simples*.

Como configurar um domínio próprio.

C.4.1 Mas primeiro um pouco de teoria

Antes de *realmente* começarmos esta seção, forneceremos alguns ensinamentos sobre o funcionamento do DNS; é preciso lê-los porque é fundamental para um administrador de rede. Caso não se queira, deve-se pelo menos pesquisá-los rapidamente, até chegar aonde se quer ir no arquivo `named.conf`.

DNS é um sistema hierárquico. O mais alto nível é representado por "." e denominado "raiz". Sob "." há diversos Domínios de Alto Nível (TLDs), sendo os mais conhecidos ORG, COM, EDU e NET, mas existem muitos mais.

Ao se procurar uma máquina, a pesquisa ocorre recursivamente dentro da hierarquia, começando no topo. Caso se queira descobrir o endereço de `prep.ai.mit.edu`, o servidor de nomes local tem que encontrar um nome de servidor que responda pelo domínio edu. Ele pergunta a um servidor "." (ele já conhece os servidores ".", a partir do arquivo `root.hints`), e o servidor "." fornecerá uma lista dos servidores do domínio edu:

```
$ nslookup
Default Server: localhost
Address:  127.0.0.1
```

Começaremos perguntando por um servidor raiz:

```
> server c.root -servers.net.
Default Server:  c.root -servers.net
Address:  192.33.4.12
```

A seguir definiremos o tipo de pesquisa que desejamos fazer. Neste caso NS (registros de servidores de nomes):

```
> set q=ns
```

A seguir perguntaremos pelos servidores que respondem pelo domínio edu:

```
> edu.
```

O ponto após edu é significativo. Ele indica ao servidor que estamos pesquisando os servidores sob os quais o domínio edu está configurado (isto de alguma maneira simplifica a busca):

```
edu     nome do servidor = A.ROOT-SERVERS.NET
edu     nome do servidor = H.ROOT-SERVERS.NET
edu     nome do servidor = B.ROOT-SERVERS.NET
edu     nome do servidor  = C.ROOT-SERVERS.NET
edu     nome do servidor = D.ROOT-SERVERS.NET
edu     nome do servidor = E.ROOT-SERVERS.NET
edu     nome do servidor = I.ROOT-SERVERS.NET
edu     nome do servidor = F.ROOT-SERVERS.NET
edu     nome do servidor = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      endereço na internet = 198.41.0.4
H.ROOT-SERVERS.NET      endereço na internet = 128.63.2.53
B.ROOT-SERVERS.NET      endereço na internet = 128.9.0.107
C.ROOT-SERVERS.NET      endereço na internet = 192.33.4.12
D.ROOT-SERVERS.NET      endereço na internet = 128.8.10.90
E.ROOT-SERVERS.NET      endereço na internet = 192.203.230.10
I.ROOT-SERVERS.NET      endereço na internet = 192.36.148.17
F.ROOT-SERVERS.NET      endereço na internet = 192.5.5.241
G.ROOT-SERVERS.NET      endereço na internet = 192.112.36.4
```

A resposta nos indica que `*.root-servers.net` serve `edu.`, podemos então continuar perguntando, por exemplo ao servidor C.ROOT-SERVERS.NET. Agora queremos saber quem serve o próximo nível do nome da máquina: `mit.edu.`:

```
> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12

Non-authoritative answer:
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
mit.edu nameserver = STRAWB.mit.edu

Authoritative answers can be found from:
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
STRAWB.mit.edu  internet address = 18.71.0.151
```

A resposta indica que `strawb`, `w20ns` e `bitsy` servem o domínio `mit`. Vamos selecionar um deles e perguntar-lhe sobre `ai.mit.edu`:

```
> servidor W20NS.mit.edu.
```

Os nomes das máquinas não são sensíveis a maiúsculas e minúsculas, mas sugerimos o uso do mouse para cortar e colar como estão na tela.

```
Servidor:  W20NS.mit.edu
Endereço:  18.70.0.160
> ai.mit.edu.
Server:  W20NS.mit.edu
Address:  18.70.0.160

Non-authoritative answer:
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU
ai.mit.edu      nameserver = LIFE.AI.MIT.EDU
ai.mit.edu      nameserver = BEET-CHEX.AI.MIT.EDU
ai.mit.edu      nameserver = MINI-WHEATS.AI.MIT.EDU
ai.mit.edu      nameserver = COUNT-CHOCULA.AI.MIT.EDU
ai.mit.edu      nameserver = MINTAKA.LCS.MIT.EDU

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
AI.MIT.EDU      nameserver = LIFE.AI.MIT.EDU
AI.MIT.EDU      nameserver = BEET-CHEX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINI-WHEATS.AI.MIT.EDU
AI.MIT.EDU      nameserver = COUNT-CHOCULA.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINTAKA.LCS.MIT.EDU
ALPHA-BITS.AI.MIT.EDU           internet address = 128.52.32.5
GRAPE-NUTS.AI.MIT.EDU           internet address = 128.52.36.4
TRIX.AI.MIT.EDU                 internet address = 128.52.37.6
MUESLI.AI.MIT.EDU               internet address = 128.52.39.7
LIFE.AI.MIT.EDU                 internet address = 128.52.32.80
BEET-CHEX.AI.MIT.EDU            internet address = 128.52.32.22
MINI-WHEATS.AI.MIT.EDU          internet address = 128.52.54.11
COUNT-CHOCULA.AI.MIT.EDU        internet address = 128.52.38.22
MINTAKA.LCS.MIT.EDU             internet address = 18.26.0.36
```

Desta forma, obtemos que `museli.ai.mit.edu` é um dos servidores de nomes de `ai.mit.edu`:

```
> server MUESLI.AI.MIT.EDU
Default Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7
```

Agora mudaremos o tipo de pergunta. Já que encontramos o servidor de nomes, agora podemos perguntar tudo o que quisermos sobre `prep.ai.mit.edu`.

```
> set q=any
> prep.ai.mit.edu.
Server:  MUESLI.AI.MIT.EDU
Address:  128.52.39.7

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
        inet address = 18.159.0.42, protocol = tcp
          ftp  telnet  smtp  finger
prep.ai.mit.edu preference = 1, mail exchanger = gnu-life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = beet-chex.ai.mit.edu
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
gnu-life.ai.mit.edu             internet address = 128.52.32.60
beet-chex.ai.mit.edu            internet address = 128.52.32.22
alpha-bits.ai.mit.edu           internet address = 128.52.32.5
mini-wheats.ai.mit.edu          internet address = 128.52.54.11
trix.ai.mit.edu                 internet address = 128.52.37.6
muesli.ai.mit.edu               internet address = 128.52.39.7
count-chocula.ai.mit.edu        internet address = 128.52.38.22
mintaka.lcs.mit.edu             internet address = 18.26.0.36
life.ai.mit.edu                 internet address = 128.52.32.80
```

Assim começando por "." fomos capazes de descobrir os nomes dos servidores do próximo nível de domínio. Caso se esteja usando um servidor DNS próprio ao invés de usar todos aqueles outros servidores, o named certamente guardaria no cache todas as informações que tenha encontrado, não sendo necessária toda esta pesquisa na próxima vez que fosse realizada uma nova pesquisa de localização desta máquina.

Um tema muito menos comentado, mas também muito importante é `in-addr.arpa`. Ele também está aninhado como um domínio "normal". in-addr.arpa permite-nos conseguir os nomes das máquinas através de seus endereços. Uma coisa importante aqui, é notar que ip#s são escritos ao contrário no campo in-addr.arpa. Caso se tenha o endereço da máquina: 192.128.52.43, named procederá exatamente como no exemplo prep.ai.mit.edu: encontrar servidores `arpa.`, `in-addr.arpa.`, `192.in-addr.arpa.`, `128.192.in-addr.arpa.`, `52.128.192.in-addr.arpa.`. Encontrar então os registros necessários para `43.52.128.192.in-addr.arpa.` Engenhoso não? (Diga 'Sim', por favor!.) Porém não se preocupe endereços reversos somente são confusos nos dois primeiros anos.

Acabamos de contar uma mentira. O DNS não funciona exatamente da maneira aqui descrita. Mas não tenha dúvida que é muito próximo disso.

C.4.2 Nosso Próprio Domínio

Agora vamos definir nosso próprio campo. Vamos criar o domínio *linux.bogus* e definir suas máquinas. Usaremos o nome de domínio bogus para estarmos certos de não estarmos perturbando ninguém.

Mais uma coisa antes de começarmos: nem todos os caracteres são permitidos nos nomes das máquinas. Estamos limitados aos caracteres do alfabeto: a-z e aos números: 0-9, além do caractere "-" (hífen). Devemos nos restringir àqueles caracteres. Os caracteres maiúsculos e minúsculos são idênticos para o DNS, assim `pat.uio.no` é igual a `Pat.UiO.No`.

Começaremos esta parte com uma linha em `named.conf`:

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "pz/127.0.0";
};
```

Por favor note a falta de "." no final dos nomes dos campos neste arquivo. Isto nos diz que podemos definir uma zona `0.0.127.in-addr.arpa`, na qual somos os servidores principais e que as informações estão guardadas em um arquivo chamado `pz/127.0.0`. Nós já configuramos este arquivo anteriormente com o seguinte conteúdo:

```
@               IN      SOA     ns.linux.bogus.hostmaster.linux.bogus. (
                                1     ; Serial
                                8H      ; Atualização
                                2H    ; Tentativas
                                1W      ; Expiração
                                1D)     ; TTL mínimo
                NS      ns.linux.bogus.
1               PTR     localhost
```

Por favor note o "." no final de todos os nomes completos de campo neste arquivo, em contraste ao arquivo acima `named.conf`. Algumas pessoas gostam de começar cada arquivo de zona com uma diretiva `$ORIGIN`, mas isto é supérfluo. A origem (onde pertence o DNS na hierarquia) de um arquivo de zona é especificado na seção de zona do arquivo `named.conf`, a qual neste caso é `0.0.127.in-addr.arpa`.

Este "arquivo de zona" contém 3 "registros de recursos" (RRs): SOA, NS e um PTR. SOA é a contração para Início de Autoridade. O "@" é uma observação especial que significa origem e desde que a coluna do campo para este arquivo diz 0.0.127.in-addr.arpa, a primeira linha realmente quer dizer

```
0.0.127.in-addr.arpa.   IN      SOA ...
```

NS é o nome do servidor RR. Não há '@' no início desta linha, está *implícito* desde que a última linha começou com o caractere '@'. Economiza-se assim alguma digitação e a possibilidade de cometer algum erro. Assim na linha NS se lê

```
0.0.127.in-addr.arpa.   IN      NS      ns.linux.bogus
```

Indicando ao DNS que a máquina é o servidor de nomes do domínio `0.0.127.in-addr.arpa` é chamada `ns.linux.bogus`. 'ns' é um nome comum para servidor de nomes, mas como em servidores web são costumeiramente chamados `www.`*domínio*, este nome pode ser qualquer coisa.

E finalmente o registro PTR diz que a máquina no endereço 1 na sub-rede `0.0.127.in-addr.arpa`, ou seja, 127.0.0,1 é denominado `localhost`.

O registro SOA é o preâmbulo para *todos* os arquivos de zona e deve haver exatamente um em cada arquivo de zona, devendo necessariamente ser o primeiro registro. Ele descreve a zona, sua origem (uma máquina servidor de nomes `ns.linux.bogus`), quem é a responsável por seu conteúdo (`hostmaster@linux.bogus`), qual a versão do arquivo de zona (serial: 1) e outras coisas que têm a ver com guarda de dados em cache e servidores secundários de DNS. Para os demais campos, Atualização, Tentativas, Expiração e TTL, pode-se usar os valores aqui indicados e se estará seguro.

Agora reinicializaremos o servidor de nomes (através do comando `ndc restart`), e usaremos nslookup para examinar o que foi feito:

```
$ nslookup

Servidor Padrão: localhost
Endereço:  127.0.0.1

> 127.0.0.1
Servidor:  localhost
Endereço:  127.0.0.1

Nome:    localhost
Endereço:  127.0.0.1
```

observamos então que é possível chegar a `localhost` a partir do endereço 127.0.0.1. Agora a nossa tarefa principal, no campo `linux.bogus`, vamos inserir uma nova seção chamada "zone" no `named.conf`:

```
zone "linux.bogus" {
        notify no;
        type master;
        file "pz/linux.bogus";
};
```

Note a ausência de "." no nome do domínio no arquivo `named.conf`.

No arquivo de zona do domínio linux.bogus colocaremos alguns dados totalmente inventados:

```
;
```

```
; Arquivo zona para linux.bogus
;
; O arquivo completo de zone
;


@  IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
               199802151       ; serial, data de hoje +  serial de hoje
                      8H       ; Atualização, segundos
                      2H       ; Tentativa, segundos
                      1W       ; Expiração, segundos
                      1D )     ; TTL, segundos
;
           NS      ns              ; Endereço Internet do nome do servidor
           MX      10 mail.linux.bogus     ; Servidor de Correio Primário
   MX      20 mail.friend.bogus.   ; Servidor de Correio Secundário
;
localhost   A       127.0.0.1
ns          A       192.168.196.2
mail        A       192.168.196.4
```

Dois aspectos devem ser observados sobre o registro SOA. ns.linux.bogus *deve* ser uma máquina real com um registro A. Não é permitido ter um registro CNAME para a máquina mencionada no registro SOA. O nome não precisa ser 'ns', pode ser qualquer nome de máquina válido. Em seguida, a hostmaster.linux.bogus deve ser lido como hostmaster@linux.bogus, o qual deve ser um nome alternativo de correio, ou caixa postal, acessado pela(s) pessoa(s) que mantém o DNS e leiam a correspondência freqüentemente. Qualquer correspondência relativa ao domínio será enviada para o endereço relacionado aqui. O nome não precisa ser 'hostmaster', pode ser qualquer endereço email válido, mas espera-se que o endereço email 'hostmaster' *funcione* bem também.

Há um novo tipo RR neste arquivo, o MX, ou registro de recurso de servidor de correio. Este arquivo diz aos sistemas de correspondência para onde enviar a correspondência endereçada para `alguém@linux.bogus`, ou seja no nosso caso `mail.linux.bogus` ou `mail.friend.bogus`. O número antes de cada nome de máquina define a prioridade. O RR com o número mais baixo tem prioridade. Caso ele não esteja ativo ou apresentar algum erro, a mensagem pode ser enviada a um outro servidor de mensagens com um número mais alto, um operador de correspondência secundário, ou seja, no nosso caso, `mail.friend.bogus` que tem prioridade 20.

Ao se reiniciar o servidor de nomes executando-se `ndc restart` obteremos os seguintes resultados com nslookup:

```
$ nslookup
> set q=any
> linux.bogus
Server:  localhost
Address:  127.0.0.1

linux.bogus
        origin = ns.linux.bogus
        mail addr = hostmaster.linux.bogus
        serial = 199802151
        refresh = 28800 (8 horas)
        retry   = 7200 (2 horas)
        expire  = 604800 (7 dias)
        minimum ttl = 86400 (1 dia)
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     preference = 10, mail exchanger = \
                                        mail.linux.bogus.linux.bogus
linux.bogus     preference = 20, mail exchanger =  \
                                        mail.friend.bogus
linux.bogus     nameserver = ns.linux.bogus
ns.linux.bogus  internet address = 192.168.196.2
mail.linux.bogus        internet address = 192.168.196.4
```

Com um exame mais apurado pode-se descobrir um pequeno problema. A linha

```
linux.bogus     preference = 10, mail exchanger = \
                                        mail.linux.bogus.linux.bogus
```

deveria ser

```
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus
```

Deliberadamente cometemos o erro para que o leitor aprenda com ele:-) Examinando o arquivo de zona, percebemos que na linha

```
   MX      10 mail.linux.bogus     ; Servidor primário de correio
```

está faltando um ponto. Ou seja, há "linux.bogus" demais. Caso um nome de máquina não seja seguido por um ponto num arquivo de zona, a origem será acrescentada ao final causando o duplo `linux.bogus.linux.bogus`. Portanto:

```
   MX      10 mail.linux.bogus.    ; Servidor primário de correio
```

ou

```
   MX      10 mail                 ; Servidor primário de correio
```

estão corretos. Particularmente, sugerimos a última forma, por ser mais econômica e menos sujeita a erros. Existem alguns bem conhecidos usuários de bind que discordam e outros que concordam com isto. Num arquivo de zona, o domínio pode tanto ser totalmente identificado e terminado com um "." ou não deve ser incluído de forma alguma, utilizando então o padrão da origem.

Devemos salientar que em um arquivo named.conf *não* deve haver "." depois dos nomes dos domínios. Você não tem idéia de quantas vezes um "." gerou uma enormidade de problemas e confundiu um punhado de administradores.

Agora que já expressamos nosso ponto de vista, estamos com o novo arquivo de zona, com informações extras também:

```
;
; Arquivo de zona para linux.bogus
;
; O arquivo de zona completo
;
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151       ; serial, data de hoje + serial de hoje #
                        8H              ; Atualizar, segundos
                        2H              ; Tentativas, segundos
                        1W              ; Expiração, segundos
                        1D )            ; TTL, segundos
;
                TXT     "Linux.Bogus, os especialistas DNS "
                NS      ns              ; Endereço Internet do servidor de nomes
                NS      ns.friend.bogus.
                MX      10 mail         ; Servidor de correio primário
                MX      20 mail.friend.bogus. ; Servidor de correio secundário

localhost       A       127.0.0.1

gw              A       192.168.196.1
                HINFO   "Cisco" "IOS"
                TXT     "O roteador"

ns              A       192.168.196.2
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "Pentium" "Linux 2.0"
www             CNAME    ns
```

```
donald          A       192.168.196.3
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "i486"  "Linux 2.0"
                TXT     "DEK"

correio         A       192.168.196.4
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 2.2"

ftp             A       192.168.196.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 2.0.36"
```

Há diversos RRs novos: HINFO (INFOrmação da Máquina) tem duas partes, sendo aconselhável indicar os dois. A primeira parte é o hardware ou CPU da máquina, e a segunda parte é o software ou OS da máquina. O servidor de nomes 'ns' tem uma CPU Pentium e executa o Linux 2.0. CNAME (NOME Canônico) é uma maneira de dar a uma mesma máquina vários nomes. Assim www é um nome alternativo para o ns.

O uso do registro CNAME é um pouco controvertido. Mas é seguro seguir a regra onde um registro MX, CNAME ou SOA *nunca* deve referir-se a um registro CNAME , e devem referir-se somente a um registro A, sendo portanto incorreto ter-se:

```
itamaracabar            CNAME   www                     ; NÃO!
```

o correto seria:

```
itamaracabar            CNAME   ns                      ; SIM!
```

É também seguro supor que um CNAME não é um nome de máquina válido para um endereço email, por exemplo `webmaster@www.linux.bogus` é um endereço ilegal, conforme a configuração acima. Não se deve esperar que muitos administradores de servidores de mensagens usem esta configuração, mesmo se ela funcionar localmente. A maneira para evitar isto é usar registros de tipo A (e talvez alguns outros também, como um registro MX):

```
www             A       192.168.196.2
```

Um grande número de magos do DNS, sugerem que o CNAME *não* seja utilizado. Por isso, devemos considerar esta sugestão *muito* seriamente.

Mas como se pode perceber, este COMO FAZER e muitos sites não seguem esta regra.

Pode carregar o novo banco de dados executando-se `ndc reload`, o que fará com que o named leia seus arquivos novamente.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.bogus
```

Isto significa que todos os registros devem ser apresentados. O resultado será:

```
[localhost]
$ORIGIN linux.bogus.
@                       1D IN SOA       ns hostmaster (
                                        199802151       ; nro. serial
                                        8H              ; atualizar
                                        2H              ; tentativas
                                        1W              ; expiração
                                        1D )            ; mínimo

                        1D IN NS     ns
                        1D IN NS     ns.friend.bogus.
                        1D IN TXT    "Linux.Bogus, os consultores DNS"
                        1D IN MX     10 mail
                        1D IN MX     20 mail.friend.bogus.
gw                      1D IN A      192.168.196.1
                        1D IN HINFO  "Cisco" "IOS"
                        1D IN TXT    "O roteador"
mail                    1D IN A      192.168.196.4
                        1D IN MX     10 mail
                        1D IN MX     20 mail.friend.bogus.
                        1D IN HINFO  "386sx" "Linux 1.0.9"
localhost               1D IN A      127.0.0.1
www                     1D IN CNAME  ns
donald                  1D IN A      192.168.196.3
                        1D IN MX     10 mail
                        1D IN MX     20 mail.friend.bogus.
                        1D IN HINFO  "i486" "Linux 1.2"
                        1D IN TXT    "DEK"
ftp                     1D IN A      192.168.196.5
                        1D IN MX     10 mail
                        1D IN MX     20 mail.friend.bogus.
                        1D IN HINFO  "P6" "Linux 1.3.59"
```

```
ns                      1D IN A      192.168.196.2
                        1D IN MX     10 mail
                        1D IN MX     20 mail.friend.bogus.
                        1D IN HINFO  "Pentium" "Linux 1.2"
@                       1D IN SOA    ns hostmaster (
                                     199802151       ; nro. serial
                                     8H              ; atualizar
                                     2H              ; tentativas
                                     1W              ; expiração
                                     1D )            ; mínimo
```

Parece ótimo. Como se pode ver parece muito com o arquivo de zona. Vamos verificar o que ele diz para www:

```
> set q=any
> www.linux.bogus.
Server:  localhost
Address:  127.0.0.1

www.linux.bogus canonical name = ns.linux.bogus
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 192.168.196.2
```

Em outras palavras, o nome real de `www.linux.bogus` é `ns.linux.bogus`, e ele fornece algumas informações adicionais que ele possua sobre ns, o suficiente para um programa conectar-se a ele.

Agora estamos no meio do caminho.

C.4.3 A zona reversa

Agora os programas podem converter os nomes em linux.bogus para endereços com os quais eles podem se conectar. Porém é pedido também uma zona reversa, que torne o DNS capaz de converter um endereço em um nome. Este nome é usado por muitos servidores de espécies diferentes (FTP, IRC, WWW e outros) para decidir se eles querem conversar com a máquina local ou não, e em caso positivo, também qual a prioridade que deve ser dada a esta máquina. Para o acesso completo a todos os serviços da Internet, uma zona reversa é necessária.

Deve-se colocar o seguinte em `named.conf`:

```
zone "196.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "pz/192.168.196";
};
```

Estes parâmetros são exatamente iguais para `0.0.127.in-addr.arpa` e os conteúdos são semelhantes:

```
@       IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151 ; Nro.Serial, data + nro. série
                        8H      ; Atualizar
                        2H    ; Tentativas
                        1W      ; Expiração
                        1D)     ; TTL mínimo

                NS    ns.linux.bogus.

1               PTR     gw.linux.bogus.
2               PTR     ns.linux.bogus.
3               PTR     donald.linux.bogus.
4               PTR     mail.linux.bogus.
5               PTR     ftp.linux.bogus.
```

Agora ao reinicializar o servidor de nomes (`ndc restart`) e examinar o trabalho realizado, utilizando-se o nslookup, teremos:

```
> 192.168.196.4
Server:  localhost
Address:  127.0.0.1

Name:    mail.linux.bogus
Address:  192.168.196.4
```

então caso tudo pareça correto, vamos examinar todas as demais informações:

```
> ls -d 196.168.192.in-addr.arpa
[localhost]
```

```
$ORIGIN 196.168.192.in-addr.arpa.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
                                        199802151       ; nro. serial
                                        8H              ; atualizar
                                        2H              ; tentativas
                                        1W              ; expiração
                                        1D )            ; ttl mínimo

                        1D IN NS        ns.linux.bogus.
1                       1D IN PTR       gw.linux.bogus.
2                       1D IN PTR       ns.linux.bogus.
3                       1D IN PTR       donald.linux.bogus.
4                       1D IN PTR       mail.linux.bogus.
5                       1D IN PTR       ftp.linux.bogus.
@                       1D IN SOA       ns.linux.bogus. hostmaster.linux.bogus. (
199802151       ; nro. serial
                                        8H              ; atualizar
                                        2H              ; tentativas
                                        1W              ; expiração
                                        1D )            ; ttl mínimo
```

Parece bom!

