

Sumário

1	O que é Software Livre	5
2	Porque GNU/Linux ?	6
3	O Kernel	6
4	A grande Família	7
5	Conceitos Básicos	8
5.1	Ambiente Modo Texto e Modo Gráfico	8
5.1.1	Modo Texto	8
5.1.2	Modo Gráfico	9
6	Ambientes Gráficos	9
6.1	Kde - The K Desktop Environment	10
6.2	Gnome - Computing made easy	10
6.3	Window Maker - Wmaker - X11 Window Manager	10
6.4	FluxBox - Fluxbox - X11 Window Manager	11
7	Entrando e Saindo do Sistema	11
7.1	Login	11
7.2	Logout	11
8	A Hierarquia	11
8.1	O Root	11
8.2	O Usuário	11
9	X Window (ambiente gráfico)	12
9.1	O servidor X	12
9.2	A organização do ambiente gráfico X Window	12
10	Diretórios	12
10.1	Diretório	12
10.2	Diretório Raíz	13
10.3	Diretório home	13
10.3.1	Diretório Superior	13
10.3.2	Diretório Anterior	13
10.3.3	Caminho na estrutura de diretórios	13
10.3.4	Nomeando Arquivos e Diretórios	14
11	Comandos	14
11.0.5	Aviso de comando (Prompt)	15
11.0.6	Interpretador de comandos	16
11.0.7	Terminal Virtual (console)	16

12 Comandos para manipulação de diretório	17
12.1 ls	17
12.2 cd	17
12.3 pwd	18
12.4 mkdir	18
12.5 rmdir	18
13 Comandos para manipulação de Arquivos	18
13.1 cat	18
13.2 rm	19
13.3 cp	19
13.4 mv	20
14 Comandos Diversos	21
14.1 clear	21
14.2 date	21
14.3 df	21
14.4 ln	22
14.5 du	22
14.6 find	23
14.7 free	23
14.8 grep	24
14.9 more	24
14.10less	24
14.11time	24
14.12uptime	25
14.13dmesg	25
14.14echo	25
14.15su	25
14.16uname	25
14.17reboot	25
14.18shutdown	25
14.19diff	26
14.20patch	28
15 Redirecionamentos e Pipe	28
15.1 >	29
15.2 >>	29
15.3 <	29
15.4 <<	29
15.5 	29
15.6 Diferença entre o e o >	30
16 Discos e Partições	30
16.1 Partições	30
16.2 Pontos de Montagem	30
16.3 Identificação de discos e partições em sistemas Linux	30
16.4 Montando (acessando) uma partição de disco	31
16.5 fstab	32

16.6	Desmontando uma partição de disco	33
17	Execução de programas	33
17.1	Executando um comando/programa	33
17.2	Path	33
17.3	Tipos de Execução de comandos/programas	34
18	Executando programas em seqüência	34
18.1	ps	34
18.2	top	35
19	Controle de execução de processos	35
19.1	Interrompendo a execução de um processo	36
19.2	Parando momentaneamente a execução de um processo	36
19.3	jobs	36
19.4	fg	36
19.5	bg	36
19.6	kill	37
19.7	killall	37
20	Fechando um programa quando não se sabe como sair	37
20.1	Eliminando caracteres estranhos	38
21	Permissões de acesso a arquivos e diretórios	39
21.1	Donos, grupos e outros usuários	39
21.2	Tipos de Permissões de acesso	40
21.3	Etapas para acesso a um arquivo/diretório	40
22	Exemplos práticos de permissões de acesso	41
22.1	Exemplo de acesso a um arquivo	41
22.2	Exemplo de acesso a um diretório	41
23	A conta root	42
23.1	chmod	43
24	Compactadores	44
24.1	O que fazem os compactadores/descompactadores?	44
24.2	Tipos de compactação	45
24.3	Extensões de arquivos compactados	45
24.4	gzip	46
24.5	zip	46
24.6	unzip	47
24.7	rar	48
24.8	bzip2	50
24.9	rar	51

25 A distribuição Debian GNU/Linux	53
25.1 Porque usar a Debian?	53
25.2 Pacotes existentes na Debian	54
25.3 que é sid/testing/frozen/stable?	54
25.4 Como obter a Debian	55
25.5 Programas de configuração	55
25.6 Sistema de gerenciamento de pacotes	56
26 dpkg	56
26.1 Pacotes	56
26.2 Instalar pacotes	56
26.3 Dependências	56
26.4 Listar pacotes existentes no sistema	56
26.5 Removendo pacotes do sistema	57
26.6 Removendo completamente um pacote	57
26.7 Mostrar descrição do pacote	57
26.8 Procura de pacotes através do nome de um arquivo	57
26.9 tatus do pacote	57
26.10Procurando pacotes com problemas de instalação	57
26.11Mostrando a lista de pacotes do sistema	58
26.12Obtendo uma lista de pacotes para instalar no sistema	58
26.13Configurando pacotes desconfigurados	58
26.14Listando arquivos de um pacote	58
27 apt	59
27.1 O arquivo /etc/apt/sources.list	59
27.2 Endereços de servidores e mirrors nacionais da Debian	60
27.3 Um modelo de arquivo sources.list	60
27.4 O arquivo /etc/apt/apt.conf	61
27.5 Copiando a lista de pacotes disponíveis	61
27.6 Utilizando CDs oficiais/não-oficiais/terceiros com o apt	61
27.7 Instalando novos pacotes	62
27.8 Removendo pacotes instalado	62
27.9 Atualizando sua distribuição	62
27.10Removendo pacotes baixados pelo apt	63
27.11Procurando por pacotes através da descrição	63
27.12Procurando um pacote que contém determinado arquivo	63
27.13Modos eficazes de compilação do código fonte para a Debian	64
27.14Verificando pacotes corrompidos	64
27.15Corrigindo problemas de dependências e outros erros	65

1 O que é Software Livre

Nós mantemos esta definição do Software Livre para mostrar claramente o que deve ser verdadeiro à respeito de um dado programa de software para que ele seja considerado software livre.

