Sumário

Resumo

Este documento tem por objetivo ser uma referência ao aprendizado do usuário e um guia de consulta, operação e configuração de sistemas Linux (e outros tipos de *ix). A última versão deste guia pode ser encontrada na Página Oficial do Foca GNU/Linux (http://www.metainfo.org/focalinux). Novas versões são lançadas com uma freqüência mensal e você pode receber avisos de lançamentos preenchendo um formulário na página Web.

1 O Linux

O 'Linux' é um sistema operacional criado em 1991 por *Linus Torvalds* na universidade de Helsinki na Finlândia. É um sistema Operacional de código aberto distribuído gratuitamente pela Internet. Seu código fonte é liberado como *Free Software* (software livre) o aviso de copyright do kernel feito por Linus descreve detalhadamente isto e mesmo ele está proibido de fazer a comercialização do sistema.

Isto quer dizer que você não precisa pagar nada para usar o Linux, e não é crime fazer cópias para instalar em outros computadores, nós inclusive incentivamos você a fazer isto. Ser um sistema de código aberto pode explicar a performance, estabilidade e velocidade em que novos recursos são adicionados ao sistema.

Para rodar o 'Linux' você precisa, no mínimo, de um computador 386 SX com 2 MB de memória e 40MB disponíveis em seu disco rígido para uma instalação básica e funcional.

O sistema segue o padrão *POSIX* que é o mesmo usado por sistemas *UNIX* e suas variantes. Assim, aprendendo o 'Linux' você não encontrará muita dificuldade em operar um sistema do tipo 'UNIX, FreeBSD, HPUX, SunOS,' etc., bastando apenas aprender alguns detalhes encontrados em cada sistema.

O código fonte aberto permite que qualquer pessoa veja como o sistema funciona (útil para aprendizado), corrija alguma problema ou faça alguma sugestão sobre sua melhoria, esse é um dos motivos de seu rápido crescimento, do aumento da compatibilidade de periféricos (como novas placas sendo suportadas logo após seu lançamento) e de sua estabilidade.

Outro ponto em que ele se destaca é o suporte que oferece a placas, CD-Roms e outros tipos de dispositivos de última geração e mais antigos (a maioria deles já ultrapassados e sendo completamente suportados pelo sistema operacional). Este é um ponto forte para empresas que desejam manter seus micros em funcionamento e pretendem investir em avanços tecnológicos com as máquinas que possui.

Hoje o 'Linux' é desenvolvido por milhares de pessoas espalhadas pelo mundo, cada uma fazendo sua contribuição ou mantendo alguma parte do kernel gratuitamente. *Linus Torvalds* ainda trabalha em seu desenvolvimento e também ajuda na coordenação entre os desenvolvedores.

O suporte ao sistema também se destaca como sendo o mais eficiente e rápido do que qualquer programa comercial disponível no mercado.

Existem centenas de consultores especializados espalhados ao redor do mundo. Você pode se inscrever em uma lista de discussão e relatar sua dúvida ou alguma falha, e sua mensagem será vista por centenas de usuários na Internet e algum irá te ajudar ou avisará as pessoas responsáveis sobre a falha encontrada para devida correção.

1.1 Algumas Características do Linux

- É de graça e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional. Muitos deles estavam cansados do excesso de propaganda (Marketing) e baixa qualidade de sistemas comerciais existentes
- Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o 'DOS', 'Windows', 'OS/2') no mesmo computador.

- Multitarefa real
- Multiusuário
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres)
- Conectividade com outros tipos de plataformas como *Apple, Sun, Macintosh, Sparc, Alpha, PowerPc, ARM, Unix, Windows, DOS, etc.*
- Proteção entre processos executados na memória RAM
- Suporte há mais de 63 terminais virtuais (consoles)
- Modularização - O 'GNU/Linux' somente carrega para a memória o que é usado durante o processamento, liberando totalmente a memória assim que o programa/dispositivo é finalizado
- Devido a modularização, os drivers dos periféricos e recursos do sistema podem ser carregados e removidos completamente da memória RAM a qualquer momento. Os drivers (módulos) ocupam pouco espaço quando carregados na memória RAM (cerca de 6Kb para a Placa de rede NE 2000, por exemplo)
- Não há a necessidade de se reiniciar o sistema após a modificar a configuração de qualquer periférico ou parâmetros de rede. Somente é necessário reiniciar o sistema no caso de uma instalação interna de um novo periférico, falha em algum hardware (queima do processador, placa mãe, etc.).
- Não precisa de um processador potente para funcionar. O sistema roda bem em computadores 386sx 25 com 4MB de memória RAM (sem rodar o sistema gráfico X, que é recomendado 8MB de RAM). Já pensou no seu desempenho em um 486 ou Pentium ;-)
- O crescimento e novas versões do sistema não provocam lentidão, pelo contrário, a cada nova versão os desenvolvedores procuram buscar maior compatibilidade, acrescentar recursos úteis e melhor desempenho do sistema (como o que aconteceu na passagem do kernel 2.0.x para 2.2.x).
- Não é requerida uma licença para seu uso. O 'GNU/Linux' é licenciado de acordo com os termos da GNU.
- Acessa sem problemas discos formatados pelo 'DOS, Windows, Novell, OS/2, NTFS, SunOS, Amiga, Atari, Mac,' etc.
- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
- NÃO EXISTEM VÍRUS NO LINUX! Em 9 anos de existência, nunca foi registrado NENHUM tipo de vírus neste sistema. Isto tudo devido a grande segurança oferecida pelas permissões de acesso do sistema que funcionam inclusive durante a execução de programas.
- Rede TCP/IP mais rápida que no Windows e tem sua pilha constantemente melhorada. O 'GNU/Linux' tem suporte nativo a redes TCP/IP e não depende de uma camada intermediária como o Winsock. Em acessos via modem a Internet, a velocidade de transmissão é 10% maior. Jogadores do 'Quake' ou qualquer outro tipo de jogo via Internet preferem o 'GNU/Linux' por causa da maior velocidade do Jogo em rede. É fácil rodar um servidor 'Quake' em seu computador e assim jogar contra vários adversários via Internet.
- Roda aplicações *DOS* através do 'DOSEMU'. Para se ter uma idéia, é possível dar o boot em um sistema *DOS* qualquer dentro dele e ao mesmo tempo usar a multitarefa deste sistema.
- Suporte a dispositivos infravermelho.
- Suporte a rede via rádio amador.
- Suporte a dispositivos Plug-and-Play.
- Suporte a dispositivos USB.
- Vários tipos de firewalls de alta qualidade e com grande poder de segurança de graça.
- Roteamento estático e dinâmico de pacotes.
- Ponte entre Redes.
- Proxy Tradicional e Transparente.

- Possui recursos para atender a mais de um endereço IP na mesma placa de rede, sendo muito útil para situações de manutenção em servidores de redes ou para a emulação de "mais computadores" virtualmente. O servidor WEB e FTP podem estar localizados no mesmo computador, mas o usuário que se conecta tem a impressão que a rede possui servidores diferentes.
- O sistema de arquivos usado pelo 'GNU/Linux' ('Ext2') organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.
- Permite a montagem de um servidor Web, E-mail, News, etc. com um baixo custo e alta performance. O melhor servidor Web do mercado, o 'Apache', é distribuído gratuitamente junto com o Linux.
- Por ser um sistema operacional de código aberto, você pode ver o que o código fonte (o que foi digitado pelo programador) faz e adapta-lo as suas necessidades ou de sua empresa. Esta característica é uma segurança a mais para empresas sérias e outros que não querem ter seus dados roubados (você não sabe o que um sistema sem código fonte faz na realidade enquanto esta processando o programa).
- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Pode ser executado em 10 arquiteturas diferentes (Intel, Macintosh, Alpha, Arm, etc.).
- Consultores técnicos especializados no suporte ao sistema espalhados por todo o mundo.
- Entre muitas outras características que você descobrirá durante o uso do sistema.

TODOS OS ÍTENS DESCRITOS ACIMA SÃO VERDADEIROS E TESTADOS PARA QUE TIVESSE PLENA CERTEZA DE SEU FUNCIONAMENTO.

1.2 Software Livre

Softwares Livres são programas que possuem o código fonte incluído (o código fonte é o que o programador digitou para fazer o programa) e você pode modificar ou distribui-los livremente. Existem algumas licenças que permitem isso, a mais comum é a *General Public Licence* (ou GPL).

Os softwares livres muitas vezes são chamados de *programas de código aberto* (ou OSS). Muito se acredita no compartilhamento do conhecimento e tendo liberdade de cooperar uns com outros, isto é importante para o aprendizado de como as coisas funcionam e novas técnicas de construção. Existe uma longa teoria desde 1950 valorizando isto, muitas vezes pessoas assim são chamadas de "Hackers Éticos".

Outros procuram aprender mais sobre o funcionamento do computador e seus dispositivos (periféricos) e muitas pessoas estão procurando por meios de de evitar o preço absurdo de softwares comerciais através de programas livres que possuem qualidade igual ou superior, devido a cooperação em seu desenvolvimento.

Você pode modificar o código fonte de um software livre a fim de melhora-lo ou acrescentar mais recursos e o autor do programa pode ser contactado sobre a alteração e os benefícios que sua modificação fez no programa, e esta poderá ser incluída no programa principal. Deste modo, milhares de pessoas que usam o programa se beneficiarão de sua contribuição.

1.3 Estrutura básica de diretórios do Sistema Linux

O sistema 'GNU/Linux' possui a seguinte estrutura básica de diretórios:

'/bin' Contém arquivos programas do sistema que são usados com freqüência pelos usuários.

'/boot' Contém arquivos necessários para a inicialização do sistema.

'/cdrom' Ponto de montagem da unidade de CD-ROM.

'/dev' Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.

'/etc' Arquivos de configuração de seu computador local.

'/floppy' Ponto de montagem de unidade de disquetes

'/home' Diretórios contendo os arquivos dos usuários.

'/lib' Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.

'/lost+found' Local para a gravação de arquivos/diretórios recuperados pelo utilitário 'fsck.ext2'. Cada partição possui seu próprio diretório 'lost+found'.

'/mnt' Ponto de montagem temporário.

‘/proc’ Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.

‘/root’ Diretório do usuário ‘root’.

‘/sbin’ Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.

‘/tmp’ Diretório para armazenamento de arquivos temporários criados por programas.

‘/usr’ Contém maior parte de seus programas. Normalmente acessível somente como leitura.

‘/var’ Contém maior parte dos arquivos que são gravados com freqüência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

1.4 Comandos

Comandos são ordens que passamos ao sistema operacional para executar uma determinada tarefa.

Cada comando tem uma função específica, devemos saber a função de cada comando e escolher o mais adequado para fazer o que desejamos, por exemplo:

- ‘ls’ - Mostra arquivos de diretórios
- ‘cd’ - Para mudar de diretório

Este guia tem uma lista de vários comandos organizados por categoria com a explicação sobre o seu funcionamento e as opções aceitas (incluindo alguns exemplos).

É sempre usado um espaço depois do comando para separa-lo de uma opção ou parâmetro que será passado para o processamento. Um comando pode receber opções e parâmetros:

Opções

As opções são usadas para controlar como o comando será executado, por exemplo, para fazer uma listagem mostrando o _dono, grupo, tamanho dos arquivos_ você deve digitar ‘ls -l’. Opções podem ser passadas ao comando através de um "-" ou "–":

- Opção identificada por uma letra. Podem ser usadas mais de uma opção com um único hífen. O comando ‘ls -l -a’ é a mesma coisa de ‘ls -la’

– Opção identificada por um nome. O comando ‘ls –all’ é equivalente a ‘ls -a’. Pode ser usado tanto "-" como "–", mas há casos em que somente "-" ou "–" esta disponível.

Parâmetros

Um parâmetro identifica o caminho, origem, destino, entrada padrão ou saída padrão que será passada ao comando. Se você digitar: ‘ls /usr/doc/copyright’, ‘/usr/doc/copyright’ será o parâmetro passado ao comando ‘ls’, neste caso queremos que ele liste os arquivos do diretório /usr/doc/copyright . É normal errar o nome de comandos, mas não se preocupe, quando isto acontecer o sistema mostrará a mensagem ‘command not found’ (comando não encontrado) e voltará ao aviso de comando. As mensagens de erro não fazem nenhum mal ao seu sistema, somente dizem que algo deu errado para que você possa corrigir e entender o que aconteceu. No ‘GNU/Linux’, você tem a possibilidade de criar comandos personalizados usando outros comandos mais simples (isto será visto mais adiante). Os comandos se encaixam em duas categorias: Comandos Internos e Comandos Externos.

Por exemplo: ‘"ls -la /usr/doc"’, ‘ls’ é o comando, ‘-la’ é a opção passada ao comando, e “/usr/doc” é o diretório passado como parâmetro ao comando ‘ls’.

1.4.1 Comandos Internos

São comandos que estão localizados dentro do interpretador de comandos (normalmente o ‘Bash’) e não no disco. Eles são carregados na memória RAM do computador junto com o interpretador de comandos.

Quando executa um comando, o interpretador de comandos verifica primeiro se ele é um Comando Interno caso não seja é verificado se é um Comando Externo.

Exemplos de comandos internos são: ‘cd, exit, echo, bg, fg, source, help’

1.4.2 Comandos Externos

São comandos que estão localizados no disco. Os comandos são procurados no disco usando o ‘path’ e executados assim que encontrados.

1.5 Aviso de comando (Prompt)

Aviso de comando (ou Prompt), é a linha mostrada na tela para *digitação de comandos* que serão passados ao ‘interpretador de comandos’ para sua execução.

A posição onde o comando será digitado é marcado um "traço" piscante na tela chamado de *cursor*. Tanto em shells texto como em gráficos é necessário o uso do cursor para sabermos onde iniciar a digitação de textos e nos orientarmos quanto a posição na tela.

O aviso de comando do usuário ‘root’ é identificado por uma "#" (tralha), e o aviso de comando de usuários é identificado pelo símbolo "$". Isto é padrão em sistemas ‘UNIX’.

Você pode retornar comandos já digitados pressionando as teclas ‘Seta para cima’ / ‘Seta para baixo’.

A tela pode ser rolada para baixo ou para cima segurando a tecla ‘SHIFT’ e pressionando ‘PGUP’ ou ‘PGDOWN’. Isto é útil para ver textos que rolaram rapidamente para cima.

Abaixo algumas dicas sobre a edição da linha de comandos (não é necessário se preocupar em decora-los):

- Pressione a tecla ‘Backspace’ ("<–") para apagar um caracter à esquerda do cursor.
- Pressione a tecla ‘Del’ para apagar o caracter acima do cursor.
- Pressione ‘CTRL’+‘A’ para mover o cursor para o inicio da linha de comandos.
- Pressione ‘CTRL’+‘E’ para mover o cursor para o fim da linha de comandos.
- Pressione ‘CTRL’+‘U’ para apagar o que estiver à esquerda do cursor. O conteúdo apagado é copiado para uso com ‘CTRL’+‘y’.
- Pressione ‘CTRL’+‘K’ para apagar o que estiver à direita do cursor. O conteúdo apagado é copiado para uso com ‘CTRL’+‘y’.
- Pressione ‘CTRL’+‘L’ para limpar a tela e manter o texto que estiver sendo digitado na linha de comando (parecido com o comando ‘clear’).
- Pressione ‘CTRL’+‘Y’ para colocar o texto que foi apagado na posição atual do cursor.

1.6 Interpretador de comandos

Também conhecido como "shell". É o programa responsável em interpretar as instruções enviadas pelo usuário e seus programas ao sistema operacional (o kernel). Ele que executa comandos lidos do dispositivo de entrada padrão (teclado) ou de um arquivo executável. É a principal ligação entre o usuário, os programas e o kernel. O ‘GNU/Linux’ possui diversos tipos de interpretadores de comandos, entre eles posso destacar o ‘bash, ash, csh, tcsh, sh,’ etc. Entre eles o mais usado é o ‘bash’. O interpretador de comandos do DOS, por exemplo, é o ‘command.com’.

Os comandos podem ser enviados de duas maneiras para o interpretador: ‘interativa’ e ‘não-interativa’:

‘Interativa’ Os comandos são digitados no aviso de comando e passados ao interpretador de comandos um a um. Neste modo, o computador depende do usuário para executar uma tarefa, ou próximo comando.

‘Não-interativa’ São usados arquivos de comandos criados pelo usuário (scripts) para o computador executar os comandos na ordem encontrada no arquivo. Neste modo, o computador executa os comandos do arquivo um por um e dependendo do término do comando, o script pode checar qual será o próximo comando que será executado e dar continuidade ao processamento.

Este sistema é útil quando temos que digitar por várias vezes seguidas um mesmo comando ou para compilar algum programa complexo.

O shell ‘Bash’ possui ainda outra característica interessante: A completação dos nomes. Isto é feito pressionando-se a tecla ‘TAB’. Por exemplo, se digitar "ls tes" e pressionar <tab>, o ‘Bash’ localizará todos os arquivos que iniciam com "tes" e completará o restante do nome. Caso a completação de nomes encontre mais do que uma expressão que satisfaça a pesquisa, ou nenhuma, é emitido um beep. A completação de nomes funciona sem problemas para comandos internos.

Exemplo: ‘ech’ (pressione ‘TAB’). ‘ls /vm’(pressione ‘TAB’)

1.7 Terminal Virtual (console)

Terminal (ou console) é o teclado e tela conectados em seu computador. O ‘GNU/Linux’ faz uso de sua característica *multi-usuária* usando os "terminais virtuais". Um terminal virtual é uma segunda seção de trabalho completamente independente de outras, que pode ser acessada no computador local ou remotamente via ‘telnet, rsh, rlogin,’ etc.

No ‘GNU/Linux’, em modo texto, você pode acessar outros terminais virtuais segurando a tecla ‘ALT’ e pressionando ‘F1 a F6’. Cada tecla de função corresponde a um número de terminal do 1 ao 6 (o sétimo é usado por padrão pelo ambiente gráfico X). O ‘GNU/Linux’ possui mais de 63 terminais virtuais, mas apenas 6 estão disponíveis inicialmente por motivos de economia de memória RAM .

Se estiver usando o modo gráfico, você deve segurar ‘CTRL’+ ‘ALT’ enquanto pressiona uma tela de <F1> a <F6>.

Um exemplo prático: Se você estiver usando o sistema no Terminal 1 com o nome "joao" e desejar entrar como "root" para instalar algum programa, segure ‘ALT’ enquanto pressiona <F2> para abrir o segundo terminal virtual e faça o login como "root". Será aberta uma nova seção para o usuário "root" e você poderá retornar a hora que quiser para o primeiro terminal pressionando ‘ALT’+<F1>.

1.8 Coringas

Coringas (ou referência global) é um recurso usado para especificar um ou mais arquivos ou diretórios do sistema de uma só vez. Este é um recurso permite que você faça a filtragem do que será listado, copiado, apagado, etc. São usados 3 tipos de coringas no ‘GNU/Linux’:

- "*" - Faz referência a um nome completo/restante de um arquivo/diretório.
- "?"- Faz referência a uma letra naquela posição.
- ‘[padrão]’ - Faz referência a um padrão contido em um arquivo. Padrão pode ser:
- ‘[a-z][1-0]’ - Faz referência a caracteres de ‘a’ até ‘z’ ou de ‘1’ até ‘10’.
- ‘[a,z][1,0]’ - Faz a referência aos caracteres ‘a’ e ‘z’ ou ‘1’ e ‘10’ naquela posição.
- ‘[a-z,1,0]’ - Faz referência aos caracteres de ‘a’ até ‘z’ e ‘1’ e ‘10’ naquela posição.

A procura de caracteres é "Case Sensitive" assim se você deseja que sejam localizados todos os caracteres alfabéticos você deve usar ‘[a-zA-Z]’.

Caso a expressão seja seguida de um ‘^’, faz referência a qualquer caracter exceto o da expressão. Por exemplo ‘[^abc]’ faz referência a qualquer caracter exceto ‘a’, ‘b’ e ‘c’.

Lembrando que os 3 tipos de coringas ("*", "?" e "[]") podem ser usados juntos. Para entender melhor vamos a prática:

Vamos dizer que tenha 5 arquivo no diretório ‘/usr/teste’:

‘teste1.txt, teste2.txt, teste3.txt, teste4.new, teste5.new’.

Caso deseje listar *todos* os arquivos do diretório ‘/usr/teste’ você pode usar o coringa "*" para especificar todos os arquivos do diretório:

‘cd /usr/teste’ e ‘ls *’ ou ‘ls /usr/teste/*’.

Não tem muito sentido usar o comando ‘ls’ com "*" porque todos os arquivos serão listados se o ‘ls’ for usado sem nenhum Coringa.

Agora para listar todos os arquivos ‘teste1.txt, teste2.txt, teste3.txt’ com excessão de ‘teste4.new’, ‘teste5.new’, podemos usar inicialmente 3 métodos:

1. Usando o comando ‘ls *.txt’ que pega todos os arquivos que começam com qualquer nome e terminam com ‘.txt’.
2. Usando o comando ‘ls teste?.txt’, que pega todos os arquivos que começam com o nome ‘teste’, tenham qualquer caracter no lugar do coringa ‘?’ e terminem com ‘.txt’. Com o exemplo acima ‘teste*.txt’ também faria a mesma coisa, mas se também tivéssemos um arquivo chamado ‘teste10.txt’ este também seria listado.
3. Usando o comando ‘ls teste[1-3].txt’, que pega todos os arquivos que começam com o nome ‘teste’, tenham qualquer caracter entre o número 1-3 no lugar da 6a letra e terminem com ‘.txt’. Neste caso se obtém uma filtragem mais exata, pois o coringa _?_ especifica qualquer caracter naquela posição e [] especifica números, letras ou intervalo que será usado.

Agora para listar somente ‘teste4.new’ e ‘teste5.new’ podemos usar os seguintes métodos:

1. ‘ls *.new’ que lista todos os arquivos que terminam com ‘.new’
2. ‘ls teste?.new’ que lista todos os arquivos que começam com ‘teste’, contenham qualquer caracter na posição do coringa _?_ e terminem com ‘.new’.
3. ‘ls teste[4,5].*’ que lista todos os arquivos que começam com ‘teste’ contenham números de 4 e 5 naquela posição e terminem com qualquer extensão.

Existem muitas outras formas de se fazer a mesma coisa, isto depende do gosto de cada um. O que pretendi fazer aqui foi mostrar como especificar mais de um arquivo de uma só vez. O uso de coringas será útil ao copiar arquivos, apagar, mover, renomear, e nas mais diversas partes do sistema. Alias esta é uma característica do ‘GNU/Linux’: permitir que a mesma coisa possa ser feita com liberdade de várias maneiras diferentes.

2 Relação entre DOS e GNU/Linux

2.1 Relacionando dispositivos

Os dispositivos também são identificados e uma forma diferente que no 'DOS' por exemplo:

DOS/Windows	Linux
A:	/dev/fd0
B:	/dev/fd1
C:	/dev/hda1
LPT1	/dev/lp0
LPT2	/dev/lp1
LPT3	/dev/lp2
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3

2.2 Comandos equivalentes entre DOS e Linux

Esta seção contém os comandos equivalentes entre estes dois sistemas e a avaliação entre ambos. Grande parte dos comandos podem ser usados da mesma forma que no 'DOS', mas os comandos 'Linux' possuem avanços para utilização neste ambiente multiusuário/multitarefa.

O objetivo desta seção é permitir as pessoas com experiência em 'DOS' fazer rapidamente no 'GNU/Linux' as tarefas que fazem no 'DOS'. A primeira coluna tem o nome do comando no 'DOS', a segunda o comando que possui a mesma função no 'GNU/Linux' e na terceira coluna as diferenças.

DOS	Linux	Diferenças
cls	clear	Sem diferenças
dir	ls -la	A listagem no Linux possui mais campos (as permissões de acesso) e o total de espaço ocupado no diretório e livre no disco deve ser visto separadamente usando o comando du e df. Permite também listar o conteúdo de diversos diretórios com um só comando (ls /bin /sbin /...).
dir/s	ls -lR	Sem diferenças.
dir/od	ls -tr	Sem diferenças.
cd	cd	Poucas diferenças. cd sem parâmetros retorna ao diretório de usuário e também permite o uso de "cd -" para retornar ao diretório anteriormente acessado.
del	rm	Poucas diferenças. O rm do Linux permite especificar diversos arquivos que serão apagados (rm arquivo1 arquivo2 arquivo3). Para ser mostrados os arquivos apagados, deve-se especificar o parâmetro "-v" ao comando, e "-i" para pedir a confirmação ao apagar arquivos.
md	mkdir	Uma só diferença: No Linux permite que vários diretórios sejam criados de uma só vez (mkdir /tmp/a /tmp/b...).
copy	cp	Poucas diferenças. Para ser mostrados os arquivos enquanto estão sendo copiados, deve-se usar a opção "-v", e para que ele pergunte se deseja substituir um arquivo já existente, deve-se usar a opção "-i".
echo	echo	Sem diferenças.
path	path	No Linux deve ser usado ":" para separar os diretórios e usar o comando "export PATH=caminho1:/caminho2:/caminho3:" para definir a variável de ambiente PATH. O path atual pode ser visualizado através do comando "echo $PATH".
ren	mv	Poucas diferenças. No Linux não é possível renomear vários arquivos de uma só vez (como "ren *.txt *.bak"). É necessário usar um shell script para fazer isto.
type	cat	Sem diferenças.
ver	uname -a	Poucas diferenças (o uname tem algumas opções a mais).
date	date	No Linux mostra/modifica a Data e Hora do sistema.
time	date	No Linux mostra/modifica a Data e Hora do sistema.
attrib	chmod	O chmod possui mais opções por tratar as permissões de acesso de leitura, gravação e execução para donos, grupos e outros usuários.
scandisk	e2fsck	O fsck é mais rápido e extensivo na checagem.
doskey	——	A edição de teclas é feita automaticamente pelo bash.

edit	vi, ae, emacs	O edit é mais fácil de usar, mas usuário experientes apreciarão os recursos do vi ou o emacs (programado em lisp).
fdisk	fdisk, cfdisk	Os particionadores do Linux trabalham com praticamente todos os tipos de partições de diversos sistemas de arquivos diferentes.
format	mkfs.ext2	Poucas diferenças, precisa apenas que seja especificado o dispositivo a ser formatado como "/dev/fd0" ou "/dev/hda10" (o tipo de identificação usada no Linux), ao invés de "A:" ou "C:".
help	man, info	Sem diferenças.
interlnk	plip	O plip do Linux permite que sejam montadas redes reais a partir de uma conexão via Cabo Paralelo ou Serial. A máquina pode fazer tudo o que poderia fazer conectada em uma rede (na realidade é uma rede e usa o TCP/IP como protocolo) inclusive navegar na Internet, enviar e-mails, irc, etc.
intersvr	plip	Mesmo que o acima.
keyb	loadkeys	Sem diferenças (somente que a posição das teclas do teclado pode ser editada. Desnecessário para a maioria dos usuários).
label	e2label	É necessário especificar a partição que terá o nome modificado.
mem	cat /proc/meminfo	Mostra detalhes sobre a quantidade de dados em buffers, cache e memória virtual (disco).
more	more, less	O more é equivalente a ambos os sistemas, mas o less permite que sejam usadas as setas para cima e para baixo, o que torna a leitura do texto muito mais agradável.
move	mv	Poucas diferenças. Para ser mostrados os arquivos enquanto estão sendo movidos, deve-se usar a opção "-v", e para que ele pergunte se deseja substituir um arquivo já existente deve-se usar a opção "-i".
scan	——	Não existem vírus no Linux devido as restrições do usuário durante execução de programas.
backup	tar	O tar permite o uso de compactação (através do parâmetro -z) e tem um melhor esquema de recuperação de arquivos corrompidos que já segue evoluindo há 30 anos em sistemas UNIX.
print	lpr	O lpr é mais rápido e permite até mesmo impressões de gráficos ou arquivos compactados diretamente caso seja usado o programa magicfilter. É o programa de Spool de impressoras usados no sistema Linux/Unix.
vol	e2label	Sem diferenças.
xcopy	cp -R	Pouca diferença, requer que seja usado a opção "-v" para mostrar os arquivos que estão sendo copiados e "-i" para pedir confirmação de substituição de arquivos.

2.3 Usando a sintaxe de comandos DOS no Linux

Você pode usar os comandos do pacote ‘mtools’ para simular os comandos usados pelo ‘DOS’ no ‘GNU/Linux’, a diferença básica é que eles terão a letra ‘m’ no inicio do nome. Os seguintes comandos são suportados:

- ‘mattrib’ - Ajusta modifica atributos de arquivos
- ‘mcat’ - Mostra os dados da unidade de disquete em formato RAW
- ‘mcd’ - Entra em diretórios
- ‘mcopy’ - Copia arquivos/diretórios
- ‘mdel’ - Exclui arquivos
- ‘mdeltree’ - Exclui arquivos, diretórios e sub-diretórios
- ‘mdir’ - Lista arquivos e diretórios
- ‘mdu’ - Mostra o espaço ocupado pelo diretório do DOS
- ‘mformat’ - Formatador de discos
- ‘minfo’ - Mostra detalhes sobre a unidade de disquetes
- ‘mlabel’ - Cria um volume para unidades DOS

- ‘mmd’ - Cria diretórios
- ‘mmount’ - Monta discos DOS
- ‘mmove’ - Move ou renomeia arquivos/subdiretórios
- ‘mpartition’ - Particiona um disco para ser usado no DOS
- ‘mrd’ - Remove um diretório
- ‘mren’ - Renomeia arquivos
- ‘mtype’ - Visualiza o conteúdo de arquivos (equivalente ao cat)
- ‘mtoolstest’ - Exibe a configuração atual do ‘mtools’
- ‘mshowfat’ - Mostra a FAT da unidade
- ‘mbadblocks’ - Procura por setores defeituosos na unidade
- ‘mzip’ - Altera modo de proteção e ejeta discos em unidades Jaz/ZIP
- ‘mkmanifest’ - Cria um shell script para restaurar nomes extensos usados no UNIX
- ‘mcheck’ - Verifica arquivos na unidade

2.4 Formatando disquetes

As subseções seguintes explicarão maneiras de formatar seus disquetes para serem usados no ‘GNU/Linux’ e ‘DOS/Windows’.

2.4.1 Formatando disquetes para serem usados no Linux

Para formatar disquetes para serem usados no ‘GNU/Linux’ use o comando:

‘mkfs.ext2 [*-c*] [*/dev/fd0*]’

Em alguns sistemas você deve usar ‘mke2fs’ no lugar de ‘mkfs.ext2’. A opção ‘-c’ faz com que o ‘mkfs.ext2’ procure por blocos danificados no disquete e ‘/dev/fd0’ especifica a primeira unidade de disquetes para ser formatada (equivalente a ‘A:’ no DOS). Mude para ‘/dev/fd1’ para formatar um disquete da segunda unidade.

OBS: Este comando cria um sistema de arquivos *ext2* no disquete que é nativo do ‘GNU/Linux’ e permite usar características como permissões de acesso e outras. Isto também faz com que o disquete NÃO possa ser lido pelo ‘DOS/Windows’. Para formatar um disquete no ‘GNU/Linux’ usando o *FAT12* (compatível com o DOS/Windows) veja próxima seção.

Exemplo: ‘mkfs.ext2 -c /dev/fd0’

2.4.2 Formatando disquetes compatíveis com o DOS/Windows

A formatação de disquetes ‘DOS’ no ‘GNU/Linux’ é feita usando o comando ‘superformat’ que é geralmente incluído no pacote ‘mtools’. O ‘superformat’ formata (cria um sistema de arquivos) um disquete para ser usado no ‘DOS’ e também possui opções avançadas para a manipulação da unidade, formatação de intervalos de cilindros específicos, formatação de discos em alta capacidade e verificação do disquete.

‘superformat [*opções*] [*dispositivo*]’

dispositivo Unidade de disquete que será formatada. Normalmente ‘/dev/fd0’ ou ‘/dev/fd1’ especificando respectivamente a primeira e segunda unidade de disquetes.

opções

-v [num] Especifica o nível de detalhes que serão exibidos durante a formatação do disquete. O nível 1 especifica um ponto mostrado na tela para cada trilha formatada. Veja a página de manual do ‘superformat’ para detalhes.

-superverify Verifica primeiro se a trilha pode ser lida antes de formata-la. Este é o padrão.

–dosverify, -B Verifica o disquete usando o utilitário ‘mbadblocks’. Usando esta opção, as trilhas defeituosas encontradas serão automaticamente marcadas para não serem utilizadas.

–verify_later, -V Verifica todo o disquete no final da formatação.

–noverify, -f Não faz verificação de leitura.

Na primeira vez que o 'superformat' é executado, ele verifica a velocidade de rotação da unidade e a comunicação com a placa controladora, pois os discos de alta densidade são sensíveis a rotação da unidade. Após o teste inicial ele recomendará adicionar uma linha no arquivo '/etc/driveprm' como forma de evitar que este teste seja sempre executado.

OBS: Esta linha é calculada de acordo com a rotação de usa unidade de disquetes, transferência de dados e comunicação com a placa controladora de disquete. Desta forma ela varia de computador para computador Note que não é necessário montar a unidade de disquetes para formata-la.

Segue abaixo exemplos de como formatar seus disquetes com o 'superformat':

- 'superformat /dev/fd0' - Formata o disquete na primeira unidade de disquetes usando os valores padrões.
- 'superformat /dev/fd0 dd' - Faz a mesma coisa que o acima, mas assume que o disquete é de Dupla Densidade (720Kb).
- 'superformat -v 1 /dev/fd0' - Faz a formatação da primeira unidade de disquetes ('/dev/fd0') e especifica o nível de detalhes para 1, exibindo um 'ponto' após cada trilha formatada.

2.4.3 Programas de Formatação Gráficos

Além de programas de formatação em modo texto, existem outros para ambiente gráfico (X11) que permitem fazer a mesma tarefa.

Entre os diversos programas destaco o 'gfloppy' que além de permitir selecionar se o disquete será formatado para o 'GNU/Linux' (ext2) ou 'DOS' (FAT12), permite selecionar a capacidade da unidade de disquetes e formatação rápida do disco.

2.5 Pontos de Montagem

O 'GNU/Linux' acessa as partições existente em seus discos rígidos e disquetes através de diretórios. Os diretórios que são usados para acessar (montar) partições são chamados de _Pontos de Montagem_. Para detalhes sobre montagem de partições, veja Seção 4.5, 'Montando (acessando) uma partição de disco'.