Há algumas coisas que devemos acrescentar. Os números IP usados nos exemplos acima foram tirados dos blocos de 'redes privadas', ou seja, eles não podem ser usados publicamente na Internet. Por isso eles são seguros para serem usados em um exemplo de um COMO FAZER. A segunda coisa é a linha `notify no;`, a qual indica que o servidor de nomes não notificará o servidor secundário (escravo), quando houver uma atualização para um dos arquivos de zona. No bind-8 o servidor de nomes pode notificar os outros servidores relacionados nos registros NS no arquivo de zona, toda vez que ela for atualizada. Isto é conveniente para o uso diário e usual, mas em nossas experiências particulares com zonas, esta característica deve ser desativada, afinal não queremos que a experiência polua toda a Internet, queremos?

E claro, este domínio é totalmente inventado, assim como todos os endereços que estão nele. Para um exemplo real de um domínio real, veja a próxima seção.

C.5 Um Exemplo de Domínio Real

Onde listamos alguns arquivos *de zona reais*

Os usuários têm sugerido que seja incluído um exemplo real de um domínio em operação, bem como um exemplo detalhado.

Usaremos este exemplo com a permissão de David Bullock da LAND-5. Estes arquivos eram atuais em 24 de setembro de 1996 e foram então editados para corresponder às restrições da bind-8 e às extensões usadas pelo autor. Assim, o que se vê aqui, difere um pouco do que se pode encontrar ao se perguntar aos servidores de nomes do LAND-5.

C.5.1 /etc/named.conf (ou /var/named/named.conf)

Aqui encontramos as seções mestre de zona para as duas zonas reversas necessárias: a rede 127.0.0 , bem como a rede LAND-5 206.6.177, além de uma linha primária para o land-5.com. Note ainda que ao invés de colocar os arquivos em um diretório chamado `pz`, como foi feito anteriormente, eles foram colocados no diretório chamado `zone`.

```
//  Arquivo de inicialização do servidor de nomes de LAND-5

options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "zone/127.0.0";
};

zone "land-5.com" {
        type master;
        file "zone/land-5.com";
};

zone "177.6.206.in-addr.arpa" {
        type master;
        file "zone/206.6.177";
};
```

Caso este arquivo seja definido como o arquivo named.conf de uma máquina local, *POR FAVOR* use o parâmetro `notify no;` nas seções de zona para as duas zonas land-5, a fim de evitar acidentes.

C.5.2 /var/named/root.hints

Deve-se ter em mente que este é um arquivo dinâmico e o aqui descrito pode não significar a realidade atual. É sugerido utilizar um modelo atualizado, produzido pelo utilitário dig, conforme explicado anteriormente.

```
; <<>> DiG 8.1 <<>> @A.ROOT-SERVERS.NET.
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
```

```
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241

;; Total query time: 215 msec
;; FROM: roke.uio.no to SERVER: A.ROOT-SERVERS.NET.  198.41.0.4
;; WHEN: Sun Feb 15 01:22:51 1998
;; MSG SIZE  sent: 17  rcvd: 436
```

C.5.3 /var/named/zone/127.0.0

Somente as informações básicas são obrigatórias, como o registro SOA e um registro que mapeie 127.0.0.1 para `localhost`. Nenhuma outra informação deve estar contida neste arquivo. Provavelmente ele nunca precisará ser atualizado, a menos que o endereço do servidor de nomes ou da máquina mestra seja alterado.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199909203       ; Nro. Serial
                                28800   ; Atualizar
                                7200    ; Tentativas
                                604800  ; Expiração
                                86400)  ; TTL Mínimo
                        NS      land-5.com.

1                       PTR     localhost.
```

C.5.4 /var/named/zone/land-5.com

Aqui teremos um registro SOA obrigatório com os registros NS necessários. Podemos ver que há um servidor secundário em ns2.psi.net. Este é o procedimento correto, *sempre* ter um site como servidor secundário que esteja fora da rede local. Podemos verificar que ele tem uma máquina mestra chamada land-5, o qual cuida de muitos serviços diferentes da Internet, e que ele foi definido com diversos registros CNAME (uma alternativa seria usar os registros de recursos do tipo A).

Como se pode verificar no registro SOA, o arquivo de zona tem origem em land-5.com e a pessoa de contato é `root@land-5.com`. O `hostmaster` é outro endereço usado com freqüência na definição de pessoa de contato. O número serial está no formato habitual ano-mês-dia com o números seriais acrescentados, sendo esta provavelmente a sexta versão do arquivo de zona datada de 20 de setembro de 1996. Lembre-se que o número serial *deve* aumentar ordenadamente, onde hoje temos apenas *um* dígito para serial#; assim depois da nona alteração no dia de hoje ele terá que esperar até amanhã antes de poder editar o arquivo novamente. É aconselhável o de dois dígitos para evitar este tipo de problema.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206  ; nro. serial, data de hoje + serial
                        8H            ; atualizar em segundos
                        2H            ; tentativas em segundos
                        1W            ; expiração em segundos
                        1D )          ; mínimo em segundos
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Servidor primário de correio

Localhost        A      127.0.0.1

Router           A      206.6.177.1

land-5.com.      A      206.6.177.2
ns               A      206.6.177.3
www              A      207.159.141.192

ftp          CNAME      land-5.com.
mail         CNAME      land-5.com.
news         CNAME      land-5.com.

funn             A      206.6.177.2

@              TXT      "Corporação LAND-5"

;
;       Estações de Trabalho
;
ws-177200       A       206.6.177.200
               MX       10 land-5.com.
ws-177201       A       206.6.177.201
                MX      10 land-5.com. ; Servidor primário de correio
ws-177202       A       206.6.177.202
                MX      10 land-5.com. ; Servidor primário de correio
ws-177203       A       206.6.177.203
                MX      10 land-5.com. ; Servidor primário de correio
ws-177204       A       206.6.177.204
                MX      10 land-5.com. ; Servidor primário de correio
ws-177205       A       206.6.177.205
                MX      10 land-5.com. ; Servidor primário de correio
                 ; {Definições repetitivas retiradas - SNIP}
ws-177250       A       206.6.177.250
                MX      10 land-5.com. ; Servidor primário de correio
ws-177251       A       206.6.177.251
                MX      10 land-5.com. ; Servidor primário de correio
ws-177252       A       206.6.177.252
                MX      10 land-5.com. ; Servidor primário de correio
```

```
ws-177253       A       206.6.177.253
                MX      10 land-5.com. ; Servidor primário de correio
ws-177254       A       206.6.177.254
                MX      10 land-5.com. ; Servidor primário de correio
```

Ao examinarmos o servidor de nomes land-5, descobriremos que os nomes das máquinas estão no formato `ws_`*número*. Como nas versões recentes do bind 4, o named começa a impor restrições nos caracteres que podem ser usados como nomes das máquinas. Por isso, o original não funcionava com bind-8 e foram substituídos então os '-'(travessões) por '_'(sublinhados).

Uma outra coisa a ser notada é que as estações operacionais não possuem nomes individuais, mas um prefixo seguido pelas duas últimas partes dos números IP. Usando-se tal convenção, pode-se simplificar significativamente a manutenção, mas pode ser um pouco impessoal, e na verdade, se tornar uma fonte de descontentamento entre os usuários.

Vemos também que `funn.land-5.com` é um nome alternativo para land-5.com, mas usando um registro A e não um registro CNAME.

C.5.5 /var/named/zone/206.6.177

Comentaremos este arquivo em seguida.

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206       ; Nro. Serial
                                28800   ; Atualizar
                                7200    ; Tentativa
                                604800  ; Expiração
                                86400)  ; TTL Mínimo
                        NS      land-5.com.
                        NS      ns2.psi.net.
;
;       Servidores
;
1       PTR     router.land-5.com.
2       PTR     land-5.com.
2       PTR     funn.land-5.com.
;
;       Estações de trabalho
```

```
;
200     PTR     ws-177200.land-5.com.
201     PTR     ws-177201.land-5.com.
202     PTR     ws-177202.land-5.com.
203     PTR     ws-177203.land-5.com.
204     PTR     ws-177204.land-5.com.
205     PTR     ws-177205.land-5.com.
; {Muitas definições repetidas foram suprimidas - SNIP}
250     PTR     ws-177250.land-5.com.
251     PTR     ws-177251.land-5.com.
252     PTR     ws-177252.land-5.com.
253     PTR     ws-177253.land-5.com.
254     PTR     ws-177254.land-5.com.
```

A zona reversa é o aspecto da configuração que parece causar a maior dificuldade. É usada para se encontrar o nome da máquina, caso se tenha o seu endereço IP. Por exemplo: caso a máquina seja um servidor IRC que aceita conexões de clientes IRC. No entanto este é um servidor Norueguês e por isso, somente serão aceitas conexões de clientes na Noruega e outros países escandinavos. Quando se obtém uma conexão de um cliente, a biblioteca C é capaz de indicar o número IP da máquina remota, porque o número IP do cliente está contido em todos os pacotes que são enviados para a rede. Pode-se então usar uma função chamada gethostbyaddr, a qual pesquisa o nome de uma máquina dado o número IP. Gethostbyaddr perguntará a um servidor DNS, o qual procurará pela máquina. Supondo-se que a conexão cliente foi originada por ws-177200.land-5.com. O número IP que a biblioteca C fornece para o servidor IRC é 206.6.177.200. Para descobrir o nome daquela máquina, precisamos encontrar 200.177.6.206.in-addr.arpa. O servidor DNS primeiramente encontrará os servidores arpa., então os servidores in-addr.arpa. seguidos pelos servidores reversos de 206, então 6 e finalmente é encontrando o servidor para a zona 177.6.206.in-addr.arpa em land-5. A partir deste se obterá a resposta, ou seja o registro 'PTR ws-177200.land-5.com', que indica que o nome de 206.6.177.200 é igual a ws-177200.land-5.com. Assim como com as explicações sobre a forma de pesquisa de prep.ai.mit.edu, esta descrição também é um pouco simplificada em relação ao que efetivamente ocorre.

Voltando ao exemplo do servidor IRC. O servidor IRC só aceita conexões de países escandinavos, ou seja, *.no, *.se ou*.dk e o nome ws-177200.land-5.com claramente não combina com qualquer uma delas, sendo então negada a conexão. Caso *não* houvesse o mapeamento reverso de 206.2.177.200 através da zona in-addr.arpa, o servidor estaria incapacitado de encontrar o nome e teria que comparar 206.2.177.200 com *.no, *.se e *.dk, onde evidentemente nenhuma das opções coincidiria.

Algumas pessoas afirmam que a pesquisa de mapeamentos reversos são importantes apenas para os servidores, ou ainda que não são importantes de forma alguma. A verdade nos parece bem diferente: muitos servidores ftp, notícias, IRC e até mesmo alguns http (WWW) *não* aceitarão conexões de máquinas das quais não seja possível encontrar o nome. Por isso os mapeamentos reversos são na verdade *obrigatórios*.

C.6 Manutenção

Mantendo o sistema funcionando.

Há uma tarefa de manutenção que se deve executar no named, além de mantê-los funcionando, que é manter o arquivo `root.hints` atualizado. A maneira mais fácil é usar o utilitário dig, o qual deve ser executado inicialmente sem argumentos, gerando um `root.hints` adequado ao servidor. A seguir deve ser perguntado a um dos servidores relacionados o seguinte: `dig@rootserver`. Pode-se notar que a saída se parecerá muitíssimo como um arquivo root.hints. Ela deve ser salva em um arquivo (`dig@e.root-servers.net.ns$>$root.hints.new`) que servirá de substituto ao `root.hints` anterior.

O servidor de nomes deverá ser então reiniciado para substituir o cache antigo.

Al Longyear enviou este programa, o qual pode ser executado automaticamente para atualizar `root.hints`; basta configurar uma entrada no crontab para executá-lo por exemplo uma vez ao mês. O programa assume que se tenha um servidor de correio funcionando e que o nome alternativo de endereço de correio eletrônico 'hostmaster' está definido.

```
#!/bin/sh
#
# Atualiza as informações do cache do servidor de nomes uma vez ao mês
# É executado automaticamente uma vez ao mês através de uma entrada no cron
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"
 echo "Subject: Atualização automática do arquivo named.conf "
 echo
```

```
 export PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 cd /var/named

 dig @rs.internic.net . ns >root.hints.new

 echo "O arquivo named.conf foi atualizado, passando a conter as seguintes informações:"
 echo
 cat root.hints.new

 chown root.root root.hints.new
 chmod 444 root.hints.new
 rm -f root.hints.old
 mv root.hints root.hints.old
 mv root.hints.new root.hints
 ndc restart
 echo
 echo "O servidor de nomes foi reinicializado para garantir que a atualização foi completada".
 echo "O arquivo root.hints anterior foi renomeado para /var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

Alguns dos leitores mais avançados podem saber que o arquivo root.hints está também disponível via ftp na Internet. Por favor *não* use ftp para atualizar root.hints, o método acima é muito mais amigável para a rede.

C.7 Converter da versão 4 para versão 8

Esta foi originalmente uma seção sobre o uso da bind 8 escrita por David E. Smith (dave@bureau42.ml.org). Ela foi editada para conter o novo nome da seção.

Não há muito a acrescentar. Exceto pelo uso do servidor named.conf ao invés de servidor named.boot, tudo mais é idêntico; bind8 vem com um programa perl que converte arquivos de estilo velho para o novo formato. Exemplo de um named.boot (velho estilo) para um servidor de nomes somente para cache:

```
directory /var/named
cache   .                                  root.hints
primary 0.0.127.IN-ADDR.ARPA          127.0.0.zone
primary localhost                     localhost.zone
```

Na linha de comando, no diretório bind8/src/bin/named (*presume-se aqui que se tenham os fontes da distribuição. Caso se localize somente o pacote binário, o programa estará por perto*), digite:

```
./named-bootconf.pl < named.boot > named.conf
```

o qual criará o seguinte named.conf:

options { directory "/var/named"; };

zone "."{ type hint; file "root.hints"; };

zone "0.0.127.IN-ADDR.ARPA"{ type master; file "127.0.0.zone"; };

zone "localhost"{ type master; file "localhost.zone"; };

Funciona para tudo o que puder estar presente em um arquivo named.boot, embora ele não acrescente todas as novas funcionalidades e opções de configuração que o bind8 permite. Aqui está um named.conf mais completo, o qual faz as mesmas coisas, mas de uma forma um pouco mais eficaz.

```
// Este é um arquivo de configuração para o named (BIND 8.1 ou mais recente).
// Deve ser instalado em /etc/named.conf.
// A única mudança feita no named.conf (à parte deste comentário:) é que a
// linha de diretório foi descomentada, uma vez que já se tinha os arquivos
// de zona em /var/named.

options {
        directory "/var/named";
        check-names master warn;                /* padrão. */
        datasize 20M;
};


zone "localhost" IN {
        type master;
        file "localhost.zone";
        check-names fail;
        allow-update { none; };
        allow-transfer { any; };
};



zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "127.0.0.zone";
```

```
        check-names fail;
        allow-update { none; };
        allow-transfer { any; };
};


zone "." IN {
        type hint;
        file "root.hints";
};
```

`bind8/src/bin/named/test` tem este conteúdo e cópias dos arquivos de zonas, que muitos podem simplesmente começar a usar.

Os formatos por arquivos de zona e dos arquivos `root.hints` são idênticos, assim como são os comandos para atualizá-los.

C.8 Perguntas e Respostas

Por favor leia esta seção com atenção antes de enviar mensagens ao autor.

1. Meu named necessita de um arquivo named.boot.

   Você está lendo o COMO FAZER errado. Por favor veja a versão antiga deste COMO FAZER que converte bind 4 em `http://www.math.uio.no/~janl/DNS/`.

2. Como usar o DNS de dentro de um firewall? Algumas dicas: 'retransmissores', 'escravo' e dê uma olhada na lista de literatura no final deste COMO FAZER.

3. Como fazer para o DNS alternar através de diversos endereços disponíveis para um serviço, digamos, www.busy.site para obter um efeito de carga balanceada ou similar?

   Faça vários registros **A** para www.busy.site e use a bind 4.9.3 ou posterior. Então bind irá fornecer as respostas, porém *não* funcionará com versões mais antigas do bind.

4. Gostaria de configurar o DNS em uma intranet (fechada). O que eu faço?

Pode-se omitir o arquivo root.hints e construir somente os arquivos de zona. Isto significa ainda que você não tem que conseguir um novo arquivo hint o tempo todo.

5. Como configurar um servidor de nomes secundário (escravo)?

   Caso o servidor primário/mestre tiver, por exemplo, o endereço 127.0.0.1, basta colocar uma linha no arquivo named.conf do secundário:

```
zone "linux.bogus" {
      type slave;
      file "sz/linux.bogus";
      masters { 127.0.0.1; };
};
```

   Pode-se relacionar vários servidores mestres alternativos. O arquivo de zona pode ser copiado de dentro de uma lista de `mestres`, separada por ";" (ponto e vírgula).

6. Eu quero executar o bind quando estiver desconectado da rede.

   Há aspectos relacionados com isto:

   - Eu recebi esta mensagem de Ian Clark <`ic@deakin.edu.au`> onde ele explica a sua maneira de fazer isto:

     Eu executo named na minha máquina "Masquerading", tendo dois arquivos root.hints, um chamado `root.hints.real` que contém os nomes dos servidores de nomes raiz reais e o outro chamado `root.hints.falso` que contém...

     ```
     ----
     ; root.hints.falso
     ; este arquivo não contém informações
     ----
     ```

     Ao desconectar-se da rede, eu copio o arquivo `root.hints.falso` para `root.hints` e reinicio o `named`. Ao conectar-se novamente à rede, o root.hints.real é copiado para root.hints e o named é reiniciado.

     Isto é feito pelos programas ip-down & ip-up respectivamente.

     A primeira vez que uma pergunta pesquisa é feita com a rede desconectada, o servidor de nomes não tem meios de obter o seu endereço e apresenta a seguinte mensagem:

```
28 de jan. 20:10:11 servidor de nomes hazchem [10147]:
Nenhum servidor de nomes raiz foi localizado para a classe IN...
```

com a qual eu posso viver.

Certamente parece funcionar para mim. É possível utilizar o servidor de nomes para máquinas locais, enquanto estiver desconectado da rede, sem o tempo de espera necessário para nomes de domínios externos e enquanto as pesquisas na rede por outros domínios "funcionam"a contento.

- Recebi de Karl-Max Wanger informações sobre como bind interage com o NFS e o portmapper numa máquina fora da rede:

  Eu executo meu próprio named em todas as máquinas que ocasionalmente estejam conectadas a Internet via modem. O servidor de nomes atua somente como um cache, ele não tem de autoridade e pergunta tudo aos servidores de nomes indicados no arquivo root.cache. Como de costume no Slackware, ele é iniciado antes em nfsd e mountd.

  Com uma de minhas máquinas (um Libretto 30) eu tive o problema de algumas vezes poder montá-lo a partir de outro sistema conectado à rede local, mas na maior parte do tempo isso não ser possível. Obtive o mesmo resultado usando, PLIP, um cartão Ethernet PCMCIA ou PP sobre uma interface serial.

  Depois de algum tempo de tentativas e experiências, descobri que aparentemente o named ficava confuso com o processo de registro do nfsd e mountd junto ao portmapper, após a sua inicialização (estes servidores sempre foram inicializados da forma usual). Inicializando o named após nfsd e mountd eliminou este problema completamente.

  Como não existem desvantagens em tal seqüência modificada de inicialização aconselho a todos que a utilizem para prevenir potenciais problemas.

7. Onde o nome do servidor somente de cache guarda seu cache? Há alguma maneira de controlar o tamanho do cache?

   O cache é mantido integralmente em memória, ele *não* é gravado em disco em nenhum momento. Toda vez que se finaliza o named, o cache é perdido. O cache *não* é controlável de nenhuma maneira. O named administra-o de acordo com algumas regras simples e é isso. Não se pode controlar o cache ou o tamanho do cache. Caso se deseje pode se alterar o programa named, porém isto não é recomendado.

8. O named salva o cache? Posso fazer com ele o salve?

   Não, o servidor de nomes *não* salva o cache quando ele é finalizado. Isto significa que o cache tem que ser construído de novo cada vez que se reinicia o servidor de nomes. Não há nenhuma maneira de fazer com que o servidor de nomes salve o cache em um arquivo. Caso se deseje pode se alterar o programa named, porém isto não é recomendado.

C.9 Como tornar-se um administrador DNS.

Documentação e Ferramentas.

A documentação real existe, on-line e impressa. A leitura de várias destas é necessária para tornar-se um administrador DNS. Em formato impresso, o livro padrão é *DNS e BIND* por C. Liu e P. Albitz de O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Eu o li e digo-lhes que é excelente. Há também uma seção sobre DNS em *TCP/IP Administração de Rede*, por Craig Huny da O'Reilly..., ISBN 0-937175-82-X. Uma outra sugestão para uma boa administração DNS (ou bom para qualquer coisa) é *Zen e a Arte da Manutenção da Motocicleta* de Robert M. Prisig :-) Disponível em ISBN 0688052304 e outros.

On-line pode-se encontrar material em `http://www.dns.net/dnsrd/`, `http://www.isc.org/bind.html`; um FAQ, uma referência manual (BOG; Guia de Operações de Bind) bem como documentos e definições de protocolos e programas DNS (estes, e a maioria, se não todos, dos rfcs mencionados abaixo, estão também contidos na distribuição de bind). Eu não li a maioria destes, mas eu também não sou um grande administrador DNS. Arnt Gulbrandsen, por outro lado, leu o BOG e ficou entusiasmado com ele :-). O grupo de notícias também trata sobre DNS. Além disso há um número de RFCs sobre DNS, sendo estes provavelmente os mais importantes:

RFC 2052

A. Gulbrandsen, P. Vixie, *Um DNS RR para a especificação da localização dos serviços(DNS SRV)*, Outubro de 1996

RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Alocação de Endereços para Internets Particulares* , 29/02/1996.

RFC 1912

D. Barr, *Erros Comuns na Operação e Configuração DNS*, 28/02/1996.

RFC 1912 Erros

B. Barr *Erros na RFC 1912*, está disponível em:
`http://www.cis.ohio-state.edu/~barr/rfc1912-errors.html`.

RFC 1713

A. Romao, *Ferramentas para depuração do DNS* , 03/11/1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *Codificação DNS para Localização Geográfica*, 01/11/1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *Novas Definições de RR DNS*, 08/10/1990.

RFC 1035

P. Mockapetris, *Domínios - implementação e especificação*, 01/11/1987.

RFC 1034

P. Mockapetris, *Domínios - conceitos e instalações*, 01/11/1987.

RFC 1033

M. Lottor, *Guia de operações de administradores de domínios*, 01/11/1987.

RFC 1032

M. Stahl, *Guia de administradores de domínios* , 01/11/1987.

RFC 974

C. Partridge, *Roteamento de correio e domínios* , 01/01/1986.

Apêndice D

Como Fazer - NFS

D.1 Preâmbulo

D.1.1 Nota Legal

 A seção de Perguntas e Respostas é baseada no FAQ NFS de Alan Cox. A seção da lista de verificações é baseada na lista de problemas de montagem compilada pela IBM Corporation.

