

**ADMINISTRAÇÃO
DE
SISTEMAS**

UNIX/LINUX

**Alexandre Pereira do Carmo
Rodolfo da Silva Villaca**

SUMÁRIO

1 - HISTÓRIA DO LINUX.....	8
2 - LINUX COMO SISTEMA OPERACIONAL.....	9
2.1 PRINCIPAIS CARACTERÍSTICAS COMENTADAS NO LINUX.....	9
2.2 LINUX COMO SOFTWARE GRATUITO.....	10
2.3 DOCUMENTAÇÃO DE PACOTES.....	10
2.4 COMO FAZER E FAQs.....	12
2.5 O COMANDO LOCATE.....	12
2.6 INFO.....	13
2.7 LISTAS DE DISCUSSÃO LINUX.....	13
3 - O QUE É UMA DISTRIBUIÇÃO.....	14
3.1 O MESMO, PORÉM DIFERENTE.....	14
3.2 AS PRINCIPAIS DISTRIBUIÇÕES.....	14
3.3 RED HAT LINUX.....	14
3.4 O QUE HÁ DE NOVO NO RED HAT 6.0.....	15
4 - O PC MÍNIMO PARA LINUX.....	16
4.1 ACESSÓRIOS IDEAIS PARA UM SERVIDOR EM SUA INTRANET.....	16
4.2 VERIFICANDO A COMPATIBILIDADE DE SEU HARDWARE.....	16
4.3 REGISTRANDO SUAS INFORMAÇÕES DE HARDWARE.....	16
5 - O QUE É UM COMANDO LINUX?	17
5.1 SU.....	17
5.2 PWD, CD.....	17
5.3 LS.....	17
5.4 MKDIR.....	18
5.5 MORE E LESS.....	18
5.6 FIND.....	19
5.7 GREP.....	19
5.8 TAR.....	20
5.9 GZIP.....	21
5.9.1 <i>Combinando gzip e tar.....</i>	<i>21</i>
5.10 CP.....	22
5.10.1 <i>Cópia avançada.....</i>	<i>22</i>
5.10.2 <i>Evitando erros.....</i>	<i>23</i>
5.11 RM.....	24
5.11.1 <i>Excluindo diretórios inteiros.....</i>	<i>24</i>
5.12 RMDIR.....	25
5.13 MV.....	25
5.14 CAT.....	26
5.15 CHGRP.....	26
5.16 CHMOD.....	27
5.17 CHOWN.....	28
5.18 DU.....	28
5.19 DATE.....	28
5.20 FILE.....	28
5.21 INIT / TELINIT.....	29
5.21.1 <i>Níveis de execução.....</i>	<i>29</i>
5.21.2 <i>Iniciando.....</i>	<i>29</i>
5.21.3 <i>Telinit.....</i>	<i>29</i>
5.22 RUNLEVEL.....	29
5.23 TYPE.....	30
5.24 DIFF.....	30

5.25	ZIP	30
5.26	SORT	30
5.27	CUT	31
5.28	TR	32
5.29	OUTROS COMANDOS	32
5.30	CRIANDO VÍNCULOS SIMBÓLICOS	32
6 -	PROCESSADORES DE TEXTOS SOB LINUX.....	33
6.1	PICO.....	33
6.1.1	Acionando o Pico	33
6.1.2	Comandos de movimentação básica do cursor.....	33
6.1.3	Comandos de movimentação da tela.....	34
6.1.4	Comandos de deleção.....	34
6.1.5	Errou? Como cancelar uma operação	34
6.1.6	Outros comandos de edição.....	34
6.1.7	Busca/substituição.....	34
6.1.8	Inserindo um arquivo no meio de um texto.....	34
6.1.9	Refresh	35
6.1.10	Salvar.....	35
6.1.11	Salvar e sair/ Sair sem salvar.....	35
7 -	ENTENDENDO O SHELL	36
7.1	COMPARANDO SHELLS.....	36
7.2	EXPERIMENTANDO DIFERENTES SHELLS.....	37
7.3	VISÃO GERAL DO SHELL BASH.....	37
7.4	DEFININDO VARIÁVEIS DE AMBIENTE NO BASH	37
7.5	SHELL SCRIPT	38
7.6	REDIRECIONAMENTO DE ENTRADA E SAÍDA	38
7.6.1	Dutos ou Pipes (/).....	39
7.7	TÉRMINO DE NOME DE ARQUIVO	39
7.8	ALIAS DE COMANDO.....	40
7.9	EDIÇÃO DE LINHA DE COMANDOS.....	40
7.10	LISTA DE HISTÓRICO DE COMANDOS	40
7.11	CONTROLE DE TAREFA.....	41
7.12	EXPANSÃO DE PADRÃO	42
7.13	EXPANSÃO DE NOME DE PATH	42
7.14	EXPANSÃO DE CHAVE.....	43
7.15	SUBSTITUIÇÃO DE COMANDOS.....	43
8 -	DICAS SOBRE MODO TEXTO	44
8.1	CONFIGURANDO SEU PATH.....	44
8.2	QUAL A VERSÃO DO LINUX QUE ESTOU A USAR NA MINHA MÁQUINA?	44
8.3	TUDO SOBRE LINUX PARA QUEM ESTÁ MIGRANDO DO DOS	44
8.3.1	Comparando os comandos.....	45
8.3.2	Arquivos.....	46
8.3.3	Links Simbólicos.....	46
8.3.4	Multi-tarefa.....	48
8.3.5	Diretórios.....	49
8.4	COMANDOS DO PROGRAMA VI	50
8.5	COMO POSSO SABER QUANTOS HARD LINKS TEM UM ARQUIVO E QUANTOS ELE PODE TER.....	51
8.6	MUDANDO O RELÓGIO DE SEU LINUX	51
8.7	MUDANDO O EDITOR DE TEXTO PADRÃO	52
8.8	SHELL SCRIPTS - UTILIDADES E MAIS UTILIDADES.....	52
8.8.1	Backup para um FTP	52
8.8.2	Comandos do DOS no Linux	53
8.9	COMO ALTERAR A COR DO FUNDO E DA LETRA NO CONSOLE?	54
8.10	COMO MUDO O IDIOMA DE MEU CONECTIVA LINUX?	54
8.11	COMO CONFIGURO A PROTEÇÃO DE TELA NA CONSOLE?	55
8.12	COMO AGENDAR TAREFAS DE SISTEMA (/ETC/CRONTAB)?	55

8.13	COMO EXECUTAR UM ARQUIVO?	56
8.14	O QUE POSSO APAGAR PARA LIBERAR ESPAÇO EM DISCO?	56
8.15	COMO USAR O TAR ?	56
8.16	COMO FAÇO PARA DEIXAR O 'LS' SEMPRE COLORIDO?	57
9 -	ADMINISTRAÇÃO DE SISTEMA GERAL.....	58
9.1	INICIALIZAÇÃO DO SISTEMA	58
9.2	PROGRAMANDO TAREFAS COM CROND.....	59
10 -	DICAS SOBRE O SISTEMA	62
10.1	ONDE IR BUSCAR OS HOWTOS E OUTRA DOCUMENTAÇÃO?.....	62
10.2	ONDE IR BUSCAR MATERIAL SOBRE O LINUX POR FTP?	63
10.3	PERMISSÕES.....	64
11 -	CONFIGURAÇÃO DO SISTEMA COM O PAINEL DE CONTROLE.. ERRO! INDICADOR NÃO DEFINIDO.	
11.1	CONFIGURAÇÃO DE IMPRESSORA	ERRO! INDICADOR NÃO DEFINIDO.
11.2	CONFIGURAÇÃO DO KERNEL.....	ERRO! INDICADOR NÃO DEFINIDO.
11.2.1	<i>Alterando as Opções de Módulos.....</i>	<i>Erro! Indicador não definido.</i>
11.2.2	<i>Alterando Módulos.....</i>	<i>Erro! Indicador não definido.</i>
11.2.3	<i>Adicionando Módulos.....</i>	<i>Erro! Indicador não definido.</i>
11.3	HORÁRIO E DATA.....	ERRO! INDICADOR NÃO DEFINIDO.
12 -	CD-ROMS, DISQUETES, DISCOS RÍGIDOS E SISTEMAS DE ARQUIVOS - UMA VISÃO GERAL	68
12.1	REVISANDO O SISTEMA DE ARQUIVOS - VISÃO GERAL.....	69
12.2	ADICIONADO PONTOS DE MONTAGEM NFS - VISÃO GERAL.....	70
12.3	OPÇÕES GERAIS	71
13 -	ADMINISTRAÇÃO DE SISTEMA GERAL.....	72
13.1	GERENCIAMENTO DE USUÁRIOS.....	72
13.2	CONFIGURANDO DIRETÓRIO DE BASE PADRÃO.....	73
13.3	REMOVENDO USUÁRIOS	73
13.4	GERENCIANDO GRUPOS.....	74
13.5	CHECANDO A CONSISTÊNCIA DE SISTEMA DE ARQUIVOS	75
13.6	INICIALIZAÇÃO DO SISTEMA	75
13.7	PROGRAMANDO TAREFAS COM CROND.....	76
13.8	GERENCIANDO LOGS	78
14 -	DICAS SOBRE HARDWARE.....	81
14.1	UTILIZANDO UM DISCO FLEXÍVEL NO LINUX	81
15 -	INSTALAÇÃO DE PLACA DE REDE ADICIONAL.....	82
16 -	CONFIGURAÇÃO DE IMPRESSORA.....	84
16.1	COMO CONFIGURAR UMA IMPRESSORA REMOTA EM UMA REDE LINUX ?	85
17 -	IMPORTANTES FUNCIONALIDADES DE REDE.....	87
17.1	O SUPERSERVIDOR INETD.....	87
17.2	A FUNCIONALIDADE TCPD DE CONTROLE DE ACESSO	89
17.3	RPC - CHAMADA DE PROCEDIMENTO REMOTO.....	92
18 -	DNS	94
18.1	INTRODUÇÃO E CONCEITOS.....	94
18.2	INSTALANDO E CONFIGURANDO O DNS.....	94
18.2.1	<i>Instalando os Pacotes.....</i>	<i>94</i>
18.3	CONFIGURANDO O SERVIDOR DNS.....	95
18.3.1	<i>Configurando Mapas de IPs Reversos.....</i>	<i>96</i>
18.3.2	<i>Configurando um Servidor Secundário.....</i>	<i>96</i>
18.3.3	<i>Forward Zones.....</i>	<i>96</i>
18.3.4	<i>Repetidores</i>	<i>97</i>

18.3.5	Funcionalidades.....	97
18.3.6	Alocação de Faixas de IP.....	97
18.4	INICIALIZANDO O SERVIÇO.....	98
18.5	ARQUIVOS DE CONFIGURAÇÃO DO BIND.....	98
18.5.1	O Arquivo <i>/etc/named.conf</i>	98
18.5.2	O Arquivo <i>/var/named/nome-do-dominio</i>	99
18.5.3	O Arquivo <i>/var/named/named.local</i>	99
18.5.4	O Arquivo <i>/var/named/named.ca</i>	99
18.6	CONFIGURAÇÃO DOS CLIENTES.....	100
18.6.1	Configuração Através do <i>Linuxconf</i>	101
18.6.2	O Arquivo <i>/etc/resolv.conf</i>	101
19 - NIS - SISTEMA DE INFORMAÇÕES EM REDE.....		102
19.1	CONHECENDO O NIS.....	102
19.2	O CLIENTE NIS.....	105
19.3	SERVIDOR NIS.....	105
19.4	CONFIGURANDO UM CLIENTE NIS COM NYS.....	107
19.5	ESCOLHENDO OS MAPAS CORRETOS.....	108
19.6	USANDO OS MAPAS <i>PASSWD</i> E <i>GROUP</i>	110
19.7	UTILIZANDO NIS COM SUPORTE A SENHAS SOMBRA.....	112
19.8	UTILIZANDO O TRADICIONAL CÓDIGO NIS.....	112
20 - NFS.....		114
20.1	INTRODUÇÃO E CONCEITOS.....	114
20.2	INSTALANDO O NFS.....	114
20.2.1	Instale o servidor NFS.....	114
20.3	CONFIGURANDO O SERVIDOR NFS.....	114
20.4	CONFIGURANDO UM CLIENTE NFS.....	116
21 - SAMBA.....		117
21.1	CONFIGURANDO O SERVIDOR SAMBA.....	117
21.2	INSTALANDO O SAMBA.....	117
21.3	CONFIGURAÇÃO.....	117
21.3.1	Senhas Criptografadas.....	118
21.3.2	Configurações Básicas.....	119
21.3.3	Compartilhando um diretório.....	120
21.4	MONTANDO UM VOLUME SAMBA.....	120
21.5	INICIANDO O SAMBA.....	120
21.6	CONFIGURAÇÃO DO CLIENTE.....	121
21.7	UTILIZANDO O SWAT.....	121
22 - SERVIDOR WEB.....		123
22.1	O PROTOCOLO HTTP.....	123
22.2	O APACHE.....	124
22.2.1	Instalando o Apache.....	124
22.2.2	Configurando o Apache.....	124
22.3	127	
23 - SERVIDOR FTP.....		128
23.1	O WU-FTPD.....	128
23.1.1	Instalação e Configuração.....	128
23.1.2	Acessos Anônimos.....	130
23.1.3	Permitindo Envio de Arquivos.....	130
23.1.4	Arquivos de Mensagens e Banners.....	132
23.1.5	Arquivo de Banner.....	132
23.1.6	O Arquivo <i>.message</i>	132
24 - SERVIDOR PROXY.....		133
24.1	CACHING.....	133

24.2 O SQUID.....	133
24.2.1 Instalação e Configuração.....	134
24.2.2 Memória para Cache.....	134
24.2.3 Arquivos de Cache.....	135
24.2.4 Controle de Acesso.....	135

25 - CONFIGURAÇÃO BÁSICA DO SENDMAIL..... 137

25.1 APRESENTAR SEU SISTEMA COMO.....	137
25.2 ACEITAR EMAIL PARA SEU DOMÍNIO.....	137
25.3 SERVIDOR DE EMAIL.....	137
25.4 ROTEADOR DE EMAIL.....	137
25.5 PROTOCOLO DO ROTEADOR DE EMAIL.....	138
25.6 FUNCIONALIDADES.....	138
25.7 DIVERSOS.....	139
25.8 APELIDOS PARA SEU SISTEMA.....	140
25.9 ROTEAMENTO ESPECIAL.....	140
25.9.1 Razões diferentes para se fazer roteamento especial.....	140
25.9.2 Tarefas.....	141
25.9.3 Arquivos.....	141
25.9.4 Mailertable é uma opção.....	141
25.10 ROTEAMENTO COMPLEXO.....	142
25.10.1 Introdução.....	142
25.10.2 Princípios.....	142
25.11 TAREFAS.....	142
25.11.1 Esta regra está ativa.....	143
25.11.2 Para: original.....	143
25.11.3 Para: novo.....	143
25.11.4 Reenviar para servidor (opc.).....	143
25.11.5 Transporte.....	143
25.11.6 Comentário.....	143
25.12 REGRAS DE MASCARAMENTO.....	144
25.12.1 Princípios.....	144
25.12.2 De: original.....	144
25.12.3 De: novo.....	144
25.12.4 Comentário.....	144
25.12.5 Alguns exemplos.....	145
25.13 DOMÍNIO VIRTUAL DE EMAIL.....	145
25.13.1 Princípios.....	145
25.14 A CAIXA DE DIÁLOGO DA DEFINIÇÃO DE DOMÍNIO VIRTUAL.....	146
25.14.1 Um nome.....	146
25.14.2 Destino de retorno.....	146
25.14.3 Arquivos de apelidos (opc.).....	147
25.14.4 Apelidos para este domínio.....	147
25.15 UMA NOTA ANTES DE INICIAR.....	147
25.16 COMO DEFINIR UM DOMÍNIO VIRTUAL DE EMAIL.....	147
25.16.1 Como adaptar os clientes POP.....	147
25.16.2 Como instalar o servidor.....	148
25.17 COMO DEPURAR UMA CONFIGURAÇÃO.....	149
25.17.1 Checando o DNS.....	149
25.17.2 Como adicionar usuários POP.....	150
25.17.3 Co-administrador.....	151
25.17.4 Como um usuário pode trocar a sua senha.....	151
25.18 ALGUNS PROBLEMAS.....	151
25.18.1 Todos os emails do domínio saem mascarados.....	151
25.19 APELIDOS.....	152
25.19.1 Tarefas.....	152
25.19.2 Pseudo-usuário.....	152
25.19.3 Lista dos pseudo-usuários mais conhecidos.....	153
25.19.4 Apelidos para contas administrativas.....	153
25.19.5 O pseudo-usuário administrador.....	153

25.19.6	Onde definir os apelidos.....	153
25.19.7	Administrador de listas de discussão	154
25.20	CON.GURAÇÃO DE FILTROS DE SPAM.....	154
25.20.1	Princípio.....	154
25.20.2	Usar nomes ou números IP	154
25.20.3	Especificando uma rede	155
25.20.4	Especificando uma máquina ou um domínio	155
25.21	DEFININDO REMETENTES REJEITADOS.....	155
25.21.1	Definindo _repetir para_ por IP	155
25.21.2	Definindo _repetir para_ por nome.....	155
25.21.3	Definindo _repetir para_ máquinas.....	155
26 -	ADICIONANDO CONEXÕES PPP / SLIP PARA MODEM.....	156

1 - História do Linux

Origem no UNIX que foi desenvolvido nos anos 70.

O UNIX é muito utilizado hoje no ambiente corporativo, educacional e agora no ambiente desktop.

Linux é uma cópia do Unix feito por Linus Torvalds, junto com um grupo de hackers pela Internet. Pretende-se que ele siga conforme com o padrão POSIX, padrão usado pelas estações UNIX.

Linus Torvalds que na época era um estudante de ciência da computação na Finlândia criou um clone do sistema Minix (sistema operacional desenvolvido por Andrew Tannenbaun que era semelhante ao UNIX) que o chamou de Linux. Hoje ele ainda detém o controle do Kernel do sistema.

Em março de 1992 surge a versão 1.0 do Linux.

Estima-se que sua base de usuários se situe hoje em torno de 10 milhões.

Ele ainda não se enquadra como sendo um sistema operacional UNIX.

Ele inclui proteção entre processos (crash protection), carregamento por demanda, redes TCP/IP, além de nomes de arquivos com até 255 caracteres, multi-tarefa real, suporte a UNICODE, shared libraries, memória virtual, etc.

<p>Processo - Resumidamente... um programa em execução. As informações sobre vários processos em execução ficam armazenadas na tabela de processos. O Sistema Operacional controla o tempo de execução de cada processo, e ele é responsável por iniciar, parar/suspender e terminar a execução de processos. Os usuários também pode interferir na administração de processos.</p> <p>Todo processo tem um processo "pai" que o chamou, portanto o processo tem nome, número de identificação, grupo ao qual pertence, etc. e está inserido na árvore hierárquica de processos. O processo pai é chamado de init.</p>

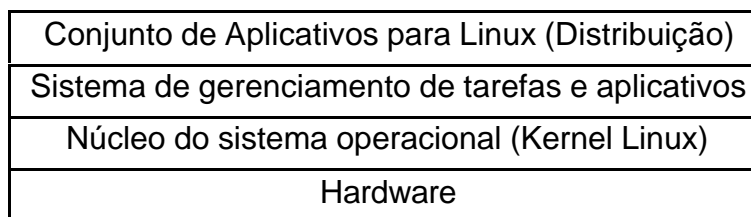
2 - Linux como sistema operacional

Linux se refere ao Kernel

O conjunto de aplicativos que são executados no Kernel se chamam distribuição.

A função do Kernel é ser a interface entre o Hardware e os sistemas de gerenciamento de tarefas e aplicativos.

O Kernel do Linux é mais leve que o de outros sistemas operacionais para servidores. São 12 milhões de linhas de código, enquanto outras versões do Unix têm 30 milhões e o Windows NT, 50 milhões. "Isso significa que o Linux é três vezes mais rápido que outros Unix e até cinco vezes mais rápido que o Windows NT".



2.1 Principais características comentadas no Linux

Multiusuário: Permite que vários usuários possam rodar o sistema operacional, e não possui restrições quanto à licença. Permite vários usuários simultâneos, utilizando integralmente os recursos de multitarefa. A vantagem disso é que o Linux pode ser distribuído como um servidor de aplicativos. Usuários podem acessar um servidor Linux através da rede local e executar aplicativos no próprio servidor.

Multiplataforma: O Linux roda em diversos tipos de computadores, sejam eles RISC ou CISC.

Multitarefa: Permite que diversos programas rodem ao mesmo tempo, ou seja, você pode estar imprimindo uma carta para sua vovó enquanto trabalha na planilha de vendas, por exemplo. Sem contar os inúmeros serviços disponibilizados pelo Sistema que estão rodando em background e você provavelmente nem sabe.

Multiprocessador: Permite o uso de mais de um processador. Já é discutida, há muitos anos, a capacidade do Linux de poder reconhecer mais de um processador e inclusive trabalhar com SMP, clusters de máquinas, na qual uma máquina central controla os processadores das outras para formar uma só máquina.

Protocolos: Pode trabalhar com diversos protocolos de rede (incluindo o TCP/IP que é nativo Unix).

Sistemas de arquivos: Suporta diversos sistemas de arquivos, incluindo o HPFS, DOS, CD-ROM, Netware, Xenix, Minix, etc.

<p>Sistema de arquivos - É uma forma de armazenamento de arquivos em estruturas (na maneira hierárquica) de diretórios. Assim, o usuário não precisa necessitar conhecer detalhes técnicos do meio de armazenamento. Ele apenas precisa necessitar conhecer a estrutura (árvore) de diretórios para poder navegar dentro dela e acessar suas informações.</p>
--

Consoles virtuais: Permite que o usuário tenha mais de um console para trabalhar, sendo que em cada console você pode ter diversas tarefas sendo executadas em background e mais em foreground (segundo plano e primeiro plano).

Fontes TrueType: Fontes TrueType são agora suportadas pelo Conectiva Linux. A carga dinâmica de fontes é suportada e pode ser usada como um servidor de fontes em uma máquina local. Nota: os usuários que estejam efetuando uma atualização terão que atualizar os caminhos padrões das fontes. Para tanto, edite o arquivo `/etc/X11/XF86Config`. Procure no arquivo até encontrar os caminhos configurados. Altere para o seguinte `unix/-1`. Deve-se ainda estar seguro de que o xfs, o Servidor de Fontes X, esteja sendo executado. Através do comando `/sbin/chkconfig -add xfs` pode-se garantir que ele seja inicializado a cada vez que o Linux seja carregado.

2.2 *Linux como software gratuito*

Uma forma de combater as práticas monopolistas da Microsoft.

Existem rumores que a Microsoft irá alterar os termos de licenciamento de seus produtos. Na nova versão, o software não será mais adquirido, e, sim, licenciado em bases anuais, exigindo pagamento de uma nova licença para uso continuado. Isso poderá provocar a marginalização das populações ou países que não tenham os recursos necessários para investimentos nesta área.

O criador do movimento pelo software aberto e livre foi Richard Stallman. Em determinada ocasião, ele precisou corrigir o driver de uma impressora que não estava funcionando. Solicitou então, ao fabricante do driver o código fonte do programa para que pudesse realizar as correções necessárias. Para sua surpresa, o pedido foi negado. Daí ele iniciou então um esforço gigantesco para conceder versões abertas para todas as categorias de software existentes, comercializadas sem acesso ao código fonte.

Richard Stallman fundou a FSF – Free Software Foundation. A FSF criou os aplicativos utilizados por todos os sistemas semelhantes ao Unix, como Linux e FreeBSD, hoje tão populares.

Para evitar que alguém obtivesse o programa com o seu código fonte, fizesse alterações e se declarasse como dono do produto, ele estabeleceu a forma sob a qual esses programas poderiam ser distribuídos.

O documento especifica que o programa pode ser usado e modificado por quem quer que seja, desde que as modificações efetuadas sejam também disponibilizadas em código fonte. Esse documento chama-se GNU (General Public License).

O Kernel do Linux também é distribuído sob a GNU (General Public License).

O Kernel do Linux, associado a esses programas, tornou possível a milhões de pessoas o acesso a um excelente ambiente computacional de trabalho e que melhora a cada dia.

O Linux, na pessoa de seu criador e coordenador, soube melhor aglutinar o imenso potencial de colaboradores da Internet em torno de seu projeto. Contribuições são aceitas, testadas e incorporadas ao sistema operacional e uma velocidade nunca vista.

2.3 *Documentação de Pacotes*

Muitos programas têm o arquivo README e outras documentações como parte integrante do pacote. O *Conectiva Linux* utiliza normalmente os subdiretórios sob `/usr/doc` como local padrão para o armazenamento, sem que seja necessário instalar todos os fontes para acessar a documentação; porém o nome do subdiretório depende do nome do pacote e

da sua versão. Por exemplo, o pacote `zip` na sua versão `2.1`, terá como caminho para acesso à sua documentação o seguinte: `/usr/doc/zip-2.1`.

Em sua maioria a documentação está em arquivos padrão ASCII, os quais podem ser visualizados com os comandos `more` `arquivo` ou `less` `arquivo`.

Caso você esteja procurando pela documentação de um comando específico (ou arquivo) e não em qual pacote ele está contido, será possível descobri-la de forma simples. Por exemplo, para conhecer onde está a documentação do arquivo `/usr/bin/at` utilize o comando:

```
rpm -qdf /usr/bin/at
```

Este comando retornará uma lista de toda a documentação (inclusive páginas de manual) do pacote que contenha o arquivos `/usr/bin/at`. O RPM é capaz ainda de uma série de outras funcionalidades. Para maiores informações sobre ele, veja o *Guia do Usuário do Conectiva Linux*.

2.4 Como Fazer e FAQs

Caso tenha sido selecionado durante a instalação, o conteúdo do Projeto de Documentação do Linux (LDP) estará disponível no diretório `/usr/doc` de seu sistema.

O diretório `/usr/doc/HOWTO` contém versões em arquivos ASCII de todos os Como Fazer disponíveis na época de impressão do CD-ROM. Estes arquivos podem ser lidos através do comando `less`.

Ex.: `less Tips-HOWTO`

Você também pode encontrar arquivos com extensão `.gz`. Eles estão compactados com `gzip` para economia de espaço, sendo necessário então descompactá-los antes de sua utilização. Para utilizá-los pode-se executar o comando `gunzip` para descompactá-los ou então utilizar o comando `zless` que lista os arquivos sem criar uma versão descompactada em seu disco :

Ex.: `zless HAM-HOWTO.gz`

O comando `zless` usa as mesmas teclas de operação que o comando `less`, permitindo a navegação pelo documento.

O diretório `/usr/doc/HOWTO/mini` contém versões ASCII de todos os mini-Como Fazer disponíveis. Não estão compactados e podem ser acessados normalmente com `more` ou `less`. `/usr/doc/HTML` contém versões HTML de todos os Como Fazer e dos guias Instalação do Linux e Linux para Iniciantes. Para visualizá-los basta utilizar um browser WWW de sua preferência. Por exemplo:

```
cd /usr/doc/HTML netscape index.html
```

O diretório `/usr/doc/FAQ` contém uma versão ASCII (e algumas versões HTML) de FAQs mais utilizados, incluindo o *faq* do *Conectiva Linux*.

O diretório `/usr/doc/HOWTO/translations/pt_BR/` possui diversos documentos traduzidos para o português.

2.5 O Comando locate

Quando não se conhece o nome completo do comando ou arquivo que se busca, pode-se facilmente encontrá-lo através do comando `locate`. Este comando utiliza uma base de dados para localizar todos os arquivos no sistema. Normalmente esta base é construída automaticamente toda noite, desde que o Linux esteja ativo. Caso isso não ocorra é possível criá-la através do comando (executado como superusuário `root`):

```
locate bison
```

E a resposta será algo como:

```
/usr/bin/bison  
/usr/include/bison2cpp.h  
/usr/info/bison.info.gz  
/usr/lib/bison.hairy
```

A resposta é fornecida através do nome e rota completa do arquivo.

2.6 Info

Enquanto as páginas de manual utilizam técnicas simples de apresentação de documentos, as funções `info` são muito mais poderosas. Elas provêm funções de hipertexto, tornando mais simples a leitura de grandes documentos, além de disponibilizarem diversas ferramentas para a criação de documentos. Há diversos documentos em formato `info` no *Conectiva Linux* (especialmente alguns do Projeto GNU).

Para acessar a documentação, basta utilizar o programa `info` sem argumentos. Será apresentada uma lista dos documentos disponíveis. Caso nada seja encontrado é porque provavelmente não foram instalados os pacotes de documentação, o que pode ser feito a qualquer momento através do utilitário RPM.

Caso se tenha conhecimento de `emacs`, pode-se acessar a documentação `info` diretamente dentro do `emacs` através da seqüência das teclas Ctrl-h i.

Todo texto que esteja destacado de forma luminosa é uma ligação que leva a alguma informação adicional. Utilize Tab para mover o cursor para a ligação e pressione Enter para ativá-lo. Pressionando-se p retorna para a página anterior, n vai para a próxima página e u sobe um nível. Para sair basta pressionar Ctrl-x Ctrl-c (control-x seguido de control-c).

A melhor maneira de aprender como utilizar a documentação do `info` é acessar o programa e verificar as informações disponíveis na primeira tela.

2.7 Listas de Discussão Linux

A Conectiva mantém listas para discussão sobre assuntos gerais do Linux:

- <http://linux-br.conectiva.com.br>
- <http://listas.conectiva.com.br/listas>

É mantida ainda a lista `Conectiva-Anúncios` (inscrições na primeira página da Conectiva) para recepção de novidades sobre os lançamentos e os trabalhos desenvolvidos pela Conectiva.

Todas as listas são abertas ao público em geral.

3 - O que é uma distribuição

Ao "kernel" é freqüentemente acrescentado uma série de aplicações, formando um sistema ou distribuição Linux.

Distribuição nada mais é que um pacote do kernel do sistema operacional mais os programas que o acompanham. Este pacote, incluindo as ferramentas necessárias para sua instalação, é chamado de distribuição.

Uma distribuição atende a uma determinada necessidade.

As distribuições podem ser produzidas em diferentes versões do Kernel, podem incluir diferentes conjuntos de aplicativos, utilitários, ferramentas e módulos de driver, e podem oferecer diferentes programas de instalação e atualização para facilitar o gerenciamento do sistema.

3.1 O mesmo, porém diferente

Na maioria das distribuições existe um conjunto comum de programas básicos, utilitários e bibliotecas, que os projetistas de aplicativos podem esperar encontrar em um sistema Linux.

Padrão seguido pelas distribuições = Linux file system standart.

3.2 As principais distribuições

Red Hat – Famoso por suas ferramentas de instalação e atualização do sistema operacional e por seu sistema bem projetado de instalação, desinstalação e controle de pacotes de aplicativos de software.

Slackware – Era a distribuição mais popular. Não vem com RPM. Ganha em performance, mas peca na interatividade. Usado mais para servidores de rede.

Caldera OpenLink – O OpenLink 1.3 inclui o "K" Desktop Environment, uma licença não-comercial do StarOffice for Linux, Suporte Netware, uma licença do DR-DOS para compatibilidade DOS.

S.u.S.E Linux – É uma conhecida distribuição de Linux, disponível principalmente na Europa e oferecida nas versões em inglês e alemão.

Debian / GNU – Não possui uma organização comercial patrocinadora. É produzida por uma equipe de voluntários. Utiliza seu próprio sistema de gerenciamento de pacotes.

3.3 Red Hat Linux

O que popularizou o Red Hat foi seu sistema de gerenciamento de pacotes. Esse sistema permite que os aplicativos de software sejam testados, configurados e fornecidos em um estado pronto para funcionar no Red Hat Linux. Usando-se ferramentas de gerenciamento de pacotes simples, novos pacotes podem ser obtidos por download, instalados e executados sem a configuração tortuosa, às vezes exigida por outros pacotes.

Outra vantagem do gerenciamento de pacotes é a capacidade de atualização: é impossível atualizar versões do Red Hat sem a necessidade de reinstalar o Linux desde o início.

Você pode usar o Red Hat em PC Intel, Alpha digital e Sun SPARC.

3.4 O que há de novo no Red Hat 6.0

Instalação aprimorada - Reconhece melhor dispositivos PCI, novo software de particionamento de disco e capacidade de escolher quais serviços serão carregados automaticamente no momento da inicialização.

Novas e melhores ferramentas de administração – Inclui uma poderosa ferramenta de configuração gráfica Linux, o *LinuxConf*. Também inclui uma ferramenta para a configuração da versão gratuita de X-Windows, XFree86, chamada de *Xconfigurator*.

Introdução do Gnome – É um ambiente desktop para X-Windows projetado para tornar mais fácil o desenvolvimento de aplicativos e proporcionar um ambiente desktop mais consistente, de qualidade profissional para usuários Linux.

Desempenho – Melhor implementação de multiprocessamento simétrico (SMP) e introdução de RAID baseado em software (fornece mecanismo para combinar múltiplos discos para melhorar a confiabilidade e o desempenho).

4 - O PC mínimo para Linux

Um 386 com 4MB, porém não pode executar X-Windows e o número de programas que ela pode executar simultaneamente é limitado pela quantidade de RAM física, seu desempenho será lento na maioria dos aplicativos de missão crítica (servidor de Web). Esse é portanto mais adequado como terminal de acesso a outro servidor Linux ou Unix; ou um servidor de baixo desempenho para serviços como DNS (converte nomes host em endereço IP reais) ou um servidor de autenticação para uma pequena empresa.

4.1 *Acessórios ideais para um servidor em sua Intranet*

Uma placa SCSI - Ideal para um sistema multiusuário (Ex.: Servidor de arquivos, servidor Web ou servidor de aplicativos). Escolha uma placa com suporte a Ultra-DMA SCSI.

O ideal é utilizar HD's em separado para dividir o processamento de dados / sistema e software.

4.2 *Verificando a compatibilidade de seu hardware*

O hardware precisa ser suportado por drivers incluídos na distribuição de Linux do usuário ou por software acessório que forneça drivers para o hardware em questão.

4.3 *Registrando suas informações de hardware*

Placa de vídeo - Fabricante e modelo; chipset de vídeo; quantidade de memória; tipo de relógio na placa.

Placa de som – Fabricante e modelo; IRQ da placa , endereço de I/O da placa e endereço de DMA.

Monitores – Fabricante e modelo; resolução mais alta de monitor; intervalo de sincronismo horizontal e intervalo de sincronismo vertical.

Mouse – Fabricante e modelo; número de botões; protocolo do mouse e porta serial.

Unidades de disco rígido – Capacidade de armazenamento total do HD; número de cilindros, número de cabeças e número de setores por trilha.

Modems - Fabricante e modelo; velocidade do modem e porta serial.

5 - O que é um comando Linux?

No MS-DOS os comandos não poderiam ser criados pelos usuários, ou sejam, eram limitados e geralmente estáticos.

No mundo Unix e por extensão, no Linux, o conceito é diferente. Um comando é qualquer arquivo executável.

5.1 *Su*

É usado geralmente para alternar entre diferentes usuários dentro de um terminal virtual.

Exemplo de comando: `$ su user2` (será solicitada a senha do user2).

Quando acabarmos de trabalhar basta usar o comando **exit** para voltar ao usuário anterior.

Se você está logado como usuário e der o comando **su** sem nome de usuário, será solicitada a senha do Root e, quando ela for fornecida, será trocada para trabalhar como usuário-root.

Se você está logado como Root e der o comando **su** <o nome de algum usuário>, não será solicitado nenhum pedido de senha. Isso é interessante para o administrador, pois ele pode precisar se tornar diferentes usuários para depurar problemas, mas não necessariamente conhecer as senhas de outros usuários.

5.2 *pwd, Cd*

Esses comandos fornecem as ferramentas básicas de que você precisa para trabalhar com diretórios e arquivos.

O comando **Pwd** informa em qual diretório está atualmente.

O comando **Cd** muda seu diretório atual para qualquer diretório acessível no sistema.

5.3 *ls*

O comando **ls** é usado para ver o conteúdo do diretório corrente.

Entre as opções mais úteis temos:

- **a** – Inclui, na listagem, todos os arquivos contidos no diretório, mesmo as referências do diretório onde estamos posicionados e do diretório “pai”, ou seja, o superior àquele onde estamos posicionados – que são representados por “. “ (diretório atual) e “..” (diretório pai).
- **F** – Anexa aos nomes dos arquivos um caractere, indicando seu tipo: diretório (/), programas executáveis (*), links simbólicos (@), para FIFOs (|), para sockets (=) e nada para arquivos comuns.
- **l** – Uso de formato longo, detalhando os dados referentes a (siga os números no exemplo abaixo): (1) permissões, (2) quantidade de sub-diretórios ou se for 1 se trata de um arquivo, (3) nome do usuário que criou o arquivo e (4) do grupo a que este usuário pertence, (5) tamanho, (6) data da última alteração e (7) nome completo do arquivo. Veja o exemplo com os números indicando estas informações:

drwxr-xr-x	2	root	root	1024	Dec 23 15:22	bin
drwxr-xr-x	2	root	root	1024	Dec 31 05:48	boot
drwxr-xr-x	2	root	root	1024	Dec 6 15:51	cdrom
drwxr-xr-x	3	root	root	8192	Mar 11 10:17	dev
drwxrwxr-x	2	root	root	1024	Feb 27 13:52	dosa
-----(1)----	(2)	(3)	(4)	(5)	------(6)-----	(7)

- R – Listagem recursiva. Irá também acessar os arquivos que estão colocados internamente nos subdiretórios, a partir do ponto em que estamos.
- u – Usa a data do último acesso ao arquivo para a classificação da saída.
- X – Usa a extensão do nome de arquivo para a ordenação.
- L – Mostra entradas apontadas pelos links simbólicos.
- n - Mostra UIDs e GIDs numéricos em vez dos nomes
- S – Ordenar pelo tamanho do arquivo

5.4 *mkdir*

Cria usado para a criação de novos diretórios.

Sintaxe : `mkdir (diretório 1) (diretório 2) ... (diretório n)`

onde (diretório 1) até (diretório n) são os diretórios a serem criados. As entradas padrão em um diretório (por exemplo, os arquivos ".", para o próprio diretório, e ".." para o diretório pai) são criadas automaticamente. A criação de um diretório requer permissão de escrita no diretório pai.

O identificador de proprietário (owner id), e o identificador de grupo (group id) dos novos diretórios são configurados para os identificadores de proprietário e de grupo do usuário efetivo, respectivamente.

Opções:

- **m** (mode) - Esta opção permite aos usuários especificar o modo a ser usado para os novos diretórios.
- **p** - Com esta opção, `mkdir` cria o nome do diretório através da criação de todos os diretórios-pai não existentes primeiro.

Exemplo: `mkdir -p diretório 1/diretório 2/diretório 3`

cria a estrutura de subdiretórios "diretório 1/diretório 2/diretório 3".

5.5 *More e Less*

O comando **more** permite que o usuário se movam uma linha ou uma tela para frente por vez, em um longo corpo de texto, assim como pesquisar esse texto. Pressiona a barra de espaço faz pular para frente uma página, enquanto pressionar Enter moverá para frente uma linha por vez.

Para pesquisar para frente o arquivo inteiro, pressione a tecla de barra (/), seguida da palavra ou frase que você deseja pesquisar e, em seguida, pressione Enter. Você pode repetir o processo pressionando a tecla **n**, após a primeira busca, evitando a necessidade de digitar a mesma palavra ou frase repetidamente.

O comando **Less** é uma versão amplamente aprimorada do comando `more`. Além das funções básicas descritas anteriormente, a seguir estão algumas das outras ações que podem ser realizadas em um corpo de texto:

- Pular diretamente para uma linha – Coloque o número da linha seguido da letra **g**.
- Pular diretamente para o início ou final do arquivo - Se digitar **g** você pula para a primeira linha do texto. Com o **G** sozinho você pula para a última linha do texto.
- Retroceder em um arquivo – A seta para cima faz mover para cima uma linha de texto por vez e a seta para baixo faz mover uma linha de texto para baixo.
- Pesquisar retroativamente em um arquivo – Uma barra normal (/) seguida de uma palavra ou frase, pesquisará para frente do texto, e um ponto de interrogação

5.6 Find

O comando **Find** pode ser usado para pesquisar arquivos pelo nome, data de criação ou modificação, proprietário, tamanho do arquivo e até o tipo do arquivo.

A sua estrutura básica é:

```
$ find [diretório inicial] [parâmetros] [ações]
```

Diretório inicial especifica onde a pesquisa vai iniciar.

Os parâmetros representam o lugar em que você especifica os critérios de busca.

A seção referente às ações indica a ação que será executada nos arquivos encontrados. Geralmente, você desejará usar a ação **-print**, o que indica que o nome e o caminho completos do arquivo devem ser apresentados.