No 'DOS' cada letra de unidade (C:, D:, E:) identifica uma partição de disco, no 'GNU/Linux' os pontos de montagem fazem parte da grande estrutura do sistema de arquivos raiz.

2.6 Identificação de discos e partições em sistemas Linux

No 'GNU/Linux', os dispositivos existentes em seu computador (como discos rígidos, disquetes, tela, portas de impressora, modem, etc) são identificados por um arquivo referente a este dispositivo no diretório '/dev'.

A identificação de discos rígidos no 'GNU/Linux' é feita da seguinte forma:

```
/dev/hda1
|    | ||
|    | ||_Número que identifica o número da partição no disco rígido.
|    | |
|    | |_Letra que identifica o disco rígido (a=primeiro, b=segundo, etc...).
|    |
|    |_Sigla que identifica o tipo do disco rígido (hd=ide, sd=SCSI, xd=XT).
|
|_Diretório onde são armazenados os dispositivos existentes no sistema.
```

Abaixo algumas identificações de discos e partições em sistemas Linux:

- */dev/fd0* - 'Primeira unidade de disquetes'.
- */dev/fd1* - 'Segunda unidade de disquetes'.
- */dev/hda* - 'Primeiro disco rígido na primeira controladora IDE do micro (primary master)'.
- */dev/hda1* - 'Primeira partição do primeiro disco rígido IDE'.
- */dev/hdb* - 'Segundo disco rígido na primeira controladora IDE do micro (primary slave)'.
- */dev/hdb1* - 'Primeira partição do segundo disco rígido IDE'.
- */dev/sda* - 'Primeiro disco rígido na primeira controladora SCSI'.

- */dev/sda1* - ‘Primeira partição do primeiro disco rígido SCSI’.
- */dev/sdb* - ‘Segundo disco rígido na primeira controladora SCSI’.
- */dev/sdb1* - ‘Primeira partição do segundo disco rígido SCSI’.
- */dev/sr0* - ‘Primeiro CD-ROM SCSI’.
- */dev/sr1* - ‘Segundo CD-ROM SCSI’.
- */dev/xda* - ‘Primeiro disco rígido XT’.
- */dev/xdb* - ‘Segundo disco rígido XT’.

As letras de identificação de discos rígidos podem ir além de ‘hdb’, em meu micro, por exemplo, a unidade de CD-ROM está localizada em ‘/dev/hdg’ (Primeiro disco - quarta controladora IDE). É importante entender como os discos e partições são identificados no sistema, pois será necessário usar os parâmetros corretos para monta-los.[1]

2.7 Montando (acessando) uma partição de disco

Você pode acessar uma partição de disco usando o comando ‘mount’.
‘mount [*dispositivo*] [*ponto de montagem*] [*opções*]’
Onde:

dispositivo Identificação da unidade de disco/partição que deseja acessar (como ‘/dev/hda1’ (disco rígido) ou ‘/dev/fd0’ (primeira unidade de disquetes).

ponto de montagem Diretório de onde a _unidade de disco/partição_ será acessado. O diretório deve estar vazio para montagem de um sistema de arquivo. Normalmente é usado o diretório ‘/mnt’ para armazenamento de pontos de montagem temporários.

-t [tipo] Tipo do sistema de arquivos usado pelo _dispositivo_. São aceitos os sistemas de arquivos:

- *ext2* - Para partições ‘GNU/Linux’.
- *vfat* - Para partições ‘Windows 95’ que utilizam nomes extensos de arquivos e diretórios.
- *msdos* - Para partições ‘DOS’ normais.
- *iso9660* - Para montar unidades de ‘CD-ROM’. É o padrão.
- *umsdos* - Para montar uma partição ‘DOS’ com recursos de partições ‘EXT2’, como permissões de acesso, links, etc.

Para mais detalhes sobre opções usadas com cada sistema de arquivos, veja a página de manual *mount*.

-r Caso for especificada, monta a partição somente para leitura.

-w Caso for especificada, monta a partição como leitura/gravação. É o padrão.

Existem muitas outras opções que podem ser usadas com o comando ‘mount’, mas aqui procurei somente mostrar o básico para "montar" seus discos e partições no ‘GNU/Linux’ (para mais opções, veja a página de manual do ‘mount’). Caso você digitar ‘mount’ sem parâmetros, serão mostrados os sistemas de arquivos atualmente montados no sistema. Esta mesma listagem pode ser vista em ‘/etc/mtab’. A remontagem de partição também é muito útil, especialmente após reparos nos sistema de arquivos do disco rígido. Veja alguns exemplos de remontagem abaixo.

É necessário permissões de root para montar partições, a não ser que tenha especificado a opção ‘user’ no arquivo ‘/etc/fstab’ (veja Seção 4.5.1, ‘fstab’).

Exemplo de Montagem:

- Montar uma partição Windows (vfat) de ‘/dev/hda1’ em ‘/mnt’ somente para leitura: ‘mount /dev/hda1 /mnt -r -t vfat’
- Montar a primeira unidade de disquetes ‘/dev/fd0’ em ‘/floppy’: ‘mount /dev/fd0 /floppy -t vfat’
- Montar uma partição DOS localizada em um segundo disco rígido ‘/dev/hdb1’ em ‘/mnt’: ‘mount /dev/hdb1 /mnt -t msdos’.
- Remontar a partição raíz como somente leitura: ‘mount -o remount,rw /’
- Remontar a partição raíz como _leitura/gravação_ (a opção -n é usada porque o ‘mount’ não conseguirá atualizar o arquivo ‘/etc/mtab’ devido ao sistema de arquivos ‘/’ estar montado como somente leitura atualmente: ‘mount -n -o remount,rw /’.

[1] maiores informações RTFS

2.7.1 fstab

O arquivo '/etc/fstab' permite que as partições do sistema sejam montadas facilmente especificando somente o dispositivo ou o ponto de montagem. Este arquivo contém parâmetros sobre as partições que são lidos pelo comando 'mount'. Cada linha deste arquivo contém a partição que desejamos montar, o ponto de montagem, o sistema de arquivos usado pela partição e outras opções. 'fstab' tem a seguinte forma:

Sistema de arquivos	Ponto de Montagem	Tipo	Opções	Dump ordem
/dev/hda1	/	ext2	defaults	0 1
/dev/hda2	/boot	ext2	defaults	0 2
/dev/hda3	/dos	msdos	defaults,noauto,rw	0 0
/dev/hdg	/cdrom	iso9660	defaults,noauto	0 0

Onde:

Sistema de Arquivos Partição que deseja montar.

Ponto de montagem Diretório do 'GNU/Linux' onde a partição montada será acessada.

Tipo Tipo de sistema de arquivos usado na partição que será montada. Para partições 'GNU/Linux' use _ext2_, para partições 'DOS' (sem nomes extensos de arquivos) use _msdos_, para partições 'Win 95' (com suporte a nomes extensos de arquivos) use _vfat_, para unidades de CD-ROM use *iso9660*.

Opções
Especifica as opções usadas com o sistema de arquivos:

- 'defaults' - Utiliza valores padrões de montagem.
- 'noauto' - Não monta os sistemas de arquivos durante a inicialização (útil para CD-ROMS e disquetes).
- 'ro' - Monta como somente leitura.
- 'user' - Permite que usuários montem o sistema de arquivos (não recomendado por motivos de segurança).
- 'sync' é recomendado para uso com discos removíveis (disquetes, zip drives, etc) para que os dados sejam gravados imediatamente na unidade (caso não seja usada, você deve usar o comando Seção 8.22, 'sync' antes de retirar o disquete da unidade.

Ordem Define a ordem que os sistemas de arquivos serão verificados na inicialização do sistema. Se usar 0, o sistema de arquivos não é verificado. O sistema de arquivos raíz que deverá ser verificado primeiro é o raíz "/" . Após configurar o '/etc/fstab', basta digitar o comando 'mount /dev/hdg' ou 'mount /cdrom' para que a unidade de CD-ROM seja montada. Você deve ter notado que não é necessário especificar o sistema de arquivos da partição pois o 'mount' verificará se ele já existe no '/etc/fstab' e caso existir, usará as opções especificadas neste arquivo. Para maiores detalhes veja as páginas de manual 'fstab' e 'mount'.

2.8 Desmontando uma partição de disco

Para desmontar um sistema de arquivos montado com o comando 'mount', use o comando 'umount'. Você deve ter permissões de root para desmontar uma partição.

'umount [*dispositivo*/*ponto de montagem*]'

Você pode tanto usar 'umount /dev/hda1' como 'umount /mnt' para desmontar um sistema de arquivos '/dev/hda1' montado em '/mnt'.

Observação: O comando 'umount' executa o 'sync' automaticamente no momento da desmontagem para garantir que todos os dados ainda não gravados serão salvos.

2.9 Execução de programas

Este capítulo explica como executar programas no 'GNU/Linux' e o uso das ferramentas de controle de execução dos programas.

2.9.1 Executando um comando/programa

Para executar um comando, é necessário que ele tenha permissões de execução (veja Seção 11.2, 'Tipos de Permissões de acesso' e Seção 6.1, 'ls') e que esteja no caminho de procura de arquivos (veja Seção 5.2, 'path').

No aviso de comando # (root) ou *$* (usuário), digite o nome do comando e tecle Enter. O programa/comando é executado e receberá um número de identificação (chamado de PID - Process Identification), este número é útil para identificar o processo no sistema e assim ter um controle sobre sua execução (será visto mais adiante neste capítulo).

Todo o programa executado no 'GNU/Linux' roda sob o controle das permissões de acesso. Recomendo ver mais tarde o Capítulo 11, 'Permissões de acesso a arquivos e diretórios'.

Exemplos de comandos: 'ls', 'df', 'pwd'.

2.10 path

Path é o caminho de procura dos arquivos/comandos executáveis. O path (caminho) é armazenado na variável de ambiente 'PATH'. Você pode ver o conteúdo desta variável com o comando 'echo $PATH'.

Por exemplo, o caminho '/usr/local/bin:/usr/bin:/bin:/usr/bin/X11' significa que se você digitar o comando 'ls', o interpretador de comandos iniciará a procura do programa 'ls' no diretório '/usr/local/bin', caso não encontre o arquivo no diretório '/usr/local/bin' ele inicia a procura em '/usr/bin', até que encontre o arquivo procurado.

Caso o interpretador de comandos chegue até o último diretório do path e não encontre o arquivo/comando digitado, é mostrada a seguinte mensagem:

'bash: ls: command not found' (comando não encontrado).

O caminho de diretórios vem configurado na instalação do Linux, mas pode ser alterado no arquivo '/etc/profile'. Caso deseje alterar o caminho para todos os usuários, este arquivo é o melhor lugar, pois ele é lido por todos os usuários no momento do login.

Caso um arquivo/comando não esteja localizado em nenhum dos diretórios do path, você deve executa-lo usando um './' na frente do comando.

Se deseja alterar o 'path' para um único usuário, modifique o arquivo '.bash_profile' em seu diretório de usuário (home).

OBSERVAÇÃO: Por motivos de segurança, não inclua o diretório atual '$PWD' no 'path'.

2.11 Tipos de Execução de comandos/programas

Um programa pode ser executado de duas formas:

1. 'Primeiro Plano' - Também chamado de *foreground*. Quando você deve esperar o término da execução de um programa para executar um novo comando. Somente é mostrado o aviso de comando após o término de execução do comando/programa.

2. 'Segundo Plano' - Também chamado de *background*. Quando você não precisa esperar o término da execução de um programa para executar um novo comando. Após iniciar um programa em *background* , é mostrado um número PID (identificação do Processo) e o aviso de comando é novamente mostrado, permitindo o uso normal do sistema.

O programa executado em background continua sendo executado internamente. Após ser concluído, o sistema retorna uma mensagem de pronto acompanhado do número PID do processo que terminou.

Para iniciar um programa em 'primeiro plano', basta digitar seu nome normalmente. Para iniciar um programa em 'segundo plano', acrescente o caracter '"&"' após o final do comando.

OBS: Mesmo que um usuário execute um programa em segundo plano e saia do sistema, o programa continuará sendo executado até que seja concluído ou finalizado pelo usuário que iniciou a execução (ou pelo usuário root).

Exemplo: 'find / -name boot.b &'

O comando será executado em segundo plano e deixará o sistema livre para outras tarefas. Após o comando 'find' terminar, será mostrada uma mensagem.

2.12 Executando programas em seqüência

Os comandos podem ser executados em seqüência (um após o término do outro) se os separarmos com ";". Por exemplo: 'echo primeiro;echo segundo;echo terceiro'

2.13 ps

Algumas vezes é útil ver quais processos estão sendo executados no computador. O comando 'ps' faz isto, e também nos mostra qual usuário executou o programa, hora que o processo foi iniciado, etc.

'ps [*opções*]'

Onde:

opções

a Mostra os processos criados por você e de outros usuários do sistema.

x Mostra processos que não são controlados pelo terminal.

u Mostra o nome de usuário que iniciou o processo e hora em que o processo foi iniciado.

m Mostra a memória ocupada por cada processo em execução.

f Mostra a árvore de execução de comandos (comandos que são chamados por outros comandos).

e Mostra variáveis de ambiente no momento da inicialização do processo.

w Mostra a continuação da linha atual na próxima linha ao invés de cortar o restante que não couber na tela.

As opções acima podem ser combinadas para resultar em uma listagem mais completa. Você também pode usar pipes "|" para 'filtrar' a saída do comando 'ps'. Para detalhes, veja Seção 12.4, '| (pipe)'.

Ao contrário de outros comandos, o comando 'ps' não precisa do hífen "-" para especificar os comandos. Isto porque ele não utiliza opções longas e não usa parâmetros.

Exemplos: 'ps', 'ps ax|grep inetd', 'ps auxf', 'ps auxw'.

2.14 top

Mostra os programas em execução ativos, parados, tempo usado na CPU, detalhes sobre o uso da memória RAM, Swap, disponibilidade para execução de programas no sistema, etc.

'top' é um programa que continua em execução mostrando continuamente os processos que estão rodando em seu computador e os recursos utilizados por eles. Para sair do 'top', pressione a tecla 'q'.

'top [*opções*]'

Onde:

-d [tempo] Atualiza a tela após o [tempo] (em segundos).

-s Diz ao 'top' para ser executado em modo seguro.

-i Inicia o 'top' ignorando o tempo de processos zumbis.

-c Mostra a linha de comando ao invés do nome do programa.

A ajuda sobre o 'top' pode ser obtida dentro do programa pressionando a tecla 'h' ou pela página de manual ('man top').

Abaixo algumas teclas úteis:

- 'espaço' - Atualiza imediatamente a tela.
- 'CTRL'+'L' - Apaga e atualiza a tela.
- 'h' - Mostra a tela de ajuda do programa. É mostrado todas as teclas que podem ser usadas com o 'top'.
- 'i' - Ignora o tempo ocioso de processos zumbis.
- 'q' - Sai do programa.
- 'k' - Finaliza um processo - semelhante ao comando 'kill'. Você será perguntado pelo número de identificação do processo (PID). Este comando não estará disponível caso esteja usando o 'top' com a opção '-s'.
- 'n' - Muda o número de linhas mostradas na tela. Se 0 for especificado, será usada toda a tela para listagem de processos.

2.15 Controle de execução de processos

Abaixo algumas comandos e métodos úteis para o controle da execução de processos no 'GNU/Linux'.

2.15.1 Interrompendo a execução de um processo

Para cancelar a execução de algum processo 'rodando em primeiro plano', basta pressionar as teclas 'CTRL'+'C'. A execução do programa será cancelada e será mostrado o aviso de comando. Você também pode usar o comando, 'kill' para interromper um processo sendo executado.

2.15.2 Parando momentaneamente a execução de um processo

Para parar a execução de um processo rodando em primeiro plano, basta pressionar as teclas ‘CTRL’+‘Z’. O programa em execução será pausado e será mostrado o número de seu job e o aviso de comando.

Para retornar a execução de um comando pausado, use ‘fg’ ou , ‘bg’.

O programa permanece na memória no ponto de processamento em que parou quando ele é interrompido. Você pode usar outros comandos ou rodar outros programas enquanto o programa atual está interrompido.

2.15.3 jobs

O comando ‘jobs’ mostra os processos que estão parados ou rodando em *segundo plano*. Processos em segundo plano são iniciados usando o símbolo ‘"&"’ no final da linha de comando ou através do comando ‘bg’.

‘jobs’

O número de identificação de cada processo parado ou em segundo plano (job), é usado com os comandos, ‘fg’ e , ‘bg’.

2.15.4 fg

Permite fazer um programa rodando em segundo plano ou parado, rodar em primeiro plano. Você deve usar o comando ‘jobs’ para pegar o número do processo rodando em segundo plano ou interrompida, este número será passado ao comando ‘fg’ para ativa-lo em primeiro plano.

‘fg [*número*]’

Onde *número* é o número obtido através do comando ‘jobs’.

Caso seja usado sem parâmetros, o ‘fg’ utilizará o último programa interrompido (o maior número obtido com o comando ‘jobs’).

Exemplo: ‘fg 1’.

2.15.5 bg

Permite fazer um programa rodando em primeiro plano ou parado, rodar em segundo plano. Para fazer um programa em primeiro plano rodar em segundo, é necessário primeiro interromper a execução do comando com ‘CTRL’+ ‘Z’, será mostrado o número da tarefa interrompida, use este número com o comando ‘bg’ para iniciar a execução do comando em segundo plano.

‘bg [*número*]’

Onde: *número* número do programa obtido com o pressionamento das teclas ‘CTRL’+‘Z’ ou através do comando ‘jobs’.

2.15.6 kill

Permite enviar um sinal a um comando/programa. Caso seja usado sem parâmetros, o ‘kill’ enviará um sinal de término ao processo sendo executado.

‘kill [*opções*] [*sinal*] [*número*]’

Onde:

número É o número de identificação do processo obtido com o comando, ‘ps’

sinal Sinal que será enviado ao processo. Se omitido usa ‘-15’ como padrão.

opções

-9 Envia um sinal de destruição ao processo ou programa. Ele é terminado imediatamente sem chances de salvar os dados ou apagar os arquivos temporários criados por ele.

Você precisa ser o dono do processo ou o usuário root para termina-lo ou destruí-lo. Você pode verificar se o processo foi finalizado através do comando ‘ps’.

Exemplo: ‘kill 500’, ‘kill -9 500’.

2.15.7 killall

Permite finalizar processos através do nome.

‘killall [*opções*] [*sinal*] [*processo*]’

Onde:

processo Nome do processo que deseja finalizar

sinal Sinal que será enviado ao processo (pode ser obtido usando a opção '-i').

opções

-i Pede confirmação sobre a finalização do processo.

-l Lista o nome de todos os sinais conhecidos.

-q Ignora a existência do processo.

-v Retorna se o sinal foi enviado com sucesso ao processo.

-w Finaliza a execução do 'killall' somente após finalizar todos os processos.

Exemplo: 'killall -HUP inetd'

2.15.8 killall5

Envia um sinal de finalização para todos os processos sendo executados.

'killall5 [*sinal*]'

2.15.9 Sinais do Sistema

Retirado da página de manual 'signal'. O 'GNU/Linux' suporta os sinais listados abaixo. Alguns números de sinais são dependentes de arquitetura.

Primeiro, os sinais descritos no *POSIX 1*:

Sinal	Valor	Ação	Comentário
HUP	1	A	Travamento detectado no terminal de controle ou finalização do processo controlado
INT	2	A	Interrupção através do teclado
QUIT	3	C	Sair através do teclado
ILL	4	C	Instrução Ilegal
ABRT	6	C	Sinal de abortar enviado pela função abort
FPE	8	C	Exceção de ponto Flutuante
KILL	9	AEF	Sinal de destruição do processo
SEGV	11	C	Referência Inválida de memória
PIPE	13	A	Pipe Quebrado: escreveu para o pipe sem leitores
ALRM	14	A	Sinal do Temporizador da chamada do sistema alarm
TERM	15	A	Sinal de Término
USR1	30,10,16	A	Sinal definido pelo usuário 1
USR2	31,12,17	A	Sinal definido pelo usuário 2
CHLD	20,17,18	B	Processo filho parado ou terminado
CONT	19,18,25		Continuar a execução, se interrompido
STOP	17,19,23	DEF	Interromper processo
TSTP	18,20,24	D	Interromper digitação no terminal
TTIN	21,21,26	D	Entrada do terminal para o processo em segundo plano
TTOU	22,22,27	D	Saída do terminal para o processo em segundo plano

As letras da coluna 'Ação' tem o seguinte significado:

* 'A' - A ação padrão é terminar o processo.
* 'B' - A ação padrão é ignorar o sinal.
* 'C' - A ação padrão é terminar o processo e mostrar o core.
* 'D' - A ação padrão é parar o processo.
* 'E' - O sinal não pode ser pego.
* 'F' - O sinal não pode ser ignorado.

Sinais não descritos no *POSIX 1* mas descritos na *SUSv2*:

Sinal	Valor	Acão	Comentário
BUS	10,7,10	C	Erro no Barramento (acesso incorreto da memória)
POLL		A	Evento executado em Pool (Sys V). Sinônimo de IO
PROF	27,27,29	A	Tempo expirado do Profiling
SYS	12,-,12	C	Argumento inválido para a rotina (SVID)
TRAP	5	C	Captura do traço/ponto de interrupção
URG	16,23,21	B	Condição Urgente no soquete (4.2 BSD)
VTALRM	26,26,28	A	Alarme virtual do relógio (4.2 BSD)
XCPU	24,24,30	C	Tempo limite da CPU excedido (4.2 BSD)
XFSZ	25,25,31	C	Limite do tamanho de arquivo excedido (4.2 BSD)

(Para os casos SIGSYS, SIGXCPU, SIGXFSZ, e em algumas arquiteturas também o SIGGUS, a ação padrão do Linux para kernels 2.3.27 e superiores é A (terminar), enquanto SYSv2 descreve C (terminar e mostrar dump core).) Seguem vários outros sinais:

Sinal	Valor	Acão	Comentário
IOT	6	C	Traço IOT. Um sinônimo para ABRT
EMT	7,-,7		
STKFLT	-,16,-	A	Falha na pilha do processador
IO	23,29,22	A	I/O agora possível (4.2 BSD)
CLD	-,-,18		Um sinônimo para CHLD
PWR	29,30,19	A	Falha de força (System V)
INFO	29,-,-		Um sinônimo para SIGPWR
LOST	-,-,-	A	Perda do bloqueio do arquivo
WINCH	28,28,20	B	Sinal de redimensionamento da Janela (4.3 BSD, Sun)
UNUSED	-,31,-	A	Sinal não usado (será SYS)

O "-" significa que o sinal não está presente. Onde três valores são listados, o primeiro é normalmente válido para o Alpha e Sparc, o do meio para i386, PowerPc e sh, o último para o Mips. O sinal 29 é SIGINFO/SIGPWR em um Alpha mas SIGLOST em um Sparc.

2.16 Fechando um programa quando não se sabe como sair

Muitas vezes quando se esta iniciando no ‘GNU/Linux’ você pode executar um programa e talvez não saber como fecha-lo. Este capítulo do guia pretende ajuda-lo a resolver este tipo de problema.

Isto pode também ocorrer com programadores que estão construindo seus programas e por algum motivo não implementam uma opção de saída, ou ela não funciona!

Em nosso exemplo vou supor que executamos um programa em desenvolvimento com o nome ‘contagem’ que conta o tempo em segundos a partir do momento que é executado, mas que o programador esqueceu de colocar uma opção de saída. Siga estas dicas para finaliza-lo:

1. Normalmente todos os programas ‘UNIX’ (o ‘GNU/Linux’ também é um Sistema Operacional baseado no ‘UNIX’) podem ser interrompidos com o pressionamento das teclas ‘<CTRL>’ e ‘<C>’. Tente isto primeiro para finalizar um programa. Isto provavelmente não vai funcionar se estiver usando um Editor de Texto (ele vai entender como um comando de menu). Isto normalmente funciona para comandos que são executados e terminados sem a intervenção do usuário. Caso isto não der certo, vamos partir para a força! ;-)

2. Mude para um novo console (pressionando ‘<ALT>’ e ‘<F2>’), e faça o *login como usuário root*.

3. Localize o PID (número de identificação do processo) usando o comando: ‘ps ax’, aparecerão várias linhas cada uma com o número do processo na primeira coluna, e a linha de comando do programa na última coluna. Caso aparecerem vários processos você pode usar ‘ps ax|grep contagem’, neste caso o ‘grep’ fará uma filtragem da saída do comando ‘ps ax’ mostrando somente as linhas que tem a palavra "contagem".

4. Feche o processo usando o comando ‘kill *PID*’, lembre-se de substituir PID pelo número encontrado pelo comando ‘ps ax’ acima. O comando acima envia um sinal de término de execução para o processo (neste caso o programa ‘contagem’). O sinal de término mantém a chance do programa salvar seus dados ou apagar os arquivos temporários que criou e então ser finalizado, isto depende do programa.

5. Alterne para o console onde estava executando o programa ‘contagem’ e verifique se ele ainda está em execução. Se ele estiver parado mas o aviso de comando não está disponível, pressione a tecla <ENTER>. Freqüentemente acontece isto com o comando ‘kill’, você finaliza um programa mas o aviso de comando não é mostrado até que se pressione <ENTER>.

6. Caso o programa ainda não foi finalizado, repita o comando ‘kill’ usando a opção -9: ‘kill -9 PID’. Este comando envia um sinal de DESTRUIÇÃO do processo, fazendo ele terminar "na marra"!

Uma última dica: todos os programas estáveis (todos que acompanham as boas distribuições 'GNU/Linux') tem sua opção de saída. Lembre-se que quando finaliza um processo todos os dados do programa em execução podem ser perdidos (principalmente se estiver em um editor de textos), mesmo usando o 'kill' sem o parâmetro '-9'.

Procure a opção de saída de um programa consultando o help on line, as páginas de manual, a documentação que acompanha o programa, info pages.

2.17 Eliminando caracteres estranhos

As vezes quando um programa 'mal comportado' é finalizado ou quando você visualiza um arquivo binário através do comando 'cat', é possível que o aviso de comando (prompt) volte com caracteres estranhos.

Para fazer tudo voltar ao normal, basta digitar 'reset' e teclar 'ENTER'. Não se preocupe, o comando 'reset' não reiniciará seu computador (como o botão reset do seu computador faz), ele apenas fará tudo voltar ao normal.

Note que enquanto você digitar 'reset' aparecerão caracteres estranhos ao invés das letras. Não se preocupe! Basta digitar corretamente e bater 'ENTER' e o aviso de comando voltará ao normal.

3 Comandos para manipulação de diretório

Abaixo comandos úteis para a manipulação de diretórios.

3.1 ls

Lista os arquivos de um diretório.

'ls [*opções*] [*caminho/arquivo*] [*caminho1/arquivo1*] ...'

onde:

caminho/arquivo Diretório/arquivo que será listado.

caminho1/arquivo1 Outro Diretório/arquivo que será listado. Podem ser feitas várias listagens de uma só vez.

opções

-a, –all Lista todos os arquivos (inclusive os ocultos) de um diretório.

-A, –almost-all Lista todos os arquivos (inclusive os ocultos) de um diretório, exceto o diretório atual e o de nível anterior.

-B, –ignore-backups Não lista arquivos que terminam com ~ (Backup).

–color=PARAM

Mostra os arquivos em cores diferentes, conforme o tipo de arquivo. PARAM pode ser:

- *never* - Nunca lista em cores (mesma coisa de não usar o parâmetro –color).
- *always* - Sempre lista em cores conforme o tipo de arquivo.
- *auto* - Somente colore a listagem se estiver em um terminal.

-d, –directory Lista os nomes dos diretórios ao invés do conteúdo.

-f Não classifica a listagem.

-F Insere um caracter após arquivos executáveis ('*'), diretórios ('/'), soquete ('='), link simbólico ('@') e pipe ('|'). Seu uso é útil para identificar de forma fácil tipos de arquivos nas listagens de diretórios.

-G, –no-group Oculta a coluna de grupo do arquivo.

-h, –human-readable Mostra o tamanho dos arquivos em Kbytes, Mbytes, Gbytes.

-H Faz o mesmo que '-h', mas usa unidades de 1000 ao invés de 1024 para especificar Kbytes, Mbytes, Gbytes.

-l Usa o formato longo para listagem de arquivos. Lista as permissões, data de modificação, donos, grupos, etc.

-n Usa a identificação de usuário e grupo numérica ao invés dos nomes.

-L, –dereference Lista o arquivo original e não o link referente ao arquivo.

-o Usa a listagem longa sem os donos dos arquivos (mesma coisa que -lG).

-p Mesma coisa que -F, mas não inclui o símbolo '*' em arquivos executáveis. Esta opção é típica de sistemas 'Linux'.

-R Lista diretórios e sub-diretórios recursivamente.

Uma listagem feita com o comando 'ls -la' normalmente é mostrada da seguinte maneira:
-rwxr-xr– 1 gleydson user 8192 nov 4 16:00 teste
Abaixo as explicações de cada parte:
'-rwxr-xr–'
São as permissões de acesso ao arquivo teste. A primeira letra (da esquerda) identifica o tipo do arquivo, se tiver um 'd' é um diretório, se tiver um "-" é um arquivo normal.

'1' Se for um diretório, mostra a quantidade de sub-diretórios existentes dentro dele. Caso for um arquivo, será 1.

'gleydson' Nome do dono do arquivo 'teste'.

'user' Nome do grupo que o arquivo 'teste' pertence.

'8192' Tamanho do arquivo (em bytes).

'nov' Mês da criação/ última modificação do arquivo.

'4' Dia que o arquivo foi criado.

'16:00' Hora em que o arquivo foi criado/modificado. Se o arquivo foi criado ha mais de um ano, em seu lugar é mostrado o ano da criação do arquivo.

'teste' Nome do arquivo.

Exemplos do uso do comando 'ls':

- 'ls' - Lista os arquivos do diretório atual.
- 'ls /bin /sbin' - Lista os arquivos do diretório /bin e /sbin
- 'ls -la /bin' - Listagem completa (vertical) dos arquivos do diretório /bin inclusive os ocultos.

3.2 cd

Entra em um diretório. Você precisa ter a permissão de execução para entrar no diretório.

'cd [*diretório*]'

onde:
diretório - diretório que deseja entrar.
Exemplos:

- Usando 'cd' sem parâmetros ou 'cd ~', você retornará ao seu diretório de usuário (diretório home).
- 'cd /', retornará ao diretório raíz.
- 'cd -', retornará ao diretório anteriormente acessado.
- 'cd ..', sobe um diretório.
- 'cd ../[*diretório*]', sobe um diretório e entra imediatamente no próximo (por exemplo, quando você está em '/usr/sbin', você digita 'cd ../bin', o comando 'cd' retorna um diretório ('/usr') e entra imediatamente no diretório 'bin' ('/usr/bin').

3.3 pwd

Mostra o nome e caminho do diretório atual. Você pode usar o comando pwd para verificar em qual diretório se encontra (caso seu aviso de comandos não mostre isso).

3.4 mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos de um determinado tipo. O diretório pode ser entendido como uma *pasta* onde você guarda seus papeis (arquivos). Como uma pessoa organizada, você utilizará uma pasta para guardar cada tipo de documento, da mesma forma você pode criar um diretório 'vendas' para guardar seus arquivos relacionados com vendas naquele local.

'mkdir [*opções*] [*caminho/diretório*] [*caminho1/diretório1*]'

onde:

caminho Caminho onde o diretório será criado.

diretório Nome do diretório que será criado.

opções:

–verbose Mostra uma mensagem para cada diretório criado. As mensagens de erro serão mostradas mesmo que esta opção não seja usada.

Para criar um novo diretório, você deve ter permissão de gravação. Por exemplo, para criar um diretório em /tmp com o nome de 'teste' que será usado para gravar arquivos de teste, você deve usar o comando '"mkdir /tmp/teste"'.

Podem ser criados mais de um diretório com um único comando ('mkdir /tmp/teste /tmp/teste1 /tmp/teste2').

3.5 rmdir

Remove um diretório do sistema. Este comando faz exatamente o contrário do 'mkdir'. O diretório a ser removido deve estar vazio e você deve ter permissão de gravação para remove-lo.

'rmdir [*caminho/diretório*] [*caminho1/diretório1*]'

onde:

caminho Caminho do diretório que será removido.

diretório Nome do diretório que será removido.

É necessário que esteja um nível acima do diretório(s) que será(ão) removido(s). Para remover diretórios que contenham arquivos, use o comando 'rm' com a opção '-r' .

Por exemplo, para remover o diretório '/tmp/teste' você deve estar no diretório 'tmp' e executar o comando 'rmdir teste'.

4 Comandos para manipulação de Arquivos

Abaixo, comandos utilizados para manipulação de arquivos.

4.1 cat

Mostra o conteúdo de um arquivo binário ou texto.

'cat [*opções*] [*diretório/arquivo*] [*diretório1/arquivo1*]'

diretório/arquivo Localização do arquivo que deseja visualizar o conteúdo.

opções

-n, –number Mostra o número das linhas enquanto o conteúdo do arquivo é mostrado.

-s, –squeeze-blank Não mostra mais que uma linha em branco entre um parágrafo e outro.

- Lê a entrada padrão.

O comando 'cat' trabalha com arquivos texto. Use o comando 'zcat' para ver diretamente arquivos compactados com 'gzip'.

Exemplo: 'cat /usr/doc/copyright/GPL'

4.2 rm

Apaga arquivos. Também pode ser usado para apagar diretórios e sub-diretórios vazios ou que contenham arquivos.

‘rm [*opções*][*caminho*][*arquivo/diretório*] [*caminho1*][*arquivo1/diretório1*]’

onde:

caminho Localização do arquivo que deseja apagar. Se omitido, assume que o arquivo esteja no diretório atual.

arquivo/diretório Arquivo que será apagado.

opções

-i, –interactive Pergunta antes de remover, esta é ativada por padrão.

-v, –verbose Mostra os arquivos na medida que são removidos.

-r, –recursive Usado para remover arquivos em sub-diretórios. Esta opção também pode ser usada para remover sub-diretórios.

-f, –force Remove os arquivos sem perguntar.