”Software Livre” é uma questão de liberdade, não de preço. Para entender o conceito, você deve pensar em ”liberdade de expressão”, não em ”cerveja grátis”.

”Software livre” se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro tipos de liberdade, para os usuários do software:

- * A liberdade de executar o programa, para qualquer propósito.
- * A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades.
- * A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
- * A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

Um programa é software livre se os usuários tem todas essas liberdades. Portanto, você deve ser livre para redistribuir cópias, seja com ou sem modificações, seja de graça ou cobrando uma taxa pela distribuição, para qualquer um em qualquer lugar. Ser livre para fazer essas coisas significa (entre outras coisas) que você não tem que pedir ou pagar pela permissão.

Você deve também ter a liberdade de fazer modificações e usá-las privativamente no seu trabalho ou lazer, sem nem mesmo mencionar que elas existem. Se você publicar as modificações, você não deve ser obrigado a avisar a ninguém em particular, ou de nenhum modo em especial.

A liberdade de utilizar um programa significa a liberdade para qualquer tipo de pessoa física ou jurídica utilizar o software em qualquer tipo de sistema computacional, para qualquer tipo de trabalho ou atividade, sem que seja necessário comunicar ao desenvolvedor ou a qualquer outra entidade em especial.

A liberdade de redistribuir cópias deve incluir formas binárias ou executáveis do programa, assim como o código-fonte, tanto para as versões originais quanto para as modificadas. Está ok se não for possível produzir uma forma binária ou executável (pois algumas linguagens de programação não suportam este recurso), mas deve ser concedida a liberdade de redistribuir essas formas caso seja desenvolvido um meio de criá-las.

De modo que a liberdade de fazer modificações, e de publicar versões aperfeiçoadas, tenha algum significado, deve-se ter acesso ao código-fonte do programa. Portanto, acesso ao código-fonte é uma condição necessária ao software livre.

Copyright (C) 1996, 1997, 1998, 1999, 2000 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA A cópia fiel e a distribuição deste artigo completo é permitida em qualquer meio, desde que esta nota seja preservada.

2 Porque GNU/Linux ?

O movimento Gnu surgiu com a criação do mesmo por Richard Stallman, criou e publicou um termo de licença denominado GNU GPL - General Public License, onde o código fonte do GNU deverá seguir os termos e condições para copiar, distribuir e modificar.



Figura 1: Stallman e Linus, figuras principais do movimento GNU.

Muitos programas surgiram sobre esta nova licença e o movimento ganhava cada vez mais força, mas ainda faltava o núcleo, mais conhecido como kernel ou sistema operacional, uma vez que até o presente momento os programas livres rodavam em sistemas operacionais proprietários, foi aí que surgiu o Linus Torvald, com o seu kernel que veio se chamar mais tarde de linux. Com o surgimento do LINUX, o GNU o adotou como núcleo para uso em conjunto com seus aplicativos, dando origem então ao atual GNU/LINUX, um sistema operacional completo com todas as funcionalidades do já conceituado UNIX, se consagrando como uma plataforma para os mais diversos perfis de usuários. Apesar do Linux depender do GNU para realmente se compor como um sistema operacional, equivocadamente as pessoas chamam este sistema operacional apenas de Linux.

3 O Kernel

O kernel Linux é o centro de toda distribuição Linux. É a fundação sobre a qual é contruída a "terra do usuário" (os programas e ambientes que você usa para interagir com o sistema). O código fonte que entra nele é muito especial – quaisquer bugs nele terão efeito em toda outra parte do sistema, provavelmente destruindo o trabalho do usuário. Como acontecem as mudanças nele? Como são feitas as decisões sobre a direção que o kernel tomará?

O kernel Linux é dividido em muitas partes. A maior divisão é a do núcleo do código do núcleo e os drivers de dispositivo. O núcleo do código é dividido em subsistemas. Cada subsistema contém código específico de arquitetura, mas a maioria dos drivers de dispositivo não são específicos de arquitetura. Pelo fato de o kernel ser dividido dessa forma, muitas pessoas podem compartilhar o trabalho de manter o kernel olhando por partes específicas. Cada driver de dispositivo tem uma pessoa que é responsável por ele; cada subsistema do núcleo do kernel tem um mantenedor que monitora seu desenvolvimento.

Os vários programadores trabalhando para manter o kernel estão em uma hierarquia destacada. Programadores experientes, que contribuíram mais e se voluntariaram para essa responsabilidade, são designados para trabalhos mais importantes. A autoridade máxima

sobre todos esses programadores é o mantenedor oficial da árvore. Durante a produção desse texto, eles são: Linus Torvalds para o 2.4.x, Alan Cox para 2.2.x e David Weinehall para o 2.0.x. O mantenedor oficial da árvore é como o diretor de uma peça – eles designam tarefas para os outros programadores, dando, as vezes, exemplos da direção que eles desejam que o código tome. Eles trabalham para fazer o kernel atingir sua visão do que ele deveria ser.