Também é possível pesquisar nomes de arquivo parciais. Por exemplo, se você sabe que o arquivo que está procurando começa com *fo*, então pode usar a expressão *fo** para indicar todos os arquivos que começam com *fo* e terminam com qualquer combinação. Quando você usa o caractere ***, é importante colocar apóstrofes em torno da expressão inteira.

```
Ex.: $ find / -name 'fo*' -print
```

5.7 Grep

Comando **Grep** é usado para verificar o conteúdo de um ou mais arquivos na tentativa de encontrar a ocorrência de um padrão de texto específico dentro dos arquivos.

Em geral, o padrão para o comando é:

```
$ grep <opções> [texto para pesquisa] [arquivos]
```

Se você quiser procurar uma frase, como “rio de janeiro”, precisará colocar o texto padrão entre aspas:

```
$ grep “rio de janeiro” *
```

Opções:

- **l** – Mostra os nomes dos arquivos que contêm o texto de busca.
- **c** – Informa o número de linhas num arquivo que atende à pesquisa feita.
- **i** – Não diferencia letras maiúsculas de minúsculas para o termo que está sendo pesquisado.

Considere a situação em que você deseja uma listagem de todos os arquivos do diretório corrente, com a data da modificação de 12 de maio. Você poderia encontrar essa informação usando pipe com `ls -l`, através de um comando `grep`:

```
$ ls -l | grep "may 12"
```

5.8 Tar

O programa **tar** era usado originalmente para criar backups de sistema em fitas. A criação de um arquivo **tar** é fácil:

```
$ tar cvf tar-nome-arquivo lista-arquivo
```

Esse comando criará um novo arquivo, especificado pelo nome de arquivo *tar-nome-arquivo* (geralmente tem extensão `.tar`), e depois armazenará todos os arquivos da lista nesse arquivo.

Cada uma dessas opções é usada para controlar diferentes aspectos do comportamento do comando `tar`. O **c** indica que estamos criando um arquivo, **v** indica que o comando deve ser executado no modo *verbose* (o que significa que cada nome de arquivo será apresentado, à medida que é copiado no arquivamento) e **f** significa que estamos gerando um arquivo (em oposição a uma unidade de fita).

comando *tar* copia todos os arquivos e subdiretórios de um diretório e um arquivo, caso o diretório faça parte da lista de arquivos. Assim, se temos um diretório chamado *vnc* e queremos que todo o conteúdo desse diretório seja copiado em um novo arquivo, chamado *vnc.tar*, podemos usar

```
$ tar cvf vnc.tar vnc
```

e obter o seguinte resultado:

```
$ tar cvf vnc.tar vnc
vnc/
vnc/license.txt
vnc/readme
vnc/classes/
vnc/classes/rfprre.class
```

Você notará que a primeira linha indica a criação do diretório *vnc* no arquivo e depois a cópia dos arquivos deste diretório do arquivo *.tar*.

Para ver o conteúdo de um arquivo *.tar* existente, substituímos a opção **c** por **t**.

```
$ tar tvf vnc.tar
```

Para extrair o conteúdo de um arquivo *.tar* no diretório corrente, substituímos o **c** ou **t** por **x**:

```
$ tar xvf vnc.tar
```

5.9 Gzip

Embora o comando *tar* seja útil para o armazenamento de arquivos, ele não realiza qualquer compactação nos exemplos anteriores. No Linux, a compactação é obtida com o comando **gzip**.

Ao contrário dos arquivos ZIP do Windows, que compacta muitos arquivos em um único arquivo compactado, o comando *gzip* compacta apenas arquivos individuais, sem compactá-lo em um arquivo.

Por exemplo, se temos um arquivo particularmente grande, chamado *test.pdf*, que não usaremos por algum tempo e queremos compactá-lo para economizar espaço em disco, usamos o comando *gzip*:

```
$ gzip test.pdf
```

Isso compactará o arquivo e incluirá a extensão *.gz* no final do nome de arquivo, mudando o nome para *test.pdf.gz*.

Para fazer a compressão máxima usamos a extensão *-9*.

```
$ gzip -9 test.pdf
```

Você pode listar o conteúdo do arquivo compactado usando a extensão *-l*.

```
$ gzip -l test.pdf
```

Para descompactar um arquivo *.gz*, retornando o arquivo ao seu estado descompactado original, com o nome *teste.pdf*.

```
$ gzip -d teste.pdf.gz
```

Um comando alternativo, *gunzip*, elimina a necessidade de usar a opção *-d*:

```
$ gunzip test.pdf.gz
```

5.9.1 Combinando *gzip* e *tar*

As versões recentes de *tar* fornecem um método para acessar diretamente e criar arquivos *tar* compactados com *gzip*.

Apenas incluindo uma opção **z** em qualquer um dos comandos *tar* discutidos anteriormente, podemos criar um arquivo compactado sem a necessidade de um segundo comando.

```
$ tar czvf vnc.tar.gz vnc (inclui todos os arquivos do diretório e subdiretórios de vnc compactando-os automaticamente)
```

```
$ tar tzvf vnc.tar.gz vnc (apresenta o conteúdo de nosso arquivo text.tar.gz compactado)
```

```
$ tar xzvf vnc.tar.gz vnc (extrai o conteúdo do arquivo).
```

5.10 cp

Para copiar um arquivo (ThisFile) do diretório corrente em um segundo arquivo (a ser chamado ThisFile-Acopy)

```
$ cp ThisFile ThisFile-Acopy
```

Se quisermos copiar ThisFile em /tmp, mas fornecer um nome diferente para o novo arquivo, podemos usar

```
$ cp ThisFile /tmp/NewFileName
```

Se você desse o comando “cp ThisFile NewFile” o conteúdo de NewFile seria sobrescrito por uma cópia de ThisFile e seria perdido para sempre.

Para evitar essa dificuldade, você pode usar o flag **-i** do comando cp, que obriga o sistema a confirmar quando qualquer arquivo for sobrescrito por uma cópia.

Você pode criar um **alias** para o comando cp executando o comando

```
$ alias cp='cp -i'
```

Podemos configurar nosso shell Bash usando o arquivo oculto **.bashrc** para garantir que, sempre que nos conectarmos, esse alias esteja definido. Para isso devemos editar esse arquivo com qualquer editor de texto (Ex.: mcedit) e incluir o alias dentro dele.

Para criar um alias para todos os usuários vá ao diretório /etc/rc.d e crie um arquivo qualquer definindo todos os alias que desejar. Ao dar boot na máquina todos os arquivos que estão abaixo desse diretório são executados, incluindo seu arquivo de alias que acaba de criar.

Podemos passar vários argumentos para o comando e o último deles será tratado como o destino e todos os arquivos precedentes serão copiados no destino.

```
$ cp FileOne FileTwo FileThree /tmp
```

Ao copiar vários arquivos desse modo, é importante lembrar-se de que o último argumento deve ser um diretório.

Se quisermos copiar um diretório inteiro e todos os seus subdiretórios, podemos usar o flag **-R** do comando cp.

```
$ cp -R SomeDir /tmp
```

Esse comando copia a totalidade do subdiretório SomeDir para o diretório /tmp criando o diretório /tmp/Somedir.

5.10.1 Cópia avançada

Quando você copia um arquivo, o arquivo resultante normalmente pertence a quem copiou, e não a quem criou o arquivo.

Analogamente, quando um arquivo é criado em um diretório, ele possui um conjunto de permissões padrão atribuídas a ele. Ao copiar um arquivo, a cópia terá as permissões definidas de acordo com o padrão do diretório de destino, em vez de manter as permissões do arquivo original. Para mantermos os atributos originais usamos o flag **-p**

```
$ cp -p /tmp/TheFile .
```

Normalmente, quando você copia um vínculo simbólico, o arquivo resultante é uma cópia do arquivo apontado pelo vínculo para o mesmo arquivo.

```
Lrwxrwxrwx  1 user2  users  2 Aps 5 13:10  TheFile - > OtherFile
```

Então a execução do comando cp

```
$cp /tmp/TheFile ~/NewFile ( o ~ significa que você irá fazer a cópia debaixo do diretório home do usuário atual)
```

resultaria em um arquivo que seria uma cópia de OtherFile.

Mas, e se quiséssemos copiar o vínculo, em vez do próprio arquivo? Para isso o comando cp tem um flag para tratar dessa situação: o flag **-d**, que indica a não-eliminação da referência ao vínculo simbólico. Poderíamos simplesmente usar o comando

```
$ cp -d /tmp/TheFile ~/NewFile
```

Dito isso, é hora de reunir tudo. *E se quisermos usar o comando cp para criar uma cópia de backup útil de um diretório existente e todos os seus subdiretórios?*

```
$ cp -pdR TheDirectory /backups (cria uma cópia exata de TheDirectory no diretório /backups/TheDirectory)
```

Porém o comando cp fornece um modo simplificado para obter isso: o flag **-a**.

```
$ cp -a TheDirectory /backups
```

5.10.2 Evitando erros

- ◆ Você pode usar o flag **-b** para criar uma cópia de backup de qualquer arquivo que vá ser sobrescrito. Por padrão, o backup será o nome de arquivo original com um til (~) depois dele.
- ◆ É possível alterar o modo como o comando cp atribui nomes aos arquivos de backup, usando dois flags diferentes: **-S** e **-V**. O flag **-s** permite que você mude o caractere de til usado em nomes de backup para outra coisa.

```
$ cp -b -S_ FileOne FileTwo
```

- ◆ O flag **-V** proporciona ainda mais flexibilidade, permitindo que o usuário especifique um dos três tipos de esquemas de atribuição de nomes de backup:
 - ❖ **t** ou **numbered** : cria backups numerados em seqüência. Se um arquivo de backup numerado já existir, então o novo arquivo de backup será numerado seqüencialmente, após o arquivo de backup existente; os nomes de arquivo resultantes são como os seguintes: FileName.~Number~ (Ex.:FileName.~2~)

```
$ cp -b -V t FileOne FileTwo
```

- ❖ **Nil** ou **existing**: se um arquivo de backup numerado já existe, então cria um arquivo de backup numerado; caso contrário, cria um arquivo de backup simples normal.
- ❖ **Never** ou **simples**: cria um arquivo de backup simples usando o til padrão ou um caractere alternativo, indicado pelo flag `-s`.

5.11 *rm*

Este comando é utilizado para apagar arquivos. É importante lembrar que quando os arquivos são apagados, no sistema Unix, é impossível recuperá-los.

Sintaxe: `rm (arquivo 1) (arquivo 2) ... (arquivo n)`

onde (arquivo 1) até (arquivo n) são os arquivos a serem apagados. Se um arquivo não possuir permissão de escrita e a saída-padrão for um terminal, todo o conjunto de permissões do arquivo será exibido, seguido por um ponto de interrogação. É um pedido de confirmação. Se a resposta começar com "y" ("yes" = sim), o arquivo será apagado, caso contrário ele será mantido no sistema.

Quando você apaga um arquivo com o comando "rm", você está apagando somente um link (ligação ou entrada) para um arquivo. Um arquivo somente será apagado verdadeiramente do sistema quando ele não possuir mais nenhuma ligação para ele, isto é, nenhum link referenciando-o. Geralmente, arquivos possuem somente um link, portanto o uso do comando "rm" irá apagar o(s) arquivo(s). No entanto, se um arquivo possuir muitos links, o uso de "rm" irá apagar somente uma ligação; neste caso, para apagar o arquivo, é necessário que você apague todos os links para este arquivo.

Você pode verificar o número de links que um arquivo possui utilizando o comando `ls`, com a opção `-l`.

Opções:

- **f** - Remove todos os arquivos (mesmo se estiverem com proteção de escrita) em um diretório sem pedir confirmação do usuário.
- **i** - Esta opção pedirá uma confirmação do usuário antes de apagar o(s) arquivo(s) especificado(s).
- **r** - Opção recursiva para remover um diretório e todo o seu conteúdo, incluindo quaisquer subdiretórios e seus arquivos.

➔ **CUIDADO** : diretórios e seus conteúdos removidos com o comando "rm -r" não podem ser recuperados.

5.11.1 *Excluindo diretórios inteiros*

Você pode remover o diretório inteiro usando o flag `-r`

```
$ rm -r tempInstall
```

Quando você está certo de que deseja excluir um diretório inteiro, vai querer usar o flag `-f` do comando `rm`.


```
$ rm -rf tempInstall
```

5.12 rmdir

é utilizado para apaga diretórios vazios.

```
Sintaxe: rmdir (diretório 1) (diretório 2) ... (diretório n)
```

onde (diretório 1) até (diretório n) são os diretórios a serem apagados. O comando "rmdir" se recusa a apagar um diretório inexistente, exibindo a mensagem:

```
rmdir : (nome-do-diretório) : No such file or directory
```

Quando usar "rmdir", lembre-se que o seu diretório de trabalho corrente não pode estar contido no(s) diretório(s) a ser(em) apagado(s). Se você tentar remover seu próprio diretório corrente, será exibida a seguinte mensagem:

```
rmdir : . : Operation not permitted
```

Se o diretório o qual você deseja remover não estiver vazio, utilize o comando "cd" para acessar os arquivos dentro do diretório, e então remova estes arquivos utilizando o comando "rm".

Opções:

-p Permite aos usuários remover o diretório e seu diretório pai, o qual se torna vazio. Uma mensagem será exibida na saída padrão informando se o caminho ("path") inteiro foi removido ou se parte do caminho persiste por algum motivo.

➔ CUIDADO : diretórios removidos com o comando "rmdir" não podem ser recuperados!

5.13 mv

Vamos começar considerando a operação básica de movimentação:

```
$ mv FileOne /tmp
```

É possível mover o arquivo para o diretório /tmp e mudar o seu nome usando o seguinte comando:

```
$ mv FileOne /tmp/NewFileName
```

Usando esse conceito, você pode renomear um arquivo. Basta mover um arquivo de seu nome existente para um novo nome no mesmo diretório:

```
$ mv FileOne NewFileName
```

Ao copiar arquivos, é possível mover vários deles de uma vez, pois o comando mv pode aceitar mais de dois argumentos e o último argumento servirá como diretório de destino da movimentação.

```
$ mv *.bak *.tmp *.old /tmp
```

É possível mover diretórios inteiros com o comando mv, sem usar nenhum flag especial. Se houvesse um subdiretório chamado TheDir no diretório atual e quiséssemos movê-lo de modo que ele se tornasse um subdiretório sob /tmp, usaríamos o comando mv exatamente como fizemos para arquivos:

```
$ mv /TheDir /tmp
```

Opções:

- **b** - Fará uma cópia de segurança de arquivos que serão sobrepostos pela movimentação, caso já existam arquivos com aqueles nomes no volume de destino.
- **u** - Atualiza apenas os arquivos que tiverem data de atualização anterior ao que está sendo movido sobre outro, já existente. Assim sendo, apenas os mais novos irão substituir as versões mais antigas.

5.14 cat

Oficialmente usado para concatenar arquivos. Também usado para exibir todo o conteúdo de um arquivo de uma só vez, sem pausa.

Sintaxe: `cat < arquivo1 > < arquivo2 >... < arquivo n >`,

onde (arquivo1) até (arquivo n) são os arquivos a serem mostrados. "cat" lê cada arquivo em seqüência e exibe-o na saída padrão. Deste modo , a linha de comando:

```
cat < arquivo >
```

exibirá o arquivo em seu terminal; e a linha de comando :

```
cat < arquivo1 > < arquivo2 > > < arquivo3 >
```

concatenará "arquivo1" e "arquivo2", e escreverá o resultado no "arquivo 3" . O símbolo ">", usado para redirecionar a saída para um arquivo, tem caráter destrutivo; em outras palavras, o comando acima escreverá por cima do conteúdo de < arquivo3 >. Se, ao invés disto, você redirecionar com o símbolo ">>", a saída será adicionada a <arquivo3 >, ao invés de escrever por cima de seu conteúdo.

5.15 chgrp

Modifica o grupo de um arquivo ou diretório.

Sintaxe: `chgrp [-f] [-h] [-R] gid nome-do-arquivo`

"chgrp" modifica o identificador de grupo ("group ID" , gid) dos arquivos passados como argumentos. "gid" pode ser um número decimal especificando o group id, ou um nome de grupo encontrado no arquivo "/etc/group". Você deve ser o proprietário do arquivo, ou o superusuário, para que possa utilizar este comando.

Opções:

- f Esta opção não reporta erros
- h Se o arquivo for um link simbólico, esta opção modifica o grupo do link simbólico. Sem esta opção, o grupo do arquivo referenciado pelo link simbólico é modificado.
- R Esta opção é recursiva. "chgrp" percorre o diretório e os subdiretórios, modificando o GID à medida em que prossegue.

5.16 *chmod*

Modifica as permissões de um arquivo ou diretório. Você deve ser o proprietário de um arquivo ou diretório, ou ter acesso ao root, para modificar as suas permissões.

Sintaxe : `chmod permissões nome_do_arquivo`

onde :

permissões - indica as permissões a serem modificadas;

nome - indica o nome do arquivo ou diretório cujas permissões serão afetadas.

As permissões podem ser especificadas de várias maneiras. Aqui está uma das formas mais simples de realizarmos esta operação :

1- Use uma ou mais letras indicando os usuários envolvidos: . u (para o usuário) . g (para o grupo) . o (para "outros") . a (para todas as categorias acima)

2- Indique se as permissões serão adicionadas (+) ou removidas (-).

3- Use uma ou mais letras indicando as permissões envolvidas : . r (para "read") (ler) . w (para "write") (escrever) . x (para "execute") (executar)

Exemplo : No exemplo a seguir, a permissão de escrita ("write") é adicionada ao diretório "dir1" para usuários pertencentes ao mesmo grupo. (Portanto, o argumento "permissões" é g+w e o argumento "nome" é dir1).

```
$ ls -l dir1
drwxr-xr-x 3 dir1 1024 Feb 10 11:15 dir1
$ chmod g+w dir1
$ ls -l dir1
drwxrwxr-x 3 dir1 1024 Feb 10 11:17 dir1
```

Como você pôde verificar, o hífen (-) no conjunto de caracteres para grupo foi modificado para "w" como resultado deste comando.

Quando você cria um novo arquivo ou diretório, o sistema associa permissões automaticamente. Geralmente, a configuração "default" (assumida) para os novos arquivos é:

```
- r w - r - - r - -
```

e para novos diretórios é:

```
d r w x r - x r - x
```

5.17 *chown*

Modifica o proprietário de um arquivo ou diretório.

Sintaxe: `chown [-fhR] (proprietário) (nome-do-arquivo)`

O argumento "proprietário" especifica o novo proprietário do arquivo. Este argumento deve ser ou um número decimal especificando o `userid` do usuário ou um "login name" encontrado no arquivo `/etc/passwd`.

Somente o proprietário do arquivo (ou o superusuário) pode modificar o proprietário deste arquivo.

Opções:

- **f** Esta opção não reporta erros.
- **h** Se o arquivo for um link simbólico, esta opção modifica o proprietário do link simbólico. Sem esta opção, o proprietário do arquivo referenciado pelo link simbólico é modificado.
- **r** Esta opção é recursiva. "chown" percorre o diretório e os subdiretórios, modificando as propriedades à medida em que prossegue.

5.18 *du*

Exibe o espaço ocupado de um diretório e de todos os seus subdiretórios, em blocos de 512 bytes; isto é, unidades de 512 bytes ou caracteres.

"du" mostra a utilização do disco em cada subdiretório.

5.19 *date*

Exibe a data configurada no sistema.

O comando "date", a nível de usuário, exibe na tela a data configurada no sistema. Ele pode ser usado com opções que mostram a data local ou data universal GMT - Greenwich Mean Time. A configuração dos dados deste comando só podem se realizadas pelo superusuário.

Para exibir a data local, basta executar "date". Caso queira a data GMT utilize a opção "-u".

```
$date  
Wed Jan 8 12:05:57 EDT 1997
```

Aqui a data é exibida em 6 campos que representam o dia da semana abreviado, o mês do ano abreviado, o dia do mês, a hora disposta em horas/minutos/segundos, a zona horária e o ano.

Podemos acertar a hora, usando o comando na seguinte forma:

```
$date -s 09:30 (formato hora:minuto)
```

ou

```
$date -s 09/18 (formato mm/dd)
```

5.20 *file*

Exibe o tipo de um arquivo.

Alguns arquivos, tais como arquivos binários e executáveis, não podem ser visualizados na tela. O comando "file" pode ser útil se você não tem certeza sobre o tipo do arquivo. O uso do comando permitirá a visualização do tipo do arquivo.

```
Exemplo : $file copyfile
copyfile: ascii text
```

5.21 *init* / *telinit*

É o pai dos processos. O seu papel principal é criar os processos a partir de programas armazenados no arquivo **/etc/inittab**. Este arquivo tem entradas que fazem com que o **init** inicie **gettys** em cada linha que os usuários podem usar para acessar o sistema. Ele controla ainda processos autônomos requeridos por qualquer sistema em particular.

5.21.1 *Níveis de execução*

É uma configuração de software do sistema que permite que um grupo selecionado de processos sejam inicializados. Os processos acionados por **init** para cada um dos níveis de execução são definidos no arquivo **/etc/inittab**. **Init** pode estar em um dos oito níveis de execução: **0-6** e **S** ou **s**.

5.21.2 *Iniciando*

Após o **init** ser iniciado com o último passo da sequência de inicialização, ele procura pelo arquivo **/etc/inittab** e verifica se há alguma entrada para o tipo **initdefault**. A entrada **initdefault** define o nível de execução inicial do sistema. Caso não haja tal entrada, um nível de execução deve ser informado na console do sistema.

5.21.3 *Telinit*

/sbin/telinit é um link simbólico de **/sbin/init**. Ele recebe um argumento de um caracter e sinaliza ao **init** para executar a ação apropriada. Os seguintes argumentos servem como diretivas para **telinit**:

0,1,2,3,4,5 ou **6** - Dizem ao **init** para mudar o nível de execução.

a,b,c - Dizem ao **init** para processar somente aquelas entradas no arquivo **/etc/inittab** que tenham os níveis de execução a,b ou c.

Q ou **q** - Dizem ao **init** para reexaminar o arquivo **/etc/inittab**.

S ou **s** - Dizem ao **init** para entrar em modo monousuário.

5.22 *runlevel*

Encontra o nível de execução anterior e o atual do sistema.

```
# runlevel [utmp]
```

O comando **runlevel** lê o arquivo **utmp** do sistema (normalmente **/var/run/utmp**) para localizar o registro do nível de execução, mostrando o nível de execução anterior e o atual na

saída padrão, separado por um espaço simples. Se não existir um nível de execução anterior, a letra **N** será impressa em seu lugar.

5.23 type

Mostra a localização de um arquivo. type
Este comando mostra a localização de um arquivo, através do caminho do sistema.

```
[marisa@guarani log]$ type bash
bash is /bin/bash
```

5.24 diff

Compara dois arquivos em formato texto linha a linha.

O comando diff procura encontrar o menor conjunto de diferenças entre as linhas dos arquivos, listando as que devem ser mudadas no primeiro arquivo para torná-lo idêntico ao segundo.

Exemplo:

```
[marisa@guarani log]$ diff linguagens linguagens.old
2c2
< java - ainda vai ser boa um dia
---
> java - ainda vai ser uma boa linguagem um dia
```

5.25 zip

Programa de compactação de arquivos.

O zip é um comando de compressão e empacotamento de arquivos. Ele é análogo à combinação dos comandos tar e compress e é compatível com o pkzip e winzip das plataformas DOS/Windows.

Exemplo:

```
[marisa@guarani log]$ l previsao
-rw-rw-r-- 1 marisa marisa 3274 jul 27 11:37 previsao
[marisa@guarani log]$ zip previsao previsao
adding: previsao (deflated 59%)
[marisa@guarani log]$ l previsao*
-rw-rw-r-- 1 marisa marisa 3274 jul 27 11:37 previsao
-rw-rw-r-- 1 marisa marisa 1497 jul 27 11:44 previsao.zip
```

5.26 sort

Ordena as linhas de arquivos texto.

O comando sort ordena as linhas de um arquivo texto. Existem diversas opções de ordenamento: ascendente, descendente, por campo do arquivo, etc.

Exemplo:

```
[marisa@guarani log]$ ls -l alunos
-rw-rw-r-- 1 marisa marisa 3274 jul 27 11:37 alunos
```

```
[marisa@guarani log]$ cat alunos
linus
alan
bill
eric
```

```
[marisa@guarani log]$ sort alunos
alan
bill
eric
linus
```

```
[marisa@guarani log]$ sort -r alunos
linus
eric
bill
alan
```

5.27 cut

Seleciona campos de uma tabela. cut

A entrada padrão é tratada como uma tabela. O comando seleciona colunas da tabela para serem removidas ou copiadas na saída padrão.

Exemplo:

```
[marisa@guarani log]$ cat linguagens
C - o assembler do passado
Java – ainda vai ser boa um dia
Perl - existe mais de um jeito de fazer isso
Php - pré processador html

[marisa@guarani log]$ cut -c1-5 linguagens
C - o
Java
Perl
Php

[marisa@guarani log]$ cut -d '-' -f 1 linguagens
C
Java
Perl
Php
```

```
[marisa@guarani log]$ cut -d '-' -f 2 linguagens
o assembler do passado
ainda vai ser boa um dia
existe mais de um jeito de fazer isso
pré processador html
```

5.28 tr

Converte ou remove caracteres. tr

Este comando copia da entrada padrão para a saída padrão substituindo ou removendo os caracteres selecionados. Qualquer caractere de entrada encontrado em expr1 é convertido para o caractere da posição correspondente em expr2.

Exemplo:

```
[marisa@guarani log]$ tr a-z A-Z < linguagens
C - O ASSEMBLER DO PASSADO
JAVA - AINDA VAI SER BOA UM DIA
PERL - EXISTE MAIS DE UM JEITO DE FAZER ISSO
PHP - PRÉ PROCESSADOR HTML
```

5.29 Outros comandos

comm: Compara dois arquivos para determinar quais linhas são comuns entre eles.

mail: Usado para receber ou enviar e-mail.

tset: Escolher o tipo de terminal.

umask: Permite que o usuário especifique uma nova criação de camuflagem.

uniq: Compara dois arquivos. Procura e exibe em linhas o que é incomparável em um arquivo.

wc: Exibe detalhes no tamanho do arquivo.

write: Usado para mandar mensagens para outro usuário.

5.30 Criando vínculos simbólicos

→ Os vínculos simbólicos (que são apenas ponteiros para um arquivo real em outra posição) são usados normalmente por administradores de sistema e projetistas de aplicativos.

→ Usa-se o comando **ln** com o flag **-s** para indicar um vínculo simbólico.

```
$ ln -s /bin/cp MyCopy (cria um vínculo chamado "MyCopy" para acessar virtualmente o diretório "/bin/cp").
```


6 - Processadores de textos sob Linux

6.1 PICO

O editor de textos **pico** é de domínio público. Foi elaborado para ser simples, amigável ao usuário, com layout similar ao programa de correio eletrônico *PINE*.

Os comandos de edição e de movimentação de cursor (além das teclas de setas) são obtidos no **pico** através da digitação de seqüências " tecla control + letra ". A designação "^" é utilizado para indicar a tecla control, de forma que a seqüência control q, por exemplo, é representada por " ^Q ", indicando que as teclas control e q devem ser pressionadas simultaneamente.

Utilizamos como convenção o carácter " % " para indicar o prompt da sua máquina. Para obter mais informações sobre o **pico**, digite :

% man pico

,ou solicite o help interativo quando estiver utilizando o editor, acionado sempre com "**^G**"

6.1.1 Acionando o Pico

Digite **pico** no prompt da sua conta:

%pico

Imediatamente aparecerá a tela de trabalho do editor. A linha de status no alto da tela mostra versão do programa que você está usando, o nome do arquivo que está sendo editado e indica se este arquivo foi ou não alterado. A terceira linha a partir do rodapé da tela é utilizada para mensagens de informação ou para comandos adicionais. As duas linhas no rodapé possuem os principais comandos do **pico**.

Se você digitar **pico** seguido de um nome, da seguinte forma:

% pico catálogo

, será criado um novo arquivo com o nome informado.

Cada caracter digitado aparece automaticamente no arquivo editado, exatamente na posição em que se encontra o cursor.

Para editar um arquivo já existente, digite **pico** seguido do nome do arquivo:

%pico teste

e o arquivo aparecerá automaticamente na tela de edição.

6.1.2 Comandos de movimentação básica do cursor

O cursor pode ser movimentado utilizando-se das teclas de setas.

^C - informa a posição do cursor ao ser digitado o comando. Informa a linha em que o cursor se encontra, o total de linhas e o caracter em que se encontra.

^F - move o cursor para o próximo caracter

^B - move o cursor para o caracter anterior

^P - move o cursor para a linha imediatamente superior

- ^N** - move o cursor para a linha imediatamente inferior
- ^E** - move o cursor para o final na linha em que se encontra
- ^A** - move o cursor para o início da linha em que se encontra

6.1.3 Comandos de movimentação da tela

- ^V** - Prev Pg - move o cursor para a tela seguinte (abaixo)
- ^Y** - Next Pg - move o cursor para a tela anterior (acima)

6.1.4 Comandos de deleção

São os comandos que permitem eliminar palavras ou trechos do arquivo.

- ^D** - apaga o caracter sob o cursor
- ^K** - Del Line - apaga toda a linha em que o cursor se encontra

A tecla de **backspace** apaga o caracter imediatamente anterior ao cursor

6.1.5 Errou? Como cancelar uma operação

O **pico** possui os seguintes comandos para cancelamento de operações:

- ^U** - Underline- volta a inserir as linhas anteriormente apagadas
- ^C** - Cancel - cancela a operação. Este comando não está disponível o tempo inteiro.

Ele pode ser usado para cancelar as seguintes operações:

- inserção de um arquivo: **^R**
- sair do editor: **^X**

6.1.6 Outros comandos de edição

- ^I** - Insere uma tabulação aonde se encontra o cursor.
- ^O** - WriteOut - escrevendo por cima . Este comando grava o arquivo em edição com novo nome. Ao ativar o comando, o programa pedirá o novo nome do arquivo.
- ^J** - Justify - justifica o texto digitado, eliminando quebras de linha, linhas em branco e o alinhamento normal a esquerda.

6.1.7 Busca/substituição

^W - Este comando realiza a busca de uma letra, palavras ou seqüências de palavras. Você digita o comando e em seguida ele solicita que informe a seqüência para busca.

6.1.8 Inserindo um arquivo no meio de um texto.

Para inserir um arquivo no meio de um texto que estiver sendo editado, utilize os seguintes procedimentos:

^R - Read File - Inserindo um arquivo no meio do texto. Este comando permite que você insira um arquivo já existente em sua conta, no corpo do texto que estiver editando. O arquivo inserido aparecerá a partir da posição em que se encontrar o cursor.

Ao digitar o comando **^R** , vão aparecer as seguintes opções:

Insert file:

^G - Get Help **^C** - Cancel **^T** - To files

Caso você já saiba o nome do arquivo, basta digita-lo ao lado da linha

Insert file:

Com os novos comandos:

^G - Get Help - você obtém uma explicação sobre o comando **^R**

^C - to cancel - cancela a operação

^T - oferece a lista dos arquivos e diretórios que existem no seu diretório corrente.

Para caminhar nesta lista, utilize as teclas com seta e para selecionar um arquivo ou diretório, pressione a tecla enter ou return.

6.1.9 Refresh

^L - comando para "limpar" a tela. É utilizado por exemplo para retirar do meio do seu texto uma chamada de *talk* ou mensagem recebida através do comando *write*.

6.1.10 Salvar

Para salvar um texto durante a edição, sem sair do editor, digite o comando **^O** e informe ao sistema o nome do arquivo em que deve ser gravado o texto digitado.

6.1.11 Salvar e sair/ Sair sem salvar

^X - saindo do **pico**

Ao ativar este comando o editor vai perguntar se você deseja salvar ou não o arquivo. Caso você diga que não, a edição é abandonada. Caso diga que sim, ele solicita que você dê um nome ao arquivo.

Oferece também a opção de help sobre o comando **X**, obtido com **G** e a opção de cancelar a operação (**^C**) ou de listar arquivos (**^T**).

7 - Entendendo o Shell

O **Shell (concha)** como o próprio nome indica serve de ponto de contato entre o utilizador e o sistema. É o ambiente de linha de comandos – um interpretador de comandos semelhante ao prompt do DOS.

Ao contrário do prompt do DOS que é um ambiente fixo com flexibilidade limitada, os shells do Unix são pequenos programas aplicativos, executados como processos quando você se conecta, que fornece uma variedade de características de interface de linha de comandos e recursos de acordo com diferentes usuários e aplicativos.

De um lado temos o utilizador que não entende o funcionamento e a complexidade inerente ao coração do sistema operativo(kernel) , e que não consegue estabelecer qualquer tipo de comunicação diretamente com este. Do outro, temos o sistema operativo que não entende a linguagem humana, e que se gere por um conjunto de regras e definições complexas. Entre os dois existe o **shell**.

Ela recebe as instruções do utilizador e passa-as para o sistema. Por exemplo, quando se executa um simples "cat", a shell lança um processo filho que irá disparar um conjunto de bibliotecas de sistema que por sua vez irão reservar recursos, mapear e proteger zonas de memória (entre outras operações complexas), no entanto, para o utilizador a única consequência visível será o de ver o conteúdo de um ficheiro na sua console.

No mundo Unix e, por extensão, no Linux existem numerosos shells para se escolher. Cada shell oferece um conjunto de características e recursos diferentes, e a maioria oferece sua própria linguagem de script (programa de execução automática).

As principais funções de uma shell são:

- ◆ *Interpretação de comandos* : quer em modo interativo, quer como interpretador de **shell script**
- ◆ *Controle de processos* : Gestão de "jobs".
- ◆ *Mecanismo de memorização de comandos*: Guarda os últimos comandos dados pelo utilizador.
- ◆ *Correção de Erros*: Correção ortográfica de erros.
- ◆ *Completa/Lista comandos/ficheiros*: Com a tecla TAB permite "adivinhar" a partir da letra inicial os possíveis comandos/ficheiros (Ótimo quando surgem os diretórios com nome extensos, típicas em Unix)

7.1 Comparando Shells

Existem duas classes principais de shell - aqueles que derivam sua sintaxe básica e design do Bourne Shell (Bash) e aqueles cujo modelo tem por base o C Shell.

Bourne Shell (sh) - Sob diversos aspectos, ele é muito limitado, carecendo de recursos como uma lista de histórico e edição de linha de comandos. Ele é capaz de testar programas quanto ao status de sucesso e falha ao terminarem a execução, o que possibilitou a existência de scripts sofisticados.

C Shell (csh) - O C-shell13.3 é ao mesmo tempo interpretador de comandos e linguagem de programação (baseada em C), tem variáveis shell e variáveis de usuário.

Oferece funções específicas como por exemplo:

- Função HISTORY: Permite que o usuário repita e manipule os comandos que constam da lista.
- Função ALIAS: Possibilita a criação de uma grande variedade de comandos simples.

Bourne Again Shell (bash) - É o shell mais comum instalado com as distribuições Linux. Tem por base o Bourne Shell, mas fornece um conjunto de recursos mais amplo, incluindo edição de linha de comandos, uma lista de histórico e término de nome de arquivo.

Korn Shell (ksh) - O Korn Shell foi provavelmente o primeiro a introduzir muitos dos recursos populares que agora vemos no bash, incluindo a edição de linha de comandos.

7.2 Experimentando diferentes shells

Você pode experimentar os shells sem torná-los padrão apenas executando-os como um programa dentro de seu shell padrão. Os shells se encontram no diretório /bin. Para voltar ao shell anterior basta digitar **exit**.

Para mudar o shell padrão que é executado quando você se conecta em seu sistema, é preciso mudar sua entrada no arquivo de senhas do Unix. Você pode fazer isso usando o comando **chsh**.

```
$ chsh -s /bin/tcsh someuser (muda o shell de someuser para /bin/tcsh)
```

7.3 Visão geral do Shell Bash

Quando você se conecta no prompt *login:*, várias coisas acontecem. A primeira delas é a ativação de seu Shell (nesse caso, o Bash), seguida da execução de todo arquivo de configuração que você possa ter criado para seu ambiente Bash pessoal.

Para fornecer uma configuração personalizada no Bash, você precisa criar um arquivo chamado **.bashrc** em seu diretório de base. Trata-se de um simples arquivo de texto, que é executado pelo Bash quando você ativa o Shell - geralmente, quando você se conecta.

É no arquivo **.bashrc** que você pode configurar o comportamento do Bash, definir variáveis de ambiente, como o seu path, e ativar todos os programas que quiser, sempre que executar o Shell Bash.

7.4 Definindo variáveis de ambiente no Bash

Cada Shell possui sua própria sintaxe para a definição de variáveis de ambiente. No Bash, geralmente isso é feito em duas etapas:

1. Definir o valor de uma variável.
2. Exportar a variável para o ambiente.

Se quisermos designar o emacs como nosso editor de textos padrão, poderemos definir a variável de ambiente EDITOR com o comando

```
$ EDITOR=emacs  
e exportá-la com  
$ export EDITOR
```

As duas etapas podem ser combinadas em uma só, onde atribuímos um valor para EDITOR e exportamos EDITOR em um único comando:

```
$ export EDITOR=emacs
```

O valor das variáveis de ambiente podem ser acessados incluindo-se o símbolo \$ no início do nome da variável em um comando Bash. Desse modo, podemos incluir informações no valor atual de uma variável de ambiente. Por exemplo, se a variável PATH contém atualmente

```
/bin:/usr/bin:/usr/X11R6/bin
```

podemos incluir /usr/local/bin no path, usando o comando

```
$ export PATH=$PATH:/usr/local/bin
```

7.5 Shell Script

O shell possibilita a interpretação tanto de comandos digitados quanto de shell script, que não é nada mais do que um arquivo texto com seqüências de comandos e com permissão de execução. Para criar o shell script, crie um arquivo de texto e ajuste suas permissões para que ele se torne executável. Este arquivo pode ser criado com um editor como o vi ou simplesmente o redirecionamento para um arquivo da saída de um comando cat. Nota: para tornar um arquivo texto em um shell script deve-se usar o comando chmod para mudar suas permissões de acesso.

7.6 Redirecionamento de entrada e saída

Normalmente os programas não-interativos recebem seus dados de entrada através da entrada padrão – normalmente o teclado. Analogamente, eles apresentam seus resultados na saída padrão – normalmente, a tela. O usuário pode, através do redirecionamento de E/S, redefinir de onde um comando ou programa receberá sua entrada e para onde enviará sua saída. A entrada de um comando são os dados sobre os quais o comando irá operar. Estes dados podem vir de um arquivo especificado pelo usuário, de um arquivo de sistema, do terminal ou da saída de outro comando. A saída de um comando é o resultado da operação que ele realiza sobre a entrada. A saída dos comandos pode ser impressa na tela do terminal, enviada a um arquivo, ou servir de entrada para outro comando.

Exemplos de Saídas Padrão:

```
[marisa@guarani marisa]$ ls -l /bin > /tmp/arquivos_bin
```

```
[marisa@guarani marisa]$ l /tmp/arquivos_bin  
-rw-rw-r-- 1 marisa marisa 5469 jul 29 10:05 /tmp/arquivos_bin
```

```
[marisa@guarani marisa]$ echo "teste de saída padrão"  
teste de saída padrão
```

```
[marisa@guarani marisa]$ echo "teste de saída padrão" > /tmp/saída
```

```
[marisa@guarani marisa]$ cat /tmp/saída  
teste de saída padrão
```

Exemplos de Entrada Padrão:

```
[marisa@guarani marisa]$ sort < /etc/fstab
/dev/fd0          /mnt/floppy  ext2  noauto    0 0
/dev/hda1        swap         swap  defaults  0 0
/dev/hda5        /            ext2  defaults  1 1
/dev/hda6        /usr        ext2  defaults  1 2
/dev/hda7        /home       ext2  defaults  1 2
none             /proc       proc  defaults  0 0
```

7.6.1 Dutos ou Pipes (|)

Utilizado como conexão de utilitários. É uma maneira de redirecionar as entradas e saídas, de modo que a saída de um comando torna-se a entrada do comando seguinte. Pode-se usar vários dutos em uma mesma linha de comando, de maneira que é possível combinar tantos comandos quantos forem necessários.

```
$ ls -l | more
```

7.7 Término de nome de arquivo

Se você digitar caracteres suficientes para identificar exclusivamente um arquivo, comando ou nome de diretório, o bash poderá completar o restante do nome

```
$ /usr/lo
```

Simplemente pressionando-se a tecla Tab, o bash tentará completar o nome para você, neste caso preenchendo o texto para indicar /usr/local.

Em algumas ocasiões você pode ter mais de uma referência e o bash ficar confuso ao completar o nome. Nesse caso você deve teclar o Tab duas vezes. Será apresentado todas as alternativas possíveis. Você tem a liberdade de digitar caracteres suficientes para identificar exclusivamente o nome desejado e pressionar Tab novamente.