Use com atenção o comando ‘rm’, uma vez que os arquivos e diretórios forem apagados, eles não poderão ser mais recuperados.
Exemplos:

- ‘rm teste.txt’ - Apaga o arquivo ‘teste.txt’ no diretório atual.
- ‘rm *.txt’ - Apaga todos os arquivos do diretório atual que terminam com ‘.txt’.
- ‘rm *.txt teste.novo’ - Apaga todos os arquivos do diretório atual que terminam com ‘.txt’ e também o arquivo ‘teste.novo’.
- ‘rm -rf /tmp/teste/*’ - Apaga todos os arquivos e sub-diretórios do diretório ‘/tmp/teste’ mas mantém o sub-diretório ‘/tmp/teste’.
- ‘rm -rf /tmp/teste’ - Apaga todos os arquivos e sub-diretórios do diretório ‘/tmp/teste’, inclusive ‘/tmp/teste’.

4.3 cp

Copia arquivos.

‘cp [*opções*] [*origem*] [*destino*]’

onde:

origem Arquivo que será copiado. Podem ser especificados mais de um arquivo para ser copiado usando "Coringas" .

destino O caminho ou nome de arquivo onde será copiado. Se o destino for um diretório, os arquivos de origem serão copiados para dentro do diretório.

opções

i, –interactive Pergunta antes de substituir um arquivo existente.

-f, –force Não pergunta, substitui todos os arquivos caso já exista.

-r Copia arquivos dos diretórios e subdiretórios da origem para o destino. É recomendável usar -R ao invés de -r.

-R, –recursive Copia arquivos e sub-diretórios (como a opção -r) e também os arquivos especiais FIFO e dispositivos.

-v, –verbose Mostra os arquivos enquanto estão sendo copiados.

O comando ‘cp’ copia arquivos da ORIGEM para o DESTINO. Ambos origem e destino terão o mesmo conteúdo após a cópia.
Exemplos:

‘cp teste.txt teste1.txt’

Copia o arquivo ‘teste.txt’ para ‘teste1.txt’.

‘cp teste.txt /tmp’

Copia o arquivo ‘teste.txt’ para dentro do diretório ‘/tmp’.

‘cp * /tmp’

Copia todos os arquivos do diretório atual para ‘/tmp’.

‘cp /bin/* .’

Copia todos os arquivos do diretório ‘/bin’ para o diretório em que nos encontramos no momento.

‘cp -R /bin /tmp’

Copia o diretório ‘/bin’ e todos os arquivos/sub-diretórios existentes para o diretório ‘/tmp’.

‘cp -R /bin/* /tmp’

Copia todos os arquivos do diretório ‘/bin’ (exceto o diretório ‘/bin’) e todos os arquivos/sub-diretórios existentes dentro dele para ‘/tmp’.

‘cp -R /bin /tmp’

Copia todos os arquivos e o diretório ‘/bin’ para ‘/tmp’.

4.4 mv

Move ou renomeia arquivos e diretórios. O processo é semelhante ao do comando ‘cp’ mas o arquivo de origem é apagado após o término da cópia.

‘mv [*opções*] [*origem*] [*destino*]’

Onde:

origem Arquivo/diretório de origem.

destino Local onde será movido ou novo nome do arquivo/diretório.

opções

-f, –force Substitui o arquivo de destino sem perguntar.

-i, –interactive Pergunta antes de substituir. É o padrão.

-v, –verbose Mostra os arquivos que estão sendo movidos.

O comando ‘mv’ copia um arquivo da origem para o destino (semelhante ao ‘cp’), mas após a cópia, o arquivo de origem é apagado.
Exemplos:

‘mv teste.txt teste1.txt’

Muda o nome do arquivo ‘teste.txt’ para ‘teste1.txt’.

‘mv teste.txt /tmp’

Move o arquivo teste.txt para ‘/tmp’. Lembre-se que o arquivo de origem é apagado após ser movido.

‘mv teste.txt teste.new’ (supondo que ‘teste.new’ já exista)

Copia o arquivo ‘teste.txt’ por cima de ‘teste.new’ e apaga ‘teste.txt’ após terminar a cópia.

5 Comandos Diversos

Comandos de uso diversos no sistema.

5.1 clear

Limpa a tela e posiciona o cursor no canto superior esquerdo do vídeo.

‘clear’

5.2 date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora.

‘date MesDiaHoraMinuto[AnoSegundos]’

Onde:

MesDiaHoraMinuto [AnoSegundos] São respectivamente os números do mês, dia, hora e minutos sem espaços. Opcionalmente você pode especificar o Ano (com 2 ou 4 dígitos) e os Segundos.

+[FORMATO]
Define o formato da listagem que será usada pelo comando ‘date’.
Os seguintes formatos são os mais usados:

- ‘%d’ - Dia do Mês (00-31).
- ‘%d’ - Mês do Ano (00-12).
- ‘%y’ - Ano (dois dígitos).
- ‘%Y’ - Ano (quatro dígitos).
- ‘%H’ - Hora (00-24).
- ‘%I’ - Hora (00-12).
- ‘%M’ - Minuto (00-59).
- ‘%j’ - Dia do ano (1-366).
- ‘%p’ - AM/PM (útil se utilizado com %d).
- ‘%r’ - Formato de 12 horas completo (hh:mm:ss AM/PM).
- ‘%T’ - Formato de 24 horas completo (hh:mm:ss).
- ‘%w’ - Dia da semana (0-6).

Outros formatos podem ser obtidos através da página de manual do ‘date’.
Para maiores detalhes, veja a página de manual do comando ‘date’.
Para ver a data atual digite: ‘date’
Se quiser mudar a Data para 25/12 e a hora para 08:15 digite: ‘date 12250815’
Para mostrar somente a data no formato dia/mês/ano: ‘date +%d/%m/%Y’

5.3 df

Mostra o espaço livre/ocupado de cada partição.

‘df [*opções*]’

onde:
opções

-a Inclui sistemas de arquivos com 0 blocos.

-h, –human-readable Mostra o espaço livre/ocupado em _MB, KB, GB_ ao invés de blocos.

-H Idêntico a ‘-h’ mas usa 1000 ao invés de 1024 como unidade de cálculo.

-k Lista em Kbytes.

-l Somente lista sistema de arquivos locais.

-m Lista em Mbytes (equivalente a –block-size=1048576).

Exemplos: ‘df’, ‘df -h’, ‘df -t vfat’.

5.4 ln

Cria links para arquivos e diretórios no sistema. O link é um mecanismo que faz referência a outro arquivo ou diretório em outra localização. O link em sistemas 'GNU/Linux' faz referência reais ao arquivo/diretório podendo ser feita cópia do link (será copiado o arquivo alvo), entrar no diretório (caso o link faça referência a um diretório), etc.

'ln [*opções*] [*origem*] [*link*]'

Onde:

origem Diretório ou arquivo de onde será feito o link.

link Nome do link que será criado.

opções

-s Cria um link simbólico. Usado para criar ligações com o arquivo/diretório de destino.

-v Mostra o nome de cada arquivo antes de fazer o link.

-d Cria um hard link para diretórios. Somente o root pode usar esta opção.

Existem 2 tipos de links: *simbólicos e hardlinks*.

- O *link simbólico* cria um arquivo especial no disco (do tipo link) que tem como conteúdo o caminho para chegar até o arquivo alvo (isto pode ser verificado pelo tamanho do arquivo do link). Use a opção '-s' para criar links simbólicos.
- O *hardlink* faz referência ao mesmo inodo do arquivo original, desta forma ele será perfeitamente idêntico, inclusive nas permissões de acesso, ao arquivo original. Ao contrário dos links simbólicos, não é possível fazer um hardlink para um diretório ou fazer referência a arquivos que estejam em partições diferentes.

Observações:

- Se for usado o comando 'rm' com um link, somente o link será removido.
- Se for usado o comando 'cp' com um link, o arquivo original será copiado ao invés do link.
- Se for usado o comando 'mv' com um link, a modificação será feita no link.
- Se for usado um comando de visualização (como o 'cat'), o arquivo original será visualizado.

Exemplos:

- 'ln -s /dev/ttyS1 /dev/modem' - Cria o link '/dev/modem' para o arquivo '/dev/ttyS1'.
- 'ln -s /tmp ~/tmp' - Cria um link '~/tmp' para o diretório '/tmp'.

5.5 du

Mostra o espaço ocupado por arquivos e sub-diretórios do diretório atual.

'du [*opções*]'

onde:

opções

-a, –all Mostra o espaço ocupado por todos os arquivos.

-b, –bytes Mostra o espaço ocupado em bytes.

-c, –total Faz uma totalização de todo espaço listado.

-D Não conta links simbólicos.

-h, –human Mostra o espaço ocupado em formato legível por humanos (Kb, Mb) ao invés de usar blocos.

-H Como o anterior mas usa 1000 e não 1024 como unidade de cálculo.

-k Mostra o espaço ocupado em Kbytes.

-m Mostra o espaço ocupado em Mbytes.

-S, –separate-dirs Não calcula o espaço ocupado por sub-diretórios.

Exemplo: 'du -h', 'du -hc'.

5.6 find

Procura por arquivos/diretórios no disco. ‘find’ pode procurar arquivos através de sua data de modificação, tamanho, etc através do uso de opções. ‘find’, ao contrário de outros programas, usa opções longas através de um ‘"-"’.

‘find [*diretório*] [*opções/expressão*]’

Onde:

diretório Inicia a procura neste diretório, percorrendo seu sub-diretórios.

opções/expressão

-name [expressão] Procura pelo nome [expressão] nos nomes de arquivos e diretórios processados.

-depth Processa os sub-diretórios primeiro antes de processar os arquivos do diretório principal.

-maxdepth [num] Faz a procura até [num] sub-diretórios dentro do diretório que está sendo pesquisado.

-mindepth [num] Não faz nenhuma procura em diretórios menores que [num] níveis.

-mount, -xdev Não faz a pesquisa em sistemas de arquivos diferentes daquele de onde o comando ‘find’ foi executado.

-size [num] Procura por arquivos que tiverem o tamanho [num]. [num] pode ser antecedido de "+" ou "-" para especificar um arquivo maior ou menor que [num]. A opção -size pode ser seguida de:

- ‘b’ - Especifica o tamanho em blocos de 512 bytes. É o padrão caso [num] não seja acompanhado de nenhuma letra.
- ‘c’ - Especifica o tamanho em bytes.
- ‘k’ - Especifica o tamanho em Kbytes.

-type [tipo] Procura por arquivos do [tipo] especificado. Os seguintes tipos são aceitos:

- ‘b’ - bloco
- ‘c’ - caracter
- ‘d’ - diretório
- ‘p’ - pipe
- ‘f’ - arquivo regular
- ‘l’ - link simbólico
- ‘s’ - sockete

A maior parte dos argumentos numéricos podem ser precedidos por "+" ou "-". Para detalhes sobre outras opções e argumentos, consulte a página de manual.

Exemplo:

- ‘find / -name grep’ - Procura no diretório raíz e sub-diretórios um arquivo/diretório chamado ‘grep’.
- ‘find / -name grep -maxdepth 3’ - Procura no diretório raíz e sub-diretórios até o 3o. nível, um arquivo/diretório chamado ‘grep’.
- ‘find . -size +1000k’ - Procura no diretório atual e sub-diretórios um arquivo com tamanho maior que 1000 kbytes (1Mbyte).

5.7 free

Mostra detalhes sobre a utilização da memória RAM do sistema.

‘free [*opções*]’

Onde:
opções

-b Mostra o resultado em bytes.

-k Mostra o resultado em Kbytes.

-m Mostra o resultado em Mbytes.

-o Oculta a linha de buffers.

-t Mostra uma linha contendo o total.

-s [num] Mostra a utilização da memória a cada [num] segundos.

O ‘free’ é uma interface ao arquivo ‘/proc/meminfo’.

5.8 grep

Procura por um texto dentro de um arquivo(s) ou no dispositivo de entrada padrão.

‘grep [*expressão*] [*arquivo*] [*opções*]’

Onde:

expressão palavra ou frase que será procurada no texto. Se tiver mais de 2 palavras você deve identifica-la com aspas "" caso contrário o ‘grep’ assumirá que a segunda palavra é o arquivo!

arquivo Arquivo onde será feita a procura.

opções

-A [número] Mostra o [número] de linhas após a linha encontrada pelo ‘grep’.

-B [número] Mostra o [número] de linhas antes da linha encontrada pelo ‘grep’.

-f [arquivo] Especifica que o texto que será localizado, esta no arquivo [arquivo].

-h, –no-filename Não mostra os nomes dos arquivos durante a procura.

-i, –ignore-case Ignora diferença entre maiúsculas e minúsculas no texto procurado e arquivo.

-n, –line-number Mostra o nome de cada linha encontrada pelo ‘grep’.

-U, –binary Trata o arquivo que será procurado como binário.

Se não for especificado o nome de um arquivo ou se for usado um hífen "-", ‘grep’ procurará a string no dispositivo de entrada padrão. O ‘grep’ faz sua pesquisa em arquivos texto. Use o comando ‘zgrep’ para pesquisar diretamente em arquivos compactados com ‘gzip’, os comandos e opções são as mesmas.
Exemplos: ‘grep "capitulo" texto.txt’, ‘ps ax|grep inetd’, ‘grep "capitulo" texto.txt -A 2 -B 2’.

5.9 head

Mostra as linhas iniciais de um arquivo texto.

‘head [*opções*]’

Onde:

-c [numero] Mostra o [numero] de bytes do inicio do arquivo.

-n [numero] Mostra o [numero] de linhas do inicio do arquivo. Caso não for especificado, o ‘head’ mostra as 10 primeiras linhas.

Exemplos: ‘head teste.txt’, ‘head -n 20 teste.txt’.

5.10 more

Permite fazer a paginação de arquivos ou da entrada padrão. O comando ‘more’ pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o ‘more’ efetua uma pausa e permite que você pressione ‘Enter’ para continuar avançando o número de páginas. Para sair do ‘more’ pressione ‘q’.

‘more [*arquivo*]’

Onde:

arquivo É o arquivo que será paginado.

O ‘more’ somente permite avançar o conteúdo do arquivo linha por linha, para um melhor controle de paginação, use o comando Seção 8.12, ‘less’.
Para visualizar diretamente arquivos texto compactados pelo ‘gzip’ ‘.gz’ use o comando ‘zmore’.
Exemplos: ‘more /etc/passwd’, ‘cat /etc/passwd|more’.

5.11 less

Permite fazer a paginação de arquivos ou da entrada padrão. O comando ‘less’ pode ser usado como comando para leitura de arquivos que ocupem mais de uma tela. Quando toda a tela é ocupada, o ‘less’ efetua uma pausa (semelhante ao ‘more’) e permite que você pressione Seta para Cima e Seta para Baixo ou PgUP/PgDown para fazer o rolamento da página. Para sair do ‘less’ pressione ‘q’.

‘less [*arquivo*]’

Onde:

arquivo É o arquivo que será paginado.

Para visualizar diretamente arquivos texto compactados pelo utilitário ‘gzip’ (arquivos ‘.gz’), use o comando ‘zless’.
Exemplos: ‘less /etc/passwd’, ‘cat /etc/passwd|less’

5.12 sort

Organiza as linhas de um arquivo texto ou da entrada padrão.

‘sort [*opções*] [*arquivo*]’

Onde:

arquivo É o nome do arquivo que será organizado. Caso não for especificado, será usado o dispositivo de entrada padrão (normalmente o teclado ou um "|").

opções

-b Ignora linhas em branco.

-d Somente usa letras, dígitos e espaços durante a organização.

-f Ignora a diferença entre maiúsculas e minúsculas.

-r Inverte o resultado da comparação.

-n Caso estiver organizando um campo que contém números, os números serão organizados na ordem aritmética. Por exemplo, se você tiver um arquivo com os números 100; 10; 50

Usando a opção ‘-n’, o arquivo será organizado desta maneira:
10; 50; 100
Caso esta opção *não* for usada com o ‘sort’, ele organizará como uma listagem alfabética (que começam de ‘a’ até ‘z’ e do ‘0’ até ‘9’)
10; 100; 50

-c Verifica se o arquivo já esta organizado. Caso não estiver, retorna a mensagem "*disorder on arquivo*".

-o *arquivo* Grava a saída do comando ‘sort’ no *arquivo*.

Abaixo, exemplos de uso do comando ‘sort’:

- ‘sort ‘texto.txt” - Organiza o arquivo ‘texto.txt’ em ordem crescente.
- ‘sort ‘texto.txt’ -r’ - Organiza o conteúdo do arquivo ‘texto.txt’ em ordem decrescente.
- ‘cat ‘texto.txt’|sort’ - Faz a mesma coisa que o primeiro exemplo, só que neste caso a saída do comando ‘cat’ é redirecionado a entrada padrão do comando ‘sort’.
- ‘sort -f ‘texto.txt” - Ignora diferenças entre letras maiúsculas e minúsculas durante a organização.

5.13 tail

Mostra as linhas finais de um arquivo texto.

‘tail [*opções*]’

Onde:

-c [numero] Mostra o [numero] de bytes do final do arquivo.

-n [numero] Mostra o [numero] de linhas do final do arquivo.

Exemplos: ‘tail teste.txt’, ‘tail -n 20 teste.txt’.

5.14 time

Mede o tempo gasto para executar um processo (programa).

‘time [*comando*]’

Onde: *comando* é o comando/programa que deseja medir o tempo gasto para ser concluído.
Exemplo: ‘time ls’, ‘time find / -name crontab’.

5.15 touch

Muda a data e hora que um arquivo foi criado. Também pode ser usado para criar arquivos vazios. Caso o ‘touch’ seja usado com arquivos que não existam, por padrão ele criará estes arquivos.

‘touch [*opções*] [*arquivos*]’

Onde:

arquivos Arquivos que terão sua data/hora modificados.

opções

-t MMDDhhmm[ANO.segundos] Usa Minutos (MM), Dias (DD), Horas (hh), minutos (mm) e opcionalmente o ANO e segundos para modificação do(s) arquivos ao invés da data e hora atual.

-a, –time=atime Faz o ‘touch’ mudar somente a data e hora do acesso ao arquivo.

-c, –no-create Não cria arquivos vazios, caso os _arquivos_ não existam.

-m, –time=mtime Faz o ‘touch’ mudar somente a data e hora da modificação.

-r [arquivo] Usa as horas no [arquivo] como referência ao invés da hora atual.

Exemplos:

- ‘touch teste’ - Cria o arquivo ‘teste’ caso ele não existir.
- ‘touch -t 10011230 teste’ - Altera da data e hora do arquivo para 01/10 e 12:30.
- ‘touch -t 120112301999.30 teste’ - Altera da data, hora ano, e segundos do arquivo para 01/12/1999 e 12:30:30.
- ‘touch -t 12011200 *’ - Altera a data e hora do arquivo para 01/12 e 12:00.

5.16 uptime

Mostra o tempo de execução do sistema desde que o computador foi ligado.

‘uptime’

5.17 dmesg

Mostra as mensagens de inicialização do kernel. São mostradas as mensagens da última inicialização do sistema.

‘dmesg|less’

5.18 echo

Mostra mensagens. Este comando é útil na construção de scripts para mostrar mensagens na tela para o usuário acompanhar sua execução.

‘echo [*mensagem*]’

A opção ‘-n’ pode ser usada para que não ocorra o salto de linha após a mensagem ser mostrada.

5.19 su

Permite o usuário mudar sua identidade para outro usuário sem fazer o logout. Útil para executar um programa ou comando como root sem ter que abandonar a seção atual.

‘su [*usuário*]’

Onde: *usuário* é o nome do usuário que deseja usar para acessar o sistema. Se não digitado, é assumido o usuário ‘root’.
Será pedida a senha do superusuário para autenticação. Digite ‘exit’ quando desejar retornar a identificação de usuário anterior.

5.20 uname

Retorna o nome e versão do kernel atual.

‘uname’

5.21 reboot

Reinicia o computador.

5.22 shutdown

Desliga/reinicia o computador imediatamente ou após determinado tempo (programável) de forma segura. Todos os usuários do sistema são avisados que o computador será desligado . Este comando somente pode ser executado pelo usuário root ou usuário autorizado no arquivo ‘/etc/shutdown.allow’.

‘shutdown [*opções*] [*hora*] [*mensagem*]’

hora Momento que o computador será desligado. Você pode usar ‘HH:MM’ para definir a hora e minuto, ‘MM’ para definir minutos, ‘+SS’ para definir após quantos segundos, ou ‘now’ para imediatamente (equivalente a +0).

O ‘shutdown’ criará o arquivo ‘/etc/nologin’ para não permitir que novos usuários façam login no sistema (com excessão do root). Este arquivo é removido caso a execução do ‘shutdown’ seja cancelada (opção -c) ou após o sistema ser reiniciado.

mensagem Mensagem que será mostrada a todos os usuários alertando sobre o reinicio/desligamento do sistema.

opções

-h Inicia o processo para desligamento do computador.

-r Reinicia o sistema

-c Cancela a execução do ‘shutdown’. Você pode acrescentar uma mensagem avisando aos usuários sobre o fato.

O ‘shutdown’ envia uma mensagem a todos os usuários do sistema alertando sobre o desligamento durante os 15 minutos restantes e assim permite que finalizem suas tarefas. Após isto, o ‘shutdown’ muda o nível de execução através do comando ‘init’ para 0 (desligamento), 1 (modo monousuário), 6 (reinicialização). É recomendado utilizar o símbolo "&" no final da linha de comando para que o ‘shutdown’ seja executado em segundo plano.

Quando restarem apenas 5 minutos para o reinicio/desligamento do sistema, o programa ‘login’ será desativado, impedindo a entrada de novos usuários no sistema.

O programa ‘shutdown’ pode ser chamado pelo ‘init’ através do pressionamento da combinação das teclas de reinicialização ‘CTRL+ALT+DEL’ alterando-se o arquivo ‘/etc/inittab’. Isto permite que somente os usuários autorizados (ou o root) possam reinicializar o sistema.

Exemplos:

- ‘"shutdown -h now"’ - Desligar o computador imediatamente.
- ‘"shutdown -r now"’ - Reinicia o computador imediatamente.
- ‘"shutdown 19:00 A manutenção do servidor será iniciada às 19:00"’ - Faz o computador entrar em modo monousuário (init 1) às 19:00 enviando a mensagem _A manutenção do servidor será iniciada às 19:00_ a todos os usuários conectados ao sistema.
- ‘"shutdown -r 15:00 O sistema será reiniciado às 15:00 horas"’ - Faz o computador ser reiniciado (init 6) às 15:00 horas enviando a mensagem _O sistema será reiniciado às 15:00 horas_ a todos os usuários conectados ao sistema.
- ‘shutdown -r 20’ - Faz o sistema ser reiniciado após 20 minutos.
- ‘shutdown -c’ - Cancela a execução do ‘shutdown’.

5.23 wc

Conta o número de palavras, bytes e linhas em um arquivo ou entrada padrão. Se as opções forem omitidas, o ‘wc’ mostra a quantidade de linhas, palavras, e bytes.

‘wc [*opções*] [*arquivo*]’

Onde:

arquivo Arquivo que será verificado pelo comando ‘wc’.

opções

-c, –bytes Mostra os bytes do arquivo.

-w, –words Mostra a quantidade de palavras do arquivo.

-l, –lines Mostra a quantidade de linhas do arquivo.

A ordem da listagem dos parâmetros é única, e modificando a posição das opções não modifica a ordem que os parâmetros são listados.

Exemplo:

- ‘wc /etc/passwd’ - Mostra a quantidade de linhas, palavras e letras (bytes) no arquivo ‘/etc/passwd’.
- ‘wc -w /etc/passwd’ - Mostra a quantidade de palavras.
- ‘wc -l /etc/passwd’ - Mostra a quantidade de linhas.
- ‘wc -l -w /etc/passwd’ - Mostra a quantidade de linhas e palavras no arquivo ‘/etc/passwd’.

6 Comandos de rede

Este capítulo traz alguns comandos úteis para uso em rede e ambientes multiusuário.

6.1 who

Mostra quem está atualmente conectado no computador. Este comando lista os nomes de usuários que estão conectados em seu computador, o terminal e data da conexão.

'who [*opções*]'

onde:

opções

-H, –heading Mostra o cabeçalho das colunas.

-i, -u, –idle Mostra o tempo que o usuário está parado em Horas:Minutos.

-m, i am Mostra o nome do computador e usuário associado ao nome. É equivalente a digitar 'who i am' ou 'who am i'.

-q, –count Mostra o total de usuários conectados aos terminais.

-T, -w, –mesg Mostra se o usuário pode receber mensagens via 'talk' (conversação).

- + O usuário recebe mensagens via talk
- - O usuário não recebe mensagens via talk.
- ? Não foi possível determinar o dispositivo de terminal onde o usuário está conectado.

6.2 Telnet

Permite acesso a um computador remoto. É mostrada uma tela de acesso correspondente ao computador local onde deve ser feita a autenticação do usuário para entrar no sistema. Muito útil, mas deve ser tomado cuidados ao disponibilizar este serviço para evitar riscos de segurança.

'telnet [*opções*] [*ip/dns*] [*porta*]'

onde:

ip/dns Endereço IP do computador de destino ou nome DNS.

porta Porta onde será feita a conexão. Por padrão, a conexão é feita na porta *23*.

opções

-8 Requisita uma operação binária de 8 bits. Isto força a operação em modo binário para envio e recebimento. Por padrão, 'telnet' não usa 8 bits.

-a Tenta um login automático, enviando o nome do usuário lido da variável de ambiente 'USER'.

-d Ativa o modo de debug.

-r Ativa a emulação de rlogin.

-l [usuário] Faz a conexão usando [usuário] como nome de usuário.

Exemplo: 'telnet 192.168.1.1', 'telnet 192.168.1.1 23'.

6.2.1 finger

Mostra detalhes sobre os usuários de um sistema. Algumas versões do 'finger' possuem bugs e podem significar um risco para a segurança do sistema. É recomendado desativar este serviço na máquina local.

'finger [*usuário*] [*usuário@host*]'

Onde:

usuário Nome do usuário que deseja obter detalhes do sistema. Se não for digitado o nome de usuário, o sistema mostra detalhes de todos os usuários conectados no momento.

usuário@host Nome do usuário e endereço do computador que deseja obter detalhes.

-l Mostra os detalhes de todos os usuários conectados no momento. Entre os detalhes, estão incluídos o *nome do interpretador de comandos* (shell) do usuário, *diretório home*, *nome do usuário*, *endereço*, etc.

-p Não exibe o conteúdo dos arquivos '.plan' e '.project'

Se for usado sem parâmetros, mostra os dados de todos os usuários conectados atualmente ao seu sistema.
Exemplo: 'finger', 'finger root'.

6.3 ftp

Permite a transferência de arquivos do computador remoto/local e vice versa. O file transfer protocol é o sistema de transmissão de arquivos mais usado na Internet. É requerida a autenticação do usuário para que seja permitida a conexão. Muitos servidores ftp disponibilizam acesso anônimo aos usuários, com acesso restrito.

Uma vez conectado a um servidor 'ftp', você pode usar a maioria dos comandos do 'GNU/Linux' para operá-lo.

'ftp [_ip/dns_]'

Abaixo alguns dos comandos mais usados no FTP:

ls Lista arquivos do diretório atual.

cd [diretório] Entra em um diretório.

get [arquivo] Copia um arquivo do servidor ftp para o computador local. O arquivo é gravado, por padrão, no diretório onde o program ftp foi executado.

hash [on/off] Por padrão esta opção está desligada. Quando ligada, faz com que o caracter "#" seja impresso na tela indicando o progresso do download.

mget [arquivos] Semelhante ao get, mas pode copiar diversos arquivos e permite o uso de coringas.

send [arquivo] Envia um arquivo para o diretório atual do servidor FTP (você precisa de uma conta com acesso a gravação para fazer isto).

prompt [on/off] Ativa ou desativa a pergunta para a cópia de arquivo. Se estiver como 'off' assume sim para qualquer pergunta.

Exemplo: 'ftp ftp.br.debian.org'.

6.4 whoami

Mostra o nome que usou para se conectar ao sistema. É útil quando você usa várias contas e não sabe com qual nome entrou no sistema :-)

'whoiam'

6.5 dnsdomainname

Mostra o nome do domínio de seu sistema.

6.6 hostname

Mostra ou muda o nome de seu computador na rede.

6.7 talk

Inicia conversa com outro usuário em uma rede local ou Internet. Talk é um programa de conversação em tempo real onde uma pessoa vê o que a outra escreve.

'talk [*usuário*] [*tty*]'

ou

'talk [*usuário@host*]'

Onde:

usuário Nome de login do usuário que deseja iniciar a conversação. Este nome pode ser obtido com o comando 'who' .

tty O nome de terminal onde o usuário está conectado, para iniciar uma conexão local.

usuário@host Se o usuário que deseja conversar estiver conectado em um computador remoto, você deve usar o nome do usuário@hostn do computador. Após o 'talk' ser iniciado, ele verificará se o usuário pode receber mensagens, em caso positivo, ele enviará uma mensagem ao usuário dizendo como responder ao seu pedido de conversa.

7 Comandos para manipulação de contas

Este capítulo traz comandos usados para manipulação de conta de usuários e grupos em sistemas 'GNU/Linux'. Entre os assuntos descritos aqui estão adicionar usuários ao sistema, adicionar grupos, incluir usuários existente em novos grupos, etc.

7.1 adduser

Adiciona um usuário ou grupo no sistema. Por padrão, quando um novo usuário é adicionado, é criado um grupo com o mesmo nome do usuário. Será criado um diretório home com o nome do usuário (a não ser que o novo usuário criado seja um usuário do sistema) e este receberá uma identificação. A identificação do usuário (UID) escolhida será a primeira disponível no sistema especificada de acordo com a faixa de UIDS de usuários permitidas no arquivo de configuração '/etc/adduser.conf'. Este é o arquivo que contém os padrões para a criação de novos usuários no sistema.

'adduser [*opções*] [*usuário/grupo*]'

Onde:

usuário/grupo Nome do novo usuário que será adicionado ao sistema.

opções

-disable-passwd Não executa o programa 'passwd' para escolher a senha e somente permite o uso da conta após o usuário escolher uma senha.

–force-badname Desativa a checagem de senhas ruins durante a adição do novo usuário. Por padrão o 'adduser' checa se a senha pode ser facilmente adivinhada.

–group Cria um novo grupo ao invés de um novo usuário. A criação de grupos também pode ser feita pelo comando 'addgroup'.

-uid [num] Cria um novo usuário com a identificação [num] ao invés de procurar o próximo UID disponível.

-gid [num] Faz com que o usuário seja parte do grupo [gid] ao invés de pertencer a um novo grupo que será criado com seu nome. Isto é útil caso deseje permitir que grupos de usuários possam ter acesso a arquivos comuns. Caso estiver criando um novo grupo com 'adduser', a identificação do novo grupo será [num].

–home [dir] Usa o diretório [dir] para a criação do diretório home do usuário ao invés de usar o especificado no arquivo de configuração '/etc/adduser.conf'.

–ingroup [nome] Quando adicionar um novo usuário no sistema, coloca o usuário no grupo [nome] ao invés de criar um novo grupo.

–quiet Não mostra mensagens durante a operação.

–system Cria um usuário de sistema ao invés de um usuário normal.

Os dados do usuário são colocados no arquivo '/etc/passwd' após sua criação e os dados do grupo são colocados no arquivo '/etc/group'.

OBSERVAÇÃO: Caso esteja usando senhas ocultas (shadow passwords), as senhas dos usuários serão colocadas no arquivo '/etc/shadow' e as senhas dos grupos no arquivo '/etc/gshadow'. Isto aumenta mais a segurança do sistema porque somente o usuário 'root' pode ter acesso a estes arquivos, ao contrário do arquivo '/etc/passwd' que possui os dados de usuários e devem ser lidos por todos.

7.2 addgroup

Adiciona um novo grupo de usuários no sistema. As opções usadas são as mesmas do Seção 10.1, 'adduser'.

'addgroup [*usuário/grupo*] [*opções*]'

7.3 passwd

Muda a senha do usuário ou grupo. Um usuário somente pode alterar a senha de sua conta, mas o superusuário (‘root’) pode alterar a senha de qualquer conta de usuário, inclusive a data de validade da conta, etc. Os donos de grupos também podem alterar a senha do grupo com este comando.

Os dados da conta do usuário como nome, endereço, telefone, também podem ser alterados com este comando.

‘passwd [*usuário/grupo*] [*opções*]’

Onde:

usuário Nome do usuário/grupo que terá sua senha alterada.

opções

-g Se especificada, a senha do grupo será alterada. Somente o root ou o administrador do grupo pode alterar sua senha. A opção -r pode ser usada com esta para remover a senha do grupo. A opção -R pode ser usada para restringir o acesso do grupo para outros usuários.

Procure sempre combinar letras maiúsculas, minúsculas, e números ao escolher suas senhas. Não é recomendado escolher palavras normais como sua senha pois podem ser vulneráveis a ataques de dicionários cracker. Outra recomendação é utilizar *senhas ocultas* em seu sistema (*shadow password*).

Você deve ser o dono da conta para poder modificar a senhas. O usuário root pode modificar/apagar a senha de qualquer usuário. Exemplo: ‘passwd root’.

7.4 userdel

Apaga um usuário do sistema. Quando é usado, este comando apaga todos os dados da conta especificado dos arquivos de contas do sistema.

‘userdel [*-r*] [*usuário*]’

Onde:

-r Apaga também o diretório HOME do usuário.

OBS: Note que uma conta de usuário não poderá ser removida caso ele estiver no sistema, pois os programas podem precisar ter acesso aos dados dele (como UID, GID) no ‘/etc/passwd’.

7.5 groupdel

Apaga um grupo do sistema. Quando é usado, este comando apaga todos os dados do grupo especificado dos arquivos de contas do sistema.

‘groupdel [*grupo*]’

Tenha certeza que não existem arquivos/diretórios criados com o grupo apagado através do comando ‘find’.

OBS: Você não pode remover o grupo primário de um usuário. Remova o usuário primeiro.

7.6 Adicionando um novo grupo a um usuário

Para incluir um novo grupo a um usuário, e assim permitir que ele acesse os arquivos/diretórios que pertencem àquele grupo, você deve estar como root e editar o arquivo ‘/etc/group’. Este arquivo possui o seguinte formato:

NomedoGrupo:senha:GID:usuários

Onde:

NomedoGrupo É o nome daquele grupo de usuários.

senha Senha para ter acesso ao grupo. Caso esteja utilizando senhas ocultas para grupos, as senhas estarão em ‘/etc/gshadow’.