D.1.2 Outros Assuntos

Este nunca será um documento finalizado, devido à dinâmica do tema. Por favor envie-nos informações sobre problemas e sucessos, que possam melhorar este Como Fazer. Por favor contribuições financeiras, comentários e questões podem ser enviadas para `janl@math.uio.no`. Caso uma mensagem seja enviada, por favor esteja *seguro* de que o endereço para resposta está correto e funcionando, pois eu recebo *muitos* emails e tentar descobrir endereços pode ser uma tarefa cansativa. Obrigado.

Agradecimentos a Olaf Kirch que me convenceu a escrever este documento e forneceu-me grandes sugestões. `:-)`

Este Como Fazer cobre o NFS nas versões 2.0 do kernel. Há melhorias significativas e mudanças do NFS nas versões subseqüentes do kernel.

Caso se deseje traduzir este Como Fazer por favor, avise-me para que eu possa estar ciente sobre a quantidade de idiomas em que eu já fui publicado :-).

D.1.3 Dedicatória

Este Como Fazer é dedicado a Anne Line Norheim Langfeldt, que provavelmente nunca o lerá, já que ela não é deste tipo de garota.

D.2 LEIAME Antes!

NFS, o Sistema de Arquivos em Rede tem três importantes características:

- Possibilita o compartilhamento de arquivos sobre uma rede local.
- Funciona bastante bem.
- Impede diversos problemas de segurança que são bem conhecidos por invasores e podem ser explorados na obtenção de acesso (leitura, gravação e remoção) de todos os arquivos de um sistema.

Abordaremos todos estes assuntos neste documento. Por favor, não deixe de ler os itens sobre segurança neste documento, o que tornará a rede menos vulnerável a riscos tolos de segurança. As passagens sobre segurança serão bastante técnicas e exigirão conhecimento sobre redes IP e sobre os termos usados. Caso não se reconheça algum dos termos aqui usados, verifique o Como Fazer - Redes ou obtenha um livro sobre administração de redes TCP/IP. Esta é uma boa idéia de qualquer forma, caso se esteja administrando máquinas Unix/Linux. Um livro muito bom é *TCP/IP Network Administration* de Craig Hunt, publicado pela O'Reilly & Associates, Inc. E após toda esta leitura, certamente você será mais valorizado no mercado de trabalho, e isso não se pode perder ;-)

Há duas seções de ajuda com problemas no NFS, chamadas *Lista de Verificação* e *FAQs*. Por favor, leia com atenção, caso algo não funcione da maneira esperada.

D.3 Configurando um Servidor NFS

D.3.1 Pré-Requisitos

Antes de continuar a leitura deste Como Fazer, será necessário poder executar-se o programa telnet *de* e *para* as máquinas que serão usadas como servidor e cliente. Caso isso não esteja funcionando, pedimos que seja checada a rede e sugerimos a leitura do Como Fazer Net-2 para configurar a rede adequadamente.

D.3.2 Primeiros Passos

Antes que se possa fazer qualquer coisa será necessário ter um servidor NFS configurado. Caso se faça parte de alguma rede de um departamento ou rede universitária provavelmente já existirão diversos servidores NFS sendo executados. Casos eles permitam o acesso, ou ao invés disso se esteja lendo este Como Fazer para se obter acesso a um servidor NFS, não é necessário ler esta seção, podendo passar-se diretamente à seção D.4 (Configurando um cliente NFS).

Caso se necessite configurar um sistema diferente do Linux para atuar como servidor, será necessário ler o manual do sistema para descobrir como habilitar o NFS e a exportação de sistemas de arquivos. Há uma seção neste documento explicando como fazer isto em muitos sistemas diferentes. Após se verificar isso tudo pode-se continuar na leitura desta seção.

Aqueles que continuaram a sua leitura estão avisados: vamos ter que configurar uma série de programas.

D.3.3 O Portmapper

O portmapper no Linux é chamado também de `portmap` ou `rpc.portmap`. A página de manual on-line diz que se trata de "mapeador de portas DARPA para números de programas RPC". Este é o primeiro problema de segurança com o qual nos deparamos neste Como Fazer. A descrição de como evitar estes problemas pode ser encontrada na Seção D.6 (Seção de segurança), a qual eu repito, deve ser lida!

Inicializando o portmapper! Ele é chamado de `portmap` ou `rpc.portmap` e deve estar localizado no diretório `/usr/sbin` (em algumas máquinas ele é chamado de rpcbind). Pode-se inicializá-lo manualmente por hora, mas ele deverá ser inicializado toda vez que o sistema operacional for ativado, sendo então necessário editar os programas rc. Este programas são explicados mais detalhadamente na página de manual do processo init e usualmente estão localizados nos diretórios `/etc/rc.d`, `/etc/init.d` ou `/etc/rc.d/init.d`. Caso haja um programa chamado `inet` ou algo similar, este provavelmente será aquele que deve ser editado. Porém, como fazê-lo está além do escopo deste documento. Deve-se iniciar o programa portmap e verificar se ele está ativo através do comando `ps aux`. Encontrou-o? Ótimo.

D.3.4 Mountd e nfsd

Os próximos programas que necessitam ser executados são chamados *mountd* e *nfsd*. Porém, antes, é necessário editar outro arquivo. Desta vez o `/etc/exports`. Digamos que se deseje que o sistema de arquivos `/mn/parolin/local`, o qual está localizado na máquina `parolin`, seja disponibilizado para a máquina chamada `batel`. Deve-se então utilizar a seguinte configuração no arquivo `/etc/exports` em parolin:

```
/mn/parolin/local       batel(rw)
```

As linhas acima fornecem a batel acesso de leitura e gravação (rw) para `/mn/parolin/local`. Ao invés de `rw` poderíamos informar `ro`, o qual fornece acesso somente para leitura e é o padrão quando este parâmetro não é informado. Há diversas opções que podem ser utilizadas e que serão discutidas juntamente com aspectos de segurança mais adiante. Elas estão descritas nas páginas de manual on-line do comando `exports`, a qual deve ser lida ao menos uma vez na vida. Há ainda formas otimizadas de se incluir diversas máquinas no arquivo exports. Pode-se por exemplo, usar grupos de rede caso se esteja utilizando NIS (ou NYS) (NIS foi conhecido como YP) e especificar sempre um domínio com caracteres de generalização, ou sub-redes IP como máquinas que têm permissão para montar algo. Porém é necessário considerar que poderá ser possível obter acesso ao servidor de forma não autorizada caso se utilize autorizações tão genéricas.

Nota: o arquivo exports não tem a mesma sintaxe que em outros "Unices". Há uma seção específica neste Como fazer sobre arquivos `exports` em outros sistemas.

Agora que configuramos o mountd (ou talvez ele seja chamado `rpc.mountd`) e o nfsd (o qual pode ser chamado `rpc.nfsd`), ambos irão ler o arquivo exports.

Caso se edite o `/etc/exports` deve-se estar seguro de que os programas nfsd e mountd fiquem cientes destas alterações. A forma tradicional é através da execução do comando `exportfs`. Muitas distribuições Linux não possuem o programa exportfs. Caso este seja o seu caso, pode-se instalar o seguinte programa na máquina local:

```
#!/bin/sh
killall -HUP /usr/sbin/rpc.mountd
killall -HUP /usr/sbin/rpc.nfsd
echo re-exportando sistemas de arquivos
```

O programa acima deve ser salvo, por exemplo como `/usr/sbin/exportfs` e deve ser executado o comando `chmod a+rx exportfs`. Agora, toda vez que uma alteração for efetuada, deve-se executar o comando exportfs a seguir, com privilégios de superusuário.

Agora deve-se checar se mountd e nfsd estão sendo adequadamente executados. Inicialmente deve-se executar o comando `rpcinfo -p`. Ele deverá apresentar uma saída similar a:

```
   programa versão protocolo   porta
    100000    2         tcp    111  portmapper
    100000    2         udp    111  portmapper
    100005    1         udp    745  mountd
    100005    1         tcp    747  mountd
    100003    2         udp    2049 nfs
    100003    2         tcp    2049 nfs
```

Como se pode perceber, o portmapper anunciou os seus serviços, assim como mountd e nfsd.

Caso se obtenha uma mensagem similar a `rpcinfo: não foi possível contatar o portmapper: RPC: Erro no sistema remoto - Conexão recusada` ou algo similar, possivelmente o portmapper não esteja sendo executado. Caso se obtenha uma mensagem similar a `Nenhum programa remoto registrado` então, ou o portmapper não deseja falar com a máquina local ou existe algum erro. Pode-se finalizar nfsd, mountd e o portmapper e tentar reiniciá-los nesta ordem novamente.

Após verificar os serviços disponíveis segundo o portmapper, pode-se fazer uma checagem através do comando ps. O portmapper continuará a reportar um serviço, mesmo após o programa responsável ter sido finalizado com erro, por exemplo. Então um comando ps poderá ser a maneira mais simples de descobrir que programas estão efetivamente sendo executados.

Evidentemente, será necessário modificar os arquivos rc do sistema para inicializar o mountd e o nfsd, assim como o portmapper, quando o sistema operacional for carregado. É muito provável que estes programas já existam na máquina local e que se deva somente descomentar as seções adequadas ou ativá-los nos níveis de execução corretos.

Páginas de manual on-line que já devem ter sido visitadas até agora: portmap, mountd, nfsd, e exports.

Bem, caso tudo tenha sido feito exatamente como foi descrito aqui, já temos à disposição todo o conjunto de ferramentas necessárias para iniciar um cliente NFS.

D.4 Configurando um cliente NFS

Inicialmente é necessário ter um kernel com o suporte a sistemas de arquivo NFS compilado ou como um módulo. Isso deve ser configurado antes da compilação do kernel. Caso não se tenha feito isto, por favor verifique o Como Fazer - Kernel para instruções sobre como proceder. Caso se esteja utilizando uma boa distribuição (como o Conectiva Linux) e nunca se tenha lidado com o kernel ou módulos, nfs está magicamente à sua disposição.

Pode-se agora, na linha de comandos como superusuário, informar o comando de montagem apropriado e o sistema de arquivos estará disponível. Continuando com nosso exemplo anterior, desejamos montar `/mn/parolin/local` a partir de parolin. Isso deve ser feito através do seguinte comando:

```
mount -o rsize=1024,wsize=1024 parolin:/mn/parolin/local /mnt
```

(Retornaremos posteriormente às opções rsize e wsize). O sistema de arquivos está agora disponível sob `/mnt` e pode-se acessá-lo através do comando `cd`, assim como verificar o seu conteúdo através do comando `ls` e observar os arquivos individualmente. Pode-se perceber que ele não é tão rápido quanto um sistema local, mas muito mais amigável que o uso do ftp. Se, ao invés de montar um sistema de arquivos, o comando mount apresente uma mensagem de erro como `mount:parolin:/mn/parolin/local falhou, razão fornecida pelo` `servidor: Permissão negada` , então o arquivo exports contém algum erro. Caso ele informe `mount clntudp_create: RPC: Programa não registrado` isso significa que os programas nfsd ou mountd não estão sendo executados no servidor.

Para desmontar o sistema de arquivos basta digitar:

```
umount /mnt
```

Para que um sistema de arquivos nfs seja montado na inicialização do sistema operacional, deve-se editar o arquivo `/etc/fstab` da forma usual. No caso de nosso exemplo, deve-se adicionar a seguinte linha:

```
# dispositivo      pto.montagem    tipo_sist_arqs  opções    dump ordem verif.
...
parolin:/mn/parolin/local  /mnt    nfs  rsize=1024,wsize=1024   0    0
...
```

Bem, parece que isso é tudo. Quase. Continue a leitura por favor.

D.4.1 Opções de Montagem

Há algumas opções que devem ser consideradas. Elas definem a forma como o cliente NFS lida com uma queda do servidor ou da rede. Um dos aspectos mais interessantes sobre NFS é que ele trata destas situações com elegância, desde que o cliente esteja corretamente configurado. Há dois tipos distintos de parâmetros de tratamento de falhas:

soft

O cliente NFS reporta um erro ao processar o acesso a um arquivo localizado em um sistema de arquivos montado via NFS. Alguns programas podem lidar com isto com compostura, outros não. Esta opção não é recomendada.

hard

O programa que acessa um arquivo em um sistema de arquivos montado via NFS irá travar sempre que o servidor não responder. O processo não pode ser interrompido ou finalizado a menos que se tenha especificado `intr`. Quando o servidor NFS estiver novamente ativo, o programa irá continuar a partir do ponto de onde tenha parado. Isso é provavelmente o que se deseja. Recomendamos o uso do parâmetro `hard,intr` em todos os sistemas de arquivos montados via NFS.

A partir do exemplo anterior, esta seria a entrada no arquivo fstab:

```
# dispositivo  pto.montagem tipo_sist_arqs  opções         dump ordem verif.
...
parolin:/mn/parolin/local  /mnt    nfs  rsize=1024,wsize=1024,hard,intr 0 0
...
```

D.4.2 Otimizando NFS

Normalmente, caso as opções rsize e wsize sejam especificadas, o NFS irá ler e gravar blocos de 4096 e 8172 bytes, respectivamente. Algumas combinações de kernel do Linux e placas de rede não podem lidar com blocos grandes e não podem ser otimizadas. Então vamos tentar descobrir como encontrar os parâmetros rsize e wsize que funcionem da maneira mais otimizada possível. É possível testar a velocidade das opções com um simples comando. Dado o comando mount conforme descrito acima, logo temos acesso de gravação ao disco, podendo executar um teste de performance de gravação seqüencial:

```
time dd if=/dev/zero of=/mnt/testfile bs=16k count=4096
```

Este comando criará um arquivo de 64 Mb de bytes zerados (que deve ser grande o suficiente para que o cache não altere significativamente a performance. Pode ser usado um arquivo maior caso o sistema local tenha muita memória). Isso pode ser feito algumas vezes (5-10?), para que se possa ter uma média bem fundamentada. Neste casos, o importante é medir o tempo de "relógio" e o tempo efetivamente gasto na conexão. Após, pode-se testar a performance da leitura ao se ler o arquivo de volta:

```
time dd if=/mnt/testfile of=/dev/null bs=16k
```

Isso pode ser feito algumas vezes. Após deve-se executar o comando mount e umount novamente com tamanhos maiores em rsize e wsize. Eles devem ser provavelmente múltiplo de 1024 e não maiores que 16384, uma vez que este é o tamanho máximo do NFS versão 2. Exatamente após a montagem de um tamanho maior, acesse o sistema de arquivos montado através do comando cd e explore-o através do comando ls, para estar seguro que ele está funcionando perfeitamente. Caso os parâmetros rsize/wsize sejam muito grandes, os sintomas não são *muito* óbvios. Um típico sintoma é uma lista incompleta dos arquivos produzida pelo comando ls e nenhuma mensagem de erro. Ou ao se ler um arquivo ele falha misteriosamente, sem mensagens de erro. Após definir que os parâmetros rsize/wsize funcionam perfeitamente deve-se executar os testes de performance. SunOS e Solaris têm a reputação de funcionar muito melhor com blocos de 4096 bytes.

kernels mais recentes do Linux (desde o 1.3) executam a leitura antecipada para rsizes maiores ou iguais ao tamanho de página da máquina. Em máquinas Intel o tamanho de página é de 4.096 bytes. A leitura adiantada aumenta *significativamente* a performance de leitura do NFS. Ou seja, sempre que possível deve-se usar o rsize de 4.096 bytes em máquinas Intel.

Lembre-se de editar o arquivo `/etc/fstab` com os valores de rsize/wsize encontrados.

Uma sugestão para incrementar a performance de gravação do NFS é desabilitar o sincronismo de gravação do servidor. A especificação NFS indica que a gravação NFS solicitada não pode ser considerada finalizada antes dos dados serem gravados em um meio não volátil (normalmente o disco rígido). Isso restringe a performance de gravação de alguma forma, enquanto que gravações assíncronas irão aumentar a velocidade do NFS. O servidor Linux nfsd nunca faz gravações síncronas, primeiro porque a própria implementação do sistema de arquivos não o faz, mas em servidores com sistemas operacionais diferentes isso pode aumentar a performance através do seguinte parâmetro no arquivo exports:

```
/dir    -async,access=linuxbox
```

ou algo similar. Por favor verifique a página de manual on-line da máquina em questão. Cabe salientar que esta opção aumenta o risco de perda de dados no caso de algum problema ocorrer antes da sua efetiva gravação.

D.5 NFS Sobre Linhas de Baixa Velocidade

Linhas de baixa velocidade incluem modems, ISDN e praticamente todas as ligações de longa distância possíveis.

Esta seção é baseada no conhecimento dos protocolos usados, mas não em experiências de campo. Meu computador pessoal esteve inativo por um longo tempo e caso você tenha alguma experiência adicional, por favor informe.

A primeira coisa para se lembrar sobre NFS é que ele é um protocolo lento e tem ainda um alto número de informações adicionais. Usar NFS é o mesmo que se utilizar o kermit para transferir arquivos. É *lento*. Praticamente qualquer coisa é mais rápida que NFS. FTP, HTTP, rcp, ssh por exemplo.

Ainda quer tentar? Ok.

Os parâmetros padrões do NFS são para linhas rápidas com baixa latência. Caso se esteja usando estes parâmetros para linhas de alta latência, certamente o NFS reportará alguns erros, encerrará operações, imaginará que arquivos são menores do que eles sejam na realidade e agirá estranhamente em alguns casos.

A primeira coisa a *não* fazer é usar a opção de montagem `soft`. Ela provocará ultrapassagem dos tempos de espera e retornos de erro para o software, o qual, na maior parte do tempo, não saberá lidar corretamente com eles. Essa é uma maneira rápida de se obter erros misteriosos. Ao invés disso deve ser usada a opção de montagem `hard`, que gera infinitas tentativas em caso de estouro de tempo de espera, ao invés de encerrar a solicitação, independentemente do que o software deseja fazer. Isso será realmente necessário nestes casos.

A próxima providência é mudar as opções de montagem timeo e retrans. Elas são descritas na página de manual do nfs(5), mas segue aqui uma cópia:

```
timeo=n
O número de décimos de segundo antes de enviar a primeira retransmissão
após findo o tempo de espera de uma RPC. O valor padrão é de 7 décimos
de segundo. Após a primeira espera, o tempo é dobrado após cada espera
sem respostas, até um máximo de 60 segundos ou um número máximo de
retransmissões ser atingido. Então, caso o sistema de arquivos esteja
montado com a opção hard, cada novo tempo de espera começa com o dobro
do tempo da anterior, novamente dobrando a cada retransmissão. O tempo
máximo de espera é sempre de 60 segundos. Uma melhor performance pode
ser atingida ao se incrementar o tempo de espera, quando se está
montando sistemas sobre uma rede com muito tráfego, utilizando-se
servidores lentos ou usando o sistema através de diversos roteadores
e portas de entrada.

retrans=n
```

```
O número de tempo limite e retransmissões que devem ocorrer antes que
um alarme de tempo de resposta seja acionado. O padrão é de 3
ocorrências. Quando um alarme de tempo de espera maior ocorre, a
operação é interrompida ou uma mensagem de ``servidor não está
respondendo'' é apresentada no console.
```

Em outras palavras: se uma resposta não for recebida no tempo de espera de 0,7 segundos (700 ms), o cliente NFS irá repetir e dobrar o tempo de espera para 1,4 segundos. Caso a resposta não seja recebida neste tempo, a requisição será enviada novamente com um tempo de espera alterado para 2,8 segundos.

A velocidade da linha pode ser medida com um ping com os mesmos parâmetros das opções rsize/wsize.

```
$ ping -s 8192 lugulbanda
PING lugulbanda.uio.no (129.240.222.99): 8192 data bytes
8200 bytes from 129.240.222.99: icmp_seq=0 ttl=64 time=15.2 ms
8200 bytes from 129.240.222.99: icmp_seq=1 ttl=64 time=15.9 ms
8200 bytes from 129.240.222.99: icmp_seq=2 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=3 ttl=64 time=14.9 ms
8200 bytes from 129.240.222.99: icmp_seq=4 ttl=64 time=15.0 ms

--- lugulbanda.uio.no ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14.9/15.1/15.9 ms
```

O parâmetro time aqui mostra quanto tempo o pacote ping levou para chegar a e retornar da máquina denominada lugulbanda. 15ms é bastante rápido. Sobre uma linha de 28.800 bps pode-se esperar algo como 4000-5000ms e, caso a linha esteja carregada, um tempo maior, chegando facilmente ao dobro. Quando o tempo é muito alto nos referimos como uma linha de alta latência. Geralmente para pacotes maiores e linhas mais carregadas, a latência tende a aumentar. Deve-se aumentar o parâmetro timeo para se adequar a esta realidade. Deve-se atentar que a latência tende a aumentar ainda mais quando se usa a linha para outros serviços como por exemplo FTP e NFS simultaneamente. Neste caso deve-se medir as respostas do comando ping ao se efetuar transferências de arquivos.

D.6 Segurança e NFS

Não me considero um expert em segurança de computadores. Porém existem algumas *sugestões* importantes. É importante ressaltar que esta não é uma lista completa de todos os aspectos relacionados com segurança e caso se imagine que implementando somente estes não se poderá ter qualquer problema relacionado com o tema segurança, por favor me envie seu email que eu tenho uma ponte e desejo vendê-la. :-)

Esta seção é provavelmente fora de questão caso se esteja em uma rede *fechada*, onde todos os usuários são conhecidos e ninguém que não seja confiável pode acessar a rede, ou seja não há forma de discar para a rede e não há forma de conectar-se a outras redes onde existam usuários não confiáveis. Isso soa como paranóia? Não sou paranóico. Isso é somente um aviso *básico* de segurança. E lembre-se, o que aqui for dito é somente uma base para o tema. Um site *seguro* necessita de um administrador diligente e com conhecimento que consiga encontrar informações sobre problemas de segurança correntes e potenciais.

NFS é um problema básico, no qual o cliente, caso não seja informado do contrário, irá confiar no servidor NFS e vice-versa. Isso pode ser ruim, pois se a senha do superusuário no servidor NFS for quebrada, a senha dos superusuários dos clientes também o será com relativa facilidade e vice-versa. Há algumas estratégias para se evitar isso, as quais mencionaremos adiante.

Uma leitura obrigatória são os avisos do CERT sobre NFS, onde muitos dos textos lidam com conselhos sobre segurança. Veja em `ftp.cert.org/01-README` uma lista atualizada dos avisos CERT. Aqui estão alguns dos relacionados com NFS:

```
CA-91:21.SunOS.NFS.Jumbo.and.fsirand                    12/06/91
Vulnerabilidade preocupa Sun Microsystems, Inc. (Sun) Sistema de
Arquivos em Rede (NFS) e o programa fsirand. Estas vulnerabilidades
afetam o SunOS versões 4.1.1, 4.1 e 4.0.3 em todas as arquiteturas.
Atualizações estão disponíveis para SunOS 4.1.1. Uma atualização
inicial para o NFS SunOS 4.1 está também disponível. A Sun irá
disponibilizar atualizações completas para as versões SunOS 4.1
e SunOS 4.0.3 em uma versão posterior.

CA-94:15.NFS.Vulnerabilidades                           12/19/94
Este aviso descreve as medidas de segurança a serem tomadas para
evitar diversas vulnerabilidades do Sistema de Arquivos em Rede
(NFS). Os avisos foram gerados devido ao incremento do
```

```
comprometimento de superusuários através de invasores usando
ferramentas que exploram estas falhas.

CA-96.08.pcnfsd                                       04/18/96
Este aviso descreve a vulnerabilidade do programa pcnfsd (também
conhecido como rpc.pcnfsd). Uma atualização está incluída.
```

D.6.1 Segurança do Cliente

No cliente, podemos decidir se desejamos ou não confiar no servidor através de algumas opções na montagem. Por exemplo, é possível proibir programas suid a funcionarem em sistemas de arquivos NFS através da opção `nosuid`. Esta pode ser uma boa idéia que deve ser considerada no uso de todos os discos montados via NFS. Esta opção indica que o superusuário do servidor não pode fazer um programa com características de suid no sistema de arquivos, o que possibilitaria que ele acessasse o cliente como um usuário normal e usasse o programa suid-superusuário para tornar-se superusuário na máquina cliente. Deve-se proibir também a execução de arquivos em sistemas de arquivos montados, através da opção `noexec`. Porém isso pode ser impraticável por vezes, assim como o `nosuid` uma vez que um sistema de arquivos normalmente contém *alguns* programas que necessitam ser executados. Estes parâmetros podem ser informados na coluna opções, juntamente com os parâmetros `rsize` e `wsize`, separados por vírgulas.