➔ O Bash consegue completar nomes de arquivo ou diretório sem fornecer um PATH completo

```
$ gr
```

O Bash procurará no PATH, nomes que combinem com os caracteres introduzidos. O Bash apresentará a você uma lista de alternativas possíveis:

```
grep  groff  grotty
```

Se você quiser o comando *grep*, basta digitar *e e*, em seguida, pressionar Tab novamente, e o Bash completará o comando automaticamente.

7.8 Alias de comando

Permite definir seus próprios comandos personalizados.

```
$ alias psa= "ps -aux | more"
```

Se você quiser criar alias de comando permanentes, deve defini-los em seu arquivo `.bashrc`

Além de oferecer um modo de criar atalhos para os comandos mais usados, o alias de comando pode ser usado para protegê-lo de erros importantes.

```
alias cp='cp -i'  
alias mv='mv -i'  
alias rm='rm -i'
```

7.9 Edição de linha de comandos

Permite que você use as teclas de seta para se mover na linha de comando atual, exclua e insira os caracteres necessários e pressione Enter para executar o comando sem mover o cursor para o final da linha.

Por padrão, o Bash tem a inserção ativada; portanto, se você digitar novos caracteres, eles serão inseridos no cursor, em vez de sobrescrever os caracteres existentes.

O Bash fornece vários atalhos de teclado úteis para acelerar a edição, especialmente com linhas de comando longas.

Atalho de teclado	Ação
Ctrl + A	Pula para o início da linha
Ctrl + E	Pula para o final da linha
Esc, B	Retrocede uma palavra
Esc, F	Avança uma palavra
Ctrl + L	Limpa a tela e apresenta o comando corrente como a primeira linha da tela
Ctrl + T	Transpõe o caractere à esquerda do cursor com o caractere de sua posição atual
Esc, T	Transpõe o palavra à esquerda do cursor com o palavra de sua posição atual
Esc, U	Transforma a palavra atual em maiúsculas
Esc, L	Transforma a palavra atual em minúsculas
Ctrl + K	Exclui da posição atual do cursor até o final da linha.

7.10 Lista de histórico de comandos

Todo comando que você executa é incluído em um buffer de lista de histórico, que pode ser acessado na ordem inversa a partir do comando mais recente executado, terminando com o último.

Assim como na edição de linha de comandos, existem alguns atalhos de tecla avançados para executar funções mais elaboradas com a lista de histórico de comandos.

Atalho de teclado	Ação
Ctrl + P	Move para o comando anterior na lista de histórico
Ctrl + N	Move para o próximo comando na lista de histórico
Esc, <	Pula para o início da lista de histórico (o comando menos recente)
Esc, >	Pula para o final da lista de histórico (o comando mais recente)
Ctrl + R	Pesquisa reversa na lista de histórico

O mais interessante deles é **Ctrl + R**, que permite pesquisar para trás na lista de histórico de maneira dinâmica e interativa: quando você começa a digitar um comando, o Bash mostra o comando mais recente que combina com o que foi digitado até o momento.

(reverse-i-search) “:

A medida que você digita as letras do comando específico, o Bash localiza o comando mais recente que corresponda à string introduzida. Demonstrado o comando que você deseja executar apenas teclando Enter.

7.11 Controle de tarefa

Usando o controle de tarefas é possível utilizar um único shell para executar e controlar vários programas que estejam em execução simultânea.

Normalmente, quando você executa um comando, ele atua em primeiro plano. Ou seja, o shell executa o comando e o prompt não retorna até que o comando esteja concluído. No caso de programas interativos, como o emacs, isso significa que o programa assume o controle da tela ou janela em que o shell está funcionando e somente quando você sai do programa é que o prompt de comando se torna novamente disponível. No caso de programas não-interativos, como o **find**, o programa será executado e, mesmo que não apresente nada na tela, o prompt de comandos não retorna até o programa termine.

```
$ find / -name '*.tmp' -print >templist
```

É aí que entra a idéia de execução de um programa em segundo plano. O modo mais fácil de colocar uma tarefa em segundo plano é incluindo um E comercial (&) no final do comando, quando você executá-lo.

```
find / -name '*.tmp' -print >templist &
```

Usando o comando **jobs** do Bash é possível controlar as tarefas que estão em execução em segundo plano.

```
[1] – Running      find / -name '*.tmp' -print >templist &
[2] + Running      ls -lR / >dirlist &
```

Se você já iniciou um programa em primeiro plano e quer colocá-lo em segundo plano, pode fazer isso. O atalho Ctrl + Z é interpretado pelo Bash como um pedido para suspender

temporariamente o processo atual. Podemos então colocar a tarefa em segundo plano com o comando **bg**

```
$ bg 1
```

onde 1 especifica o número da tarefa. Se houver apenas uma tarefa interrompida, não será necessário especificar o seu número. Quando uma tarefa interrompida for colocada em segundo plano, digitar *jobs* você verá ativar novamente (running).

Às vezes, é útil suspender temporariamente uma tarefa sem colocá-la em segundo plano. Isso é particularmente verdade quando você está usando um aplicativo interativo, como o emacs ou outro editor de textos, e quer executar um ou mais comandos e depois retornar à sua edição. Em vez de sair do editor, é mais fácil usar Ctrl + Z para interromper a tarefa do editor, executar seus comandos desejados e depois retornar a tarefa interrompida para o primeiro plano.

Para enviar uma tarefa interrompida para o primeiro plano, você pode usar o comando **fg**.

```
$ fg número da tarefa
```

Existem ocasiões em que você desejará finalizar ou eliminar uma tarefa interrompida ou de segundo plano. O comando **kill** pode ser usado para eliminar um comando de acordo com uma ID de processo (**PID**) ou número da tarefa.

```
$ kill %2 (é necessário o sinal de “%” para identificar a tarefa)
```

7.12 Expansão de padrão

Uma das vantagens do Bash é a capacidade de usar padrões poderosos para especificar um ou mais comandos ou arquivos.

Suponhamos que queiramos encontrar um arquivo específico e tudo que sabemos é que o nome de arquivo começa com a letra z. Então, o comando

```
$ ls -l z*
```

O que acontece aqui é que o Bash constrói uma lista de todos os nomes que começam com z e depois substitui z* por essa lista, efetivamente passando todos os nomes de arquivo como argumentos para o comando ls -l.

7.13 Expansão de nome de Path

Suponhamos que queiramos listar todos os arquivos cujos nomes tenham três letras de comprimento e comecem com a e terminem com z.

```
$ ls -l a?z
```

Suponhamos que queiramos listar todos os arquivos que começam com as letras **a**, **b**, **c** ou **d**.

```
$ ls -l [abcd]*
```

Entretanto, como **a**, **b**, **c** e **d** são uma seqüência contínua de letras, poderíamos usar

```
$ ls - [a-d]*
```

Considere os casos em que estamos produzindo um arquivo compactado de todos os diretórios de base em um sistema para propósitos de backup. Digamos que queremos produzir um arquivo de todos os diretórios de base, exceto aqueles que começam com a letra **m**.

```
$ tar czvf home.tar.gz /home/[a-l]* /home/[n-z]*
```

Entretanto, podemos tornar as coisas mais fáceis com

```
$ tar czvf home.tar.gz /home/[!m]*
```

7.14 Expansão de chave

Fornece um método pelo qual é possível expandir uma expressão independentemente de os nomes que estejam sendo gerados existirem realmente como arquivos ou diretórios.

```
$ mkdir testedir {1,2,3,4} (Cria os diretórios testedir 1, 2, 3 e 4)
```

É importante lembrar que expressão de chave deve conter pelo menos uma vírgula.

Um recurso interessante da expansão de nome de path e da expansão de chave é que elas podem ser usadas dentro de outra expressão de chave. Isso é possível porque a primeira expansão a ocorrer é a expansão de chave.

```
$ chmod 644 testefile.{tx?,Bak,0[0-9]}
```

Isso mudaria as permissões em uma série de arquivos, incluindo todo arquivo que combinasse com a expressão *testefile.tx?*, *testefile.Bak* e *testefile.00* a *testefile.09*.

7.15 Substituição de comandos

Com a substituição de comandos, a saída padrão de um comando se torna um argumento ou parâmetro para outro comando.

```
$ gzip $(find / -name '*.bak' -print)
```

Nesse caso compactaremos todos os arquivos que tenham a extensão *.bak*.

8 - Dicas sobre modo texto

8.1 Configurando seu PATH

Para ver os atuais diretórios que estão como PATH, digite o seguinte:

```
echo $PATH
```

Se o diretório desejado não estiver na lista, coloque-o assim:

```
PATH=$PATH:/diretório/a/ser/colocado/no/path
```

Isso colocará o /diretório/a/ser/colocado/no/path no PATH.

Obs.: Essas instruções são válidas somente para uma seção! Ou seja, são temporários. Se você quiser colocar um PATH permanente, coloque num profile pessoal. Se quiser ser um PATH GLOBAL, coloque o diretório no arquivo /etc/profile aonde indicado.

8.2 Qual a versão do Linux que estou a usar na minha máquina?

Para saber qual a versão do Linux existente na sua máquina, digite:

```
uname -a
```

8.3 Tudo sobre Linux para quem está migrando do DOS

Vamos agora aprender coisas simples:

- Como sair do Linux. Se você estiver no modo texto (terminal), é só digitar CTRL+ALT+DEL, se você estiver no X-Window, você terá primeiro que digitar CTRL+ALT+BACKSPACE, depois você digita CTRL+ALT+DEL. Nunca dê Reset na "tora", pois isso pode danificar seu sistema de arquivos, e algumas coisas você fez não vão ser salvas.
- O Linux tem uma coisa que o DOS não tem, permissões, acessos. Você está logado como um usuário normal, e de repente quer executar algum programa ou editar algum arquivo mas quando tenta, dá "Permisson Denied". Quer dizer o que você está tentando não é possível fazer por você como esse usuário. O usuário que pode fazer tudo, eu disse TUDO no sistema, é o root, ou seja, o administrador do sistema.
- Você agora está no prompt. Se o prompt terminar em \$ você estará como usuário normal, e quando estiver terminando em #, você está como root. Você agora quer obter ajuda, tente o bom e velho:

```
$ help
```

Este comando lhe dá ajuda sobre o bash (uma shell), se você quiser ajuda sobre um determinado comando, tente os manuais online:

\$ man comando

Isso invoca o manual do comando. Você pode tentar também:

\$ apropos comando

\$ whatis comando

e pressione 'q' para sair.

- Quando você vê a sintaxe do comando, você terá que saber que:

Na sintaxe do comando: \$ tar -tf < file.tar > [> redir_file]

o < ... > significa uma coisa essencial ao comando

o (...) significa uma coisa opcional

No exemplo acima, "file.tar" tem que ser identificado, e "> redir_file" é opcional.

8.3.1 Comparando os comandos

Veja a tabela a seguir:

DOS	Linux	Notas
BACKUP	tar -Mcvf device dir/	completamente diferente
CD dirname\	cd dirname/	quase a mesma sintaxe
COPY file1 file2	cp file1 file2	igual
DEL file	rm file	igual
DELTREE dirname	rm -R dirname/	igual
DIR	ls	não é exatamente a mesma sintaxe
EDIT file	vi file	eu acho que você não vai gostar
	emacs file	este é melhor
	jstar file	este é tipo o edit do DOS
FORMAT	fdformat, mount, umount	sintaxe um pouco diferente
HELP command	man command	a mesma filosofia
MD dirname	mkdir dirname/	quase a mesma sintaxe
MOVE file1 file2	mv file1 file2	igual
NUL	/dev/null	igual
PRINT file	lpr file	igual
PRN	/dev/lp0, /dev/lp1	igual
RD dirname	rmdir dirname/	quase a mesma sintaxe
REN file1 file2	mv file1 file2	não é pra arquivos múltiplos
RESTORE	tar -Mxpvf device	sintaxe diferente
TYPE file	less file	MUITO melhor
WIN	startx	poles apart!

8.3.2 Arquivos

A estrutura de arquivos do Linux é similar ao do DOS, são estocados em diretórios, alguns executáveis outros não...

Aqui vai alguns conceitos básicos:

- No DOS, os arquivos são de forma 8.3, ou seja, não podem passar de 8 dígitos.3 dígitos. Um exemplo: NOTENOUG.TXT. No Linux, se você instalou o Linux usando uma partição ext2 ou umsdos, você pode fazer melhor, pode colocar nomes de arquivos longos (no máximo 255 caracteres), um exemplo de arquivo que o Linux pode fazer e o DOS não pode:

Este.é.um.arquivo.MUITO_grande

- No DOS, os caracteres MAIÚSCULOS e minúsculos são tratados da mesma forma. No Linux, eles são completamente diferentes, exemplo: ARQUIVO.tar.gz e arquivo.tar.gz são dois arquivos diferentes, ls é um comando e LS é um erro.
- No Linux não existem extensões .EXE, .COM especial para programas como o DOS. Os programas executáveis no Linux são marcados com um asterisco no final do arquivo. Por exemplo:

```
$ ls -F
letter_to_Joe cindy.jpg cjpg* I_am_a_dir/ my_1st_script* old~
```

- Os arquivos cjpg* e my_1st_script* são executáveis. No DOS, arquivos de backup terminam com extensão .BAK, no Linux, eles terminam com um ~ (tio). No Linux, os arquivos que começam com um ponto são considerados ocultos. Por exemplo: o arquivo .eu.sou.um.arquivo.oculto não é mostrado com um comando ls normal;

8.3.3 Links Simbólicos

No Unix, existe um tipo de arquivo que não existe no DOS: O link simbólico. Ele pode funcionar como um redirecionador para um arquivo ou um diretório, e pode ser usado em arquivos ou diretórios também. É similar com os atalhos do Windows95. Exemplo de links simbólicos: /usr/X11, que redireciona para /usr/X11R6; /dev/modem, que redireciona para /dev/cua0 ou /dev/cua1

Para criar um link simbólico:

```
$ ln -s <file_or_dir> <linkname>
```

Exemplo:

```
$ ln -s /usr/doc/g77/DOC g77manual.txt
```

Agora você pode referir para g77manual.txt ao invés de /usr/doc/g77/DOC.

Operadores de Redireção e Direção: < > >> |

Wildcards: * ?

nul: /dev/null

prn, lpt1: /dev/lp0 or /dev/lp1; lpr

- EXAMPLES -

DOS	Linux
C:\HUGO>copy joe.txt joe.doc	\$ cp joe.txt joe.doc
C:\HUGO>copy *.* total	\$ cat * > total
C:\HUGO>copy fractals.doc prn	\$ lpr fractals.doc
C:\HUGO>del temp	\$ rm temp
C:\HUGO>del *.bak	\$ rm *~
C:\HUGO>move paper.txt tmp\	\$ mv paper.txt tmp/
C:\HUGO>ren paper.txt paper.asc	\$ mv paper.txt paper.asc
C:\HUGO>print letter.txt	\$ lpr letter.txt
C:\HUGO>type letter.txt	\$ more letter.txt
C:\HUGO>type letter.txt	\$ less letter.txt
C:\HUGO>type letter.txt > nul	\$ cat letter.txt > /dev/null
n/a	\$ more *.txt *.asc
n/a	\$ cat section*.txt less

Notas:

- * é melhor no Linux;
 - * mostra todos os arquivos exceto os ocultos;
 - .* mostra todos os arquivos ocultos; *.* mostra somente os que tiverem;
 - um "." (sem aspas) no meio, seguido de caracteres;
 - p*r mostra tudo que começa com p e termina com r;
 - *c* mostra todos os arquivos que tiverem um c no meio.
- Quando usado more, pressione SPACE para ler o arquivo, q ou CTRL-C para sair, less é melhor e deixa que você use as setas do teclado.
 - Não há UNDELETE, então pense duas vezes antes de apagar alguma coisa;
 - Adicionando aos < > >> do DOS, o Linux tem 2> para redirecionar mensagens de erro (stderr); 2>&1 redireciona stderr para stdout, enquanto 1>&2 redireciona stdout para stderr;
 - O Linux tem mais um wildcard L o [].
 - ❖ Use [abc]* mostra arquivos começando com a, b, c;
 - ❖ *[I-N,1,2,3] mostra arquivos terminando com I,J,K,L,M,N,1,2,3;
 - Não existe um DOS RENAME; para isso se utiliza mv *.xxx *.yyy;
 - Use cp -i e mv -i para ser avisado quando um arquivo está para ser sobrescrito.

8.3.4 Multi-tarefa

O Linux é um sistema multi-tarefa, por isso, ele pode ser acessado por vários consoles ao mesmo tempo, assim como pode ser rodado vários programas ao mesmo tempo. Para mudar o console do 1 a 6, utilize:

ALT+N (Onde N é o número do console)

Exemplo:

ALT+1, ALT+2, ALT+3, ALT+4, ALT+5, ALT+6

Agora você pode ir para o próximo console e o antecedente com:

ALT+RIGHT (Vai pra 1 console A FRENTE)

ALT+LEFT (Vai pra 1 console ATRÁS)

Se você quiser ir para outra sessão em sair do console, utilize o comando su:

su < usuário >

Exemplo:

su root

Para sair da sessão:

\$ exit

Cada programa executado, seja pelo boot ou a manualmente mesmo, fica identificado com um PID. Para visualizar estes PIDs, use o comando:

\$ ps -a

E para terminar esses processos (fechar o programa), use:

\$ kill < PID >

Quando algo é suspenso, ou seja, deixado temporariamente (a maioria dos programas são suspensos com CTRL+Z). Depois de suspenso, você pode retornar a eles através do comando:

fg < job >

Onda job é o programa que você quer retornar.

Para saber quais programas estão suspensos, tente o comando:

jobs

Para killar, ou seja, terminar algum programa suspenso:


```
kill < %job >
```

8.3.5 Diretórios

A estrutura de diretórios do Linux é similar ao do DOS, mas existem algumas diferenças entre o do DOS e o do Linux. Agora vou mostrar um exemplo de diferença:

```
DOS:    C:\DOCS\LINUX\LINUXMAN.TEX
Linux:  /home/hugo/docs/linuxmanual.tex
DIR:    ls, find, du
CD:     cd, pwd
MD:     mkdir
RD:     rmdir
DELTREE: rm -R
MOVE:   mv
```

- EXAMPLES -

DOS	Linux
C:\GUIDO>dir	\$ ls
C:\GUIDO>dir file.txt	\$ ls file.txt
C:\GUIDO>dir *.h *.c	\$ ls *.h *.c
C:\GUIDO>dir/p	\$ ls more
C:\GUIDO>dir/a	\$ ls -l
C:\GUIDO>dir *.tmp /s	\$ find / -name "*.tmp"
C:\GUIDO>cd	\$ pwd
n/a - veja nota	\$ cd
igual	\$ cd ~
igual	\$ cd ~/temp
C:\GUIDO>cd \other	\$ cd /other
C:\GUIDO>cd ..\temp\trash	\$ cd ../temp/trash
C:\GUIDO>md newprogs	\$ mkdir newprogs
C:\GUIDO>move prog ..	\$ mv prog ..
C:\GUIDO>md \progs\turbo	\$ mkdir /progs/turbo
C:\GUIDO>deltree temp\trash	\$ rm -R temp/trash
C:\GUIDO>rd newprogs	\$ rmdir newprogs
C:\GUIDO>rd \progs\turbo	\$ rmdir /progs/turbo

Notas:

1. Quando usando rmdir, o diretório para remover tem que estar vazio.

Para deletar o diretório com o que contêm dentro, use rm -R (em seu risco)

2. O caractere '~' é um atalho para o nome do seu diretório home. Os comandos cd ou cd ~ fazem você ir para seu home de onde você estiver. o comando cd ~/tmp leva você para /home/você/tmp.

8.4 Comandos do programa vi

Comandos do editor de textos vi do UNIX

MODOS DE TEXTO

Subcomandos de inserção de texto:

i	insere texto antes do cursor
r	insere texto no início da linha onde se encontra o cursor
a	insere texto depois do cursor
A	insere texto no fim da linha onde se encontra o cursor
o	adiciona linha abaixo da linha corrente
O	adiciona linha acima da linha corrente
Ctrl + h	apaga último carácter
Ctrl + w	apaga última palavra minúscula
Esc	passa para o modo comando

MODOS DE COMANDO:

Subcomandos para Movimentação pelo Texto:

Ctrl+f	passa para a tela seguinte.
Ctrl+b	passa para a tela anterior.
H	move o cursor para a primeira linha da tela.
M	move o cursor para o meio da tela.
L	move o cursor para a última linha da tela.
h	move cursor para carácter a esquerda.
j	move cursor para linha abaixo.
k	move o cursor para linha acima.
l	move cursor para carácter a direita.
w	move cursor para início da próxima palavra (Ignora pontuação).
W	move cursor para início da próxima palavra (Não ignora pontuação).
b	move cursor para início da palavra anterior (Ignora pontuação).
B	move cursor para início da palavra anterior (Não ignora pontuação).
0 (zero)	move cursor para início da linha corrente.
^	move cursor para o primeiro carácter não branco da linha.
\$	move cursor para o fim da linha corrente.
nG	move para a linha n.
G	move para a última linha do arquivo.

Subcomandos para Localização de Texto:

/palavra	procura pela palavra ou carácter acima ou abaixo do texto.
?palavra	move para a ocorrência anterior da palavra(para repetir a busca usar n).
n	repete o ultimo / ou ? comando.
N	repete o ultimo / ou ? comando na direção reversa.
Ctrl+g	mostra o nome do arquivo, o número da linha corrente e o total de linhas.

Subcomandos para Alteração de Texto:

x	deleta um carácter que esta sobre o cursor.
dw	deleta a palavra, do inicio da posição do cursor até o fim.
dd	deleta a linha inteira onde o cursor estiver.
D	deleta a linha a partir da posição do cursor em diante.

rx	substitui o caracter sob o cursor pelo especificado x (é opcional indicar o caracter).
Rtexto	substitui o texto corrente pelo texto indicado (opcional indicar o texto adicionado).
cw	substitui a palavra corrente. Pode-se inserir o novo conteúdo da palavra automaticamente.
cc	substitui a linha corrente. Pode-se inserir o novo conteúdo da linha automaticamente.
C	substitui restante da linha corrente. Pode-se inserir o texto logo após o comando.
u	desfaz a última modificação.
U	desfaz todas as modificações feitas na linha (se o cursor não mudou de linha).
J	une a linha corrente a próxima.

Subcomandos para Salvar o Texto:

:wq	salvar as mudanças feitas no arquivo e sai do editor.
:w < nome-arq >	salva o arquivo corrente com o nome especificado. Continua edição normalmente.
:w! < nome-arq >	salva (de modo forçado) o arquivo corrente no arquivo especificado
:q	sai do editor. Se mudanças não foram salvas é apresentada mensagem de advertência
:q!	sai do editor sem salvar as mudanças realizadas.

8.5 Como posso saber quantos hard links tem um arquivo e quantos ele pode ter

O número de hardlinks de um arquivo aparece no comando ls -l:

```
drwxr-sr-x 4 user group 1024 Feb 26 1997 xtar
-rwxr-xr-x 1 user group 942 Jun 30 1995 xterm.login
```

Aquele "4" logo depois das permissões é o número de links para o arquivo.

```
> ln: cannot link `/bin/ls' to `/home/user/bin/ls': Too many links
```

Este erro, em geral, é um erro no próprio comando que ao tentar resolver um "link" acha um "link" que aponta para ele mesmo.

8.6 Mudando o relógio de seu Linux

Para mudar o horário do relógio de seu linux digite:

```
date mmddhhmm[yy]
```

Onde...

```
mm = mes
dd = dia
hh = hora
```

mm = minuto
yy = ano

Então... Salve com o comando:

```
clock -w
```

Isso tudo tem que ser como root, claro.

8.7 Mudando o Editor de Texto padrão

Edite o arquivo /etc/profile e coloque as seguintes linhas:

```
export EDITOR=pico  
export VISUAL=pico
```

pico é o nome do programa. Você pode substituir pelo seu editor favorito (joe, jove, jed, etc)

8.8 Shell Scripts - Utilidades e mais Utilidades

8.8.1 Backup para um FTP

Este shell script faz com que se compacte os arquivos mais importantes do Linux e passe compactado para um ftp.

```
-----[CORTE AQUI]-----  
#!/bin/bash  
  
# Faz um TAR com os diretórios mais importantes do Linux  
  
tar cvf /tmp/backup.tar /home /var/spool/mail /var/spool/cron /etc /usr/local  
  
# Compacta o TAR  
  
gzip /tmp/backup.tar  
  
# Faz o ftp para ftp-host.domain.etc  
  
ftp -in <<EOF  
open ftp-host.domain.etc  
user usuário senha  
bin  
hash  
prompt  
cd /home/backup  
lcd /tmp  
put backup.tar.gz
```

bye

-----[CORTE AQUI]-----

8.8.2 Comandos do DOS no Linux

Este Script que fará com que seu ambiente Linux se pareça um pouco com o ambiente do DOS, perfeito para quem confundi comandos do dos no linux. Você deve colocar as linhas no /etc/bashrc ou /etc/profile (global) ou então no .profile do home do usuário.

-----[CORTE AQUI]-----

```
# /etc/bashrc
# -----
# Colocar este script no /etc/bashrc
# Este Script que fará' com que seu ambiente Linux se pareça um
# pouco com o ambiente do DOS, perfeito para quem confundi comandos
# do dos no linux.
# -----
# Begin!

# System wide functions and aliases
# Environment stuff goes in /etc/profile

export PS1='\u\$ $PWD> '      # Prompt (ex.fica 'user$ /etc>')
alias which="type -path"      # Onde esta' arquivo
alias ls="ls -F --color=tty"  # ls colorido
alias dir="ls -FlA --color=tty" # dir colorido
alias rd="rmdir"              # Remove diretório
alias md="mkdir"              # Cria diretório
alias cd..="cd .."            # cd junto com .. :)
alias copy="cp"                # Copiar arquivos
alias move="mv"                # Mover arquivos
alias ren="mv"                 # Renomear arquivos (mover)
alias rename="mv"              # Renomear arquivos (mover)
alias win="startx"             # Iniciar o x-windows
alias edit="pico"              # Um dos melhores editores (o joe é bom tb)
alias path="env | grep ^PATH"  # Mostra o path
alias home="cd ~"              # Vai pra o homedir do usuário
alias cdrom.on="mount /mnt/cdrom" # Monta o CD-ROM (declara no fstab)
alias cdrom.off="umount /mnt/cdrom" # Desmonta o CD-ROM
alias cdrom="cd /mnt/cdrom"    # Entre no CD-ROM
alias cls="clear"              # Apagar a tela (pra que? :/)
alias del="rm"                 # Remove arquivos
alias remove="rm"              # Remove arquivos
alias deltree="rm -r"          # Remove diretório cheio
alias xcopy="cp -R"            # Copia diretório + subdirectórios
alias config="setup"           # Setup :PP
alias diskon="mount /dev/fd0 /mnt/disk" # Monta disquete
alias diskoff="umount /dev/fd0" # Desmonta disquete

# End
```

-----[CORTE AQUI]-----

Ordena linhas de arquivos alfabeticamente

Ordena linhas de um arquivo em ordem alfabética

-----[CORTE AQUI]-----

```
#!/bin/sh
# Eitch
#-----
# Ordenar - Tkz to Levy, a friend of mine ;)
# Ordena linhas de um arquivo em ordem alfabetica
#
# Sintaxe: $0 <arquivo origem> <arquivo destino>
#-----
# Begin
echo Ordenando $1 no arquivo $2
cat $1 | sort > $1
# End
```

8.9 Como alterar a cor do fundo e da letra no console?

```
[usuario@localhost]$ setterm -background cyan -foreground black
```

As cores possíveis são: black, blue, green, cyan, red, magenta, yellow, white, default

Detalhes: o ls com cores limpa sempre o terminal, então as opções de cores do ls devem ser desabilitadas para que funcione, e caso se tenha um prompt colorido, a mesma coisa, volte-o para o padrão. Para tornar essa mudança definitiva, inclua essa linha no seu arquivo `/.bashrc`

8.10 Como mudo o idioma de meu Conectiva Linux?

Edite o arquivo `/etc/sysconfig/i18n` e altere as variáveis para a língua desejada.

As variáveis para língua portuguesa são:

```
LANG="pt_BR"
LC_ALL="pt_BR"
LC_CTYPE="ISO-8859-1"
LESSCHARSET="latin1"
```

Para inglês, apenas comente essas linhas:

```
# LANG="pt_BR"
# LC_ALL="pt_BR"
# LC_CTYPE="ISO-8859-1"
# LESSCHARSET="latin1"
```

Para língua espanhola:

```
LANG="es_ES"
LC_ALL="es_ES"
LC_CTYPE="ISO-8859-1"
LESSCHARSET="latin1"
```

Em outras distribuições/instalações, que não da Conectiva, coloque estas variáveis em `/etc/profile` ou equivalente e as exporte, assim:

```
[root@localhost]# export LANG LC_ALL LC_CTYPE LESSCHARSET
```

Existe suporte a outras línguas também, mas algumas delas tem muito poucas traduções, ou nenhuma... Você pode ver os códigos das línguas em /usr/share/locale numa máquina com Marumbi.

Marumbi

As configurações são as mesmas, mas o arquivo de configuração fica em /etc/sysconfig/lang

8.11 Como configuro a proteção de tela na console?

Execute o comando:

```
[usuario@localhost]$ setterm -blank "número"
```

Sendo "número" o número de minutos de inatividade para se esperar antes de rodar a proteção (tela preta).

0 - para desligar 60 - número máximo

8.12 Como agendar tarefas de sistema (/etc/crontab)?

Veja o /etc/crontab:

```
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Simplesmente coloque uma tarefa (um executável) para ser executada, dentro do diretório específico:

```
/etc/cron.hourly : de hora em hora
/etc/cron.daily : todo dia
/etc/cron.weekly : uma vez por semana
/etc/cron.monthly : uma vez por mês
```

Agora, se quiser algo mais específico, apenas adicione linhas ao /etc/crontab seguindo a mesma lógica.

Por exemplo, executar um programa de backup do sistema, passando para ele o parâmetro "geral", como superusuário (root), de segunda a sexta-feira, de março a novembro, às 3:44 da manhã:

```
44 3 * 3-11 1-5 root /root/backup geral
| | | | | | |
| | | | | | +--- comando a ser executado (com a rota)
| | | | | +----- usuário que executará o comando
| | | | +----- dia da semana
| | | +----- mês do ano
| | +----- dia do mês
| +----- hora
+----- minuto
```

Eventuais mensagens de erro são mandadas para a caixa postal do superusuário (root).

para maiores informações, limites, parâmetros e sintaxe:
man 5 crontab

8.13 Como executar um arquivo?

No Linux não existem extensões como forma de indicar se um arquivo é um programa executável. Verifique os atributos do arquivo com o comando `ls -la` e veja se o caractere "x" aparece. se não aparecer execute:

```
[root@localhost]# chmod +x nome-do-programa
```

Outro detalhe: por motivos de segurança o diretório corrente não faz parte do PATH. Para executar programas no diretório corrente basta:

```
[root@localhost]# ./nome-do-programa
```

Ou incluir o diretório corrente (.) no PATH, editando o arquivo `/etc/profile`

8.14 O que posso apagar para liberar espaço em disco?

A maioria dos programas já exclui seus próprios arquivos temporários.

Exceto o KDE, que deixa vários arquivos no `/tmp` com nomes `kio*` e `kfm*`, que podem ser excluídos sem problemas.

Pode-se excluir também o diretório `$HOME/.netscape/cache` que é o cache em disco do netscape.

E ainda resta o `/var/log` que é o diretório onde são guardados os arquivos de registro (log) do sistema, que crescem infinitamente e podem ser cortados/editados (ou apagados) de vez em quando. principais: `cron` `httpd/access_log` `lastlog` `maillog` `messages` `wtmp`

Quanto a esses arquivos de registro, pode-se usar o aplicativo `logrotate` para gerenciá-los automaticamente (dividi-los e apagá-los quando muito grande/antigo).

Pode-se excluir também os HOWTOs

```
[root@localhost]# rpm -e `rpm -qa | grep howto`
```

e numa atitude desesperada e não recomendada, excluir toda a documentação de todos os pacotes:

```
[root@localhost]# rm -rf /usr/doc/*
```

E é claro, por último mas o mais importante: Desinstale pacotes que não são utilizados. Muito dificilmente alguém utilizará TODOS os pacotes que estão na distribuição (dezenas de jogos, servidores, editores, etc), principalmente os maiores como `tetex-*`, `octave`, `emacs`...

8.15 Como usar o tar ?

Supondo o nome do arquivo "grande.tgz" e disquetes de 1,44 Mb

```
[root@localhost]# tar cvM -f /dev/fd0H1440 grande.tgz
```

```
|||| | | |
|||| | | |
|||| | | |
|||| | | |
|||| +-----+
arquivo/dispositivo | | +-----+ salvar no
                       | | +-----+
                       | | +-----+ volumes múltiplos
                       | | +-----+ detalhado
                       +-----+ criar
```

Para descompactar:

```
[root@localhost]# tar xvM -f /dev/fd0
```


8.16 Como faço para deixar o 'ls' sempre colorido?

Coloque em seu `/etc/bashrc` a linha `alias ls="ls --color"`

- Dentro das aspas, pode-se colocar qualquer outra opção do 'ls';
- Essa alteração só terá efeito na próxima console que você se logar;
- Ou, para incluí-las na console atual, digite: `[root@localhost]$. /etc/bashrc`

9 - Administração de sistema geral

O gerenciamento de usuários de qualquer sistema Linux é fundamental, desde um servidor de rede até um computador pessoal compartilhado. Isso vai desde a criação de novas contas de usuário até a mudança de senhas de usuário.

O acesso dos usuários aos recursos do sistema será governado de acordo com cada usuário e com cada grupo, onde um grupo consiste em vários usuários associados em uma entidade organizacional comum, com um nome único.

Outro item administrativo fundamental é a *automação de tarefas*, tanto na inicialização como nas tarefas programadas.

O Linux oferece sofisticados recursos de log, que tornam possível saber exatamente o que está ocorrendo em seu sistema.

9.1 Inicialização do sistema

As mensagens de inicialização são salvas no arquivo de log de sistema `/var/log/messages`.

O que acontece durante a inicialização

Existem dois estágios no processo de inicialização:

1. Inicialização do kernel. Durante essa fase, o Kernel é carregado na memória e imprime mensagens à medida que inicializa cada driver de dispositivo.
2. Execução do programa *init*. Esse programa manipula a ativação de todos os programas, incluindo daemons de sistema essenciais e outro software especificado para ser carregado no momento da inicialização.

O programa Init

Aqui você pode personalizar facilmente quais programas são carregados durante o ciclo de inicialização.

O Linux tem um sistema de *níveis de execução*. Um nível de execução é um número que identifica o estado atual do sistema e quais processos o *init* deve executar e manter em execução nesse estado do sistema. No arquivo `/etc/inittab`, a primeira entrada especifica o nível de execução padrão que é carregado durante a inicialização.

id:3:initdefault: (nível de execução 3)

Assim, para o nível de execução 3, as seguintes linhas são relevantes:

```
13:3:wait:/etc/rc.d/rc 3
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

As últimas seis linhas configuram os seis consoles virtuais fornecidos no Linux. A primeira linha executa o script de inicialização */etc/rc.d/rc 3*. Isso executará todos os scripts contidos no diretório */etc/rc.d/rc3.d*. Esses scripts representam programas que precisam ser iniciados na inicialização do sistema, como *sendmail*, serviços PCMCIA, o daemon de impressora e *crond*.

Daemons - São processos servidores, inicialmente inicializados durante o boot, que rodam continuamente enquanto o sistema estiver ativo, esperando, em background, até que um processo requisiute seus serviços. Por exemplo, network daemons em estado idle até que um processo requisiute serviços de rede.

O último script a ser executado será o *AS99local*. Esse script é, na verdade, um vínculo para o arquivo */etc/rc.d/rc.local*; é aqui que você pode colocar todos os programas de inicialização personalizada que quiser ativar no momento da inicialização.

9.2 Programando tarefas com crond

O *crond* é um daemon que é geralmente é instalado para iniciar sempre que o sistema for inicializado. Quando ele inicia é ativado a cada minuto e verifica se alguma tarefa foi programada para executar durante esse minuto.

Todas as tarefas programadas são armazenadas em um arquivo de configuração individual (conhecido como arquivo *crontab*) do usuário, com cada linha representando uma tarefa programada.

Cada entrada tem a forma

time-date command

A entrada *time-date* consiste em cinco campos numéricos, cada um separado por espaços, que indicam quando uma tarefa deve ser executada. Os cinco campos em ordem, são:

- **Minute:** De 0 a 59.
- **Hour:** De 0 a 23.
- **Day of month:** De 0 a 31.
- **Month:** De 0 a 12.
- **Day os week:** De 0 a 7, onde 0 e 7 representam domingo.

Para todos esses campos, várias regras proporcionam flexibilidade:

- Intervalos de números podem ser usados (Ex.: 1 – 3 no campo horas diz para programar o comando para 1:00, 2:00 e 3:00 da manhã. 2 – 4 no campo de dia da semana programa a tarefa para terça, quarta e quinta-feira.

O campo command

Às vezes, os comandos (como o *mail*) exigem que informações sejam introduzidas através da entrada padrão. Isso é feito usando-se sinais de porcentagem (%). O primeiro sinal de porcentagem marca o início da entrada padrão e cada sinal subsequente serve como um caractere de nova linha nessa entrada.

```
30 14 * * fri /bin/mail -s "TGIF" arna@lala.edu %Bom dia!!!!%% Tudo Bem???
```

Isso enviará a seguinte mensagem de correio eletrônico:

```
Bom bia!!!
```

```
Tudo bem???
```

Carregando entradas de um arquivo

Par carregar entradas de um arquivo, primeiro é necessário criar um arquivo que contenha todas as entradas que você queira que apareçam em seu arquivo *crontab*. Um exemplo de arquivo poderia conter duas entradas:

```
0 1 * * * /usr/local/bin/backup
30 14 * * fri /bin/mail -s "TGIF" arna@lala.edu %Bom dia!!!! %% Tudo Bem???
```

Esses arquivo precisa ser salvo com um nome conveniente, como *cronjobs*.

Uma vez criado e salvo o arquivo, ele pode ser carregado no arquivo *crontab* de um usuário, executando-se o comando:

```
$ crontab cronjobs
```

O conteúdo de *cronjobs* sobrescreverá todas as entradas correntes do arquivo *crontab* do usuário.

Usando o flag **-u**, o usuário-root pode especificar que o arquivo *crontab* de outro usuário, em vez do seu próprio, deve ser alterado.

```
$ crontab -u username cronjobs
```

Carregando arquivos crontab diretamente

Em vez de criar um arquivo separado e carregá-lo no arquivo *crontab*, o comando *crontab* fornece o flag **-e**, que permite ao usuário editar o arquivo *crontab* diretamente.

```
$ crontab -e
```

Assim como aconteceu quando carregamos um arquivo no arquivo *crontab*, usando o flag **-u**, o usuário-root pode editar diretamente o arquivo *crontab* de outro usuário:

```
$ crontab -u username -e
```

Vendo o conteúdo do arquivo crontab

Para ver o conteúdo do arquivo *crontab*, basta usar o flag **-l**

```
$ crontab -l
```

Assim como na edição do arquivo *crontab*, o usuário-root pode ver o conteúdo do arquivo *crontab* de qualquer usuário com o flag **-u**:

```
# crontab -u username -l
```

Removendo o arquivo crontab

Para apagar o conteúdo do arquivo *crontab* de um usuário, ele pode usar o flag **-r**:

```
# crontab -r
```

Analogamente, o usuário-root pode apagar o arquivo *crontab* de qualquer usuário com o flag **-u**:

10 - Dicas sobre o sistema

10.1 Onde ir buscar os HOWTOs e outra documentação?

Dê uma vista de olhos nestes lugares:

- ftp.funet.fi (128.214.6.100) : /pub/OS/Linux/doc/HOWTO
- tsx-11.mit.edu (18.172.1.2) : /pub/linux/docs/HOWTO
- sunsite.unc.edu (152.2.22.81) : /pub/Linux/docs/HOWTO

Para uma lista completa dos sites de FTP, veja - Onde ir buscar material sobre o Linux por FTP?

Se não tem acesso a FTP, tente usar servidores de FTP por mail em ftpmail@decwrl.dec.com, ftpmail@doc.ic.ac.uk ou ftp-mailer@informatik.tu-muenchen.de.

Uma lista completa dos HOWTOs está disponível no arquivo HOWTO.INDEX no diretório docs/HOWTO dos sites de FTP, ou na Web em <http://sunsite.unc.edu/mdw/HOWTO/HOWTO-INDEX.html>

Esta é uma lista (provavelmente incompleta) dos HOWTOs:

- Linux INFO-SHEET
- Linux META-FAQ
- Bootdisk HOWTO
- CDROM HOWTO
- DOSEMU HOWTO
- Distribution HOWTO
- Ethernet HOWTO
- Firewall HOWTO
- Hardware HOWTO
- Installation HOWTO
- Kernel HOWTO
- Electronic Mail HOWTO
- PCI-HOWTO
- Portuguese HOWTO
- PPP HOWTO
- Printing HOWTO
- Serial HOWTO
- Sound HOWTO
- XFree86 HOWTO

Outros documentos destes estão sempre em preparação. Se não encontrar a informação que precisa num destes documentos procure noutros diretórios próximos nos sites de FTP. Além destes HOWTOs ainda existem os mini HOWTOs, no diretório docs/HOWTO/mini.