Você pode estar pensando como um mantenedor de árvore decide sobre a direção para o kernel; de onde sua visão vem. Ela vem dos usuários. Os usuários pedem por funções, reportam erros que interferem em seu trabalho e geralmente provêm um retorno sobre o que o kernel está fazendo para eles. Baseados nisso e em motivação pessoal, os três mantenedores decidem em um conjunto de objetivos concretos para o kernel. Essa é sua visão.

A direção do desenvolvimento do kernel não é estática através da vida de um braço específico do código. Quando as necessidades dos usuários mudam, os objetivos para a árvore mudam. A discussão dessas mudanças se dá na lista de email do kernel Linux. A lista recebe quase todas as discussões públicas entre os vários desenvolvedores assim como relatórios de erros gerais, pedidos de funções e patches. Com no mínimo 200 mensagens por dia ela não é para os fracos do coração. A visão para a árvore 2.4.x envolveu consertos e melhoras ao subsistema de VM, expansão de escalabilidade em ambientes corporativos, aumento a portabilidade e melhora de eficiência. Essas mudanças foram motivadas pelas foram motivadas pela avaliação humilhante da mindcraft em 1999 assim como o desejo de dar aos usuários corporativos do Linux a habilidade de usar toda a sua capacidade com o Linux. Acima do curso do desenvolvimento da árvore (a série 2.3.x do kernel), mudanças não relacionadas diretamente aos objetivos do Linus para a árvore foram aceitos porque eles limpavam código, endereçavam código mal-comportado ou pavimentavam o caminho para futuras limpezas de código.

No mesmo estágio de desenvolvimento, alguns conflitos apareceram também. Andrea Arcangeli e Rik van Riel tiveram idéias sobre como o subsistema de memória virtual deveria ser desenhado. Alan queria ver qual das duas visões na melhora do VM era mais efetiva no seu braço AC da série 2.4.0-test. A solução foi fazer com que ambos enviassem seus patches e lançar duas versões do próximo kernel AC test. O mais estável foi então selecionado e incluído no outro lançamento. Esse processo foi repetido algumas vezes para dar uma boa chance de competição entre os programadores. O vencedor foi o algoritmo de Rik que Linus incluiu no patch da árvore AC do Alan e ele se tornou o patch oficial de VM que entrou nos lançamentos do 2.4.x estável.

O kernel Linux é um dos maiores projetos Open Source que existem. O kernel 2.4.1 é um arquivo tar de 24 megabytes; expandido, o fonte tem quase 100 megabytes em tamanho. É um grande crescimento sobre o fonte original lançado na usenet em 1991 pelo Linus – esse kernel tinha a metade de um megabyte quando descomprimido.

”Arreando” as idéias dos desenvolvedores e usuários, o processo de desenvolvimento do kernel Linux permitiu a ele ultrapassar muitos sistemas operacionais tradicionais em performance e confiabilidade. Esse processo, que enfatiza código confiável e o correto acima dos sentimentos e políticas de indivíduos é uma parte essencial do sucesso do Linux.

- Por Dylan Griffiths - Traduzido por Gustavo Noronha Silva - kov@debian.org

4 A grande Família

O Linux juntamente com o movimento de software livre GNU, são mantidos e contruídos por pessoas de toda parte do mundo, imagine você, um software sendo testado diariamente

por milhares de usuários cada um com uma forma diferente de interagir com o mesmo, é inevitável que apareça bug e que o mesmo seja corrigido logo em seguida.

Tudo isso leva a uma qualidade muito alta dos softwares, como dizia Linus Torvads: A olhos de muitos não existe erros intrinsecos em programas de computadores. Na internet o espaço desaparece e o que temos é somente o tempo, pessoas de várias partes do mundo se comunicam para escolherem que rumo tomar em relação a um software livre qualquer. Uma empresa por maior que seja jamais terá em mãos um força desse tamanho, porque ela não pode contar com a comunidade que usa seu programas, para isso ela teria que abrir os cadeados que prendem o código fonte dos programas. Devido a essa grande variedade de pessoas de culturas diferentes e visões diferentes, surgem a diversidade encontrada no mundo GNU/Linux, indo contra a monocultura estabelecida, tudo você pode, e tem o domínio e a liberdade de fazer o que bem quiser com os seu programas e seu sistema operacional. As forma de participar neste grande movimento são muitas (caso queira é claro), desde pessoas que não sabe quase nada de computadores até pessoas que trabalham a anos com informática em algum empresa de grande porte. Artistas que colaboram com a criação de ícones, músicos para a criação de arranjos para algum Ambiente Gráfico voltado para desktop, principalmente dicas de usuários comuns ou finais como são conhecidos, para melhora a logística por exemplo de uma gerenciador de arquivos, de forma que se possa trabalhar mais rapido e exato com o mesmo, em tudo por tudo, as possibilidade de participar e as formas como as pessoas participam é muito diversificado.

