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

UNIX é uma marca registrada da Univel.
`Linux` não é uma marca registrada, e não tem conexão com UNIX™ ou Univel.

Kattreinstr. 38, 64295 Darmstadt, Germany
`okir@monad.swb.de`

Sumário

Lista de Figuras

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 fu-

turo.

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).

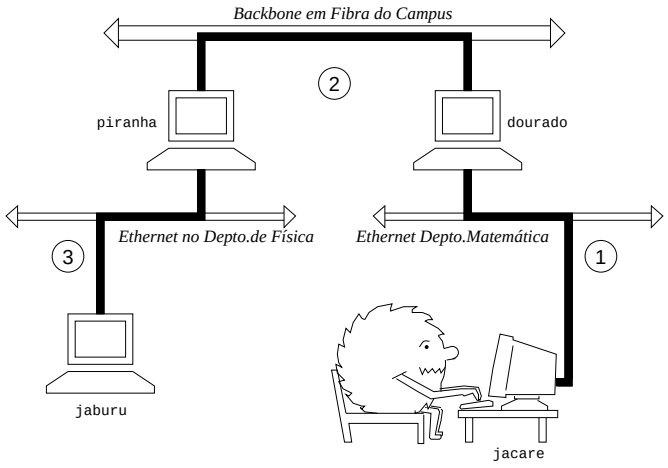

Figura 1.1: Os três passos para se enviar um datagrama de `jacare` a `jaburu`

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.

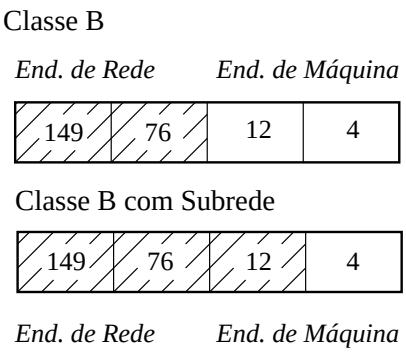

Figura 2.1: Criando sub-redes em uma rede classe B

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.

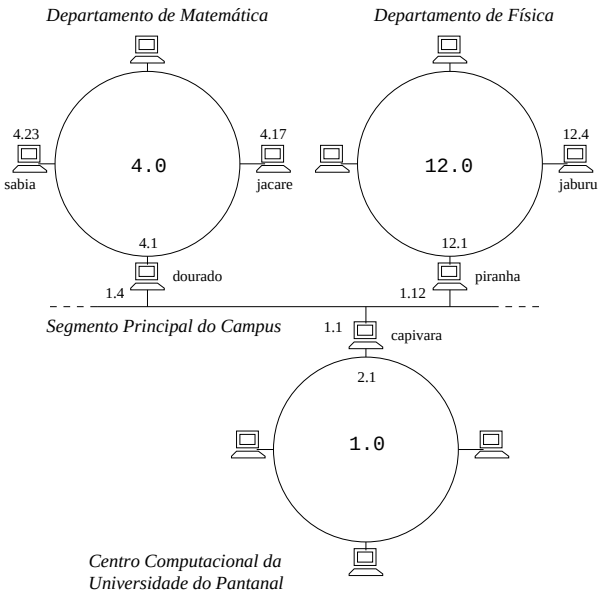

Figura 2.2: Parte da topologia de rede da Universidade do Pantanal

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 subdivida. 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

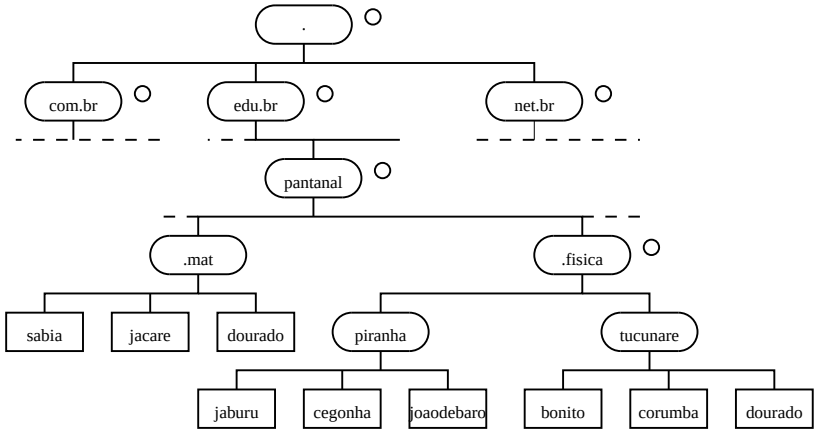

Figura 2.3: Parte do Espaço de Nome de Domínio

`.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

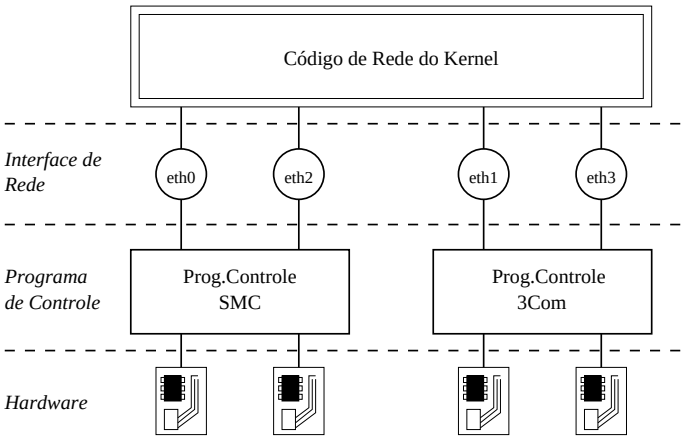

Figura 3.1: O relacionamento entre programas de controle, interfaces e o hardware

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 co-processador 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 dispo-

sitivos. 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