O arquivo WRITING contém informações sobre como escrever um novo HOWTO.

Os HOWTOs são coordenados por Greg Hankins gregh@cc.gatech.edu. Os livros produzidos pelo Linux Documentation Project estão disponíveis em /pub/Linux/docs/LDP em

sunsite.unc.edu. Por favor leia-os se está a iniciar-se no UNIX e no Linux. Principalmente o Installation and Getting Started Guide. Os livros produzidos até ao momento são:

- The Linux Documentation Project manifesto
- Installation and Getting Started Guide
- The Kernel Hacker's Guide
- Network Administration Guide
- Linux System Administrator's Guide

10.2 Onde ir buscar material sobre o Linux por FTP?

Os três sites principais do Linux são:

- ftp.funet.fi (Finland, 128.214.6.100) : /pub/OS/Linux
- sunsite.unc.edu (US, 152.2.22.81) : /pub/Linux
- tsx-11.mit.edu (US, 18.172.1.2) : /pub/linux

O melhor lugar para atualizar as versões do kernel é ftp.cs.helsinki.fi em /pub/Software/Linux/Kernel;

Linus Torvalds disponibiliza as versões mais recentes do kernel nesse lugar. A distribuição Debian está disponível em ftp.debian.org e a distribuição Red Hat em ftp.redhat.com.

O conteúdo destes sites é "mirrorado" (copiado, em geral diariamente) por outros sites. Por favor use aquele que estiver mais perto de si -- será mais rápido para si e mais fácil para a rede.

- src.doc.ic.ac.uk : /packages/Linux (UK)
- sunacm.swan.ac.uk : /pub/Linux (UK)
- ftp.ibp.fr : /pub/linux (França)
- ftp.cc.gatech.edu : /pub/linux (EUA - sudeste: Suranet)
- wuarchive.wustl.edu : /systems/linux (EUA)
- uiarchive.cso.uiuc.edu : /pub/systems/linux (EUA)
- ftp.cdrom.com : /pub/linux (EUA)
- ftp.informatik.tu-muenchen.de : /pub/comp/os/linux (Alemanha)
- ftp.ibr.cs.tu-bs.de : /pub/os/linux (Alemanha)
- ftp.dfv.rwth-aachen.de : /pub/linux (Alemanha)
- ftp.informatik.rwth-aachen.de : /pub/Linux (Alemanha)
- bond.edu.au : /pub/OS/Linux (Austrália)
- ftp.cc.monash.edu.au : /pub/linux (Austrália)
- ftp.dstc.edu.au : /pub/Linux (Austrália: Queensland)
- ftp.sun.ac.za : /pub/linux (África do Sul)
- ftp.inf.utsm.cl : /pub/Linux (Chile)
- ftp.zel.fer.hr : /pub/Linux (Croácia)

Os pacotes de instalação poderão ser encontrados nos ftps:

ftp://ftp.ufsm.br/pub/linux/slackware (BR)

ftp://ftp.cdrom.com/pub/linux/ (US)

Aqui vai uma descrição dos pacotes a ser pegados:

A(*) - O Básico do sistema para rodar.

AP(*) - Aplicativos em geral

D - Linguagens de programação /GCC/G++/Perl/C/ e outros...

E - GNU Emacs 19.25.

F(*) - Coleção de FAQs e outros documentos.

I - Documentação de vários programas

N - Networking. TCP/IP, UUCP, mailx, dip, deliver, elm, pine, smail, cnews, nn, tin, trn.

(necessário para comunicação internet/rede em geral)

OOP - Programas Orientado a Objeto

K(*) - Kernel do linux (necessário para compilar do kernel, p/ atualização do hardware)

TCL - Tcl, Tk, TclX, blt, itcl.

Y - Games. The BSD games collection, and Tetris for terminals.

X - XFree86 2.1.1 system (X-Window tipo o Windows convencional)

XAP - Aplicativos para X : X11 ghostscript, libgr13, seyon, workman, xfilemanager, xv 3.01, GNU chess and xboard, xfm 1.2, ghostview, e varios X games.

XD - X11 program development. X11 libraries, server linkkit, PEX support.

XV - Xview 3.2 release 5. XView libraries, and the Open Look virtual and non-virtual window managers.

IV - Interviews libraries, include files, and the doc and idraw apps. These run unreasonably slow on my machine, but they might still be worth looking at.

OI - ParcPlace's Object Builder 2.0 and Object Interface Library 4.0, generously made available for Linux developers according to the terms in the "copying" notice found in these directories. Note that these only work with libc-4.4.4, but a new version may be released once gcc 2.5.9 is available.

T - The TeX and LaTeX2e text formatting systems.

10.3 Permissões

Para saber se um programa é executável ou não, execute um 'ls -l' e veja no lado esquerdo se o arquivo tem X nos seus argumentos, como no exemplo abaixo:

```
drwxr-xr-x 2 root root 1024 Dec 23 15:22 bin
drwxr-xr-x 2 root root 1024 Dec 31 05:48 boot
drwxr-xr-x 2 root root 1024 Dec 6 15:51 cdrom
drwxr-xr-x 3 root root 8192 Mar 11 10:17 dev
drwxrwxr-x 2 root root 1024 Feb 27 13:52 dosa
dr-xr-xr-x 11 root root 2048 Mar 11 10:19 etc
drwxr-xr-x 11 root root 2048 Feb 23 19:08 home
drwxr-xr-x 3 root root 1024 Feb 23 19:13 lib
drwxr-xr-x 2 root root 12288 Nov 2 11:25 lost+found
-rwxr--r-- 1 root root 57 Mar 10 03:44 make-backup
-rw-rw-r-- 1 killer users 2342 Mar 10 03:12 teste.txt
-rw-rw-rw- 1 fernando visits 23412 Mar 09 22:22 teste2.doc
```

No exemplo acima todos os arquivos tem como dono root e como grupo também root, com exceção do 'teste.txt' que o dono é 'killer' e o grupo é 'users', e também 'teste2.doc', no qual 'fernando' é o dono e o grupo 'visits' também é dono.

Como você pode ver do lado esquerdo de cada arquivo/diretório existe um série de letras r, w, x ou d! Vamos ver o que representa cada uma delas:

```
drwxrwxrwx
0111222333
```

No caso acima, a primeira coluna significa (número 0) se o nome listado é um diretório ou não, caso não seja um diretório ele será exibido da seguinte maneira:

```
-rwxr--r-- 1 root  root    57 Mar 10 03:44 make-backup
|
\-----> Não contém a letra 'd', não é diretório, e sim arquivo!!!
```

O exemplo abaixo mostra o que seria um diretório:

```
drwxr--r-- 1 root  root    1 Mar 10 01:12 bin
|
\-----> Contém a letra 'd' na primeira coluna, é um diretório!!!
```

Continuando, na segunda coluna (números 1 de acordo com o exemplo mais acima) temos as definições para o dono do arquivo, como mostra o exemplo:

```
-rwXr--r-- 1 killer  users   1231 Mar 09 12:12 teste.txt
| | |
| | \-----> O dono do arquivo (killer) pode executar o arquivo, x=executable!
| \-----> O dono do arquivo (killer) pode gravar no arquivo, w=writable!
\-----> O dono do arquivo (killer) pode ler o arquivo, r=readable!
```

Seguindo, na terceira coluna (composto pelos números 2) temos as definições para o grupo que é dono do arquivo, como mostra o exemplo:

```
-r--rwXr-- 1 fernando visits   212 Mar 01 12:42 exemplo.doc
| | |
| | \-----> O grupo dono do arquivo (visits) pode executar o arquivo!
| \-----> O grupo dono do arquivo (visits) pode gravar no arquivo!
\-----> O grupo dono do arquivo (visits) pode ler o arquivo!
```

Finalmente, temos a quarta coluna (composto pelos números 3), essa coluna se refere as permissões para todos os outros usuários do sistema, sem ser os donos e grupos-donos dos mesmos, exemplo:

```
-r--r--rwX 1 fernando visits    1231 Mar 03 12:42 exemplo2.doc
| | |
| | \--> Todos os usuários (exceto fernando e usuários do grupo visits)
| |     tem permissão para acessar o arquivo!
| | \---> Todos os usuários (exceto fernando e usuários do grupo visits)
| |     tem permissão para gravar no arquivo!
| | \----> Todos os usuários (exceto fernando e usuários do grupo visits)
| |     tem permissão para ler o arquivo!
```

Quando nos referimos a diretório invés de arquivos, o FLAG x (executável) diz se o diretório é ou não acessível, já que não podemos "EXECUTAR" diretórios... Exemplo:

```
drwxr--r-- 1 root    root    2134 Mar 01 12:54 exemplo3
| | | | | |
| | | | | | \----> Todos os usuários podem ler o interior do diretório, mas não
| | | | | |     podem usar o comando 'cd' para entrar nele, pois não existe
| | | | | |     o FLAG 'x' para a quarta coluna!
| | | | | | \-----> Usuários do grupo 'root' podem ler o interior do diretório,
| | | | | |     mas também não podem usar 'cd' para entrar no diretório!
| | | | | | \-----> O usuário 'root' pode usar 'cd' para entrar no diretório!
| | | | | | \-----> O usuário 'root' pode gravar arquivos nesse diretório!
| | | | | | \-----> O usuário 'root' pode ler o interior desse diretório!
| | | | | | \-----> Indica que o nome listado é um diretório!
```

O comando `chmod` pode ser usado para mudar os FLAGS 'rwx' dos arquivos e/ou diretórios, a sintaxe básica é:

```
chmod [ugoa]{-+}[rwx] nome_do_arquivo_ou_diretório
```

Exemplo:

```
chmod u+rw arquivo1.txt
```

No exemplo você mudará a permissão para o dono do arquivo (u = user) pode ler e gravar (rw) no 'arquivo1.txt'...

Caso você queira desfazer o comando, você faria:

```
chmod u-rw arquivo1.txt
```

Como se vê, o + ou - define se os FLAGS serão ativados ou desativados!

Outros exemplos:

```
chmod a+r arquivo2.txt (Todos usuários (a=all) podem ler o 'arquivo2.txt')
```

```
chmod o+w arquivo3.txt (Outros usuários (o=others) sem ser o dono e o grupo dono do arquivo, podem gravar o 'arquivo3.txt')
```

```
chmod g+x netscape (O grupo-dono do arquivo (g=group) pode executar o arquivo 'netscape')
```

O comando `chmod` pode também ser usado com números, em vez dos flags, como mostra o exemplo:

```
chmod 664 arquivo.txt
```

O que quer dizer cada um desses números? Veja abaixo:

- 0 = nenhuma permissão
- 1 = permissão para executar
- 2 = permissão para gravar
- 3 = permissão para gravar e executar
- 4 = permissão para ler
- 5 = permissão para ler e executar
- 6 = permissão para ler e gravar
- 7 = permissão para ler, gravar e executar

No exemplo o comando informou que o 'arquivo.txt' pode ser lido e gravado pelo seu dono (numero 6 na primeira coluna), informou que pode também ser lido e gravado pelos usuários que compõem o grupo-dono (numero 6 na segunda coluna), e informou que pode ser lido por todos os outros usuários do sistema (numero 4 na ultima coluna).

O comando `chown` é simples e pode ser usado da seguinte maneira:

```
chown usuário.grupo nome_do_arquivo_ou_diretório
```

Como exemplo, vamos definir que um arquivo 'teste4.txt' terá como dono 'killer' e como grupo 'users':

```
chown killer.users teste4.txt
```

Outros exemplos:

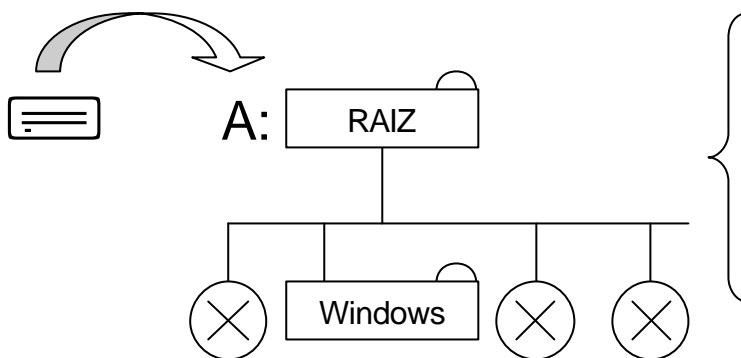
```
chown mrdvs.visits teste5.txt  
chown jackie.jackie teste6.txt
```

11 - CD-ROMs, Disquetes, Discos Rígidos e Sistemas de Arquivos - Uma Visão Geral

Um sistema de arquivos é composto por arquivos e diretórios, iniciando em um único diretório denominado raiz. Este diretório pode conter qualquer número de arquivos ou de diretórios, com cada diretório por sua vez seguindo o mesmo conceito e padrões. Um sistema de arquivos padrão normalmente se parece com uma árvore invertida, com os diretórios como galhos e os arquivos como folhas. Sistemas de arquivos residem em unidades de armazenamento de massa como disquetes, discos rígidos e CD-ROMs.

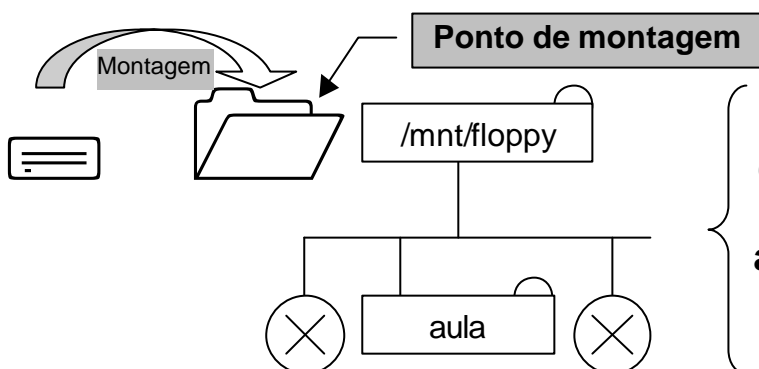
Por exemplo, uma unidade de disquetes no DOS ou Windows é normalmente referenciada como `A:`. Isso descreve o dispositivo (`A:`) e o diretório raiz do dispositivo. O disco rígido primário, em sistemas similares, é tipicamente referenciado como `C:` uma vez que a especificação de dispositivos para o primeiro disco rígido é `C:`. Para especificar o diretório raiz do dispositivo `C:`, pode-se utilizar `C:`.

Neste caso, teremos então dois sistemas de arquivos - um em `A:` e o outro em `C:`. Para especificar qualquer arquivo em um sistema de arquivos DOS/Windows, deve-se especificar o dispositivo no qual ele reside, ou ele deve residir no dispositivo padrão do sistema (o qual é a origem do indicador DOS de linha de comando - é o dispositivo padrão em um sistema com uma única unidade de disco rígido).



O Sistema de Arquivos contido em um dispositivo é acessado a partir de uma identificação de unidade lógica.

Sob Linux é possível definir sistemas de arquivos residentes em diferentes meios de armazenamento como se fossem um único e grande sistema de arquivos. Isso pode ser feito através da definição de um dispositivo dentro de um sistema de arquivos. Por exemplo, enquanto um sistema de arquivos de um diretório raiz de um disquete em DOS pode ser referenciado como `A:`, o mesmo dispositivo pode ser acessado no Linux com um diretório denominado, por exemplo como `/mnt/floppy`.



O Sistema de Arquivos contido em um dispositivo é acessado a partir de uma identificação de diretório.

O processo de mesclar sistemas de arquivos desta forma é conhecido como montagem. Quando um dispositivo está montado significa que ele pode ser acessado pelos usuários do sistema. O diretório através do qual o sistema de arquivos pode ser acessado é conhecido como ponto de montagem. No exemplo anterior, `/mnt/floppy` era o ponto de montagem do disquete. Note que não há restrições (além das convenções normais) de nome de pontos de montagem. Poderíamos facilmente denominar o ponto de montagem com `/longo/caminho/para/a/unidade/de/disquete` ou simplesmente `/A`. Um ponto a ser lembrado é que todos os diretórios e arquivos de um dispositivo têm a sua localização no sistema relacionada com o ponto de montagem.

Para montar um sistema de arquivos, esteja seguro de estar acessando o sistema como super-usuário ou de usar o comando `su` (`man su` - em português). Uma vez tendo os privilégios de super-usuário, execute o comando `mount` (`man mount` - em português) seguido pelo dispositivo e pelo ponto de montagem. Por exemplo, para montar a primeira unidade de disquete em `/mnt/floppy`, pode-se digitar o seguinte comando `mount /dev/fd0 /mnt/floppy`. Para acessar os dados em um disquete formatado em `ext2`, basta digitar `cd /mnt/floppy`. Na instalação o *Conectiva Linux* irá criar um arquivo chamado `/etc/fstab`. Este arquivo contém informações que permitem sintetizar os comandos de montagem de dispositivos. Usando-se as informações contidas naquele arquivo, pode-se comandar somente `mount` e então, ou o ponto de montagem ou o dispositivo. O comando `mount` irá então procurar o restante das informações em `/etc/fstab`. É possível modificar manualmente o arquivo ou utilizar-se o `Linuxconf` conforme descrito nas próximas seções.

11.1 Revisando o Sistema de Arquivos - Visão Geral

Inicialmente vamos verificar a estrutura de diretórios:

- Inicie o `Linuxconf`
- Informe a senha do super-usuário quando solicitado (caso já não seja o `root`).
- Abra `[Configurar] [Sistemas de Arquivos] [Acessar Dispositivos Locais]`.

Os campos são:

- Origem - o dispositivo físico: `hd` indica um disco rígido IDE, `fd` indica uma unidade de disquete, e `cdrom` indica uma unidade de CD-ROM. Caso o sistema possua um dispositivo SCSI, será apresentada então a indicação `sd`. Caso mais de um dispositivo do mesmo tipo esteja presente, estes serão ordenados por letras, como por exemplo `hda` representa o primeiro dispositivo IDE, enquanto `hdb` representa o segundo. Em alguns casos, serão apresentados números em lugar de letras; em unidades de discos rígidos eles representam as partições, enquanto que em unidades de disquetes, o número de referência da unidade atual.
- Ponto de Montagem - nome com o qual o dispositivo será mapeado dentro do sistema de arquivos.
- Tipo - indicação do tipo do sistema de arquivos. Uma partição Linux padrão usará um tipo `ext2`. Um sistema de arquivos DOS terá um tipo `vfat` com suporte a nomes longos, ou `fat` para suporte somente a nomes DOS tradicionais. O sistema de arquivos ISO 9660 indica um dispositivo CD-ROM.

Nota: o <i>Conectiva Linux</i> pode acessar sistemas de arquivos do tipo FAT32 utilizando o tipo <code>vfat</code> .
--

- Tamanho - apresenta o tamanho do sistema de arquivos em Mb. Para mídias removíveis como disquetes e CD-ROM é listado um tamanho de 0 Mb.
- Tipo Partição - uma descrição e um código do tipo sistema de arquivos usados naquela partição.
- Estado - condição atual do sistema de arquivos, ou seja se está disponível, portanto montado ou indisponível, conseqüentemente desmontado.

Sistemas de arquivos de outras máquinas na rede podem também estar disponíveis. Eles podem variar de um único diretório a um volume inteiro. Nenhuma informação sobre Tamanho ou Tipo da Partição estará disponível para essas partições. Informações adicionais desses sistemas de arquivos podem ser encontradas na opção [Configurar] [Sistemas de Arquivos] [Acessar volumes NFS].

A tela é similar a de Volume Local, com as seguintes diferenças:

- Origem - o nome da máquina em que está disponibilizado o sistema de arquivos, seguida pelo diretório remoto. Por exemplo: `cnc:/var/spool/mail` onde `cnc` é a máquina que contém o diretório `/var/spool/mail` que está sendo disponibilizado.
- Tipo - sempre igual a `nfs`.

11.2 Adicionado Pontos de Montagem NFS - Visão Geral

NFS é a sigla para *Sistemas de Arquivos Remotos*. É uma forma de computadores compartilharem partes de seus sistemas de arquivos através de uma rede. Estas partes podem ser um simples diretório até milhares de arquivos em uma vasta hierarquia de diretórios. Por exemplo, muitas empresas poderão ter um único servidor de correio eletrônico compartilhando os diretórios de mensagens com os usuários do sistema através de montagens NFS.

Para criar um ponto de montagem NFS:

- Inicie o Linuxconf
- Informe a senha do super-usuário quando solicitado (caso já não seja o `root`).
- Abra [Configurar] [Sistemas de Arquivos] [Acessar volumes NFS].
- Na tela Especificação de Volume, selecione Aceitar.

Os três campos são:

- Servidor - nome da máquina onde residem os sistemas de arquivos a serem montados. Por exemplo `guarani.cnc.com.br`.
- Volume - o nome do sistema de arquivos que se deseja adicionar. Por exemplo, `/var/spool/mail`.
- Ponto de Montagem - nome do caminho no qual o sistema remoto será montado. Por exemplo, `/mnt/mail`.

Isso é tudo o que se precisa saber para se ter um ponto de montagem criado. Linuxconf atualizará o arquivo `/etc/fstab` da forma adequada. Caso você tenha algum requisito adicional, podem ser acionadas as alternativas disponíveis na janela opções, a saber:

11.3 Opções gerais

As opções gerais não são necessárias na maioria das vezes. Elas proporcionam maior flexibilidade e segurança.

- Somente leitura - é possível bloquear a escrita em uma partição. Até mesmo o superusuário não poderá escrever ali. Esta opção é raramente usada numa partição de disco rígido normal.
- Montável pelo usuário - esta opção é geralmente usada em conjunto com a opção seguinte e é útil para mídias removíveis, como por exemplo disquetes. Ela permite a um usuário normal ativar a conexão a qualquer hora. Normalmente, apenas o superusuário (root) pode fazer uma montagem.
- Não montar na inicialização - especialmente útil para mídia removível, esta opção impede que o sistema tente fazer uma montagem na inicialização.
- Nenhum programa pode ser executado - esta é uma funcionalidade de segurança, especialmente útil para mídia removível. Se você definir a opção montável pelo usuário em uma mídia removível, qualquer usuário pode instalar arquivos especiais para dar-lhe acesso total ao seu sistema (privilégios de administrador). Esta opção lhe previne disto acontecer.
- Sem suporte a arquivos de dispositivos especiais - esta é uma opção relacionada à segurança. Dispositivos especiais são geralmente criados com direitos de acesso apropriados no diretório. Eles podem ser criados em outros lugares também através do comando `mknod`. Esta funcionalidade impede a montagem de uma mídia com dispositivo especial criada sem preocupações de segurança. Esses dispositivos poderiam arruinar toda a segurança do sistema.
- Sem permissão a programas com `setuid` - mais uma funcionalidade de segurança. É um meio termo entre o acesso total e a opção acima (nenhum programa pode ser executado). Selecionando-se esta opção, o sistema não deixará programas privilegiados usarem seus direitos especiais. Um programa privilegiado é aquele que muda seu usuário para outra identificação (geralmente para o superusuário) enquanto está sendo executado. Isto permite que o usuário normal execute tarefas especiais que apenas o superusuário poderia fazer.
- ativar quota por usuário - esta opção diz ao `kernel` para ativar a contabilidade de quota no sistema de arquivos. A contabilidade de quota é usada para cada usuário em tempo real, controlando o espaço em disco usado por ele e a quantidade de arquivos e diretórios que ele possui. Os limites podem ser aplicados para alguns ou todos os usuários. Há um controle separado para cada sistema de arquivos. O arquivo `quota.user` é criado na raiz do sistema de arquivos (o Configurador Linux o criará para você se esta opção for selecionada). O utilitário `quotacheck` é executado para inicializar o arquivo com o estado corrente do sistema de arquivos. Então o `kernel` o atualizará silenciosamente para cada conta de usuário. Isto é útil para impedir que um simples usuário ocupe todo o disco.
- ativar quota por grupo - é a mesma funcionalidade da "ativar quota por usuário", mas adequada para grupos. As quotas para grupos definem a soma das cotas de todos os membros do grupo. O arquivo `quota.group` é criado quando esta opção é selecionada e o utilitário `quotacheck` é usado para inicializar o arquivo. Enquanto um usuário pode estar sobre sua quota pessoal, a quota de seu grupo pode ultrapassar este limite. O usuário será advertido ao criar arquivos novos e/ou aumentá-los.

12 - Administração de sistema geral

O gerenciamento de usuários de qualquer sistema Linux é fundamental, desde um servidor de rede até um computador pessoal compartilhado. Isso vai desde a criação de novas contas de usuário até a mudança de senhas de usuário.

O acesso dos usuários aos recursos do sistema será governado de acordo com cada usuário e com cada grupo, onde um grupo consiste em vários usuários associados em uma entidade organizacional comum, com um nome único.

Outro item administrativo fundamental é a *automação de tarefas*, tanto na inicialização como nas tarefas programadas.

O Linux oferece sofisticados recursos de log, que tornam possível saber exatamente o que está ocorrendo em seu sistema.

12.1 Gerenciamento de usuários

Criação de usuários

É feita através do **useradd** ou **adduser**.

```
# adduser usuário1
```

Esse comando cria o usuário executando as seguintes ações:

- Criação de uma entrada para o usuário no arquivo */etc/passwd* sem uma senha.
- Atribuição de uma ID para o usuário. Geralmente, essa é a próxima ID de usuário disponível, em ordem numérica. No Red Hat, o padrão é usar o menor número maior do que 500, que seja maior do que as IDs de todos os usuários.
- Inclusão do usuário no grupo apropriado. No Red Hat, isso significa criar um grupo para o usuário, ao qual apenas esse usuário pertence. Em outras distribuições Linux, todos os usuários podem pertencer ao mesmo grupo por padrão.
- Criação de um diretório de base para o usuário (em */home/usuário1*, na maioria dos sistemas Linux) e cópia do conteúdo de */etc/skel* no diretório de base.

Para adicionar um usuário modificando seu ID usa-se o flag **-u**:

```
# adduser -u 10001 usuário1
```

e você quer impor a atribuição de um grupo em particular para o usuário use o flag **-g**:

```
# adduser -g users usuário1
```

Considere agora que o *usuário1* pertença aos grupos *group1* e *group2*, além do grupo padrão *users*. O comando *adduser* oferece o flag **-G**, que permite a especificação de grupos adicionais para que se inclua o novo usuário, quando a conta for criada:

```
# adduser -g users -G group1,group2 usuário1
```


Para especificar um diretório de base alternativo para um usuário, utilize o flag **-d**:

```
# adduser -d /other/home/directory usuário1
```

Mudando os padrões de Useradd

Existem alguns padrões usados por useradd que talvez você queira anular sempre que criar um usuário. Por exemplo, talvez queira que todos os diretórios de base sejam criados em */users*, em vez de */home*.

Esses padrões podem ser redefinidos usando-se o flag **-D** do comando useradd e vários flags suplementares. O flag **-D** indica que o comando não deve criar um novo usuário, mas sim atribuir novos padrões.

```
# useradd -D -b /users (Defini o path do diretório de base padrão como /users)
```

```
# useradd -D -g /users (Defini o grupo padrão de todos os novos usuários como /users)
```

Mudando senhas

A mudança de senhas é feita com o comando **passwd**. Qualquer usuário pode mudar sua senha simplesmente digitando o comando no prompt.

Em muitas versões de Linux, o comando passwd verifica se uma senha é curta demais, simples demais, semelhante demais ao nome de usuário ou semelhante demais à senha anterior.

O usuário-root tem o poder de mudar a senha de qualquer usuário fornecendo o nome de usuário como argumento para o programa *passwd*.

12.2 Configurando diretório de base padrão

Quando a conta do usuário é criada, seu diretório de base é criado e populado com um conjunto de arquivos padrão. Esse conjunto de arquivos padrão é copiado do diretório */etc/skel*, que contém o diretório de esqueleto para novos diretórios de base.

Para incluir um arquivo em cada novo diretório de base, basta criar o arquivo e colocá-lo em */etc/skel*, com o mesmo nome que você deseja que ele tenha nos diretórios de base dos usuários.

Todos os usuários incluídos depois que você colocar o arquivo lá encontrarão esse arquivo em seus diretórios de base quando suas contas forem criadas.

12.3 Removendo usuários

A exclusão de usuários é um processo paralelo à inclusão de usuários: você usa o comando *userdel*.

```
# userdel usuário1
```

O problema é que os arquivos do usuário não são excluídos. Para excluir o diretório de base do usuário simultaneamente, forneça o flag **-r** :

```
# userdel -r usuário1
```

E se o usuário possuir arquivos em outra parte do sistema que precisam ser excluídos? Isso pode ser feito usando-se o comando **find**, depois da exclusão do usuário. Para fazer isso, tome nota da ID do usuário a partir do arquivo de senha, antes de excluí-lo, e depois use o comando *find*:

```
# find / -type f -uid 503 -print -exec rm {} \ ou
```

O *-type f* indica que o comando deve procurar apenas arquivos; *-uid 503* indica que apenas os arquivos pertencentes ao usuário com ID 503 devem ser retornados (esse número deve ser substituído pela ID do usuário que você está excluindo); *-exec rm {} *; indica que o comando *rm* deve ser executado em cada arquivo encontrado, removendo os arquivos efetivamente.

12.4 Gerenciando grupos

Criando grupos

Você pode incluir novos grupos em seu sistema usando o comando *groupadd* (esse comando se chama *addgroup* em algumas distribuições)

```
# groupadd groupname
```

O grupo será criado e receberá um novo número de usuário de acordo com a seguinte regra: “o padrão é usar o menor valor de ID que seja maior do que 500 e maior do que qualquer outro grupo”.

Se você quiser especificar o número do grupo, basta usar o flag **-g** para indicar o número:

```
# groupadd -g 503 groupname
```

Incluindo usuários em grupos

Não existe um programa padrão disponível para se incluir facilmente usuários em um grupo. O modo mais fácil é editar diretamente o arquivo */etc/group*. Cada linha desse arquivo representa a definição de um grupo e assume a forma

```
groupname:password:groupid:userlist
```

```
Ex.: group1::505:user1,user2,user3
```

Normalmente as senhas não são aplicadas aos grupos, de modo que normalmente isso fica em branco.

O *groupid* é o ID numérica do grupo e deve ser única para ela.

Se quiser incluir usuários em um grupo existente, basta editar o arquivo */etc/group* com seu editor de textos predileto e incluir os nomes dos usuários no final da lista, separando cada usuário com uma vírgula.

Excluindo grupos

A exclusão de grupos é feita com o comando *groupdel*.

```
# groupdel groupname
```

Contudo, toda essa simplicidade traz alguns inconvenientes:

- Os arquivos pertencentes ao grupo não serão excluídos ou mudarão de grupo.
- Se o grupo serve como grupo principal de um usuário (em outras palavras, é indicado como grupo do usuário no arquivo de senha), então ele não será excluído.

Para resolver essa questão, devemos primeiramente anotar a ID do grupo que estamos excluindo (isso pode ser encontrado no arquivo */etc/group*).

Quando o grupo for excluído com *groupdel*, podemos usar o comando *find* para mudar a posse de grupo de todos os arquivos pertencentes ao grupo excluído:

```
# find / -type f -gid 503 -print -exec chgrp newgroupname {} \;
```

12.5 Checando a consistência de sistema de arquivos

O que acontece quando o Sistema de arquivos está com problemas? Toda vez que o Linux é iniciado, um programa chamado “fsck” é iniciado também. Esse programa analisa e conserta qualquer problema no seu sistema de arquivos. Por esse motivo, os usuários Linux não tem com que se preocupar (aqui não ocorre fragmentação de arquivos como no DOS). Somente em casos mais sérios, como falhas de hardware ou interrupções bruscas, pode haver danos. Nesses casos, o fsck pára na inicialização e pede intervenção do administrador. Esse programa pode verificar múltiplos sistemas de arquivos usando paralelismo.

Durante a inicialização, esse programa verifica todos os sistemas de arquivos, Caso ocorra um erro, o usuário administrador recebe um prompt, entra no sistema e digita fsck com algumas opções, geralmente -a, que realiza reparos automaticamente. Lembre-se que realizar reparos em sistemas de arquivos montados pode causar problemas. Onde está o backup?

12.6 Inicialização do sistema

As mensagens de inicialização são salvas no arquivo de log de sistema */var/log/messages*.

O que acontece durante a inicialização

Existem dois estágios no processo de inicialização:

3. Inicialização do kernel. Durante essa fase, o Kernel é carregado na memória e imprime mensagens à medida que inicializa cada driver de dispositivo.

4. Execução do programa *init*. Esse programa manipula a ativação de todos os programas, incluindo daemons de sistema essenciais e outro software especificado para ser carregado no momento da inicialização.

O programa Init

Aqui você pode personalizar facilmente quais programas são carregados durante o ciclo de inicialização.

O Linux tem um sistema de *níveis de execução*. Um nível de execução é um número que identifica o estado atual do sistema e quais processos o *init* deve executar e manter em execução nesse estado do sistema. No arquivo */etc/inittab*, a primeira entrada especifica o nível de execução padrão que é carregado durante a inicialização.

id:3:initdefault: (nível de execução 3)

Assim, para o nível de execução 3, as seguintes linhas são relevantes:

```
13:3:wait:/etc/rc.d/rc 3
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

As últimas seis linhas configuram os seis consoles virtuais fornecidos no Linux. A primeira linha executa o script de inicialização */etc/rc.d/rc 3*. Isso executará todos os scripts contidos no diretório */etc/rc.d/rc3.d*. Esses scripts representam programas que precisam ser iniciados na inicialização do sistema, como *sendmail*, serviços PCMCIA, o daemon de impressora e *crond*.

Daemons - São processos servidores, inicialmente inicializados durante o boot, que rodam continuamente enquanto o sistema estiver ativo, esperando, em background, até que um processo requisite seus serviços. Por exemplo, network daemons em estado idle até que um processo requisite serviços de rede.

O último script a ser executado será o *AS99local*. Esse script é, na verdade, um vínculo para o arquivo */etc/rc.d/rc.local*; é aqui que você pode colocar todos os programas de inicialização personalizada que quiser ativar no momento da inicialização.

12.7 Programando tarefas com crond

O *crond* é um daemon que é geralmente instalado para iniciar sempre que o sistema for inicializado. Quando ele inicia é ativado a cada minuto e verifica se alguma tarefa foi programada para executar durante esse minuto.

Todas as tarefas programadas são armazenadas em um arquivo de configuração individual (conhecido como arquivo *crontab*) do usuário, com cada linha representando uma tarefa programada.

Cada entrada tem a forma

time-date command

A entrada *time-date* consiste em cinco campos numéricos, cada um separado por espaços, que indicam quando uma tarefa deve ser executada. Os cinco campos em ordem, são:

- **Minute:** De 0 a 59.
- **Hour:** De 0 a 23.
- **Day of month:** De 0 a 31.
- **Month:** De 0 a 12.
- **Day of week:** De 0 a 7, onde 0 e 7 representam domingo.

Para todos esses campos, várias regras proporcionam flexibilidade:

- Intervalos de números podem ser usados (Ex.: 1 – 3 no campo horas diz para programar o comando para 1:00, 2:00 e 3:00 da manhã. 2 – 4 no campo de dia da semana programa a tarefa para terça, quarta e quinta-feira.

O campo command

Às vezes, os comandos (como o *mail*) exigem que informações sejam introduzidas através da entrada padrão. Isso é feito usando-se sinais de porcentagem (%). O primeiro sinal de porcentagem marca o início da entrada padrão e cada sinal subsequente serve como um caractere de nova linha nessa entrada.

```
30 14 * * fri /bin/mail -s "TGIF" arna@lala.edu %Bom dia!!!!%% Tudo Bem???
```

Isso enviará a seguinte mensagem de correio eletrônico:

Bom dia!!!

Tudo bem???

Carregando entradas de um arquivo

Para carregar entradas de um arquivo, primeiro é necessário criar um arquivo que contenha todas as entradas que você queira que apareçam em seu arquivo *crontab*. Um exemplo de arquivo poderia conter duas entradas:

```
0 1 * * * /usr/local/bin/backup
30 14 * * fri /bin/mail -s "TGIF" arna@lala.edu %Bom dia!!!! %% Tudo Bem???
```

Esse arquivo precisa ser salvo com um nome conveniente, como *cronjobs*.

Uma vez criado e salvo o arquivo, ele pode ser carregado no arquivo *crontab* de um usuário, executando-se o comando:

```
$ crontab cronjobs
```

O conteúdo de *cronjobs* sobrescreverá todas as entradas correntes do arquivo *crontab* do usuário.

Usando o flag **-u**, o usuário-root pode especificar que o arquivo *crontab* de outro usuário, em vez do seu próprio, deve ser alterado.

```
$ crontab -u username cronjobs
```

Carregando arquivos crontab diretamente

Em vez de criar um arquivo separado e carregá-lo no arquivo *crontab*, o comando *crontab* fornece o flag **-e**, que permite ao usuário editar o arquivo *crontab* diretamente.

```
$ crontab -e
```

Assim como aconteceu quando carregamos um arquivo no arquivo *crontab*, usando o flag **-u**, o usuário-root pode editar diretamente o arquivo *crontab* de outro usuário:

```
$ crontab -u username -e
```

Vendo o conteúdo do arquivo crontab

Para ver o conteúdo do arquivo *crontab*, basta usar o flag **-l**

```
$ crontab -l
```

Assim como na edição do arquivo *crontab*, o usuário-root pode ver o conteúdo do arquivo *crontab* de qualquer usuário com o flag **-u**:

```
# crontab -u username -l
```

Removendo o arquivo crontab

Para apagar o conteúdo do arquivo *crontab* de um usuário, ele pode usar o flag **-r**:

```
# crontab -r
```

Analogamente, o usuário-root pode apagar o arquivo *crontab* de qualquer usuário com o flag **-u**:

```
# crontab -u username -r
```

12.8 Gerenciando logs

O Linux permite registrar em logs a atividade dos numerosos daemons e programas que estão em execução no sistema. Esses logs podem ser usados para depurar problemas do sistema, bem como para controlar sua utilização, cobrindo tudo, desde possíveis brechas na segurança até aviso avançado de possíveis falhas de hardware.

O que é registrado em logs?

Basicamente dois tipos de logs: logs de sistema e logs de aplicativos.

Os logs de sistema são gerados pelo daemon *syslogd*, que é carregado no momento da inicialização.

O daemon acessa mensagens em oito níveis de seriedade dos vários processos de sistema, como o kernel, o sistema de correio, programas de usuários configurados para usar *syslogd* e programas de autenticação, como o programa *login*.

Esses níveis de mensagens são, em ordem crescente de seriedade:

- debug
- info
- notice
- warning
- err
- crit
- alert
- emerg

Esses níveis são usados no arquivo */etc/syslog.conf* para informar ao *syslogd* onde deve criar logs para diferentes tipos de informação. O arquivo */etc/syslog.conf* tem várias entradas, uma em cada linha, contendo cada uma dois campos separados por um ou mais espaços: uma lista recurso-nível e uma localização do arquivo de log.

```
*.info;mail.none;authpriv.none /var/log/messages
```

Essa linha registra as mensagens de informação de todos os recursos, exceto correio e autenticação no arquivo */var/log/message*.

```
authpriv.* /var/log/secure
```

Coloca todas as mensagens de autenticação em */var/log/secure*.

```
mail.* /var/log/maillog
```

Coloca todas as mensagens de log de correio em */var/log/maillog*.

```
uucp,news.crit /var/log/spooler
```

Coloca todas as mensagens relacionadas a correio e a notícias em */var/log/spooler*.

Se você quiser mudar sua estratégia de registro de log, editando o arquivo */etc/syslog.conf*, pode fazer isso editando esse arquivo e depois dizendo ao *syslogd* para que recarregue a configuração, com o comando

```
# kill -HUP `cat /var/run/syslogd.pid`
```

O flag **-HUP** do comando *Kill* indica que o processo deve reler sua configuração, mas continuar em execução.

Alternando logs

Para que os logs permaneçam úteis, eles precisam ser alterados regularmente. Em um sistema monousuário podemos apenas removê-lo e recriá-lo.

```
# rm /var/log/messages
# Kill -HUP 'cat /var/run/syslogd.pid'
```

Em servidores multiusuários, as informações de histórico são importantes. Nesse caso você deve criar um histórico de logs.

```
# mv /var/log/message /var/log/message.1
# aKill -HUP 'cat /var/run/syslogd.pid'
```

Na maioria dos sistemas, você pode querer automatizar esse procedimento, executando-o a cada semana em determinado horário. Para fazer isso, primeiro você precisa criar um script que realize as ações necessárias para alternar seus arquivos de log.