5 Conceitos Básicos

5.1 Ambiente Modo Texto e Modo Gráfico

5.1.1 Modo Texto

Se vocês já utilizaram um computador uma vez em suas vidas devem estar acostumados com o sistema de arrastar e clicar através do mouse. Através desse sistema vocês passaram suas instruções para o sistema operacional realizar tarefas como abrir um programa, copiar ou apagar arquivos, desligar o computador, etc. Essa forma de utilizar o computador pode até ser cômoda mas com certeza não é prática. Você também poderá trabalhar no Linux desta forma

No modo texto, temos acesso a essência do sistema operacional linux, através de uma "shell". Sua interação com o sistema não passa de um **Prompt** onde você declara e envia ordens a serem executadas pelo o kernel. Parece muito frio e sem graça usa o linux assim dessa maneira, mas devo dizer que a **Shell** do linux é extremamente rica em comandos e possibilidades, e recursos pra simplificar a vida do usuário. Para muitos com certeza vai ser um trauma usar a shell pela primeira vez, mas é incrível o que podemos fazer com um pouco de treino usando uma shell simples, coisa que você perderia horas usando o seu querido mouse no seu ambiente gráfico predileto. Para os usuários mais experiêntes (você não quer ser um futuramente ?) é imprecidível o uso do shell, tanto em modo gráfico como somente em texto onde é tudo o que você tem. Em outras palavras, numa abordagem visual, o *modo texto* seria somentes a tela escura onde você troca informação com o sistema a respeito do seus dados

```

top - 22:13:43 up 1:05, 3 users, load average: 0.96, 0.47, 0.36
Tasks: 68 total, 1 running, 67 sleeping, 0 stopped, 0 zombie
Cpu(s): 7.9% user, 6.3% system, 0.0% nice, 85.8% idle
Mem: 117900k total, 113804k used, 4096k free, 1804k buffers
Swap: 192740k total, 7992k used, 184748k free, 39724k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  754 merovigi  16   0  1024  1024  820  R   5.0   0.9   0:01.40 top
  462 root      15  -10 29484  19m 3028  S  -2.5  16.9  2:53.91 XFree86
  527 merovigi  16   0  1548  1388 1252  S   2.2   1.2   0:21.44 wmusic
  586 merovigi  11   0  5348  4800 2760  S   1.6   4.1   0:12.50 xmms
  755 merovigi  10   0  6632  6632 5428  S   1.6   5.6   0:01.53 screenshot
  748 merovigi  10   0  5348  4800 2760  S   1.3   4.1   0:00.94 xmms
  525 merovigi  10   0  1120  1016  956  S   0.3   0.9   0:01.11 wmpd
  594 merovigi   9   0  8212  6592 5312  S   0.3   5.6   0:00.46 xchat
    1 root       8   0    492   464   440  S   0.0   0.4   0:04.40 init
    2 root       9   0     0     0     0  S   0.0   0.0   0:00.27 keventd
    3 root      19  19     0     0     0  S   0.0   0.0   0:00.06 ksoftirqd_CPU0
    4 root       9   0     0     0     0  S   0.0   0.0   0:00.68 kswapd
    5 root       9   0     0     0     0  S   0.0   0.0   0:00.00 bdflush
    6 root       9   0     0     0     0  S   0.0   0.0   0:00.00 kupdated
    7 root       9   0     0     0     0  S   0.0   0.0   0:00.00 i2cevt
  109 root       9   0     0     0     0  S   0.0   0.0   0:00.11 kjournald
  196 root       9   0    624   584   508  S   0.0   0.5   0:00.00 dhclient

```

Figura 2: Uma Shell aberta em modo gráfico, mais conhecida como terminal virtual

5.1.2 Modo Gráfico

No linux o ambiente gráfico é independente do sistema operacional, ambos são desenvolvidos separadamente, você pode muito bem usa o Linux sem o ambiente gráfico, o que é normal, mas caso queiro,

O sistema X Window (ou simplesmente X) é um ambiente de trabalho gráfico utilizado em quase todos os desktops Unix existentes no mundo. Foi desenvolvido inicialmente pelo MIT (Massachusetts Institute of Technology), sendo o projeto posteriormente repassado para um conjunto de empresas denominado The X Consortium e mantidos por eles desde então.

O X é constituído de um programa chamado servidor gráfico, responsável por desenhar os elementos da tela e de uma série de bibliotecas auxiliares. Os programas utilitários são denominados clientes. O browser netscape é um exemplo de programa cliente.

No Linux, o X Window é implementado por um conjunto de programas e bibliotecas denominado XFree86. O XFree86 é totalmente gratuito e dá suporte às mais diversas placas de vídeo disponíveis no mercado.

O X Window traz para o usuário Unix o conforto de trabalhar em um ambiente gráfico, com ícones, editores de texto e imagem etc e possibilita, entre outras coisas.

É com o ambiente gráfico que torna possível o uso do gnome ou kde entre outro gerenciadores de janelas, que só fazem sentido com o uso do X.

Agostinho de M. Brito Jr. 2000-09-02

6 Ambientes Gráficos

Aqui faremos uma breve introdução do que é cada ambiente gráfico mais usados no linux, é claro que durante o curso não poderemos ter acesso a todos, sendo que usaremos o por default o gnome ou então o kde, que são mais simples e intuitivos.



6.1 Kde - The K Desktop Environment

O KDE é um poderoso ambiente gráfico de Código Aberto para estações Linux. Ele combina facilidade de uso, com funcionalidade e fantástico design gráfico com toda a superioridade tecnológica que o UNIX oferece.

É uma Suíte robusta com uma base sólida para um número cada vez maior de aplicações para estações Linux. Ele possui um modelo de desenvolvimento de alta qualidade, permitindo que novas aplicações sejam criadas de modo rápido e eficiente. Tem um ótimo gerenciador de arquivos conhecido como **Konqueror** e vários aplicativos interessante que pode ser usado com muita facilidade pelo o usuario inicial, não é a toa que é muito usado pelos iniciantes no mundo linux, por outro lado é um dos ambientes gráficos mais pesados que temos para o X.

6.2 Gnome - Computing made easy



Ambiente gráfico gratuito e open source destinado aos utilizadores de plataformas Unix, é hoje em dia incluído em muitas das distribuições GNU/Linux e BSD.