GID Identificação numérica do grupo de usuário.

usuários Lista de usuários que também fazem parte daquele grupo. Caso exista mais de um nome de usuário, eles devem estar separados por vírgula.

Deste modo para acrescentar o usuário "joao" ao grupo ‘audio’ para ter acesso aos dispositivos de som do Linux, acrescente o nome no final da linha: "audio:x:100:joao". Pronto, basta digitar ‘logout’ e entrar novamente com seu nome e senha, você estará fazendo parte do grupo ‘audio’ (confira digitando ‘groups’ ou ‘id’).

Outros nomes de usuários podem ser acrescentados ao grupo ‘audio’ bastando separar os nomes com vírgula.

7.7 id

Mostra a identificação atual do usuário, grupo primário e outros grupos que pertence.

‘id [*opções*] [*usuário*]’

Onde:

usuário É o usuário que desejamos ver a identificação, grupos primários e complementares.

opções

-g, –group Mostra somente a identificação do grupo primário.

-G, –groups Mostra a identificação de outros grupos que pertence.

-n, –name Mostra o nome do usuário e grupo ao invés da identificação numérica.

-u, –user Mostra somente a identificação do usuário (user ID).

-r, –real Mostra a identificação real de usuário e grupo, ao invés da efetiva. Esta opção deve ser usada junto com uma das opções: -u, -g, ou -G. Caso não sejam especificadas opções, ‘id’ mostrará todos os dados do usuário.

Exemplo: ‘id’, ‘id –user’, ‘id -r -u’.

7.8 users

Mostra os nomes de usuários usando atualmente o sistema. Os nomes de usuários são mostrados através de espaços sem detalhes adicionais.

‘users’

Os nomes de usuários atualmente conectados ao sistema são obtidos do arquivo ‘/var/log/wtmp’.

7.9 groups

Mostra os grupos que o usuário pertence.

‘groups [*usuário*]’

Exemplo: ‘groups’, ‘groups root’

8 Permissões de acesso a arquivos e diretórios

A permissão de acesso protege o sistema de arquivos Linux do acesso indevido de pessoas ou programas não autorizados.

A permissão de acesso do ‘GNU/Linux’ também impede que um programa mal intencionado, por exemplo, apague um arquivo que não deve, envie arquivos para outra pessoa ou forneça acesso da rede para que outros usuários invadam o sistema. O sistema ‘GNU/Linux’ é muito seguro e como qualquer outro sistema seguro e confiável impede que usuários iniciantes (ou mal intencionados) instalem programas enviados por terceiros sem saber para que eles realmente servem e causem danos irreverssíveis em seus arquivos, seu micro ou sua empresa.

Esta seção pode se tornar um pouco difícil de se entender, então recomendo ler e ao mesmo tempo prática-la para uma ótima compreensão.

8.1 Donos, grupos e outros usuários

O princípio da segurança no sistema de arquivos ‘GNU/Linux’ é definir o acesso aos arquivos por donos, grupos e outros usuários:

dono É a pessoa que criou o arquivo ou o diretório. O nome do dono do arquivo/diretório é o mesmo do usuário usado para entrar no sistema ‘GNU/Linux’. Somente o dono pode modificar as permissões de acesso do arquivo. As permissões de acesso do dono de um arquivo somente se aplicam ao dono do arquivo/diretório. A identificação do dono também é chamada de user id (UID). A identificação de usuário e o nome do grupo que pertence são armazenadas respectivamente nos arquivos ‘/etc/passwd’ e ‘/etc/group’. Estes são arquivos textos comuns e podem ser editados em qualquer editor de texto, mas tenha cuidado para não modificar o campo que contém a senha do usuário encriptada (que pode estar armazenada neste arquivo caso não estiver usando senhas ocultas).

grupo Para permitir que vários usuários diferentes tivessem acesso a um mesmo arquivo (já que somente o dono poderia ter acesso ao arquivo), este recurso foi criado. Cada usuário pode fazer parte de um ou mais grupos e então acessar arquivos que pertençam ao mesmo grupo que o seu (mesmo que estes arquivos tenham outro *dono*). Por padrão, quando um novo usuário é criado, o grupo ele pertencerá será o mesmo de seu grupo primário (exceto pelas condições que explicarei adiante) . A identificação do grupo é chamada de 'gid (*group id*)'. Um usuário pode pertencer a um ou mais grupos.

outros É a categoria de usuários que não são donos ou não pertencem ao grupo do arquivo. Cada um dos tipos acima possuem três tipos básicos de permissões de acesso que serão vistas na próxima seção.

OBSERVAÇÕES :

- O usuário 'root' não tem nenhuma restrição de acesso ao sistema.
- Se você tem permissões de gravação no diretório e tentar apagar um arquivo que você não tem permissão de gravação, o sistema perguntará se você confirma a exclusão do arquivo apesar do modo leitura. Caso você tenha permissões de gravação no arquivo, o arquivo será apagado por padrão sem mostrar nenhuma mensagem de erro (a não ser que seja especificada a opção -i com o comando 'rm').
- Por outro lado, mesmo que você tenha permissões de gravação em um arquivo mas não tenha permissões de gravação em um diretório, a exclusão do arquivo será negada.

Isto mostra que é levado mais em consideração a permissão de acesso do diretório do que as permissões dos arquivos e sub-diretórios que ele contém. Este ponto é muitas vezes ignorado por muitas pessoas e expõem seu sistema a riscos de segurança. Imagine o problema que algum usuário que não tenha permissão de gravação em um arquivo mas que a tenha no diretório pode causar em um sistema mal administrado.

8.2 Permissões de Acesso Especiais

Em adição as três permissões básicas (rwx), existem permissões de acesso especiais (stX) que afetam arquivos executáveis e diretórios:

- 's' - Quando é usado na permissão de acesso do *Dono* , ajusta a identificação efetiva do usuário do processo durante a execução de um programa, também chamado de *bit setuid* . Não tem efeito em diretórios. Quando 's' é usado na permissão de acesso do *Grupo* , ajusta a identificação efetiva do grupo do processo durante a execução de um programa, chamado de *bit setgid* . É identificado pela letra 's' no lugar da permissão de execução do grupo do arquivo/diretório. Em diretórios, força que os arquivos criados dentro dele pertençam ao mesmo grupo do diretório, ao invés do grupo primário que o usuário pertence. Ambos *setgid* e *setuid* podem aparecer ao mesmo tempo no mesmo arquivo/diretório. A permissão de acesso especial 's' somente pode aparecer no campo *Dono e Grupo* .
- 't' - Salva a imagem do texto do programa no dispositivo swap, assim ele será carregado mais rapidamente quando executado, também chamado de *stick bit*. Em diretórios, impede que outros usuários removam arquivos dos quais não são donos. Isto é chamado de colocar o diretório em modo 'append-only'. Um exemplo de diretório que se encaixa perfeitamente nesta condição é o '/tmp', todos os usuários devem ter acesso para que seus programas possam criar os arquivos temporários lá, mas nenhum pode apagar arquivos dos outros. A permissão especial 't', pode ser especificada somente no campo outros usuários das permissões de acesso.
- 'X' - Se você usar 'X' ao invés de 'x', a permissão de execução somente é afetada se o arquivo já tiver permissões de execução. Em diretórios ela tem o mesmo efeito que a permissão de execução 'x'.

Exemplo da permissão de acesso especial 'X':

1. Crie um arquivo 'teste' (digitando 'touch teste') e defina sua permissão para 'rw-rw-r–' ('chmod ug=rw,o=r teste' ou 'chmod 664 teste').
2. Agora use o comando 'chmod a+X teste'
3. digite 'ls -l'
4. Veja que as permissões do arquivo não foram afetadas.
5. agora digite 'chmod o+x teste'
6. digite 'ls -l', você colocou a permissão de execução para os outros usuários.
7. Agora use novamente o comando 'chmod a+X teste'

8. digite ‘ls -l’
9. Veja que agora a permissão de execução foi concedida a todos os usuários, pois foi verificado que o arquivo era executável (tinha permissão de execução para outros usuários).
10. Agora use o comando ‘chmod a-X teste’
11. Ele também funcionará e removerá as permissões de execução de todos os usuários, porque o arquivo ‘teste’ tem permissão de execução (confira digitando ‘ls -l’).
12. Agora tente novamente o ‘chmod a+X teste’
13. Você deve ter reparado que a permissão de acesso especial ‘X’ é semelhante a ‘x’, mas somente faz efeito quanto o arquivo já tem permissão de execução para o dono, grupo ou outros usuários.

Em diretórios, a permissão de acesso especial ‘X’ funciona da mesma forma que ‘x’, até mesmo se o diretório não tiver nenhuma permissão de acesso (‘x’).

8.3 A conta root

Esta seção foi retirada do Manual de Instalação da Debian .

A conta root é também chamada de *super usuário* , este é um login que não possui restrições de segurança. A conta root somente deve ser usada para fazer a administração do sistema, e usada o menor tempo possível.

Qualquer senha que criar deverá conter de 6 a 8 caracteres, e também poderá conter letras maiúsculas e minúsculas, e também caracteres de pontuação. Tenha um cuidado especial quando escolher sua senha root, porque ela é a conta mais poderosa. Evite palavras de dicionário ou o uso de qualquer outros dados pessoais que podem ser adivinhados.

Se qualquer um lhe pedir senha root, seja extremamente cuidadoso. Você normalmente nunca deve distribuir sua conta root, a não ser que esteja administrando um computador com mais de um administrador do sistema.

Utilize uma conta de usuário normal ao invés da conta root para operar seu sistema. Porque não usar a conta root? Bem, uma razão para evitar usar privilégios root é por causa da facilidade de se cometer danos irreparáveis como root. Outra razão é que você pode ser enganado e rodar um programa *Cavalo de Tróia* – que é um programa que obtém poderes do *super usuário* para comprometer a segurança do seu sistema sem que você saiba.

8.4 chmod

Muda a permissão de acesso a um arquivo ou diretório. Com este comando você pode escolher se usuário ou grupo terá permissões para ler, gravar, executar um arquivo ou arquivos. Sempre que um arquivo é criado, seu dono é o usuário que o criou e seu grupo é o grupo do usuário (exceto para diretórios configurados com a permissão de grupo ‘"s"’, será visto adiante).

‘chmod [*opções*] [*permissões*] [*diretório/arquivo*]’

Onde:

diretório/arquivo Diretório ou arquivo que terá sua permissão mudada.

opções

-v, –verbose Mostra todos os arquivos que estão sendo processados.

-f, –silent Não mostra a maior parte das mensagens de erro.

-c, –change Semelhante a opção -v, mas só mostra os arquivos que tiveram as permissões alteradas.

-R, –recursive Muda permissões de acesso do _diretório/arquivo_ no diretório atual e sub-diretórios.

ugoa+-=rwxXst

ugoa Controla que nível de acesso será mudado. Especificam, em ordem, usuário (u), grupo (g), outros (o), todos (a).

+-= + coloca a permissão, - retira a permissão do arquivo e = define a permissão exatamente como especificado.

rwx *r* permissão de leitura do arquivo. *w* permissão de gravação. *x* permissão de execução (ou acesso a diretórios).

‘chmod’ não muda permissões de links simbólicos, as permissões devem ser mudadas no arquivo alvo do link. Também podem ser usados códigos numéricos octais para a mudança das permissões de acesso a arquivos/diretórios.

DICA: É possível copiar permissões de acesso do arquivo/diretório, por exemplo, se o arquivo ‘teste.txt’ tiver a permissão de acesso ‘r-xr——’ e você digitar ‘chmod o=u’, as permissões de acesso dos outros usuários (o) serão idênticas ao do dono (u). Então a nova permissão de acesso do arquivo ‘teste.txt’ será ‘r-xr–r-x’

Exemplos de permissões de acesso:

‘chmod g+r *’

Permite que todos os usuários que pertençam ao grupo dos arquivos (g) tenham (+) permissões de leitura (r) em todos os arquivos do diretório atual.

‘chmod o-r teste.txt’

Retira (-) a permissão de leitura (r) do arquivo ‘teste.txt’ para os outros usuários (usuários que não são donos e não pertencem ao grupo do arquivo ‘teste.txt’).

‘chmod uo+x teste.txt’

Inclui (+) a permissão de execução do arquivo ‘teste.txt’ para o dono e outros usuários do arquivo.

‘chmod a+x teste.txt’

Inclui (+) a permissão de execução do arquivo ‘teste.txt’ para o dono, grupo e outros usuários.

‘chmod a=rw teste.txt’

Define a permissão de todos os usuários exatamente (=) para leitura e gravação do arquivo ‘teste.txt’.

8.5 chgrp

Muda o grupo de um arquivo/diretório.

‘chgrp [_*opções*_] [*grupo*] [*arquivo/diretório*]’

Onde:

grupo Novo grupo do *arquivo/diretório*.

arquivo/diretório Arquivo/diretório que terá o grupo alterado.

opções

-c, –changes Somente mostra os arquivos/grupos que forem alterados.

-f, –silent Não mostra mensagens de erro para arquivos/diretórios que não puderam ser alterados.

-v, –verbose Mostra todas as mensagens e arquivos sendo modificados.

-R, –recursive Altera os grupos de arquivos/sub-diretórios do diretório atual.

8.6 chown

Muda dono de um arquivo/diretório. Opcionalmente pode também ser usado para mudar o grupo.

‘chown [*opções*] [*dono.grupo*] [*diretório/arquivo*]’

onde:

dono.grupo Nome do *dono.grupo* que será atribuído ao *diretório/arquivo* . O grupo é opcional.

diretório/arquivo Diretório/arquivo que o dono.grupo será modificado.

opções

-v, –verbose Mostra os arquivos enquanto são alterados.

-f, –supress Não mostra mensagens de erro durante a execução do programa.

-c, –changes Mostra somente arquivos que forem alterados.

-R, –recursive Altera dono e grupo de arquivos no diretório atual e sub-diretórios.

O *dono.grupo* pode ser especificado usando o nome de grupo ou o código numérico correspondente ao grupo (GID).

Você deve ter permissões de gravação no diretório/arquivo para alterar seu dono/grupo.

- 'chown joao teste.txt' - Muda o dono do arquivo 'teste.txt' para 'joao'.
- 'chown joao.users teste.txt' - Muda o dono do arquivo 'teste.txt' para 'joao' e seu grupo para 'users'.
- 'chown -R joao.users *' - Muda o dono/grupo dos arquivos do diretório atual e sub-diretórios para 'joao/users' (desde que você tenha permissões de gravação no diretórios e sub-diretórios).

8.7 Modo de permissão octal

Este modo de permissão é o mais fácil de ser compreendido e composto para as permissões de usuário, grupo e outros. Segue abaixo a correlação:

$r = 2^2 = 4$
$w = 2^1 = 2$
$x = 2^0 = 1$

Desta forma $rwx = 7$. Então para a seguinte notação seria :
$-rwxr - xr - x = 755$

8.8 umask

A umask (*user mask*) são 3 números que definem as permissões iniciais do 'dono', 'grupo' e 'outros usuários' que o arquivo/diretório receberá quando for criado ou copiado. Digite 'umask' sem parâmetros para retornar o valor de sua umask atual.

A umask tem efeitos diferentes caso o arquivo que estiver sendo criado for *binário* (um programa executável) ou *texto* .

Um *arquivo texto* criado com o comando 'umask 012;touch texto.txt' receberá as permissões '-rw-rw-r–', pois 0 (dono) terá permissões 'rw-', 1 (grupo), terá permissões 'rw-' e 2 (outros usuários) terão permissões 'r–'. Um *arquivo binári*o copiado com o comando 'umask 012;cp /bin/ls /tmp/ls' receberá as permissões '-r-xr–r-x' .

Por este motivo é preciso um pouco de atenção antes de escolher a umask, um valor mal escolhido poderia causar problemas de acesso a arquivos, diretórios ou programas não sendo executados. O valor padrão da umask na maioria das distribuições atuais é 022 . A umask padrão no sistema Debian é a 022 .

A umask é de grande utilidade para programas que criam arquivos/diretórios temporários, desta forma pode-se bloquear o acesso de outros usuários desde a criação do arquivo, evitando recorrer ao 'chmod'.

9 Redirecionamentos e Pipe

Esta seção explica o funcionamento dos recursos de direcionamento de entrada e saída do sistema 'GNU/Linux'.

9.1 >

Redireciona a saída de um programa/comando/script para algum dispositivo ou arquivo ao invés do dispositivo de saída padrão (tela). Quando é usado com arquivos, este redirecionamento cria ou substitui o conteúdo do arquivo.

Por exemplo, você pode usar o comando 'ls' para listar arquivos e usar 'ls >listagem' para enviar a saída do comando para o arquivo 'listagem'. Use o comando 'cat' para visualizar o conteúdo do arquivo 'listagem'.

O mesmo comando pode ser redirecionado para o segundo console '/dev/tty2' usando: 'ls >/dev/tty2', o resultado do comando 'ls' será mostrado no segundo console (pressione 'ALT' e 'F2' para mudar para o segundo console e 'ALT' e 'F1' para retornar ao primeiro).

9.2 > >

Redireciona a saída de um programa/comando/script para algum dispositivo ou final de arquivo ao invés do dispositivo de saída padrão (tela). A diferença entre este redirecionamento duplo e o simples, é se caso for usado com arquivos, adiciona a saída do comando ao final do arquivo existente ao invés de substituir seu conteúdo.

Por exemplo, você pode acrescentar a saída do comando 'ls' ao arquivo 'listagem' do capítulo anterior usando 'ls / > >listagem'. Use o comando 'cat' para visualizar o conteúdo do arquivo 'listagem'.

O mesmo comando pode ser redirecionado para o segundo console ‘/dev/tty2’ usando: ‘ls >/dev/tty2’, o resultado do comando ‘ls’ será mostrado no segundo console (pressione ‘ALT’ e ‘F2’ para mudar para o segundo console e ‘ALT’ e ‘F1’ para retornar ao primeiro).

9.3 <

Direciona a entrada padrão de arquivo/dispositivo para um comando.

Este comando faz o contrário do anterior, ele envia dados ao comando.

Você pode usar o comando ‘cat <teste.txt’ para enviar o conteúdo do arquivo ‘teste.txt’ ao comando ‘cat’ que mostrará seu conteúdo (é claro que o mesmo resultado pode ser obtido com ‘cat teste.txt’ mas este exemplo serviu para mostrar a funcionalidade do <).

9.4 | (pipe)

Envia a saída de um comando para a entrada do próximo comando para continuidade do processamento. Os dados enviados são processados pelo próximo comando que mostrará o resultado do processamento.

Por exemplo: ‘ls -la|more’, este comando faz a listagem longa de arquivos que é enviado ao comando ‘more’ (que tem a função de efetuar uma pausa a cada 25 linhas do arquivo).

Outro exemplo é o comando ‘"locate find|grep bin/"’, neste comando todos os caminhos/arquivos que contém _find_ na listagem serão mostrados (inclusive man pages, bibliotecas, etc.), então enviamos a saída deste comando para ‘grep bin/’ para mostrar somente os diretórios que contém binários. Mesmo assim a listagem ocupe mais de uma tela, podemos acrescentar o ‘more’: ‘locate find|grep bin/|more’.

Podem ser usados mais de um comando de redirecionamento (<, >, |) em um mesmo comando.

9.5 Diferença entre o "|" e o ">"

A principal diferença entre o "|" e o ">", é que o Pipe envolve processamento entre comandos, ou seja, a saída de um comando é enviado a entrada do próximo e o ">" redireciona a saída de um comando para um arquivo/dispositivo.

Você pode notar pelo exemplo acima (‘ls -la|more’) que ambos ‘ls’ e ‘more’ são comandos porque estão separados por um "|"! Se um deles não existir ou estiver digitado incorretamente, será mostrada uma mensagem de erro.

Um resultado diferente seria obtido usando um ‘">"’ no lugar do ‘"|"’; A saída do comando ‘ls -la’ seria gravada em um arquivo chamado ‘more’.

9.6 tee

Envia o resultado do programa para a saída padrão (tela) e para um arquivo ao mesmo tempo. Este comando deve ser usado com o pipe "|".

‘ *comando* |tee [*arquivo*]’

Exemplo: ‘ls -la|tee listagem.txt’, a saída do comando será mostrada normalmente na tela e ao mesmo tempo gravada no arquivo ‘listagem.txt’.

10 Impressão

Este capitulo descreve como imprimir em seu sistema ‘GNU/Linux’ e as formas de impressão via spool.

Antes de seguir os passos descritos neste capítulo, tenha certeza que seu kernel foi compilado com o suporte a impressora paralela ativado, caso contrário até mesmo a impressão direta para a porta de impressora falhará.

10.1 Portas de impressora

Uma porta de impressora é o local do sistema usado para se comunicar com a impressora. Em sistemas ‘GNU/Linux’, a porta de impressora é identificada como ‘lp0, lp1, lp2’ no diretório ‘/dev’, correspondendo respectivamente a ‘LPT1’, ‘LPT2’ e ‘LPT3’ no ‘DOS’ e ‘Windows’. Recomendo que o suporte a porta paralela esteja compilado como módulo no kernel.

10.2 Imprimindo diretamente para a porta de impressora

Isto é feito direcionando a saída ou o texto com '>' diretamente para a porta de impressora no diretório '/dev'. Supondo que você quer imprimir o texto contido do arquivo 'trabalho.txt' e a porta de impressora em seu sistema é '/dev/lp0', você pode usar os seguintes comandos:

- 'cat trabalho.txt >/dev/lp0' - Direciona a saída do comando 'cat' para a impressora.
- 'cat <trabalho.txt >/dev/lp0'. Faz a mesma coisa que o acima.
- 'cat -n trabalho.txt >/dev/lp0' - Numera as linhas durante a impressão.
- 'head -n 30 trabalho.txt >/dev/lp0' - Imprime as 30 linhas iniciais do arquivo.
- 'cat trabalho.txt|tee /dev/lp0' - Mostra o conteúdo do 'cat' na tela e envia também para a impressora.

Os métodos acima servem somente para imprimir em modo texto (letras, números e caracteres semi-gráficos).

11 Rede

Este capítulo descreve o que é uma rede, os principais dispositivos de rede no 'GNU/Linux', a identificação de cada um, como configurar os dispositivos, escolha de endereços IP, roteamento.

11.1 O que é uma rede

Rede é a conexão de duas ou mais máquinas com o objetivo de compartilhar recursos entre uma máquina e outra. Os recursos podem ser:

- Compartilhamento do conteúdo de seu disco rígido (ou parte dele) com outros usuários. Os outros usuários poderão acessar o disco como se estivesse instalado na própria máquina). Também chamado de servidor de arquivos.
- Compartilhamento de uma impressora com outros usuários. Os outros usuários poderão enviar seus trabalhos para uma impressora da rede. Também chamado de servidor de impressão.
- Compartilhamento de acesso a Internet. Outros usuários poderão navegar na Internet, pegar seus e-mails, ler noticias, bate-papo no IRC, ICQ através do servidor de acesso Internet. Também chamado de servidor Proxy.
- Servidor de Internet/Intranet. Outros usuários poderão navegar nas páginas Internet localizadas em seu computador, pegar e-mails, usar um servidor de IRC para chat na rede, servidor de ICQ, etc.

Com os itens acima funcionando é possível criar permissões de acesso da rede, definindo quem terá ou não permissão para acessar cada compartilhamento ou serviço existente na máquina (www, ftp, irc, icq, etc), e registrando/avisando sobre eventuais tentativas de violar a segurança do sistema, firewalls, pontes, etc.

Entre outras ilimitadas possibilidades que dependem do conhecimento do indivíduo no ambiente 'GNU/Linux', já que ele permite muita flexibilidade para fazer qualquer coisa funcionar em rede.

A comunicação entre computadores em uma rede é feita através do *Protocolo de Rede*.

11.2 Protocolo de Rede

O protocolo de rede é a linguagem usada para a comunicação entre um computador e outro. Existem vários tipos de protocolos usados para a comunicação de dados, alguns são projetados para pequenas redes (como é o caso do NetBios) outros para redes mundiais (TCP/IP que possui características de roteamento).

Dentre os protocolos, o que mais se destaca atualmente é o TCP/IP devido ao seu projeto, velocidade e capacidade de roteamento.

11.3 Endereço IP

O *endereço IP* são números que identificam seu computador em uma rede. Inicialmente você pode imaginar o IP como um número de telefone. O IP é compostos por quatro bytes e a convenção de escrita dos números é chamada de "notação decimal pontuada". Por convenção, cada interface (placa usada p/ rede) do computador ou roteador tem um endereço IP. Também é permitido que o mesmo endereço IP seja usado em mais de uma interface de uma mesma máquina mas normalmente cada interface tem seu próprio endereço IP.

As Redes do Protocolo Internet são seqüências contínuas de endereços IP's. Todos os endereços dentro da rede tem um número de dígitos dentro dos endereços em comum. A porção dos endereços que são comuns entre todos os endereços de uma rede são

chamados de *porção da rede*. Os dígitos restantes são chamados de *porção dos hosts*. O número de bits que são compartilhados por todos os endereços dentro da rede são chamados de *netmask* (máscara da rede) e o papel da *netmask* é determinar quais endereços pertencem ou não a rede. Por exemplo, considere o seguinte:

Endereço do Host 192.168.110.23

Máscara da Rede 255.255.255.0 Porção da Rede 192.168.110.

Porção do Host .23

Endereço da Rede 192.168.110.0

Endereço Broadcast 192.168.110.255

Qualquer endereço que é finalizado em zero em sua *netmask*, revelará o *endereço da rede* que pertence. O endereço e rede é então sempre o menor endereço numérico dentro da escalas de endereços da rede e sempre possui a *porção host* dos endereços codificada como zeros.

O endereço de *broadcast* é um endereço especial que cada computador em uma rede "escuta" em adição a seu próprio endereço. Este é um endereço onde os datagramas enviados são recebidos por todos os computadores da rede. Certos tipos de dados como informações de roteamento e mensagens de alerta são transmitidos para o endereço *broadcast*, assim todo computador na rede pode recebe-las simultaneamente.

Existe dois padrões normalmente usados para especificar o endereço de *broadcast*. O mais amplamente aceito é para usar o endereço 'mais alto' da rede como endereço broadcast. No exemplo acima este seria 192.168.110.255. Por algumas razões outros sites tem adotado a convenção de usar o 'endereço de rede' como o endereço broadcast. Na prática não importa muito se usar este endereço, mas você deve ter certeza que todo computador na rede esteja configurado para escutar o mesmo *endereço broadcast*.

11.3.1 Classes de Rede IP

Por razões administrativas após algum pouco tempo no desenvolvimento do protocolo IP alguns grupos arbitrários de endereços foram formados em redes e estas redes foram agrupadas no que foram chamadas de *classes*. Estas classes armazenam um tamanho padrão de redes que podem ser usadas. As faixas alocadas são:

Classede rede	Máscara de Rede	Endereço da Rede
A	255.0.0.0	0.0.0.0- 127.255.255.255
B	255.255.0.0	128.0.0.0 - 191.255.255.255
C	255.255.255.0	192.0.0.0 - 223.255.255.255
Multicast	240.0.0.0	224.0.0.0 - 239.255.255.255

O tipo de endereço que você deve utilizar depende exatamente do que estiver fazendo.

11.3.2 Para instalar uma máquina usando o Linux em uma rede existente

Se você quiser instalar uma máquina 'GNU/Linux' em uma rede TCP/IP existente então você deve contatar qualquer um dos administradores da sua rede e perguntar o seguinte:

- Endereço IP de sua máquina
- Endereço IP da rede
- Endereço IP de broadcast
- Máscara da Rede IP
- Endereço do Roteador
- Endereço do Servidor de Nomes (DNS)

Você deve então configurar seu dispositivo de rede 'GNU/Linux' com estes detalhes. Você não pode simplesmente escolhe-los e esperar que sua configuração funcione.

11.3.3 Endereços reservados para uso em uma rede Privada

Se você estiver construindo uma rede privada que nunca será conectada a Internet, então você pode escolher qualquer endereço que quiser. No entanto, para sua segurança e padronização, existem alguns endereços IP's que foram reservados especificamente para este propósito. Eles estão especificados no RFC1597 e são os seguintes:

	ENDEREÇOS RESERVADOS PARA REDES PRIVADAS	
Classede rede	Máscara de Rede	Endereço da Rede
A	255.0.0.0	10.0.0.0 - 10.255.255.255

B	255.255.0.0	172.16.0.0 - 172.31.255.255
C	255.255.255.0	192.168.0.0 - 192.168.255.255

Você deve decidir primeiro qual será a largura de sua rede e então escolher a classe de rede que será usada.

11.4 Interface de rede

As interfaces de rede no 'GNU/Linux' estão localizadas no diretório '/dev' e a maioria é criada dinamicamente pelos softwares quando são requisitadas. Este é o caso das interfaces 'ppp' e 'plip' que são criadas dinamicamente pelos softwares.

Abaixo a identificação de algumas interfaces de rede no Linux (a '?' significa um número que identifica as interfaces seqüencialmente, iniciando em 0):

- 'eth?' - Placa de rede Ethernet e WaveLan.
- 'ppp?' - Interface de rede PPP (protocolo ponto a ponto).
- 'slip?' - Interface de rede serial
- 'eql' - Balanceador de tráfego para múltiplas linhas
- 'plip?' - Interface de porta paralela
- 'arc?e, arc?s' - Interfaces Arcnet
- 'sl?, ax?' - Interfaces de rede AX25 (respectivamente para kernels 2.0.xx e 2.2.xx.
- 'fddi?' - Interfaces de rede FDDI.
- 'dlci??, sdla?' - Interfaces Frame Relay, respectivamente para para dispositivos de encapsulamento DLCI e FRAD.
- 'nr?' - Interface Net Rom
- 'rs?' - Interfaces Rose
- 'st?' - Interfaces Strip (Starmode Radio IP)
- 'tr?' - Token Ring

Para maiores detalhes sobre as interfaces acima, consulte o documento *NET3-4-HOWTO*.

11.4.1 A interface loopback

A interface *loopback* é um tipo especial de interface que permite fazer conexões com você mesmo. Todos os computadores que usam o protocolo TCP/IP utilizam esta interface e existem várias razões porque precisa fazer isto, por exemplo, você pode testar vários programas de rede sem interferir com ninguém em sua rede. Por convenção, o endereço IP 127.0.0.1 foi escolhido especificamente para a loopback, assim se abrir uma conexão telnet para 127.0.0.1, abrirá uma conexão para o próprio computador local.

A configuração da interface loopback é simples e você deve ter certeza que fez isto (mas note que esta tarefa é normalmente feita pelos scripts padrões de inicialização existentes em sua distribuição).

```
ifconfig lo 127.0.0.1
```

Caso a interface loopback não esteja configurada, você poderá ter problemas quando tentar qualquer tipo de conexão com as interfaces locais, tendo problemas até mesmo com o comando 'ping'.

11.4.2 Atribuindo um endereço de rede a uma interface (ifconfig)

Após configurada fisicamente, a interface precisa receber um endereço IP para ser identificada na rede e se comunicar com outros computadores, além de outros parâmetros como o endereço de *broadcast* e a *máscara de rede*. O comando usado para fazer isso é o 'ifconfig' (interface configure).