D.6.2 Segurança no Servidor: nfsd

No servidor pode-se decidir sobre a possibilidade de confiar na conta do superusuário do cliente. Isso é definido através do uso da opção root_squash no arquivo exports:

```
/mn/parolin/local batel(rw,root_squash)
```

Agora caso um usuário com número de identificação igual a 0 (UID) tentar acessar (ler, gravar, remover) o sistema de arquivos, o servidor substituirá o UID pela identificação de conta "nobody" (ninguém). Isso faz com que o superusuário da máquina cliente não possa acessar arquivos ou executar mudanças autorizadas somente para o superusuário do servidor. Isso é aconselhável e provavelmente devase usar `root_squash` em todos os sistemas exportados. "Porém o superusuário cliente pode ainda usar o comando 'su' para tornar-se qualquer outro usuário e acessar e alterar quaisquer arquivos", é o que se pode pensar à primeira vista. A resposta é: sim, é desta forma que as coisas funcionam com Unix e NFS. Isso traz uma implicação importante: todos os binários e arquivos importantes devem pertencer ao superusuário `root`, e não a `bin` ou outra conta diferente, uma vez que somente a conta do superusuário da máquina cliente pode acessar a conta do superusuário no servidor. Na página de manual on-line do nfsd há diversas outras opções squash que podem ser usadas, então o administrador deve decidir quem não pode ter acesso à conta do superusuário. Existem opções de se evitar o uso de faixas ou de qualquer UID ou GID que se deseje. Isso está descrito na mesma página de manual.

root_squash é na verdade o padrão do nfsd do Linux. Para permitir acesso a um sistema de arquivos como superusuário deve-se usar a opção `no_root_squash`.

Outro aspecto importante é garantir que o nfsd verifique que todas as requisições sejam provenientes de uma porta autorizada. Caso se aceite requisições de qualquer porta antiga de um usuário sem privilégios especiais, torna-se simples acessar o sistema de arquivos através da Internet, por exemplo. Basta usar o protocolo nfs e identificar-se como qualquer usuário que se deseje. Ooopss. O nfsd do Linux realiza esta verificação por padrão, em outros sistemas operacionais deve-se habilitar esta opção. Isso deverá estar descrito na página de manual do servidor nfs do sistema.

Um dado adicional. Nenhum sistema de arquivos deve ser exportado para o 'localhost' ou 127.0.0.1. Acredite em mim.

D.6.3 Segurança no Servidor: o portmapper

O portmapper básico em combinação com o nfsd tem um problema de desenho que torna possível obter-se arquivos em servidores NFS sem a necessidade de quaisquer privilégios. Felizmente o portmapper do Linux é relativamente seguro contra este tipo de ataque, o que pode ser evitado através da configuração de uma lista de acessos em dois arquivos.

Inicialmente editaremos o `/etc/hosts.deny`. Ele deverá conter a seguinte linha:

```
portmap: ALL
```

através da qual o acesso será bloqueado para *todos* os clientes. Isto talvez seja um pouco drástico, então podemos tornar as definições um pouco mais maleáveis através da edição do arquivo `/etc/hosts.allow`. Inicialmente é necessário definir o que será colocado nele. Ele contém basicamente uma lista de todas as máquinas que podem acessar o portmapper local. Em um sistema Linux há normalmente poucas máquinas que necessitem este tipo de acesso, qualquer que seja a razão. O portmapper administra os programas nfsd, mountd, ypbind/ypserv, pcnfsd e serviços 'r' como ruptime e rusers. Todas as máquinas que necessitam acessar os serviços da máquina local devem ter permissão para tanto. Digamos que o endereço da máquina local seja 129.240.223.254 e que ela está conectada à sub-rede 129.240.223.0, a qual deve ter acesso à máquina local (em caso de dúvida verifique o Como Fazer - Redes para refrescar a memória sobre estes conceitos). Para tanto basta digitar:

```
portmap: 129.240.223.0/255.255.255.0
```

no arquivo `hosts.allow`. Este é o mesmo endereço de rede fornecido para o comando route e a máscara de sub-rede informada no ifconfig. No dispositivo `eth0` desta máquina `ifconfig` mostraria:

```
...
eth0      Link encap:10Mbps Ethernet  HWaddr 00:60:8C:96:D5:56
          inet addr:129.240.223.254  Bcast:129.240.223.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:360315 errors:0 dropped:0 overruns:0
          TX packets:179274 errors:0 dropped:0 overruns:0
          Interrupt:10 Base address:0x320
...
```

e `netstat -rn` apresentaria

```
Tabela de Roteamento do Kernel
Destinação     Cam.Padrão        Máscara    Indics Métrica Ref  Uso   Iface
...
129.240.223.0   0.0.0.0        255.255.255.0   U     0      0   174412 eth0
...
```

o endereço de rede na primeira coluna.

Os arquivos `hosts.deny` e `hosts.allow` são descritos nas página de manual de mesmo nome.

IMPORTANTE: *não* coloque *nada* exceto *ENDEREÇOS IP* nas linhas do portmap nestes arquivos. Pesquisas por nomes podem indiretamente causar atividade do portmap o qual acionará a pesquisa de nomes de máquinas a qual indiretamente irá causar atividade no portmap, o qual....

As sugestões acima certamente deixarão o servidor mais seguro. As questões restantes residem em alguém que tenha descoberto a senha do superusuário (ou inicializando um MS-DOS) em uma máquina confiável e usando este privilégio para enviar requisições a partir de uma porta segura como qualquer outro usuário real.

D.6.4 NFS e Firewalls

É uma boa idéia proteger o servidor nfs e as portas portmap no roteador ou no firewall. O nfsd opera normalmente na porta 2049, nos protocolos udp e tcp. O portmapper na porta 111, tcp e udp e o mountd na porta 745 e 747, tcp e udp. Estas informações devem ser checadas através do comando `rpcinfo -p`.

Por outro lado, caso se deseje permitir o acesso ao NFS através de um firewall, há opções em programas mountd e nfsd mais recentes que permitem o uso específico e não padronizado de portas que podem ser abertas através de um firewall.

D.6.5 Resumo

Caso se utilize hosts.allow/deny, root_squash, nosuid e funcionalidades de portas privilegiadas para os softwares portmapper e nfs pode-se evitar muitos dos problemas atualmente conhecidos sobre segurança e pode sentir-se quase seguro sobre *estes problemas*, no mínimo. Porém há mais ainda: quando um intruso tem acesso à rede, ele pode incluir comandos estranhos nos arquivos `.forward` ou nos arquivos de mensagens, quando `/home` ou `/var/spool/mail` são montados via NFS. Pela mesma razão, nunca se deve dar acesso às chaves privadas PGPP sobre nfs. Ou no mínimo, deve-se saber dos riscos envolvidos. Pelo menos isso você já sabe.

NFS e o portmapper criam um subsistema complexo e adicionalmente há problemas que são descobertos e que devem ser solucionados, além da necessidade de se ter em mente o desenho básico de implementação a ser usado. Para estar ciente do que está ocorrendo pode-se acessar o grupo de notícias `comp.os.linux.announce` e `comp.security.announce` eventualmente.

D.7 Pontos de Verificação de Montagem

Esta seção é baseada na lista de verificação de problemas da IBM Corp. Meus agradecimentos a eles por tornarem ela disponível para este Como Fazer. Caso o leitor esteja com algum problema em montar sistemas de arquivos NFS, por favor consulte esta lista. Cada item descreve um problema específico e a sua solução.

1. O sistema de arquivos não foi exportado, ao menos para a máquina cliente em questão.

   Solução: Incluí-lo no arquivo exports.

2. A resolução de nomes não confere com a lista de exports.

   Por exemplo: a lista em export indica uma exportação para `johnmad`, mas o nome `johnmad` é resolvido como `johnmad.austin.ibm.com`, fazendo com que a permissão de montagem seja negada.

   Solução: Exportar em ambos os formatos de nomes.

   Isso pode ocorrer ainda quando o cliente tem 2 interfaces com nomes diferentes para cada um dos dois dispositivos e o comando export especifica somente um deles.

   Solução: Exporta para ambas as interfaces.

   Isso pode ocorrer também quando o servidor não consegue executar um chamada lookuphostbyname ou lookuphostbyaddr (são funções da biblioteca) no cliente. Esteja seguro de que o cliente pode executar `máquina <nome>`; `máquina <endereço_ip>` e que ambos mostram a mesma máquina.

   Solução: ajustar a resolução de nomes no cliente.

3. O sistema de arquivos foi montado após a inicialização do NFS (no servidor). Neste caso o sistema de arquivos está exportado sob um ponto de montagem.

   Solução: Desativar nfsd e reinicializá-lo.

   Nota: os clientes que tenham pontos de montagem sob sistemas de arquivos terão problemas no acesso após a reinicialização.

   Nestes casos é recomendada a execução do comando mount -a, como superusuário, na máquina cliente.

4. As datas estão estranhamente diferentes em ambas as máquinas (o que pode gerar inconsistências com os arquivos).

Solução: Ajustar as datas.

O autor do Como Fazer sugere o uso do NTP para sincronismo de relógios. Uma vez que existem restrições de exportação do NTP para fora dos EUA, pode-se obter uma cópia em uma distribuição Linux ou em `ftp://ftp.hacktic.nl/pub/replay/pub/linux` ou em um site espelho.

5. O servidor não aceita uma montagem de um usuário presente em mais de 8 grupos.

   Solução: diminuir o número de grupos aos quais o usuário pertença ou alterar o usuário na montagem.

D.8 FAQ

Esta é uma seção de perguntas e respostas. Muito do que está contido aqui foi escrito por Alan Cox.

1. Obtive uma série de erros de manipulação de arquivos nfs ao usar o Linux como servidor.

   Isso é causado por uma antiga versão do nfsd. Está corrigida a partir da versão nfs-server2.2beta16.

2. Ao tentar montar um sistema de arquivos, surge a mensagem:

   ```
   não foi possível registrar-se no portmap: erro do sistema no envio
   ```

   Provavelmente se está utilizando o sistema da Caldera. Há um problema com os programas rc. Por favor entre em contato com eles para correção do problema.

3. Por que não é possível executar um arquivo após copiá-lo para o servidor NFS?

   A questão reside no fato do nfsd criar caches de manipulação de arquivos por questões de performance (lembre-se que ele é executado em um espaço de usuário). Enquanto nfsd tem um arquivo aberto (como no caso em que ele esteja sendo gravado), o kernel não permite a sua execução. Os programas NFSd a partir de 95 liberam os arquivos após alguns segundos, já versões mais antigas podem levar dias.

4. Os arquivos NFS estão todos com permissões somente de leitura.

   O padrão do servidor NFS Linux é somente fornecer permissões de leitura para arquivos montados. O arquivo `/etc/exports` deve ser alterado caso se deseje algo diferente.

5. Existe um sistema de arquivos montado a partir de um servidor NFS Linux e enquanto o comando ls trabalha, a leitura e gravação de arquivos não funcionam.

   Em versões antigas do Linux, deve-se montar um servidor NFS com os parâmetros `rsize=1024`,`wsize=1024`.

6. Ao montar a partir de um servidor NFS Linux com um bloco de tamanho entre 3500-4000 ele trava regularmente.

   Bem...não faça mais isso!

7. Pode Linux executar NFS sobre TCP?

   Não, no momento.

8. Ao se montar a partir de uma máquina Linux, obtém-se inúmeros erros.

   Esteja certo de que os usuários estarão presentes em no máximo 8 grupos. Servidores mais antigos requerem isso.

9. Ao reinicializar a máquina, ela algumas vezes trava ao tentar desmontar um servidor NFS.

   Não desmonte servidores NFS ao reinicializar ou desligar. Simplesmente ignore-os. Isso não irá machucar ninguém. O comando é `umount -avt nonfs`.

10. Clientes Linux NFS são muito lentos ao se tentar gravar em sistemas Sun e BSD.

    NFS executa gravações síncronas (o que pode ser desabilitado caso não haja nenhum grande problema em se perder algum dado). Kernels derivados do BSD tendem a trabalhar mal com pequenos blocos. Porém ao se gravar blocos de 4 Kb de dados a partir de uma máquina Linux, usando pacotes de 1 Kb, faz com que o Linux use a rotina BSD na seguinte forma:

    ```
    ler página de 4K;
    altera para 1K;
    gravar 4K no disco rígido;
    ler página de 4;K
    ```

```
altera para 1K;
gravar 4K no disco rígido;
etc..
```

D.9 Exportando Sistemas de Arquivos

A forma de exportar sistemas de arquivos com NFS não é totalmente consistente quando utilizada entre plataformas distintas. No caso Linux e Solaris 2 são distintos. Esta seção lista superficialmente a forma de como executar esta tarefa na maioria dos sistemas. Caso o seu sistema não esteja aqui descrito, deve-se checar as páginas de manual do sistema em questão. Palavras chaves são: nfsd, ferramentas de administração de sistemas, programas rc, programas de inicialização, seqüência de inicialização, /etc/exports, exportfs. Usaremos como exemplo nesta seção como exportar /mn/parolin/local para a máquina batel com permissões de leitura e gravação.

D.9.1 IRIX, HP-UX, Digital-UNIX, Ultrix, SunOS 4 (Solaris 1), AIX

Estes sistemas usam o formato tradicional de exportação. Em `/etc/exports` deve ser incluído:

```
/mn/parolin/local -rw=batel
```

A documentação completa de `exports` pode ser encontrada na página de manual. Após editar este arquivo deve ser executado o comando `exportfs -av` para exportar os sistemas de arquivos.

Em alguns sistemas a linha anterior pode ter o seguinte formato:

```
/mn/parolin/local batel
```

ou mesmo algo como:

```
/mn/parolin/local rw=batel
```

Recomenda-se a forma usual. O risco da próxima versão do `exportfs` ser diferente é grande e algumas coisas podem parar de funcionar.

D.9.2 Solaris 2

A Sun reinventou completamente a roda quando fez o Solaris 2, já que ele é completamente diferente de todos os outros sistemas operacionais. Deve-se editar o arquivo `/etc/dfs/dfstab`. Neste arquivo são colocados os comandos compartilhados, conforme documentado na página de manual share (1 Mb). A sintaxe será algo como:

```
share -o rw=batel -d "Parolin  Local" /mn/parolin/local
```

Após a edição deve-se executar o programa `shareall` para exportar o sistema de arquivos.

D.10 PC-NFS

Não se deve rodar o PC-NFS. Neste caso o melhor é executar o samba.

Desculpe, mas não conheço nada sobre o PC-NFS. Caso alguém queira colaborar, por favor envie-me algumas informações e elas serão incluídas.

Apêndice E

Licença Pública GNU

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

This is an unofficial translation of the GNU General Public License into Portuguese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL – only the original English text of the GNU GPL does that. However, we hope that this translation will help Portuguese speakers understand the GNU GPL better.

E.1 Introdução

As licenças de muitos softwares são desenvolvidas para cercear a liberdade de uso, compartilhamento e mudanças. A GNU Licença Pública Geral ao contrário, pretende garantir a liberdade de compartilhar e alterar softwares de livre distribuição - tornando-os de livre distribuição também para quaisquer usuários. A Licença Pública Geral aplica-se à maioria dos softwares da Free Software Foundation e a qualquer autor que esteja de acordo em utilizá-la (alguns softwares da FSF são cobertos pela GNU Library General Public License).

Quando nos referimos a softwares de livre distribuição, referimo-nos à liberdade e não ao preço. Nossa Licença Pública Geral foi criada para garantir a liberdade de distribuição de cópias de softwares de livre distribuição (e cobrar por isso caso seja do interesse do distribuidor), o qual recebeu os códigos fonte, o qual pode ser alterado ou utilizado em parte em novos programas.

Para assegurar os direitos dos desenvolvedores, algumas restrições são feitas, proibindo a todas as pessoas a negação desses direitos ou a solicitação de sua abdicação. Essas restrições aplicam-se ainda a certas responsabilidades sobre a distribuição ou modificação do software.

Por exemplo, ao se distribuir cópias de determinado programa, por uma taxa determinada ou gratuitamente, deve-se informar sobre todos os direitos incidentes sobre esse programa, assegurando-se que os fontes estejam disponíveis assim como a Licença Pública Geral GNU.

A proteção dos direitos envolve dois passos: (1) copyright do software e (2) licença que dá permissão legal para cópia, distribuição e/ou modificação do softwares.

Ainda para a proteção da FSF e do autor é importante que todos entendam que não há garantias para softwares de livre distribuição. Caso o software seja modificado por alguém e passado adiante, este software não mais refletirá o trabalho original do autor não podendo portanto ser garantido por aquele.

Finalmente, qualquer programa de livre distribuição é constantemente ameaçado pelas patentes de softwares. Buscamos evitar o perigo de que distribuidores destes programas obtenham patentes individuais, tornado-se seus donos efetivos. Para evitar isso foram feitas declarações expressas de que qualquer solicitação de patente deve ser feita permitindo o uso por qualquer indivíduo, sem a necessidade de licença de uso.

Os termos e condições precisas para cópia, distribuição e modificação seguem abaixo:

E.1.1 Termos e Condições para Cópia, Distribuição e Modificação

0. Esta licença se aplica a qualquer programa ou outro trabalho que contenha um aviso colocado pelo detentor dos direitos autorais dizendo que aquele poderá ser distribuído nas condições da Licença Pública Geral. O Programa, abaixo refere-se a qualquer software ou trabalho e a um trabalho baseado em

um Programa e significa tanto o Programa em si como quaisquer trabalhos derivados de acordo com a lei de direitos autorais, o que significa dizer, um trabalho que contenha o Programa ou uma parte deste, na sua forma original ou com modificações ou traduzido para uma outra língua (tradução está incluída sem limitações no termo `modificação`).

Atividades distintas de cópia, distribuição e modificação não estão cobertas por esta Licença, estando fora de seu escopo. O ato de executar o Programa não está restringido e a saída do Programa é coberta somente caso seu conteúdo contenha trabalhos baseados no Programa (independentemente de terem sidos gerados pela execução do Programa). Se isso é verdadeiro depende das funções executadas pelo Programa.

1. O código fonte do Programa, da forma como foi recebido, pode ser copiado e distribuído, em qualquer mídia, desde que seja providenciado um aviso adequado sobre os copyrights e a negação de garantias, e todos os avisos que se referem à Licença Pública Geral e à ausência de garantias estejam inalterados e que qualquer produto oriundo do Programa esteja acompanhado desta Licença Pública Geral.

   É permitida a cobrança de taxas pelo ato físico de transferência ou gravação de cópias, e podem ser dadas garantias e suporte em troca da cobrança de valores.

2. Pode-se modificar a cópia ou cópias do Programa de qualquer forma que se deseje, ou ainda criar-se um trabalho baseado no Programa, e copiá-la e distribuir tais modificações sob os termos da seção 1 acima e do seguinte:

   a. Deve existir aviso em destaque de que os dados originais foram alterados nos arquivos e as datas das mudanças;

   a. Deve existir aviso de que o trabalho distribuído ou publicado é, de forma total ou em parte derivado do Programa ou de alguma parte sua, e que pode ser licenciado totalmente sem custos para terceiros sob os termos desta Licença.

   a. Caso o programa modificado seja executado de forma interativa, é obrigatório, no início de sua execução, apresentar a informação de copyright e da ausência de garantias (ou de que a garantia corre por conta de terceiros), e que os usuários podem redistribuir o programa sob estas condições, indicando ao usuário como acessar esta Licença na sua íntegra.

Esses requisitos aplicam-se a trabalhos de modificação em geral. Caso algumas seções identificáveis não sejam derivadas do Programa, e podem ser consideradas como partes independentes, então esta Licença e seus Termos não se aplicam àquelas seções quando distribuídas separadamente. Porém ao distribuir aquelas seções como parte de um trabalho baseado no Programa, a distribuição como um todo deve conter os termos desta Licença, cujas permissões estendem-se ao trabalho como um todo, e não a cada uma das partes independentemente de quem os tenha desenvolvido.

Mais do que tencionar contestar os direitos sobre o trabalho desenvolvido por alguém, esta seção objetiva propiciar a correta distribuição de trabalhos derivados do Programa.

Adicionalmente, a mera adição de outro trabalho ao Programa, porém não baseado nele nem a um trabalho baseado nele, a um volume de armazenamento ou media de distribuição não obriga a utilização desta Licença e de seus termos ao trabalho.

3. São permitidas a cópia e a distribuição do Programa (ou a um trabalho baseado neste) na forma de código objeto ou executável de acordo com os termos das Seções 1 e 2 acima, desde que atendido o seguinte:

    a. Esteja acompanhado dos códigos fonte legíveis, os quais devem ser distribuídos na forma da Seções 1 e 2 acima, em mídia normalmente utilizada para manuseio de softwares ou

    b. Esteja acompanhado de oferta escrita, válida por, no mínimo 3 anos, de disponibilizar a terceiros, por um custo não superior ao custo do meio físico de armazenamento, uma cópia completa dos códigos fonte em meio magnético, de acordo com as Seções 1 e 2 acima.

    c. Esteja acompanhada com a mesma informação recebida em relação à oferta da distribuição do código fonte correspondente. (esta alternativa somente é permitida para distribuições não comerciais e somente se o programa recebido na forma de objeto ou executável tenha tal oferta, de acordo com a subseção 2 acima).

O código-fonte é a melhor forma de produzirem-se alterações em um trabalho. Códigos-fonte completos significam todos os fontes de todos os módulos, além das definições de interfaces associadas, arquivos, scripts utilizados na compilação e instalação do executável. Como uma exceção, o código-fonte distribuído não poderá incluir alguns componentes que não se encontrem em seu escopo, tais como compilador, kernel, etc. para o sistema operacional onde o trabalho seja executado.

Caso a distribuição do executável ou objeto seja feita através de acesso a um determinado ponto, então oferta equivalente de acesso deve ser feita aos códigos fonte, mesmo que terceiros não sejam obrigados a copiarem os fontes juntos com os objetos simultaneamente.

4. Não é permitida a cópia, modificação, sublicenciamento ou distribuição do Programa, exceto sob as condições expressas nesta Licença. Qualquer tentativa de cópia, modificação, sublicenciamento ou distribuição do Programa é proibida, e os direitos descritos nesta Licença cessarão imediatamente. Terceiros que tenham recebido cópias ou direitos na forma desta Licença não terão seus direitos cessados desde que permaneçam dentro das cláusulas desta Licença.

5. Não é necessária aceitação formal desta Licença, apesar de que não haverá documento ou contrato que garanta permissão de modificação ou distribuição do Programa ou seus trabalhos derivados. Essas ações são proibidas por lei, caso não se aceitem as condições desta Licença. A modificação ou distribuição do Programa ou qualquer trabalho baseado neste implica na aceitação desta Licença e de todos os termos desta para cópia, distribuição ou modificação do Programa ou trabalhos baseados neste.

6. Cada vez que o Programa seja distribuído (ou qualquer trabalho baseado neste), o recipiente automaticamente recebe uma licença do detentor original dos direitos de cópia, distribuição ou modificação do Programa objeto deste termos e condições. Não podem ser impostas outras restrições nos recipientes.