O projeto de GNOME nasceu como um esforço para criar um ambiente desktop inteiramente livre para sistemas livres.

No começo, o objetivo principal do GNOME foi fornecer uma suite amigável ao usuário de aplicações e um desktop fácil-de-usar.

Como a maioria dos programas GNU, GNOME foi projetado para funcionar em todas as variações modernas de sistemas semelhantes ao Unix.

Como o projeto tem progredido nos meses passados, os objetivos do projeto foram estendidos para lidar com um número de problemas na infraestrutura existente no Unix. O projeto GNOME é como um guarda-chuva, os componentes principais do GNOME são: O desktop do GNOME, A plataforma do desenvolvimento do GNOME e O Office do GNOME.

6.3 Window Maker - Wmaker - X11 Window Manager

O Window Maker é um elegante gerenciador de janelas X11 que emula o desktop do OpenStep(tm) apresentando várias melhorias. Tem um conceito muito diferente dos gerenciadores de janelas convencionais, que no caso até agora apresentado não passa de ambientes voltados para desktops. O window maker traz várias inovações e um visual muito bonito apesar



de ser simples e leve, usa os lindo **Dockapps** tornando-o muito mais elegante, é um dos gerenciadores gráficos leves mais bonitos que temos. Vale apenas ressaltar que é totalmente desenvolvido pelo brasileiro Alfredo Kojima.

6.4 FluxBox - Fluxbox - X11 Window Manager

Baseado no Blackbox, o Fluxbox possui algumas características adicionais, entre as quais destacam-se o suporte a dock de aplicativos (incluindo os do KDE e Gnome), Xinerama (possibilidade de uso em dois monitores ao mesmo tempo), Window Tabs e teclas de atalhos configuráveis. É com certeza o mais leve dos gerenciadores gráficos e muito bonito visualmente, o tamanho do arquivo fonte é de apenas 500K, sem abrir mão da beleza.

7 Entrando e Saindo do Sistema

7.1 Login

Para que você possa ter acesso ao sistema, é necessário **logar-se**, e para isso é necessário ter uma conta de usuário. O ato de fazer um *login* em uma máquina, significa entrar no sistema, seja como usuário ou como administrador.

7.2 Logout

Nada mais é que encerrar uma conta aberta.

8 A Hierarquia

8.1 O Root

Por se tratar de um sistema multiusuário baseado no Unix, o linux também tem uma hierarquia de acesso ao sistema. O usuário que tem acesso a todos os outros e administra todo o sistema tendo acesso para fazer o que bem quiser no sistema, como a fiscalização das contas de usuário é o **Root**, geralmente quem tem a senha e o login de root é o responsável pela a integridade do sistema. Deve-se tomar muito cuidado ao utilizar o sistema logado como root, porque qualquer coisa feita sem pensar pode apagar ou modificar um arquivo de configuração importante que afeta de forma drástica o funcionamento de todo o sistema operacional.

8.2 O Usuário

Na hierarquia é o mais ordinário, são as pessoas que podem acessar o sistema mas somente isso, não tendo o controle do mesmo. O máximo que pode é modificar e gerenciar a sua

conta com os respectivos arquivos encontrados e configurações. É muito mais seguro você usar o Linux logado como usuário comum, isso pode evitar muito aborrecimentos, por isso que se aconselha a criar uma conta de usuário quando se instala o sistema GNU/Linux .

9 X Window (ambiente gráfico)

9.1 O servidor X

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. É justamente o que gera a parte visual de um programa, aquele que você vê pelo monitor, a janela, os ícones e etc. Ele se comunica com os programas cliente através de diversos métodos de comunicação.

9.2 A organização do ambiente gráfico X Window

Em geral o ambiente gráfico X Window é dividido da seguinte forma:

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.

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.

10 Diretórios

Como já falei várias vezes que a grande "sacada" do linux é **terminal virtual** ou **shell**. Então o nosso curso será com ênfase a manipulação de arquivos, diretórios e programas usando o terminal. Isso porque para qualquer usuário de linux, não é muita vantagem gastar nosso tempo precioso aprendendo a usar um ambiente gráfico, sendo que basta "fuçar" para aprender o que quiser, se tratando que os mais conhecidos são os mais interativos.

10.1 Diretório

Diretório é o local utilizado para armazenar conjuntos arquivos para melhor organização e localização. O diretório, como o arquivo, também é **Case Sensitive** (diretório /teste é completamente diferente do diretório /Teste).

Não podem existir dois arquivos com o mesmo nome em um diretório, ou um sub-diretório com um mesmo nome de um arquivo em um mesmo diretório.

Um diretório nos sistemas Linux/UNIX são especificados por uma /.

10.2 Diretório Raíz

Este é o diretório principal do sistema. Dentro dele estão todos os diretórios do sistema. O diretório Raíz é representado por uma /, assim se você digitar o comando `cd /` você estará acessando este diretório.

Nele estão localizados outros diretórios como o `/bin`, `/sbin`, `/usr`, `/usr/local`, `/mnt`, `/tmp`, `/var`, `/home`, etc. Estes são chamados de sub-diretórios pois estão dentro do diretório /. A estrutura de diretórios e sub-diretórios pode ser identificada da seguinte maneira:

```

          * /           * /bin           * /sbin           * /usr           * /usr/local
* /mnt       * /tmp       * /var           * /home

```

A estrutura de diretórios também é chamada de **Árvore de Diretórios** porque é parecida com uma árvore de cabeça para baixo. Cada diretório do sistema tem seus respectivos arquivos que são armazenados conforme regras definidas pela **FHS** (FileSystem Hierarchy Standard - Hierarquia Padrão do Sistema de Arquivos) versão 2.0, definindo que tipo de arquivo deve ser armazenado em cada diretório.