Para configurar a interface de rede Ethernet ('eth0') com o endereço 192.168.1.1, máscara de rede 255.255.255.0, podemos usar o comando:

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
```

O comando acima ativa a interface de rede. A palavra ‘up’ pode ser omitida, pois a ativação da interface de rede é o padrão. Para desativar a mesma interface de rede, basta usar usar o comando:

```
ifconfig eth0 down
```

Digitando ‘ifconfig’ são mostradas todas as interfaces ativas no momento, pacotes enviados, recebidos e colisões de datagramas. Para mostrar a configuração somente da interface eth0, use o comando: ‘ifconfig eth0’

Para mais detalhes, veja a página de manual do ‘ifconfig’ ou o *NET3-4-HOWTO*.

11.5 Roteamento

Roteamento é quando uma máquina com múltiplas conexões de rede decide onde entregar os pacotes IP que recebeu, para que cheguem ao seu destino.

Pode ser útil ilustrar isto com um exemplo. Imagine um simples roteador de escritório, ele pode ter um link intermitente com a Internet, um número de segmentos ethernet alimentando as estações de trabalho e outro link PPP intermitente fora de outro escritório.

Quando o roteador recebe um datagrama de qualquer de suas conexões de rede, o mecanismo que usa determina qual a próxima interface deve enviar o datagrama. Computadores simples também precisam rotear, todos os computadores na Internet tem dois dispositivos de rede, um é a interface *loopback* (explicada acima) o outro é um usado para falar com o resto da rede, talvez uma ethernet, talvez uma interface serial PPP ou SLIP.

OK, viu como o roteamento funciona? cada computador mantém uma lista de regras especiais de roteamento, chamada *tabela de roteamento*. Esta tabela contém colunas que tipicamente contém no mínimo três campos, o primeiro é o *endereço de destino*, o segundo é o *nome da interface* que o datagrama deve ser roteado e o terceiro é opcionalmente o *endereço IP* da outra máquina que levará o datagrama em seu próximo passo através da rede. No ‘GNU/Linux’ você pode ver a tabela de roteamento usando um dos seguintes comandos:

```
cat /proc/net/route
route -n
netstat -r
```

O processo de roteamento é muito simples: um datagrama (pacote IP) é recebido, o endereço de destino (para quem ele é) é examinado e comparado com cada ítem da tabela de roteamento. O ítem que mais corresponder com o endereço é selecionado e o datagrama é direcionado a interface especificada.

Se o campo *gateway* estiver preenchido, então o datagrama é direcionado para aquele computador pela interface especificada, caso contrário o endereço de destino é assumido sendo uma rede suportada pela interface.

11.5.1 Configurando uma rota no Linux

A configuração da rota é feita através da ferramenta ‘route’. Para adicionar uma rota para a rede 192.168.1.0 acessível através da interface eth0 basta digitar o comando:

```
route add -net 192.168.1.0 eth0
```

Para apagar a rota acima da *tabela de roteamento*, basta substituir a palavra ‘add’ por ‘del’. A palavra ‘net’ quer dizer que 192.168.1.0 é um endereço de rede para especificar uma máquina de destino, basta usar a palavra ‘-host’. Endereços de máquina de destino são muito usadas em conexões de rede apenas entre dois pontos (como ppp, plip, slip). Por padrão, a interface é especificada como último argumento. Caso a interface precise especifica-la em outro lugar, ela deverá ser precedida da opção ‘-dev’.

Para adicionar uma rota padrão para um endereço que não se encontre na tabela de roteamento, utiliza-se o *gateway padrão da rede*. Através do gateway padrão é possível especificar um computador (normalmente outro gateway) que os pacotes de rede serão enviados caso o endereço não confira com os da tabela de roteamento. Para especificar o computador 192.168.1.1 como *gateway padrão* usamos:

```
route add default gw 192.168.1.1 eth0
```

O *gateway padrão* pode ser visualizado através do comando ‘route -n’ e verificando o campo ‘gateway’. A opção ‘gw’ acima, especifica que o próximo argumento é um endereço IP (de uma rede já acessível através das tabelas de roteamento).

O computador *gateway* está conectado a duas ou mais redes ao mesmo tempo. Quando seus dados precisam ser enviados para computadores fora da rede, eles são enviados através do computador *gateway* e o *gateway* os encaminham ao endereço de destino. Desta forma, a resposta do servidor também é enviada através do *gateway* para seu computador (é o caso de uma típica conexão com a Internet).

A nossa configuração ficaria assim:

```
route add -net 192.168.1.0 eth0
route add default gw 192.168.1.1 eth0
```

Para mais detalhes, veja a página de manual do ‘route’ ou o *NET3-4-HOWTO*.

11.6 Resolvedor de nomes (DNS)

DNS significa Domain Name System (sistema de nomes de domínio). O *DNS* converte os nomes de máquinas para endereços IPs que todas as máquinas da Internet possuem. Ele faz o mapeamento do nome para o endereço e do endereço para o nome e algumas outras coisas. Um mapeamento é simplesmente uma associação entre duas coisas, neste caso um nome de computador, como www.metainfo.org, e o endereço IP desta máquina (ou endereços) como 200.245.157.9.

O *DNS* foi criado com o objetivo de tornar as coisas mais fáceis para o usuário, permitindo assim, a identificação de computadores na Internet ou redes locais através de nomes (é como se tivéssemos apenas que decorar o nome da pessoa ao invés de um número de telefone). A parte responsável por traduzir os nomes como 'www.nome.com.br' em um endereço IP é chamada de *resolvedor de nomes*.

O *resolvedor de nomes* pode ser um banco de dados local (controlador por um arquivo ou programa) que converte automaticamente os nomes em endereços IP ou através de *servidores DNS* que fazem a busca em um banco de dados na Internet e retornam o endereço IP do computador desejado. Um servidor DNS mais difundido na Internet é o 'bind'.

Através do DNS é necessário apenas decorar o endereço sem precisar se preocupar com o endereço IP (alguns usuários simplesmente não sabem que isto existe...). Se desejar mais detalhes sobre *DNS*, veja o documento DNS-HOWTO.

11.6.1 O que é um nome?

Você deve estar acostumado com o uso dos nomes de computadores na Internet, mas pode não entender como eles são organizados. Os nomes de domínio na Internet são uma estrutura hierárquica, ou seja, eles tem uma estrutura semelhante aos diretórios de seu sistema.

Um *domínio* é uma família ou grupo de nomes. Um domínio pode ser colocado em um *sub-domínio*. Um *domínio principal* é um domínio que não é um sub-domínio. Os domínios principais são especificados na RFC-920. Alguns exemplos de domínios principais comuns são:

- 'COM' - Organizações Comerciais
- 'EDU' - Organizações Educacionais
- 'GOV' - Organizações Governamentais
- 'MIL' - Organizações Militares
- 'ORG' - Outras Organizações
- 'NET' - Organizações relacionadas com a Internet
- 'Identificador do País' - São duas letras que representam um país em particular.

Cada um dos domínios principais tem sub-domínios. Os domínios principais baseados no nome do país são freqüentemente divididos em sub-domínios baseado nos domínios '.com', '.edu', '.gov', '.mil' e '.org'. Assim, por exemplo, você pode finaliza-lo com: 'com.au' e 'gov.au' para organizações comerciais e governamentais na Austrália; note que isto não é uma regra geral, as organizações de domínio atuais dependem da autoridade na escolha de nomes de cada domínio. Quando o endereço não especifica o domínio principal, como o endereço 'www.unicamp.br', isto quer dizer que é uma organização acadêmica.

O próximo nível da divisão representa o nome da organização. Subdomínios futuros variam em natureza, freqüentemente o próximo nível do sub-domínio é baseado na estrutura departamental da organização mas ela pode ser baseada em qualquer critério considerado razoável e significantes pelos administradores de rede para a organização.

A porção mais a direita do nome é sempre o nome único da máquina chamado *hostname*, a porção do nome a direita do hostname é chamado *nome de domínio* e o nome completo é chamado *nome do domínio completamente qualificado* (*Fully Qualified Domain Name*).

Usando o computador 'www.debian.org.br' como exemplo:

- 'br' - País onde o computador se encontra
- 'org' - Domínio principal
- 'debian' - Nome de Domínio
- 'www' - Nome do computador

A localização do computador 'www.debian.org.br' através de servidores DNS na Internet obedece exatamente a seqüência de procura acima. Os administradores do domínio 'debian.org.br' podem cadastrar quantos sub-domínios e computadores quiserem (como 'www.non-us.debian.org.br' ou 'cvs.debian.org.br').

11.6.2 Arquivos de configuração usados na resolução de nomes

Abaixo a descrição dos arquivos usados no processo de resolver um nome no sistema 'GNU/Linux'.

- /etc/resolv.conf

O '/etc/resolv.conf' é o arquivo de configuração principal do código do resolvedor de nomes. Seu formato é um arquivo texto simples com um parâmetro por linha e o endereço de servidores DNS externos são especificados nele. Existem três palavras chaves normalmente usadas que são:

domain Especifica o nome do domínio local.

search Especifica uma lista de nomes de domínio alternativos ao procurar por um computador, separados por espaços. A linha search pode conter no máximo 6 domínios ou 256 caracteres.

nameserver Especifica o endereço IP de um servidor de nomes de domínio para resolução de nomes. Pode ser usado várias vezes.

Como exemplo, o '/etc/resolv.conf' se parece com isto: domain maths.wu.edu.au search maths.wu.edu.au wu.edu.au nameserver 192.168.10.1 nameserver 192.168.12.1

Este exemplo especifica que o nome de domínio a adicionar ao nome não qualificado (ie hostnames sem o domínio) é 'maths.wu.edu.a e que se o computador não for encontrado naquele domínio então a procura segue para o domínio 'wu.edu.au' diretamente. Duas linhas de nomes de servidores foram especificadas, cada uma pode ser chamada pelo código resolvedor de nomes para resolver o nome.

- /etc/host.conf

O arquivo '/etc/host.conf' é o local onde é possível configurar alguns itens que gerenciam o código do resolvedor de nomes. O formato deste arquivo é descrito em detalhes na página de manual resolv+. Em quase todas as situações, o exemplo seguinte funcionará:

```
order hosts,bind multi on
```

Este arquivo de configuração diz ao resolvedor de nomes para checar o arquivo '/etc/hosts' (parâmetro 'hosts') antes de tentar verificar um *servidor de nomes* (parâmetro 'bind') e retornar um endereço IP válido para o computador procurado.

- /etc/hosts

O arquivo '/etc/hosts' é onde você coloca o endereço IP e o nome dos computadores locais. Se colocar um computador neste arquivo então você não precisará consultar o servidor de nomes para obter o endereço IP. A desvantagem de fazer isto é que você mesmo precisará manter este arquivo atualizado e se o endereço IP, se algum computador for modificado. Em um sistema bem gerenciado, os únicos endereços de computadores que aparecerão neste arquivo serão da interface loopback e os nomes de computadores.

```
# /etc/hosts
127.0.0.1 localhost
loopback 192.168.0.1 this.host.name
```

Você pode especificar mais que um nome de computador por linha como demonstrada pela primeira linha, a que identifica a interface loopback.

- /etc/networks

O arquivo '/etc/networks' tem uma função similar ao arquivos '/etc/hosts'. Ele contém um banco de dados simples de nomes de redes contra endereços de redes. Seu formato se difere por dois campos por linha e seus campos são identificados como:
Nome da Rede *Endereço da Rede*
Abaixo um exemplo de como se parece este arquivo:

```
loopnet 127.0.0.0
localnet 192.168.1.0
amprnet 44.0.0.0
```

Quando usar comandos como 'route', se um destino é uma rede e esta rede se encontra no arquivo '/etc/networks', então o comando 'route' mostrará o *nome da rede* ao invés de seu endereço.

11.6.3 Executando um servidor de nomes

Se você planeja executar um servidor de nomes, você pode fazer isto facilmente. Por favor veja o documento 'DNS-HOWTO' e quaisquer documentos incluídos em sua versão do BIND (Berkeley Internet Name Domain).

11.7 Serviços de Rede

Serviços de rede é o que está disponível para ser acessado pelo usuário. No TCP/IP, cada serviço é associado a um número chamado *porta* que é onde o servidor espera pelas conexões dos computadores clientes. Uma porta de rede pode se referenciada tanto pelo número como pelo nome do serviço.

Abaixo, alguns exemplos de portas padrões usadas em serviços TCP/IP:

- '21' - FTP (transferência de arquivos)
- '23' - Telnet (terminal virtual remoto)
- '25' - Smtp (envio de e-mails)
- '53' - DNS (resolvedor de nomes)
- '79' - Finger (detalhes sobre usuários do sistema)
- '80' - http (protocolo www - transferência de páginas Internet)
- '110' - Pop-3 (recebimento de mensagens)
- '119' - Nntp (usado por programas de noticias)

O arquivo padrão responsável pelo mapeamento do nome dos serviços e das portas mais utilizadas é o '/etc/services' .

11.7.1 Serviços iniciados como Daemons de rede

Serviços de rede iniciados como *daemons* ficam residente o tempo todo na memória 'esperando' que alguém se conecte (também chamado de *modo standalone*). Um exemplo de *daemon* é o servidor proxy 'squid' e o servidor web 'Apache' operando no modo *daemon*.

Alguns programas servidores oferecem a opção de serem executados como *daemons* ou através do *inetd*. É recomendável escolher *daemon* se o serviço for solicitado freqüentemente (como é o caso dos servidores web ou proxy).

Para verificar se um programa está rodando como *daemon*, basta digitar 'ps ax' e procurar o nome do programa, em caso positivo ele é um *daemon*.

Normalmente os programas que são iniciados como daemons possuem seus próprios recursos de segurança/autenticação para decidir quem tem ou não permissão de se conectar.

11.7.2 Serviços iniciados através do inetd

Serviços iniciados pelo *inetd* são carregados para a memória somente quando são solicitados. O controle de quais serviços podem ser carregados e seus parâmetros, são feitos através do arquivo '/etc/inetd.conf'.

Um *daemon* chamado 'inetd' lê as configurações deste arquivo e permanece residente na memória, esperando pela conexão dos clientes. Quando uma conexão é solicitada, o daemon *inetd* verifica as permissões de acesso nos arquivos '/etc/hosts.allow' e '/etc/hosts.deny' e carrega o programa servidor correspondente no arquivo '/etc/inetd.conf'. Um arquivo também importante neste processo é o '/etc/services' que faz o mapeamento das portas e nomes dos serviços.

Alguns programas servidores oferecem a opção de serem executados como *daemons* ou através do *inetd*. É recomendável escolher *inetd* se o serviço não for solicitado freqüentemente (como é o caso de servidores 'ftp', 'telnet', 'talk', etc). Abaixo segue o arquivo de configuração *inet*.

- /etc/inetd.conf

O arquivo '/etc/inetd.conf' é um arquivo de configuração para o daemon servidor _inetd_. Sua função é dizer ao 'inetd' o que fazer quando receber uma requisição de conexão para um serviço em particular. Para cada serviço que deseja aceitar conexões, você precisa dizer ao *inetd* qual daemon servidor executar e como executa-lo.

Seu formato é também muito simples. É um arquivo texto com cada linha descrevendo um serviço que deseja oferecer. Qualquer texto em uma linha seguindo uma "#" é ignorada e considerada um comentário. Cada linha contém sete campos separados por qualquer número de espaços em branco (tab ou espaços). O formato geral é o seguinte:

serviço *tipo soquete* proto opções usuário *caminho serv*. *opções serv*.

serviço É o serviço relevante a este arquivo de configuração pego do arquivo '/etc/services'.

tipo *soquete* Este campo descreve o tipo do soquete que este item utilizará, valores permitidos são: ‘stream’, ‘dgram’, ‘raw’, ‘rdm’, ou ‘seqpacket’. Isto é um pouco técnico de natureza, mas como uma regra geral, todos os serviços baseados em *tcp* usam ‘stream’ e todos os protocolos baseados em *udp* usam ‘dgram’. Somente alguns tipos de daemons especiais de servidores usam os outros valores.

protocolo O protocolo é considerado válido para esta item. Isto deve bater com um item apropriado no arquivo ‘/etc/services’ e tipicamente será tcp ou udp. Servidores baseados no Sun RPC (*Remote Procedure Call*), utilizam rpc/tcp ou rpc/udp.

opções Existem somente duas configurações para este campo. A configuração deste campo diz ao *inetd* se o programa servidor de rede libera o soquete após ele ser iniciado e então se inetd pode iniciar outra cópia na próxima requisição de conexão, ou se o inetd deve aguardar e assumir que qualquer servidor já em execução pegará a nova requisição de conexão. Este é um pequeno truque de trabalho, mas como uma regra, todos os servidores tcp devem ter este parâmetro ajustado para *nowai*t e a maior parte dos servidores udp deve tê-lo ajustado para *wait*. Foi alertado que existem algumas excessões a isto, assim deixo isto como exemplo se não estiver seguro.

usuário Este campo descreve que conta de usuário usuário no arquivo ‘/etc/passwd’ será escolhida como *dono* do daemon de rede quando este for iniciado. Isto é muito útil se você deseja diminuir os riscos de segurança. Você pode ajustar o usuário de qualquer item para o usuário *nobody*, assim se a segurança do servidor de redes é quebrada, a possibilidade de problemas é minimizada. Normalmente este campo é ajustado para *root*, porque muitos servidores requerem privilégios de usuário root para funcionarem corretamente.

caminho servidor Este campo é o caminho para o programa servidor atual que será executado.

argumentos servidor Este campo inclui o resto da linha e é opcional. Você pode colocar neste campo qualquer argumento da linha de comando que deseje passar para o daemon servidor quando for iniciado. Uma dica que pode aumentar significativamente a segurança de seu sistema é comentar (colocar uma ‘#’no inicio da linha) os serviços que não serão utilizados.

Abaixo um modelo de arquivo ‘/etc/inetd.conf’ usado em sistemas ‘Debian’:

```
# /etc/inetd.conf: veja inetd(8) para mais detalhes.
#
# Banco de Dados de configurações do servidor Internet
#
#
# Linhas iniciando com "#:LABEL:" ou "#<off>#" não devem
# ser alteradas a não ser que saiba o que está fazendo!
#
#
# Os pacotes devem modificar este arquivo usando update-inetd(8)
#
# <nome_serviço> <tipo_soquete> <proto> <opções> <usuário> <caminho_servidor> <args>
#
#:INTERNO: Serviços internos
#echo stream tcp nowait root internal
#echo dgram udp wait root internal
#chargen stream tcp nowait root internal
#chargen dgram udp wait root internal
#discard stream tcp nowait root internal
#discard dgram udp wait root internal
#daytime stream tcp nowait root internal
#daytime dgram udp wait root internal time stream tcp nowait root internal
#time dgram udp wait root internal
#:PADRÕES: Estes são serviços padrões.
#:BSD: Shell, login, exec e talk são protocolos BSD.
#shell stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rshd
#login stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind
#exec stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rexecd
talk dgram udp wait nobody.tty /usr/sbin/tcpd /usr/sbin/in.talkd
ntalk dgram udp wait nobody.tty /usr/sbin/tcpd /usr/sbin/in.ntalkd
#:MAIL: Mail, news e serviços uucp. smtp stream tcp nowait mail /usr/sbin/exim exim -bs
#:INFO: Serviços informativos
#:BOOT: O serviço Tftp é oferecido primariamente para a inicialização. Alguns sites
# o executam somente em máquinas atuando como "servidores de inicialização".
```

#:RPC: Serviços baseados em RPC
#:HAM-RADIO: serviços de rádio amador
#:OTHER: Outros serviços

11.8 Segurança da Rede e controle de Acesso

Deixe-me iniciar esta seção lhe alertando que a segurança da rede em sua máquina e ataques maliciosos são uma arte complexa. Uma regra importante é: "Não ofereça serviços de rede que não deseja utilizar".

Muitas distribuições vem configuradas com vários tipos de serviços que são iniciados automaticamente. Para melhorar, mesmo que insignificantemente, o nível de segurança em seu sistema você deve editar se arquivo '/etc/inetd.conf' e comentar (colocar uma "#") as linhas que contém serviços que não utiliza.

Bons candidatos são serviços tais como: 'shell', 'login', 'exec', 'uucp', 'ftp' e serviços de informação tais como 'finger', 'netstat' e 'sysstat'.

Existem todos os tipos de mecanismos de segurança e controle de acesso, eu descreverei os mais importantes deles.

11.8.1 /etc/ftpusers

O arquivo '/etc/ftpusers' é um mecanismo simples que lhe permite bloquear a conexão de certos usuários via *ftp*. O arquivo '/etc/ftpusers' é lido pelo programa daemon ftp (*ftpd*) quando um pedido de conexão é recebido. O arquivo é uma lista simples de usuários que não tem permissão de se conectar. Ele se parece com:

/etc/ftpusers - login de usuários bloqueados via ftp root uucp bin mail

11.8.2 /etc/securetty

O arquivo '/etc/securetty' lhe permite especificar que dispositivos 'tty' que o usuário *root* pode se conectar. O arquivo /etc/securetty é lido pelo programa login (normalmente '/bin/login'). Seu formato é uma lista de dispositivos 'tty' onde a conexão é permitida, em todos os outros, a entrada do usuário *root* é bloqueada.

/etc/securetty - terminais que o usuário root pode se conectar tty1 tty2 tty3 tty4

11.8.3 O mecanismo de controle de acessos tcpd

O programa 'tcpd' que você deve ter visto listado no mesmo arquivo '/etc/inetd.conf', oferece mecanismos de registro e controle de acesso para os serviços que esta configurado para proteger. Ele é um tipo de firewall simples e fácil de configurar que pode evitar tipos indesejados de ataques e registrar possíveis tentativas de invasão.

Quando é executado pelo programa inetd, ele lê dos arquivos contendo regras de acesso e permite ou bloqueia o acesso ao servidor protegendo adequadamente.

Ele procura nos arquivos de regras até que uma regra confira. Se nenhuma regra conferir, então ele assume que o acesso deve ser permitido a qualquer um. Os arquivos que ele procura em seqüência são: '/etc/hosts.allow' e '/etc/hosts.deny'. Eu descreverei cada um destes arquivos separadamente.

Para uma descrição completa desta facilidade, você deve verificar a página de manual apropriada (hosts_access (5) é um bom ponto de partida).

11.8.4 Arquivos de controle de acesso a rede

- /etc/hosts.allow

O arquivo '/etc/hosts.allow' é um arquivo de configuração do programa '/usr/sbin/tcpd'. O arquivo 'hosts.allow' contém regras descrevendo que hosts tem permissão de acessar um serviço em sua máquina.

O formato do arquivo é muito simples:

/etc/hosts.allow
#
lista de serviços: lista de hosts : comando

lista de serviços É uma lista de nomes de serviços separados por vírgula que esta regra se aplica. Exemplos de nomes de serviços são: 'ftpd', 'telnetd' e 'fingerd'.

lista de hosts É uma lista de nomes de hosts separada por vírgula. Você também pode usar endereços IP's aqui. Adicionalmente, você pode especificar nomes de computadores ou endereço IP usando caracteres coringas para atingir grupos de hosts. Exemplos incluem: 'gw.vk2ktj.ampr.org' para conferir com um endereço de computador específico, '.uts.edu.au' para atingir qualquer endereço de computador finalizando com aquele string. Use 200.200.200. para conferir com qualquer endereço IP iniciando com estes dígitos. Existem alguns parâmetros especiais para simplificar a configuração, alguns destes são: 'ALL' atinge todos endereços, 'LOCAL' atinge qualquer computador que não contém um "." (ie. está no mesmo domínio de sua

máquina) e ‘PARANOID’ atinge qualquer computador que o nome não confere com seu endereço (falsificação de nome). Existe também um último parâmetro que é também útil: o parâmetro ‘EXCEPT’ lhe permite fazer uma lista de exceções. Isto será coberto em um exemplo adiante.

comando É um parâmetro opcional. Este parâmetro é o caminho completo de um comando que deverá ser executado toda a vez que esta regra conferir. Ele pode executar um comando para tentar identificar quem esta conectado pelo host remoto, ou gerar uma mensagem via E-Mail ou algum outro alerta para um administrador de rede que alguém está tentando se conectar. Existem um número de expansões que podem ser incluídas, alguns exemplos comuns são: %h expande o endereço do computador que está conectado ou endereço se ele não possuir um nome, %d o nome do daemon sendo chamado. Se o computador tiver permissão de acessar um serviço através do ‘/etc/hosts.allow’, então o ‘/etc/hosts.deny’ não será consultado e o acesso será permitido.

Como exemplo:

```
# /etc/hosts.allow
#
# Permite que qualquer um envie e-mails
in.smtpd: ALL
# Permitir telnet e ftp somente para hosts locais e myhost.athome.org.au
telnetd, ftpd: LOCAL, myhost.athome.org.au
# Permitir finger para qualquer um mas manter um registro de quem é fingerd:
ALL: (finger @%h | mail -s "finger from %h" root)
```

Qualquer modificação no arquivo ‘/etc/hosts.allow’ entrará em ação após reiniciar o daemon *inetd*. Isto pode ser feito com o comando ‘kill -HUP [pid do inetd]’, o ‘pid’ do *inetd* pode ser obtido com o comando ‘ps ax|grep inetd’.

- /etc/hosts.deny

O arquivo ‘/etc/hosts.deny’ é um arquivo de configuração das regras descrevendo quais computadores não tem a permissão de acessar um serviço em sua máquina.

Um modelo simples deste arquivo se parece com isto:

```
# /etc/hosts.deny
#
# Bloqueia o acesso de computadores com endereços suspeitos
ALL: PARANOID
#
# Bloqueia todos os computadores
ALL: ALL
```

A entrada ‘PARANOID’ é realmente redundante porque a outra entrada nega tudo. Qualquer uma destas linhas pode fazer uma segurança padrão dependendo de seu requerimento em particular.

Tendo um padrão *ALL: ALL* no arquivo */etc/hosts.deny* e então ativando especificamente os serviços e permitindo computadores que você deseja no arquivo ‘/etc/hosts.allow’ é a configuração mais segura.

Qualquer modificação no arquivo ‘/etc/hosts.deny’ entrará em ação após reiniciar o daemon *inetd*. Isto pode ser feito com o comando ‘kill -HUP [pid do inetd]’, o ‘pid’ do *inetd* pode ser obtido com o comando ‘ps ax|grep inetd’.

- /etc/hosts.equiv

O arquivo ‘/etc/hosts.equiv’ é usado para garantir/bloquear certos computadores e usuários o direito de acesso aos serviços "r*" (rsh, rexec, rcp, etc) **sem precisar fornecer uma senha**. Isto é útil em um ambiente seguro onde você controla todas as máquinas, no entanto isto **é um perigo de segurança**. O formato deste arquivo é o seguinte:

```
#Acesso        Máquina              Usuário
   +      maquina2.dominio.com.br   usuario1
   +      maquina3.dominio.com.br   usuario1
   +      maquina4.dominio.com.br   usuario1
   -      maquina2.dominio.com.br   usuario2
   -      maquina4.dominio.com.br   usuario2
```

O primeiro campo especifica se o acesso será permitido ou negado caso o segundo e terceiro campo confiram. Por razões de segurança deve ser especificado o FQDN no caso de nomes de máquinas.

Para aumentar a segurança, não use este mecanismo e encoraje seus usuários a também não usar o arquivo ‘.rhosts’.

ATENÇÃO O uso do sinal "+" sozinho significa permitir acesso livre a qualquer pessoa de qualquer lugar. Se este mecanismo for mesmo necessário, tenha muita atenção na especificação de seus campos.

- Verificando a segurança do TCPD e a sintaxe dos arquivos

O utilitário ‘tcpdchk’ é útil para verificar problemas nos arquivos ‘hosts.allow’ e ‘hosts.deny’. Quando é executado ele verifica a sintaxe destes arquivos e relata problemas, caso eles existam.

Outro utilitário útil é o ‘tcpdmatch’, o que ele faz é permitir que você simule a tentativa de conexões ao seu sistema e observar ser ela será permitida ou bloqueada pelos arquivos ‘hosts.allow’ e ‘hosts.deny’.

É importante mostrar na prática como o ‘tcpdmatch’ funciona através de um exemplo simulando um teste simples em um sistema com a configuração padrão de acesso restrito:

O arquivo ‘hosts.allow’ contém as seguintes linhas:

```
ALL: 127.0.0.1
in.talkd, in.ntalkd: ALL
in.fingerd: 192.168.1. EXCEPT 192.168.1.30
```

A primeira linha permite o loopback (127.0.0.1) acessar qualquer serviço TCP/UDP em nosso computador, a segunda linha permite qualquer um acessar os servidor TALK (nós desejamos que o sistema nos avise quando alguém desejar conversar) e a terceira somente permite enviar dados do ‘finger’ para computadores dentro de nossa rede privada (exceto para 192.168.1.30).

O arquivo ‘hosts.deny’ contém a seguinte linha:

```
ALL: ALL
```

Qualquer outra conexão será explicitamente derrubada.

Vamos aos testes, digitando: "tcpdmatch in.fingerd 127.0.0.1" (verificar se o endereço 127.0.0.1 tem acesso ao finger):

```
client: address 127.0.0.1
server: process in.fingerd
matched: /etc/hosts.allow line 1
access: granted
```

Ok, temos acesso garantido com especificado pela linha 1 do ‘hosts.allow’ (a primeira linha que confere é usada). Agora "tcpdmatch in.fingerd 192.168.1.29":

```
client: address 192.168.1.29
server: process in.fingerd
matched: /etc/hosts.allow line 3
access: granted
```

O acesso foi permitido através da linha 3 do ‘hosts.allow’. Agora "tcpdmatch in.fingerd 192.168.1.29":

```
client: address 192.168.1.30
server: process in.fingerd
matched: /etc/hosts.deny line 1
access: denied
```

O que aconteceu? como a linha 2 do ‘hosts.allow’ permite o acesso a todos os computadores 192.168.1.* exceto 192.168.1.30, ela não bateu, então o processamento partiu para o ‘hosts.deny’ que nega todos os serviços para qualquer endereço. Agora um último exemplo: "tcpdmatch in.talkd www.debian.org"