7. No caso de decisões judiciais ou alegações de uso indevido de patentes ou direitos autorais, restrições sejam impostas que contradigam esta Licença, estes não isentam da sua aplicação. Caso não seja possível distribuir o Programa de forma a garantir simultaneamente as obrigações desta Licença e outras que sejam necessárias, então o Programa não poderá ser distribuído.

   Caso esta Seção seja considerada inválida por qualquer motivo particular ou geral, o seu resultado implicará na invalidação geral desta licença na cópia, modificação, sublicenciamento ou distribuição do Programa ou trabalhos baseados neste.

   O propósito desta seção não é, de forma alguma, incitar quem quer que seja a infringir direitos reclamados em questões válidas e procedentes, e sim proteger as premissas do sistema de livre distribuição de software. Muitas pessoas têm feito contribuições generosas ao sistema, na forma de programas, e é necessário garantir a consistência e credibilidade do sistema, cabendo a estes e não a terceiros decidirem a forma de distribuição dos softwares.

Esta seção pretende tornar claro os motivos que geraram as demais cláusulas destas Licença.

8. Caso a distribuição do Programa dentro dos termos desta Licença tenha restrições em algum País, quer por patentes ou direitos autorais, o detentor original dos direitos autorais do Programa sob esta Licença pode adicionar explicitamente limitações geográficas de distribuição, excluindo aqueles Países, fazendo com que a distribuição somente seja possível nos Países não excluídos.

9. A Fundação de Software de Livre Distribuição (FSF - Free Software Foundation) pode publicar versões revisadas ou novas versões desta Licença Pública Geral de tempos em tempos. Estas novas versões manterão os mesmos objetivos e o espírito da presente versão, podendo variar em detalhes referentes a novas situações encontradas.

   A cada versão é dada um número distinto. Caso o Programa especifique um número de versão específico desta Licença a qual tenha em seu conteúdo a expressão "ou versão mais atualizada", é possível optar pelas condições daquela versão ou de qualquer versão mais atualizada publicada pela FSF.

10. Caso se deseje incorporar parte do Programa em outros programas de livre distribuição de softwares é necessária autorização formal do autor. Para softwares que a FSF detenha os direitos autorais, podem ser abertas exceções desde que mantido o espírito e objetivos originais desta Licença.

AUSÊNCIA DE GARANTIAS

11. UMA VEZ QUE O PROGRAMA É LICENCIADO SEM ÔNUS, NÃO HÁ QUALQUER GARANTIA PARA O PROGRAMA. EXCETO QUANDO TERCEIROS EXPRESSEM-SE FORMALMENTE O PROGRAMA É DISPONIBILIZADO EM SEU FORMATO ORIGINAL, SEM GARANTIAS DE QUALQUER NATUREZA, EXPRESSAS OU IMPLÍCITAS, INCLUINDO, MAS NÃO LIMITADAS, AS GARANTIAS COMERCIAIS E DO ATENDIMENTO DE DETERMINADO FIM. A QUALIDADE E A PERFORMANCE SÃO DE RISCO EXCLUSIVO DOS USUÁRIOS, CORRENDO POR

SUAS CONTA OS CUSTOS NECESSÁRIOS A EVENTUAIS ALTERAÇÕES, CORREÇÕES E REPAROS JULGADOS NECESSÁRIOS.

12. EM NENHUMA OCASIÃO, A MENOS QUE REQUERIDO POR DECISÃO JUDICIAL OU POR LIVRE VONTADE, O AUTOR OU TERCEIROS QUE TENHAM MODIFICADO O PROGRAMA, SERÃO RESPONSÁVEIS POR DANOS OU PREJUÍZOS PROVENIENTES DO USO OU DA FALTA DE HABILIDADE NA SUA UTILIZAÇÃO (INCLUINDO, MAS NÃO LIMITADA, A PERDA DE DADOS OU DADOS ERRÔNEOS), MESMO QUE TENHA SIDO EMITIDO AVISO DE POSSÍVEIS ERROS OU DANOS.

FIM DA LICENÇA

E.2 Apêndice: Como Aplicar Estes Termos a Novos Programas?

Caso se tenha desenvolvido um novo programa e se deseje a sua ampla distribuição para o público, a melhor forma de consegui-lo é torná-lo um software de livre distribuição, o qual qualquer um possa distribuí-lo nas condições desta Licença.

Para tanto basta anexar este aviso ao programa. É aconselhável indicar ainda no início de cada arquivo fonte a ausência de garantias e um apontamento para um arquivo contendo o texto geral desta Licença, como por exemplo:

> ⟨*uma linha para dar o nome do programa e uma breve idéia do que ele faz.*⟩ Copyright ©19yy ⟨*nome do autor*⟩
>
> Este programa é um software de livre distribuição, que pode ser copiado e distribuído sob os termos da Licença Pública Geral GNU, conforme publicada pela Free Software Foundation, versão 2 da licença ou (a critério do autor) qualquer versão posterior.
>
> Este programa é distribuído na expectativa de ser útil aos seus usuários, porém NÃO TEM NENHUMA GARANTIA, EXPLÍCITAS OU IMPLÍCITAS, COMERCIAIS OU DE ATENDIMENTO A UMA DETERMINADA FINALIDADE. Consulte a Licença Pública Geral GNU para maiores detalhes.
>
> Deve haver uma cópia da Licença Pública Geral GNU junto com este

software em inglês ou português. Caso não haja escreva para Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Inclua também informações de como contatar você através de correio eletrônico ou endereço comercial/residencial.

Caso o programa seja interativo, apresente na sua saída um breve aviso quando de seu início como por exemplo:

```
Gnomovision versão 69, Copyright © nome do autor
Gnomovision NÃO POSSUI NENHUMA GARANTIA;
para detalhes digite ‘mostre garantia’.  Este é um software
de livre distribuição e você está autorizado
a distribui-lo dentro de certas condições.  Digite ‘mostre
condição’ para maiores detalhes.
```

Os comandos hipotéticos ‘mostre garantia’ e ‘mostre condição’ apresentarão as partes apropriadas da Licença Pública Geral GNU. Evidentemente os comandos podem variar ou serem acionados por outras interfaces como clique de mouse, etc..

Esta Licença Pública Geral não permite a incorporação de seu programa em programas proprietários. Se o seu programa é uma sub-rotina de biblioteca, você pode achar mais interessante permitir a “ligação” de aplicações proprietárias com sua biblioteca. Se é isso que você deseja fazer, use a Licença Pública Geral Gnu para Bibliotecas no lugar desta Licença.

E.3 BSD Copyright

Copyright ©1991, 1992, 1993, 1994 The Regents of the University of California. Todos os direitos reservados.

Redistribuição e uso nas formas de código fonte ou binários, com ou sem modificação são permitidas dentro das seguintes condições:

1. A redistribuição do software deve conter todas as informações sobre direitos autorais, esta lista de condições e o aviso abaixo;

2. A redistribuição de binários ou executáveis deve conter todas as informações sobre direitos autorais, listas de condições e o aviso abaixo anúncio na documentação e/ou em outros materiais constantes da distribuição;
3. Todos os comerciais e anúncios mencionando funcionalidades deste software devem apresentar o seguinte texto: Este produto inclui software desenvolvido pela Universidade da Califórnia, Berkeley e seus contribuintes;
4. O nome da Universidade ou de seus contribuintes não pode ser utilizado para endossar ou promover produtos derivados deste software sem expressa autorização por escrito.

ESTE SOFTWARE É DISTRIBUÍDO POR SEUS MONITORES E CONTRIBUINTES NA FORMA EM QUE SE ENCONTRA, E QUALQUER GARANTIA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITADAS AS GARANTIAS COMERCIAIS E ATENDIMENTO DE DETERMINADOS PROPÓSITOS QUE NÃO SÃO RECONHECIDAS. EM NENHUMA HIPÓTESE OS MONITORES OU SEUS CONTRIBUINTES SERÃO RESPONSÁVEIS POR QUALQUER DANO DIRETO, INDIRETO, ACIDENTAL, ESPECIAL, INCLUINDO, MAS NÃO LIMITADO À SUSBTITUIÇÃO DE MERCADORIAS OU SERVIÇOS, IMPOSSIBILIDADE DE USO, PERDA DE DADOS, LUCROS CESSANTES OU INTERRUPÇÃO DE ATIVIDADES COMERCIAIS, CAUSADOS EM QUALQUER BASE PELO USO DESTE SOFTWARE.

E.3.1 X Copyright

Copyright ©1987 X Consortium

É concedida e garantida a qualquer pessoa, livre de custos, a obtenção de cópia deste software e dos arquivos de documentação associados (o Software), podendo lidar com o Software sem restrições, incluindo os direitos de uso, cópia, modificação, unificação, publicação, distribuição, sublicenciamento e/ou venda de cópias do Software, e a permissão para as pessoas às quais o Software for fornecido, dentro das seguintes condições:

As informações de copyright a seguir devem estar presentes em todas as cópias ou partes substanciais do Software:

O SOFTWARE SERÁ DISPONIBILIZADO NA FORMA EM QUE SE ENCONTRE, SEM GARANTIAS DE QUALQUER ESPÉCIE, EXPRESSAS OU ÍMPLICITAS, INCLUÍDAS, MAS NÃO LIMITADAS, AS GARANTIAS COMERCIAIS,

Glossário

Uma dificuldade no tema de redes de computadores é relembrar o significado de todos os termos e abreviações. A seguir apresentamos uma lista das mais freqüentes, junto com uma breve explicação.

ACU Unidade de discagem automática. Um modem.[1]

ARP Protocolo de Resolução de Endereços. Usado para mapear endereços IP para endereços Ethernet.

ARPA Agência de Projetos de Pesquisa Avançada, posteriormente denominada DARPA. Criadora da Internet.

ARPANET O antecessor da Internet atual, uma rede experimental fundada pela Agência de Projetos de Pesquisa Avançada do Departamento de Defesa dos Estados Unidos (DARPA).

BBS Sistema de Divulgação de Mensagens. Um sistema de mensagens via linha discada.

BGP Protocolo de Caminhos Padrões Limites. Um protocolo para troca de informações de roteamento entre sistemas autônomos.

BIND O servidor de Nomes de Domínios Internet de Berkeley. Uma implementação do servidor DNS.

BNU Utilidades Básicas de Redes. É a variedade mais comum de UUCP no momento. Também conhecida como UUCP HoneyDanBer, nome derivado dos nomes dos autores: P. Honeyman, D.A. Novitz e B.E. Redman.

BSD Distribuição do Software de Berkeley. Um sabor de `Unix`.

[1] Alternativamente pode ser uma adolescente com um telefone.

caminho Normalmente usado em redes UUCP como um sinônimo para *rotas*. Veja ainda *caminho bang*.

caminho bang Em redes UUCP, uma notação especial para o caminho de um site UUCP para outro. O nome deriva do uso de pontos de exclamação ('bangs' em inglês) na separação dos nomes das máquinas. Exemplo: emailsapucaia!sleopoldo!canoas!gustavo indica uma mensagem para o usuário `gustavo`, passando (nesta ordem) pelos sites `sapucaia`, `sleopoldo` e `canoas`.

cerveja virtual A bebida favorita de todo usuário Linux. A primeira menção à cerveja virtual que eu lembre, foi feita na nota de liberação do kernel `Linux` 0.98.X, onde Linus listou a "Oxford Beer Trolls" na seção de créditos pelo envio de algumas cervejas virtuais.

CCITT Comitê Consultivo Internacional de Telefonia e Telegrafia. Uma organização internacional de serviços telefônicos, etc.

CSLIP IP Comprimido em Linha Serial. Um protocolo para troca de pacotes IP sobre uma linha serial, usando-se compressão de cabeçalho em datagramas TCP/IP.

Definição de Números O nome de uma *RFC* publicada regularmente, que lista os números alocados publicamente em diversas atividades de uma rede TCP/IP. Por exemplo, ela contém a lista de todos os números de portas de serviços bem conhecidos como `rlogin`, `telnet`, etc. A versão mais recente é a RFC 1340.

DNS Sistema de Nome de Domínios. Base de dados distribuída usada na Internet para mapear nomes de máquinas para endereços IP.

EGP Protocolo de Caminho Padrão Externo. Um protocolo para troca de informações de roteamento entre sistemas autônomos.

Ethernet Em termos coloquiais, o nome de um tipo de equipamento de rede. Tecnicamente, Ethernet é parte de um conjunto de especificações definidas pelo IEEE. O hardware Ethernet usa um simples cabo para conectar diversas máquinas, permitindo a transferência a taxas de até 100 Mbps. O protocolo Ethernet define a maneira pela qual as máquinas podem se comunicar sobre um cabo.[2]

[2] O *protocolo* Ethernet comumente usado pelo TCP/IP *não* é exatamente o mesmo que o definido pelo IEEE 802.3. Datagramas Ethernet têm um campo de tipo de adicional de tamanho.

FQDN Nome de Domínio Totalmente Qualificado. Um nome de máquina com um nome de domínio anexo, consistindo em uma entrada válida em uma base de dados de nome de domínios.

FTP Protocolo de Transferência de Arquivos. Um dos mais conhecidos serviços de transferência de arquivos.

FYI "Para Sua Informação". Série de documentos com informações informais sobre tópicos Internet.

GNU Gnu não é Unix – este acrônimo recursivo é o nome de projeto da Associação de Software de Livre Distribuição que busca prover um conjunto de ferramentas `Unix` que podem ser utilizadas e copiadas livremente. Todos os softwares GNU são cobertos por uma licença especial de direitos autorais, também chamada de Licença Pública Geral GNU (GPL). A GPL está reproduzida na seção E.

HoneyDanBer O nome de uma variedade do UUCP. Veja também BNU.

ICMP Protocolo de Mensagens de Controle Internet. Um protocolo de rede usado pelo IP para retornar informações de erro para a máquina de origem, etc.

IEEE Instituto de Engenheiros Elétricos e Eletrônicos. Uma organização destinada a definir padrões. Do ponto de vista do Unix, sua contribuição mais importante são os padrões POSIX que definem os diversos aspectos de um sistema UNIX, desde as interfaces de chamadas ao sistema e sua semântica até ferramentas de administração.

Além disso, o IEEE desenvolveu as especificações para redes Ethernet, Token Ring e Token Bus networks. Uma representação largamente utilizada de formato binário para números reais é também de autoria do IEEE.

IETF Força Tarefa de Engenharia da Internet.

internet Uma rede de computadores formada por uma série de redes individuais menores.

Internet Um rede em especial de abrangência mundial.

IP Protocolo Internet. Um protocolo de rede.

ISO Organização Internacional de Padrões.

ISDN Rede de Serviços Digitais Integrados (RSDI). Tecnologia de telecomunicações que utiliza um meio digital de envio de dados.

LAN Rede Local de Computadores. Normalmente uma pequena rede de microcomputadores.

máquina Geralmente um nó de rede: algo que é capaz de receber e transmitir mensagens em rede. Normalmente é um computador, mas podem ser ainda uma impressora inteligente ou um Terminal X.

MX Servidor de Correio. Um recurso DNS referente ao tipo de registro usado para definir uma máquina como um caminho para as mensagens de um domínio.

NFS Sistema de Arquivos em Rede. Um protocolo padrão de rede e conjunto de softwares para acesso transparente a dados localizados em discos remotos.

NIS Sistemas de Informações em Rede. Uma aplicação baseada em RPC que permite o compartilhamento de arquivos de configuração, como arquivos de senhas, entre diversas máquinas. Veja também as informações sobre YP.

NNTP Protocolo de Transferência de Notícias em Rede. Usado na transferência de notícias sobre conexões em redes TCP.

nome canônico de máquina O nome primário de uma máquina em um sistema de nomes de domínio. Este é o único nome que está associado ao registro tipo A associado à máquina, o qual é retornado quando se executa uma pesquisa reversa.

octeto Na Internet, este termo técnico se refere a quantidade de oito bits. É usado no lugar da palavra *byte*, porque há máquinas na Internet que têm bytes de outros tamanhos.

OSI Interconexão de Sistemas Abertos. Um padrão ISO para programas de rede.

PLIP IP sobre Linhas Paralelas. Um protocolo de troca de pacotes IP sobre uma linha paralela como uma porta de impressora.

porta, TCP ou UDP Portas são abstrações TCP e UDP que representam uma ponta de um serviço. Antes que um processo possa prover ou acessar algum serviço de rede, ele deve solicitar ("bind") uma porta. Junto com os endereços IP de uma máquina, portas individualizam uma das partes de uma conexão TCP.

portmapper O programa portmapper é o mediador entre os números do programas utilizados pelo RPC para identificar os servidores individualmente e os números de portas TCP e UDP aonde os serviços estão sendo disponibilizados.

PPP O protocolo ponto a ponto. PPP é um protocolo flexível e rápido a nível de conexão. Capaz de transportar diversos protocolos de rede como IP ou IPX através de uma conexão ponto a ponto. Além de ser usado em conexões seriais (modem), pode ser utilizado ainda sobre protocolos de conexão como o ISDN.

pesquisa reversa O ato de procurar o nome de uma máquina baseado no endereço IP. No DNS é feito através da pesquisa do endereço IP no domínio `in-addr.arpa`.

propagação O envio de um datagrama de uma estação de uma rede para todas as demais estações simultaneamente.

RARP Protocolo de Resolução de Endereço Reverso. Permite às máquinas encontrarem seu endereço IP durante a inicialização do sistema.

rede, orientada a pacotes Um tipo de rede que provê reenvio instantâneo de dados através da divisão dos dados em pequenos pacotes, os quais são transportados individualmente. Estas redes baseiam-se em conexões permanentes ou semi-permanentes.

rede, armazena e reenvia Oposto da rede orientada a pacotes. Transferem dados como arquivos inteiros e não utiliza conexões permanentes. As máquinas conectam-se umas às outras em certos intervalos de tempo e transferem todos os dados de uma única vez. Requerem o armazenamento de dados localmente até que a conexão seja estabelecida.

resolvedor Biblioteca responsável pelo mapeamento de nomes de máquinas em endereços IP e vice e versa.

registro de recurso Unidade básica de informação de uma base de dados DNS, comumente abreviada como RR. Cada registro tem um certo tipo e uma classe associada a ele, por exemplo um registro de mapeamento de um nome de máquina para um endereço IP tem um tipo A e uma classe IN (para protocolo Internet).

RFC Requisição Para Comentários. Série de documentos descrevendo os padrões Internet.

RIP Protocolo de Informações de Roteamento. Este é o protocolo de roteamento que dinamicamente ajusta as rotas dentro de uma (pequena) rede.

rota A seqüência de máquinas na forma de um caminho a ser seguido entre a origem e o destino. A busca de uma rota adequada é também chamada de *roteamento*.

RPC Chamada de Procedimento Remoto. Protocolo para execução de procedimentos dentro de um processo em uma máquina remota.

RR Forma abreviada de *registro de recurso* usado no DNS.

RS-232 Padrão comum utilizado nas interfaces seriais.

RTS/CTS Um nome coloquial para a negociação de hardware executada por dois dispositivos que se comunicam através de uma RS-232. Os nomes derivam dos dois circuitos envolvidos, RTS ("Pronto Para Enviar") e CTS ("Apto a Enviar").

serviços bem conhecidos Este termo é freqüentemente usado para referenciar-se a tarefas comuns de serviços de rede como `telnet` e `rlogin`. Em um senso mais técnico, são os serviços que têm números de portas definidos na RFC "Definindo Números".

servidor de roteamento Em redes maiores, a topologia da rede muda de uma forma muito rápida e difícil de ser adaptada manualmente em todas as demais estações da rede. Para facilitar a atualização das máquinas são distribuídas informações de roteamento atualizadas. Isso é chamado de roteamento dinâmico, e as informações sobre as rotas são trocadas por *servidores de roteamento* que são executados em máquinas centrais da rede. Os protocolos empregados são chamados *protocolos de roteamento*.

site Um aglomerado de máquinas que, vistas do exterior, comportam-se como um único nó de rede. Por exemplo, de um ponto de vista da Internet, pode-se denominar a Universidade do Pantanal como um site, independentemente da complexidade da rede interna.

SLIP IP sobre Linha Serial. Este é um protocolo para troca de pacotes IP sobre uma linha serial. Veja também CSLIP.

SMTP Protocolo Simples de Transporte de Mensagens. Usado para transporte de correio eletrônico sobre conexões TCP e também para transporte de lotes de mensagens sobre conexões UUCP (SMTP em lotes).

SOA Início de Autoridade. Um tipo de registro de recurso do DNS.

System V Um sabor de `Unix`.

TCP Protocolo de Controle de Transmissão. Um protocolo de rede.

TCP/IP Nome dado a um conjunto de produtos de protocolos internet.

UDP Protocolo de Datagramas de Usuário. Um protocolo de rede.

UUCP Cópia de Unix para Unix. Um conjunto de comandos de transporte de rede para redes via linha discada.

UUCP Versão 2 Uma antiga variedade do UUCP.

YP Páginas Amarelas. Um antigo nome para o NIS o qual não é mais usado, já que este nome foi registrado pela British Telecom. De qualquer forma, muitos utilitários NIS mantiveram o prefixo `yp`.

Bibliografia

Livros

A seguir apresentamos uma lista de livros que podem ser consultados caso deseje-se aprofundar nos temas cobertos por este Guia. Não pretende ser uma lista completa ou sistemática, eu simplesmente os achei muito úteis. Quaisquer sugestões de novos títulos serão bem-vindas.

Livros sobre a Internet

[Kehoe92] Brendan P. Kehoe: *Zen and the Art of the Internet.* .

"Zen" foi um dos primeiros, se não *o* primeiro guia sobre a Internet, apresentado aos usuários iniciantes os diversos aspectos, serviços, folclores da Net. Com aproximadamente 100 páginas, cobre tópicos de email até notícias Usenet e "Vermes". Está disponível para FTP anônimo a partir de diversos sites. Uma cópia impressa está também disponível na Prentice-Hall.

Administração

[Hunt92] Craig Hunt: *TCP/IP Network Administration.* O'Reilly and Associates, 1992. ISBN 0-937175-82-X.

Caso o Guia de Administrador de Redes não seja suficiente para as suas necessidades, consulte este livro. Ele aborda todos os aspectos, desde a obtenção de um endereço IP até um guia de resolução de problemas de rede, incluindo temas de segurança.

Ele focaliza a configuração TCP/IP, ou seja, a configuração da interface, configuração do roteamento e resolução de nomes. Inclui ainda uma descrição detalhada das facilidades oferecidas pelos servidores de roteamento `routed` e `gated`, os quais disponibilizam rotinas dinâmicas de roteamento.

escreve ainda a configuração de programas de aplicação e servidores de rede, como o `inetd`, comandos `r`, NIS e NFS.

O apêndice traz referências detalhadas sobre `gated`, `named` e uma descrição da configuração do programa `sendmail` de Berkeley.

[Stern92] Hal Stern: *Managing NIS and NFS*. O'Reilly and Associates, 1992. ISBN 0-937175-75-7.

Este é um complemento do livro de Craig Hunt denominado "TCP/IP Network Administration". Ele cobre o uso do NIS, o Sistema de Informações em Rede, e NFS, o Sistema de Arquivos em Rede, extensivamente, incluindo configuração e montagem automática, assim como o PC/NFS.

[OReilly89] Tim O'Reilly and Grace Todino: *Managing UUCP and Usenet, 10th ed.* O'Reilly and Associates, 1992. ISBN 0-93717593-5.

Este é o livro padrão de redes baseadas em UUCP. Cobre a versão 2 do UUCP, assim como a BNU. Auxilia a configurar um nó UUCP a partir de seu início, fornecendo dicas práticas e soluções para diversos problemas, como teste de conexões, ou programas de conversação. Lida ainda com tópicos mais complexos como configuração de nó UUCP, ou as sutilezas dos diferentes sabores do UUCP.

A segunda parte do livro lida com a Usenet e softwares de notícias. Explica a configuração do Bnews (versão 2.11) e do

C news, apresentando as tarefas de manutenção de um servidor de notícias.