10.3 Diretório home

Também chamado de diretório de usuário. Em sistemas GNU/Linux cada usuário (inclusive o root) possui seu próprio diretório onde poderá armazenar seus programas e arquivos pessoais.

Este diretório está localizado em `/home/[login]`, neste caso se o seu login for "joao" o seu diretório home será `/home/joao`. O diretório home também é identificado por um `(til)`, você pode digitar tanto o comando `ls /home/joao` como `ls ~` para listar os arquivos de seu diretório home. O diretório home do usuário root (na maioria das distribuições GNU/Linux) está localizado em `/root`.

10.3.1 Diretório Superior

O diretório superior (Upper Directory) é identificado por `..` (2 pontos). Caso estiver no diretório `/usr/local` e quiser listar os arquivos do diretório `/usr` você pode digitar, `ls ..`. Este recurso também pode ser usado para copiar, mover arquivos/diretórios, etc.

10.3.2 Diretório Anterior

O diretório anterior é identificado por `-`. É útil para retornar ao último diretório usado.

Se estive no diretório `/usr/local` e digitar `cd /lib`, você pode retornar facilmente para o diretório `/usr/local` usando `cd -`.

10.3.3 Caminho na estrutura de diretórios

São os diretórios que teremos que percorrer até chegar no arquivo ou diretório que procuramos. Se desejar ver o arquivo `/usr/doc/copyright/GPL` você tem duas opções:

1. Mudar o diretório padrão para `/usr/doc/copyright` com o comando `cd /usr/doc/copyright` e usar o comando `cat GPL`
2. Usar o comando "cat" especificando o caminho completo na estrutura de diretórios e o nome de arquivo: `cat /usr/doc/copyright/GPL`.

As duas soluções acima permitem que você veja o arquivo GPL. A diferença entre as duas é a seguinte:

* Na primeira, você muda o diretório padrão para `/usr/doc/copyright` (confira digitando `pwd`) e depois o comando `cat GPL`. Você pode ver os arquivos de `/usr/doc/copyright` com o comando `ls`. `/usr/doc/copyright` é o caminho de diretório que devemos percorrer para chegar até o arquivo GPL.

* Na segunda, é digitado o caminho completo para o `cat` localizar o arquivo GPL: `cat /usr/doc/copyright/GPL`. Neste caso, você continuará no diretório padrão (confira digitando `pwd`). Digitando `ls`, os arquivos do diretório atual serão listados.

O caminho de diretórios é necessário para dizer ao sistema operacional onde encontrar um arquivo na "árvore" de diretórios.

10.3.4 Nomeando Arquivos e Diretórios

No GNU/Linux, os arquivos e diretórios podem ter o tamanho de até 255 letras. Você pode identificá-lo com uma extensão (um conjunto de letras separadas do nome do arquivo por um ".").

Os programas executáveis do GNU/Linux, ao contrário dos programas de DOS e Windows, não são executados a partir de extensões `.exe`, `.com` ou `.bat`. O GNU/Linux (como todos os sistemas POSIX) usa a permissão de execução de arquivo para identificar se um arquivo pode ou não ser executado.

No exemplo anterior, nosso trabalho de história pode ser identificado mais facilmente caso fosse gravado com o nome `trabalho.text` ou `trabalho.txt`. Também é permitido gravar o arquivo com o nome `Trabalho de Historia.txt` mas não é recomendado gravar nomes de arquivos e diretórios com espaços. Porque será necessário colocar o nome do arquivo entre "aspas" para acessá-lo (por exemplo, `cat "Trabalho de Historia.txt"`). Ao invés de usar espaços, prefira capitalizar o arquivo (usar letras maiúsculas e minúsculas para identificá-lo): `TrabalhodeHistoria.txt`.

11 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 separá-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`.

11.0.5 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 um **jogo da velha** (tralha), e o aviso de comando de usuários é identificado pelo símbolo o símbolo cifrão. 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 `Back Space` ("`<`") 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 início 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.

11.0.6 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 completção dos nomes. Isto é feito pressionando-se a tecla TAB . Por exemplo, se digitar "ls tes" e pressionar `␣`, o Bash localizará todos os arquivos que iniciam com "tes" e completará o restante do nome. Caso a completção de nomes encontre mais do que uma expressão que satisfaça a pesquisa, ou nenhuma, é emitido um beep. Se você apertar novamente a tecla TAB imediatamente depois do beep, o interpretador de comandos irá listar as diversas possibilidades que satisfazem a pesquisa, para que você possa escolher a que lhe interessa. A completção de nomes funciona sem problemas para comandos internos.

Exemplo: `ech (pressione TAB). ls /vm (pressione TAB)`

11.0.7 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>.

12 Comandos para manipulação de diretório

Abaixo comandos úteis para a manipulação de diretórios.

12.1 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.

-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.

-R

Lista diretórios e sub-diretórios recursivamente.

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.

12.2 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).

12.3 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).

12.4 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 papéis (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 [ caminho/diretório ]
```

onde:

caminho

Caminho onde o diretório será criado.

diretório

Nome do diretório que será criado.

Exemplo:

```
"mkdir /tmp/teste".
```

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

12.5 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` .

13 Comandos para manipulação de Arquivos

Abaixo, comandos utilizados para manipulação de arquivos.

13.1 cat

Mostra o conteúdo de um arquivo binário ou texto.