```
client: address www.debian.org
server: process in.talkd
matched: /etc/hosts.allow line 2
access: granted
```

Ok, na linha 2 qualquer computador pode te chamar para conversar via talk na rede, mas para o endereço DNS conferir com um IP especificado, o ‘GNU/Linux’ faz a resolução DNS, convertendo o endereço para IP e verificando se ele possui acesso.

No lugar do endereço também pode ser usado a forma ‘daemon@computador’ ou ‘cliente@computador’ para verificar respectivamente o acesso de daemons e cliente de determinados computadores aos serviços da rede.

Como pode ver o TCPD ajuda a aumentar a segurança do seu sistema, mas não confie nele além do uso em um sistema simples, é necessário o uso de um firewall verdadeiro para controlar minuciosamente a segurança do seu sistema e dos pacotes que atravessam os protocolos, roteamento e as interfaces de rede. Se este for o caso aprenda a trabalhar a fundo com firewalls e implemente a segurança da sua rede da forma que melhor planejar.

11.8.5 Firewall

Dentre todos os métodos de segurança, o *Firewall* é o mais seguro. A função do Firewall é bloquear determinados tipos de tráfego de um endereço ou para uma porta local ou permitir o acesso de determinados usuários mas bloquear outros, bloquear a falsificação de endereços, redirecionar tráfego da rede, ping da morte, etc.

A implementação de um bom firewall dependerá da experiência, conhecimentos de rede (protocolos, roteamento, interfaces, endereçamento, masquerade, etc), da rede local, e sistema em geral do Administrador de redes, a segurança de sua rede e seus dados dependem da escolha do profissional correto, que entenda a fundo o TCP/IP, roteamento, protocolos, serviços e outros assuntos ligados a rede.

Freqüentemente tem se ouvido falar de empresas que tiveram seus sistemas invadidos, em parte isto é devido a escolha do sistema operacional indevido mas na maioria das vezes o motivo é a falta de investimento da empresa em políticas de segurança, que algumas simplesmente consideram a segurança de seus dados e sigilo interno como uma 'despesa a mais'.

Um bom firewall que recomendo é o 'ipchains', 'Sinus' e o 'TIS'. Particularmente gosto muito de usar o 'ipchains' e o 'Sinus' e é possível fazer coisas inimagináveis programando scripts para interagirem com estes programas...

11.9 Outros arquivos de configuração relacionados com a rede

11.9.1 /etc/services

O arquivo '/etc/services' é um banco de dados simples que associa um nome amigável a humanos a uma porta de serviço amigável a máquinas. É um arquivo texto de formato muito simples, cada linha representa um item no banco de dados. Cada item é dividido em três campos separados por qualquer número de espaços em branco (tab ou espaços). Os campos são:

nome porta/protocolo apelido # comentário

name Uma palavra simples que representa o nome do serviço sendo descrito.

porta/protocolo Este campo é dividido em dois sub-campos.

- 'porta' - Um número que especifica o número da porta em que o serviço estará disponível. Muitos dos serviços comuns tem designados um número de serviço. Estes estão descritos no RFC-1340.
- 'protocolo' - Este sub-campo pode ser ajustado para *tcp* ou *udp*. É importante notar que o item *18/tcp* é muito diferente do item *18/udp* e que não existe razão técnica porque o mesmo serviço precisa existir em ambos. Normalmente o senso comum prevalece e que somente se um serviço esta disponível em ambos os protocolos *tcp* e *udp*, você precisará especificar ambos. apelidos Outros nomes podem ser usados para se referir a entrada deste serviço. comentário Qualquer texto aparecendo em uma linha após um caracter "#" é ignorado e tratado como comentário.

11.9.2 /etc/protocols

O arquivo '/etc/protocols' é um banco de dados que mapeia números de identificação de protocolos novamente em nomes de protocolos. Isto é usado por programadores para permiti-los especificar protocolos por nomes em seus programas e também por alguns programas tal como *tcpdump* permitindo-os mostrar *nomes ao invés de números* em sua saída. A sintaxe geral deste arquivo é:

nomeprotocolo número apelidos

12 Kernel e Módulos

Este capítulo descreve em detalhes o que é o kernel, módulos, sua configuração e programas relacionados.

12.1 O Kernel

É o sistema operacional (o 'GNU/Linux') é ele controla os dispositivos e demais periféricos do sistema (como memória, placas de som, vídeo, discos rígidos, disquetes, sistemas de arquivos, redes e outros recursos disponíveis). Muitos confundem isto e chamam a distribuição de sistema operacional, isto é errado!

O *kernel* faz o controle dos periféricos do sistema e para isto ele deve ter o seu suporte incluído. Para fazer uma placa de som *Sound Blaster* funcionar, por exemplo, é necessário que o kernel ofereça suporte a este placa e você deve configurar seus parâmetros (como interrupção, I/O e DMA) com comandos específicos para ativar a placa e faze-la funcionar corretamente. Existe um documento que contém quais são os periféricos suportados/ não suportados pelo 'GNU/Linux', ele se chama 'Hardware-HOWTO'.

Suas versões são identificadas por números como 2.0.36, 2.0.38, 2.1.10, 2.2.12, as versões que contém um número par entre o primeiro e segundo ponto são versões estáveis e que contém números ímpares neste mesmo local são versões instáveis (em desenvolvimento). Usar versões instáveis não quer dizer que ocorrerá travamentos ou coisas do tipo, mas algumas partes do kernel podem não estar testadas o suficiente ou alguns controladores podem ainda estar incompletos para obter pleno funcionamento. Se opera sua máquina em um ambiente crítico, prefira pegar versões estáveis do kernel.

Após inicializar o sistema, o kernel e seus arquivos podem ser acessados ou modificados através do ponto de montagem '/proc'. Para detalhes veja 'O sistema de Arquivos '/proc".

Caso você tenha um dispositivo (como uma placa de som) que tem suporte no 'GNU/Linux' mas não funciona veja , 'Como adicionar suporte a Hardwares e outros dispositivos no kernel'.

12.2 Módulos

São partes do kernel que são carregadas somente quando são solicitadas por algum aplicativo ou dispositivo e descarregadas da memória quando não são mais usadas. Este recurso é útil por 2 motivos: Evita a construção de um kernel grande (estático) que ocupe grande parte da memória com todos os drivers compilados e permite que partes do kernel ocupem a memória somente quando forem necessários.

Os módulos do kernel estão localizados no diretório '/lib/modules/*versão do kernel*/*' (onde '*versão do kernel*' é a versão atual do kernel em seu sistema, caso seja '2.2.10' o diretório que contém seus módulos será '/lib/modules/2.2.10'.

Os módulos são carregados automaticamente quando solicitados através do programa 'kmod' ou manualmente através do arquivo '/etc/modules' , 'insmod' ou 'modprobe'. Atenção: Não compile o suporte ao seu sistema de arquivos raíz como módulo, isto o tornará inacessível.

12.3 Como adicionar suporte a Hardwares e outros dispositivos no kernel

Quando seu hardware não funciona mas você tem certeza que é suportado pelo 'GNU/Linux', é preciso seguir alguns passos para faze-lo funcionar corretamente:

- Verifique se o kernel atual foi compilado com suporte ao seu dispositivo. Também é possível que o suporte ao dispositivo esteja compilado como módulo. Dê o comando 'dmesg' para ver as mensagens do kernel durante a inicialização e verifique se aparece alguma coisa referente ao dispositivo que deseja instalar (alguma mensagem de erro, etc). Caso não aparecer nada é possível que o driver esteja compilado como módulo, para verificar isto entre no diretório '/lib/modules/*versão do kernel*' e veja se encontra o módulo correspondente ao seu dispositivo (o módulo da placa *NE 2000* tem o nome de 'ne.o' e o da placa *Sound Blaster* de 'sb.o', por exemplo).

Caso o kernel não tiver o suporte ao seu dispositivo, você precisará recompilar seu kernel ativando seu suporte. Veja 'Recompilando o Kernel'.

- Caso seu hardware esteja compilado no kernel, verifique se o módulo correspondente está carregado (com o comando 'lsmod'). Caso não estiver, carregue-o com o 'modprobe' (por exemplo, 'modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpuio=0x330'), para detalhes veja Seção 16.8, 'modprobe'.

O uso deste comando deverá ativar seu hardware imediatamente, neste caso configure o módulo para ser carregado automaticamente através do programa 'modconf' ou edite os arquivos relacionados com os módulos (veja Seção 16.12, 'Arquivos relacionados com o Kernel e Módulos'). Caso não tenha sucesso, será retornada uma mensagem de erro.

12.4 kmod

Este é o programa usado para carregar os módulos automaticamente quando são requeridos pelo sistema. Ele é um daemon que funciona constantemente fazendo a monitoração, quando verifica que algum dispositivo ou programa está solicitando o suporte a algum dispositivo, ele carrega o módulo correspondente.

Ele pode ser desativado através da recompilação do kernel, dando um 'kill' no processo ou através do arquivo '/etc/modules' . Caso seja desativado, é preciso carregar manualmente os módulos através do 'modprobe' ou 'insmod'.

12.5 lsmod

Lista quais módulos estão carregados atualmente pelo kernel. O nome 'lsmod' é uma contração de 'ls'+'módulos' - Listar Módulos. A listagem feita pelo 'lsmod' é uma alternativa ao uso do comando 'cat /proc/modules'.

A saída deste comando tem a seguinte forma:

Module	Size	Pages	Used by
nls_iso8859_1	8000	1	1 (autoclean)
nls_cp437	3744	1	1 (autoclean)
ne	6156	2	1
8390	8390	1	[ne] 0

A coluna *Module* indica o nome do módulo que está carregado, a coluna *Used* mostra qual módulos está usando aquele recurso. O parâmetro *(autoclean)* no final da coluna indica que o módulo foi carregado manualmente (pelo 'insmod' ou 'modprobe') ou através do 'kmod' e será automaticamente removido da memória quando não for mais usado.

No exemplo acima os módulos *ne* e *8390* não tem o parâmetro *(autoclean)* porque foram carregados pelo arquivo '/etc/modules' . Isto significa que não serão removidos da memória caso estiverem sem uso.

Qualquer módulo carregado pode ser removido manualmente através do comandos 'rmmod'.

12.6 insmod

Carrega um módulo manualmente. Para carregar módulos que dependem de outros módulos para que funcionem, você duas opções: Carregar os módulos manualmente ou usar o 'modprobe' que verifica e carrega as dependências correspondentes.

A sintaxe do comando é: 'insmod [*módulo*] [*opções módulo*]'

Onde:

módulo É o nome do módulo que será carregado.

opções módulo Opções que serão usadas pelo módulo. Variam de módulo para módulo, alguns precisam de opções outros não, tente primeiro carregar sem opções, caso seja mostrada uma mensagem de erro verifique as opções usadas por ele. Para detalhes sobre que opções são suportadas por cada módulo, veja a sua documentação no código fonte do kernel em '/usr/src/linux/Documentat

Exemplo: 'insmod ne io=0x300 irq=10'

12.7 rmmod

Remove módulos carregados no kernel. Para ver os nomes dos módulos atualmente carregados no kernel digite 'lsmod' e verifique na primeira coluna o nome do módulo. Caso um módulo tenha dependências e você tentar remover suas dependências, uma mensagem de erro será mostrada alertando que o módulo está em uso.

Exemplo: 'rmmod ne'

12.8 modprobe

Carrega um módulo e suas dependências manualmente. Este comando permite carregar diversos módulos e dependências de uma só vez. O comportamento do 'modprobe' é modificado pelo arquivo '/etc/modules.conf' .

A sintaxe deste comando é: 'modprobe [*módulo*] [*opções módulo*]'

Onde:

módulo É o nome do módulo que será carregado.

opções módulo Opções que serão usadas pelo módulo. Variam de módulo para módulo, alguns precisam de opções outros não, tente primeiro carregar sem opções, caso seja mostrada uma mensagem de erro verifique as opções usadas por ele. Para detalhes sobre que opções são suportadas por cada módulo, veja a sua documentação no código fonte do kernel em '/usr/src/linux/Documentat

Nem todos os módulos são carregados corretamente pelo 'modprobe', o 'plip', por exemplo, mostra uma mensagem sobre porta I/O inválida mas não caso seja carregado pelo 'insmod'.

Exemplo: 'modprobe ne io=0x300 irq=10', 'modprobe sb io=0x220 irq=5 dma=1 dma16=5 mpuio=0x330'

12.9 depmod

Verifica a dependência de módulos. As dependências dos módulos são verificadas pelos scripts em '/etc/init.d' usando o comando 'depmod -a' e o resultado gravado no arquivo '/lib/modules/versão_do_kernel/modules.dep'. Esta checagem serve para que todas as dependências de módulos estejam corretamente disponíveis na inicialização do sistema. O comportamento do 'depmod' pode ser modificado através do arquivo '/etc/modules.conf' .

Exemplo: 'depmod -a'

12.10 modconf

Este programa permite um meio mais fácil de configurar a ativação de módulos e opções através de uma interface através de menus. Selecione a categoria de módulos através das setas acima e abaixo e pressione enter para selecionar os módulos existentes. Serão pedidas as opções do módulo (como DMA, IRQ, I/O) para que sua inicialização seja possível, estes parâmetros são específicos de cada módulo e devem ser vistos na documentação do código fonte do kernel no diretório '/usr/src/linux/Documentation'. Note que também existem módulos com auto-detecção mas isto deixa o sistema um pouco mais lento, porque ele fará uma varredura na faixa de endereços especificados pelo módulo para achar o dispositivo. As opções são desnecessárias em alguns tipos de módulos.

As modificações feitas por este programa são gravadas no diretório '/etc/modutils' em arquivos separados como '/etc/modutils/alias' - alias de módulos, '/etc/modutils/modconf' - opções usadas por módulos, '/etc/modutils/paths' - Caminho onde os módulos do sistema são encontrados. Dentro de '/etc/modutils' é ainda encontrado um sub-diretório chamado 'arch' que contém opções específicas por arquiteturas.

A sincronização dos arquivos gerados pelo ‘modconf’ com o ‘/etc/modules.conf’ é feita através do utilitário ‘update-modules’. Ele é normalmente executado após modificações nos módulos feitas pelo ‘modconf’.

12.11 Recompilando o Kernel

Será que vou precisar recompilar o meu kernel? você deve estar se perguntando agora. Abaixo alguns motivos para esclarecer suas dúvidas:

- Melhora o desempenho do kernel. O kernel padrão que acompanha as distribuições ‘GNU/Linux’ foi feito para funcionar em qualquer tipo de sistema e garantir seu funcionamento e inclui suporte a praticamente tudo. Isto pode gerar desde instabilidade até uma grade pausa do kernel na inicialização quando estiver procurando pelos dispositivos que simplesmente não existem em seu computador!

A compilação permite escolher somente o suporte aos dispositivos existentes em seu computador e assim diminuir o tamanho do kernel, desocupar a memória RAM com dispositivos que nunca usará e assim você terá um desempenho bem melhor do que teria com um kernel pesado.

- Incluir suporte a alguns hardwares que estão desativados no kernel padrão (SMP, APM, Firewall, drivers experimentais, etc).
- Se aventurar em compilar um kernel (sistema operacional) personalizado em seu sistema.
- Impressionar os seus amigos, tentando coisas novas.

Serão necessários uns 70Mb de espaço em disco disponível para copiar e descompactar o código fonte do kernel e alguns pacotes de desenvolvimento como o ‘gcc’, ‘cpp’, ‘binutils’, ‘gcc-i386-gnu’, ‘bin86’, ‘make’, ‘dpkg-dev’, ‘perl’, ‘kernel-package’ (os três últimos somente para a distribuição ‘Debian’).

Para recompilar o kernel usando o método padrão, siga os seguintes passos:

1. Descompacte o código fonte do kernel (através do arquivo linux-2.2.XX) para o diretório ‘/usr/src’. O código fonte do kernel pode ser encontrado em ftp://ftp.kernel.org.
2. Após isto, entre no diretório onde o código fonte do kernel foi instalado com ‘cd /usr/src/linux’ (este será assumido o lugar onde o código fonte do kernel se encontra).
3. Como usuário ‘root’, digite ‘make config’. Você também pode usar ‘make menuconfig’ (configuração através de menus) ou ‘make xconfig’ (configuração em modo gráfico) mas precisará de pacotes adicionais.
   Serão feitas perguntas sobre se deseja suporte a tal dispositivo, etc. Pressione ‘Y’ para incluir o suporte diretamente no kernel, ‘M’ para incluir o suporte como módulo ou ‘N’ para não incluir o suporte. Note que nem todos os drivers podem ser compilados como módulos. Escolha as opções que se encaixam em seu sistema. se estiver em dúvida sobre a pergunta digite ‘?’ e tecle Enter para ter uma explicação sobre o que aquela opção faz. Se não souber do que se trata, recomendo incluir a opção (pressionando ‘Y’ ou ‘M’. Este passo pode levar entre 5 minutos e 1 Hora (usuários que estão fazendo isto pela primeira vez tendem a levar mais tempo lendo e conhecendo os recursos que o ‘GNU/Linux’ possui antes de tomar qualquer decisão). Não se preocupe se esquecer de incluir o suporte a alguma coisa, você pode repetir o passo ‘make config’, recompilar o kernel e instalar em cima do antigo a qualquer hora que quiser.
4. Digite o comando ‘make dep’ para verificar as dependências dos módulos.
5. Digite o comando ‘make clean’ para limpar construções anteriores do kernel.
6. Digite o comando ‘make zImage’ para iniciar a compilação do kernel estático (outro comando compila os módulos). Aguarde a compilação, o tempo pode variar dependendo da quantidade de recursos que adicionou ao kernel, a velocidade de seu computador e a quantidade de memória RAM disponível. Caso tenha acrescentado muitos itens no Kernel, é possível que o comando ‘make zImage’ falhe no final (especialmente se o tamanho do kernel estático for maior que 505Kb). Neste caso use ‘make bzImage’. A diferença entre *zImage* e *bzImage* é que o primeiro possui um limite de tamanho porque é descompactado na memória básica (recomendado para alguns Notebooks), já a *bzImage*, é descompactada na memória estendida e não possui as limitações da *zImage*. Existe a opção bzlilo, que compila o kernel e o coloca no diretório raiz ’/’ .
7. Após terminada a compilação do kernel estático, execute ‘make modules’ para compilar os módulos referentes àquele kernel. A velocidade de compilação pode variar de acordo com os motivos do passo anterior.
8. A compilação neste ponto está completa, você agora tem duas opções para instalar o kernel: Substituir o kernel anterior pelo recém compilado ou usar os dois. A segunda questão é recomendável se você não tem certeza se o kernel funcionará corretamente e deseja iniciar pelo antigo no caso de alguma coisa dar errado.

Se você optar por substituir o kernel anterior:

1. É recomendável renomear o diretório `/lib/modules/versão_do_kernel' para `/lib/modules/versão do kernel.old, isto será útil para a restauração completa dos módulos antigos caso alguma coisa der errado.

2. Execute o comando `make modules_install' para instalar os módulos do kernel recém compilado em `/lib/modules/versão do kernel'.

3. Copie o arquivo `zImage' que contém o kernel de `/usr/src/linux/arch/i386/boot/zImage' para `/boot/vmlinuz-2.XX.XX' (2.XX.XX é a versão do kernel anterior)

4. Veri que se o link simbólico `/vmlinuz' aponta para a versão do kernel que compilou atualmente (com `ls -la /'). Caso contrário, apague o arquivo `/vmlinuz' do diretório raíz e crie um novo link com `ln -s /boot/vmlinuz-2.XX.Xx /vmlinuz' apontando para o kernel correto.[2]

5. Execute o comando `lilo' para gerar um novo setor de partida no disco rígido.

6. Reinicie o sistema (`shutdown -r now').

Caso tudo esteja funcionando normalmente, apague o diretório antigo de módulos que salvou e o kernel antigo de `/boot'. Caso algo tenha dado errado e seu sistema não inicializa, inicie a partir de um disquete, apague o novo kernel, apague os novos módulos, renomeie o diretório de módulos antigos para o nome original, ajuste o link simbólico `/vmlinuz' para apontar para o antigo kernel e execute o `lilo'. Após reiniciar seu computador voltará como estava antes.

Se você optar por manter o kernel anterior e selecionar qual será usado na partida do sistema (útil para um kernel em testes):

1. Execute o comando `make modules_install' para instalar os módulos recém compilados do kernel em `/lib/modules/versão do kernel'.

2. Copie o arquivo `zImage' que contém o kernel de `/usr/src/linux/arch/i386/boot/zImage' para `/boot/vmlinuz-2.XX.XX' (2.XX.XX é a versão do kernel anterior)

3. Crie um link simbólico no diretório raíz (`/') apontando para o novo kernel. Como exemplos será usado `/vmlinuz-novo'.

4. Modi que o arquivo `/etc/lilo.conf' para incluir a nova imagem de kernel.
   Por exemplo:
   Antes da modi cação:
   boot=/dev/hda
   prompttimeout=200
   delay=200
   map=/boot/map
   install=/boot/boot.b
   image= /vmlinuz
   root = /dev/hda1
   label = 1
   read-only
   Depois da modi cação:
   boot=/dev/hda
   prompttimeout=200
   delay=200
   map=/boot/map
   install=/boot/boot.b
   image= /vmlinuz
   root = /dev/hda1
   label = 1
   read-only
   image= /vmlinuz-novo
   root = /dev/hda1
   label = novo
   read-only

13 Arquivos e daemons de Log

A maioria das atividades dos programas são registradas em arquivos localizados em '/var/log' . Estes arquivos de registros são chamados de *logs* e contém a data, hora e a mensagem emitida pelo programa (violações do sistema, mensagens de error, alerta e outros eventos) entre outros campos. Enfim, muitos detalhes úteis ao administrador tanto para acompanhar o funcionamento do seu sistema, comportamento dos programas ou ajudar na solução e prevenção de problemas.

Alguns programas como o 'Apache', 'exim', 'ircd' e 'squid' criam diversos arquivos de log e por este motivo estes são organizados em sub-diretórios (a mesma técnica é usada nos arquivos de configuração em '/etc', conforme a padrão FHS atual).

13.1 Daemons de log do sistema

Os daemons de log do sistema registram as mensagens de saída do kernel ('klogd') e sistema ('syslogd') nos arquivos em '/var/log' .

A classificação de qual arquivo em '/var/log' receberá qual tipo de mensagem é controlado pelo arquivo de configuração '/etc/syslog.conf' através de *facilidades e níveis*

13.1.1 syslogd

Este daemon controla o registro de logs do sistema.

'syslogd [*opções*]'

opções

-f Especifica um arquivo de configuração alternativo ao '/etc/syslog.conf'.

-h Permite o redirecionar mensagens recebidas por outros daemons a servidores porta UDP 514 de outros servidores 'syslog' definidos.

-I [computadores] Especifica um ou mais computadores (separados por ":") que deverão ser registrados somente com o hostname ao invés do FQDN (nome completo, incluindo domínio).

-m [minutos] Intervalo em _minutos_ que o syslog mostrará a mensagem '–MARK–'. O valor padrão padrão é 20 minutos, 0 desativa.

-n Evita que o processo caia automaticamente em background. Necessário principalmente se o 'syslogd' for controlado pelo 'init'.

-p [soquete] Especifica um soquete UNIX alternativo ao invés de usar o padrão '/dev/log'.

-r Permite o recebimento de mensagens através da rede através da porta UDP 514. Esta opção é útil para criar um servidor de logs centralizado na rede. Por padrão, o servidor 'syslog' rejeitará conexões externas.

-s [domínios] Especifica a lista de domínios (separados por ":") que deverão ser retirados antes de enviados ao log.

13.1.2 Arquivo de configuração 'syslog.conf'

O arquivo de configuração '/etc/syslog.conf' possui o seguinte formato:

facilidade.nível destino

A *facilidade e nível* são separadas por um "." e contém parâmetros que definem o que será registrado nos arquivos de log do sistema:

* 'facilidade' - É usada para especificar que tipo de programa está enviando a mensagem. Os seguintes níveis são permitidos (em ordem alfabética):

- 'auth' - Mensagens de segurança/autorização (é recomendável usar authpriv ao invés deste).
- 'authpriv' - Mensagens de segurança/autorização (privativas).
- 'cron' - Daemons de agendamento ('cron' e 'at').
- 'daemon' - Outros daemons do sistema que não possuem facilidades específicas.
- 'ftp' - Daemon de ftp do sistema.
- 'kern' - Mensagens do kernel.
- 'lpr' - Subsistema de impressão.

- ‘local0 a local7’ - Reservados para uso local.
- ‘mail’ - Subsistema de e-mail.
- ‘news’ - Subsistema de notícias da USENET.
- ‘security’ - Sinônimo para a facilidade ‘auth’ (evite usa-la).
- ‘syslog’ - Mensagens internas geradas pelo ‘syslogd’.
- ‘user’ - Mensagens genéricas de nível do usuário.
- ‘uucp’ - Subsistema de UUCP.
- ‘*’ - Confere com todas as facilidades. Mais de uma facilidade pode ser especificada na mesma linha do ‘syslog.conf’ separando-as com ",".
- ‘nível’ - Especifica a importância da mensagem. Os seguintes níveis são permitidos (em ordem de importância invertida; da mais para a menos importante):

  ‘emerg’ - O sistema está inutilizável.

  ‘alert’ - Uma ação deve ser tomada imediatamente para resolver o problema.

  ‘crit’ - Condições críticas.

  ‘err’ - Condições de erro.

  ‘warning’ - Condições de alerta.

  ‘notice’ - Condição normal, mas significante.

  ‘info’ - Mensagens informativas.

  ‘debug’ - Mensagens de depuração.

  ‘*’ - Confere com todos os níveis.

  ‘none’ - Nenhuma prioridade.

Além destes níveis os seguintes sinônimos estão disponíveis:

- ‘error’ - Sinônimo para o nível err.
- ‘panic’ - Sinônimo para o nível emerg.
- ‘warn’ - Sinônimo para o nível warning.
- ‘destino’ - O destino das mensagens pode ser um arquivo, um pipe (se iniciado por um "|"), um computador remoto (se iniciado por uma "@"), determinados usuários do sistema (especificando os logins separados por vírgula) ou para todos os usuários logados via ‘wall’ (usando "*").

Todas as mensagens com o nível especificado e superiores a esta especificadas no ‘syslog.conf’ serão registradas, de acordo com as opções usadas. Conjuntos de *facilidades e níveis* podem ser agrupadas separando-as por ";".

OBS1: Sempre use TABS ao invés de espaços para separar os parâmetros do ‘syslog.conf’.

OBS2: Algumas facilidades como ‘security’, emitem um beep de alerta no sistema e enviam uma mensagem para o console, como forma de alerta ao administrador e usuários logados no sistema.

Existem ainda 4 caracteres que garantes funções especiais: "*", "=", "!" e "-":

- "*" - Todas as mensagens da *facilidade* especificada serão redirecionadas.
- "=" - Somente o *nível* especificado será registrado.
- "!" - Todos os *níveis* especificados e maiores NÃO serão registrados.
- "-" - Pode ser usado para desativar o sync imediato do arquivo após sua gravação. Os caracteres especiais "=" e "!" podem ser combinados em uma mesma regra.

Exemplo: Veja abaixo um exemplo de um arquivo ‘/etc/syslog.conf’ .