[Spaf93] Gene Spafford and Simson Garfinkel: *Practical UNIX Security*. O'Reilly and Associates, 1992. ISBN 0-937175-72-2.

Um dos livros indispensáveis para todo aquele que administra um sistema de rede, assim como para outros tipos de usuários. O livro discute todos os aspectos relevantes da segurança na computação, indo de aspectos básicos de segurança sob `Unix` até aspectos de segurança física. De qualquer forma, a parte referente a redes é a mais interessante no nosso contexto. Além das políticas básicas que se relacionam aos serviços Berkeley tais como (`telnet`, `rlogin`, etc), NFS e NIS, ainda aborda aspectos avançados de segurança como o Kerberos do MIT, RPC de segurança da Sun e o uso de firewalls.

[AlbitzLiu92] Paul Albitz and Cricket Liu: *DNS and BIND*. O'Reilly and Associates, 1992. ISBN 1-56592-010-4.

Este livro é útil para aqueles que necessitam administrar servidores de nome DNS. Explica todas as facilidades do DNS em detalhes, fornecendo exemplos que tornam todas aquelas opções do BIND mais inteligíveis do que parecem à primeira vista. Eu achei muito divertido lê-lo e realmente aprendi muito com ele.

[NISPlus] Rick Ramsey: *All about Admnistering NIS+*. Prentice-Hall, 1993. ISBN 0-13-068800-2.

Suporte

A seguinte lista de livros pode ser interessante para aqueles que desejam conhecer mais sobre como o TCP/IP e suas aplicações funcionam, mas não querem ler as RFCs[3].

[Stevens90] Richard W. Stevens: *UNIX Network Programming.* Prentice-Hall International, 1990. ISBN 0-13-949876-X.

Este é provavelmente *o* mais usado livro sobre programação em redes TCP/IP, o qual detalha os aspectos referentes à programação e o protocolo IP.[4]

[Tanen89] Andrew S. Tanenbaum: *Computer Networks.* Prentice-Hall International, 1989. ISBN 0-13-166836-6.

Este livro fornece uma visão interna sobre o tema de redes de computadores. Usando o modelo de referência OSI, explica como é o desenho de cada camada e os algoritmos usados para obtê-las. Em cada camada, as implementações de cada tipo de rede, desde a ARPAnet, são comparadas com as demais.

Um ponto um pouco desagradável neste livro é a abundância de abreviações, o que algumas vezes dificulta o entendimento da mensagem do autor. Isso provavelmente seja uma herança da rede.

[Comer88] Douglas R. Comer: *Internetworking with TCP/IP: Principles, Protocols, and Architecture.* Prentice-Hall International, 1988.

[3] documentos que definem padrões de diversos aspectos técnicos

[4] Note que Stevens acabou de escrever um novo livro chamado *TCP/IP Illustrated, Volume 1, The Protocols*, publicado por Addison Wesley. Infelizmente ainda não pude lê-lo.

Como Fazer

A lista a seguir foi retirada do site do Projeto de Documentação Linux, versão 2.10.96 de 6 de dezembro de 1998, mantida por Tim Bynum.

O que são os Como Fazer `Linux`?

Como Fazer são pequenos documentos que descrevem detalhadamente um certo aspecto de configuração do uso do sistema `Linux`. Há por exemplo, o Como fazer de Instalação, o Como Fazer Mail, o qual descreve como configurar e instalar o correio eletrônico sob `Linux`, etc. Outros exemplos incluem o Como Fazer NET-2 (conhecido anteriormente como FAQ NET-2) e o Como Fazer sobre Impressão.

Informações nos Como Fazer geralmente incluem informações mais detalhadas que nos FAQs `Linux`. Por esta razão o `Linux` FAQ está sendo reescrito. Uma grande quantidade de informações ainda estará presente somente nos Como Fazer. O FAQ será uma pequena lista das perguntas mais freqüentes sobre `Linux`, cobrindo pequenos tópicos.

Como Fazer são documentos simples, similares a FAQs, mas geralmente em um formato distinto daquele de perguntas e respostas. De qualquer forma, muitos Como Fazer contêm uma seção FAQ ao seu final, como por exemplo o Como Fazer NET-2.

Onde Obter os Como Fazer do `Linux`

Os Como Fazer podem ser obtidos via FTP anônimo de um dos seguintes sites:

- `metalab.unc.edu:/pub/Linux/docs/HOWTO`
- `tsx-11.mit.edu:/pub/linux/docs/HOWTO`

assim como em um dos muitos espelhos que estão listados no META-FAQ `Linux` (veja abaixo).

O índice apresentado abaixo lista todos os Como Fazer disponíveis em Inglês. No Brasil existe um esforço de tradução também neste sentido, sendo que já encontram-se traduzidos os principais Como Fazer. Eles podem ser encontrados em `http://ldp-br.conectiva.com.br/documentos/comofazer` ou em sua edição impressa no Guia do Servidor Linux impresso pela Conectiva Informática Ltda.

Os Como Fazer são postados regularmente no grupo de notícias `comp.os.linux` e `comp.os.linux.announce`. Adicionalmente, alguns Como Fazer são enviados também para `news.answers`, podendo ser encontrados nos arquivos `news.answers` no site `rtfm.mit.edu`.

Índice dos Como Fazer Disponíveis

Os Como Fazer já traduzidos ou em tradução pela Conectiva Informática possuem um sinal com (*) ao lado do nome.

- 3Dfx, COMO FAZER, por Bernd Kreimeier `bk@gamers.org`. Como utilizar o suporte a chips de aceleração gráfica. Atualizado em 06.02.98.
- AX25, COMO FAZER(*), por Terry Dawson `terry@perf.no.itg.telecom.com.au`. Como configurar uma rede AX25 para Linux. Atualizado em 17.10.97.
- Access, COMO FAZER, por Michael De La Ru `access-howto@ed.ac.uk`. Como usar tecnologia adaptativa com Linux. Atualizado em 28.03.97.
- Alpha, COMO FAZER(*), por David Mosberge `davidm@azstarnet.com`. Visão geral de sistemas Alpha Digital e processadores. Última atualização em 06.06.97.
- Assembler, COMO FAZER, por François-René Ridea `rideau@ens.fr`. Informações sobre a programação Assembler em procesadores x86. Atualizado em 16.11.97.
- Bash, linha de comando, COMO FAZER(*), por Giles Or `giles@interlog.com`. Como criar e controlar linhas de comando em terminais em modo texto ou gráficos. Atualizado em 01.12.98.
- Benchmarking, COMO FAZER(*), por André D. Bals `andrewbalsa@usa.net`. Como executar medidas básicas de performance. Atualizado em 15.08.97.
- Beowulf, COMO FAZER(*), por Jacek Radajewsk `jacek@usq.edu.au` e Douglas Eadline. Apresenta a arquitetura do Supercomputador Beowulf e provê informações sobre programação paralela. Atualizado em 22.11.98.

- Linha de Comando de Inicialização, COMO FAZER(*), por Paul Gortmake `gpg109@rsphy1.anu.edu.au`. Lista dos argumentos possíveis em tempo de inicialização do sistema e visão geral do software de inicialização. Atualizado em 01.02.98.

- Disco de Inicialização, COMO FAZER(*), por Tom Fawcet `fawcett@croftj.net`. Como criar e manter um disco de inicialização para Linux. Atualizado em 01.02.98.

- Busmouse, COMO FAZER, por Chris Bagwel `cbagwell@sprynet.com`. Informações sobre a compatibilidade de bus mouse com Linux. Atualizado em 15.06.98.

- Gravação de CDs, COMO FAZER, por Winfried Trümpe `winni@xpilot.org`. Como gravar CDs no Linux. Atualizado em 16.12.97.

- CDROM, COMO FAZER, por Jeff Trante `jeff_tranter@pobox.com`. Informações sobre a compatibilidade de dispositivos de CDROM para Linux. Atualizado em 23.02.98.

- Chinês, COMO FAZER, por Chih-Wei Huan `cwhuang@phys.ntu.edu.tw`. Como configurar o Linux para utilizar o conjunto de caracteres chineses. Atualizado em 02.06.98.

- Comercial, COMO FAZER(*), por Martin Michlmay `tbm@cyrius.com`. Lista de softwares comerciais para Linux. Atualizado em 21.09.98.

- Configuração, COMO FAZER(*), por Guido Gonzat `guido@ibogfs.cineca.it`. Como customizar e fazer ajustes finos no Linux. Atualizado 10.04.98.

- Consultores, COMO FAZER(*), por Martin Michlmay `tbm@cyrius.com`. Lista de consultores Linux em todo o mundo. Atualizado em 08.11.98.

- Cirílico, COMO FAZER, por Alexander L. Belikof `abel@bfr.co.il`. Como configurar o Linux para o uso com caracteres cirílicos. Atualizado em 23.01.98.

- DNS, COMO FAZER(*), por Nicolai Langfeld `janl@math.uio.no`. Como configurar o DNS. Atualizado em 12.11.98.

- DOS/Windows para Linux, COMO FAZER(*), por Guido Gonzat `guido@ibogfs.cineca.it`. Como migrar de DOS/Windows para Linux. Atualizado em 15.04.98.

- DOSEMU, COMO FAZER(*), por Uwe Bonne bon@elektron.ikp.physik.th-darmstadt.de. COMO FAZER sobre o emulador Linux do MS-DOS, DOSEMU. Atualizado em 15.03.97 para o dosemu-0.64.4 (em progresso uma atualização).

- Dinamarquês, COMO FAZER, por Niels Kristian Bech Jense nkbj@image.dk. Como configurar o Linux para uso com o conjunto de caracteres dinamarqueses. Atualizado em 01.12.98.

- Distribuição, COMO FAZER, por Eric S. Raymond esr@snark.thyrsus.com. Uma lista das distribuições Linux. Atualizado em 10.09.98.

- ELF, COMO FAZER(*), por Daniel Barlo daniel.barlow@linux.org. Como instalar e migrar para o formato binário ELF. Atualizado em 14.07.96.

- Emacspeak, COMO FAZER, por Jim Van Zand jrv@vanzandt.mv.com. Como usar 'emacspeak' com Linux. Atualizado em 21.12.97.

- Esperanto, COMO FAZER, por Wolfram Dieste diestel@rzaix340.rz.uni-leipzig.de. Como usar Esperanto em formato geral e ISO-8859-3 com Linux. Atualizado em 06.98.

- Ethernet, COMO FAZER(*), por Paul Gortmake gpg109@rsphy1.anu.edu.au. Informações sobre compatibilidade de hardwares Ethernet para Linux. Atualizado em 06.07.98.

- Finlandês, COMO FAZER, por Pekka Taipal pjt@iki.fi. Como configurar o Linux para uso com o conjunto de caracteres do idioma finlandês. Atualizado em 14.02.96.

- Firewall, COMO FAZER(*), por Mark Grenna markg@netplus.net. Como configurar um firewall usando Linux. Atualizado em 08.11.96.

- Francês, COMO FAZER, por Guylhem Azna guylhem@danmark.linux.eu.org. Como configurar o Linux para uso com o conjunto de caracteres do idioma francês.

- Ftape, COMO FAZER(*), por Kevin Johnso kjj@pobox.com. Informações sobre a compatibilidade de dispositivos de fita com o Linux. Atualizado em 08.98.

- GCC, COMO FAZER(*), por Daniel Barlo daniel.barlow@linux.org. Como configurar o compilador GNU C e as bibliotecas de desenvolvimento. Atualizado em 28.02.96.

- Alemão, COMO FAZER, por Winfried Trümpe `winni@xpilot.org`. Informações sobre o uso do Linux com funcionalidades específicas para o idioma alemão. Atualizado em 19.03.97.

- Glibc2, COMO FAZER(*), por Eric Gree `ejg3@cornell.edu`. Como instalar e migrar para a biblioteca glibc2. Atualizado em 08.02.98.

- HAM, COMO FAZER(*), por Terry Dawso `terry@perf.no.itg.telecom.com.au`. COMO FAZER de configuração de software para rádio amador no Linux. Atualizado em 01.04.97.

- COMO FAZER Índice(*), por Tim Pornu `linux-howto@sunsite.unc.edu`. Índice dos documentos COMO FAZER para Linux. Atualizado em 06.12.98.

- Compatibilidade de Hardware, COMO FAZER(*), por Patrick Reijne `antispam.patrickr@antispam.bart.nl`. Uma lista dos hardwares conhecidos e que funcionam com Linux. Atualizado em 30.07.98.

- Hebreu, COMO FAZER, por Yair G. Rajwa `yair@hobbes.jct.ac.il`. Como configurar o Linux para uso com o conjunto de caracteres do idioma hebraico. Atualizado e 12.09.95.

- Planilha de Informações (INFO-SHEET) (*), por Michael K. Johnso `johnsonm@redhat.com`. Informações genéricas de introdução ao uso do sistema Linux. Atualizado em 01.09.98.

- Correntes IP, COMO FAZER(*), por Paul Russel `Paul.Russell@rustcorp.com.au`. Como instalar e configurar o software aprimorado de correntes IP para Firewall. Atualizado em 27.10.98.

- IPX, COMO FAZER(*), por Terry Dawso `terry@perf.no.itg.telecom.com.au`. Como instalar e configurar uma rede IPC. Atualizado em 06.05.98.

- IR, COMO FAZER, por Werner Heuse `r2d2c3po@zedat.fu-berlin.de`. Uma introdução ao software disponibilizado pelo Projeto IR. Atualizado em 27.09.98.

- Provedor de Acesso Internet, COMO FAZER(*), por Egil Kvaleber `egil@kvaleberg.no`. Introdução básica à instalação de um provedor de acesso a Internet. Atualizado em 05.03.98.

- Instalação, COMO FAZER(*), por Eric S. Raymond `esr@snark.thyrsus.com`. Como obter e instalar Linux. Atualizado em 20.11.98.

- Servidor Intranet, COMO FAZER(*), por Pramod Karna `karnad@indiamail.com`. Como configurar um servidor Linux para Intranet. Atualizado em 07.08.97.

- Italiano, COMO FAZER, por Marco "Gaio" Gaiari `gaio@dei.unipd.it`. Como configurar o Linux para uso com o conjunto de caracteres do idioma italiano. Atualizado em 03.11.98.

- Java-CGI, COMO FAZER(*), por David H. Silbe `dhs@orbits.com`. Como configurar programas CGI baseados em java. Atualizado em 01.12.98.

- Kernel, COMO FAZER(*), por Brian War `ward@blah.math.tu-graz.ac.at`. Atualização e compilação do kernel do Linux. Atualizado em 26.05.97.

- Teclado e Console, COMO FAZER(*), por Andries Brouwe `aeb@cwi.nl`. Informações sobre o teclado, console e caracteres não ASCII em Linux. Atualizado em 25.02.98.

- Instalação Rápida, COMO FAZER(*), por Martin Hamilto `martinh@gnu.org`. Descrição rápida de como configurar o sistema de instalação rápida do Red Hat Linux para instalação de grandes quantidades de máquinas idênticas. Atualizado em 28.09.98.

- LinuxDoc+Emacs+Ispell, COMO FAZER(*), por Philippe Marti `feloy@wanadoo.fr`. Auxílio a escritores e tradutores dos COMO FAZER Linux e quaisquer outros documentos do Projeto de Documentação Linux. Atualizado em 27.02.98.

- META-FAQ(*), por Michael K. Johnso `johnsonm@redhat.com`. Uma lista das fontes de informação sobre Linux. Atualizado em 25.10.97.

- MGR, COMO FAZER, por Vincent Broma `broman@nosc.mil`. Informações sobre a interface gráfica MGR para Linux. Atualizado em 30.05.96.

- MILO, COMO FAZER(*), por David A. Ruslin `david.rusling@reo.mts.dec.com`. Como utilizar o carregador Linux para computadores Alpha Digital (MILO). Atualizado em 06.12.96.

- Correio Eletrônico, COMO FAZER(*), por Guylhem Azna `guylhem@danmark.linux.eu.org`. Informações sobre servidores e clientes de correio eletrônico. Atualizado em 01.98.

- Modem, COMO FAZER(*), por David S. Lawye bf347@lafn.org. Auxílio para seleção, conexão, configuração, resolução de problemas e entendimento de modems para PC. Atualizado 12.98.
- Múltiplos Discos, COMO FAZER(*), por Stein Gjoe sgjoen@nyx.net. Como configurar múltiplos discos rígidos no Linux. Atualizado em 03.02.98.
- Multicast, COMO FAZER(*), por Juan-Mariano de Goyenech jmseyas@dit.upm.es. Este COMO FAZER cobre os aspectos relacionados com a propagação de mensagens sobre redes TCP/IP. Atualizado em 20.03.98.
- NET-3, COMO FAZER(*), por Terry Dawso terry@perf.no.itg.telecom.com.au. Como configurar redes TCP/IP no Linux. Atualizado em 08.98.
- NFS, COMO FAZER(*), por Nicolai Langfeld janl@math.uio.no. Como configurar servidores e clientes NFS. Atualizado em 03.11.97.
- NIS, COMO FAZER(*), por Thorsten Kuku kukuk@vt.uni-paderborn.de. Informações sobre o uso de NIS/YP em Linux. Atualizado em 12.06.98.
- Visão Geral de Redes, COMO FAZER(*), por Daniel López Ridruej ridruejo@esi.us.es. O propósito deste documento é fornecer uma visão geral das capacidades de rede do sistema operacional Linux, providenciando indicativos para maiores informações e detalhes de implantação. Atualizado em 10.07.98.
- Disco Ótico, COMO FAZER(*), por Skip Ry Skip_Rye@faneuil.com. Como usar dispositivos de discos óticos com Linux. Atualizado em 01.09.98.
- Oracle, COMO FAZER(*), por Paul Haig paul@nailed.demon.co.uk. Como configurar um servidor de banco de dados Oracle. Atualizado em 04.08.98.
- PCI, COMO FAZER(*), por Michael Wil Michael.Will@student.uni-tuebingen.de. Informações sobre compatibilidade da arquitetura PCI com Linux. Atualizado em 30.03.97.
- PCMCIA, COMO FAZER(*), por Dave Hind dhinds@allegro.stanford.edu. Como instalar e utilizar cartões PCMCIA. Atualizado em 13.08.98.
- PPP, COMO FAZER(*), por Robert Har hartr@interweft.com.au. Informações sobre o uso de redes PPP utilizando Linux. Atualizado em 31.03.97.
- PalmOS, COMO FAZER(*), por David H. Silbe pilot@orbits.com. Como utilizar o Palm OS com Linux. Atualizado em 20.09.98.

- Processamento Paralelo, COMO FAZER(*), por Hank Diet `pplinux@ecn.purdue.edu`. Discussão sobre as abordagens de processamento paralelo em Linux. Atualizado em 05.01.98.
- Plug and Play, COMO FAZER(*), por David Lawye `bf347@lafn.org`. Como obter suporte a Plug-and-Play no sistema Linux. Atualizado em 11.98.
- Polonês, COMO FAZER, por Sergiusz Pawlowic `ser@arch.pwr.wroc.pl`. Informações sobre o uso de Linux utilizando funcionalidades específicas da língua polonesa. Atualizado em 01.06.98.
- Português, COMO FAZER(*), por Carlos Augusto Moreira dos Santo `casantos@cpmet.ufpel.tche.br`. Este documento pretende ser um guia de referência de configuração do Linux e seus programas na língua portuguesa. Atualizado em 28.10.98.
- PostgreSQL, COMO FAZER(*), por Al Dev (Alavoor Vasudevan) `aldev@hotmail.com`. Como configurar um servidor de banco de dados PostgreSQL. Atualizado em 24.11.98.
- Impressão, COMO FAZER(*), por Grant Taylo `gtaylor+pht@picante.com`. Abordagem sobre softwares de impressão para Linux. Atualizado em 06.06.98.
- Impressão, Uso de, COMO FAZER(*), por Mark Komarinsk `markk@auratek.com`. Como utilizar o sistema de impressão para uma grande variedade de tipos de arquivos e opções. Atualizado em 06.02.98.
- Quake, COMO FAZER(*), por Bob Zimbinsk `bobz@mr.net` Thomas Mike Hallock `mikeh@medina.net`. Este documento explica como instalar, executar e principais problemas na execução do Quake, QuakeWorld e Quake II em um sistema Linux para Intel. Atualizado em 30.08.98.
- RPM, COMO FAZER(*), por Donnie Barne `djb@redhat.com`. Como utilizar o sistema de gerenciamento de pacotes da Red Hat (.rpm). Atualizado em 08.04.97.
- Lista de Leituras, COMO FAZER(*), por Eric S. Raymond `esr@snark.thyrsus.com`. Livros interessantes sobre temas relacionados com Linux. Atualizado em 22.11.98.
- RAID do Raiz, COMO FAZER(*), por Michael A. Robinto `michael@bzs.org`. Como criar um sistema de arquivos raiz baseado em RAID. Atualizado em 25.03.98.

- Programação SCSI, COMO FAZER, por Heiko Eissfeld `heiko@colossus.escape.de`. Informações sobre programação em interfaces SCSI genéricas. Atualizado em 07.05.96.

- SMB, COMO FAZER(*), por David Woo `dwood@plugged.net.au`. Como usar o protocolo Blocos de Mensagem de Sessão (SMB) com Linux. Atualizado em 10.08.96.

- SRM COMO FAZER(*), por David Mosberge `davidm@azstarnet.com`. Como inicializar sistemas Linux em equipamentos Alpha Digital através do firmware SRM. Atualizado em 17.08.97.

- Segurança, COMO FAZER(*), por Kevin Fenz `kevin@scrye.com`. Visão geral de aspectos de segurança do sistema. Atualizado em 01.05.98.

- Serial, COMO FAZER(*), por David Lawye `bf347@lafn.org`. Como usar dispositivos seriais (modems, terminais, etc.) com Linux. Atualizado em 07.98.

- Serial, Programação, COMO FAZER, por Peter H. Bauma n `Peter.Baumann@dlr.de`. Como usar portas seriais em programas. Atualizado em 22.01.98.

- Senhas Sombra, COMO FAZER(*), por Michael H. Jackso `mhjack@tscnet.com`. Como obter, instalar e configurar senhas sombra. Atualizado em 03.04.96.

- Esloveno, COMO FAZER, por Primoz Peterli `primoz.peterlin@biofiz.mf.uni-lj.si`. Informações sobre o uso de funcionalidade específicas do idioma esloveno com Linux. Atualizado em 30.10.96.

- Softwares para Linux, Liberação de, COMO FAZER(*), por Eric S. Raymond `esr@snark.thyrsus.com`. Descreve práticas aconselháveis de liberação de softwares para projetos de fontes abertos. Atualizado em 21.11.98.

- Som, COMO FAZER(*), por Jeff Trante `jeff_tranter@pobox.com`. Hardwares e softwares para som no sistema operacional Linux. Atualizado em 23.01.98.

- Som, Reproduzindo, COMO FAZER(*), por Yoo C. Chun `wacko@laplace.snu.ac.kr`. Como reproduzir diversos formatos de som sob Linux. Atualizado em 11.08.98.