```
cat [ diretório/arquivo ]
```

diretório/arquivo

Localização do arquivo que deseja visualizar o conteúdo. 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

13.2 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 .

* rm -f --arquivo- - Remove o arquivo de nome --arquivo- .

13.3 cp

Copia arquivos.

cp [opções] [origem] [destino]

onde:

origem

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

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 .

13.4 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.

-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.

14 Comandos Diversos

Comandos de uso diversos no sistema.

14.1 clear

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

```
clear
```

14.2 date

Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora. Muitos programas do sistema, arquivos de registro (log) e tarefas agendadas funcionam com base na data e hora fornecidas pelo sistema, assim esteja consciente das modificações que a data/hora pode trazer a estes programas (principalmente em se tratando de uma rede com muitos usuários) .

```
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.

14.3 df

Mostra o espaço livre/ocupado de cada partição.

```
df [ opções ]
```

onde:

```
opções
```

-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). Não lista sistemas de arquivos do tipo tipo .

Exemplos: df , df -H .

14.4 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` .

14.5 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.

-h, -human

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

-m Mostra o espaço ocupado em Mbytes.

Exemplo: du -h , du -a .

14.6 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.

-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 raiz e sub-diretórios um arquivo/diretório chamado grep .

* find / -name grep -maxdepth 3 - Procura no diretório raiz 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).

* find / -mmin 10 - Procura no diretório raiz e sub-diretórios um arquivo que foi modificado há 10 minutos atrás.

* find / -links 4 - Procura no diretório raiz e sub-diretórios, todos os arquivos que possuem 4 links como referência.

14.7 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 .

14.8 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

Foram ignorados, por fugir ai conteúdo abordado.

Exemplos: grep "capitulo"texto.txt , ps ax—grep inetd , grep "capitulo"texto.txt -A 2 -B 2 .

14.9 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 ou espaço para continuar avançando no arquivo sendo visualizado. Para sair do more pressione q .

```
more [ arquivo ]
```

Onde: arquivo É o arquivo que será paginado.

Para visualizar diretamente arquivos texto compactados pelo gzip .gz use o comando zmore .

Exemplos: more /etc/passwd , cat /etc/passwd—more .

14.10 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

14.11 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 .

14.12 uptime

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

uptime

14.13 dmesg

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

dmesg—less

14.14 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.

14.15 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 sessão 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.

14.16 uname

Retorna o nome e versão do kernel atual.

uname

14.17 reboot

Reinicia o computador.

14.18 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 quando é usada a opção -a

pelos usuários cadastrados no arquivo `/etc/shutdown.allow` que estejam logados no console virtual do sistema.

```
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 exceção 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 reinício/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.

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 .

* shutdown -t 30 -r 20 - Reinicia o sistema após 20 minutos, espera 30 segundos após o sinal de término para enviar o sinal KILL a todos os programas abertos.

14.19 diff

Compara dois arquivos e mostra as diferenças entre eles. O comando diff é usado somente para a comparação de arquivos em formato texto. As diferenças encontradas podem ser redirecionadas para um arquivo que poderá ser usado pelo comando patch para aplicar as alterações em um arquivo que não contém as diferenças. Isto é útil para grandes textos porque é possível copiar somente as modificações (geradas através do diff, que são muito pequenas) e aplicar no arquivo para atualiza-lo (através do patch) ao invés de copiar a nova versão. Este é um sistema de atualização muito usado na atualização dos código fonte do kernel do GNU/Linux .