```
# !/bin/sh
# mv /var/log/messages /var/log/messages.1
# mv /var/log/secure /var/log/secure.1
# mv /var/log/maillog /var/log/maillog.1
# mv /var/log/spooler /var/log/spooler.1
# Kill -HUP 'cat /var/run/syslogd.pid'
```

Esse script precisa ser criado com um editor de textos em uma localização lógica (como `/usr/local/bin/newlogs`) e depois transformado em um arquivo executável:

```
# chmod 755 /usr/local/bin/newlogs
```

Em seguida, você precisa editar o arquivo *crontab* do usuário-root e incluir uma entrada apropriada. Por exemplo, para executar o script a cada domingo de manhã, às 12h01 min:

```
1 12 * * sun /usr/local/bin/newlogs
```


13 - Dicas sobre hardware

13.1 Utilizando um disco flexível no Linux

Em primeiro lugar, um disquete é uma unidade de volume assim como o CD, e deverá ser montado antes do uso.

Para formatar um disquete com sistema de arquivos ext2:

```
# fdformat /dev/fd0H1440
```

Para criar o sistema de arquivos ext2:

```
# mkfs -t ext2 -c -m 0 /dev/fd0H1440
```

Para montar um floppy disk, isto é, um disco flexível, você terá que utilizar o comando 'mount'. Você terá que ter o driver e o device respectivamente (fd0, fd1, fd2, etc). Então você deverá digitar:

```
# mount -t ext2 /dev/fd0 /mnt/floppy
```

Para montar um disquete com o sistema de arquivos msdos:

```
# mount -t msdos /dev/fd0 /mnt/floppy
```

Isto fará com que você acesse o disquete que está no drive atualmente.

Quando você quiser retirar o disco geralmente deve-se 'desmontá-lo' primeiro. Digite:

```
umount /mnt/floppy
```

Você pode também fazer o seguinte, criar um script, que se chama, por exemplo de 'diskon' (Para ativar) e 'diskoff' (Para desativar).

Então para melhor utilização, coloque este arquivo em um diretório PATH, ou então coloque o PATH no diretório onde você quiser colocar os scripts.

14 - Instalação de placa de rede adicional.

Verificar a marca e o modelo das placas de rede a serem instaladas em cada estação, bem como dados como interrupção e endereço de memória. Caso as placas sejam do tipo PCI essa informação é desnecessária, porém no caso de placas ISA ela é essencial. Anote numa tabela os dados encontrados.

Dica: Caso seja uma placa nova verificar no disquete de instalação se a mesma é NE2000 compatível e se existem drivers para Linux da mesma. Ainda com o disquete, caso necessário escolher e ajustar um endereço de memória (I/O) e interrupção (IRQ) para a mesma.

No Linux, usando o console ou a interface gráfica – à sua escolha - entre no Linuxconf e vá em *Configuração -> Ambiente de Rede -> Tarefas de Cliente -> Configuração Básica desta Máquina*.

Na aba *Nome da máquina* preencher o campo *Nome da máquina*. Escolher um nome para a máquina na rede, de preferência preencher em letras minúsculas.

As abas *Adaptador 1, Adaptador 2, etc...* serão preenchidas da seguinte forma:

- a. Selecionar o botão *ativo* para que a placa seja ativada durante o boot.
- b. Escolher o *Modo de Configuração* (endereço IP). *Manual*: atribuição de IP fixo (ideal para o caso de servidores), *DHCP*, *BOOTP*: atribuição de IP dinâmico usando protocolos DHCP ou BOOTP.
- c. Preencher o campo *Primeiro nome + domínio*. Nesse caso, caso a sua máquina vá estar na internet, colocar o nome principal dela (mesmo do item 3) acrescido de .domínio. Caso a sua máquina não vá participar de nenhum domínio preencher somente o nome da máquina, de preferência coincidindo com o nome dado no item 3. Se sua máquina fará parte da internet usando conexão a um provedor, sem domínio próprio pode-se colocar aqui o domínio do provedor.
- d. Em *Endereços IP* preencher somente se o *Modo de Configuração* selecionado foi *manual*. Numa intranet prefira sempre os endereços IP's reservados. Ex. 192.168.X.Y, 10.X.Y.Z. Caso sua máquina tenha ligação direta com a internet preencher aqui somente com endereços válidos.
- e. *Máscara* de sub-rede. Preencher de acordo com o projeto da sua rede. Se não for preenchido, será assumido o valor default 255.255.255.255. Embora esteja marcando como opcional, é sempre bom tê-la preenchido.
- f. Selecionar *Dispositivo de Rede*. No caso de redes ethernet, que é o padrão mais usado, selecionar *eth0* para o *Adaptador1*, *eth1* para o *Adaptador2* e assim por diante.
- g. Em *Módulo do Kernel* selecionar o apropriado para a sua placa de acordo com o modelo da mesma. Caso não exista o módulo referente à sua placa, caso ela seja NE2000 Compatível selecione-o (*ne* para o caso de placas ISA ou *ne2k-pci* para o caso de placas PCI). Nesse caso elas não funcionaram a 100Mbps. Se preferir, siga as orientações do manual da sua placa para instalar o módulo do kernel correto, será necessário na maioria das vezes recompilação do mesmo, adicionando o módulo necessário.

- h. Os campos de *Porta E/S* e *IRQ* devem ser preenchidos na seguinte situação: duas placas de rede usando o mesmo módulo do kernel. Ainda que a sua placa seja ISA NE2000, caso ela seja a única instalada na sua máquina o Linux é capaz de detectá-las automaticamente. Se falhar, preencha manualmente.
- i. Confira tudo, após sair do Linuxconf confirmando as alterações, no arquivo */etc/conf.modules*. Aqui é guardada a informação de quais módulos do kernel serão carregados no boot e a qual dispositivo esse módulo está associado. No console, usando o comando *ifconfig* verifique se ambas estão funcionando. O comando *ifconfig* seria usado para instalar a placa caso não tivéssemos o Linuxconf.
- j.

Outras opções de configuração seriam a edição direta dos arquivos: */etc/sysconfig/network-scripts/ifcfg-ethX* ou a utilização do programa *netconfig*.

A instalação dos módulos compilados é feita usando-se o comando:

insmod modulo.o

15 - Configuração de Impressora

A ferramenta de configuração de impressoras que iremos utilizar é o printtool. Ele atua sobre o arquivo `/etc/printcap`, os diretórios de tarefas de impressão e os filtros de impressão. Os filtros permitem que se imprimam diferentes tipos de arquivos incluindo:

- ASCII (texto).
- PostScript.
- Tex .dvi.
- RPMs.
- GIF, JPEG, TIFF e outros formatos gráficos.

Em outras palavras, ao imprimir arquivos GIF ou RPM utilizando-se o comando `lpr` os arquivos serão tratados adequadamente.

Para criar-se uma fila de impressão nova, deve-se escolher Add e então selecionar o tipo de impressora.

Há quatro tipos de filas de impressão que podem ser criadas:

- Local: filas de impressão para impressoras instaladas diretamente na máquina local.
- Remote: filas de impressão direcionadas para outras estações de uma rede TCP/IP.
- SMB: filas de impressão direcionadas para sistemas que utilizem uma rede tipo SMB (Windows© por exemplo).
- NCP: filas de impressão direcionadas para sistemas de impressão baseados em Novell Netware©.

Após escolher o tipo de impressora, uma janela de diálogo solicitará maiores informações sobre a fila de impressão. *Todos* os tipos de filas requerem as seguintes informações:

- Queue Name: nome da fila de impressão. Vários nomes podem ser especificados utilizando-se o separador.
- Spool Directory: diretório no sistema local onde ficarão armazenados os arquivos antes que a impressão ocorra. Nota: não deve ser definido o mesmo diretório para mais de uma fila de impressão.
- File Limit: tamanho máximo permitido para o arquivo de impressão. Um tamanho igual a zero indica que não há limite .
- Input Filter: filtros são ferramentas de conversão de arquivos de impressão para formatos que a impressora selecionada pode tratar. Pressione Select para escolher o filtro que mais fique adequado à impressora. Além de impressoras capazes de imprimir gráficos e PostScript, é possível configurar dispositivos que imprimam somente arquivos em formato texto. A maioria dos arquivos de controle de impressoras são capazes de imprimir arquivos ASCII, sem convertê-los para PostScript. Para habilitar esta funcionalidade selecione Fast text printing ao se configurar o filtro. Nota: esta funcionalidade somente está habilitada para impressoras que não sejam PostScript.
- Suppress Headers: verifica se há necessidade de imprimir uma página de início antes de cada impressão.

Para impressoras locais, os seguintes dados são necessários:

- Printer Device: normalmente /dev/lp1 é o nome da porta à qual a impressora está conectada. Impressoras seriais estão normalmente em portas /dev/ttyS?, sendo necessário definir manualmente os parâmetros de configuração deste tipo de impressoras.

Para impressoras remotas são necessárias as seguintes informações:

- Remote Host: nome do servidor remoto ao qual a impressora está conectada.
- Remote Queue: nome da fila de impressão no servidor remoto.

O servidor remoto deverá estar configurado de forma a permitir que a máquina local utilize a sua fila de impressão. O arquivo /etc/hosts.lpd controla estas informações.

Para impressoras SMB e NCP, são necessárias as seguintes informações:

- Hostname of Printer Server: nome da máquina à qual a impressora está conectada.
- IP number of Server: o endereço IP da máquina à qual a impressora está conectada (opcional).
- Printer Name: nome da impressora no sistema Windows, por exemplo.
- User: nome do usuário para acessar a impressora (normalmente guest em servidores Windows ou nobody para servidores samba).
- Password: senha (se necessária) para utilizar a impressora (normalmente brancos).

É recomendável que o usuário e sua senha, quando utilizados, sejam diferentes de usuários e senhas do *Conectiva Linux* ou de usuários de compartilhamento de arquivos, para que se tenha um maior nível de segurança de acessos. O mesmo procedimento é indicado para usuários utilizados no compartilhamento de arquivos em um servidor SMB ou Novell. Isso se deve ao fato de que usuários e senhas para acesso a filas de impressão são armazenados no sistema local em formato transparente, ou seja, sem criptografia.

Após adicionar a fila de impressão é necessário reinicializar o servidor de impressão. Para tanto basta clicar sobre Restart lpd no menu lpd.

Pode-se imprimir uma página de teste em qualquer fila de impressão selecionada. Selecione o tipo de página de teste no menu Tests.

Caso o teste de impressão gere apenas uma linha, selecione a impressora, opções Edit, select e na opção input filter, marque a opção fix stair-stepping.

Para imprimir a partir da linha de comando do interpretador ou de um terminal xterm, pode ser executado o comando <lpr -P nome-da-impressora arquivo-para-impressão>, onde <nome-da-impressora> é o nome cadastrado na fila de impressão e é sempre o nome na primeira coluna da saída do comando printtool.

15.1 Como configurar uma impressora remota em uma rede linux ?

Edite o arquivo /etc/hosts das máquinas ou configure o DNS de sua rede:

```
127.0.0.1 localhost.localdomain localhost
192.168.255.1 servidor.dominio servidor
192.168.255.2 cliente.dominio cliente
```

Configure a impressora da máquina servidora normalmente, usando o printtool.

Edite o arquivo `/etc/hosts.lpd` da máquina servidora que máquinas da rede terão acesso ao servidor:

```
cliente.dominio cliente2.dominio ...
```

Adicione uma impressora "Remote Unix (lpd) Queue" em cada máquina cliente, pelo programa `printtool`.

No campo Remote Host, digite o nome completo do servidor, ex.: `servidor.dominio`

No campo Remote Queue, digite o nome da fila de impressão que foi definida no servidor, ex.: `lp`

Feito isto, basta experimentar imprimir algo no servidor, por exemplo:

```
[usuario@localhost]$ lpr .bashrc
```

16 - Importantes Funcionalidades de Rede

Após configurar com sucesso o endereço IP e o resolvedor de nomes, deve-se configurar os serviços que serão disponibilizados através da rede. Este capítulo cobre a configuração de algumas aplicações importantes de rede, inclusive o servidor `inetd` e os programas da família `rlogin`. A interface RPC - Chamada de Procedimentos Remotos, na qual serviços como o Sistema de Arquivos em Rede - NFS e Sistemas de Informações em Rede - NIS estão baseados.

16.1 O Superservidor `inetd`

Freqüentemente, serviços são executados por programas denominados servidores. Um servidor é um programa que abre uma determinada porta e fica aguardando por solicitações de conexão. Quando uma solicitação é recebida, ele cria um processo filho, o qual trata aquela conexão específica, enquanto o processo pai continua a escutar na porta aguardando novas solicitações. Este conceito sintetiza, na sua essência e de maneira simplificada, a forma como os serviços são oferecidos em uma máquina Linux, ou seja um servidor escutando em uma porta, aguardando pedidos de conexão, o que geralmente pode significar a perda de alguns recursos de sistema, como por exemplo a área de troca.

Porém, praticamente todo sistema Unix executa uma espécie de superservidor, o qual é capaz de criar conectores para uma série de serviços e ouvir todas as portas simultaneamente, utilizando para tanto uma chamada ao sistema denominada `select(2)`. Quando uma máquina remota solicita algum de seus serviços, o superservidor percebe o fato e aciona o servidor específico da porta envolvida. O superservidor normalmente utilizado é conhecido por `inetd`,

Ele inicia a sua execução quando o sistema é inicializado, recebendo a lista de serviços a serem monitorados a partir de um arquivo denominado `/etc/inetd.conf`. Além dos serviços que envolvem outros servidores, existe uma série de serviços simples que são executados pelo próprio `inetd` denominados serviços internos. Eles incluem a função `chargen`, a qual simplesmente gera uma cadeia de caracteres e a função `daytime`, a qual retorna o conceito do sistema da hora do dia.

Um registro neste arquivo consiste de uma linha simples composta pelos seguintes campos:

serviço tipo protocolo espera usuário servidor linha_de_comando

O significado dos campos é o seguinte:

serviço Fornece o nome do serviço a ser disponibilizado. Ele deve ser traduzido em um número de porta através de uma pesquisa no arquivo `/etc/services`. Este arquivo será descrito na seção Os Arquivos `services` e `protocols`, a seguir.

tipo Especifica o tipo de conexão que será utilizada, stream (para conexões orientadas a protocolo) ou dgram (para protocolos que utilizem datagramas). Serviços baseados em TCP devem sempre ser especificados como stream, enquanto que serviços baseados em UDP devem sempre ser definidos como dgram.

protocolo Especifica o nome do protocolo usado pelo serviço. Deve ser um nome válido que possa ser encontrado no arquivo protocols, também explicado a seguir.

espera Esta opção aplica-se somente a conexões por datagramas. Ela pode ser igual a wait ou nowait. Caso wait seja especificado, inetd irá executar somente um servidor por vez para a porta especificada. De outra forma, ele imediatamente voltará a ouvir a porta após atender a uma requisição. Isso é útil para servidores `_mono-executáveis_` que necessitam ler todos os

datagramas até que mais nenhum seja enviado, e então finaliza. Muitos servidores RPC são deste tipo e devem ter esta configuração. O tipo oposto, os servidores `_multi-executáveis_` permitem um número ilimitado de instâncias do programa sendo executadas concorrentemente. Estes são utilizados mais raramente. Estes servidores devem receber o parâmetro nowait. Conexões stream devem sempre usar o parâmetro nowait.

usuário Esta é a identificação de acesso do usuário sob o qual o processo será executado. Como frequência ele será igual ao superusuário root, porém alguns serviços podem utilizar contas diferentes. É aconselhável aplicar os princípios de uso do usuário menos privilegiado, o que significa que não se deve executar um comando com uma conta com privilégios que não sejam realmente necessários na sua execução. Por exemplo, o servidor de notícias NNTP será executado com o usuário news, enquanto serviços que podem produzir riscos de segurança, como por exemplo o tftp ou finger, são normalmente executados com o usuário nobody.

servidor Fornece o caminho completo do programa servidor a ser utilizado. Serviços internos terão um valor igual a internal neste campo.

linha_de_comando Esta é a linha de comando a ser enviada para o servidor. Isso inclui o argumento 0, que é o nome do comando. Normalmente conterá o nome do programa servidor, a menos que o programa comporte-se diferentemente quando acionado com um nome diferente. Este campo não deverá conter nenhuma informação para serviços internos.

Um exemplo do arquivo `\file{/etc/inetd.conf}`.

```
#
# serviços inetd
ftp stream tcp nowait root /usr/sbin/ftpd in.ftpd -l.p p
telnet stream tcp nowait root /usr/sbin/telnetd in.telnetd -b/etc/issue
#finger stream tcp nowait bin /usr/sbin/fingerd in.fingerd
#tftp dgram udp wait nobody /usr/sbin/tftpd in.tftpd
#tftp dgram udp wait nobody /usr/sbin/tftpd in.tftpd /boot/diskless
login stream tcp nowait root /usr/sbin/rlogind in.rlogind
shell stream tcp nowait root /usr/sbin/rshd in.rshd
exec stream tcp nowait root /usr/sbin/rexecd in.rexecd
#
# serviços internos inetd
```



```
#
daytime stream tcp nowait root internal
daytime dgram udp nowait root internal
time stream tcp nowait root internal
time dgram udp nowait root internal
echo stream tcp nowait root internal
echo dgram udp nowait root internal
discard stream tcp nowait root internal
discard dgram udp nowait root internal
chargen stream tcp nowait root internal
chargen dgram udp nowait root internal
```

Um exemplo de um arquivo `inetd.conf` foi mostrado anteriormente. O serviço `finger` está comentado (contém um caractere `#` no início da linha), portanto não está disponível. Isso é feito com frequência, por motivos de segurança, uma vez que ele pode ser usado por intrusos para obter nomes válidos de usuários do sistema local.

16.2 A Funcionalidade `tcpd` de Controle de Acesso

Abrir um computador para acesso pela rede envolve muitos riscos de segurança. Algumas aplicações foram escritas para proteger o sistema contra diversos tipos de ataques. Alguns desses porém podem ser bastante frágeis, ou não conseguem distinguir entre máquinas seguras a partir das quais a requisição de um determinado serviço será aceita e máquinas inseguras cujas solicitações serão rejeitadas. Já comentamos rapidamente os serviços `finger` e `tftp` acima. Deve-se limitar o acesso a estes serviços somente a máquinas confiáveis, o que é impossível com a configuração usual, pois o programa `inetd` disponibiliza um serviço a todos os

clientes ou a nenhum. Uma ferramenta útil nestes casos é o servidor `tcpd`, um servidor de observação.

Para os serviços TCP que se deseje monitorar ou proteger, ele deverá ser acionado ao invés do programa servidor. O programa `tcpd` registra todas as requisições através do servidor de mensagens do sistema denominado `syslog`, verifica se a máquina remota tem permissão de usar este serviço e somente se a resposta for positiva executará o real servidor do serviço. Cabe ressaltar que esta funcionalidade não está disponível para serviços baseados em UDP.

Por exemplo, para observar o serviço `finger`, deve-se alterar a linha correspondente no arquivo `inetd.conf` para:

```
# servidor de observação do serviço finger
finger stream tcp nowait root /usr/sbin/tcpd in.fingerd
```

Sem a adição de qualquer controle de acesso, o cliente não perceberá qualquer diferença de um serviço `finger` usual, exceto pelo fato de que todas as requisições serão registradas pelo `syslog`.

O controle de acesso é implementado através de dois arquivos denominados `/etc/hosts.allow` e `/etc/hosts.deny`. Eles contêm informações que permitem e negam o acesso, respectivamente. Quando o servidor `tcpd` manuseia uma requisição para um serviço, como por exemplo uma chamada ao `finger` a partir de um cliente chamado `itabaiana.cvirtual.com.br`, ele pesquisa nos arquivos `hosts.allow` e `hosts.deny` (nesta ordem) buscando uma entrada que coincida com o serviço e com a máquina cliente. Caso uma entrada seja encontrada no arquivo `hosts.allow`, o acesso será permitido, independente de qualquer referência no arquivo `hosts.deny`.

Caso alguma entrada seja encontrada no arquivo `hosts.deny`, a requisição será rejeitada e a conexão encerrada. Caso nenhuma entrada seja encontrada, a requisição será aceita.

Registros nos arquivos de acesso têm a seguinte aparência:

```
lista_de_serviços: lista_de_máquinas [:comando]
```

`lista_de_serviços` é uma relação de nomes de serviços existentes no arquivo `/etc/services` ou a palavra chave `ALL`. Para definir todos os serviços exceto

```
finger e tftp, use ALL EXCEPT finger, tftp_.
```

`lista_de_máquinas` é uma lista de nomes de máquinas ou endereços IP, ou as palavras chave `ALL`, `LOCAL`, ou `UNKNOWN`. `ALL` significa qualquer máquina, enquanto `LOCAL` é utilizado para qualquer máquina cujo nome não contenha um ponto.

`UNKNOWN` significa qualquer máquina cujo nome ou endereço não seja localizado. Um nome começando com um ponto significa todas as máquinas de um determinado domínio. Por exemplo, `.cvirtual.com.br` faz com que a máquina `itabaiana.cvirtual.com.br` tenha acesso ao serviço. Há ainda o uso de endereços IP para redes e sub-redes. Por favor consulte a página de manual `hosts_access(5)` para maiores detalhes.

Para evitar o acesso aos serviços `finger` e `tftp` para todas as máquinas, exceto as máquinas locais, deve ser criado o arquivo `/etc/hosts.deny` com o seguinte conteúdo, deixando o arquivo `/etc/hosts.allow` vazio:

```
in.tftpd, in.fingerd: ALL EXCEPT LOCAL, .seu.dominio
```

O campo opcional `comando` pode conter um comando que pode ser acionado quando a entrada coincidir. Isso é útil para configurar armadilhas que exponham intrusos em potencial:

```
in.ftpd: ALL EXCEPT LOCAL, .cvirtual.com.br :
echo "origem da solicitação em %d@%h" >> /var/log/finger.log;
if [ %h != "aracaju.cvirtual.com.br" ]; then
finger -l @%h >> /var/log/finger.log
fi
```

Os argumentos `%h` e `%d` são expandidos para o nome da máquina cliente e o nome do serviço pelo programa `tcpd`, respectivamente. Por favor consulte a página de manual do `hosts_access(5)` para maiores detalhes. Normalmente somente nomes de máquinas obtidos de pesquisas no arquivo `/etc/hosts` não contêm pontos.

O número das portas de certos serviços padrão são definidos na RFC denominada Definindo Números. Para viabilizar que programas servidores ou clientes convertam os nomes de serviços para estes números, no mínimo parte desta lista deve ser mantida em cada máquina. Estas definições são armazenadas em um arquivo chamado /etc/services. Uma entrada neste arquivo tem o seguinte formato:

```
serviços porta/protocolo [apelidos]
```

O parâmetro serviço especifica o nome do serviço, porta define a porta onde o serviço é oferecido e protocolo define qual o protocolo de transporte a ser usado. Comumente, ele será udp ou tcp. É possível que um serviço seja oferecido por mais de um protocolo, assim como serviços diferentes podem ser oferecidos na mesma porta. O campo apelidos permite a especificação de nomes alternativos para o mesmo serviço.

Usualmente, não se deve alterar o arquivo de serviços que vem com o software de rede de seu sistema Linux. De qualquer forma, segue uma demonstração do conteúdo do arquivo.

```
# Arquivo de serviços:
#
# serviços bem conhecidos
echo 7/tcp # Eco
echo 7/udp #
discard 9/tcp sink null # Descartar
discard 9/udp sink null #
daytime 13/tcp # Hora do Dia
daytime 13/udp #
chargen 19/tcp ttytst source # Gerador de Caracteres
chargen 19/udp ttytst source #
ftp-data 20/tcp # Prot.de Transf. de Arquivos(Dados)
ftp 21/tcp # Prot.de Transf. de Arquivos(Control)
telnet 23/tcp # Protocolo de Terminais Virtuais
smtp 25/tcp # Prot. Simples de Transf. de Mensagens
nntp 119/tcp readnews # Prot. de Transf. de Rede de Notícias
#
# serviços UNIX
exec 512/tcp # rexecd BSD
biff 512/udp comsat # notificação de correio
login 513/tcp # acesso remoto
who 513/udp whod # comandos remotos who e uptime
shell 514/tcp cmd # comando remoto,sem senha
syslog 514/udp # sistema remoto de registro
printer 515/tcp spooler # sistema de impressão remota
route 520/udp router routed # protocolo de informações de roteamento.p p
```

Note que o serviço echo, por exemplo, é oferecido na porta 7 por ambos os protocolos TCP e UDP, e que a porta 512 é usada para dois serviços denominados servidor COMSAT (o qual notifica o usuário da chegada de mensagens; veja por exemplo o programa xbiff(1x)), sobre UDP, e é usada para a execução de comandos remotos (rexec(1)), usando TCP.

Similar ao arquivo `deserviços`, a biblioteca de rede necessita traduzir nomes de protocolos, por exemplo aqueles usados no arquivo `serviços`, em números conhecidos pela camada IP de outras máquinas. Isso é feito através de uma pesquisa no arquivo `/etc/protocols`. Ele contém uma entrada por linha, constituída pelo nome do protocolo e o número associado. É muito improvável que este arquivo deva ser alterado. Um exemplo é apresentado a seguir:

```
#
# Protocolos Internet (IP)
#
ip 0 IP # protocolo internet, número do pseudo-protocolo
icmp 1 ICMP # protocolo de controle de mensagens internet
igmp 2 IGMP # protocolo de propagação de grupos internet
tcp 6 TCP # protocolo de controle de transmissão
udp 17 UDP # protocolo de datagrama de usuário
raw 255 RAW # interface de IP não tratado
```

16.3 RPC - Chamada de Procedimento Remoto

Um mecanismo muito genérico de aplicações cliente-servidor é disponibilizado pelo pacote RPC Chamada de Procedimento Remoto. Ele foi desenvolvido pela Sun Microsystems e é uma coleção de ferramentas e bibliotecas. Aplicações importantes construídas sobre a RPC são NFS - Sistemas de Arquivos em Rede e NIS - Sistemas de Informações em Rede, ambos os quais serão melhor descritos em capítulos posteriores.

Um servidor RPC consiste de uma coleção de procedimentos que um cliente utiliza enviando uma solicitação RPC ao servidor, junto com os parâmetros do procedimento. O servidor irá acionar o procedimento indicado pelo cliente, retornando para aquele os valores obtidos, caso haja algum que tenha sido retornado pelo programa acionado. Para tornar este serviço independente de plataforma, todos os dados trocados entre o cliente e o servidor são convertidos para o formato XDR - Representação Externa de Dados pelo emissor e convertido para a representação da plataforma local pelo receptor.

De qualquer forma, programas RPC têm números de versão definidos, normalmente iniciados por 1, e onde cada nova versão da interface incrementa a versão como em um contador. Frequentemente, um servidor é oferecido em diversas versões simultaneamente, permitindo que os clientes possam então indicar o número da versão desejada nas suas requisições. A comunicação de rede entre servidores RPC e cliente é bastante peculiar. Um servidor RPC oferece uma ou mais coleções de procedimentos, onde cada conjunto é chamado por um programa e é identificado por um número de programa. Uma lista mapeando o nome do serviço para o número de programa é normalmente mantida no arquivo `/etc/rpc`, o qual é parcialmente apresentado a seguir.

```
Um exemplo do arquivo /etc/rpc.
#
# /etc/rpc - diversos serviços baseados em RPC
#
```

```
portmapper 100000 portmap sunrpc
rstatd 100001 rstat rstat_svc rup perfmeter
rusersd 100002 rusers
nfs 100003 nfsprog
ypserv 100004 ypprog
mountd 100005 mount showmount
ypbind 100007
walld 100008 rwall shutdown
yppasswdd 100009 yppasswd
bootparam 100026
ypupdated 100028 ypupdate
```

Em redes TCP/IP, os autores do RPC confrontaram-se com o problema de mapeamento de números de programas com serviços genéricos de rede. A solução foi definida de forma que cada servidor atende aos protocolos TCP e UDP para uma versão específica de um determinado programa. Geralmente aplicações RPC utilizarão UDP para o envio de dados e utilizarão TCP quando os dados a serem transferidos não caibam em um único datagrama UDP. Obviamente, programas clientes têm que encontrar uma forma de saber em qual porta o programa está mapeado. Usar um arquivo de configuração para isso poderia ser um tanto ineficiente, uma vez que aplicações RPC não utilizam portas reservadas. Não haveria nenhuma garantia de que a porta originalmente definida pela base de dados seria a utilizada pelo processo. Ou seja, aplicações RPC utilizam qualquer porta que possam e as registram no denominado servidor de mapeamento de portas

Este age como um negociador de serviços para todos os programas RPC em execução na máquina: um cliente que deseje contactar um serviço com um determinado número de programa, irá inicialmente contactar o servidor de mapeamento de portas do servidor, o qual retornará os números das portas TCP e UDP de serviço desejado.

Este método tem uma desvantagem particular, pois introduz um ponto de falha, similar ao mesmo introduzido pelo servidor inetd de serviços de Berkeley. De qualquer forma ele é um pouco pior pois o programa portmapper pode terminar anormalmente e as informações das portas RPC serão perdidas. Isso significa que os servidores RPC deverão ser reinicializados manualmente ou ainda todo o sistema deverá ser reinicializado. No Linux, o portmapper é chamado `pc.portmap` e reside no diretório `/usr/sbin`. Além de estar certo de que ele é inicializado pelo programa `rc.inet2`, o portmapper não requer qualquer esforço de configuração.

17 - DNS

Neste capítulo apresentamos informações sobre instalação, configuração e manutenção de um servidor DNS (Domain Name System) em uma máquina com Linux. Você aprenderá como criar um domínio e administrar as máquinas liga-das a ele. Veremos como fazer com que estações possam utilizar o servidor DNS para acessar umas às outras através de seus respectivos nomes. Além disso, você aprenderá um pouco sobre como o DNS consegue organizar um número absurdamente grande de máquinas conectadas possibilitando que se enxerguem, umas às outras, em uma rede.

17.1 Introdução e Conceitos

O DNS converte nomes de máquinas em endereços IP. Ele mapeia nomes para IPs e IPs para nomes. Ele é um banco de dados distribuído, permitindo, assim, que uma seção seja gerenciada localmente e esteja, mesmo assim, disponível para todo o mundo.

Os servidores de nomes compõem a parte servidor do mecanismo cliente-servidor do DNS. Os servidores de nomes contêm informações sobre uma parte do banco de dados e as torna disponíveis para os clientes ou *resolvedores*. A estrutura do banco de dados do DNS é semelhante à estrutura do sistema de arquivos do Conectiva Linux, representada por uma árvore invertida. No sistema de arquivos, tudo parte do diretório raiz; no DNS tudo parte de um nulo (""), representado em texto como um ponto ("."). Cada ramo da árvore representa uma partição do banco de dados geral - um *diretório* no sistema de arquivos do Conectiva Linux ou um *domínio* no DNS. Cada domínio pode ser dividido em mais partes chamadas *subdomínios*. Cada domínio tem um nome. Este nome é dividido em duas partes. A primeira identifica sua relação ao domínio pai. A segunda parte é o *nome do domínio*, que identifica sua posição no banco de dados. Por exemplo, podemos ver que em kepler.minhaorganizacao.com.br, o nome de domínio é minhaorganizacao.com.br e o nome da máquina é kepler.

17.2 Instalando e Configurando o DNS

17.2.1 Instalando os Pacotes

1. Acesse o diretório de pacotes do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS/ #
```

2. Instale o servidor de nomes BIND:

```
# rpm -ivh bind-8.*.rpm
```

```
bind ##### #
```

17.3 Configurando o Servidor DNS

A configuração do seu servidor de DNS pode ser realizada com facilidade através do Linuxconf, o utilitário de configurações do Conectiva Linux. É possível configurar o serviço de servidor de nome apenas através da edição manual de arquivos de configuração, mas este capítulo se concentrará na configuração que utiliza o Linuxconf. Entre no Linuxconf e vá, então, para o menu Ambiente de Rede ! Tarefas de Servidor ! DNS.

A partir desta tela, você pode configurar o seu servidor DNS inteiro sem a necessidade de edição dos arquivos de configuração. Para começar, é necessária a configuração básica do servidor, isto é, nome do domínio, endereço de correio eletrônico do administrador e máquina servidor. Para realizar estas configurações, pressione o botão domínios. Uma tela aparecerá

onde você pode adicionar, editar ou excluir os domínios de DNS. Pressione Adicionar para que possamos adicionar um domínio. Nesta tela, você deve informar os dados referentes ao seu domínio.

Domínio: este é o nome do domínio, no caso, estamos criando um domínio chamado de minhaorganizacao. Vide a seção *Domínios* para mais informações.

Servidor principal: este é o nome da máquina onde o servidor de nomes estará sendo executado. No caso de nosso exemplo, o domínio minhaorganizacao será controlado pela máquina ns.minhaorganizacao.

Email do administrador: este é o endereço de correio eletrônico do administrador de sistema. Em caso de problemas, este administrador poderá ser avisado. Note que usa-se um ponto (".") no lugar de arroba ("@") neste campo.

Há, ainda, algumas outras configurações que podem ser feitas nesta tela, mas que não cobriremos neste livro. São elas:

Divulgando o DNS(NS): em uma configuração simples, basta o nome do servidor principal. Além disso, você deverá informar aqui quais serão os servidores secundários de seu domínio.

Divulgando o EMAIL(MX): aqui você pode definir o servidor que encaminha as mensagens de correio eletrônico do seu domínio para a Internet.

IPs padrão: aqui você pode definir um ou mais endereços de IP de máquinas que serão acessadas através do domínio. É normal pesquisas em servidores de nome se referirem apenas ao domínio, mas os domínios não possuem IP's, apenas máquinas os têm, assim, definindo IPs padrão; uma pesquisa ao domínio minhaorganizacao irá resultar naquele IP padrão.

Funcionalidades: aqui podem ser definidas algumas funcionalidades do domínio. Por exemplo, pode-se definir de quanto em quanto tempo os servidores secundários serão atualizados.

Access control: você pode definir algumas opções de segurança para seu servidor de nomes. Estas configurações já são suficientes para que o servidor de nomes possa funcionar corretamente. Mas há muito mais configurações que podem ser feitas ainda em seu servidor de DNS.

17.3.1 Configurando Mapas de IPs Reversos

A tarefa principal do servidor de nomes é fazer o mapeamento entre nomes de máquinas e endereços IP. Ele realiza automaticamente a tradução de um nome de máquina para um endereço IP. Porém, às vezes é necessário fazer a tradução de um endereço IP para um nome. Isto é feito através de mapas de IPs reversos.

A configuração de um mapa de IP reverso é muito semelhante à configuração de um domínio, basta comparar a Figura 4-3 e a Figura 4-4 para verificar as similaridades.

17.3.2 Configurando um Servidor Secundário

A configuração de servidores secundários é bastante simples, basta especificar o nome do domínio para o qual este servidor será secundário e informar o IP do servidor primário de onde este servidor buscará as informações. Veja a seção *Servidores de Nomes* para mais informações sobre servidores primários e secundários.

17.3.3 Forward Zones

Existem casos em que não é interessante que o servidor de nomes envie uma grande quantidade de pacotes para fora da rede local. Isto pode ocorrer por diversos motivos, como, por exemplo, no caso de empresas que se conectam com a matriz através de antenas. Seria ineficiente que o servidor ficasse tentando localizar algo através do servidor da matriz. Para isso, pode-se criar servidores de nomes que servem para criar um grande *cache* (veja a seção *Cache*). Estes

servidores são chamados de *forwarders*.

Por exemplo, se sua empresa tem um servidor 200.20.57.1 e deseja fazer um *cache* dos servidores 200.20.57.12 e 200.20.57.13, então você poderia configurar as zonas de *forward* conforme a Figura 4-6. Para configurar um servidor de nomes como *forwarder*, não é necessária qualquer configuração especial. Você só precisa configurar os outros servidores de nomes para encaminharem suas requisições para o *forwarder*. Desta forma, os servidores tentam resolver nomes através do *forwarder*, que, por sua vez, tem grandes possibilidades de já ter a resposta em seu *cache*, evitando, assim, que a requisição tenha de sair da rede local.

17.3.4 Repetidores

O seu servidor de nomes pode ter dificuldades para resolver nomes fora do domínio por ele gerenciado. Isso pode acontecer se o DNS estiver atrás de um *firewall* e, portanto, não pode ver a Internet, ou se o DNS estiver ligado a uma conexão lenta. Isto pode ser resolvido criando-se repetidores. Os *repetidores* são servidores de nomes que podem ser utilizados para resolver nomes externos em lugar do servi-dor primário.

Quando o servidor primário não consegue resolver um nome, ele delega a tarefa para o repetidor. Como o repetidor é muito mais usado, ele tem muito mais informações guardadas, tendo uma chance muito maior de conseguir resolver nomes. Para configurar um servidor de nomes como repetidor não é necessária nenhuma configuração especial. A configuração é toda feita no servidor que utilizará o repetidor.

Para fazer com que este servidor utilize um ou mais repetidores, você deve in-formar os endereços IP dos mesmos na tela mostrada na Figura 4-7. Note que **a ordem é importante**, pois o servidor tentará sempre enviar as requisições para o primeiro, depois para o segundo e assim por diante. Assim, sempre coloque o melhor primeiro para obter uma performance desejável.

17.3.5 Funcionalidades

Na tela de funcionalidades do servidor (Figura 4-8), você pode fazer configura-ções diversas para alterar o funcionamento de seu servidor.

17.3.6 Alocação de Faixas de IP

Um servidor de nomes geralmente gerencia uma grande quantidade de máquinas associadas a endereços IP. Cada vez que uma nova máquina é adicionada ao do-mínio, um endereço IP tem de ser associado a ela e, por isso, o DNS tem de saber muito sobre a organização da rede para decidir qual endereço IP será associado à nova máquina.

Para simplificar este processo, o Linuxconf lhe dá a possibilidade de criar faixas de endereços IP que podem ser identificadas de maneira a organizar a rede.

Por exemplo, você pode criar uma faixa de endereços a serem utilizados em má-
quinas

da matriz e faixas para máquinas das filiais:

10.0.1.1-27

Matriz

10.0.2.1-15

Filial 1

10.0.3.1-21

Filial 2

Isto facilita e organiza o processo de decisão de qual IP será designado a qual máquina.

Para criar uma faixa de endereços IP, digite a faixa no campo Uma faixa de IP a descrição (nome) no campo Identificação/descrição e pressione Aceitar. Você pode criar múltiplas faixas através do botão Adicionar.

17.4 Inicializando o Serviço

Para inicializar o serviço de servidor de nomes em seu servidor, você deve abrir um terminal como superusuário e inicializar o **named**:

```
# cds
atd gpm keytable lpd nfs sendmail syslog
crond halt killall mars-nwe pcmcia single xfs
dhcpd httpd kudzu named network portma snmpd
ypbind functions inet linuxconf netfs random sshd
# ./named start Inicializando named: [ OK ]
```

Note que isso não fará com que o **named** seja reinicializado junto com o servidor. Para fazer isso, você deve utilizar o comando **ntsysv** e marcar o serviço **named** para ser inicializado sempre que o servidor for reiniciado conforme a Figura 4-10.

Marcando o **named** para reinicializar junto com o servidor, sempre que o sistema operacional entrar em ação, o servidor de nomes será inicializado com ele.

17.5 Arquivos de Configuração do BIND

Como já foi mencionado anteriormente neste capítulo, a configuração do BIND pode ser feita sem o auxílio do Linuxconf através da edição de diversos arquivos de configuração. Embora este capítulo tenha se dedicado a demonstrar a configuração do BIND através do Linuxconf, algumas notas devem ser feitas em relação aos arquivos de configuração por ele modificados. É importante conhecer estes arquivos até por uma questão de segurança. Perca ou danifique estes arquivos e você estará automaticamente dizendo adeus ao seu domínio. Assim sendo, lembre-se de manter cópias de segurança destes arquivos!

17.5.1 O Arquivo */etc/named.conf*

O */etc/named.conf* é o primeiro arquivo que você deve conhecer. Ele é um arquivo novo no BIND, pois até o BIND 4 a configuração era feita pelo arquivo */etc/named.boot*, enquanto que o BIND 8 faz tudo no */etc/named.conf*. O arquivo */etc/named.conf* do domínio *minhaorganizacao.com.br* se parece com isso:

```
// Configuração para o domínio minhaorganizacao.com.br
options { directory "/var/named"; };
zone "." { type hint; file "root.hints"; };
zone "0.0.127.in-addr.arpa" { type master; file "zone/127.0.0"; };
zone "minhaorganizacao.com.br" { type master; file
"zone/minhaorganizacao.com.br"; };
zone "177.6.206.in-addr.arpa" { type master; file "zone/206.6.177";
};
```

Note as duas últimas seções deste arquivo. A primeira define o domínio mestre (este é um domínio primário). A segunda define o IP reverso, que será usado pelo servidor de nomes para fazer o caminho inverso, ou seja, a resolução de um nome a partir de um endereço IP.