- Espanhol, COMO FAZER, por Gonzalo Garcia Agull Gonzalo.Garcia-Agullo@jrc.es. Informações sobre o uso do Linux com funcionalidades específicas do idioma castelhano. Atualizado em 20.08.96.
- Tcl/Tk, COMO FAZER(*), por Luca Rossett lukaros@tin.it. Descreve a abordagem do Linux na linguagem interpretada Tcl.
- teTeX, COMO FAZER(*), por Robert Kieslin kiesling@terracom.net. Como instalar o pacote teTeX (TeX and LaTeX) sob Linux. Atualizado em 09.11.98.
- Terminal em modo texto, COMO FAZER(*), por David S. Lawye bf347@lafn.org. explica o conceito, o funcionamento, a instalação e configuração de terminais baseados no modo texto. Atualizado em 11.98.
- Tailandês, COMO FAZER, por Poonlap Veeratanabut poon-v@fedu.uec.ac.jp. Como configurar o Linux para uso com o conjunto de caracteres do idioma tailandês. Atualizado em 04.08.98.
- Dicas, COMO FAZER(*), por Paul Anderso paul@geeky1.ebtech.net. COMO FAZER sobre diversas dicas e sugestões no uso do Linux. Atualizado em 06.98.
- UMSDOS, COMO FAZER(*), por Jacques Gelina jacques@solucorp.qc.ca. Como instalar e manter o sistema de arquivos UMSDOS. Atualizado em 13.11.95.
- No Breaks, COMO FAZER(*), por Harvey J. Stei abel@netvision.net.il. Informações sobre o uso de no breaks com Linux. Atualizado em 18.11.97.
- UUCP, COMO FAZER(*), por Guylhem Azna. Informações sobre o software UUCP para Linux. Atualizado em 06.02.98.
- Unix e Internet, Fundamentos sobre, COMO FAZER(*), por Eric S. Raymond esr@snark.thyrsus.com. Descreve o funcionamento básico de computadores PC, sistemas operacionais Unix e Internet em uma linguagem não técnica. Atualizado em 03.12.98.
- Grupos de Usuários, COMO FAZER(*), por Kendall Grant Clar kclark@ntlug.org. Dicas sobre a fundação, manutenção e crescimento de um Grupo de Usuários Linux. Atualizado em 24.04.98.
- VARs, COMO FAZER(*), por Martin Michlmay tbm@cyrius.com. Listas de revendas Linux com valor agregado. Atualizado em 25.10.98.

- VME, COMO FAZER(*), por John Huggins and Michael Wyric `vmelinux@va.net`. Como executar o Linux em Pentium com barramento VME e outros barramentos PCI baseados em VME. Atualizado em 30.07.98.

- VMS para Linux, COMO FAZER, por Guido Gonzat `guido@ibogfs.cineca.it`. Como migrar do VMS para Linux. Atualizado em 20.04.98.

- Serviços Virtuais, COMO FAZER(*), por Brian Ackerma `brian@nycrc.net`. Como configurar os serviços de hospedagem de domínios virtuais. Atualizado em 15.08.98.

- WWW, COMO FAZER(*), por Wayne Leiste `n3mtr@qis.net`. Como configurar servidores e clientes WWW. Atualizado em 19.11.97.

- WWW mSQL, COMO FAZER(*), por Oliver Corf `corff@zedat.fu-berlin.de`. Como configurar um banco de dados para um servidor Web com mSQL. Atualizado em 17.09.97.

- XFree86, COMO FAZER(*), por Eric S. Raymond `esr@snark.thyrsus.com`. Como obter, instalar e configurar Xfree86 3.2 (X11R6). Atualizado em 27.10.98.

- XFree86, Configuração de Vídeo, COMO FAZER(*), por Eric S. Raymond `esr@snark.thyrsus.com`. Como compor uma linha de comando de configuração para o XFree86. Atualizado em 20.02.98.

- X Window, Usando o, COMO FAZER(*), por Ray Brigle `ray@croftj.net`. Informações sobre a configuração do ambiente X Window para usuários Linux. Atualizado em 29.11.98.

MINI-COMO FAZER Estes são documentos mais simples, especializados em algum tema, que descrevem inúmeros aspectos do uso do Linux.

- Mouse com 3 Botões, MINI-COMO FAZER, por Geoff Shor `geoff@kipper.york.ac.uk`. Como configurar o mouse para uso ou emulação de 3 botões. Atualizado em 31.05.98.

- Cópias de Segurança, ADSM, MINI-COMO FAZER, por Thomas Koeni `Thomas.Koenig@ciw.uni-karlsruhe.de`. Como instalar e usar o programa de criação de cópias de segurança ADSM. Atualizado em 15.01.97.

- ADSL, MINI-COMO FAZER, por David Fanni dfannin@dnai.com. Endereços para aquisição, instalação e configuração de linhas digitais assimétricas. Atualizado em 07.06.98.

- AI, MINI-COMO FAZER, por John A. Eikenberr jae@ai.uga.edu. Informações sobre softwares Ai-Alife para Linux. Atualizado em 13.01.98.

- Defesa do Linux, MINI-COMO FAZER, por Paul L. Roger Paul.L.Rogers@li.org. Sugestões de como divulgar e disseminar o uso do Linux. Atualizado em 07.05.98.

- Apache SSL PHP/FI, MINI COMO FAZER, por Marcus Faur marcus@faure.de. Construindo um servidor web de múltiplo uso, com segurança e desenvolvimento de programas. Atualizado em 07.98.

- Montagem Automática, por Do don@sabotage.org. Este arquivo descreve o processo de montagem automática denominada autofs, como configurar, evitar alguns problemas. Atualizado em 07.09.98.

- Cópias de Segurança com MSDOS, MINI COMO FAZER, por Christopher Neufel neufeld@physics.utoronto.ca. Como criar cópias de segurança de máquinas Linux com MSDOS. Atualizado em 05.08.97.

- Alimentado por Bateria, MINI COMO FAZER, por Hanno Muelle hanno@lava.de. Como reduzir o consumo de energia em um sistema Linux. Atualizado em 21.12.97.

- Boca, MINI COMO FAZER, por David H Denni david@freelink.net. Como instalar uma placa serial de 16 portas Boca (Boca 2016). Atualizado em 01.08.97.

- BogoMips, MINI COMO FAZER, por Wim C.A. van Dors baron@clifton.hobpor.nl. Informações sobre BogoMips. Atualizado em 13.12.97.

- Bridge, MINI COMO FAZER, por Chris Col cole@lynkmedia.com. Como configurar uma bridge Ethernet. Atualizado em 07.09.98.

- Bridge+Firewall, MINI COMO FAZER, por Peter Breue ptb@it.uc3m.es. Como configurar uma bridge Ethernet e um firewall. Atualizado em 19.12.97.

- Bzip2, MINI COMO FAZER, por David Fette dfetter@best.com. Como usar o novo programa de compressão bzip2. Atualizado em 29.06.98.

- Cable Modem, MINI COMO FAZER, por Vladimir Vuksa `vuksan@veus.hr`. Como usar um cable modem com um provedor de acesso Internet. Atualizado em 06.12.98.

- Cipe+Masquerading, MINI COMO FAZER, por Anthony Ciaraval `acj@home.com`. Como configurar uma rede privativa virtual entre redes locais usando cipe através de máquinas firewall Linux. Atualizado em 28.10.98.

- Relógio, MINI COMO FAZER, por Ron Bea `rbean@execpc.com`. Como configurar e manter o relógio atualizado. Atualizado em 12.98.

- Café, MINI COMO FAZER, por Georgatos Photi `gef@ceid.upatras.gr`. Pensamento sobre como fazer café com Linux (humor). Atualizado em 15.01.98.

- Comando ls em corfes, MINI COMO FAZER, por Thorbjoern Ravn Anderse `ravn@dit.ou.dk`. Como configurar as cores no comando ls. Atualizado em 07.08.97.

- IMAP Cyrus, MINI COMO FAZER, por Kevin Mitchel `kevin@iserv.net`. Como instalar o servidor IMAP Cyrus. Atualizado em 21.01.98.

- DHCP, MINI COMO FAZER, por Vladimir Vuksa `vuksan@veus.hr`. como configurar um servidor e cliente DHCP. Atualizado em 14.11.98.

- RAID com hardware DPR, MINI COMO FAZER, por Ram Samudral `me@ram.org`. Como configurar hardware RAID. Atualizado em 15.12.97.

- Diald, MINI COMO FAZER, por Harish Pilla `h.pillay@ieee.org`. Como usar o diald para conexões discadas com um provedor de acesso. Atualizado em 03.06.96.

- Diskless, MINI COMO FAZER, por Robert Nemki `buci@math.klte.hu`. Como configurar uma máquina Linux sem disco rígido. Atualizado em 12.09.96.

- Ext2fs recuperação de arquivos apagados, MINI COMO FAZER, por Aaron Cran `aaronc@pobox.com`. Como recuperar um arquivo apagado em um sistema de arquivos ext2. Atualizado em 04.08.97.

- Servidor de Fax, MINI COMO FAZER, por Erez Straus `erez@newplaces.com`. Como configurar um servidor de fax. Atualizado em 08.11.97.

- Firewall, Uso de, MINI COMO FAZER, por François-René Ridea rideau@ens.fr. Como usar conexões ppp com telnet transparente sobre um firewall Internet. Atualizado em 27.11.98.

- GIS-GRASS, MINI COMO FAZER, por David A. Hasting dah@ngdc.noaa.gov. Como instalar um Sistema de Informações Geográficas (GIS). Atualizado em 13.11.97.

- GTEK BBS-550, MINI COMO FAZER, por Wajihuddin Ahme wahmed@sdnpk.undp.org. Como configurar uma placa multiserial GTEK BBS-550 com Linux. Atualizado em 20.08.97.

- Atualização de Discos Rígidos, MINI COMO FAZER, por Yves Bellefeuill yan@ottawa.com. Como copiar um sistema Linux de um disco para outro. Atualizado em 31.01.98.

- E/S, Programação de portas de, MINI COMO FAZER, por Riku Saikkone Riku.Saikkonen@hut.fi. Como usar portas de entrada e saída em programa C. Atualizado em 28.12.97.

- Apelidos IP, MINI COMO FAZER, por Harish Pilla h.pillay@ieee.org. Como usar apelidos IP. Atualizado em 13.01.97.

- IP Mascarado, MINI COMO FAZER, por Ambrose A ambrose@writeme.com. Como usar IP mascarado. Atualizado em 10.11.97.

- Sub-redes IP, MINI COMO FAZER, por Robert Har hartr@interweft.com.au. Porque e como usar sub-redes em redes IP. Atualizado em 31.03.97.

- Conectividade com Provedores Internet, MINI COMO FAZER, por Michael Strate mstrates@croftj.net. Como obter mensagens e notícias sobre uma conexão discada. Atualizado em 06.11.97.

- Instalação a partir de um Zip Drive, MINI COMO FAZER, por Kevin Snivel k.snively@seaslug.org. Como instalar o Linux a partir de um Zip Drive em uma porta paralela. Atualizado em 29.04.98.

- Kerneld, MINI COMO FAZER, por Henrik Storne storner@osiris.ping.dk. Como usar o 'kerneld' (carga dinâmica de módulos). Atualizado em 19.07.97.

- LBX, MINI COMO FAZER, por Paul D. Smit psmith@baynetworks.com. Como usar Banda baixa X (LBX). Atualizado em 11.12.97.

- LILO, MINI COMO FAZER, por Alessandro Rubin `rubini@linux.it`. Exemplos de típicas instalações do Linux Loader. Atualizado em 16.08.98.

- Discos Grandes, MINI COMO FAZER, por Andries Brouwe `aeb@cwi.nl`. Como usar discos grandes com mais de 1024 cilindros. Atualizado em 18.05.98.

- Linhas Dedicadas, MINI COMO FAZER, por Rob van der Putte `rob@sput.webster.nl`. Como configurar modems com linhas dedicadas. Atualizado em 07.98.

- Linux+DOS+Win95+OS2, MINI COMO FAZER, por Mike Harla `r3mdh@raex.com`. como usar Linux e DOS e OS/e e Windows 95 juntos. Atualizado em 11.11.97.

- Linux+FreeBSD, MINI COMO FAZER, por Niels Kristian Bech Jense `nkbj@image.dk` Como usar Linux e FreeBSD juntos. Atualizado em 01.12.98.

- Linux+NT-Loader, MINI COMO FAZER, por Bernd Reicher `reichert@dial.eunet.ch`. Como usar Linux e o carregador de sistemas do Windows NT. Atualizado em 02.09.97.

- Linux+Win95, MINI COMO FAZER, por Jonathan Kat `jkatz@in.net`. Como usar Linux e Windows 95 juntos. Atualizado em 26.10.96.

- Loadlin+Win95, MINI COMO FAZER, por Chris Fische `protek@brigadoon.com`. Como usar o Linux e o Windows 95 juntos, utilizando o utilitário loadlin. Atualizado em 09.11.98.

- Terminal Mac, MINI COMO FAZER, por Robert Kieslin `kiesling@terracom.net`. Como usar um Apple Macintosh como um terminal serial. Atualizado em 09.11.97.

- fila de Mensagens de Correio, MINI COMO FAZER, por Leif Erlingsso `leif@lege.com`. Como colocar mensagens remotas em fila e entregá-las localmente. Atualizado em 03.09.97.

- Mail2News, MINI COMO FAZER, por Robert Har `iweft@ipax.com.au`. Como configurar um caminho do correio eletrônico para notícias. Atualizado em 04.11.96.

- Páginas de Manual, MINI COMO FAZER, por Jens Schweikhard `schweikh@noc.dfn.de`. Como escrever páginas de manual. Atualizado em 07.98.

- Módulos, MINI COMO FAZER, por Riley H. William `rhw@bigfoot.com`. Como configurar e instalar módulos do kernel. Atualizado em 14.11.97.

- Múltipla Inicialização usando LILO, MINI COMO FAZER, por Renzo Zanell `rzanelli@southeast.net`. Múltipla inicialização entre Windows 95, Windows NT e Linux. Atualizado em 26.03.98.

- Terminal X NCD, MINI COMO FAZER, por Ian Hodg `ihodge@nortel.ca`. Descreve como conectar um terminal NCD a um servidor Unix. Atualizado em 03.04.98.

- Raiz NFS, MINI COMO FAZER, por Andreas Kostyrk `andreas@ag.or.at`. Como configurar máquinas Linux sem discos rígidos. Atualizado em 08.08.97.

- Clientes com Raiz NFS, MINI COMO FAZER, por Ofer Mao `ofer@hadar.co.il`. Como configurar máquinas Linux sem discos usando NFS. Atualizado em 01.07.97.

- Nó Netrom, MINI COMO FAZER, por Karl Larse `k5di@yahoo.com`. Como configurar o pacote de utilitários ax25 para rádio amadores como Netrom. Atualizado em 19.10.98.

- Netscape+Proxy, MINI COMO FAZER, por Sarma Seetamraj `sarma@usa.net`. Como configurar um servidor proxy para Netscape. Atualizado em 15.08.97.

- Netstation, MINI COMO FAZER, por Kris Buytaer `Kris.Buytaert@advalvas.be`. Como configurar um IBM Netstation em uma rede local utilizando um Linux como servidor. Atualizado em 22.02.98.

- Site de Notícias Leaf, MINI COMO FAZER, por Florian Kuehner `sutok@gmx.de`. Como configurar um site de notícias leaf. Atualizado em 04.01.98.

- Mail Desconectado, MINI COMO FAZER, por Gunther Voe `freaker@tuc.ml.org`. Como configurar endereços de correio eletrônico sem uma conexão dedicada com a Internet. Atualizado em 04.06.98.

- PLIP, MINI COMO FAZER, por Andrea Controzz `controzz@cli.di.unipi.it`. Como configurar o PLIP (Protocolo de Interface de Linha Paralela). Atualizado em 12.03.98.

- Particionamento, MINI COMO FAZER, por Kristian Koehntop `kris@koehntopp.de`. Como escolher as partições de disco. Atualizado em 03.11.97.

- Recuperação de Partições, MINI COMO FAZER, por Rolf Klause `rolfk@romsdal.vgs.no`. Como recuperar partições apagadas do Linux. Atualizado em 22.10.97.

- Variável Path, MINI COMO FAZER, por Esa Turtiaine `etu@dna.fi`. Como usar a variável de ambiente Path. Atualizado em 15.11.97.

- Verificação de Pré-Instalação, MINI COMO FAZER, por S. Parthasarath `algolog@hd1.vsnl.net.in`. Questionário e itens de verificação antes da instalação. Atualizado em 29.08.98.

- Contabilidade de Processos, MINI COMO FAZER, por Albert M.C. Ta `bertie@scn.org`. Como configurar a contabilidade de processos. Atualizado em 08.08.97.

- Sub-rede Proxy ARP, MINI COMO FAZER, por Bob Edward `bob@faceng.anu.edu.au`. Como usar sub-redes com Proxy ARP. Atualizado em 08.97.

- Navegador Público, MINI COMO FAZER, por Donald B. Marti Jr. `dmarti@best.com`. Como configurar uma conta de visitante para usar um navegador Web. Atualizado em 05.01.98.

- Qmail+MH, MINI COMO FAZER, por Christopher Richardso `rdn@tara.n.eunet.de`. Como instalar o qmail e MH. Atualizado em 05.03.98.

- Quota, MINI COMO FAZER, por Albert M.C. Ta `bertie@scn.org`. Como configurar quotas de uso de disco. Atualizado em 08.08.97.

- RCS, MINI COMO FAZER, por Robert Kieslin `kiesling@terracom.net`. Como usar o Sistema de Controle de Revisões - RCS. Atualizado em 14.08.97.

- RPM+Slackware, MINI COMO FAZER, por Dave Whitinge `dave@whitinger.net`. Como instalar o Gerenciador de Pacote Red Hat (RPM) no Slackware. Atualizado em 13.04.98.

- Red Hat, CD MINI COMO FAZER, por Morten Kjeldgaar `mok@imsb.au.dk` Peter von der Ahé `pahe+rhcd@daimi.au.dk`. Como criar CDs com a distribuição Red Hat Linux. Atualizado em 09.09.98.

- Inicialização Remota, MINI COMO FAZER, por Marc Vuilleumier Stückelber `Marc.VuilleumierStuckelberg@cui.unige.ch`. Como configurar um selecionador de inicialização baseada em servidores. Atualizado em 06.98.

- Aplicações Gráficas Remotas, MINI COMO FAZER, por Vincent Zweij `zweije@xs4all.nl`. Como executar aplicações X remotas. Atualizado em 14.07.98.

- Emulador SLIP-PPP, MINI COMO FAZER, por Iris `irish@eskimo.com`. Como usar emuladores SLIP-PPP com Linux. Atualizado em 07.08.97.

- Endereços Sendmail, MINI COMO FAZER, por Thomas Roessle `roessler@guug.de`. Como configurar os arquivos de controle do sendmail para uso doméstico através de uma linha discada. Atualizado em 06.05.98.

- Sendmail+UUCP, MINI COMO FAZER, por Jamal Hadi Sali. Como usar sendmail e UUCP juntos. Atualizado em 08.98.

- Mail seguro via SSH, MINI COMO FAZER, por Manish Sing `yosh@gimp.org`. Como configurar conexões POP seguras usando ssh. Atualizado em 30.09.98.

- Pouca Memória, MINI COMO FAZER, por Todd Burges `tburgess@uoguelph.ca`. Como executar o Linux em um sistema com pouca memória. Atualizado em 29.10.97.

- Construindo Software, MINI COMO FAZER, por Mendel Leo Coope `thegrendel@theriver.com`. Como construir pacotes de softwares. Atualizado em 06.07.98.

- RAID por Software, MINI COMO FAZER, por Linas Vepsta `linas@fc.net`. Como configurar um sistema de RAID por software. Atualizado em 21.11.98.

- Soundblaster AWE, MINI COMO FAZER, por Marcus Brinkman `Marcus.Brinkmann@ruhr-uni-bochum.de`. Como instalar uma placa de som Soundblaster AWE 32/64. Atualizado em 11.01.98.
- StarOffice, MINI COMO FAZER, por Matthew Borowsk `mkb@poboxes.com`. Informações sobre a instalação do suíte StarOffice suite. Atualizado em 02.06.98.
- Terminal via Firewall, MINI COMO FAZER, por Barak Pearlmutte `bap@cs.unm.edu`. Como utilizar terminais através de firewall. Atualizado em 15.07.97.
- TkRat, MINI COMO FAZER, por Dave Whitinge `dave@whitinger.net`. Como instalar e utilizar o programa de mensagens TkRat. Atualizado em 02.02.98.
- Token Ring, MINI COMO FAZER, por Mike Eckhof `mike.e@emissary.aus-etc.com`. Como utilizar placas de redes Token Ring. Atualizado em 07.01.98.
- Ultra-DMA, MINI COMO FAZER, por Brion Vibbe `brion@pobox.com`. Como utilizar controladoras e dispositivos Ultra-DMA. Atualizado em 06.07.98.
- Novas Versões, MINI COMO FAZER, por Stein Gjoe `sgjoen@nyx.net`. Como estar informado das últimas atualizações do desenvolvimento do Linux. Atualizado em 03.02.98.
- Atualizações, MINI COMO FAZER, por Greg Loui `glouis@dynamicro.on.ca`. Como atualizar uma instalação de uma distribuição. Atualizado em 06.06.96.
- VAIO, MINI COMO FAZER, por Hideki Sait `hideki@chatlink.com`. Como instalar Linux em equipamentos Sony VAIO. Atualizado em 16.09.98.
- Vesafb, MINI COMO FAZER, por Alex Buel `alex.buell@tahallah.demon.co.uk`. Como utilizar o dispositivo vesafb. Atualizado em 02.08.98.
- VPN, MINI COMO FAZER, por Árpád Magosány `mag@bunuel.tii.matav.hu`. Como usar uma Rede Privada Virtual - VPN. Atualizado em 07.08.97.

- Sinais Visuais, MINI COMO FAZER, por Alessandro Rubin `rubini@linux.it`. Como desabilitar o alto falante do PC e habilitar sinais visuais. Atualizado em 11.11.97.

- Compartilhamento de Modems com Windows, MINI COMO FAZER, por Friedemann Baitinge `baiti@toplink.net`. Como configurar o Windows para utilizar um modem compartilhado em uma máquina Linux. Atualizado em 02.11.97.

- WordPerfect, MINI COMO FAZER, por Wade Hampto `whampton@staffnet.com`. Como configurar o Wordperfect para Linux. Atualizado em 13.08.97.

- Grande Cursor, MINI COMO FAZER, por Joerg Schneide `schneid@ira.uka.de`. Como utilizar grandes cursores com X Window. Atualizado em 11.08.97.

- XFree86-XInside MINI COMO FAZER, por Marco Melgazz `marco@techie.com`. Como converter XFree86 em modelos Xinside. Atualizado em 09.97.

- Título de um terminal X, MINI COMO FAZER, por Ric Liste `ric@giccs.georgetown.edu`. Como colocar textos na barra de títulos de um terminal X. Atualizado em 07.01.98.

- Instalação ZIP, MINI COMO FAZER, por John Wiggin `jwiggins@comp.uark.edu`. Como instalar o Linux em um dispositivo ZIP. Atualizado em 26.01.98.

- ZIP Drive, MINI COMO FAZER, por Kyle Dansi `dansie@ibm.net`. Provê uma referência rápida sobre a configuração e o uso do dispositivo ZIP drive com Linux. Atualizado em 26.08.98.

Como Fazer Especiais

- Alta Disponibilidade, COMO FAZER, por Harald Mil `hm@seneca.muc.de` está disponível em `http://metalab.unc.edu/pub/Linux/ALPHA/linux-ha/High-Availability-HOWTO.html`.

  Gráficos, MINI COMO FAZER, por Michael J. Hamme `mjhammel@graphics-muse.org` está disponível em `http://www.graphics-muse.org/linux/lgh.html`.

Itens Diversos e Notícias Legais

Caso você tenha alguma dúvida, sinta-se à vontade para escrever em inglês para `mdw@metalab.unc.edu`. A revisão do FAQ `Linux` está sendo coordenada por Ian Jackson, `ijackson@nyx.cs.du.edu`, com a ajuda de terceiros. Em português as mensagens podem ser enviadas para `info@conectiva.com.br`.

A menos que outras instruções sejam emitidas, os documentos Como Fazer `Linux` têm os direitos autorais reservados para os seus autores. Eles podem ser reproduzidos e distribuídos no todo ou em parte, em qualquer meio físico ou eletrônico, sem a permissão do autor. Traduções e trabalhos derivados têm permissão similar sem expressa autorização. Redistribuições comerciais são permitidas e encorajadas, porém o autor deverá ser notificado.