```
# # Primeiro alguns arquivos de log padrões. Registrados por facilidade #
auth,authpriv.* /var/log/auth.log
*.*;auth,authpriv.none -/var/log/syslog
cron.* /var/log/cron.log
daemon.* -/var/log/daemon.log
kern.* -/var/log/kern.log
lpr.* -/var/log/lpr.log
mail.* /var/log/mail.log
user.* -/var/log/user.log
uucp.* -/var/log/uucp.log
#
# Registro de logs do sistema de mensagens. Divididos para facilitar
# a criação de scripts para manipular estes arquivos.
# mail.info -/var/log/mail.info
mail.warn -/var/log/mail.warn
mail.err /var/log/mail.err
# Registro para o sistema de news INN #
news.crit /var/log/news/news.crit
news.err /var/log/news/news.err
news.notice -/var/log/news/news.notice
#
# Alguns arquivos de registro "pega-tudo".
# São usadas "," para especificar mais de uma prioridade (por # exemplo, "auth,authpriv.none") e ";" para especificar
mais de uma
# facilidade.nível que será gravada naquele arquivo.
# Isto permite deixar as regras consideravelmente menores e mais legíveis #
*.=debug;\
auth,authpriv.none;\
news.none;mail.none -/var/log/debug *.=info;*.=notice;*.=warn;\
auth,authpriv.none;\
cron,daemon.none;\
mail,news.none -/var/log/messages
#
# Emergências são enviadas para qualquer um que estiver logado no sistema. Isto
# é feito através da especificação do "*" como destino das mensagens e são
# enviadas através do comando wall. # *.emerg *
#
# Eu gosto de ter mensagens mostradas no console, mas somente em consoles que
# não utilizo.
#
#daemon,mail.*;\
# news.=crit;news.=err;news.=notice;\
# *.=debug;*.=info;\
# *.=notice;*.=warn /dev/tty8
# O pipe /dev/xconsole é usado pelo utilitário "xconsole". Para usa-lo,
# você deve executar o "xconsole" com a opção "-file":
#
# $ xconsole -file /dev/xconsole [...]
# # NOTA: ajuste as regras abaixo, ou ficará maluco se tiver um um site
# muito movimentado...
# daemon.*;mail.*;\
news.crit;news.err;news.notice;\
*.=debug;*.=info;\
*.=notice;*.=warn |/dev/xconsole
# A linha baixo envia mensagens importantes para o console em que
# estamos trabalhando logados (principalmente para quem gosta de ter
# controle total sobre o que está acontecendo com seu sistema).
m *.err;kern.debug;auth.notice;mail.crit /dev/console
```

13.1.3 klogd

Este daemon controla o registro de mensagens do kernel. Ele monitora as mensagens do kernel e as envia para o daemon de monitoramento 'syslogd', por padrão.

'klogd [*opções*]'

opções

-d Ativa o modo de depuração do daemon

-f [*arquivo*] Envia as mensagens do kernel para o arquivo especificado ao invés de enviar ao daemon do 'syslog'

-i Envia um sinal para o daemon recarregar os símbolos de módulos do kernel.

-I Envia um sinal para o daemon recarregar os símbolos estáticos e de módulos do kernel.

-n Evita a operação em segundo plano. Útil se iniciado pelo 'init' -k [*arquivo*] Especifica o arquivo que contém os símbolos do kernel. Exemplos deste arquivo estão localizados em '/boot/System.map-xx.xx.xx'. A especificação de um arquivo com a opção '-k' é necessária se desejar que sejam mostradas a tabela de símbolos ao invés de endereços numéricos do kernel.

13.2 logger

Este comando permite enviar uma mensagem nos log do sistema. A mensagem é enviada aos logs via daemon 'syslogd' ou via soquete do sistema, é possível especificar a prioridade, nível, um nome identificando o processo, etc. Seu uso é muito útil em shell scripts ou em outros eventos do sistema.

'logger [*opções*] [*mensagem*]'

Onde:

mensagem Mensagem que será enviada ao daemon *syslog*

opções

-i Registra o PID do processo

-s Envia a mensagem ambos para a saída padrão (STDOUT) e syslog.

-f [*arquivo*] Envia o conteúdo do arquivo especificado como *mensagem* ao syslog.

-t [*nome*] Especifica o nome do processo responsável pelo log que será exibido antes do PID na mensagem do syslog.

-p [*prioridade*] Especifica a prioridade da mensagem do syslog, especificada como 'facilidade.nível'. O valor padrão *prioridade.nível* é *user.notice*

Exemplos:

'logger -i -t focalinux Teste teste teste'
'logger -i -f /tmp/mensagem -p security.emerg'

14 Compactadores

Esta seção explica o que são e como usar programas compactadores no 'GNU/Linux', as características de cada um, como identificar um arquivo compactado e como descompactar um arquivo compactado usando o programa correspondente.

A utilização de arquivos compactados é método útil principalmente para reduzir o consumo de espaço em disco ou permitir grandes quantidades de texto serem transferidas para outro computador através de disquetes.

14.1 O que fazem os compactadores/descompactadores?

Compactadores são programas que diminuem o tamanho de um arquivo (ou arquivos) através da substituição de caracteres repetidos. Para entender melhor como eles funcionam, veja o próximo exemplo:

> compactadores compactam e deixam arquivos compactados.
> **– após a compactação da frase –**
> %dores %m e deixam arquivos %dos

O que aconteceu realmente foi que a palavra 'compacta' se encontrava 3 vezes na frase acima, e foi substituída por um sinal de '%'. Para descompactar o processo seria o contrário: Ele substituiria % por 'compacta' e nós temos a frase novamente restaurada.

Você deve ter notado que o tamanho da frase 'compactada' caiu quase pela metade. A quantidade de compactação de um arquivo é chamada de *taxa de compactação*. Assim se o tamanho do arquivo for diminuído a metade após a compactação, dizemos que conseguiu uma *taxa de compactação* de 2:1 (lê-se dois para um), se o arquivo diminuiu 4 vezes, dizemos que conseguiu uma compactação de 4:1 (quatro para um) e assim por diante.

Para controle dos caracteres que são usados nas substituições, os programas de compactação mantém cabeçalhos com todas as substituições usadas durante a compactação. O tamanho do cabeçalho pode ser fixo ou definido pelo usuário, depende do programa usado na compactação.

Este é um exemplo bem simples para entender o que acontece durante a compactação, os programas de compactação executam instruções muito avançadas e códigos complexos para atingir um alta taxa de compactação.

Observações:

- Não é possível trabalhar diretamente com arquivos compactados! É necessário descompactar o arquivo para usa-lo. Note que alguns programas atualmente suportam a abertura de arquivos compactados, mas na realidade eles apenas simplificam a tarefa descompactando o arquivo, abrindo e o recompactando assim que o trabalho estiver concluído.
- Arquivos de texto tem uma taxa de compactação muito melhor que arquivos binários, porque possuem mais caracteres repetidos. É normal atingir taxas de compactação de 10 para 1 ou mais quando se compacta um arquivo texto. Arquivos binários, como programas, possuem uma taxa de compactação média de 2:1.
- Note que também existem programas compactadores especialmente desenvolvidos para compactação de músicas, arquivos binários, textos.

14.2 Extensões de arquivos compactados

As extensões identificam o tipo de um arquivo e assim o programa o programa necessário para trabalhar com aquele tipo de arquivo. Existem dezenas de extensões que identificam arquivos compactados. Quando um arquivo (ou arquivos) é compactado, uma extensão correspondente ao programa usado é adicionada ao nome do arquivo (caso o arquivo seja compactado pelo 'gzip' receberá a extensão '.gz', por exemplo). Ao descompactar acontece o contrário: a extensão é retirada do arquivo. Abaixo segue uma listagem de extensões mais usadas e os programas correspondentes:

- '.gz' - Arquivo compactado pelo 'gzip'. Use o programa 'gzip' para descompacta-lo. '.bz2' - Arquivo compactado pelo 'bzip2'. Use o programa 'bzip2' para descompacta-lo .
- '.Z' - Arquivo compactado pelo programa 'compress'. Use o programa 'uncompress' para descompacta-lo.
- '.zip' - Arquivo compactado pelo programa 'zip'. Use o programa 'unzip' para descompacta-lo.
- '.rar' - Arquivo compactado pelo programa 'rar'. Use o programa 'rar' para descompacta-lo.
- '.tar.gz' - Arquivo compactado pelo programa 'gzip' no utilitário de arquivamento 'tar'. Para descompacta-lo, você pode usar o 'gzip' e depois o 'tar' ou somente o programa 'tar' usando a opção '-z'.
- '.tgz' - Abreviação de '.tar.gz'.
- '.tar.bz2' - Arquivo compactado pelo programa 'bzip2' no utilitário de arquivamento 'tar'. Para descompacta-lo, você pode usar o 'bzip2' e depois o 'tar' ou somente o programa 'tar' usando a opção '-j'.
- '.tar.Z' - Arquivo compactado pelo programa 'compress' no utilitário de arquivamento 'tar'. Para descompacta-lo, você pode usar o 'uncompress' e depois o 'tar' ou somente o programa 'tar' usando a opção '-Z'.

14.3 gzip

É praticamente o compactador padrão do 'GNU/Linux', possui uma ótima taxa de compactação e velocidade. A extensão dos arquivos compactados pelo 'gzip' é a '.gz', na versão para 'DOS', 'Windows NT' é usada a extensão '.z'.

'gzip [*opções*] [*arquivos*]'

Onde:

arquivos Especifica quais arquivos serão compactados pelo 'gzip'. Caso seja usado um '-', será assumido a entrada padrão. Coringas podem ser usados para especificar vários arquivos de uma só vez .

Opções

-d, –decompress [arquivo] Descompacta um arquivo.

-f Força a compactação, compactando até mesmo links.

-l [arquivo] Lista o conteúdo de um arquivo compactado pelo 'gzip'.

-r Compacta diretórios e sub-diretórios.

-c [arquivo] Descompacta o arquivo para a saída padrão.

-t [arquivo] Testa o arquivo compactado pelo 'gzip'.

-[num], –fast, –best Ajustam a taxa de compactação/velocidade da compactação. Quanto melhor a taxa menor é a velocidade de compactação e vice versa. A opção '–fast' permite uma compactação rápida e tamanho do arquivo maior. A opção '–best' permite uma melhor compactação e uma velocidade menor. O uso da opção '-[número]' permite especificar uma compactação individualmente usando números entre 1 (menor compactação) e 9 (melhor compactação). É útil para buscar um bom equilibro entre taxa de compactação/velocidade (especialmente em computadores muito lentos). Quando um arquivo é compactado pelo 'gzip', é automaticamente acrescentada a extensão '.gz' ao seu nome.

O 'gzip' também reconhece arquivos compactados pelos programas 'zip', 'compress', 'compress -H' e 'pack'. As permissões de acesso dos arquivos são também armazenadas no arquivo compactado.

Exemplos:

- 'gzip -9 texto.txt' - Compacta o arquivo 'texto.txt' usando a compactação máxima (compare o tamanho do arquivo compactado usando o comando 'ls -la').
- 'gzip -d texto.txt.gz' - Descompacta o arquivo 'texto.txt'
- 'gzip -c texto.txt.gz' - Descompacta o arquivo 'texto.txt' para a tela
- 'gzip -9 *.txt' - Compacta todos os arquivos que terminam com '.txt'
- 'gzip -t texto.txt.gz' - Verifica o arquivo 'texto.txt.gz'.

14.4 zip

Utilitário de compactação compatível com 'pkzip' (do 'DOS') e trabalha com arquivos de extensão '.zip'. Possui uma ótima taxa de compactação e velocidade no processamento dos arquivos compactados (comparando-se ao 'gzip').

'zip [*opções*] [*arquivo-destino*] [*arquivos-origem*]'

Onde:

arquivo-destino Nome do arquivo compactado que será gerado.

arquivos-origem Arquivos/Diretórios que serão compactados. Podem ser usados coringas para especificar mais de um arquivo de uma só vez.

opções

-r Compacta arquivos e sub-diretórios.

-e Permite encriptar o conteúdo de um arquivo '.zip' através de senha. A senha será pedida no momento da compactação.

-f Somente substitui um arquivo compactado existente dentro do arquivo '.zip' somente se a versão é mais nova que a atual. Não acrescenta arquivos ao arquivo compactado. Deve ser executado no mesmo diretório onde o programa 'zip' foi executado anteriormente.

-F Repara um arquivo '.zip' danificado.

-[NUM] Ajusta a qualidade/velocidade da compactação. Pode ser especificado um número de 1 a 9. O 1 permite mínima compactação e máxima velocidade, 9 permite uma melhor compactação e menor velocidade.

-i [arquivos] Compacta somente os [arquivos] especificados.

-j Se especificado, não armazena caminhos de diretórios.

-m Apaga os arquivos originais após a compactação.

-T [arquivo] Procura por erros em um arquivo '.zip'. Caso sejam detectados problemas, utilize a opção '-F' para corrigi-los.

-y Armazena links simbólicos no arquivo '.zip'. Por padrão, os links simbólicos são ignorados durante a compactação.

-k [arquivo] Modifica o [arquivo] para ter compatibilidade total com o 'pkzip' do 'DOS'.

-l Converte saltos de linha 'UNIX' (LF) para o formato CR+LF (usados pelo 'DOS'). Use esta opção com arquivos Texto.

-ll Converte saltos de linha 'DOS' (CR+LF) para o formato 'UNIX' (LF). Use esta opção com arquivos texto.

-n [extensão] Não compacta arquivos identificados por [extensão]. Ele é armazenado sem compactação no arquivo '.zip', muito útil para uso com arquivos já compactados. Caso sejam especificados diversas extensões de arquivos, elas devem ser separadas por ':' - Por exemplo, 'zip -n .zip:.tgz arquivo.zip *.txt'.

-q Não mostra mensagens durante a compactação do arquivo.

-u Atualiza/adiciona arquivos ao arquivo '.zip'

-X Não armazena detalhes de permissões, UID, GID e datas dos arquivos.

-z Permite incluir um comentário no arquivo '.zip'. Caso o nome de arquivo de destino não termine com '.zip', esta extensão será automaticamente adicionada. Para a descompactação de arquivos '.zip' no 'GNU/Linux', é necessário o uso do utilitário 'unzip'.

Exemplos:

- 'zip textos.zip *.txt' - Compacta todos os arquivos com a extensão '.txt' para o arquivo 'textos.zip' (compare o tamanho do arquivo compactado digitando 'ls -la').
- 'zip -r textos.zip /usr/*.txt' - Compacta todos os arquivos com a extensão '.txt' do diretório '/usr' e sub-diretórios para o arquivo 'textos.zip'.
- 'zip -9 textos.zip *' - Compacta todos os arquivos do diretório atual usando a compactação máxima para o arquivo 'textos.zip'.
- 'zip -T textos.zip' - Verifica se o arquivo 'textos.zip' contém erros.

14.5 unzip

Descompacta arquivos '.zip' criados com o programa 'zip'. Este programa também é compatível com arquivos compactados pelo 'pkzip' do 'DOS'.

'unzip [*opções*] [*arquivo.zip*] [*arquivos-extrair*] [*-d diretório*]'

Onde:

arquivo.zip Nome do arquivo que deseja descompactar. Podem ser usados coringas para especificar mais de um arquivo para ser descompactado.

arquivos-extrair Nome dos arquivos (separados por espaço) que serão descompactados do arquivo '.zip'. Caso não seja especificado, é assumido '*' (todos os arquivos serão descompactados). Se for usado '-x arquivos', os arquivos especificados não serão descompactados. O uso de coringas é permitido.

-d diretório Diretório onde os arquivos serão descompactados. Caso não for especificado, os arquivos serão descompactados no diretório atual.

opções

-c Descompacta os arquivos para stdout (saída padrão) ao invés de criar arquivos. Os nomes dos arquivos também são mostrados (veja a opção '-p').

-f Descompacta somente arquivos que existam no disco e mais novos que os atuais.

-l Lista os arquivos existentes dentro do arquivo '.zip'.

-M Efetua uma pausa a cada tela de dados durante o processamento (a mesma função do comando 'more').

-n Nunca substitui arquivos já existentes. Se um arquivo existe ele é pulado.

-o Substitui arquivos existentes sem perguntar. Tem a função contrária a opção '-n'.

-P [SENHA] Permite descompactar arquivos '.zip' usando a [SENHA]. **CUIDADO!** qualquer usuário conectado em seu sistema pode ver a senha digitada na linha de comando digitada.

-p Descompacta os arquivos para stdout (saída padrão) ao invés de criar arquivos. Os nomes dos arquivos não são mostrados (veja a opção '-c').

-q Não mostra mensagens.

-t Verifica o arquivo '.zip' em busca de erros.

-u Idêntico a opção '-f' só que também cria arquivos que não existem no diretório.

-v Mostra mais detalhes sobre o processamento do 'unzip'.

-z Mostra somente o comentário existente no arquivo. Por padrão o 'unzip' também descompacta sub-diretórios caso o arquivo '.zip' tenha sido gerado com 'zip -r'.

Exemplos:

- 'unzip texto.zip' - Descompacta o conteúdo do arquivo 'texto.zip' no diretório atual.
- 'unzip texto.zip carta.txt' - Descompacta somente o arquivo 'carta.txt' do arquivo 'texto.zip'.
- 'unzip texto.zip -d /tmp/texto' - Descompacta o conteúdo do arquivo 'texto.zip' para o diretório '/tmp/texto'.
- 'unzip -l texto.zip' - Lista o conteúdo do arquivo 'texto.zip'.
- 'unzip -t texto.zip' - Verifica o arquivo 'texto.zip'.

14.6 tar

Na verdade o 'tar' não é um compactador e sim um "arquivador" (ele junta vários arquivos em um só), mas pode ser usado em conjunto com um compactar (como o 'gzip' ou 'zip') para armazena-los compactados. O 'tar' também é muito usado para cópias de arquivos especiais ou dispositivos do sistema. É comum encontrar arquivos com a extensão '.tar', '.tar.gz', '.tgz', '.tar.bz2', '.tar.Z', '.tgZ', o primeiro é um arquivo normal gerado pelo 'tar' e todos os outros são arquivos gerados através 'tar' junto com um programa de compactação ('gzip' ('.gz'), 'bzip2' ('.bz2') e 'compress' ('.Z').

'tar [*opções*] [*arquivo-destino*] [*arquivos-origem*]'

Onde:

arquivo-destino É o nome do arquivo de destino. Normalmente especificado com a extensão '.tar' caso seja usado somente o arquivamento ou '.tar.gz'/'.tgz' caso seja usada a compactação (usando a opção '-z').

arquivos-origem Especifica quais arquivos/diretórios serão compactados.

opções

-c, –create Cria um novo arquivo '.tar'

-t, –list Lista o conteúdo de um arquivo ‘.tar’

-u, –update Atualiza arquivos compactados no arquivo ‘.tar’

-f, –file [HOST:]F Usa o arquivo especificado para gravação ou o dispositivo ‘/dev/rmt0’.

-j, –bzip2 Usa o programa ‘bzip2’ para processar os arquivos do ‘tar’

-l, –one-file-system Não processa arquivos em um sistema de arquivos diferentes de onde o ‘tar’ foi executado.

-M, –multi-volume Cria/lista/descompacta arquivos em múltiplos volumes. O uso de arquivos em múltiplos volumes permite que uma grande cópia de arquivos que não cabe em um disquete, por exemplo, seja feita em mais de um disquete.

-o Grava o arquivo no formato VT7 ao invés do ANSI.

-O, –to-stdout Descompacta arquivos para a saída padrão ao invés de gravar em um arquivo.

–remove-files Apaga os arquivos de origem após serem processados pelo ‘tar’.

-R, –record-number Mostra o número de registros dentro de um arquivo ‘tar’ em cada mensagem.

–totals Mostra o total de bytes gravados com a opção ‘–create’.

-v Mostra os nomes dos arquivos enquanto são processados.

-V [NOME] Inclui um [NOME] no arquivo ‘tar’.

-W, –verify Tenta verificar o arquivo gerado pelo ‘tar’ após grava-lo.

x Extrai arquivos gerados pelo ‘tar’ -X [ARQUIVO] Tenta apagar o [ARQUIVO] dentro de um arquivo compactado ‘.tar’.

-Z Usa o programa ‘compress’ durante o processamento dos arquivos.

-z Usa o programa ‘gzip’ durante o processamento dos arquivos.

–use-compress-program [PROGRAMA] Usa o [PROGRAMA] durante o processamento dos arquivos. Ele deve aceitar a opção ‘-d’.

-[0-7][lmh] Especifica a unidade e sua densidade. A extensão precisa ser especificada no arquivo de destino para a identificação correta:

- Arquivos gerados pelo ‘tar’ precisam ter a extensão ‘.tar’
- Caso seja usada a opção ‘-j’ para compactação, a extensão deverá ser ‘.tar.bz2’
- Caso seja usada a opção ‘-z’ para compactação, a extensão deverá ser ‘.tar.gz’ ou ‘.tgz’
- Caso seja usada a opção ‘-Z’ para a compactação, a extensão deverá ser ‘.tar.Z’ ou ‘.tgZ’ É importante saber qual qual o tipo de compactador usado durante a geração do arquivo ‘.tar’ pois será necessário especificar a opção apropriada para descompacta-lo

Exemplos:

- ‘tar -cf index.txt.tar index.txt’ - Cria um arquivo chamado ‘index.txt.tar’ que armazenará o arquivo ‘index.txt’. Você pode notar digitando ‘ls -la’ que o arquivo ‘index.txt’ foi somente arquivado (sem compactação), isto é útil para juntar diversos arquivos em um só.
- ‘tar -xf index.txt.tar’ - Desarquiva o arquivo ‘index.txt’ criado pelo comando acima.
- ‘tar -czf index.txt.tar.gz index.txt’ - O mesmo que o exemplo de arquivamento anterior, só que agora é usado a opção ‘-z’ (compactação através do programa ‘gzip’). Você agora pode notar digitando ‘ls -la’ que o arquivo ‘index.txt’ foi compactado e depois arquivado no arquivo ‘index.txt.tar.gz’ (você também pode chama-lo de ‘index.txt.tgz’ que também identifica um arquivo ‘.tar’ compactado pelo ‘gzip’)
- ‘tar -xzf index.txt.tar.gz’ - Descompacta e desarquiva o arquivo ‘index.txt.tar.gz’ criado com o comando acima.
- ‘gzip -dc index.tar.gz | tar -xf -’ - Faz o mesmo que o comando acima só que de uma forma diferente: Primeiro descompacta o arquivo ‘index.txt.tar.gz’ e envia a saída do arquivo descompactado para o ‘tar’ que desarquivará o arquivo ‘index.txt’.
- ‘tar -cjf index.txt.tar.bz2 index.txt’ - Arquiva o arquivo ‘index.txt’ em ‘index.txt.tar.bz2’ compactando através do ‘bzip2’ (opção -j).

- ‘tar -xjf index.txt.tar.bz2’ - Descompacta e desarquiva o arquivo ‘index.txt.tar.bz2’ criado com o comando acima.
- ‘bzip2 -dc index.txt.tar.bz2 | tar -xf -’ - Faz o mesmo que o comando acima só que de uma forma diferente: Primeiro descompacta o arquivo ‘index.txt.tar.bz2’ e envia a saída do arquivo descompactado para o ‘tar’ que desarquivará o arquivo ‘index.txt’.
- ‘tar -t index.txt.tar’ - Lista o conteúdo de um arquivo ‘.tar’.
- ‘tar -tz index.txt.tar.gz’ - Lista o conteúdo de um arquivo ‘.tar.gz’.

14.7 bzip2

É um novo compactador que vem sendo cada vez mais usado porque consegue atingir a melhor compactação em arquivos texto se comparado aos já existentes (em consequencia sua velocidade de compactação também é menor; quase duas vezes mais lento que o ‘gzip’). Suas opções são praticamente as mesmas usadas no ‘gzip’ e você também pode usa-lo da mesma forma. A extensão dos arquivos compactados pelo ‘bzip2’ é a ‘.bz2’

‘bzip2 [*opções*] [*arquivos*]’

Onde:

arquivos Especifica quais arquivos serão compactados pelo ‘bzip2’. Caso seja usado um ‘-’, será assumido a entrada padrão. Coringas podem ser usados para especificar vários arquivos de uma só vez (veja Seção 2.3, ‘Coringas).

Opções

-d, –decompress [arquivo] Descompacta um arquivo.

-f Força a compactação, compactando até mesmo links.

-l [arquivo] Lista o conteúdo de um arquivo compactado pelo ‘bzip2’.

-r Compacta diretórios e sub-diretórios.

-c [arquivo] Descompacta o arquivo para a saída padrão.

-t [arquivo] Testa o arquivo compactado pelo ‘bzip2’.

-[num], –fast, –best Ajustam a taxa de compactação/velocidade da compactação. Quanto melhor a taxa menor é a velocidade de compactação e vice versa. A opção ‘–fast’ permite uma compactação rápida e tamanho do arquivo maior. A opção ‘–best’ permite uma melhor compactação e uma velocidade menor. O uso da opção ‘-[número]’ permite especificar uma compactação individualmente usando números entre 1 (menor compactação) e 9 (melhor compactação). É útil para buscar um bom equilibro entre taxa de compactação/velocidade (especialmente em computadores muito lentos). Quando um arquivo é compactado pelo ‘bzip2’, é automaticamente acrescentada a extensão ‘.bz2’ ao seu nome. As permissões de acesso dos arquivos são também armazenadas no arquivo compactado.

Exemplos:

- ‘bzip2 -9 texto.txt’ - Compacta o arquivo ‘texto.txt’ usando a compactação máxima (compare o tamanho do arquivo compactado usando o comando ‘ls -la’).
- ‘bzip2 -d texto.txt.bz2’ - Descompacta o arquivo ‘texto.txt’
- ‘bzip2 -c texto.txt.bz2’ - Descompacta o arquivo ‘texto.txt’ para a saída padrão (tela)
- ‘bzip2 -9 *.txt’ - Compacta todos os arquivos que terminam com ‘.txt’
- ‘bzip2 -t texto.txt.bz2’ - Verifica o arquivo ‘texto.txt.bz2’.

14.8 rar

'rar' é um compactador desenvolvido por 'Eugene Roshal' e possui versões para 'GNU/Linux', 'DOS', 'Windows', 'OS/2' e 'Macintosh'. Trabalha com arquivos de extensão '.rar' e permite armazenar arquivos compactados em vários disquetes (múltiplos volumes). Se trata de um produto comercial, mas possui boas versões Shareware e pode ser muito útil em algumas situações.

'rar [*ações*] [*opções*] [*arquivo-destino.rar*] [*arquivos-origem*]'

Onde:

arquivo-destino.rar É o nome do arquivo de destino

arquivos-origem Arquivos que serão compactados. Podem ser usados coringas para especificar mais de um arquivo.

ações

a Compacta arquivos.

x Descompacta arquivos.

d Apaga arquivos especificados.

t Verifica o arquivo compactado em busca de erros.

c Inclui comentário no arquivo compactado.

r Repara um arquivo '.rar' danificado.

l Lista arquivos armazenados no arquivo compactado.

u Atualiza arquivos existentes no arquivo compactado.

m Compacta e apaga os arquivos de origem (move).

e Descompacta arquivos para o diretório atual.

p Mostra o conteúdo do arquivo na saída padrão.

rr Adiciona um registro de verificação no arquivo.

s Converte um arquivo '.rar' normal em arquivo auto-extractil. Arquivos auto-extracteis são úteis para enviar arquivos a pessoas que não tem o programa 'rar'. Basta executar o arquivo e ele será automaticamente descompactado (usando o sistema operacional que foi criado). Note que esta opção requer que o arquivo 'default.sfx' esteja presente no diretório home do usuário. Use o comando 'find' para localiza-lo em seu sistema.

opções

o+ Substitui arquivos já existentes sem perguntar.

o- Não substitui arquivos existentes.

sfx Cria arquivos auto-extracteis. Arquivos auto-extracteis são úteis para enviar arquivos a pessoas que não tem o programa 'rar'. Basta executar o arquivo e ele será automaticamente descompactado. Note que este processo requer que o arquivo 'default.sfx' esteja presente no diretório home do usuário. Use o comando 'find' para localiza-lo em seu sistema.

y Assume 'sim' para todas as perguntas.

r Inclui sub-diretórios no arquivo compactado.

x [ARQUIVO] Processa tudo menos o [ARQUIVO]. Pode ser usados coringas.

v [TAMANHO] Cria arquivos com um limite de tamanho. Por padrão, o tamanho é especificado em bytes, mas o número pode ser seguido de 'k' (kilobytes) ou 'm'(megabytes). Exemplo: 'rar a -v1440k ...' ou 'rar a -v10m ...'

p [SENHA] Inclui senha no arquivo. **CUIDADO**, pessoas conectadas em seu sistema podem capturar a linha de comando facilmente e descobrir sua senha.

m [0-5] Ajusta a taxa de compactação/velocidade de compactação. 0 não faz compactação alguma (mais rápido) somente armazena os arquivos, 5 é o nível que usa mais compactação (mais lento).

ed Não inclui diretórios vazios no arquivo.

isnd Ativa emissão de sons de alerta pelo programa.

ierr Envia mensagens de erro para stderr.

inul Desativa todas as mensagens.

ow Salva o dono e grupo dos arquivos.

ol Salva links simbólicos no arquivo ao invés do arquivo físico que o link faz referência.

mm[f] Usa um método especial de compactação para arquivos multimídia (sons, vídeos, etc). Caso for usado 'mmf', força o uso do método multimídia mesmo que o arquivo compactado não seja deste tipo. Os arquivos gerados pelo 'rar' do 'GNU/Linux' podem ser usados em outros sistemas operacionais, basta ter o 'rar' instalado. Quando é usada a opção '-v' para a criação de múltiplos volumes, a numeração dos arquivos é feita na forma: 'arquivo.rar', 'arquivo.r00', 'arquivo.r01', etc, durante a descompactação os arquivos serão pedidos em ordem. Se você receber a mensagem 'cannot modify volume' durante a criação de um arquivo '.rar', provavelmente o arquivo já existe. Apague o arquivo existente e tente novamente.

Exemplos:

- 'rar a texto.rar texto.txt' - Compacta o arquivo 'texto.txt' em um arquivo com o nome 'texto.rar'
- 'rar x texto.rar' - Descompacta o arquivo 'texto.rar'
- 'rar a -m5 -v1400k textos.rar *' - Compacta todos os arquivos do diretório atual, usando a compactação máxima no arquivo 'textos.rar'. Note que o tamanho máximo de cada arquivo é 1440 para ser possível grava-lo em partes para disquetes.
- 'rar x -v -y textos.rar' - Restaura os arquivos em múltiplos volumes criados com o processo anterior. Todos os arquivos devem ter sido copiados dos disquetes para o diretório atual antes de prosseguir. A opção '-y' é útil para não precisar-mos responder 'yes' a toda pergunta que o 'rar' fizer. * 'rar t textos.rar' - Verifica se o arquivo 'textos.rar' possui erros.
- 'rar r textos.rar' - Repara um arquivo '.rar' danificado.

15 Personalização do Sistema

Este capítulo ensina como personalizar algumas características de seu sistema 'GNU/Linux'.

15.1 Variáveis de Ambientes

É um método simples e prático que permite a especificação de opções de configuração de programas sem precisar mexer com arquivos no disco ou opções. Algumas variáveis do 'GNU/Linux' afetam o comportamento de todo o Sistema Operacional, como o idioma utilizado e o path . Variáveis de ambientes são nomes que contém algum valor e tem a forma 'Nome=Valor'. As variáveis de ambiente são individuais para cada usuário do sistema ou consoles virtuais e permanecem residentes na memória RAM até que o usuário saia do sistema (logo-off) ou até que o sistema seja desligado.

As variáveis de ambiente são visualizadas/criadas através do comando 'set' ou 'echo $NOME' (apenas visualiza) e exportadas para o sistemas com o comando 'export NOME=VALOR'.

Nos maioria dos sistemas GNU/Linux , o local usado para especificar variáveis de ambiente é o '/etc/profile' . Todas as variáveis especificadas neste arquivos serão inicializadas e automaticamente exportadas na inicialização do sistema.

Exemplo: Para criar uma variável chamada 'TESTE' que contenha o valor '123456' digite: 'export TESTE=123456'. Agora para ver o resultado digite: 'echo $TESTE' ou 'set|grep TESTE'. Note que o '$' que antecede o nome 'TESTE' serve para identificar que se trata de uma variável e não de um arquivo comum.

15.2 Modificando o Idioma usado em seu sistema

O idioma usado em seu sistema pode ser modificado facilmente através das variáveis de ambiente. Atualmente a maioria dos programas estão sendo *localizados*. A localização é um recurso que especifica arquivos que contém as mensagens do programas em outros idiomas. Você pode usar o comando 'locale' para listar as variáveis de localização do sistema e seus respectivos valores. As principais variáveis usadas para determinar qual idioma os programas 'localizados' utilizarão são:

- 'LANG' - Especifica o *idioma_PAIS* local. Podem ser especificados mais de um idioma na mesma variável separando-os com ':', desta forma caso o primeiro não esteja disponível para o programa o segundo será verificado e assim por diante. A língua Inglesa é identificada pelo código 'C' e usada como padrão caso nenhum locale seja especificado. Por exemplo: 'export LANG=pt_BR', 'export LANG=pt_BR:pt_PT:C'

- ‘LC_MESSAGES’ - Especifica o idioma que serão mostradas as mensagens dos programas. Seu formato é o mesmo de ‘LANG’.
- ‘LC_ALL’ - Configura todas as variáveis de localização de uma só vez. Seu formato é o mesmo de ‘LANG’.

As mensagens de localização estão localizadas em arquivos individuais de cada programa em ‘/usr/share/locale/[Idioma]/LC_MESSAGE . Elas são geradas através de arquivos ‘potfiles’ (arquivos com a extensão ‘.po’ ou ‘.pot’ e são gerados catálogos de mensagens ‘.mo’. As variáveis de ambiente podem ser especificadas no arquivo ‘/etc/environment’ desta forma as variáveis serão carregadas toda a vez que seu sistema for iniciado. Você também pode especificar as variáveis de localização em seu arquivos de inicialização ‘.bash_profile’, ‘.bashrc’ ou ‘.profile’ assim toda a vez que entrar no sistema, as variáveis de localização personalizadas serão carregadas.

Na maioria das distribuições , você pode especificar estas variáveis no arquivo ‘/etc/profile’, desta forma toda a vez que seu sistema for iniciado as variáveis de localização serão carregadas e exportadas para o sistema. Para as mensagens e programas do X-Window usarem em seu idioma local, é preciso colocar as variáveis no arquivo ‘~/.xserverrc’ do diretório home de cada usuário e dar a permissão de execução neste arquivo (‘chmod 755 .xserverrc’). Lembre-se de incluir o caminho completo do arquivo executável do seu gerenciador de janelas na última linha deste arquivo (sem o ‘&’ no final), caso contrário o Xserver será finalizado logo após ler este arquivo.

Abaixo exemplos de localização com as explicações:

- ‘export LANG=pt_BR’ - Usa o idioma pt_BR como língua padrão do sistema. Caso o idioma Português do Brasil não esteja disponível, C é usado (Inglês).
- ‘export LANG=C’ - Usa o idioma Inglês como padrão (é a mesma coisa de não especificar ‘LANG’, pois o idioma Inglês é usado como padrão).
- ‘export LANG=pt_BR:pt_PT:es_ES:C’ - Usa o idioma Português do Brasil como padrão, caso não esteja disponível usa o Português de Portugal, se não estiver disponível usa o Espanhol e por fim o Inglês. É recomendável usar a variável ‘LC_ALL’ para especificar o idioma, desta forma todos os outras variáveis (‘LANG, MESSAGES, LC_MONETARY, LC_NUMERIC, LC_COLLATE, LC_CTYPE e LC_TIME’) serão configuradas automaticamente.

15.3 alias

Permite criar um apelido a um comando ou programa. Por exemplo, se você gosta de digitar (como eu) o comando ‘ls –color=auto’ para ver uma listagem longa e colorida, você pode usar o comando ‘alias’ para facilitar as coisas digitando: ‘alias ls=’ls –color=auto” (não se esqueça da meia aspa ’para identificar o comando’). Agora quando você digitar ‘ls’, a listagem será mostrada com cores.

Se você digitar ‘ls -la’, a opção ‘-la’ será adicionada no final da linha de comando do alias: ‘ls –color=auto -la’, e a listagem também será mostrada em cores.

15.4 Arquivo ‘/etc/profile’

Este arquivo contém comandos que são executados para *todos* os usuários do sistema no momento do login, além de determinar todas as variáveis do sistema. Somente o usuário root pode ter permissão para modificar este arquivo.

Este arquivo é lido antes do arquivo de configuração pessoal de cada usuário (‘.profile’(root) e ‘.bash_profile’).

Quando é carregado através de um shell que requer login (nome e senha), o ‘bash’ procura estes arquivos em seqüência e executa os comandos contidos, caso existam:

1. ‘/etc/profile’
2. ‘~/.bash_profile’
3. ‘~/.bash_login’
4. ‘~/.profile’

Caso o ‘bash’ seja carregado através de um shell que não requer login (um terminal no X, por exemplo), o seguinte arquivo é executado: ‘~/.bashrc’.

Observação: Nos maioria dos sistemas GNU/Linux, o profile do usuário root está configurado no arquivo ‘/root/.profile’. A razão disto é porque se o ‘bash’ for carregado através do comando ‘sh’, ele fará a inicialização clássica deste shell lendo primeiro o arquivo ‘/etc/profile’ e após o ‘~/.profile’ e ignorando o ‘.bash_profile’ e ‘.bashrc’ que são arquivos de configuração usados somente pelo ‘Bash’. Exemplo, inserindo a linha ‘mesg y’ no arquivo ‘/etc/profile’ permite que todos os usuários do sistema recebam pedidos de ‘talk’ de outros usuários. Caso um usuário não quiser receber pedidos de ‘talk’, basta somente adicionar a linha ‘"mesg y"’ no arquivo pessoal ‘.bash_profile’.

15.5 Arquivo ‘.bash_profile’

Este arquivo reside no diretório pessoal de cada usuário. É executado por shells que usam autenticação (nome e senha). ‘.bash_profile’ contém comandos e variáveis de ambiente que são executados para o usuário no momento do login no sistema após o ‘/etc/profile’. Note que este é um arquivo oculto pois tem um "." no inicio do nome.

Por exemplo colocando a linha: ‘alias ls=’ls –colors=auto” no ‘.bash_profile’, cria um apelido para o comando ‘ls –colors=auto’ usando ‘ls’, assim toda vez que você digitar ‘ls’ será mostrada a listagem colorida.

15.6 Arquivo ‘.bashrc’

Possui as mesmas características do ‘.bash_profile’ mas é executado por shells que não requerem autenticação (como uma seção de terminal no X).

Os comandos deste arquivo são executados no momento que o usuário inicia um shell com as características acima. Note que este é um arquivo oculto pois tem um "." no inicio do nome.

15.7 Arquivo ‘.hushlogin’

Deve ser colocado no diretório pessoal do usuário. Este arquivo faz o ‘bash’ pular as mensagens do ‘/etc/motd’, número de e-mails, etc. Exibindo imediatamente o aviso de comando após a digitação da senha.

15.8 Diretório ‘/etc/skel’

Este diretório contém os modelos de arquivos ‘.bash_profile’ e ‘.bashrc’ que serão copiados para o diretório pessoal dos usuários no momento que for criada uma conta no sistema. Desta forma você não precisará configurar estes arquivos separadamente para cada usuário.

16 Configuração do sistema

Este capítulo traz explicações sobre algumas configurações úteis que podem ser feitas no sistema. Neste documento assumimos que o kernel do seus sistema já possui suporte a página de código 860 (Portuguesa) e o conjunto de caracteres ‘ISO-8859-1’.

16.1 Acentuação

Permite que o ‘GNU/Linux’ use a acentuação. A acentuação do modo texto é independente do modo gráfico; você pode configurar tanto um como o outro ou ambos.

Note que os mapas de teclado usados em modo texto são diferentes dos usados em modo gráfico. Geralmente os mapas de teclados para o modo gráfico tem uma letra ‘X’ no nome.

16.1.1 Acentuação em modo Texto

Caso sua distribuição esteja acentuando corretamente no modo texto você não precisará ler esta seção. Antes de prosseguir, verifique se você possui o pacote ‘kbd’ ou ‘console-tools’ instalado em seu sistema com o comando: ‘rpm -qa|grep kbd’. Caso não existam, alguns programas de configuração e arquivos de fontes não estarão disponíveis.

Siga os passos abaixo para colocar e acentuação em funcionamento para o modo Texto :

Mapa de Teclados Verifique se possui o arquivo de *mapa de teclado* correspondente ao seu modelo de teclado. Um mapa de teclado são arquivos com a extensão ‘.map’ ou ‘.kmap’ que fazem a tradução do código enviado pelo teclado para um caracter que será exibido na tela além de outras funções como o "Dead Keys" (pressionamento de uma tecla que não gera nenhum caracter mas afetará o próximo caracter gerado - como na acentuação, quando você aperta o ‘” não aparece nada mas após apertar a letra ‘A’ um caracter ‘Á’ é exibido. A combinação ‘” + ‘A’ é um *Dead Key* e está definido no arquivo do mapa de teclados).

Os tipos de teclados mais usados aqui no Brasil são o *padrão EUA* e o *ABNT2*. O teclado *padrão EUA* é o modelo usado nos Estados Unidos e você precisará apertar ‘”+‘C’ para gerar um *Cedilha (ç)*, enquanto o teclado ABNT2 possui todas as teclas usadas no Brasil (semelhante a uma máquina de escrever) e o *Cedilha* possui sua própria tecla após a letra ‘L’. O mapa de teclados correspondente ao teclado *padrão EUA* é o ‘br-latin1’ enquanto o *ABNT2* é o ‘br-abnt2’.

Se não tiver o arquivo correspondente ao seu teclado ou não encontra-lo, você poderá copia-lo de http://www.metainfo.org/focalinux/c este arquivo possui 3 mapas de teclados para os 2 teclados Brasileiros mais usados e um de Portugal (raramente usado no Brasil). Descompacte o arquivo ‘Consolemaps_tar.gz’ para um local em seu sistema (por exemplo: ‘/tmp’) com o comando: ‘tar -xzvf Consolemaps_tar.gz’. Note que este arquivo serve somente para a configuração no modo texto (console), veja a seção seguinte para configurar a acentuação no modo gráfico.

Configurando o Mapa de Teclados Se o arquivo do mapa de teclados possuir a extensão '.gz', descompacte-o com o comando: 'gzip -dc *arquivo.gz* > /etc/sysconfig/console/default.map' ou 'gzip -d *arquivo.gz*' para descompactar e depois o comando 'cp *arquivo.kmap* /etc/sysconfig/console/default.map'.

Se o arquivo possuir a extensão '.tar.gz', descompacte-o com o comando: 'tar -zxvf *arquivo.tar.gz*' e depois use o comando 'cp *arquivo.kmap* /etc/sysconfig/console/default.kmap'.

Faça isto substituindo *arquivo.gz* ou *arquivo.tar.gz* com o nome do arquivo compactado que contém o mapa de teclados.

Você pode manter o arquivo '/etc/sysconfig/console/default.map.gz', pois este arquivo é lido pelos scripts de inicialização somente se o arquivo '/etc/sysconfig/console/default.map' não for encontrado.

Se desejar usar o comando 'loadkeys', você precisa copiar o mapa de teclados para um local conhecido no sistema, então copie o arquivo 'arquivo.kmap' para '/usr/lib/kbd/keymaps/i386/qwerty' (na maioria dos sistemas GNU/Linux) ou algum outro local apropriado. Note que o arquivo pode ser compactado pelo 'gzip' e copiado para '/usr/lib/kbd/keymaps/i386/qwerty' que será lido sem problemas pelo sistema encarregado de configurar o teclado e acentuação.

Configurando a fonte de Tela Descomente a linha 'FONT=iso01.f16' e modifique-a para 'FONT=lat1u-16.psf' no arquivo '/etc/sysconf

Esta linha diz ao sistema que *fonte* deve carregar para mostrar os caracteres na tela. A fonte de caracteres deve ser compatível com o idioma local, pois nem todas suportam caracteres acentuados. A fonte preferível para exibir os caracteres acentuados é a 'lat1u-16', o '-16' no nome do arquivo significa o tamanho da fonte. As fontes de tela estão disponíveis no diretório '/usr/lib/kbd/consolefonts'.

Neste ponto você pode verificar se o seu sistema esta reconhecendo corretamente a acentuação entrando no editor de textos 'ae' e digitando: 'áãâà'. Se todos os acentos apareceram corretamente, parabéns! você já passou pela parte mais difícil. Agora o próximo passo é a acentuação no 'Bash'.

Acentuação no aviso de comando ('bash') Para acentuar no 'Bash' (interpretador de comandos) é necessário alterar o arquivo '/etc/inputrc' e fazer as seguintes modificações:

1. Descomente a linha: '"#set convert-meta off"' você faz isto apagando o símbolo "#" antes do nome.
   Um comentário faz com que o programa ignore linha(s) de comando. É muito útil para descrever o funcionamento de comandos/programas (você vai encontrar muito isso no sistema 'GNU/Linux', tudo é muito bem documentado).

2. Inclua a seguinte linha no final do arquivo:
   set meta-flag on

3. O conteúdo deste arquivo deve ficar assim:
   set convert-meta off
   set input-meta on
   set output-meta on
   set meta-flag on

4. Digite 'exit' ou pressione 'CTRL'+'D' para fazer o logout. Entre novamente no sistema para que as alterações façam efeito.

Pronto! você já esta acentuando em modo texto!. Talvez seja necessário que faça alguma alteração em arquivos de configuração de outros programas para que possa acentuar corretamente (veja se existe algum arquivo com o nome correspondente ao programa no diretório '/etc').

Na maioria das distribuições traz o utilitário 'kbdconfig' que também faz a configuração do mapa de teclados de forma interativa e gravando automaticamente o mapa de teclados em '/etc/sysconfig/console/default.map.gz'. Se preferir usar o 'kbdconfig' ainda será necessário executar os passos acima para habilitação da fonte 'lat1u-16' e acentuação no 'bash'.

16.1.2 Acentuação em modo gráfico

A acentuação no modo gráfico é feita de maneira simples:

Mapa de Teclados Verifique se possui o arquivo de mapa de teclado para o modo gráfico que corresponde ao seu teclado. Um arquivo de mapa de teclado faz a tradução do código enviado pelo teclado para um caracter que será exibido na tela. Este tipo de arquivo é identificado com a extensão '.map'. Se não tiver este arquivo ou não encontra-lo, você pode copia-lo de http://www.metainfo.org/focalinux/download/outros/Xmodmaps_tar.gz, este arquivo possui 3 mapas de teclados para os 2 teclados Brasileiros mais usados e um de Portugal. Descompacte o arquivo 'Xmodmaps_tar.gz' para um local em seu sistema (por exemplo: '/tmp') com o comando: 'tar -xzvf Xmodmaps-tar.gz'. Note que os mapas de teclado do 'Xmodmaps_tar.gz' somente servem para a configuração no modo gráfico (X-Window).

Acentuação no X Para acentuar no X você precisará descompactar e copiar o arquivo de mapa de teclado adequado ao seu computador em cima do arquivo '/etc/X11/Xmodmap' que já está em seu sistema. No meu caso, eu usei o seguinte comando (após descompactar o arquivo): '"cp 'Xmodmap.us+' '/etc/X11/Xmodmap'"'. Agora você precisa reiniciar o servidor X para que as alterações façam efeito (ou digite 'xmodmap /etc/X11/Xmodmap' no 'xterm' para aplicar as alterações na seção atual).

17 Compilação

Este capítulo explica o que é compilação, os principais compiladores e como compilar programas e principalmente o Kernel do 'GNU/Linux' com o objetivo de personaliza-lo de acordo com os dispositivos usados em seu computador e/ou os recursos que planeja utilizar.

17.1 O que é compilação?

É a transformação de um programa em código fonte (programa escrito pelo programador) em linguagem de máquina (programa executável).

Existem centenas de linguagens de programação diferentes umas das outras, cada uma oferece recursos específicos para atender melhor uma necessidade ou características particulares, algumas são voltadas para bancos de dados, outras somente para a criação de interfaces comunicação (*front-ends*), aprendizado, etc. Cada linguagem de programação possui comandos específicos que desempenham alguma função, mas todas trabalham com variáveis de memória para a manipulação de dados de entrada/processamento.

17.2 Compilador

É o programa que converte o programa feito pelo programador em linguagem de máquina. Após o processo de compilação o programa estará pronto para ser executado como um arquivo binário.

Existem muitos compiladores no ambiente 'GNU/Linux', um dos mais usados é o 'gcc', o compilador para linguagem C.

18 Manutenção do Sistema

Este capítulo descreve como fazer a manutenção de seu sistema de arquivos e os programas de manutenção automática que são executados periodicamente pelo sistema.

18.1 Checagem dos sistemas de arquivos

A checagem do sistema de arquivos permite verificar se toda a estrutura para armazenamento de arquivos, diretórios, permissões, conectividade e superfície do disco estão funcionando corretamente. Caso algum problema exista, ele poderá ser corrigido com o uso da ferramenta de checagem apropriada. As ferramentas de checagem de sistemas de arquivos costumam ter seu nome iniciado por 'fsck' e terminados com o nome do sistema de arquivos que verifica, separados por um ponto:

- 'fsck.ext2' - Verifica o sistema de arquivos 'EXT2'. Pode também ser encontrado com o nome 'e2fsck'.
- 'fsck.minix' - Verifica o sistema de arquivos 'Minix'.
- 'fsck.msdos' - Verifica o sistema de arquivos 'Msdos'. Pode também ser encontrado com o nome 'dosfsck'.

Para verificar um sistema de arquivos é necessário que ele esteja desmontado caso contrário poderá ocorrer danos em sua estrutura. Para verificar o sistema de arquivos raíz (que não pode ser desmontado enquanto o sistema estiver sendo executado) você precisará inicializar através de um disquete e executar o 'fsck.ext2'.

18.1.1 fsck.ext2

Este utilitário permite verificar erros em sistemas de arquivos 'EXT2' (*Linux Native*).

'fsck.ext2 [*opções*] [*dispositivo*]'

Onde:

dispositivo É o local que contém o sistema de arquivos 'EXT2' que será verificado (partições, disquetes, arquivos).

opções

-c Faz o 'fsck.ext2' verificar se existem agrupamentos danificados na unidade de disco durante a checagem.

-d Debug - Mostra detalhes de processamento do 'fsck.ext2'.

-f Força a checagem mesmo se o sistema de arquivos aparenta estar em bom estado. Por padrão, um sistema de arquivos que aparentar estar em bom estado não são verificados.

-F Grava os dados do cache no disco antes de iniciar.

-l [arquivo] Inclui os blocos listados no [arquivo] como blocos defeituosos no sistema de arquivos. O formato deste arquivo é o mesmo gerado pelo programa 'badblocks'.

-L [arquivo] Faz o mesmo que a opção '-l', só que a lista de blocos defeituosos do dispositivo é completamente limpa e depois a lista do [arquivo] é adicionada.

-n Faz uma verificação de 'somente leitura' no sistema de arquivos. Com esta opção é possível verificar o sistema de arquivos montado. Será assumido 'não' para todas as perguntas e nenhuma modificação será feita no sistema de arquivos. Caso a opção '-c' seja usada junto com '-n', '-l' ou '-L', o sistema de arquivos será verificado e permitirá somente a atualização dos setores danificados não alterando qualquer outra área.

-p Corrige automaticamente o sistema de arquivos sem perguntar. É recomendável fazer isto manualmente para entender o que aconteceu, em caso de problemas com o sistema de arquivos.

-v Ativa o modo verbose (mais mensagens são mostradas durante a execução do programa).

-y Assume 'sim' para todas as questões.

Caso sejam encontrados arquivos problemáticos e estes não possam ser recuperados, o 'fsck.ext2' perguntará se deseja salva-los no diretório 'lost+found'. Este diretório é encontrado em todas as partições *ext2* .

Após sua execução é mostrado detalhes sobre o sistema de arquivos verificado como quantidade de blocos livres/ocupados e taxa de fragmentação.

Exemplos:
'fsck.ext2 /dev/hda2';
'fsck.ext2 -f /dev/hda2';
'fsck.ext2 -vrf /dev/hda1'.

18.2 fsck.minix

Verifica o sistema de arquivos *minix* em um dispositivo.

'fsck.minix [*opções*] [*dispositivo*]'

Onde:

dispositivo Partição, disquete ou arquivo que contém o sistema de arquivos 'Minix' que será verificado

opções

-f Verifica o sistema de arquivos mesmo se ele estiver perfeito.

-r Permite reparo manual do sistema de arquivos

-a Permite um reparo automático do sistema de arquivos. É recomendado fazer o reparo manual.

-v Verbose - Mostra detalhes durante a execução do programa

-s Exibe detalhes sobre os blocos de root.

Exemplo:
'fsck.minix -f /dev/hda8';
'fsck.minix -vf /dev/hda8'

18.3 badblocks

Procura blocos defeituosos em um dispositivo.

'badblocks [*opções*] [*dispositivo*]'

Onde:

dispositivo Partição, disquete ou arquivo que contém o sistema de arquivos que será verificado.

opções

-b [tamanho] Especifica o [tamanho] do bloco do dispositivo em bytes

-o [arquivo] Gera uma lista dos blocos defeituosos do disco no [arquivo]. Este lista pode ser usada com o programa 'fsck.ext2' junto com a opção '-l'.

-s Mostra o número de blocos checados durante a execução do 'badblocks'.

-v Modo verbose - São mostrados mais detalhes.

-w Usa o modo leitura/gravação. Usando esta opção o 'badblocks' procura por blocos defeituosos gravando alguns padrões (0xaa, 0x55, 0xff, 0x00) em cada bloco do dispositivo e comparando seu conteúdo. Nunca use a opção '-w' em um dispositivo que contém arquivos pois eles serão apagados!

Exemplo:
'badblocks -s /dev/hda6';
'badblocks -s -o bad /dev/hda6'

18.4 defrag

Permite desfragmentar uma unidade de disco. A fragmentação é o armazenamento de arquivos em áreas não seqüenciais (uma parte é armazenada no começo a outra no final, etc), isto diminui o desempenho da unidade de disco porque a leitura deverá ser interrompida e feita a movimentação da cabeça para outra região do disco onde o arquivo continua, por este motivo discos fragmentados tendem a fazer um grande barulho na leitura e o desempenho menor.

A desfragmentação normalmente é desnecessária no 'GNU/Linux' porque o sistema de arquivos *ext2* procura automaticamente o melhor local para armazenar o arquivo. Mesmo assim, é recomendável desfragmentar um sistema de arquivos assim que sua taxa de fragmentação subir acima de 10%. A taxa de fragmentação pode ser vista através do 'fsck.ext2'. Após o 'fsck.ext2' ser executado é mostrada a taxa de fragmentação seguida de 'non-contiguos'.

A ferramenta de desfragmentação usada no 'GNU/Linux' é o 'defrag' que vem com os seguintes programas:

- 'e2defrag' - Desfragmenta sistemas de arquivos *Ext2* .
- 'defrag' - Desfragmenta sistemas de arquivos *Minix* .
- 'xdefrag' - Desfragmenta sistemas de arquivos *Xia* .

O sistema de arquivos deve estar desmontado ao fazer a desfragmentação. Se quiser desfragmentar o sistema de arquivos raíz ('/'), você precisará inicializar através de um disquete e executar um dos programas de desfragmentação apropriado ao seu sistema de arquivos. A checagem individual de fragmentação em arquivos pode ser feita com o programa 'frag'.

ATENÇÃO: Retire cópias de segurança de sua unidade antes de fazer a desfragmentação. Se por qualquer motivo o programa de desfragmentação não puder ser completado, você poderá perder dados!

'e2defrag [*opções*] [*dispositivo*]'

Onde:

dispositivo Partição, arquivo, disquete que contém o sistema de arquivos que será desfragmentado.

opções

-d Debug - serão mostrados detalhes do funcionamento

-n Não mostra o mapa do disco na desfragmentação. É útil quando você inicializa por disquetes e recebe a mensagem "Failed do open term Linux" ao tentar executar o 'e2defrag'.

-r Modo somente leitura. O defrag simulará sua execução no sistema de arquivos mas não fará nenhuma gravação. Esta opção permite que o defrag seja usado com sistema de arquivos montado.

-s Cria um sumário da fragmentação do sistema de arquivos e performance do desfragmentador.

-v Mostra detalhes durante a desfragmentação do sistema de arquivos. Caso mais de uma opção -v seja usada, o nível de detalhes será maior.

-i [arquivo] Permite definir uma lista de prioridades em que um arquivo será gravado no disco, com isto é possível determinar se um arquivo será gravado no começo ou final da unidade de disco. Esta lista é lida do [arquivo] e deve conter uma lista de prioridades de -100 a 100 para cada inodo do sistema de arquivos. Arquivos com prioridade alta serão gravados no começo do disco. Todos os inodos terão prioridade igual a zero caso a opção '-i' não seja usada ou o inodo não seja especificado no [arquivo]. O [arquivo] deverá conter uma série de linhas com um número (inodo) ou um número prefixado por um sinal de igual seguido da prioridade.

-p [numero] Define o [numero] de buffers que serão usados pela ferramenta de desfragmentação na realocação de dados, quanto mais buffers mais eficiente será o processo de realocação. O número depende de quantidade memória RAM e Swap você possui. Por padrão 512 buffers são usados correspondendo a 512Kb de buffer (em um sistema de arquivos de blocos com 1Kb).

Exemplo:
‘e2defrag -n -v /dev/hdb4’;
‘e2defrag -r /dev/hda1’

18.5 Limpando arquivos de LOGS

Tudo que acontece em sistemas ‘GNU/Linux’ pode ser registrado em arquivos de log em ‘/var/log’, como vimos anteriormente. Eles são muito úteis por diversos motivos, para o diagnóstico de problemas, falhas de dispositivos, checagem da segurança, alerta de eventuais tentativas de invasão, etc.

O problema é quando eles começam a ocupar muito espaço em seu disco. Verifique quantos Megabytes seus arquivos de LOG estão ocupando através do comando ‘cd /var/log;du -hc’. Antes de fazer uma limpeza nos arquivos de LOG, é necessário verificar se eles são desnecessários e só assim zerar os que forem dispensáveis.

Não é recomendável apagar um arquivo de log pois ele pode ser criado com permissões de acesso indevidas (algumas distribuições fazem isso). Você pode usar o comando: ‘echo -n >arquivo’ ou o seguinte shell script para zerar todos os arquivos de LOG de uma só vez (as linhas iniciante com ‘#’ são comentários):