17.5.2 O Arquivo */var/named/nome-do-dominio*

Os domínios sobre os quais este servidor tem autoridade possuem um arquivo no diretório */var/named* com o nome do próprio domínio. Em nosso exemplo, portanto, este arquivo chama-se */var/named/minhaorganizacao*. Nele estão as informações do domínio que foram configuradas na tela mostrada na Figura 4-11.

O arquivo se parece com:

```
@ IN SOA
ns.minhaorganizacao. hostmaster.ns.minhaorganizacao. (
2000051001 ; serial 3600 ; refresh 900 ; retry 1209600 ; expire
43200 ; default_ttl ) @ IN MX 5 mx.minhaorganizacao. @ IN NS
ns.minhaorganizacao. ns IN A 10.0.0.1
```

17.5.3 O Arquivo */var/named/named.local*

O arquivo */var/named/named.local* define o domínio local da máquina. Ele é um arquivo padronizado, não sendo necessária a sua configuração. Basicamente, ele é uma versão de */var/named/nome-do-dominio* adaptado para o domínio e a máquina local (*localdomain* e *localhost*, respectivamente).

```
@ IN SOA localhost.
root.localhost. ( 1997022700 ; serial 28800 ; refresh 14400 ;
retry 3600000 ; expire 86400 ; default_ttl ) @ IN NS localhost.
1 IN PTR localhost.
```

17.5.4 O Arquivo */var/named/named.ca*

O arquivo /var/named/named.ca é um arquivo fornecido pela InterNIC com informações necessárias para iniciar o *cache* do servidor de nomes. Este arquivo e deve ser atualizado periodicamente, o que pode ser feito através de FTP no servi-dor FTP ftp://ftp.rs.internic.net/ da InterNIC, no diretório /domain

```
; This file holds the information on root name servers
needed to ; initialize cache of Internet domain name servers ;
(e.g. reference this file in the "cache . arquivo" ;
configuration file of BIND domain name servers). ; ; This file
is made available by InterNIC registration services ; under
anonymous FTP as ; file /domain/named.root ; on server
FTP.RS.INTERNIC.NET ; -OR- under Gopher at RS.INTERNIC.NET ;
under menu InterNIC Registration Services (NSI) ; submenu
InterNIC Registration Archives ; file named.root ; ; last
update: Aug 22, 1997 ; related version of root zone: 1997082200
; ; ; formerly NS.INTERNIC.NET ; . 3600000 IN NS
A.ROOT-SERVERS.NET. A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4 ;
; formerly NS1.ISI.EDU ; . 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A 128.9.0.107 ; ; formerly
C.PSI.NET ; . 3600000 NS C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000 A 192.33.4.12 ; ; formerly
TERP.UMD.EDU ; . 3600000 NS D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000 A 128.8.10.90 ; ; formerly
NS.NASA.GOV ; . 3600000 NS E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000 A 192.203.230.10 ; ; formerly
NS.ISC.ORG ; . 3600000 NS F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000 A 192.5.5.241 ; ; formerly
NS.NIC.DDN.MIL ; . 3600000 NS G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000 A 192.112.36.4 ; ; formerly
AOS.ARL.ARMY.MIL ; . 3600000 NS H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000 A 128.63.2.53 ; ; formerly
NIC.NORDU.NET ; . 3600000 NS I.ROOT-SERVERS.NET.
121.Capítulo 4. DNS
I.ROOT-SERVERS.NET. 3600000 A 192.36.148.17 ; ; temporarily
housed at NSI (InterNIC) ; . 3600000 NS J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000 A 198.41.0.10 ; ; housed in LINX,
operated by RIPE NCC ; . 3600000 NS K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000 A 193.0.14.129 ; ; temporarily
housed at ISI (IANA) ; . 3600000 NS L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000 A 198.32.64.12 ; ; housed in Japan,
operated by WIDE ; . 3600000 NS M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000 A 202.12.27.33 ; End of File
```

17.6 Configuração dos Clientes

17.6.1 Configuração Através do Linuxconf

A configuração de máquinas clientes de DNS é muito mais simples do que a configuração do servidor. Aquela configuração envolve basicamente a edição do arquivo `/etc/resolv.conf` ou pequenas configurações no Linuxconf. Novamente, vamos nos concentrar na configuração via Linuxconf por ser ela muito mais robusta e menos propensa a erros. Entre no Linuxconf e siga para Ambiente de Rede ! Tarefas do Cliente ! DNS para abrir a tela de especificação do servidor de nomes que pode ser vista na Figura 4-12.

Nesta tela você pode configurar as opções do servidor de nomes a ser utilizado pela estação. São elas:

Uso do DNS: esta opção serve apenas para indicar ao Linuxconf se o mesmo deve, ou não, preocupar-se com o DNS. Ela não altera o funcionamento da máquina.

Domínio Padrão: o domínio padrão é o domínio mais utilizado. Normalmente é o domínio de sua empresa. Quando você procura um nome sem domínio, o servidor de nomes procura pelo nome no domínio padrão.

Servidor de Nomes: aqui você pode definir até 3 endereços IPs de servidores de nomes. Lembre-se de que a ordem é importante, portanto coloque sempre o melhor servidor (o mais rápido) no topo.

Procurar domínio: você pode definir vários domínios nos quais um nome poderá ser encontrado. Quando um nome sem domínio não é encontrado no domínio padrão, o servidor de nomes começa a procurar nestes domínios. A ordem é importante, mas não da mesma forma como nos servidores de nomes. Aqui, você não precisa colocar os servidores mais rápidos antes, mas sim, os servidores mais utilizados.

17.6.2 O Arquivo `/etc/resolv.conf`

A configuração via Linuxconf é muito menos propensa a erros e muito mais intuitiva do que a edição do arquivo `resolv.conf`. Mesmo assim, este arquivo é extremamente simples, contendo apenas informações sobre o(s) servidor(es) a ser(em) utilizado(s).

Geralmente, este arquivo será bem pequeno e semelhante a este:

```
domain minhaorganizacao
search minhaorganizacao.com.br
nameserver 10.0.0.5
nameserver 10.0.0.7
```

O que este arquivo está nos dizendo é que esta estação está no domínio `minhaorganizacao`, está utilizando os servidores `10.0.0.5` e `10.0.0.7` para a resolução de nomes e que o servidor deve procurar por nomes no domínio `minhaorganizacao.com.br` se o mesmo não puder ser encontrado no domínio local.

18 - NIS - Sistema de Informações em Rede

Hoje em dia, NIS está disponível virtualmente para todos os Unices e existem diversas implementações gratuitas dele. Uma delas é o Net-2 da distribuição BSD que foi derivada de uma implementação de domínio público doada pela Sun. O código da biblioteca cliente desta versão está na GNU libc há um bom tempo, enquanto os programas administrativos foram recentemente portados para o Linux por Swen Thummler.

Um servidor NIS está faltando nas referências de implementações: Tobias Reber escreveu outro pacote NIS incluindo todas as ferramentas e um servidor, chamados yps.

Atualmente, uma nova versão do código NIS chamada NYS está sendo desenvolvida por Peter Eriksson, a qual suporta tanto o NIS puro quanto a revisão da Sun conhecida como NIS + . NYS não provê somente um conjunto de ferramentas NIS e um servidor, mas também adiciona um novo conjunto de funções em uma biblioteca, as quais provavelmente serão convertidas em um padrão da libc padrão. Ele inclui um novo esquema de configuração para a resolução de nome de máquina que substitui o sistema atual de uso do host.conf. As facilidades destas funções serão discutidas a seguir.

Ele pode ser encontrado em swen@uni-paderborn.de. Os clientes NIS estão disponíveis em

`yp-linux.tar.gz` no domínio `metalab.unc.edu` em `system/Network`.

A versão atual é denominada `yp-tools` e pode ser obtida em <ftp.lysator.liu.se> no diretório `/pub/NYS`.

Ele pode ser encontrado em pen@lysator.liu.se.

Este capítulo irá focar o NYS ao invés de observar os outros pacotes, os quais serão referenciados como código NIS tradicional. Caso se deseje executar quaisquer destes pacotes, as instruções deste capítulo talvez sejam suficientes. Para obter informações adicionais, por favor obtenha um livro padrão para NIS, como o `_ NFS and NIS_` de Hal Stern (ver [Stern92]), ou veja o `Como Fazer _ NIS`, traduzido pela Conectiva Informática; veja na Bibliografia, na página 459 os documentos que podem ser pesquisados. Quando este livro foi escrito originalmente, o NYS ainda estava sobdesenvolvimento, e diversos utilitários de rede do Linux como os programas `login` ainda não estavam cientes do esquema de configuração do NYS. Até que ele seja absorvido pela libc principal, ainda será necessário recompilar os binários, caso se deseje que eles utilizem o NYS. Em qualquer destas aplicações os arquivos `Makefiles`, devem

Especificar `Insl` como última opção antes da libc para a ligação do programa. Assim pode-se utilizar as funções relevantes a partir da `libnsl`, a biblioteca do NYS, ao invés da biblioteca C padrão.

18.1 Conhecendo o NIS

NIS mantém uma base de dados de informações denominadas mapas, que contém pares de chaves. Mapas são armazenados em uma máquina central que executa o servidor NIS e a partir da qual, os clientes recuperam as informações através de diversas chamadas RPC. Muito freqüentemente, mapas são armazenados no formato DBM.

Os mapas em si são gerados a partir de arquivos textos mestres, como por exemplo os arquivos `/etc/hosts` ou `/etc/passwd`. Para alguns arquivos, diversos mapas são criados, um para cada tipo de chave. Por exemplo, pode-se pesquisar o arquivo `hosts` na busca por um nome de máquina ou por um endereço IP. Neste caso, dois mapas são gerados a partir deste arquivo, os mapas `hosts.byname` e `hosts.byaddr`, respectivamente. A tabela 10.1 lista os mapas mais comuns e os seus arquivos de origem.

Há outros arquivos e mapas que podem encontrar suporte em um ou outro pacote NIS e que podem conter informações para aplicações não discutidas neste livro, como o mapa `bootparams` que pode ser usado por alguns servidores BOOTP, DBM é uma biblioteca de gerenciamento de bases de dados que usa técnicas numéricas para acelerar as operações de pesquisa. Esta implementação livre do DBM no projeto GNU é chamada de `gdbm`, a qual é parte de muitas distribuições Linux..Arquivo Mestre Mapa(s) Mapa(s)

```
/etc/hosts hosts.byname hosts.byaddr
/etc/networks networks.byname networks.byaddr
/etc/passwd passwd.byname passwd.byuid
/etc/group group.byname group.bygid
/etc/services services.byname services.bynumber
/etc/rpc rpc.byname rpc.bynumber
/etc/protocols protocols.byname protocols.bynumber
/usr/lib/aliases mail.aliases
```

ou mapas que não têm atualmente qualquer função no Linux (como os mapas

`ethers.byname` e `ethers.byaddr`).

Para alguns mapas, as pessoas comumente utilizam nomes curtos, os quais são mais simples de serem memorizados e digitados. Para se obter uma lista completa dos nomes curtos conhecidos pelas ferramentas NIS, deve-se executar o seguinte comando:

```
$ ypcat -x
```

NIS map nickname translation table:

```
"passwd" -> "passwd.byname"
"group" -> "group.byname"
"networks" -> "networks.byaddr"
"hosts" -> "hosts.byname"
"protocols" -> "protocols.bynumber"
"services" -> "services.byname"
"aliases" -> "mail.aliases"
"ethers" -> "ethers.byname"
"rpc" -> "rpc.bynumber"
"netmasks" -> "netmasks.byaddr"
"publickey" -> "publickey.byname"
"netid" -> "netid.byname"
"passwd.adjunct" -> "passwd.adjunct.byname"
```

```
"group.adjunct" -> "group.adjunct.byname"  
"timezone" -> "timezone.byname"
```

O servidor NIS é tradicionalmente chamado de ypserv. Para uma rede média, um único servidor será suficiente, porém redes grandes podem executar diversos servidores em máquinas diferentes e em segmentos diferentes da rede permitindo maior segurança e balanceamento entre servidores e roteadores. Estes servidores são sincronizados definindo-se um como servidor mestre e os demais como servidores escravos. Mapas serão criados somente na máquina onde for executado o servidor mestre. A partir deste, eles serão distribuídos para todos os escravos. O leitor mais atento pode ter percebido que o termo rede foi colocado de forma muito vaga até aqui e isso se deve ao fato de NIS ter um conceito distinto para se referir a uma rede, ou seja, o conjunto de todas as máquinas que compartilham parte de suas informações e dados de configuração do sistema através do NIS, é o chamado: domínio NIS. Infelizmente, domínios NIS não têm absolutamente nada em comum com os domínios encontrados no DNS. A fim de evitar esta ambigüidade ao longo deste capítulo, sempre especificaremos o tipo de domínio a que se está referindo. Os domínios NIS têm exclusivamente uma função administrativa. São invisíveis para a maioria dos usuários, exceto para aqueles que compartilham senhas entre todas as máquinas do domínio. Desta forma, o nome dado a um domínio NIS é relevante somente para administradores. Normalmente, qualquer nome servirá, desde que ele seja diferente de qualquer outro nome de domínio NIS existente na rede local. Por exemplo, caso o administrador da rede da Cervejaria Virtual

resolva criar dois domínios NIS, um para a Cervejaria e outro para a Vinícola, eles podem chamar-se, por exemplo *cervejaria* e *vinicola*, respectivamente. Outro esquema comumente utilizado é o de simplificar o nome do domínio NIS, chamando-o somente de NIS. Para configurar e mostrar o nome do domínio NIS de uma máquina, deve-se utilizar o comando `domainname`. Ao ser acionado sem argumentos, ele imprime o nome do domínio NIS. Para configurar o nome do domínio deve-se executá-lo como superusuário e digitar-se:

```
# domainname cervejaria
```

Domínios NIS determinam que servidor NIS deverá ser pesquisado pela aplicação.

Por exemplo, o programa `login` em uma máquina da Vinícola pode, obviamente, somente pesquisar o servidor NIS da Vinícola (ou um deles, caso haja mais de um) para descobrir a informação de senha de um usuário, enquanto uma aplicação em uma máquina da Cervejaria deve pesquisar o servidor NIS da Cervejaria. Um mistério porém permanece sem solução: como os clientes descobrem a qual servidor eles devem se conectar. A abordagem mais simples poderia ser um arquivo de configuração que define o nome da máquina onde o servidor é executado. De qualquer forma, esta abordagem é pouco ²exível, porque não permite que os clientes usem diferentes servidores (para o mesmo domínio), dependendo de sua disponibilidade. De qualquer forma, implementações tradicionais do NIS baseiam-se em um servidor especial chamado `ybind` para detectar um servidor NIS adequado para o seu domínio NIS. Antes de estar apta a executar quaisquer pesquisas NIS, uma aplicação deve encontrar qual servidor `ybind` pode ser usado. O servidor `ybind` testa os servidores através da propagação na rede local. O primeiro servidor que responder é assumido como sendo o potencialmente mais rápido e será usado nas pesquisas NIS subsequentes. Após um certo intervalo ou se o servidor se tornar indisponível, `ybind` irá repetir o teste para servidores ativos novamente. Agora o ponto de discórdia sobre a definição dinâmica do servidor NIS reside no fato de ela ser raramente utilizada e que introduz um problema sério de segurança: `ybind` cegamente acredita em qualquer máquina que responda, o qual pode ser um perfeito servidor NIS, assim como um intruso mal

intencionado. Desnecessário dizer que isto se torna especialmente problemático quando se administram bases de dados de senhas sobre NIS. Para proteger-se disso, NIS não usa `yplib` por padrão, mas escolhe o nome da máquina servidora a partir de um arquivo de `con.guracão`.

18.2 O Cliente NIS

Caso se esteja familiarizado no desenvolvimento ou porte de aplicações de rede, pode-se notar que os mapas NIS listados acima correspondem a funções de uma biblioteca C. Por exemplo, para obter informações a partir do arquivo `passwd`, geralmente são utilizadas as funções `getpwnam(3)` e `getpwuid(3)`, as quais retornam informações sobre a conta associada a determinado nome de usuário ou a uma identificação numérica, respectivamente. Sob condições normais, estas funções executam uma pesquisa no arquivo padrão, no caso o `/etc/passwd`.

Uma aplicação NIS que necessite destas funções, irá modificar seu comportamento, e colocar uma chamada RPC para fazer com que o servidor NIS execute as pesquisas de nomes de usuários ou identifi.cações. Isso ocorre de maneira totalmente transparente para a aplicação. A função pode ou anexar o mapa NIS ou substituir o arquivo original por ele. Obviamente, as alterações não são realizadas diretamente no arquivo original, porém para a aplicação elas aparentam terem sido.

Para implementações NIS tradicionais, há certas convenções a serem usadas, assim como na substituição de mapas ou sobre aqueles que foram anexados à informação original. Alguns mapas, como o `passwd`, requerem modi.cações no arquivo `passwd`, as quais se forem realizadas de forma errônea, podem gerar problemas de segurança. Para evitar estes problemas, NYS usa um sistema geral de configuração, que determina se um determinado conjunto de funções cliente usa os arquivos originais, mapas NIS ou NIS + e em qual ordem. Ele será descrito em maiores detalhes em uma seção posterior.

18.3 Servidor NIS

Após tanta teoria tecno-babel, é tempo de *_sujar as mãos_* com algum trabalho de `con.guracão`. Nesta seção, cobriremos a `con.guracão` de um servidor NIS. Caso se estejam executando somente testes com o servidor, esteja certo de não configurar um nome de domínio NIS que já esteja em uso na rede. Isso pode corromper todos os serviços de rede e provocar tristeza e ira em diversas pessoas.

Há atualmente dois servidores NIS de livre distribuição disponíveis para Linux, um deles no pacote `ypserv` de Peter Eriksson. Não importa qual será executado, independente de se usar NYS ou o NIS padrão cujo código cliente utiliza a biblioteca `libc`. Quando este livro foi escrito, o código para o gerenciamento do servidores escravos NIS parecia ser mais completo em `yps`. Então, caso se tenha a possibilidade de lidar com diversos servidores escravos, `yps` poderá ser uma sábia escolha.

Após instalar o programa servidor (ypserv) no diretório /usr/sbin, deve-se criar o diretório que conterá os arquivos de mapas que serão distribuídos. Ao configurar o domínio NIS para o domínio cervejaria, os mapas irão para o diretório /var/yp/cervejaria. O servidor determinará se está servindo a um domínio NIS em particular ao checar se o diretório de mapas está presente. Caso o serviço não esteja habilitado para algum domínio NIS, deve-se estar seguro de que o diretório foi removido.

Mapas são atualmente armazenados em arquivos DBM para agilizar as pesquisas. Eles são criados a partir de arquivos mestres usando um programa chamado makedbm (no servidor de Tobias) ou dbmload (no servidor de Peter). Eles não podem ser intercambiados. Transformar um arquivo mestre em um formato utilizável pelo programa dbmload normalmente requer alguma mágica dos utilitários awk ou sed, o qual tende a ser um pouco tedioso para digitar e difícil de lembrar.

De qualquer forma, o pacote ypserv de Peter Eriksson contém um arquivo Makefile (chamado ypMakefile) que executará todas as tarefas necessárias. Deve-se instalar o Makefile no mapa de diretórios e editá-lo para re²etir os mapas que se deseja distribuir. A partir do topo do arquivo, pode se encontrar o parâmetro all que lista os serviços que o servidor ypserv oferece. Por padrão, a linha terá a seguinte aparência:

```
all: ethers hosts networks protocols rpc services passwd group netid
```

Caso não se deseje produzir os mapas ethers.byname e ethers.byaddr, por exemplo, basta remover os pré-requisitos ethers para esta regra. Para testar a configuração, deve-se iniciar com somente um ou dois mapas, como por exemplo com os mapas services.*.

Após editar o Makefile, enquanto ele já esteja no diretório de mapas, basta digitar `_make_`. Este procedimento irá gerar automaticamente os mapas e instalá-los. Deve-se estar seguro de atualizar os mapas toda vez que os mestres forem alterados, caso contrário eles permanecerão invisíveis para o restante da rede. A próxima seção explica como configurar o cliente NIS. Caso a configuração não funcione, deve-se tentar descobrir se alguma requisição foi recebida do servidor ou não. Caso se especifique o indicador `-D` ou `-debug` na linha de comando do servidor NYS, ele apresentará uma série de mensagens informativas na console sobre todas as pesquisas NIS recebidas e o resultado retornado. Certamente isso será de extrema utilidade na busca da causa de um problema. O servidor de Tobias não tem tal opção disponível.

10.5 Segurança em Um Servidor NIS

O uso do NIS tem aspectos de segurança bastante delicados: ele pode deixar o arquivo de senhas da rede acessível virtualmente a todos os usuários da rede local e a outras que possam estar interconectadas, o que pode facilitar o acesso a um grande número de intrusos. Assim que um intruso souber o nome do domínio NIS e o endereço do servidor, ele pode simplesmente enviar uma requisição do mapa passwd.byname e instantaneamente receber todas as senhas do sistema. Com um programa de quebra de senhas ágil e um bom dicionário, descobrir a senha de alguns usuários do sistema não será problema. Devido a isso, foram criadas as opções conhecidas como securenets. Elas restringem o acesso ao servidor NIS a somente algumas máquinas, baseado no seu endereço IP ou nos seus números de rede. A última versão do ypserv implementa esta funcionalidade de uma forma bastante simples, utilizando os arquivos etc/hosts.allow e etc/hosts.deny, descrito no capítulo

9. Por exemplo, para restringir o acesso ao servidor NIS somente às máquinas da rede da Cervejaria Virtual, o administrador de redes deverá informar o seguinte no arquivo `hosts.allow`: `ypserv: 172.16.2`

Isso faz com que todas as máquinas na rede IP 172.16.2.0 tenham acesso ao servidor NIS. Para evitar que qualquer outra máquina acesse o servidor NIS deve-se incluir a seguinte linha no arquivo `hosts.deny`:

```
ypserv: all
```

Números IP não são a única forma de especificar máquinas ou redes nos arquivos `hosts.allow` e `hosts.deny`. Verifique na página de manual de `hosts.access(5)` para maiores detalhes. De qualquer forma é importante saber que não é possível usar nomes de máquinas ou domínios como uma entrada de `ypserv`. Caso seja especificado um nome de máquina, o servidor tentará resolver o endereço IP, mas o resolvedor chama o servidor NYS, provocando um círculo vicioso sem fim. Pode-se ainda utilizar um programa `portmaster` seguro ao invés das opções

`securenets`. O `portmap-3.0` também utiliza o arquivo `hosts.allow`, disponibilizando-o para todos os servidores RPC e não somente para `ypserv`. Não se deve utilizar os dois sistemas de segurança ao mesmo tempo, pois este procedimento causaria uma sobrecarga nos processos de autorização.

18.4 Configurando um Cliente NIS com NYS

O primeiro passo para a configuração de um cliente NYS é informar qual o servidor NIS que deve ser utilizado para os serviços NIS, configurando o arquivo de configuração `/etc/yp.conf`. Uma entrada muito simples para a máquina da rede da Vinícola teria a seguinte aparência:

```
# yp.conf - configuração YP para a biblioteca NYS. #
domainname vinicola
server garibaldi
```

O primeiro comando indica a todos os clientes NIS que eles pertencem ao domínio NIS chamado `vinicola`. Caso esta linha seja omitida, NYS irá utilizar o nome de domínio definido no sistema através do comando `domainname`. O comando `server` no arquivo acima define o nome do servidor NIS a ser utilizado. Obviamente, o endereço IP de `garibaldi` deve estar configurado no arquivo `hosts`. Alternativamente pode-se usar o endereço IP no comando `server`. Na forma mostrada acima, o comando `server` indica ao NYS qual o nome do servidor a ser usado, independente do conteúdo do campo `nome de domínio`. Caso Disponível via FTP anônimo em `metalab.unc.edu` sob o diretório `Linux/systems/Network`.haja mudanças frequentes entre diferentes domínios NIS para a mesma máquina pode-se desejar manter a informação de diversos domínios no arquivo `yp.conf`. Para tanto é necessário adicionar o nome do domínio ao comando `server`. Por exemplo, pode-se alterar o exemplo acima para um equipamento portátil, que teria o seguinte conteúdo:

```
# yp.conf - YP configuração para biblioteca NYS
#
```

```
server garibaldi vinicola
server aracaju cervejaria
```

Isto permite que o portátil possa ser conectado a qualquer um dos domínios NIS, simplesmente alterando-se o nome do domínio através do comando `domainname` em tempo de inicialização do sistema. Após a criação de arquivos de configuração básica e assegurando-se de que eles

estejam acessíveis a todos os usuários deve-se executar os primeiros testes, a fim de verificar se a máquina consegue conectar-se com o servidor NIS. Deve-se escolher qualquer mapa que seja realmente distribuído pelo servidor, como por exemplo o `hosts.byname` e tentar recuperá-lo através do uso do utilitário `ypcat`, que assim como todas as outras ferramentas administrativas NIS, deverá estar no diretório `/usr/sbin`.

```
# ypcat hosts.byname
191.72.2.2 caxias caxias.vinicola.com.br
191.72.2.3 gramado gramado.vinicola.com.br
191.72.1.1 aracaju aracaju.cvirtual.com.br
191.72.2.1 aracaju aracaju.vinicola.com.br
191.72.1.2 maceio maceio.cvirtual.com.br
191.72.1.3 jessoa jessoa.cvirtual.com.br
191.72.2.4 garibaldi garibaldi.cvirtual.com.br
```

A saída obtida deverá ter uma aparência similar à lista acima apresentada. Caso se obtenha uma mensagem de erro como por exemplo `_ _` ou algo similar, as causas podem ser: ou o nome do servidor NIS do domínio definido em `yp.conf` não existe, ou o servidor não pode ser alcançado por alguma razão. Neste caso, deve ser executado um comando `ping` para as máquinas envolvidas, a fim de verificar as condições de conectividade com o servidor NIS e caso a conexão esteja funcionando corretamente deve-se checar se o servidor NIS está ativo. Isto pode ser feito através do comando `rpcinfo`, o qual deve produzir a seguinte saída:

```
# rpcinfo -u
servidor ypserv programa 100004 versão 2 pronto e aguardando
```

18.5 Escolhendo os Mapas Corretos

Estando-se seguro de que a comunicação com o servidor NIS está em perfeitas condições, deve-se decidir quais arquivos de configuração serão substituídos ou incrementados com os mapas NIS. Comumente, utilizam-se os arquivos NIS para funções de pesquisas de máquinas e senhas. O primeiro é especialmente útil caso não se execute o BIND. O último permite que todos os usuários acessem qualquer máquina do domínio utilizando a mesma conta e senha. Isto normalmente exige que o diretório pessoal do usuário esteja localizado de forma central e seja compartilhado por todas as máquinas via NFS. Isso é explicado de forma detalhada na seção 10.8 a seguir. Outros mapas, como `services.byname`, não provocam um ganho tão visível, mas economizam a edição de alguns arquivos, caso se tenha instalado qualquer aplicação de rede que use o nome do serviço que não esteja no arquivo padrão `services`.

Geralmente, é desejável ter-se alguma liberdade de escolha quando uma função de pesquisa deve utilizar arquivos locais e quando deve utilizar o servidor NIS. NYS permite a con.guração da ordem em que uma função acessa estes serviços. Isso é controlado através do arquivo `/etc/nsswitch.conf`, que significa Troca de Serviços de Nomes, o qual obviamente não está limitado somente a serviços de nome. Para qualquer uma das funções de pesquisa suportadas pelo NYS, ele conterá uma linha denominando os serviços a serem utilizados.

A ordem correta dos serviços depende do tipo de dados. Porém não é como se as entradas contidas no mapa `services.byname` contêm entradas diferentes do arquivo `services local`, na verdade ele somente contém mais dados. Então uma boa opção pode residir em selecionar os arquivos locais inicialmente e checar os mapas NIS somente se o nome do serviço não for encontrado. As informações de nomes de máquinas, por outro lado, podem mudar freqüentemente; por outro lado DNS ou o servidor NIS deve ter sempre as informações mais atualizadas, enquanto o arquivo `localhosts` é mantido somente como uma cópia de segurança caso DNS e NIS falhem. Neste caso, os arquivos locais devem ser checados por último. O exemplo abaixo mostra como configurar as funções `gethostbyname(2)`, `Name Service Switch`.

`gethostbyaddr(2)` e `getservbyname(2)` conforme descrito acima. Elas tentarão os serviços listados na ordem apresentada, onde caso uma pesquisa seja bem sucedida, o resultado será informado, caso contrário o próximo serviço será testado.

```
# pequeno exemplo do arquivo /etc/nsswitch.conf
#
hosts: nis dns files
services: files nis
```

A lista completa de serviços pode ser definida como uma entrada no arquivo `nsswitch.conf` mostrado. Os mapas correntes, arquivos, servidores e objetos podem ser pesquisados dependendo do nome da entrada. `nisplus` ou `nis+` Usa o servidor NIS + para este domínio. A localização do ser-vidor é obtida no arquivo `/etc/nis.conf.nis` Usa o servidor NIS atual deste domínio. A localização do servidor pesquisado é configurado no arquivo `yp.conf`, conforme apresentado na seção anterior. Para as entradas em `hosts`, os mapas `hosts.byname` e `hosts.byaddr` serão pesquisados.

`dns` Usa o servidor de nomes DNS. Este tipo de serviço é somente útil com uma entrada no arquivo `hosts`. As pesquisas de servidores de nomes são ainda determinadas pelo arquivo padrão `resolv.conf`.

`files` Usa o arquivo local, como o arquivo `/etc/hosts` com entrada de `hosts`.

`dbm` Pesquisa os arquivos DBM localizados em `/var/dbm`. O nome usado pelo arquivo é o correspondente no mapa NIS.

Atualmente, NYS suporta as seguintes entradas no arquivo `nsswitch.conf`: `hosts`, `networks`, `passwd`, `group`, `shadow`, `gshadow`, `services`, `protocols`, `rpc` e `ethers`. Outras entradas similares podem ser adicionadas.

A descrição abaixo apresenta um exemplo completo, que introduz uma outra funcionalidade no arquivo `nsswitch.conf`: o parâmetro `[NOTFOUND=return]` no parâmetro `hosts` informa ao NYS para retornar caso o item desejado não seja encontrado na base de

dados do NIS ou DNS. Ou seja, NYS irá continuar a pesquisa nos arquivos locais somente se o acionamento do NIS ou do servidor DNS falhar por algum motivo. Neste caso os arquivos locais podem ser usados em tempo de inicialização e como uma cópia de segurança quando o servidor NIS estiver inativo

```
Exemplo do arquivo nsswitch.conf:  
#/etc/nsswitch.conf  
#  
hosts: nis dns [NOTFOUND=return] files  
networks: nis [NOTFOUND=return] files  
services: files nis  
protocols: files nis  
rpc: files nis
```

18.6 Usando os Mapas passwd e group

Uma das maiores aplicações do NIS é a sincronização de usuários e informações de contas em todas as máquinas de um domínio NIS. Portanto, normalmente mantêm-se somente um pequeno arquivo `/etc/passwd` local, no qual as informações de todo o domínio são anexadas a partir dos mapas NIS. De qualquer forma, a simples habilitação de pesquisas NIS para este serviço no arquivo `nsswitch.conf` pode não ser suficiente.

Ao basear-se nas informações de senhas distribuídas pelo NIS, deve-se estar seguro de que a identificação numérica de um usuário de qualquer conta do arquivo `passwd` local não coincida com a identificação numérica dos usuários do servidor NIS. Pode-se utilizar estas facilidades para outros propósitos, como a montagem de volumes NFS de outras máquinas na rede local.

Caso as identificações numéricas nos arquivos `/etc/passwd` ou `/etc/group` tenha alguma inconsistência com os mapas NIS, isso provocará a necessidade de ajustes de propriedade de todos os arquivos de usuários com identificação numérica duplamente referenciada em ambos os arquivos. Inicialmente deve-se alterar as identificações numéricas de usuários e grupos nos arquivos `passwd` e `group` para novos valores, em segundo lugar deve-se localizar todos os arquivos pertencentes ao usuário recém-alterado, e finalmente alterar a propriedade para a nova identificação. Assumindo que o usuário `news` tenha uma identificação numérica igual a

9 e o usuário `sandro` tenha uma identificação igual a 103, a qual foi mudada para outro valor, devem então ser empregados os seguintes comandos:

```
# find / -uid 9 -print >/tmp/uid.9  
# find / -uid 103 -print >/tmp/uid.103  
# cat /tmp/uid.9 | xargs chown news  
# cat /tmp/uid.103 | xargs chown sandro
```

É importante que estes comandos sejam executados com o novo arquivo `passwd` instalado e que sejam identificados todos os nomes de arquivos antes de se alterar a propriedade de qualquer um deles. Para atualizar os grupos proprietários dos arquivos deve-se usar um comando similar.

Após este procedimento, a identificação numérica dos usuários e grupos do sistema deverá estar em acordo com os demais sistemas do domínio NIS. O próximo passo a ser executado, será a adição das linhas de configuração do arquivo `nsswitch.conf` que habilita pesquisas NIS para informações de usuários e grupos:

```
# /etc/nsswitch.conf - tratamento de passwd e group
passwd: nis files
group: nis files
```

Isso habilita a pesquisa em primeiro lugar nos mapas NIS dos comandos `login` e todos os seus amigos, ao tentar acessar o sistema e caso a pesquisa falhe, indica a utilização dos arquivos locais. Normalmente praticamente todos os usuários serão removidos dos arquivos locais e somente as entradas para o superusuário `root` e contas genéricas como `mail` estarão presentes. Isso se deve ao fato de que tarefas vitais do sistema podem requerer o mapeamento de identificações numéricas para nomes de usuários e vice-versa. Por exemplo, tarefas administrativas do programa `cron` devem ser executadas pelo comando `su` para temporariamente tornar-se o usuário `news`, ou o subsistema `UUCP` pode gerar um relatório de condição. Caso os usuários `news` e `uucp` não tenham entradas no arquivo `localpasswd`, estas tarefas irão falhar lamentavelmente caso o NIS não esteja ativo.

Há dois importantes aspectos de tudo o que foi apresentado até aqui: por um lado, a configuração conforme descrita neste guia funciona somente para conjuntos de programas que não utilizem senhas sombra, como aqueles incluídos no pacote `util-linux`. Os detalhes de se utilizar senhas sombra com NIS serão mostrados a seguir. Por outro lado, os comandos de acesso não são os únicos a utilizarem o arquivo `passwd` — por exemplo o comando `ls` utilizado constantemente por um grande número de usuários. Sempre que seja construída uma longa lista, `ls` irá mostrar os nomes simbólicos dos usuários e grupos proprietários de arquivos, os

quais são na verdade o que o comando encontrou para cada identificação numérica de usuário e grupo no servidor NIS na primeira pesquisa efetuada. Isso irá tornar o processamento terrivelmente mais lento caso a rede local esteja congestionada, ou ainda pior, quando o servidor NIS não estiver na mesma rede física, fazendo com que os datagramas tenham que passar através de um roteador. Bem, esta ainda não é a história completa. Imaginemos o que acontece quando um usuário muda a sua senha. Normalmente, ele aciona o comando `passwd`, o qual receberá a nova senha e atualizará o arquivo `passwd` local. Isso é impossível com

o NIS, uma vez que na verdade ele não está disponível localmente, porém fazer com que os usuários acessem diretamente o servidor NIS pode não ser uma opção aceitável. Desta forma, NIS provê um substituto ao programa `passwd` chamado `yppasswd`, o qual faz um trabalho análogo na presença do NIS. Para alterar a senha na máquina servidora, deve-se contactar o servidor `yppasswd` via RPC e prover as informações da senha alterada. Normalmente deve-se instalar o programa `yppasswd` sobre o programa `passwd` normal da seguinte forma:

```
# cd /bin
# mv passwd passwd.old
# ln yppasswd passwd
```

Simultaneamente deve-se instalar o programa `rpc.yppasswd` no servidor e iniciá-lo a partir do `rc.inet2`. Isto irá efetivamente esconder todos os detalhes de tratamento de senhas e usuários pelo NIS.

18.7 Utilizando NIS com Suporte a Senhas Sombra

Ainda não há suporte a senhas sombra no NIS. John F. Haugh, o autor do conjunto de softwares de senhas sombra, liberou uma versão da biblioteca de funções sombra através da licença GPL no grupo comp.sources.misc. Há algum suporte a NIS, mas ainda está incompleto e os arquivos não podem ser adicionados à biblioteca padrão C. Por outro lado, a publicação de informações do arquivo `/etc/shadow` via NIS é uma forma de de.ciência aos propósitos das ferramentas de senhas sombra. Apesar das funções de pesquisas de senhas NYS não utilizarem o mapa `shadow.by name` ou qualquer outro similar, NYS suporta esta sistemática usando um arquivo `/etc/shadow` de forma transparente. Quando a implementação NYS de `getpwnam` é acionada para pesquisar as informações especificadas em um nome de acesso fornecido, as facilidades especificadas pela entrada `passwd` no arquivo `nsswitch.conf` são utilizadas.

O serviço `nis` irá simplesmente pesquisar o nome no mapa `passwd.byname` no servidor NIS. O serviço `files` irá verificar se o arquivo `/etc/shadow` está presente e em caso positivo, tentará abri-lo. Caso não esteja, ou caso não tenha privilégios de `root`, usará o comportamento tradicional de pesquisar somente o arquivo `/etc/passwd`. De qualquer forma, caso o arquivo `shadow` exista e possa ser aberto, NYS irá extrair a senha de usuário do arquivo `shadow`. A função `getpwuid` é implementada de forma similar. Desta forma, os binários compilados com NYS irão lidar com um conjunto de ferramentas de senhas sombra de forma transparente.

18.8 Utilizando o Tradicional Código NIS

Caso se esteja utilizando o código tradicional que pode ser encontrado na `libc`, a con.guração de um cliente NIS pode ser um pouco diferente. Por um lado, pode-se usar um servidor `ybind` para propagar os servidores ativos, ao invés de divulgar isso a partir de um arquivo de con.guração. Deve-se estar seguro de que o programa `ybind` será acionado na inicialização do sistema. Ele deve ser acionado após o domínio NIS ter sido configurado e o `portmapper` RPC ter sido inicializado. Acionando-se o `ypcat` para testar o servidor pode funcionar conforme o mostrado anteriormente.

Recentemente, tem havido algumas informações de que o NIS pode falhar com a seguinte mensagem de erro `_ clntudp_create: RPC: falha de portmapper - RPC: incapaz de receber_`. Isso se deve a uma incompatibilidade na forma como `ybind` comunica-se com as informações de construção das funções da biblioteca. Neste caso deve-se utilizar uma versão atualizada do NIS e recompilá-la para que o problema seja solucionado.

Adicionalmente cabe citar que a forma como o NIS decide se deve mesclar as informações NIS com os arquivos locais é distinta daquela utilizada pelo NYS. Por exemplo, para utilizar os mapas de senhas NIS, deve-se incluir a seguinte linha no mapa `/etc/passwd`:

```
+*:0:0:::
```


Isto assinala o local onde as funções de pesquisa de senhas `_inserir_` os mapas NIS. Deve-se inserir uma linha similar (menos as últimas duas colunas) em um 7

O fonte do yp-linux pode ser obtido em [ftp.uni-paderborn.de](ftp://ftp.uni-paderborn.de) no diretório `/pub/Linux/LOCAL`

arquivo `/etc/group` para se obter o mesmo resultado. Para usar os `mapashosts.*` distribuídos pelo NIS, deve-se mudar a linha `order` no arquivo `host.conf`. Por exemplo, caso se deseje usar NIS, DNS e o arquivo `/etc/hosts` (nesta ordem) será necessária a seguinte alteração na linha para `order yp bind hosts`

A implementação tradicional do NIS não suporta outros mapas no momento.

19 - NFS

19.1 Introdução e Conceitos

A maior vantagem do uso de redes de computadores é o compartilhamento de informações e recursos. Na verdade, este é o próprio propósito de se utilizar uma rede. O compartilhamento de informações pode ser feito de diversas maneiras, entre elas está o uso de FTP, web, etc.. Já o compartilhamento de recursos por vários computadores em uma rede, como discos, pode ser realizado através do NFS.

O NFS é um acrônimo para *Network File System*, ou, em português, Sistema de Arquivos de Rede. O NFS foi criado para permitir o acesso transparente a discos remotos. Ele também permite uma maior centralização da administração de discos, pois é possível ter diretórios em uma única máquina (o servidor NFS) e compartilhados em todos os sistemas conectados à rede. Além disso, o NFS abre a possibilidade de existirem clientes sem disco.

Um esquema NFS bem configurado será totalmente transparente ao usuário. Não será relevante para o usuário saber em qual servidor um diretório realmente está. Basta que seus programas funcionem corretamente sem que ele tenha de realizar qualquer tipo de configuração especial.