Resumidamente, pretendemos promover o uso destas informações por todos os canais possíveis. De qualquer forma, os direitos autorais são mantidos para os autores dos documentos Como Fazer, e gostaríamos de sermos notificados de qualquer redistribuição. Caso haja alguma dúvida, por favor contate em inglês Matt Welsh, o coordenador do Como Fazer `Linux`, em `mdw@metalab.unc.edu` ou em português a Conectiva Informática em `info@conectiva.com.br`.

RFCs

A seguinte lista de RFCs (Solicitações para Comentários) são mencionadas no decorrer deste Guia. Todas as RFCs estão disponíveis para recepção via FTP anônimo em `nic.ddn.mil`, `ftp.uu.net`. Para obter uma RFC via correio eletrônico, envie uma mensagem para `service@nic.ddn.mil`, colocando a expressão `send RFC-number.TXT` na linha de assunto da mensagem.

1340 Definindo números, *Postel, J.*, and *Reynolds, J.* A RFC Definindo Números define o significado dos números usados nos vários protocolos como os padrões dos números de portas TCP e UDP e os números de protocolo usados no cabeçalho do datagrama IP.

1144 Cabeçalhos comprimidos TCP/IP para ligações seriais de baixa velocidade. *Jacobson, V.* Este documento descreve o algoritmo usado na compressão de cabeçalhos TCP/IP nos protocolos CSLIP e PPP. Uma leitura muito útil.

1033 Guia de Operação de Administradores de Domínios, *Lottor, M.* Junto com as RFCs: 1034 e RFC 1035 é a fonte definitiva para o Sistema de Nomes de Domínios.

1034 Nomes de Domínios - Conceitos e Facilidades, *Mockapetris, P.V.* Parceira da RFC 1033.

1035 Nomes de Domínios - Implementação e especificação, *Mockapetris, P.V.* Parceira da RFC 1033.

974 Roteamento de mensagens e o sistema de domínios- *Partridge, C.* Esta RFC descreve o roteamento de mensagens na Internet. Leia para conhecer a história completa dos registro MX ...

977 Protocolo de Transferência de Notícias em Rede - NNTP *Kantor, B.*, and *Lapsley, P.* As definições do NNTp, o protocolo comum de transporte de notícias usado na Internet.

1094 NFS: Especificação do Protocolo de Sistemas de Arquivos em Rede. *Nowicki, B.* A especificação formal do NFS e protocolos de montagens (versão 2).

1055 Transmissões de datagramas IP sobre linhas seriais SLIP não padronizadas, *Romkey, J.L.* Descreve o SLIP, Protocolo Internet de Linhas Seriais.

1057 RPC: Especificação de Protocolo de Chamada de Procedimentos Remotos, Versão 2, *Sun Microsystems, Inc*

1058 Protocolo de Informações de Roteamento, *Hedrick, C.L.* Descreve o RIP, o qual é usado na troca dinâmica de informações de roteamento com redes locais e de grande abrangência.

821 Protocolo de Simples Transferência de Mensagens - SMTP, *Postel, J.B.* Define o SMTP, o protocolo de transferência de mensagens sobre TCP/IP.

1036 Padrões para troca de mensagens na USENET, *Adams, R.*, and *Horton, M.R.* Esta RFC descreve o formato de mensagens de notícias Usenet e como elas devem ser trocadas, inclusive em uma rede UUCP. Uma revisão desta RFC é esperada para breve.

822 Padrão das mensagens no formato ARPA Internet, *Crocker, D.* Esta é a fonte definitiva da sabedoria, bem, das mensagens de acordo com as RFC. Todo mundo conhece o tema, pouca gente leu a RFC.

Índice Remissivo

Símbolos
16450 UART, 79
16550 UART, 79
8250 UART, 79

A
accesso
 conceder, 206
 restringir, 207
acessando
 arquivos remotos, 202
acesso
 fornecendo, 135, 143
 UUCP, 233, 241
acesso remoto, 18, 24
algoritmo de alimentação, 326
`aliases`, 289
alimentação, notícias, 326
`amd`, 208
apelido
 de nome de máquina, 123
apelidos
 mensagem, 289
`aliases`, 290
ARPANET, 17
arquivo de caixa postal, 287
automontador, 208
autoritativo
 servidor de nomes, 54
AX.25, 21
Azevedo, Roberto, 142

B
Barber, Stan, 357
base de dados
 DNS, 53
BBS, 76
Berkeley Internet Name Domain, 111
biblioteca socket BSD, 26
BIND, 111, 117, 130
Biro, Ross, 27

C
C News, 329, 353
 `history` arquivo, 345
 arquivos de registros, 347
 atualizando a marca d'água, 366
 atualizando arquivo `active`, 348
 compactando lotes, 341
 enviando notícias , 342
 eutenho/meenvie, 341
 expirando, 342–345
 grupos moderados, 346
 guardando, 342
 limitando um envio de notícias, 346
 lista dos grupos atuais, 346
 loteando, 342
 parâmetros de lotes , 342
 Rede Local, 350
 suporte a NNTP, 360
 UUCP, 342
`cache` (opção BIND), 119

caixas postais
  montadas via NFS, 275
canônico
  nome, 53
`chat`, 153
CNAME (registro DNS), 123
colisão, Ethernet, 20
Collyer, Geoff, 323
Como fazer
  Serial, 75
compactação de cabeçalho de Van Jacobson, 146
compartilhando arquivo, 199
Comprimindo IP na Linha Serial, 133
comprimindo pacotes TCP/IP , 133
Conectiva Informática, 11
conector BNC, 19
configuração
  porta serial, 81
  servidor de nomes somente para cache, 132
  `smail`, *veja* `smail`
configuração de linha, 77
configurando
  C News, 353
  C News em uma Rede Local, 350
  caminho padrão de mensagens, 276
  correio em uma rede local, 274
  `dip`, 134
  DNS sobre SLIP/PPP, 132
  domínio padrão, 115
  leitor de notícias, 363
  mensagens em uma rede local, 278
  mensagens UUCP, 272
  notícias Usenet, 353
  o uso do servidor de nomes, 114
  porta serial, 80
  resolução de nomes de máquinas, 130
  `sendmail`, 295, 322
  servidor de nomes, 111, 117, 130
  servidor SLIP, 142
  SLIP, 131, 143
  UUCP, 209, 249
conversação
  SLIP, 137
Cox, Alan, 27
criação
  de zonas DNS, 57
criando
  sub-redes, 56
CSLIP, 23, 133, 143
C News
  atualizar a marca d'água, 345

D

DDI, 28
delegando
  subdomínios DNS, 56
depurando
  bases de dados DNS, 130
`/dev/cua*`, 78, 79
`/dev/modem`, 79
`/dev/ttyS*`, 78, 79
`dip`, 134, 143
`diphosts`, 142
`diplogin`, 142
direcionando uma mensagem para um arquivo, 287
discado IP, 131
dispositivo de discagem, 78
dispositivo de recebimento de ligações, 78
dispositivo, serial, 75
dispositivo,serial, 81

DNS, 57
  configurando o servidor, 117
  configurando um servidor, 130
  convertendo o arquivo `/etc/hosts`, 130
  criação de zonas, 57
  depurando as bases de dados, 130
  ferramentas, 130
  mapeamento reverso, 55
  mapeamento reverso de), 57
  registro de recurso, 120
  registro de recursos, 53
  RR, *veja* DNS, registro de recursos
  servidores de nomes raiz, 124, 129
  tempo de vida, 52, 121
  ttl, *veja* DNS, tempo de vida
  verificando, 127
  zona, 54, 56, 121
  zona de, 52
`dnswalk`, 130
domínio
  nível superior, 48
`domainname`, 184

E

endereço
  Ethernet, 20
  registro de recurso DNS, 123
endereço
  IP, 22
endereços
  mapeamento de nomes de máquinas para, 55
enfileirando mensagens, 280
entregando
  notícias, 327, 328
enviando mensagens para um comando, 287
envio
  IP, 21
espaço de nomes (DNS), 48
Ethernet, 19, 20
  colisão, 20
  endereço, 20
  endereço X endereço IP, 23
  thin, 19
evitando roteamentos circulares de mensagens, 282
execução
  remota, 15
exibir
  configuração da interface, 101
exportar um volume NFS, 206
`exports`, 206

F

FDDI, 20
FidoNet, 76
fila de mensagens, 282
Flintstone, Fred, 13
`fstab`, 202
FTP, localização do código do `Linux`, 28
funcionamento de linha, 132

G

`gated`, 44
`getty`, 236

H

Hankins, Greg, 75
hardware
  negociação de comunicação, 79, 80
  serial, 75, 81
hardware de acesso a rede, *veja* interface

HDLC, 145
`host.conf`, 112, 196
`hostcvt`, 130
`hosts`
  convertendo para arquivos mestres, 130
HOWTO, 3
  UUCP, 211

I
IDA, *veja* `sendmail`, IDA
IMAP, 275
`in-addr.arpa`, 55
Início de Autoridade, 54
`inetd`, 277
INN, 324, 357, 364
inter redes, 22
interface, 35
interface de dispositivo de controle, *veja* DDI
interface de rede, *veja* interface
Internet, 17
  conectando-se a, 131
  X inter redes, 22
InterNet News (INN), 324
IP, 21, 23
  discado, 131
  endereço, 22, 38
    e nomes de máquina, 47
  endereço X nome de servidor, 23
  endereços
    e nomes de máquinas, 55
  endereços multicast, 104
  envio, 21
  linha serial, 131
  máscara de rede, 102
  ponto de passagem, 21
  redes, 36, 39, 55
  roteamento, 22
  sub-rede, 40, 56
  sub-redes, *veja* IP, sub-rede

K
Kempen, Fred van, 18, 27

L
LAN, 17
  conectando, 155
  senhas, 193, 196
Lapsley, Phil, 357
LCP, *veja* Protocolo de Controle de Conexão
leitor de notícias
  configurando, 363
  criando bancos de dados de notícias, 367
  criando bases de dados de assuntos, 365
  criando bases de dados de temas, 365
  `nn`, 367
  `tass`, 364
  tema, 364
  temas, 364
  `tin`, 364
  `trn`, 365
linha serial
  arquivo de dispositivo, 78
  negociação de comunicação, 79, 80
  velocidade, 77
Linha Serial Comprimida IP, *veja* CSLIP
linha serial IP, *veja* PPP, *veja* SLIP
listando
  a configuração UUCP, 222
loteando
  notícias, 327, 342

M

mail
- bounce, 300
- postmaster, 282

manutenção do sistema, 29
mapeamento reverso, 55, 57
mensagem
- direcionando para um arquivo, 287
- enviando para um comando, 287

`cancel` mensagem de controle, 348
`rmgroup` mensagem de controle, 348
mensagens
- devolução, 282
- em rede local, 275
- escondendo o site, 275
- evitando a entrega via UUCP, 319
- fila, 280
- fila de, 271
- forçando a entrega via UUCP, 319
- nomes alternativos, 289, 290
- postmaster, 282
- reenviando, 288, 289
- servidor, 271

`mgetty`, 236, 238
modem
- velocidade, 77

montagem
- somente para leitura, 207

montando
- automaticamente, 208
- um volume NFS, 202

`mountd`, 205, 206
`mthreads`, 365
MX (Registro DNS), 123

N

`named`, 111, 117, 130
`named.boot`, 117, 120
negociação de comunicação, hardware, 79, 80
Net-1, 27
Net-2d, 27
Net-2d Depurado, 27
Net-2e, 28
Net-3, 28
Net-BSD, 28
news, 323
- mensagens de controle, 347
- Usenet, 324

NFS, 199, 208
- automontador, 208
- combinando identificações de usuários e grupos, 208
- Comparando identificações de usuários e grupos, 205
- expiração, 204
- exportar um volume, 206
- `exports`, 206
- limitações, 201
- Montagem direta vs. montagem lógica, 204
- montando um volume, 200, 202
- Restrições de tamanho de bloco, 203
- servidor, 201, 205
- tempo de espera, 204
- volume somente de leitura, 207

`nfsd`, 200, 205
NIS
- base de dados, 182
- código tradicional, 196
- e o resolver, 112
- escolhendo os mapas, 191
- localizando o servidor, 184
- mapas, 182

mostrar nomes curtos de mapas, 183
segurança, 188
senhas sombra, 195
servidor, 183, 188
`nn`, 367
nome alternativo, 53
nome canônico, 53
nome de domínio
NIS, 184
padrão, 115
nome de máquina
apelidos de, 123
canônico, 123
tratando nomes não qualificados, 292
nome de máquina canônico, 123
nomes de máquinas
obtendo endereços de, 55
pesquisa, 127
nomes e máquina
mapeamento para endereços de, 47
notícias, 328
acompanhamento, 364
adicionar um novo grupo, 348
algoritmo de alimentação, 326
alimentando, 325, 326, 328
alimentando as, 324
arquivando artigos, 342
`active` arquivo, 328
arquivos temporários, 328
artigo, 325
atualizando arquivo `active`, 348
cancelar um artigo, 348
distribuição, 327
eliminando notícias antigas, 328
enviando, 328, 356
eutenho/meenvie, 327
expiração, 366, 368
expiração de artigos antigos, 328
falsificando, 356
grupos, 325
história, 327
identificações de mensagens, 327
leitor, *veja* leitor de notícias
limitando a transferência, 327
loteando, 327
recebendo, 328, 356
remover um grupo antigo, 348
troca, 325
trocando, 324, 326, 328
versão C, *veja* Notícias C
Notícias C, 323
notação de quatro segmentos, 22
`nslookup`, 127, 129
`nsswitch.conf`, 191

O

obtendo o código fonte, 28
ordem em que os serviços de resolução são usados, 112

P

Pacotes Montagem/Desmontagem, 21
PAD, 21
padrão rota, *veja* rota, padrão
pesquisando endereços, 55
ponto de passagem, 21
POP, 275
porta, *veja* porta de rede
COM, 78
número de, 25
porta COM, 78
PPP, 23, 76, 131, 145
arquivos de opções, 149
autenticação, 159

programa de conversação , 153
roteando, 157
usando CHAP, 161
`pppd`, 165
`.ppprc`, 149
prevenção de spoofing, 113
`primary` (opção do BIND), 119
programa de conversação
UUCP, 224
programas de terminal, 76
protocol
X.25, 21
protocolo, 14
AX.25, 21
CSLIP, 23
Ethernet, 19
IP, *veja* IP
PPP, 23
SLIP, 23
UUCP, 242
protocolo
UDP, 24
Protocolo de Agência de Correio, 275
Protocolo de Controle de Transmissão, *veja* TCP
Protocolo de Datagrama de Usuário, *veja* UDP
Protocolo de Transferência de Notícias em Rede, *veja* NNTP
Protocolo Interativo de Acesso a Mensagens, 275
Protocolo Internet , *veja* IP
Protocolo Ponto-a-Ponto, *veja* PPP
protocolo TCP, 23
protocolos de
rede, 14
Protocolos de Controle de Redes, 146
PTR (registro DNS), 123

R
rádio amador, 21
rádio ham, 21
`rc.inet`, 205
rede, 13
interconexão, *veja* inter redes
interface de programação de, 26
mensagens em, 275
porta de, 24, 26
resolução de nome de máquina, 196
senhas, 196
troca de pacotes, 17
rede
Internet, 17
TCP/IP, *veja* TCP/IP
Rede Local, *veja* LAN
notícias, 350
redes
classes de, 36
conexões de, *veja* porta de rede
números de portas, 25
portas de, 25
serviços de, *veja* porta
UUCP, *veja* UUCP
reenviando
mensagens, 288, 289
registro de recursos , *veja* DNS, registro de recursos
remetendo
UUCP, 16
remota
sessão X11, 18
remoto
acesso a arquivo, 15
acesso arquivo, 199
sistema de arquivos, 202
resolução de nomes de máquinas, 47

`resolv.conf`, 114
resolvedor
robustez do, 116
variáveis de ambiente do, 114
resolver
biblioteca, 112
configurando, 112, 117
usando o NIS, 112
usando um servidor de nomes, 112
restringindo o acesso, 31
restringir acesso do superusuário `root`, 207
RIP, *veja* Protocolo de Informações de Rotas
`rmail`, 233
`rnews`, 233, 342
rota padrão, 37
roteamento
dinâmico, 157
evitando círculos, 282
roteamento
datagramas IP , *veja* IP, roteamento
`routed`, 44
RR, *veja* DNS, registro de recursos
RS-232, 79
RTS/CTS, 79

S
Salz, Rich, 357
`secundário` (opção BIND), 119
segurança, 143
acessos UUCP, 239, 241
nome falso do servidor, 113
PPP, 151
sistema de, 30
SLIP, 135
spoofing, 113
UUCP, 233, 241
sendmail
arquivos de suporte, 297
`sendmail`, 295, 322
CF, 296, 297, 303, 311, 312
DECnet, 309
`deliver`, 299
escrevendo mensagens em um arquivo, 308
estatísticas, 320
evitando a entrega via UUCP, 319
executando, 312
forçando a entrega via UUCP, 319
forçando mensagens, 318
forma de transporte, 307
formas de entrega, 319
gerando `sendmail.cf`, 311
geerando `sendmail.cf`, 312
gerenciando a fila, 320
IDA, 296
instalando, 310
localizações de Arquivos, 321
máquina retransmissora, 302
`mailertable`, 304
meios de entrega de mensagem, 306
nome de domínio totalmente qualificado, 307
nome de máquina desqualificado, 306, 307
nome de máquina local , 301
nomes alternativos, 308
nomes alternativos de usuários, 308, 309
nomes da máquina local, 301
nomes de máquinas sem qualificação, 319
nomes totalmente qualificados, 306

operando uma fila, 320
postmaster, 308
`postmaster`, 300
roteamento, 307
domínio, 317
máquina de roteamento otimizado, 302
UUCP, 306, 309, 319
`sendmail.cf`, 295
servidor de correio, 299
servidor de mensagens, 302, 317
servidores de correio, 304
site folha UUCP, 303
site Internet, 303
site remoto mal configurado, 318
tabelas, 302, 309, 315
tables, 296
testando, 312, 317
transporte, *veja* servidor de correio299
UUCP, 301, 306, 319
versão, 310
`sendmail.cf`, 295
server
`nfsd`, 200
serviços, 25
serviços conhecidos, 25
serviços de restrição de acessos, 30
serviços e números de portas, 25
servidor, 13
`sendmail`, 312
UUCP, 236, 241
servidor de nomes, 53
escravo, 120
autoritativo, 52, 56
cache, 119
configurando, 117, 130
primário, 119
resolução, 111
secundário, 119
somente para cache, 132
verificando o, 127
servidor de nomes autoritativo, 52, 54, 56, 121
servidor de nomes primário, 52
servidor de nomes secundário, 52
servidor de nomes somente para cache, 53, 132
servidores de nomes
raiz, 124, 129
sincronizando, 52
`setserial`, 80
sincronizando servidores de nomes, 52
Sistema de Arquivos de Redes, *veja* NFS
Sistema de Nomes de Domínio, *veja* DNS
sistema de segurança, 30
sistema, manutenção do, 29
site, 13
`slattach`, 133
SLIP, 23, 131, 143
deixando o usuário iniciar o, 135
SLIPDISC, 132
`smail`, 271, 293
apelidos de usuários, 289
`config` arquivo, 273, 274
arquivo `config`, 282
`paths` arquivo, 284, 286
arquivos de histórico, 278
BSMTP, 272, 291
caixa postal de usuário, 287
compilando, 280
`config` arquivo, 271
direcionando uma mensagem para um arquivo, 287

directors, 283
e SLIP/PPP, 285
em uma rede local, 274, 278
endereços locais, 286, 290
enfileirando mensagens, 280
enviando uma mensagem para um comando, 287
executando a fila, 281
gerenciando mensagens de um domínio, 275
listas de mensagens, 290
modos de entrega, 280
nome da máquina local, 273
nomes alternativos, 290
nomes de máquinas não qualificadas, 292
postmaster, 282
problemas & soluções, 278
reenviando, 288
routers, 283
roteamento, 282, 283, 286
evitando círculos, 282
Internet, 285
smart-host, 274
UUCP, 286
UUCP vs. SLIP, 285
roteando
UUCP, 284
SMTP, 277, 278, 291, 292
transports, 283
utilidades, 272
UUCP, 272, 274, 284–286, 290, 291
verificar fila de mensagens, 281
SMTP, 256
em lotes, 233
serviços, 277
SOA (registro DNS), 54, 121
socket, 26
somente leitura em volume NFS, 207
Spencer, Henry, 323
spoofing, 113
Storm, Kim F., 367
Política 'sub-redes são Locais', 110
subdomínios (DNS), 56
syslog, 208

T

T'so, Theodore, 80
tass, 364
taxa de bit, 77
taxa de transmissão, 77
TCP, 23, 24
UUCP, 232
TCP/IP, 17, 26, 35
telefone, enviando dados pelo, 23
telefone, enviando dados por, 131
thinnet, 19
tin, 364
tripwire, 31
trn, 365
troca
notícias, 325
trocando
notícias, 324, 326
tty, 77, 81
configuração de linha, 77
funcionamento de linha, 132

U

UART, 79
UDP, 24, 25
Um (registro DNS), 123
Universidade do Pantanal, 18
Urlichs, Matthias, 28
Usenet, 324

Utilidades Básicas de Rede, *veja* UUCP, HDB
`uucico`, 214
UUCP, 14, 17, 76, 249
  índice de tarefas temporárias, 228
  índice de trabalhos provisórios, 228
  acessando, 224
  acessando e depurando, 248, 249
  acionando, 214
  alternativas, 226
  anônimo, 241
  arquivo `passwd`, 238
  arquivos de configuração, 215, 219
  BNU, 209
  arquivo `dialcode`, 223
  checagem de seqüência de chamadas , 240
  `config` arquivo, 222
  configurando acessos, 237, 239
  configurando um servidor, 236, 242
  contas, 237
  conversação de acesso, 224
  correio, 233
  `dial` file, 230
  diretório temporário de tarefas, 212
  dispositivo, 224, 228, 230
  e getty `getty`, 236
  estatísticas, 248
  execução de comandos, 233
  HDB, 209, 249
  horário de chamada, 227
  intervalos de tentativas, 227
  linhas diretas, 233
  modem, 228, 230, 231
  número de telefone, 223
  negociação, 213
  nome de máquina, 222, 223
  notícias, 233
  `port` arquivo, 228
  prioridades, 212, 228
  problemas & Soluções, 246
  programas de conversação, 224, 226
  Projeto de Mapeamento, 220
  protocolo, 213, 245
    ajustes, 244
    seleção, 245
  protocolos, 242
  recebendo chamadas, 236
  reenvio, 235
  remetendo, 16
  restrições
    reenvio, 235
  restringindo
    horário de acesso, 227
    transferência de arquivos, 234
  restrita
    execução de comandos, 233
  segurança de acesso, 239, 241
  sistema remoto, 223, 228
  sistemas remotos, 222
  sobre TCP/IP, 232
  `sys` arquivo, 222
  tarefa, 211, 212
  Taylor, 210
  transferência de arquivos, 234
  usando `smail`, 272
  `uucico`, 213–215
  validando, 246
  verificação da sequência de conexão, 213
  verificação de seqüência de chamadas, 241

Versão 2, 209
UUCP anônimo , 241
`uugetty`, 236
`uux`, 342

V

Van Jacobson cabeçalho de compressão, 133
verificando
  `smail` configuração, 279
  nomes de máquinas, 127
  o servidor de nomes, 127
  `sendmail`, 312, 317
  UUCP, 222
verificando fila de mensagens, 281

X

X.25, 21

Z

Zen, 324
zona de, DNS, *veja* DNS, zona de