```
#! /bin/sh
cd /var/log
for l in ‘ls -p|grep ’/’‘;
do echo -n >$l &>/dev/null
echo Zerando arquivo $l...
done
echo Limpeza dos arquivos de log concluída!
```

Copie o conteúdo acima em um arquivo com a extensão ‘.sh’, dê permissão de execução com o ‘chmod’ e o execute como usuário ‘root’. É necessário executar este script para zerar arquivos de log em subdiretórios de ‘/var/log’, caso sejam usados em seu sistema.

Algumas distribuições GNU/Linux, fazem o arquivamento automático de arquivos de LOGs em arquivos ‘.gz’ através de scripts disparados automaticamente pelo ‘cron’.

ATENÇÃO: LEMBRE-SE QUE O SCRIPT ACIMA APAGARÁ TODOS OS ARQUIVOS DE LOGs DO SEU SISTEMA SEM POSSIBILIDADE DE RECUPERAÇÃO. TENHA ABSOLUTA CERTEZA DO QUE NÃO PRECISARÁ DELES QUANDO EXECUTAR O SCRIPT ACIMA!

18.6 Tarefas automáticas de manutenção do sistema

Os arquivos responsáveis pela manutenção automática do sistema se encontram em arquivos individuais localizados nos diretórios ‘/etc/cron.daily’, ‘/etc/cron.weekly’ e ‘/etc/cron.montly’. A quantidade de arquivos depende da quantidade de pacotes instalado em seu sistema, porque alguns programam tarefas nestes diretórios e não é possível descrever todas, para detalhes sobre o que cada arquivo faz veja o cabeçalho e o código de cada arquivo.

Estes arquivos são executados pelo ‘cron’ através do arquivo ‘/etc/crontab’. Você pode programar quantas tarefas desejar. Alguns programas mantém arquivos do ‘cron’ individuais em ‘/var/spool/cron/crontabs’ que executam comandos periodicamente.

18.7 cron

O ‘cron’ é um daemon que permite o agendamento da execução de um comando/programa para um determinado dia/mes/ano/hora. É muito usado em tarefas de arquivamento de logs, checagem da integridade do sistema e execução de programas/comandos em horários determinados.

As tarefas são definidas no arquivo ‘/etc/crontab’ e por arquivos individuais de usuários em ‘/var/spool/cron/crontabs/[usuário]’ (criados através do programa ‘crontab’). Adicionalmente as distribuições utilizam os arquivos no diretório ‘/etc/cron.d’ como uma extensão para o ‘/etc/crontab’.

Para agendar uma nova tarefa, basta editar o arquivo ‘/etc/crontab’ com qualquer editor de texto (como o ‘ae’ e o ‘vi’), ou o comando ’crontab -e’ , e definir o mes/dia/hora que a tarefa será executada. Não é necessário reiniciar o daemon do ‘cron’ porque ele verifica seus arquivos a cada minuto.

18.7.1 O formato de um arquivo crontab

O arquivo '/etc/crontab' tem o seguinte formato:

52	18	1	*	*	root	run-parts –report /etc/cron.montly
Minuto	Hora	Dia do Mês (1-31)	Mes (1-12)	Dia da semana (0-7)	UID que executará o comando	Comando que será executado

Onde:

Minuto Valor entre 0 e 59

Hora Valor entre 0 e 23

Dia do Mes Valor entre 0 e 31

Mes Valor entre 0 e 12 (identificando os meses de Janeiro a Dezembro)

Dia da Semana Valor entre 0 e 7 (identificando os dias de Domingo a Sábado). Note que tanto 0 e 7 equivalem a Domingo.

usuário O usuário especificado será usado para executar o comando (o usuário deverá existir).

comando Comando que será executado. Podem ser usados parâmetros normais usados na linha de comando.

Os campos do arquivo são separados por um ou mais espaços ou tabulações. Um asterisco '*' pode ser usado nos campos de data e hora para especificar todo o intervalo disponível. O hífen '-' serve para especificar períodos de execução (incluindo a o número inicial/final). A vírgula serve para especificar lista de números. Passos podem ser especificados através de uma '/'. Veja os exemplos no final desta seção.

O arquivo gerado em '/var/spool/cron/crontabs/[usuário]' pelo 'crontab' tem o mesmo formato do '/etc/crontab' exceto por não possuir o campo 'usuário (UID)', pois o nome do arquivo já identifica o usuário no sistema.

Para editar um arquivo de usuário em '/var/spool/cron/crontabs' ao invés de editar o '/etc/crontab' use 'crontab -e', para listar as tarefas daquele usuário 'crontab -l' e para apagar o arquivo de tarefas do usuário 'crontab -r' (adicionalmente você pode remover somente uma tarefa através do 'crontab -e' e apagando a linha correspondente).

OBS: Não esqueça de incluir uma linha em branco no final do arquivo, caso contrário o último comando não será executado.

O 'cron' define o valor de algumas variáveis automaticamente durante sua execução; a variável 'SHELL' é definida como '/bin/sh', 'PATH' como '/usr/bin:/bin', 'LOGNAME', 'MAILTO' e 'HOME' são definidas através do arquivo '/etc/passwd'. Os valores padrões destas variáveis podem ser substituídos especificando um novo valor nos arquivos do 'cron'.

Exemplos de um arquivo '/etc/crontab':

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
00 10 * * * root sync
# Executa o comando sync todo o dia as 10:00
00 06 * * 1 root updatedb
# Executa o comando updatedb toda segunda-feira as 06:00.
10,20,40 * * * * root runq
# Executa o comando runq todos os dias e a toda a hora em 10, 20 e 40 minutos.
*/10 * * * * root fetchmail
# Executa o comando fetchmail de 10 em 10 minutos todos os dias
15 0 25 12 * root echo "Feliz Natal"|smail john
# Envia um e-mail as 0:15 todo o dia 25/12 para john desejando um feliz natal.
30 5 * * 1-6 root poff
# Executa o comando poff automaticamente as 5:30 de segunda-feira a sábado.
```

19 X Window (ambiente gráfico)

Este capítulo do guia traz explicações sobre o ambiente gráfico X Window System.

19.1 O que é X Window?

É um sistema gráfico de janelas que roda em uma grande faixa de computadores, máquinas gráficas e diferentes tipos de máquinas e plataformas Unix. Pode tanto ser executado em máquinas locais como remotas através de conexão em rede.

19.2 A organização do ambiente gráfico X Window

Em geral o ambiente gráfico X Window é dividido da seguinte forma:

- 'O Servidor X' - É o programa que controla a exibição dos gráficos na tela, mouse e teclado. Ele se comunica com os programas cliente através de diversos métodos de comunicação. O servidor X pode ser executado na mesma máquina que o programa cliente esta sendo executado de forma transparente ou através de uma máquina remota na rede.
- 'O gerenciador de Janelas' - É o programa que controla a aparência da aplicação. Os gerenciadores de janelas (window managers) são programas que atuam entre o servidor X e a aplicação. Você pode alternar de um gerenciador para outro sem fechar seus aplicativos.

Existem vários tipos de gerenciadores de janelas disponíveis no mercado entre os mais conhecidos posso citar o 'Window Maker (feito por um Brasileiro)', o 'After Step', 'Gnome', 'KDE', 'twm' (este vem por padrão quando o servidor X é instalado), 'Enlightenment', 'IceWm', etc.

A escolha do seu gerenciador de janelas é pessoal, depende muito do gosto de cada pessoa e dos recursos que deseja utilizar.

- 'A aplicação cliente' - É o programa sendo executado.

Esta organização do ambiente gráfico X traz grandes vantagens de gerenciamento e recursos no ambiente gráfico UNIX, uma vez que tem estes recursos você pode executar seus programas em computadores remotos, mudar totalmente a aparência de um programa sem ter que fecha-lo (através da mudança do gerenciador de janelas), etc.

19.3 Iniciando o X

O sistema gráfico X pode ser iniciado de duas maneiras:

- 'Automática' - Usando o programa 'xdm' que é um programa que roda no ambiente gráfico X e apresenta uma tela pedindo nome e senha para entrar no sistema (login). Após entrar no sistema, o X executará um dos gerenciadores de janelas configurados.
- 'Manual' - Através do comando 'startx', ou 'xstart' . Neste caso o usuário deve entrar com seu nome e senha para entrar no modo texto e então executar um dos comandos acima. Após executar um dos comandos acima, o servidor X será iniciado e executará um dos gerenciadores de janelas configurados no sistema.

19.4 Servidor X

Como dito acima, o servidor X controla o teclado, mouse e a exibição dos gráficos em sua tela. Para ser executado, precisa ser configurado através do arquivo '/etc/X11/XF86Config' ou usando o utilitário 'xf86config' (modo texto).

A finalização do servidor X é feita através do pressionamento simultâneo das teclas 'CTRL', 'ALT', 'BackSpace'. O servidor X é imediatamente terminado e todos os gerenciadores de janelas e programas clientes são fechados.

> CUIDADO: Sempre utilize a opção de saída de seu gerenciador de janelas para encerrar normalmente uma seção X11 e salve os trabalhos que estiver fazendo antes de finalizar uma seção X11. A finalização do servidor X deve ser feita em caso de emergência quando não se sabe o que fazer para sair de um gerenciador de janelas ou de um programa mal comportado.

20 Como obter ajuda no sistema

Dúvidas são comuns durante o uso do 'GNU/Linux' e existem várias maneiras de se obter ajuda e encontrar a resposta para algum problema. O 'GNU/Linux' é um sistema bem documentado, provavelmente tudo o que imaginar fazer ou aprender já esta disponível para leitura e aprendizado. Abaixo segue algumas formas úteis para encontrar a solução de sua dúvida, vale a pena conhece-las.

20.1 Páginas de Manual

As *páginas de manual* acompanham quase todos os programas 'GNU/Linux'. Elas trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento de opção. Uma página de manual é visualizada na forma de texto único com rolagem vertical. Também documenta parâmetros usados em alguns arquivos de configuração.

A utilização da página de manual é simples, digite:

'man [*seção*] [*comando/arquivo*]'

onde:

seção É a seção de manual que será aberta, se omitido, mostra a *primeira* seção sobre o comando encontrada (em ordem crescente).

comando/arquivo Comando/arquivo que deseja pesquisar.

A navegação dentro das páginas de manual é feita usando-se as teclas:

- q - Sai da página de manual
- PageDown ou f - Rola 25 linhas abaixo
- PageUP ou w - Rola 25 linhas acima
- SetaAcima ou k - Rola 1 linha acima
- SetaAbaixo ou e - Rola 1 linha abaixo
- r - Redesenha a tela (refresh)
- p ou g - Inicio da página
- h - Ajuda sobre as opções da página de manual
- s - Salva a página de manual em formato texto no arquivo especificado (por exemplo: '/tmp/ls').

Exemplo, 'man ls', 'man 5 hosts_access'.

20.2 Info Pages

Idêntico as páginas de manual, mas é usada navegação entre as páginas. Se pressionarmos <Enter> em cima de uma palavra destacada, a 'info pages' nos levará a seção correspondente. A *info pages* é útil quando sabemos o nome do comando e queremos saber para o que ele serve. Também traz explicações detalhadas sobre uso, opções e comandos.

Para usar a info pages, digite:

'info [*comando/programa*]'

Se o nome do *comando/programa* não for digitado, a info pages mostra a lista de todos os manuais de *comandos/programas* disponíveis. A navegação da info pages é feita através de nomes marcados com um "*" (hipertextos) que se pressionarmos <Enter>, nos levará até a seção correspondente. A *info pages* possui algumas teclas de navegação úteis:

- q - Sai da info pages
- ? - Mostra a tela de ajuda (que contém a lista completa de teclas de navegação e muitos outras opções).
- n - Avança para a próxima página
- p - Volta uma página
- u - Sobre um nível do conteúdo (até checar ao índice de documentos)
- m - Permite usar a localização para encontrar uma página do 'info'. Pressione 'm', digite o comando e tecle <Enter> que será levado automaticamente a página correspondente.
- d - Volta ao índice de documentos.

Existem muitos outras teclas de navegação úteis na info pages, mas estas são as mais usadas. Para mais detalhes, entre no programa 'info' e pressione '?'.

Exemplo, 'info cvs'.

20.3 Help on line

Ajuda rápida, é útil para sabermos quais opções podem ser usadas com o comando/programa. Quase todos os comandos/programas 'GNU/Linux' oferecem este recurso que é útil para consultas rápidas (e quando não precisamos dos detalhes das páginas de manual). É útil quando se sabe o nome do programa mas deseja saber quais são as opções disponíveis e para o que cada uma serve. Para acionar o *help on line* , digite:

'[*comando*] –help'

comando É o comando/programa que desejamos ter uma explicação rápida.

O Help on Line não funciona com comandos internos (embutidos no Bash), para ter uma ajuda rápida sobre os comandos internos, veja Seção 16.4, 'help'.

Por exemplo, 'ls –help'.

20.4 help

Ajuda rápida, útil para saber que opções podem ser usadas com os *comandos internos* do interpretador de comandos. O comando 'help' somente mostra a ajuda para comandos internos. Para usar o 'help' digite:

'help [*comando*]'

Por exemplo, 'help echo', 'help exit'

20.5 apropos/whatis

Apropos procura por *programas/comandos* através da descrição. É útil quando precisamos fazer alguma coisa mas não sabemos qual comando usar. Ele faz sua pesquisa nas páginas de manual existentes no sistema e lista os comandos/programas que atendem a consulta. Para usar o comando 'apropos' digite:

'apropos [*descrição*]'

Digitando 'apropos copy', será mostrado todos os comandos que tem a palavra 'copy' em sua descrição (provavelmente os programas que copiam arquivos, mas podem ser mostrados outros também).

20.6 locate

Localiza uma palavra na estrutura de arquivos/diretórios do sistema. É útil quando queremos localizar onde um comando ou programa se encontra (para copia-lo, curiosidade, etc). A pesquisa é feita em um banco de dados construído com o comando 'updatedb' sendo feita a partir do diretório raíz '/' e sub-diretórios. Para fazer uma consulta com o 'locate' usamos:

'locate [*expressão*]'

A *expressão* deve ser o nome de um arquivo diretório ou ambos que serão procurados na estrutura de diretórios do sistema. Como a consulta por um programa costuma localizar também sua página de manual, é recomendável usar *"pipes"* para filtrar a saída do comando .

Por exemplo, para listar os diretórios que contém o nome *" cp "*: 'locate cp'. Agora mostrar somente arquivos binários, usamos: 'locate cp|grep bin/'

20.7 which

Localiza um programa na estrutura de diretórios do path. É muito semelhante ao 'locate', mas a busca é feita no 'path' do sistema e somente são mostrados arquivos executáveis .

'which [*programa/comando*]'.

20.8 Documentos HOWTO's

São documentos em formato *texto* , *html* , etc, que explicam como fazer determinada tarefa ou como um programa funciona. Normalmente são feitos na linguagem 'SGML' e convertidos para outros formatos (como o texto, HTML, PostScript) depois de prontos.

Estes trazem explicações detalhadas desde como usar o 'bash' até sobre como funciona o modem ou como montar um *servidor internet completo* .

Os HOWTO´s podem ser encontrados no diretório do projeto de documentação do 'GNU/Linux' (LDP) em ftp://metalab.unc.edu/pub ou traduzidos para o Português pelo LDP-BR em http://ldp-br.conectiva.com.br/comofazer. Caso tenha optado por instalar o pacote de HOWTO's de sua distribuição 'GNU/Linux', eles podem ser encontrados em: '/usr/doc/how-to'

20.9 Documentação de Programas

São documentos instalados junto com os programas. Alguns programas também trazem o *aviso de copyright, changelogs, modelos, scripts, exemplos e FAQs (perguntas frequêntes)* junto com a documentação normal. Seu princípio é o mesmo do How-to; documentar o programa. Estes arquivos estão localizados em:

'/usr/doc/'[*programa*]".

programa É o nome do programa ou comando procurado.

20.10 FAQ

FAQ é um arquivo de perguntas e respostas mais frequêntes sobre o programa. Os arquivos de FAQ estão localizados em:

'/usr/doc/FAQ/'[*programa*]".

programa É o nome do programa ou comando procurado.

20.11 Listas de discussão

São grupos de usuários que trocam mensagens entre si, resolvem dúvidas, ajudam na configuração de programas, instalação, etc. É considerado o melhor suporte ao 'GNU/Linux' pois qualquer participante pode ser beneficiar das soluções discutidas. Existem milhares de listas de discussões sobre o 'GNU/Linux' espalhadas pelo mundo, em Português existem algumas dezenas.

Algumas listas são específicas a um determinado assunto do sistema, algumas são feitas para usuários iniciantes ou avançados, outras falam praticamente de tudo. Existem desde usuários iniciantes, hackers, consultores, administradores de redes experientes e gurus participando de listas e oferecendo suporte de graça a quem se aventurar em instalar e usar o sistema 'GNU/Linux'.

A lista de discussão funciona da seguinte forma: você se inscreve na lista enviando uma mensagem ao endereço de inscrição, será enviada um pedido de confirmação por e-mail, simplesmente dê um reply na mensagem para ser cadastrado. Pronto! agora você estará participando do grupo de usuários e receberá todas as mensagens dos participantes do grupo. Assim você poderá enviar sua mensagem e ela será vista por todos os participantes da lista.

Da mesma forma, você pode responder uma dúvida de outro usuário da lista ou discutir algum assunto, tirar alguma dúvida sobre a dúvida de outra pessoa, etc.

Não tenha vergonha de enviar sua pergunta, participar de listas de discussão é uma experiência quase obrigatório de um 'Linuxer'. Abaixo segue uma relação de listas de discussão em Português com a descrição, endereço de inscrição, e o que você deve fazer para ser cadastrado:

<debian-user-portuguese@lists.debian.org> Lista de discussão para usuários Portugueses da 'Debian'. Também são discutidos assuntos relacionados ao Linux em geral. A inscrição é aberta a todos os interessados.

Para se inscrever, envie uma mensagem para <debian-user-portuguese-request@lists.debian.org> contendo a palavra 'subscribe' no assunto da mensagem. Será enviada uma mensagem a você pedindo a confirmação da inscrição na lista de discussão, simplesmente dê um reply na mensagem (responder) e você estará cadastrado e poderá enviar e receber mensagens dos participantes.

<debian-news-portuguese@lists.debian.org> A 'Debian' é extremamente bem estruturada quanto a divulgações e notícias, várias listas de email e várias páginas compõe essa base. A *Debian Weekly News* é especialmente importante pois dá uma visão geral do que se passou na 'Debian' durante a semana. Sua versão Brasileira é editada e traduzida por *Gustavo Noronha Silva (kov)* e conta com a lista de pacotes traduzida por *Adriano Freitas (afreitas)* e revisada por *Hilton Fernandes* e não traz apenas traduções mas também adições dos acontecimentos atuais da 'Debian' no Brasil, ou projetos concluídos ou lançados pela equipe *Debian-br* (http://debian-br.sourceforge.net).

Essa lista 'NÃO' é usada para resolução de dúvidas e problemas, apenas para o RECEBIMENTO de notícias relacionadas a Debian. Não poste mensagens nela!

Para se inscrever, envie uma mensagem para <debian-news-portuguese-request@lists.debian.org> contendo a palavra 'subscribe' no assunto da mensagem. Será enviada uma mensagem a você pedindo a confirmação da inscrição na lista de discussão, simplesmente dê um reply na mensagem (responder) e você passará a receber as notícias sobre a Debian em Português.

<linux-br@unicamp.br> Lista de discussão que cobre assuntos diversos. Esta lista é voltada para usuários com bons conhecimentos no 'GNU/Linux', são abordados assuntos como redes, configurações, etc. Esta é uma lista moderada, o que significa que a mensagem que envia passam por uma pessoa que verifica (modera) e a libera caso estejam dentro das normas adotada na lista. É uma lista de alto nível e recomendada para quem deseja fugir de mensagens como 'não consigo instalar o Linux', 'não sei compilar o kernel', 'o que eu faço quando vejo uma tela com o nome login:?', etc.

Para se inscrever nesta lista, envie uma mensagem para:

<linux-br-request@unicamp.br> contendo a palavra ‘subscribe’ no assunto da mensagem e aguarde o recebimento da confirmação da inscrição. Apenas responda a mensagem de confirmação para se inscrever. Para se descadastrar envie uma mensagem para o mesmo endereço mas use a palavra ‘unsubscribe’.

<dicas-l@unicamp.br> Esta lista envia diariamente uma dica de ‘Unix’, sistemas da Microsoft ou novidades da Internet.

Para se inscreve nesta lista de discussão, envie uma mensagem para: <dicas-l-request@unicamp.br> contendo a palavra ‘subscribe’ no corpo da mensagem e aguarde o recebimento da confirmação da inscrição. Apenas responda a mensagem de confirmação para confirmar sua inscrição na lista. Para se descadastrar envie uma mensagem para o mesmo endereço mas use a palavra ‘unsubscribe’.

<linux-br@listas.conectiva.com.br> Discute todos os aspectos relacionados ao uso, instalação, atualização e operação do ‘GNU/Linu

Para se inscreve nesta lista de discussão, envie uma mensagem para: <linux-br-request@listas.conectiva.com.br> contendo a palavra ‘subscribe’ no corpo da mensagem e aguarde o recebimento da confirmação da inscrição. Apenas responda a mensagem de confirmação para confirmar sua inscrição na lista. Para se descadastrar envie uma mensagem para o mesmo endereço mas use a palavra ‘unsubscribe’.

<debian-news-portuguese@lists.debian.org> Lista de discussão que veicula semanalmente a tradução de noticias postadas a listas ‘debian-news’ (em Inglês), listas de novos pacotes incluídos na distribuição com a descrição traduzida e noticias relacionadas aos esforços da Internacionalização da ‘Debian’ no Brasil.

Para se inscrever, envie uma mensagem para <debian-news-portuguese-request@lists.debian.org> contendo a palavra ‘subscribe’ no assunto da mensagem. Será enviada uma mensagem a você pedindo a confirmação da inscrição na lista de discussão, simplesmente dê um reply na mensagem (responder) e você estará cadastrado e passará a receber as noticias semanais enviadas a lista.

Esta listagem deveria estar mais completa, mas eu não lembro de todas as listas!.

RECOMENDAÇÕES AO PARTICIPAR DE LISTAS DE DISCUSSÕES

- Não envie mensagens em maiúsculo porque VAI PARECER QUE VOCÊ ESTÁ GRITANDO!. Isto é uma regra de etiqueta na internet. Recebi muitas mensagens de gente escrevendo TODO o e-mail desta forma e ignorei a todas!
- Sempre coloque um assunto (subject) na mensagem. O assunto serve como um resumo do problema ou dúvida que tem. Alguns usuários, principalmente os que participam de várias listas de discussão, verificam o assunto da mensagem e podem simplesmente descartar a mensagem sem lê-la porque as vezes ele não conhece sobre aquele assunto.
- Nunca use "Socorro!", "Help!" ou coisa do gênero como assunto, seja objetivo sobre o problema/dúvida que tem: *"Falha ao carregar módulo ne do kernel"* , *"SMAIL retorna a mensagem Access denied"* , *"Novidades: Nova versão do guia Foca Linux"* ;-).
- Procure enviar mensagens em formato ‘texto’ ao invés de ‘HTML’ para as listas de discussão pois isto faz com que a mensagem seja vista por todos os participantes (muitos dos usuários ‘GNU/Linux’ usam leitores de e-mail que não suportam formato html) e diminui drasticamente o tamanho da mensagem porque o formato texto não usa tags e outros elementos que a linguagem HTML contém (muitos dos usuários costumam participar de várias listas de discussão, e mensagens em HTML levam a um excesso de tráfego e tempo de conexão).
- Muitas pessoas reclamam do excesso de mensagens recebidas das listas de discussão. Se você recebe muitas mensagens, procure usar os *filtros de mensagens* para organiza-las. O que eles fazem é procurar por campos na mensagem, como o remetente, e enviar para um local separado. No final da filtragem, todas as mensagens de listas de discussão estarão em locais separados e as mensagens enviadas diretamente a você entrarão na caixa de correio principal, por exemplo.

O Netscape também tem recursos de filtros de mensagem que podem ser criadas facilmente através da opção "Arquivo/Nova Sub-Pasta" ("File/New Subfolder") do programa de E-mail. Então defina as regras através do menu "Editar/Filtros de Mensagens" ("Edit/Message filters") clicando no botão "Novo"("New").

Guia Foca GNU/Linux
Gleydson Mazioli da Silva <gleydson@escelsanet.com.br>
Versão 3.70 - sábado, 15 de setembro de 2001