19.2 Instalando o NFS

19.2.1 Instale o servidor NFS

1. Acesse o CD 1 do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS/  
#
```

2. Instale os pacotes do NFS:

```
# rpm -ivh nfs-  
nfs-server #####  
nfs-utils #####  
#
```

19.3 Configurando o Servidor NFS

A configuração de um servidor NFS é muito simples e pode ser feita tanto com o Linuxconf quanto através da edição de arquivos de configuração. Na verdade, só o que você terá de fazer é definir os diretórios a serem exportados. Para configurar o seu servidor, entre

no Linuxconf e dirija-se ao menu Ambiente de Rede ! Tarefas de Servidor ! NFS para que a tela de configuração lhe seja mostrada.

Nesta tela você pode definir um caminho a ser exportado. As opções disponíveis são:

Rota para exportar: aqui você define o diretório a ser exportado. Este diretório poderá ser montado por um cliente.

Comentário: apenas um comentário ilustrativo. Pode ser usado para documentar alguma observação sobre o diretório.

Nome do cliente: neste campo você pode definir quais máquinas-clientes (separadas por vírgula) poderão acessar este diretório. Se nenhum cliente for especificado, então qualquer máquina poderá conectar-se. Você pode ainda utilizar coringas para definir as máquinas-clientes. Por exemplo, se você quer dar acesso a todas as máquinas de seu domínio, você pode especificar algo como: *.minhaorganizacao.com.br

Pode escrever: indica se o diretório será exportado apenas para leitura ou se será possível para os usuários gravarem nele.

Privilégios de superusuário: normalmente, o superusuário acessa diretórios remotos com privilégios de `nobody`. Você pode especificar esta opção para que o superusuário continue tendo acesso de superusuário quando acessar o diretório.

Seguir links simbólicos: quando esta opção está ativa, os links simbólicos absolutos (aqueles iniciados por /) são convertidos em links relativos, de forma a atingir a raiz do servidor.

Requisitar acesso da porta segura: você pode especificar que apenas conexões seguras (com número baixo de porta de origem) possam montar o diretório. Como mencionado anteriormente, a configuração pode também ser feita através da edição de arquivos de configuração. Esta edição é bastante simples, envolvendo apenas a edição do arquivo `/etc/exports`

O arquivo `/etc/exports` é bastante simples. Cada linha representa um diretório exportado e as informações são separadas por espaços. Por exemplo:

```
/share kepler(rw) copernico
```

Este exemplo exporta o diretório `/share` especificando duas máquinas (kepler e copernico) com opções diferentes (kepler pode gravar no diretório, enquanto que copernico só pode ler).

<code>/etc/exports</code>	Linuxconf
<code>rw</code>	"pode escrever"
<code>no_root_squash</code>	Marcar "privilégios de superusuário"
<code>link_relative</code>	Marcar "seguir links simbólicos"
<code>insecure</code>	Desmarcar "requisitar acesso da porta segura"

Após feitas as configurações, você deve iniciar o serviço **nfs**. Para fazê-lo, é necessário abrir um terminal e digitar o comando (o serviço portmap já deverá estar rodando):

```
# /etc/rc.d/init.d/nfs start
Iniciando serviços NFS : [ OK ]
Iniciando quotas (NFS) [ OK ]
Iniciando mountd (NFS) [ OK ]
Iniciando rpc.nfsd (NFS) [ OK ]
```

19.4 Configurando um cliente NFS

É muito simples fazer uma máquina utilizar um disco remoto. Na verdade, não é necessário qualquer tipo de configuração para tornar uma máquina um cliente NFS. Você só precisa montar o diretório remoto da mesma maneira como monta um diretório local.

1. Antes de mais nada, crie um diretório na máquina local. Vamos montar o diretório remoto neste diretório local. Em nosso exemplo, montaremos o diretório `/usr/local` da máquina asterix dentro do diretório local `/mnt/asterix`

```
# mkdir /mnt/asterix #
```

2. Agora monte o diretório remoto com o comando **mount**.

```
# mount asterix:/usr/local /mnt/asterix
#
```

Feito isso, você poderá perceber que o conteúdo do diretório local `/mnt/asterix` é o mesmo do diretório `/usr/local` do servidor asterix.

É claro que, em muitos casos, você desejará que os diretórios remotos sejam automaticamente montados quando da inicialização da máquina-cliente. Para fazê-lo, você deve utilizar o Linuxconf e entrar em Sistemas de Arquivos ! Acessar volumes NFS ! Adicionar para que a tela de adição de volumes NFS lhe seja mostrada (Figura 5-2).

20 - Samba

O SMB é o protocolo utilizado em redes Windows® para compartilhamento de recursos, como impressoras e discos. É através do SMB que o Windows® permite que uma máquina acesse o disco e a impressora de outra na rede.

Com o crescimento do Linux, tornou-se necessário fazer com que máquinas Windows® e as redes Unix trabalhassem de maneira harmoniosa. Infelizmente, isso não era muito simples, já que as duas plataformas vinham de culturas muito diferentes e tinham dificuldades em trabalhar de maneira conjunta sem o auxílio do *Samba*.

O Samba, então, é uma implementação livre do protocolo SMB. Com o Samba será possível simular, de maneira transparente, um servidor Windows®. Isso torna possível o uso de estações Windows® em uma rede Conectiva Linux utilizando o protocolo NetBIOS.

20.1 Configurando o Servidor Samba

A configuração de um servidor Samba é um tópico bastante abrangente. Tudo depende do tipo de serviços que você deseja executar. Por exemplo, você deve decidir se deseja que o servidor Samba seja um controlador de domínio ou não.

20.2 Instalando o Samba

A instalação do Samba é simples:

1. Acesse o CD 1 da distribuição do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS
```

2. Instale o pacote do Samba:

```
# rpm -ivh samba-*
```

```
samba #####
```

```
samba-clients #####
```

20.3 Configuração

Esta seção concentrar-se-á na configuração do Samba através do Linuxconf. Porém, existem outras maneiras de fazê-la, como através da edição do arquivo `/etc/smb.conf` e, ainda, através do SWAT. Cobriremos o SWAT no final desta seção. Os exemplos de configuração serão sempre sobre o Linuxconf, não sendo necessária qualquer edição direta do arquivo `etc/smb.conf`.

20.3.1 Senhas Criptografadas

O problema mais comum durante a instalação do Samba é relacionado com as senhas dos usuários.

A partir da versão 98, o Windows® começou a fazer a transmissão de senhas criptografadas

pela rede NetBIOS. O Windows 95® e as versões iniciais do Windows NT®, porém, transmitiam sem encriptar as senhas. Essa falta de padronização entre estes sistemas torna a configuração do Samba um tanto mais complexa.

A encriptação de senhas visa proteger as mesmas de tentativas primitivas de descoberta.

Infelizmente, os algoritmos de encriptação utilizados não são particularmente fortes.

Senhas Descritografadas no Windows® 95.

O padrão do Windows® 95 é não encriptar as senhas. Assim, é provável que esta configuração não seja necessária para você. Porém, versões mais recentes do Windows® 95 (como, por exemplo, o Windows® 95 OSR2) mudaram seu comportamento.

Se você tiver problemas com senhas, siga o procedimento descrito nesta seção.

A configuração do Windows 95® pode ser feita de duas maneiras. Uma delas seria a utilização do arquivo Win95_PlainPassword.reg que acompanha o Samba.

Este arquivo pode ser localizado no diretório docs/ da documentação *on-line* do Samba:

```
# cd /usr/doc/samba*/docs
```

Você pode copiá-lo para as estações Windows 95®. Depois, basta abrir o arquivo (dar dois cliques sobre o mesmo) para que ele seja instalado.

Outra maneira de habilitar senhas é editar o registro através do utilitário **regedit** do Windows® para alterar ou incluir a chave:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

Senhas Descritografadas no Windows® 98

Como padrão, o Windows® 98 encripta todas as senhas que trafegam pela rede. Para poder utilizar estações Windows® 98 com sua rede, siga o procedimento descrito nesta seção.

A configuração de senhas não encriptadas no Windows 98® pode ser feita de duas maneiras. Uma delas seria a utilização do arquivo Win98_PlainPassword.reg que acompanha o Samba. Este arquivo pode ser localizado no diretório docs/ da documentação *on-line* do Samba:

```
# cd /usr/doc/samba*/docs
```

Você pode copiá-lo para as estações Windows 98®. Depois, basta abrir o arquivo (dar dois cliques sobre o mesmo) para que ele seja instalado.

Outra maneira de habilitar senhas é editar o registro através do utilitário **regedit** do Windows® para alterar ou incluir a chave:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

Senhas Descritografadas no Windows NT®

O Windows® NT encripta senhas em algumas versões e não encripta em outras. Notadamente, após o terceiro pacote de consertos (*Service Pack 3, ou SP3*), ele passou a encriptar todas as senhas trafegadas pela rede. Assim, recomenda-se seguir os procedimentos descritos nesta seção para assegurar que ele funcione como cliente de uma rede Samba. Além disso, recomendamos que você tenha, pelo menos, o Windows® NT SP3 instalado.

A configuração de senhas não encriptadas no Windows® 98 pode ser feita de duas maneiras. Uma delas seria a utilização do arquivo NT4_PlainPassword.reg que acompanha o Samba. Este arquivo pode ser localizado no diretório docs/ da documentação *online* do Samba:

```
# cd /usr/doc/samba*/docs
```

Você pode copiá-lo para as estações Windows® NT. Depois, basta abrir o arquivo (dar dois cliques sobre o mesmo) para que ele seja instalado. Outra maneira de habilitar senhas é editar o registro através do utilitário **regedit** do Windows® para alterar ou incluir a chave:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001
```

20.3.2 Configurações Básicas

Uma das primeiras configurações que você provavelmente desejará fazer é definir o grupo de trabalho. Para isso, abra o Linuxconf e vá para as Tarefas de Servidor no Ambiente de Rede e entre em Samba para começar a configuração. Neste momento, você deve ver uma tela semelhante à Figura 5-3.

Clique em Padrões para configurar o grupo de trabalho.

No campo Grupo de Trabalho, digite o nome desejado.

Como o Windows® normalmente utiliza o método de segurança por recurso, você deve alterar o comportamento padrão do Samba, que é o de fazer a autenticação por usuário. Para fazer isso, apenas altere a opção Modo de Autenticação para Compartilhar. Esta opção pode ser encontrada na pasta Senhas.

20.3.3 Compartilhando um diretório

Nesta seção, vamos mostrar como compartilhar o diretório /home/samba para que o mesmo possa ser montado em uma estação Windows®.

1. Abra a caixa de diálogo Compartilhamento de Disco (Figura 5-5).
Compartilhando um diretório através do Samba

Note que estamos compartilhando o diretório /home/samba, com o nome de compartilhamento (aquele que aparecerá no Ambiente de Redes do Windows®) de “publico”. Permitimos, também, que o usuário, qualquer usuário, possa gravar no diretório. Além disso, definimos que apenas máquinas em nosso domínio possam acessá-lo.

Para compartilhar um diretório, você só precisa especificar um nome de compartilhamento,
o diretório a ser exportado e definir opções de acesso simples.

20.4 Montando um volume Samba

Você pode ter a necessidade de acessar um volume de uma máquina Windows® a partir do seu Conectiva Linux. Fazer isto não só é possível como é muito simples. O processo deve ser feito através da linha de comando. Imaginando que você queira montar o disco C: da máquina copernico, que foi compartilhado com o nome C, em seu diretório Conectiva Linux /mnt/copernico:

```
# smbmount //copernico/c /mnt/copernico  
#
```

Note que em alguns casos (você está montando um volume de um servidor Windows NT®, por exemplo), será necessário que você informe um usuário e uma senha para poder utilizar o comando **smbmount**.

```
# smbmount //copernico/c /mnt/copernico -o username=mara,password=selva
```

Isso funcionará se o usuário mara estiver cadastrado na máquina jaguarica com a senha selva.

20.5 Iniciando o Samba

Para que o seu servidor Conectiva Linux possa funcionar como um servidor Samba, o serviço **smbd** deve ser inicializado. Para tanto, abra um terminal e digite:

```
# /etc/rc.d/init.d/smb start  
Iniciando serviços SMB : [ OK ]  
Iniciando serviços NMB : [ OK ]
```


20.6 Configuração do cliente

Esta seção descreve a configuração passo a passo de um cliente Windows® 98.

A instalação nas outras versões do Windows® deverá ter uma lista de passos semelhantes. Em caso de dúvidas, você deverá consultar a documentação do Windows® ao que se refere aos passos de instalação de uma rede ponto a ponto.

1. Instale normalmente a sua placa de rede conforme a documentação do Windows®. Você precisará do protocolo TCP/IP e do serviço **Cliente para Redes Microsoft**. Novamente, consulte a documentação do Windows® para informações sobre como instalar este serviço.
2. Clique na pasta Identificação na tela de configuração de redes e digite o nome de sua máquina e o nome do grupo de trabalho. Em nosso exemplo, a máquina se chamará kepler e fará parte do grupo de trabalho minhaorganizacao.
3. Clique na pasta Controle de Acesso e verifique que a tela está configurada como Figura 5-8.
4. Voltando para a tela de configuração de redes, abra a tela de propriedades do proto-colo TCP/IP; clique em Configuração DNS e adicione o endereço IP do servidor Samba à lista de servidores DNS. Se já não o tiver feito, ative o DNS (opção Ativar DNS na mesma tela). Note que esta configuração somente se aplica a redes que possuem um servidor DNS. Você deverá, também, editar a pasta Endereço IP, informando o endereço de IP a ser utilizado pela estação (se sua rede possuir um servidor DHCP, você não precisará desta configuração).
5. Clique em OK e deixe a configuração de rede. O Windows® não pode ser configurado imediatamente e, assim, você terá de reiniciá-lo quando ele assim o solicitar. Esses passos deverão permitir-lhe colocar uma estação Windows® 98 em sua rede Linux. Após a máquina ter reiniciado, você poderá abrir o Ambiente de rede para navegar pelas máquinas disponíveis na rede.

20.7 Utilizando o SWAT

Além do Linuxconf, é possível configurar o Samba através de uma outra interface amigável: o SWAT. O SWAT é uma interface *web* para a configuração. Uma grande vantagem do SWAT é a de que ele permite a configuração remota de um servidor Samba, já que só é necessário acessar a máquina pela Internet. Para habilitar o SWAT a partir de seu servidor, é necessário editar o arquivo `/etc/inetd.conf` e descomentar (retirar o “#” inicial) da linha:

```
swat stream tcp nowait.400 root /usr/sbin/swat swat
```

que está, normalmente, no final do arquivo. Caso você não encontre uma linha semelhante, pode-se adicionar a linha acima.

Após feita a edição, o serviço **inetd** deve ser reiniciado:

```
killall -HUP inetd
```

ou ainda:

```
/etc/rc.d/init.d/inet restart
```

A diferença entre os dois comandos é que o segundo irá iniciar o serviço, caso não esteja iniciado.

A partir do momento em que o **inetd** for reiniciado, o SWAT será acessível através da porta 901 de seu servidor. Você pode acessar o SWAT <http://localhost:901> através de qualquer navegador

21 - Servidor Web

21.1 O Protocolo HTTP

Para que você possa visualizar um documento de hipertexto na Internet, é necessário um protocolo para fazer a comunicação entre você e o servidor. Este protocolo é o HTTP 1. Uma sessão HTTP geralmente envolve poucos passos. Por exemplo:

1. O cliente estabelece a conexão com o servidor (isso é feito através de outros protocolos de rede, geralmente TCP/IP) e solicita um documento:

```
GET /index.html HTTP/1.0
```

Neste exemplo, o cliente está solicitando o arquivo /index.html que é, normalmente, o arquivo padrão. Veremos mais sobre isso posteriormente. O importante agora é notar que estamos apenas enviando uma requisição ao servidor HTTP.

2. Após a requisição, o servidor responde. Esta resposta é dividida em três partes:

- Um código de retorno. A grande maioria das implementações do HTTP retornam este código seguido de uma mensagem (geralmente **OK** ou uma descrição do erro, se for o caso). Além disso, é retornado um texto identificando a versão do protocolo. Em nosso exemplo, esta linha é:

```
HTTP/1.1 200 OK
```

significando que o servidor está se comunicando através do **HTTP 1.1** e a operação (a requisição do arquivo) foi bem sucedida.

- Um cabeçalho. Este cabeçalho contém diversas informações sobre o arquivo sendo enviado, como tamanho e informações sobre o próprio servidor:

```
Date: Fri, 19 May 2000 20:53:51 GMT
Server: Apache/1.3.12 (Unix) (Conectiva/Linux) mod_ssl/2.6.0
OpenSSL/0.9.4
Last-Modified: Thu, 15 Apr 1999 16:38:13 GMT
ETag: "177c3-508-371615f5"
Accept-Ranges: bytes
Content-Length: 1288
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

As informações acima demonstram que estamos conectados a uma máquina executando o Apache 1.3.12. Temos a data da última modificação do arquivo, seu tamanho e seu tipo. Estas informações podem ser utilizadas pelo programa cliente para decidir o que fazer com determinado arquivo.

- O arquivo propriamente dito. Após o cabeçalho, uma linha em branco é enviada pelo servidor, indicando que, a partir daquele ponto, tudo o que for mandado pelo mesmo será

parte do arquivo requisitado. Um programa cliente pode controlar o progresso da transmissão monitorando quanto já foi transmitido e comparando esta informação com o tamanho informado no cabeçalho.

21.2 O Apache

O Apache é o servidor web mais popular do mundo. Uma pesquisa recente demonstrou que o Apache é mais utilizado do que todos os outros servidores disponíveis juntos.

A popularização da *web* nos últimos anos tem obrigado as empresas a disponibilizarem conteúdo na mesma. Para isso, a utilização do Apache é imperativa para qualquer empresa que deseje manter uma presença na rede.

Além de permitir que você disponibilize conteúdo na Internet, o Apache lhe permite a publicação de informações em sua Intranet, tornando-o realmente um produto indispensável.

21.2.1 Instalando o Apache

1. Acesse o CD da distribuição do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS
```

2. Instale o pacote do Apache:

```
# rpm -ivh apache-*
```

```
apache #####
```

```
apache-devel #####
```

```
apache-doc #####
```

21.2.2 Configurando o Apache

O Apache que acompanha a distribuição do Conectiva Linux já vem configurado com as opções mais utilizadas, de forma que você só precisa iniciar o servidor para que seu Conectiva Linux seja um servidor *web*. Quando você inicializa o servidor *web*, você pode acessar sua máquina via HTTP e visualizar a página padrão como mostrada na Figura 6-1:

Para inicializar o servidor Apache, abra um terminal e digite:

```
# cds  
atalk functions keytable lpd network  
atd gpm killall mars-nwe nfs  
autofs halt kudzu mysql nfslock  
crond httpd ldap named pcmcia  
dhcpd inet linuxconf-setup netfs portmap  
# ./httpd start
```

Iniciando httpd: [OK]

#

A configuração do Apache depende muito do perfil de servidor *web* desejado. Nesta seção você verá como configurar o Apache da maneira mais simples possível. Depois, você saberá um pouco sobre os módulos mais populares do Apache e como instalá-los e configurá-los.

Para configurar o Apache, você deve acessar o Linuxconf e entrar em Ambiente de rede ! Tarefas de servidor ! Apache - servidor www. Você verá o menu inicial da configuração do Apache como mostrada na Figura 6-2.

Clique em Padrões para alterar as opções básicas de seu servidor WWW.

Há um grande número de opções a serem configuradas nesta tela, mas nos ateremos às mais comuns:

- **Endereço de email do administrador:** aqui você poderá informar o endereço de correio eletrônico do administrador do *site*.

- **Domain IP address:** neste campo você poderá informar um endereço IP padrão para o domínio. Este endereço IP é, geralmente, o endereço IP do servidor www, e será utilizado quando alguém tentar acessar a página através do nome do domínio ao invés do nome da máquina. Por exemplo, imagine que o servidor Apache está sendo executado na máquina `kepler.minhaorganizacao`, que tem o endereço IP 10.0.0.1. Se você informar o mesmo endereço IP da máquina `kepler` (10.0.0.1) como **Domain IP address**, quando alguém tentar acessar o endereço `minhaorganizacao` (o domínio), ele irá, na verdade, acessar `kepler.minhaorganizacao` (a máquina que realmente possui o endereço de IP especificado).

- **Nome do servidor:** você poderá informar o nome da máquina onde o Apache está sendo executado. Em condições normais, este campo não é necessário, já que o Apache é capaz de descobrir o nome através do **DNS** ou do próprio arquivo `hosts`.

- **Diretório raiz dos documentos:** você poderá informar aqui o diretório onde estarão os arquivos do seu *site*. O diretório padrão é `/home/httpd/html`.

- **Diretório alternativo de scripts:** você poderá informar vários *apelidos* para os diretórios de CGI. Isso quer dizer que você vai definir um diretório virtual e associá-lo a um diretório real. Por exemplo, você pode definir que quando uma página fizer um referência ao diretório virtual `/cgi-bin`, ele estará, na realidade, acessando os arquivos do diretório `/home/httpd/cgi-bin`.

Uma das características mais utilizadas no Apache é a sua capacidade de lidar com *máquinas virtuais* 2 . Uma máquina virtual é, na verdade, um pequeno truque envolvendo o Apache e o serviço de nomes do servidor (DNS ou o arquivo `hosts`). Basicamente, uma máquina virtual é um apelido para a máquina real. Este apelido deve ter um IP próprio. Com isso, pode-se fazer com que apenas um servidor Apache sirva diversos *sites* separadamente. Em nosso exemplo, assumimos que nossa máquina chama-se `kepler`.

Agora vamos supor que queiramos disponibilizar informações de suporte técnico através de outro nome de máquina, por exemplo `suporte.minhaorganizacao`. O conteúdo deste outro *site* deverá estar localizado na mesma máquina kepler.

Primeiramente, você deve estabelecer um apelido de IP para a máquina kepler. Para fazer isso, você deve pressionar o botão **Apelidos de IP para máquinas virtuais** no menu Ambiente de Rede do Linuxconf. A Figura 6-4 mostra a tela de configurações dos apelidos de IP. Note que você deve escolher a *interface* de rede a ser utilizada. Esta deve ser a *interface* utilizada para acessar a máquina 2. No Apache, este conceito é chamado de *VirtualHost*

Basta informar o novo endereço de IP para a máquina. No exemplo da Figura 6-4, definimos que a máquina kepler (o nosso servidor) responderá ao endereço IP 10.0.1.1 além do seu endereço de IP *real*. Note, porém que, antes de continuarmos, você deve configurar o seu serviço de nomes para que este associe o nome `suporte` a este endereço IP. Não faz parte do escopo deste capítulo o processo de atualização do DNS, verifique a documentação disponível sobre servidores de nomes para aprender como fazer esta configuração.

Agora você pode voltar à tela de configuração do Apache no Linuxconf e pressionar **Máquinas Virtuais** seguido de **Adicionar**. Na tela da Figura 6-5, você poderá definir as opções de sua máquina virtual.

Note que as opções disponíveis são semelhantes às utilizadas na configuração dos padrões do Apache. Mas vamos ver o significado dos valores definidos em nosso exemplo:

- **Nome da máquina virtual:** aqui definimos que nossa máquina virtual se chamará `suporte`. Novamente, lembre-se de que o seu DNS deve estar configurado para reconhecer este nome e mapeá-lo para o apelido de IP que escolhemos.

- **Endereço email do administrador:** aqui apenas definimos que o endereço de correio eletrônico do administrador é `root@minhaorganizacao`.

- **Domain IP address:** especificamos o endereço do nosso domínio. Este endereço será utilizado quando for feita uma tentativa de acesso à máquina através do nome do domínio sem o nome da máquina.

- **Nome do servidor:** este é o nome da máquina servidor.

- **Diretório raiz dos documentos:** este é o diretório onde nós iremos colocar os arquivos da máquina `suporte`. Assim, todas as páginas e figuras do suporte técnico estarão no diretório `/home/httpd/suporte/html`.

- **Registros de erro:** quando algum erro ocorrer durante uma tentativa de acesso às páginas de `suporte`, o Apache salvará informações sobre o mesmo neste arquivo.

- **Registros de transferência:** este arquivo contém um registro dos acessos à máquina virtual `suporte`. Estas informações podem ser utilizadas para a obtenção de informações de segurança.

• **Nomes alternativos de *script*:** aqui você pode definir diretórios dos *scripts* CGI a serem utilizados pela máquina support. Note que é possível utilizar o mesmo diretório para todas as máquinas, sejam elas virtuais ou reais.

21.3

22 - Servidor FTP

Nesta seção trataremos do servidor FTP. No Conectiva Linux o WU-FTPD é quem realiza os serviços de FTP. Assim, esta seção irá ser voltada ao WU-FTPD.

22.1 O WU-FTPD

FTP é um acrônimo para *File Transfer Protocol* ou *Protocolo de Transferência de Arquivos*. O protocolo FTP permite a transferências de arquivos binários e arquivos texto com alta eficiência através de uma rede.

22.1.1 Instalação e Configuração

1. Para instalar o WU_FTPD, acesse o diretório de pacotes do CD da distribuição de seu Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS
```

2. Instale o pacote do WU_FTPD:

```
# rpm -ivh wu-ftp-*
```

```
wu-ftp-#####
```

3. Certifique-se que a linha abaixo esteja presente no arquivo `/etc/inetd.conf` e que ela não esteja comentada:

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
```

4. Se você houver feito alterações ao arquivo `/etc/inetd.conf`, você deve reiniciar o **inetd**:

```
# cds
```

```
atd gpm keytable lpd nfs sendmail syslog
```

```
crond halt killall mars-nwe pcmcia single xfs
```

```
dhcpd httpd kudzu netfs portmap snmpd ypbind
```

```
functions inet linuxconf-setup network random sshd
```

```
# ./inet stop
```

```
Interrompendo os serviços INET: [ OK ]
```

```
# ./inet start
```

```
Iniciando os serviços INET: [ OK ]
```

Com o servidor FTP devidamente instalado, você pode acessar o Linuxconf para iniciar o processo de configuração.

No Linuxconf, vá para Ambiente de Rede ! Tarefas de Servidor ! Wu-ftp - servidor de ftp para abrir a tela inicial de configuração do servidor FTP.

Clique em Configuração básica ! Diversos. Estas são as opções mais básicas do servidor FTP. Com elas, você poderá definir o comportamento geral de seu servidor.

- **Email do administrador:** endereço de correio eletrônico do administrador do sistema.

- **Grupo de convidados:** aqui você pode informar o grupo do Linux ao qual o usuário **anonymous** pertencerá. Você terá mais informações sobre acesso anônimo ao servidor na seção *Acessos Anônimos*. Se este campo não for preenchido, o grupo **nobody** será utilizado.

- **Arquivo de banner:** o conteúdo do arquivo de *banner* será mostrado aos usuários no *168. Capítulo 6. Servidor Internet* momento em que os mesmos acessarem com sucesso o seu servidor.

- **Mensagem de encerramento:** se o arquivo informado existir, sempre que um usuário tentar acessar o servidor, seu conteúdo será mostrado e o servidor irá fechar a conexão. O arquivo é bastante útil sempre que você quiser interromper o serviço de FTP temporariamente.

- **Permitir acesso anônimo:** aqui você pode permitir ou proibir os acessos anônimos. Mais sobre o assunto na seção *Acessos Anônimos*. Agora pressione Controle nesta mesma tela para acessar as opções de controle de acesso do servidor FTP.

Existem, na realidade, três telas de configuração de controle de acesso:

- **Usuários reais:** controle de acesso dos usuários reais do seu sistema. Esses são os usuários que têm contas em sua rede.

- **Usuários convidados:** controle de acesso a usuários convidados.

- **Anônimos:** controle de acesso de usuários que se conectam anonimamente. As três telas são absolutamente idênticas umas às outras.

As opções disponíveis

- **pode requisitar arquivos comprimidos:** define se o usuário pode requisitar que seus arquivos sejam compactados. O WU-FTPD permite a compactação de arquivos durante a transmissão.

- **pode requisitar arquivos tar:** especifica se o usuário tem permissão de solicitar o arquivamento de arquivos transmitidos com o tar.

- **pode usar chmod para arquivos:** especifica se o usuário pode modificar as permissões de arquivos localizados no servidor FTP.

- **pode excluir arquivos:** define se o usuário tem permissão de apagar arquivos localizados no servidor FTP.

- **pode atualizar arquivos:** define se o usuário pode sobrescrever arquivos no servidor.

- **pode renomear arquivos:** especifica se o usuário pode modificar o nome dos arquivos localizados no servidor.

- **registrar transferências recebidas:** especifica se o servidor deve manter um registro de arquivos recebidos.

- **registrar transferências expedidas:** especifica se o servidor deve manter um registro de arquivos enviados.

22.1.2 Acessos Anônimos

Pode ser permitido ao usuário acessar seu servidor de maneira anônima. Isso é muito útil no caso de você desejar disponibilizar arquivos a pessoas de fora de sua organização. Como essas pessoas não possuem contas de usuário em seu sistema, a única forma de acessar seus arquivos é através do acesso anônimo.

Para permitir o acesso anônimo ao seu servidor:

1. Acesse o diretório de pacotes do CD do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS
```

2. Instale o pacote de acesso anônimo:

```
# rpm -ivh anonftp-*
```

```
anonftp #####
```

A partir desse momento, para acessar o servidor FTP anonimamente, o usuário deverá fornecer o nome de usuário **anonymous** e seu endereço de correio eletrônico como senha.

```
$ ftp localhost
```

```
Conectado na máquina localhost.
```

```
220 einstein.minhaorganizacao FTP server
```

```
(Version wu-2.6.0(1) Fri May 12 11:05:03 BRST 2000) ready.
```

```
Usuário (localhost:albert): anonymous
```

```
331 Guest login ok, send your complete e-mail address as password.
```

```
Senha:
```

```
230 Guest login ok, access restrictions apply.
```

```
O tipo do sistema remoto é UNIX.
```

```
Usando modo binary para transmitir/receber arquivos.
```

```
ftp>
```

Note que no exemplo acima, a senha não é mostrada, mas o usuário teve de digitar seu endereço de e-mail para poder ter acesso ao servidor. O usuário **anonymous** não precisa (nem deve) ser cadastrado em seu Conectiva Linux, já que ele é um usuário especial para o servidor FTP. Quando é feita uma tentativa de acesso com o usuário **anonymous**, o servidor automaticamente trata o acesso como anônimo, aceitando o endereço de correio eletrônico como senha.

22.1.3 Permitindo Envio de Arquivos

Normalmente, você não precisará (e provavelmente nem desejará) que os usuários possam gravar arquivos em seu servidor FTP. Porém, em alguns casos, pode haver interesse em disponibilizar-se uma área para que os usuários possam guardar arquivos.

Tornou-se um costume fazer com que os usuários tenham um local específico nos servidores de FTP para gravarem arquivos. Esse local é o diretório `/incoming`.

Permitir que usuários gravem em seu servidor FTP é um grande risco e, por isso, deve-se pensar muito bem antes de fazê-lo.

Um dos maiores problemas com isso é que você não tem um grande controle sobre aquilo que é gravado em seu servidor. Você não pode facilmente impedir que usuários guardem material ilegal em seu servidor.

Recomendamos, portanto, que você não permita o acesso de escrita em seu servidor.

Caso você não tenha escolha, você deve ter alguém responsável por monitorar os arquivos guardados em seu servidor para evitar arquivos que possam trazer problemas no futuro.

Para criar o `/incoming`:

1. Acesse o diretório raiz do FTP:

```
# cd /home/ftp
```

2. Crie o diretório `incoming`

```
# mkdir incoming
```

3. Agora você deve fazer com que o diretório criado seja de propriedade de um usuário

e grupo diferente de **root** e **dftp**. Você pode criar um usuário e grupo específicos para isto se desejar, mas utilizaremos **nobody.nobody** em nosso exemplo.

```
# chown nobody incoming
```

```
# chgrp nobody incoming
```

```
# chmod 3773 incoming
```

4. Agora você deve editar o arquivo `/etc/ftpaccess` para permitir a escrita ao diretório `/incoming`. Note que, se você houver criado um usuário e grupo para ser dono do diretório, você deverá informar isso na linha abaixo no lugar de **nobody**.

```
upload /home/ftp /incoming yes nobody nobody 0400 dirs
```

O formato desta linha é:

```
upload HOME DIR GRAVA USUARIO GRUPO PERMS DIRS
```

Onde:

- **HOME**: diretório *home* do usuário. Em nosso exemplo, informamos `/home/ftp`. Isso quer dizer que esta linha se aplica a qualquer usuário cujo diretório *home* seja `/home/ftp`.

- **DIR**: diretório ao qual se refere a linha. Esse diretório é relativo à raiz do diretório do FTP. Em nosso exemplo, informamos que esta linha se aplica ao diretório `/incoming`.

- **GRAVA**: aqui você informa se é ou não é permitida a gravação no diretório ao qual a linha se refere. Os valores permitidos são **YES** ou **NO**. Em nosso exemplo, estamos permitindo a escrita.

- **USUARIO**: é o nome do usuário a quem todos os arquivos gravados no diretório pertencerão. Em nosso exemplo esse usuário é **nobody**.

• **GRUPO:** é o grupo a quem todos os arquivos gravados no diretório pertencerão. Em nosso exemplo esse grupo é **nobody**.

• **PERMS:** após gravados, os arquivos terão as permissões trocadas para estas. Em nosso exemplo definimos que os arquivos gravados neste diretório passariam a ter permissão 0400, ou seja, apenas para leitura do dono (no caso, o usuário **nobody.nobody**). Recomendamos estas permissões, já que elas não permitem que usuários utilizem o seu servidor para troca de arquivos ilegais. Isso assegura que alguém terá de verificar o arquivo e trocar suas permissões antes que alguém possa acessá-lo via FTP.

• **DIRS:** especifica se o usuário anônimo pode criar diretórios dentro do diretório `/incoming`. Em nosso exemplo o usuário pode criar. Os valores possíveis são **dirs** e **nodirs**.

22.1.4 Arquivos de Mensagens e Banners

Se você já acessou algum servidor FTP, então já deve ter notado que é comum aparecer mensagens informativas quando se conecta ou quando se muda de diretório. Estas mensagens são muito úteis para informar o usuário das possibilidades e regras a serem seguidas no servidor.

22.1.5 Arquivo de Banner

Quando você configura o WU-FTPD no Linuxconf, há uma opção chamada Arquivo de banner. Você pode informar um arquivo cujo conteúdo será mostrado ao usuário antes da conexão em si. Ele é útil para uma breve mensagem de boas-vindas ao servidor. Pode-se fazer uma breve explicação sobre o servidor e os direitos de acesso ao mesmo.

Lembre-se que o arquivo de *banner* é apresentado ao usuário antes do *login*. Assim, evite dar muita informação sobre o servidor através deste arquivo.

22.1.6 O Arquivo `.message`

Um dos arquivos mais comuns e mais úteis em um servidor FTP é o arquivo `.message`. Quando o servidor encontra este arquivo em um diretório, ele mostra seu conteúdo para o usuário antes de mostrar o conteúdo do diretório em si.

Você pode utilizar este arquivo para dar breves explicações sobre os propósitos dos diretórios sendo acessados. Além disso, você poderia ajudar o usuário a encontrar o que ele está procurando.

Por exemplo, digamos que um usuário queira acessar o seu servidor para encontrar uma atualização para um programa. O usuário, porém, não tem muita experiência com servidores FTP e não sabe ao certo como encontrar aquilo de que ele precisa. Assim, você poderia criar arquivos `.message` para guiar o usuário. Quando o usuário acessa o diretório `/pub` do servidor FTP, por exemplo, você poderia explicar-lhe o que está disponível ali. Um exemplo de como a mensagem é mostrada para o usuário está na Figura 6-9.

23 - Servidor Proxy

Nesta seção iremos tratar da implementação de um servidor *proxy* em um Conectiva Linux. O *software* servidor *proxy* que acompanha o Conectiva Linux é o Squid, e é nele que esta documentação se centralizará.

23.1 Caching

Quando você acessa uma página da *web* ou um arquivo de FTP, uma requisição parte de sua máquina até o servidor; só então os dados são transmitidos para a sua máquina. Como muitas vezes a distância entre o servidor e a sua máquina é muito grande e a qualidade das linhas de transmissão muito irregulares, este processo acaba por tornar-se bastante lento.

Além disso, a maioria dos dados requisitados é estática; eles não mudam com o tempo. Os logotipos que as empresas colocam em suas páginas, por exemplo, não tendem a mudar. Entretanto, eles são, muitas vezes, bastante grandes. Isso é um enorme desperdício de recursos da rede, além de tempo. Uma solução encontrada foi o chamado *caching*. Sempre que é feita uma requisição de algum objeto da Internet, o servidor *proxy* consulta o *cache* para verificar se este objeto já não foi requisitado previamente. Se ele foi, então o servidor *proxy* pode responder à requisição utilizando sua própria cópia local do objeto.

Isso acelera significativamente as operações na Internet, já que grande parte dos objetos acaba trafegando apenas localmente.

O servidor *proxy* verifica se a sua cópia é atualizada com o objeto original. Caso não for, o *proxy* atualiza sua cópia. Naturalmente, um servidor de *cache* não poderia guardar todos os objetos acessados para sempre, pois isso iria rapidamente saturá-lo. A solução é simples: o servidor mantém apenas os arquivos utilizados a menos tempo. Isso garante, de uma forma indireta, que os objetos mais freqüentemente utilizados sempre estejam no *cache*.

23.2 O Squid

O Squid é o servidor *proxy* do Conectiva Linux. Ele oferece alta performance de *cache* para servidores *web*.

O Squid oferece grandes vantagens em comparação com outros servidores *proxy*:

- ele realiza, além do *cache* de objetos como arquivos de FTP e páginas da *web*, um *cache* de procuras de DNS. Isso quer dizer que ele guarda informações sobre o mapeamento entre endereços IP e nomes de máquinas da Internet. Isso acelera a procura de máquinas;

- ele mantém os objetos mais utilizados na memória RAM (cujo uso pode ser limitado pela configuração);
- ele suporta SSL (acesso a páginas criptografadas) para segurança em transações;
- pode ser organizado em hierarquias de servidores de *cache* para uma melhora significativa de performance;
- responde às requisições em um único processo de acesso a disco.

Todo o servidor *proxy* Squid consiste de um programa principal (**Squid**) e de seu próprio programa de resolução de nomes (**dnsserver**). Quando o Squid é inicializado, ele cria o processo do dnsserver, diminuindo o tempo de espera pela resposta do DNS.

23.2.1 Instalação e Configuração

Para instalar o Squid:

1. Acesse o diretório de pacotes do CD do Conectiva Linux:

```
# cd /mnt/cdrom/conectiva/RPMS
```

2. Instale o pacote do Squid:

```
rpm -ivh squid-*
```

```
squid #####
```

3. Inicialize o programa:

```
# cds
```

```
atd gpm killall named pcmcia single
```

```
autofs halt kudzu netfs portmap smb
```

```
crond httpd linuxconf-setup network postgresql snmpd
```

```
dhcpcd inet lpd nfs random squid
```

```
functions keytable mars-nwe nfslock sendmail sshd
```

```
[root@gnu init.d]# ./squid start
```

```
creating directories at /var/spool/squid, wait... [ OK ]
```

```
Iniciando squid [ OK ]
```

Para realizar configurações no Squid é necessário editar o seu arquivo de configuração que está localizado em `/etc/squid/squid.conf`.

23.2.2 Memória para Cache

O Squid armazena os objetos mais utilizados na memória RAM. Isso faz com que a performance seja muito melhor do que se os objetos fossem armazenados em disco.

A quantidade de memória a ser utilizada para *cache* do Squid pode (e deve) ser limitada, de forma a não interferir com outros processos no sistemas. O padrão é 8MB.

Note que este limite de memória refere-se à memória usada para *cache*, e não a memória total utilizada pelo Squid. Na realidade, a memória utilizada pelo Squid deverá ficar em torno de três vezes este valor.

Para especificar a quantidade de memória máxima a ser utilizada para *cache*, você deve utilizar o parâmetro `cache_mem` no arquivo `/etc/squid/squid.conf`:

```
cache_mem MEM
```

Onde **MEM** é a quantidade de memória máxima a ser utilizada para *cache*.

Exemplo:

```
cache_mem 32 MB
```

Este exemplo especifica que o Squid deverá limitar a 32MB a quantidade de memória utilizada para *cache*.

23.2.3 Arquivos de Cache

Os objetos guardados pelo Squid são guardados em arquivos no disco rígido. Você pode especificar algumas opções para definir como o Squid trabalhará com arquivos.

```
cache_dir TIPO NOMEDIR MB N1 N2
```

Onde:

- **TIPO**: você deve especificar o tipo do sistema de armazenamento que o Squid deverá utilizar. Normalmente, você deve utilizar **ufs**. Você pode tentar utilizar **asynccufs** para obter melhor performance. Se, por acaso, o **asynccufs** não funcionar corretamente em seu sistema, volte a usar o **ufs**.

- **NOMEDIR**: especifique o diretório onde os arquivos serão gravados. Note que este diretório não será criado automaticamente pelo Squid.

- **MB**: você pode modificar o espaço máximo a ser utilizado para *cache* neste diretório. O valor padrão é 100 MB.

- **N1**: você pode especificar o número máximo de diretório de primeiro nível que serão criados dentro do diretório de *cache*. O padrão é 16.

- **N2**: número máximo de diretórios de segundo nível que serão criados dentro de cada diretório de primeiro nível. O valor padrão é 256.

Exemplo:

```
cache_dir ufs /var/spool/squid 100 16 256
```

Este exemplo define que o diretório de *cache* será o `/var/spool/squid`, que ele poderá ter até 100MB, 16 diretórios de primeiro nível e 256 diretórios de segundo nível dentro de cada diretório de primeiro nível.

23.2.4 Controle de Acesso

Por razões de segurança, a configuração padrão do Squid do Conectiva Linux é bastante conservadora; ela nega acesso a qualquer máquina.

Você deve alterar este comportamento para poder utilizar o Squid. Deve, portanto, estabelecer regras de acesso ao servidor *proxy*. As regras têm o formato:
http_access PERM QUEM

Onde:

- **PERM**: indica se a linha é uma permissão ou uma proibição de acesso. Os valores permitidos são: **allow**, para permitir e **deny**, para negar acesso.

- QUEM**: a quem se refere esta permissão. Pode ser uma máquina ou um domínio ou uma classe. Pode-se ainda usar **ALL** para indicar que a permissão definida na linha se refere a todas as máquinas.

Exemplo:

```
http_access allow all
```

Este exemplo permite que todas as máquinas da rede possam utilizar este servidor *proxy*.

Note que, como mencionado anteriormente, o Squid é pré-configurado para não permitir acesso de nenhum usuário ou máquina. Assim, você deve procurar a linha do arquivo `/etc/squid/squid.conf` parecida com:

```
http_acces DENY all
```

E removê-la ou editá-la no arquivo.

24 - Configuração básica do Sendmail

24.1 Apresentar seu sistema como

Mesmo que o email venha de máquinas diferentes de sua empresa, você pode querer esconder esse fato e apresentar os emails como se eles tivessem vindo de apenas uma máquina ou domínio.

Este campo é normalmente usado na maioria das configurações e simplesmente contém o nome oficial de seu domínio.

24.2 Aceitar email para seu domínio

Se você está configurando o servidor principal de uma empresa, você deve selecionar essa opção. Normalmente, um servidor sendmail aceitará apenas emails enviados ao seu nome@domínio exato. Se este for o servidor principal, ele receberá também emails endereçados ao domínio, e não ao nome do servidor. Selecionando essa opção, o servidor aceitará ambos.

Por exemplo, se seu domínio é foo.com.br e seu servidor é mail.foo.com.br, então um email para joão@foo.com.br e um para joão@mail.foo.com.br serão aceitos e guardados localmente neste servidor (caso o usuário joão exista).

24.3 Servidor de email

Em muitas empresas, há uma máquina que guarda os emails de todos os usuários, mesmo se o email for originário de máquinas diferentes. As máquinas Unix são multiusuário. É possível um usuário mandar um email para outro usuário da mesma máquina. Às vezes isso é inapropriado. As máquinas Unix além de serem multiusuário, são orientadas à rede. Esse usuário (o destinatário) pode ter contas em outras máquinas da rede (até em todas as máquinas) e pode escolher uma outra máquina para ler seus emails. Ter essas mensagens de email espalhadas por todas as máquinas da rede não é muito desejável.

Você especifica aqui o endereço completo do servidor da empresa e todas as mensagens de email serão enviadas para ele.

24.4 Roteador de email

Muitas máquinas em uma rede não têm conectividade suficiente para repetir emails apropriadamente. Você pode especificar o nome da máquina que tem. Em muitas empresas, geralmente é a mesma máquina onde está o servidor de email.

Diferentemente do servidor de email, o roteador de email pode ser acessado por vários protocolos de envio diferentes como o ESMTP e o UUCP.

24.5 Protocolo do roteador de email

Você deve especificar como acessar o roteador de email. Se você escolher o ESMTP, então você deve especificar um nome completo para o roteador de email. Se você escolher UUCP, então você deve especificar um nome UUCP para o roteador de email.

24.6 Funcionalidades

Ativar controle de envio (spammers)

O controle de envio lhe deixa decidir quem pode usar seu servidor SMTP (sendmail) para repetir mensagens. Os spammers costumam *_emprestar_* servidores SMTP para espalhar uma mensagem para uma lista enorme de pessoas. Configurando o controle de envio, você pode limitar isso. Infelizmente, regras de reenvio mal configuradas podem impedir que alguns de seus usuários mandem email..

Esta opção lhe deixa ativar/desativar as regras. Um arquivo `sendmail.cf` novo será gerado, sem perder a configuração das regras. Note que essa opção **NÃO** controla a funcionalidade de remetentes rejeitados, que lhe permite rejeitar emails vindos de spammers conhecidos. Isso não limita a habilidade de repetir.

Tamanho máximo das mensagens

As mensagens podem ser limitadas; assim, mensagens muito grandes serão rejeitadas. Definindo o tamanho para 0, você desativa essa opção (sem limite).

Não tentar enviar imediatamente

Esta funcionalidade é feita para pessoas que não têm uma conexão permanente com a Internet. Ela diz ao sendmail para gravar o email sem checá-lo. Mais tarde, você pode disparar a entrega do email. Geralmente se faz isso num comando de pós-conexão (veja a seção de Discagem PPP) como `_/usr/sbin/sendmail -q_`, que força o sendmail a processar e enviar as mensagens.

Processar consulta a cada(min)

De vez em quando o sendmail se levanta e tenta enviar os emails que ainda lhe restam. Você pode ver essa fila de saída com o comando `mailq` sem argumentos. Pessoas

sem conectividade contínua com a Internet, podem querer usar essa opção. Definindo o intervalo para 0, você desativa essa opção. Se sua conectividade é por demanda, você pode querer colocar aqui um número bem grande, como por exemplo, 30 minutos. Quando o sendmail se levantar, ele irá/deverá disparar a conexão.

Aqueles que desativarem essa opção, podem querer controlar o sendmail `_na mão_` (ou via cron) com o comando: `/usr/sbin/sendmail -q`.

24.7 Diversos

Não usar batch para email UUCP

Quando usa-se UUCP repetidamente, as mensagens são normalmente requisitadas e entregues em intervalos regulares. Você pode querer processá-las e enviá-las imediatamente. Para um tráfego pouco intenso, isto é às vezes necessário, pois o email é repetido imediatamente e os usuários ficam felizes..

Tamanho máx. das mensagens UUCP

As mensagens UUCP podem ser limitadas. Mensagens muito grandes serão rejeitadas. Isso é útil para uma conexão via modem muito lenta. Definindo o tamanho para 0, você desativa essa opção (sem limite).

Entrega local para usuários

Coloque aqui o nome dos usuários para quem você quer ignorar a definição de Servidor de email. As mensagens sempre serão guardadas localmente para estes usuários. Essa funcionalidade é normalmente usada para pseudo-usuários administrativos (root por exemplo).

Não mascarar a origem de

Você pode querer ignorar a definição de `_Apresentar seu sistema como_` para alguns usuários. Um é o root. Isso permite ao destinatário diferenciar facilmente de qual sistema (qual root?) a mensagem se origina.

Banco de roteamentos especiais

Ative caso você pretenda usar roteamento de domínio especial. Infelizmente, o Configurador Linux não pode checar se a versão do sendmail instalada em seu sistema suporta o banco de roteamentos especiais e em quais formatos. Essas são opções de tempo de compilação para o sendmail e nenhuma opção de linha de comando do sendmail pode dar o suporte a elas.

Formato do banco de roteamentos

O Sendmail pode usar muitos formatos para algumas de suas bases de dados. O padrão geralmente é o formatodbm. Note que alguns Sendmail têm suporte apenas a um formato (dbm), que é uma opção de tempo de compilação..

Agente de entrega local

Utilitários diferentes são usados para gerenciar emails enviados para usuários locais. Esses utilitários provêm várias extensões que permitem ao usuário fazer uma entrega personalizada de emails. O procmail por exemplo suporta essa filtragem detalhada. Alguns usam o procmail para espalhar as mensagens que chegam, em várias caixas e seguindo vários critérios.

Você pode deixar esse campo vazio e o Configurador Linux localizará o utilitário apropriado disponível em seu sistema.

Suporte a clientes bogus (HELO)

Alguns clientes de email que usam SMTP para mandar emails e estão mal configurados, geram seqüências HELO mal formadas. O sendmail normalmente rejeita a conexão. Com esta opção, o sendmail fica mais tolerante e passa a aceitar essas conexões.

O Sendmail e o DNS

O Sendmail faz um uso pesado do DNS. Máquinas sem conexão direta com a Internet (mesmo atrás de um firewall) podem querer desativar o uso do DNS. Por outro lado, máquinas com uma conectividade normal do DNS devem forçar o uso do DNS.

24.8 Apelidos para seu sistema

As mensagens de email que são especificamente para o seu domínio são aceitas pelo Sendmail. Outras ou serão retornadas ou repetidas. Você pode querer aceitar mensagens de email para domínios diferentes. Uma situação é quando você se registra a um provedor de acesso e ele lhe dá um de seus subdomínios. Você ainda quer aceitar o antigo, porque muitas pessoas lhe conhecem por aquele endereço.

Você pode colocar aqui tantas linhas quanto precise. Use o botão Adicionar caso queira mais linhas..

24.9 Roteamento especial

24.9.1 Razões diferentes para se fazer roteamento especial

Há muitas razões técnicas para o porquê de alguém querer ter o controle sobre o funcionamento normal do Sendmail. Aqui vão algumas:

- _ A máquina de destino não é acessível de seu computador. Seu computador está atrás de um firewall.
- _ O DNS da máquina de destino está malfeito ou incompleto (sem um registro MX válido).
- _ A máquina é acessível apenas por UUCP. Você passa por essa situação quando uma máquina tem um domínio válido, mas não possui conexão direta com a Internet. Seu computador pode ter sido designado como o receptor de email para esse domínio e como tal, ele receberá todos os emails para este domínio. Ele precisará de regras especiais para repetir as mensagens apropriadamente.
- _ Você tem uma rota melhor (menos cara, rápida...) do que a padrão para esta máquina.

24.9.2 Tarefas

Você pode ter regras especiais. Cada regra contém a informação seguinte:

Destino

Este é o domínio de destino que você quer controlar.

Repetidor

Esta é a máquina que receberá (e repetirá) as mensagens para a máquina de destino. O repetidor pode ser um nome de domínio completo ou nome UUCP dependendo da máquina que será usada para alcançar esse repetidor..Gerenciar subdomínio da mesma forma Esta regra pode-se aplicar a um domínio de destino e a todas as máquinas/subdomínios abaixo ele. Note que você pode selecionar esta opção e ainda informar outro roteamento especial para uma máquina/subdomínio deste domínio. Essa regra terá precedência sobre esta opção.

Transporte Você escolhe qual o mecanismo que será usado para encontrar o repetidor.

24.9.3 Arquivos

As regras são guardadas no arquivo `/var/lib/mailertable`. O utilitário `/usr/sbin/makemap` é usado para criar uma base de dados indexada fora desse arquivo. O Configurator Linux cuida disso sempre que você grava as regras.

24.9.4 Mailertable é uma opção

Muitos Sendmail não suportam o mecanismo mailertable. Esta é uma opção de tempo de compilação e algumas distribuições de Linux preferiram não incluí-la. Na tela `_informação básica da máquina_`, há uma opção para ativar o seu suporte (banco de roteamento especial). Se você colocar alguma entrada na tabela, o Configurator Linux lhe avisará se a

funcionalidade não tiver sido escolhida. Uma vez selecionada a funcionalidade, você deve regenerar o arquivo sendmail.cf.

Se seu Sendmail falhou ao iniciar depois disso, cheque os registros de erros em /var/log. Seu Sendmail pode estar configurado para não suportar o conceito de mailertable.

24.10 Roteamento complexo

24.10.1 Introdução

O Sendmail roteia email usando o domínio do destinatário. Se for o mesmo domínio da máquina (usuário local), ele usa o nome do usuário para entregar o email. Às vezes, a decisão deveria ser feita usando-se o nome do usuário e o domínio de destino como uma coisa só..p

24.10.2 Princípios

Uma realidade da Internet nos dias de hoje são vários domínios virtuais na mesma máquina. Isso funciona bem para serviços de Internet. Um simples servidor simula vários servidores virtuais.

Para o email, o Sendmail tem a habilidade de aceitar emails para vários domínios. Geralmente, todos esses domínios são apelidos (veja Apelidos para seu sistema na configuração básica do sendmail) para o seu sistema. Se seu sistema tem os apelidos seguintes:

- _ domínio1.com.br
- _ domínio2.com.br

Então o usuário@domínio1.com.br e usuário@domínio2.com.br apontam para a mesma conta de usuário. Às vezes isso pode estar errado. Roteamento complexo de usuários pode ser entendido dessa maneira: Se você receber um email para um.usuário@um.domínio, então re-
pita esse email para outro.usuário@outro.domínio usando outro.servidor.de.email.

24.11 Tarefas

Para cada regra, você deve preencher todos os campos de uma tela composta por estes campos:

24.11.1 *Esta regra está ativa*

Você pode desativar uma regra sem excluí-la. Você deve regerar o arquivo sendmail.cf e reiniciar o Sendmail (o Configurador Linux lhe dirá para fazê-lo de qualquer maneira).

24.11.2 *Para: original*

O campo Para: original contém o destino original do email, incluindo o nome completo do domínio. É possível redirecionar todo um domínio informando o nome do domínio precedido de um caracter @ (@domínio.com.br). Como redirecionar um domínio é normalmente feito com roteamento especial, via Configurador Linux, usando o roteamento complexo de usuários, pode-se redirecionar todo um domínio para apenas uma conta de usuário em outro domínio! Fácil.

24.11.3 *Para: novo*

Informe aqui o email de destino. Se você quer redirecionar esse email para outro usuário local, simplesmente informe o nome da conta do usuário, sem o domínio. Isso se parece com um apelido simples (veja Apelidos para email). Embora esse roteamento não seja um substituto, ele é mais geral. Para uma base de usuários muito grande, os apelidos para email são mais eficientes pois usam uma base de dados. O roteamento complexo de usuários deve ser usado para lidar com exceções que não podem ser resolvidas pelos apelidos para email, como por exemplo:

- info@domvirtual1.com.br - > josé
- info@domvirtual2.com.br - > joão

24.11.4 *Reenviar para servidor (opc.)*

Normalmente o Sendmail encontrará o servidor apropriado baseando-se no campo Para: novo. Às vezes, você pode querer impor um servidor. Apenas coloque o nome completo desse servidor aqui.

24.11.5 *Transporte*

Se você especificar um servidor, você deve especificar que protocolo utilizar. Normalmente o esmtp deve ser usado. Você também pode escolher o uucp-dom caso esteja repetindo email através de um roteador UUCP. Esse é um caso onde você sempre especificará um servidor de destino.

24.11.6 *Comentário*

Apenas um campo para comentários. Escreva o que quiser aqui.

24.12 Regras de mascaramento

Às vezes passamos por um problema difícil de resolver: as mensagens de email saem para a Internet com endereço de retorno não apropriado. Quase sempre não há o que se fazer, porque o sistema de origem dessas mensagens não pode ser arrumado, devido a todo tipo de razões. As regras de mascaramento oferecem um jeito de se arrumar isso no roteador de email.

24.12.1 Princípios

Você pode definir regras de transformação simples que serão aplicadas a todos os emails enviados pelo sendmail. No geral, você especifica um De: e como este De: deve ser convertido.

Você tem alguma flexibilidade aqui. Você pode converter de:

- _ Um domínio para outro.
- _ Um usuário de um domínio para outro usuário de outro domínio.
- _ Qualquer usuário de um domínio para um usuário de outro domínio (estranho).

Aqui vai uma apresentação dos diferentes campos da caixa de diálogo das regras.

Você pode ativar/desativar uma regra sem apagá-la usando esta opção.

24.12.2 De: original

Você deve colocar aqui um endereço de email. Opcionalmente, você pode omitir o nome do usuário e apenas informar um nome de domínio precedido de um '@'. Isso significa que a regra se aplica para qualquer usuário deste domínio.

24.12.3 De: novo

O endereço de email do campo `_De:_` será reescrito com o conteúdo deste campo. Novamente, você deve especificar um endereço de email. Novamente, você pode omitir o nome do usuário e apenas informar um nome de domínio precedido de um '@'.

24.12.4 Comentário

Coloque aqui o que quiser para lembrá-lo da razão de se usar esta funcionalidade (regra de mascaramento). Normalmente as regras de mascaramento são um último recurso. É melhor arrumar o problema na sua origem. Infelizmente, isso nem sempre é possível, devido a todo tipo de boas (e estúpidas) razões.

24.12.5 Alguns exemplos

De: original : @domínio_escondido
De: novo : @domínio_da_internet.com.br

Esta regra converterá qualquer email vindo de usuário@domínio_escondido para que este tenha um De: de usuário@domínio_da_internet.com.br

De: original : @domínio_escondido
De: novo : joão@domínio_da_internet.com.br

Esta regra converterá qualquer email vindo de usuário@domínio_escondido para que este tenha um De: de joão@domínio_da_internet.com.br. Não sei se alguém algum dia vai precisar disso...

De: original : joão@domínio_escondido
De: novo : info@domínio_da_internet.com.br

Esta regra converterá qualquer email vindo de joão@domínio_escondido para que este tenha um De: de info@domínio_da_internet.com.br..p g ç

24.13 Domínio virtual de email

Com o poder dos computadores pessoais atualmente, as tarefas antigamente feitas por vários servidores hoje são facilmente feitas por apenas um. Com o sucesso da Internet e do sistema de email, somos levados a gerenciar vários domínios de email independentes na mesma máquina. E aí entra o Linux!

24.13.1 Princípios

Domínios virtuais de email são um jeito de se gerenciar listas independentes de usuários no mesmo servidor. Cada domínio virtual tem seu próprio arquivo de senhas, seu próprio diretório de email e seu próprio arquivo de apelidos. Para cada domínio virtual de email, o Configurador Linux irá definir:

```
/etc/vmail/passwd.domínio_virtual  
/etc/vmail/shadow.domínio_virtual  
/etc/vmail/aliases.domínio_virtual  
/var/spool/vmail/domínio_virtual/
```

/vhome/domínio_virtual/

onde domínio_virtual é um domínio, como foo.com.br por exemplo.

24.14 A caixa de diálogo da definição de domínio virtual

Para criar um domínio virtual de email novo, você deve completar uma caixa de diálogo simples. Há outras tarefas relacionadas ao DNS e a apelidos de IP, que são descritas em outra seção desta ajuda. Segue-se a descrição da caixa de diálogo principal.

24.14.1 Um nome

Você deve dar um nome ao domínio. Isso é tudo.

24.14.2 Destino de retorno

É um campo opcional. Normalmente, quando uma mensagem de email é enviada para uma conta de um domínio virtual de email, o processamento seguinte ocorre:

- Checa se existe um apelido com esse nome. Se sim, manda a mensagem para todos os membros da lista. Apelidos podem apontar para outros apelidos.
- Caso não haja um apelido, então checa a lista do domínio virtual. O email é encaminhado à caixa do usuário correspondente.
- Caso não haja nem apelido nem usuário, a mensagem é rejeitada e o remetente receberá uma mensagem de erro.

Se esse campo estiver definido, o email ao invés de ser rejeitado, será enviado a esse endereço de retorno, que pode ser

vazio

É o padrão. A mensagem é rejeitada.

outro_usuario@outro_domínio

A mensagem é enviada a um usuário específico de outro domínio.

@outro_domínio.

A mensagem é enviada para a mesma conta, mas em outro domínio. Por exemplo, um email enviado para desconhecido@este_domínio será repetido para desconhecido@outro_domínio.

conta

A mensagem é enviada para outra conta no mesmo domínio. Esta conta pode ser um apelido.

24.14.3 Arquivos de apelidos (opc.)

Cada domínio virtual tem implícito um arquivo de apelidos nomeado `/etc/vmail/aliases.domínio` onde `_domínio_` é o nome do domínio. Você pode definir mais um. Eles serão usados pelo programa `vdeliver`. O implícito tem a prioridade mais alta. O `vdeliver` olha no primeiro, depois no segundo, até achar o nome.

Note que, como os apelidos normais (`/etc/aliases`) processados pelo `sendmail`, a definição de apelidos pode apontar para outros apelidos, listas de discussão podem ser definidas, etc.

Os arquivos de apelidos são gerenciados pela mesma caixa de diálogo dos apelidos do `sendmail`, e como tal, oferece as mesmas capacidades.

24.14.4 Apelidos para este domínio

É possível ter vários nomes de domínio que apontam para o mesmo diretório de email. Você pode adicionar quantos forem necessários. Por exemplo, alguém define o domínio virtual `foo.com` e depois registra o `foo.com.br`. Usando apelidos de domínio, ambos os domínios serão equivalentes.

24.15 Uma nota antes de iniciar

O domínio virtual de email é um suplemento às capacidades normais do email de seu servidor. O domínio de email normal ainda é gerenciado apropriadamente e os emails são guardados em `/var/spool/mail`.

Então, se sua máquina está correntemente aceitando emails para o domínio `foo.com.br` e você quer receber separados os emails para `foo1.com.br` e `foo2.com.br`, você apenas terá que definir estes dois domínios virtuais de email. A configuração de `foo.com.br` permanece inalterada.

24.16 Como definir um domínio virtual de email

24.16.1 Como adaptar os clientes POP

Nada de especial deve ser feito no lado do usuário POP, o que é uma coisa excelente: _queremos abrigar vários servidores de email em uma máquina, e não queremos que todos saibam disso_. :)

24.16.2 *Como instalar o servidor*

O truque para ler o email é exatamente o mesmo de um domínio virtual de WWW: você precisa de endereços IP. Suponha que se queira criar três domínios virtuais de email: va.foo.com.br, vb.foo.com.br e vc.foo.com.br. Pense neles como se você fosse instalar três servidores independentes, cada um servindo um único domínio. Isso é o que será descrito, e depois mostrado que esses três servidores podem ser colocados em apenas uma máquina.

No DNS

Do ponto de vista do DNS, tem-se um servidor de email por domínio. Então o registro MX de cada domínio será:

```
_ va.foo.com.br - > email.va.foo.com.br
_ vb.foo.com.br - > email.vb.foo.com.br
_ vc.foo.com.br - > email.vc.foo.com.br
```

Com o DNS, é isso o que se informa ao mundo e aos usuários de email. No mais, usuários do va.foo.com.br, buscarão seu email em email.va.foo.com.br, usuários do vb.foo.com.br, buscarão seu email em email.vb.foo.com.br, e usuários do vc.foo.com.br, buscarão seu email em email.vc.foo.com.br..Com essa con.guração, pode-se muito bem ter um servidor (real) por domínio de email (o estado corrente antes do domínio virtual de email).

Instalando os servidores

Para continuar a con.guração (seja real ou virtual), vá ao DNS e defina um número IP para cada servidor (esse é o ponto-chave). Serão usados números IP privados como exemplo, e todos estarão na mesma rede.

```
_ email.va.foo.com.br - > 172.16.0.1
_ email.vb.foo.com.br - > 172.16.0.2
_ email.vc.foo.com.br - > 172.16.0.3
```

Então pode-se instalar três servidores Linux com estes IPs e dizer ao Sendmail de cada um para aceitar um desses três domínios.

Tornando-se virtual

Ao invés de se instalar três servidores Linux, instale apenas um. Para cada domínio virtual de email, deve-se:

```
_ Definí-lo usando o Configurador Linux, o que envolve apenas nomear o domínio.
_ Definir um apelido de IP (com o Configurador Linux) para que a máquina responda por esse número IP também. Isso é feito no menu "Ambiente de rede/Apelidos de IP para máquinas virtuais"
```

_ Instalar o `/usr/lib/linuxconf/lib/vpop3d` como um substituto ao `/usr/sbin/in.pop3d` no `/etc/inetd.conf`. O `vpop3d` é um bom substituto ao `pop3d` mesmo que você não use domínios virtuais.

O apelido de IP é o ponto-chave. O protocolo POP não tem como identificar o destino de uma requisição, exceto com o número IP de destino. É por isso que clientes POP devem usar um nome diferente (na verdade um IP diferente) para ler as mensagens de domínios de email diferentes, o que é algo já esperado..

Como instalar o `vpop3d`

O `vpop3d` é um substituto ao servidor POP que você usa em sua distribuição?

Nem sempre. Várias distribuições vêm com um `pop3d` diferente, que tem suporte a NIS, PAM e outros métodos de autenticação. A melhor maneira de se ter suporte a tudo isso facilmente é deixar o servidor `pop3d` nativo cuidando do domínio de email principal, e o `vpop3d` gerenciando apenas os virtuais.

Para obter este resultado, simplesmente passe como um argumento ao `vpop3d` a rota do servidor `pop3` nativo. O `vpop3d` lhe dará o controle das requisições POP feitas ao domínio principal. Aqui vai um exemplo de como ajustar o `/etc/inetd.conf`:

```
pop-3 stream tcp nowait root /usr/sbin/tcpd /usr/lib/linuxconf/lib/vpop3d /usr/sbin/ipop3d
```

Então para instalá-lo, não retire a chamada do `pop3d`, mas simplesmente coloque o `/usr/lib/linuxconf/lib/vpop3d` na sua frente. Essa linha pode variar um pouco de distribuição para distribuição.

24.17 Como depurar uma configuração

Muitos erros cometidos na definição de domínios virtuais são relacionados ao DNS. Aqui vão alguns testes que você pode fazer para verificar se sua configuração está correta.

Um aviso: usar clientes POP (programa de email) não adianta para testar esse tipo de configuração. Esses programas não são muito informativos para esse tipo de tarefa. Ou funcionam ou não.

24.17.1 Checando o DNS

Para cada domínio virtual, você deve fazer algo no DNS. Usando-se o domínio `va.foo.com.br` como exemplo, aqui vão os passos:

O MX

O comando `nslookup -q=mx va.foo.com.br` deverá mostrar algo útil. Pelo menos o nome do servidor de email deve ser obtido com seus números IP. Ou será

email.va.foo.com.br, ou o nome oficial do servidor. O nome obtido não precisa fazer parte do domínio va.foo.com.br, apenas deve apontar para o servidor físico apropriado.

O servidor de email virtual

O email.va.foo.com.br deve ser definido no DNS. O comando `nslookup email.va.foo.com.br`

deve produzir um número IP. Além disso, o comando `nslookup o-número-IP-obtido` deve mostrar email.va.foo.com.br. Se você não obtiver isso como resposta, então o servidor POP virtual não funcionará, não mesmo e ponto final. Você precisa ter o mapeamento reverso correto desse número IP. O Configurador Linux faz isso automaticamente se o mapeamento especial do reverso do domínio estiver definido no mesmo DNS do domínio virtual. Então, se você teve a saída correta desses dois comandos, você já fez a parte pior. Há um servidor ouvindo?

Depois você faz o seguinte comando:
`telnet email.va.foo.com.br`

que deve se conectar ao servidor físico. Isso prova que o apelido de IP está instalado corretamente.

Há um servidor POP virtual ouvindo?

Execute o seguinte comando e veja se o `vpop3d` está instalado apropriadamente. Em caso afirmativo, nada mais pode dar errado.

`telnet email.va.foo.com.br pop-3`

Você deve obter

`+OK Virtual va.foo.com.br POP3 Server (Version 1.004) ready.`

O `_va.foo.com.br_` é o ponto-chave aqui. Se você não o obteve como resposta, então o domínio virtual ou não está definido, ou o `vpop3d` não está instalado no `/etc/inetd.conf`.

Uma ferramenta para se fazer uma checagem rápida

O script `/usr/lib/linuxconf/lib/checkvdomain` pode ser usado para se fazer uma checagem da instalação do `vpop3d` para um domínio virtual. Rode-o sem argumentos para aprender mais.

O script `/usr/lib/linuxconf/lib/testalldomain` lê o arquivo `/etc/named.boot.p` e extrai todos os domínios nele definidos. Então ele roda o script `checkvdomain` (assumindo que há uma máquina virtual de email para cada domínio) em todos os domínios. Ele diz então se o domínio está configurado apropriadamente ou não. É muito útil para um administrador que gerencia muitos domínios virtuais.

24.17.2 Como adicionar usuários POP

Uma vez que um domínio virtual de email está funcionando, você ainda tem que adicionar usuários POP a ele. Há uma entrada de menu em Contas de usuários chamada Contas POP virtuais (somente email) . Ele lhe deixa escolher um domínio virtual e adicionar ou editar contas novas nele.

24.17.3 Co-administrador

Para cada domínio virtual de email, há um privilégio novo adicionado ao Configurador Linux. Você pode garantir esse privilégio a qualquer usuário normal. Ele poderá gerenciar a lista de usuários (apenas usuários POP) do domínio virtual. Isso é totalmente operacional usando-se a interface HTML também.

24.17.4 Como um usuário pode trocar a sua senha

Um problema com os usuários POP (e também usuários PPP) é que eles não têm acesso a uma conta shell de onde eles poderiam facilmente trocar sua senha. O Configurador Linux provê uma solução a esse problema, mas está disponível apenas na interface HTML. Ela tem suporte total a domínios virtuais de email. Se você apontar seu navegador para a seguinte URL http://seu_servidor:98/htmlmod:userpass

você acessará uma tela simples que deixa qualquer um trocar sua própria senha. O `_seu_servidor_` pode ser qualquer um dos servidores POP virtuais ou o nome normal de seu servidor. O Configurador Linux gerenciará o arquivo de senhas apropriado baseado no número IP usado para alcançá-lo..p g ç

É uma boa idéia esconder essa URL em uma de suas páginas de informações de seu servidor (é meio chata de digitar).

24.18 Alguns problemas

24.18.1 Todos os emails do domínio saem mascarados

Isso é causado por uma má configuração do DNS. Todo email que sai do servidor é reescrito, então parece que veio do domínio principal do servidor. Esse problema se origina de uma funcionalidade nova do sendmail. No seu início, o sendmail varre todas as interfaces de rede (e apelidos de IP) e pega os nomes associados com seus números IP. Ele assume que todos esses nomes são equivalentes ao domínio principal do servidor. Qualquer email originário de um desses nomes será mascarado como vindo do domínio principal.

O problema no DNS é simplesmente que você definiu o mapeamento reverso de um número IP para um nome de domínio, ao invés de uma máquina deste domínio. O Configurador Linux não deixará você cometer esse erro, pois ele nunca faz um mapeamento

reverso que aponte para um nome de domínio. Mas no caso de se fazer o DNS na mão, está-se sujeito a isso. Então certifique-se de que todos os números IP associados a apelidos de IP apontam para uma máquina de um domínio, e não para o próprio domínio. Reinicie seu DNS e seu sendmail e tudo ficará correto.

24.19 Apelidos

Às vezes é necessário redirecionar o email de um usuário para outro(s) lugar(es), outro(s) usuário(s) ou até para programas.

24.19.1 Tarefas

Para um dado nome que pode ou não ser de um usuário real de seu sistema, você pode informar:

Um programa de filtro

É possível fazer um `_pipe_` de toda a mensagem para um programa, que pode ser um programa (`&:l`), um script shell ou uma construção shell como:

```
cat >> /tmp/mail.log
```

Este comando irá adicionar todas as mensagens que chegarem para o usuário no arquivo `/tmp/mail.log`.

É assim que se instala pacotes de servidores de email.

Arquivo de listagem

Você pode passar a administração de uma lista de discussão para alguém sem privilégios. Você pode especificar um arquivo texto que contenha endereços de email. Este arquivo pode pertencer a qualquer usuário. Um editor de texto simples é o necessário para gerenciá-lo.

Um ou vários endereços de email

Cada endereço de email pode ser simplesmente um nome de usuário, outro apelido (apelidos podem ser aninhados), ou um endereço de email completo. Você pode definir uma lista de discussão desta maneira. Estas funcionalidades são independentes. Isso significa que você pode definir um programa de filtro e um arquivo de listagem e vários endereços de email para um único apelido..

24.19.2 Pseudo-usuário

Às vezes é necessário conversar com alguém de uma empresa sem saber seu nome real, como por exemplo, apenas querer falar de um serviço. Para isto configura-se o pseudo-usuário.

24.19.3 *Lista dos pseudo-usuários mais conhecidos*

A Internet criou muitos pseudo-usuários. Aqui vai uma lista de alguns que você pode definir em seu sistema (no servidor de email).

- hostmaster
Cuida de seu DNS.
- postmaster
Este é o endereço que se usa quando se precisa de informações sobre alguém (o endereço de email correto) ou caso se queira apontar um problema relativo a email.
- usenet
Usado para apontar problemas no servidor news de seu sistema.
- webmaster
Cuida de seu site na Internet.

24.19.4 *Apelidos para contas administrativas*

O sistema Linux tem várias contas administrativas que recebem email. É no mínimo incômodo se ter que entrar no sistema como um desses usuários apenas para ver os emails. Por isso é uma boa idéia redirecionar esses emails para um usuário real (você?).

Aqui vai uma lista dessas contas:

- _ root
- _ uucp
- _ news.g ç p

24.19.5 *O pseudo-usuário administrador*

Numa empresa pequena, há apenas um administrador para toda a rede. Ao invés de definir todos aqueles apelidos para ele, economize tempo e aponte-os para um pseudo-usuário administrador que é ele mesmo um apelido para o administrador. Isso pode lhe economizar tempo.

24.19.6 *Onde definir os apelidos*

Os apelidos são geralmente definidos no servidor de email de uma empresa. É possível também definir apelidos em várias máquinas.

24.19.7 Administrador de listas de discussão

Você pode definir uma lista de discussão usando apelidos e arquivos com apelidos. Isso é útil e servirá para listas de discussão pequenas com tráfego limitado (sem muitos membros entrando/saindo). Numa lista de discussão, você eventualmente terá membros que têm o email mal configurado. Cada vez que alguém mandar um email para a lista, receberá respostas de erro relativas a esses usuários. Isso é ruim pois esse alguém nada tem a fazer (a ver) para resolver esse problema.

Aqui vai a dica. Se você criar uma lista de discussão, por exemplo, lista-produtos, então crie outro apelido owner-lista-produtos que aponte para você. O Sendmail mandará as mensagens de erro automaticamente para você ao invés de mandá-las para o remetente da mensagem.

24.20 Con.guração de filtros de spam

Com o crescimento da Internet, emails indesejados, também conhecidos como spam, viraram um problema. Não basta apenas proteger seus usuários contra emails spam vindos de sites conhecidos, você também precisa controlar quem pode usar seu servidor de email como repetidor. Alguns sujeitos inconvenientes que gostam de enviar emails spam, conhecidos como spammers, tentarão usar seu servidor de email para enviar spams para o mundo todo. Você pode controlar isso.

24.20.1 Princípio

Quando seu servidor de email recebe uma mensagem, há dois caminhos a seguir: ou o email é enviado a um de seus domínios e é aceito, ou o email não é destinado a qualquer um de seus domínios e o servidor de email tentará redirecioná-lo ao servidor de email apropriado.

Há uma terceira possibilidade: a mensagem é enviada a um de seus usuários, mas é originária de um remetente que sabe-se que é um spammer. Você pode excluí-la na hora.

24.20.2 Usar nomes ou números IP

Os emails podem vir de máquinas que são identificadas pelo nome e às vezes pelo número IP. Às vezes é mais prático classificar as máquinas pelo seu número IP ao invés do nome. Na seção seguinte você verá que às vezes pode informar um nome, às vezes um número IP e às vezes ambos.

24.20.3 *Especificando uma rede*

Você pode especificar um número de rede informando um prefixo de número IP para esta rede. Por exemplo, para a rede classe C 192.168.1.0, você informará 192.168.1.

24.20.4 *Especificando uma máquina ou um domínio*

Para especificar um nome de máquina, você deve informar seu nome completo, como: máquina1.domínio.com.br. Para especificar um domínio, simplesmente o informe. Isso significa que qualquer máquina que faça parte desse domínio será processada da mesma maneira.

24.21 *Definindo remetentes rejeitados*

Simplesmente informe uma lista de endereços de email, nomes de máquina, domínio ou números IP. Todas as possibilidades seguintes são aceitas:

- _ spammer@lixo.domínio.p g ç
- _ lixo.domínio
- _ D.X.Y.Z
- _ C.X.Y
- _ B.X
- _ A

Você também deve informar uma mensagem de erro que será enviada de volta ao remetente.

24.21.1 *Definindo _repetir para_ por IP*

Aqui você informa uma lista de números IP (máquinas ou redes) para quem seu servidor poderá repetir os emails. Geralmente coloca-se aqui os vários IPs da rede local.

24.21.2 *Definindo _repetir para_ por nome*

Aqui você informa uma lista de nomes de máquina e domínios para quem seu servidor poderá repetir os emails. Geralmente coloca-se aqui os vários IPs da rede.

24.21.3 *Definindo _repetir para_ máquinas*

Aqui você informa uma lista de servidores de email os quais usam o seu servidor

como roteador.

25 - Adicionando Conexões PPP / SLIP para Modem

A primeira coisa a definir na configuração de rede é se você está conectado a uma rede local, com um grupo de computadores em um escritório, ou a uma rede de grande abrangências, como a Internet. Antes de continuar é importante ainda saber qual hardware será utilizado para a conexão. Caso se utilize conexões via modem ou placas de rede, esteja seguro de que o hardware está adequadamente instalado e que os cabos estão corretamente conectados. Independente da especificação do tipo de rede que seja utilizada, caso os cabos e equipamentos não estejam bem conectados e configurados, nenhuma configuração fará o sistema funcionar. Iniciaremos pelas conexões via modem.

Há algumas informações que serão obtidas a partir de seu Provedor de Acesso Internet ou administrador de sistema antes de ter a sua conexão PPP ou SLIP funcionando. Estes são os dados para ter o seu *Conectiva Linux* conectado ao seu Provedor:

- O endereço IP para o servidor de nomes do domínio (DNS).
- O número de telefone de conexão.
- Nome de acesso e senha.
- Um endereço IP para sua máquina, caso a rede à qual esteja conectado não defina automaticamente um endereço dinâmico.
- Utilização (ou não) de métodos de conexão, ATSI como PAP, CHAP ou MS-CHAP. Nos casos afirmativos, é necessário conhecer a seqüência de caracteres ou palavra que compõe a chave de acesso. CHAP e MS-CHAP **não** são atualmente suportados por Linuxconf, e são raramente usados.

Informações adicionais podem ser úteis, mas não fundamentais, tais como endereço do servidor secundário e domínio de pesquisa. De posse de todas as informações você estará apto a conectar-se.

- Inicie o Linuxconf
- Informe a senha do superusuário quando solicitado (caso já não seja o `root`).
- Abra [Configuração] [Ambiente de Rede] [Tarefas de Cliente] [PPP / SLIP / PLIP].
- Selecione Adicionar.

Inicialmente não haverá qualquer configuração especificada. Ao selecionar a opção **Adicionar** serão apresentadas as opções PPP, SLIP e PLIP.

PPP é a interface mais comum e é a padrão. Para configurar uma conexão PPP selecione PPP e pressione Aceitar.

Você verá os seguintes campos:

- Número de telefone - nome utilizado para o acesso remoto.
- Porta do modem - indica a localização do modem. Já deve estar configurada.
- Caixa de verificação de uso de autenticação PAP - deve ser acionada caso o sistema de destino utilize PAP.

- Nome de acesso - nome usado na conexão ao sistema remoto.
- Senha: a senha da conta PPP.

Note que a janela tem o título Interface PPP `ppp0.ppp0` é a primeira interface PPP, `ppp1` será a segunda e assim por diante. É importante estar atento em relação a qual interface se está configurando. Conexões SLIP usam o prefixo `sl` ao invés de `ppp`. Com exceção da caixa de verificação de PAP, as opções da caixa de entrada são idênticas para os dois métodos.

Em quaisquer das interfaces que tenha sido escolhida informe o número completo do telefone da máquina remota e esteja seguro de incluir eventuais números para obter uma linha externa, caso se esteja ligando de um ramal de uma central telefônica. Por exemplo, caso seja necessário discar 0 e outro número 3322074, deve ser informado 03322074. No item porta do modem tem-se à disposição um menu. Caso se utilize a dupla inicialização Linux/Windows pode-se traçar uma relação entre os padrões do Linux e do DOS da seguinte forma:

- `cua0/ttyS0` - COM1 sob MS-DOS
- `cua1/ttyS1` - COM2 sob MS-DOS
- `cua2/ttyS2` - COM3 sob MS-DOS
- `cua3/ttyS3` - COM4 sob MS-DOS

O nome de acesso é o nome da conta PPP no servidor remoto. A senha deverá ser informada em texto simples. Ao finalizar com estas informações selecione o botão Personalizar no rodapé da tela. Todas as demais informações são solicitadas em diversas telas (Hardware, Comunicação, Ambiente de Rede e PAP).

Selecione a opção PAP e informe o nome de usuário e senha que o Provedor tenha disponibilizado no campo de segredo. Os demais padrões devem ser suficientes, mas caso necessário pode-se editar as configurações iniciais na opção Personalizar.