```
diff [ diretório1/arquivo1 ] [ diretório2/arquivo2 ] [ opções ]
```

Opções:

diretório1/arquivo1 diretório2/arquivo2

Arquivos /diretórios que serão comparados. Normalmente é usado como primeiro arquivo/diretório o mais antigo e o mais novo como segundo.

opções

-lines [num]

Gera a diferença com [num] linhas de contexto. Por padrão o diff gera um arquivo com 2 linhas que é o mínimo necessário para o correto funcionamento do patch .

-a

Compara os dois arquivos como arquivos texto.

-b

Ignora espaços em branco como diferenças.

-B

Ignora linhas em branco inseridas ou apagadas nos arquivos.

-i

Ignora diferenças entre maiúsculas e minúsculas nos arquivos.

-H

Usa análise heurística para verificar os arquivos.

-N

Em uma comparação de diretórios, se o arquivo apenas existe em um diretório, trata-o como presente mas vazio no outro diretório.

-P

Em uma comparação de diretórios, se o arquivos apenas existe no segundo diretório, trata-o como presente mas vazio no primeiro diretório.

-q

Mostra somente se os dois arquivos possuem diferenças. Não mostra as diferenças entre eles.

-r

Compara diretórios e sub-diretórios existentes.

-S [nome]

Inicia a comparação de diretórios pelo arquivo [nome]. É útil quando cancelamos uma comparação.

-t

Aumenta a tabulação das diferenças encontradas.

-u

Usa o formato de comparação unificado. Use o comando zdiff para comparar diretamente arquivos compactados pelo utilitário gzip

Use o comando sdiff para visualizar as linhas diferentes entre os dois arquivos em formato texto simples.

Exemplo:

* diff texto.txt texto1.txt - Compara o arquivo texto.txt com texto1.txt e exibe suas diferenças na tela.

* diff -Bu texto.txt texto1.txt - Compara o arquivo texto.txt com texto1.txt ignorando linhas em branco diferentes entre os dois arquivos e usando o formato unificado.

* diff texto.txt texto1.txt ;texto.diff - Compara o arquivo texto.txt com texto1.txt e gera um arquivo chamado texto.diff contendo a diferença entre eles. Este arquivo poderá ser usado pelo patch para aplicar as diferenças existente entre os dois no arquivo texto.txt .

* `diff -r /usr/src/linux-2.2.13 /usr/src/linux-2.2.14 >patch-2.2.14.diff` - Compara o diretório e sub-diretórios linux-2.2.13 e linux-2.2.14 e grava as diferenças entre eles no arquivo patch-2.2.14.diff .

14.20 patch

Atualiza arquivos texto através das diferenças geradas pelo comando diff .

`patch [opções] [arquivo.diff]` ou `patch [opções] -i [arquivo.diff]`

Onde:

arquivo.diff

Arquivo contendo as diferenças geradas pelo comando diff .

opções

-p [num]

Nível do diretório onde o patch será aplicado, se igual a 0, o patch assume que os arquivos que serão atualizados estão no diretório atual, se 1, assume que os arquivos que serão atualizado estão no diretório acima (..), se 2, 2 diretórios acima ...

-b

Cria cópias de segurança dos arquivos originais ao aplica o patch.

-binary

Lê e grava arquivo usando modo binário.

-d [dir]

Muda para o diretório [dir] antes de aplica o patch.

-E

Remove arquivos vazios após a aplicação do patch.

-n

Interpreta o arquivo de patch como um .diff normal.

-N

Não desfaz patches já aplicados.

-s

Não mostra mensagens de erro.

-u

Interpreta o patch em formato unificado. As diferenças são aplicadas em arquivos originais gerados pelo comando diff . É importante entender os comandos patch e diff pois são comandos muito utilizados para desenvolvimento feito por equipes de pessoas.

Exemplo:

* `patch -p0 <texto.diff` - Aplica as diferenças contidas no arquivo texto.diff nos arquivos originais.

* `patch -p0 texto.txt texto.diff` - Aplica as diferenças contidas no arquivo texto.diff nos arquivos originais. Faz a mesma coisa que o comando anterior.

15 Redirecionamentos e Pipe

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

15.1 >

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).

15.2 >>

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`.

15.3 <

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 `<`).

15.4 <<

Este redirecionamento serve principalmente para marcar o fim de exibição de um bloco. Este é especialmente usado em conjunto com o comando `cat`, mas também tem outras aplicações. Por exemplo:

```
cat << final
```

este arquivo será mostrado até que a palavra `final` seja localizada no início da linha `final`.

15.5 |

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.

15.6 Diferença entre `|` e `>`

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` .

16 Discos e Partições

Este capítulo traz explicações de como manipular discos rígidos e partições no sistema GNU/Linux e como acessar seus discos de CD-ROM e partições DOS , Windows 95/98 no GNU/Linux .

16.1 Partições

São divisões existentes no disco rígido que marcam onde começa onde termina um sistema de arquivos. Por causa destas divisões, nós podemos usar mais de um sistema operacional no mesmo computador (como o GNU/Linux , Windows e DOS), ou dividir o disco rígido em uma ou mais partes para ser usado por um único sistema operacional.

16.2 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.

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.

16.3 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, xt=XT).
|
| Diretório onde são armazenados os dispositivos existentes no sistema.

```

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.

16.4 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 usando o Extended File System versão 2 (a mais comum).

* ext3 - Para partições GNU/Linux usando o Extended File System versão 3, com suporte a journaling.

* reiserfs - Para partições reiserfs, com suporte a journaling.

* 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/mstab . 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` .

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 /` .

16.5 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
<code>/dev/hda1</code>	<code>/</code>	<code>ext2</code>	<code>defaults</code>	<code>0</code>	<code>1</code>
<code>/dev/hda2</code>	<code>/boot</code>	<code>ext2</code>	<code>defaults</code>	<code>0</code>	<code>2</code>
<code>/dev/hda3</code>	<code>/dos</code>	<code>msdos</code>	<code>defaults,noauto,rw</code>	<code>0</code>	<code>0</code>
<code>/dev/hdg</code>	<code>/cdrom</code>	<code>iso9660</code>	<code>defaults,noauto</code>	<code>0</code>	<code>0</code>

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).

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 raiz que deverá ser verificado primeiro é o raiz `"/`.

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`.

16.6 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.

17 Execução de programas

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

17.1 Executando um comando/programa

No aviso de comando (root) ou (usuário), digite o nome do comando e tecla 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`.

17.2 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 executá-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`.

17.3 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.

18 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`

18.1 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.

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 .

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 .

18.2 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.

19 Controle de execução de processos

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

19.1 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.

19.2 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.

19.3 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 . Um processo interrompido pode ser finalizado usando-se o comando kill %[num] , onde [num] é o número do processo obtido pelo comando jobs .

19.4 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 .

19.5 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 .

19.6 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`. Também pode ser o número após o sinal de `%` obtido pelo comando `jobs` para matar uma tarefa interrompida.

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 terminá-lo ou destruí-lo. Você pode verificar se o processo foi finalizado através do comando `ps`.

Exemplo: `kill 500`, `kill -9 500`, `kill`

19.7 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`

20 Fechando um programa quando não se sabe como sair

Muitas vezes quando se está iniciando no GNU/Linux você pode executar um programa e talvez não saber como fechá-lo. Este capítulo do guia pretende ajudá-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 finalizá-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> + <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>`. Frequentemente 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`. Para detalhes de como encontrar a ajuda dos programas.

20.1 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.

21 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 irreversíveis 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. Não se preocupe, também coloquei exemplos para ajudá-lo a entender o sistema de permissões de acesso do ambiente GNU/Linux .

21.1 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. Para detalhes de como incluir o usuário em mais grupos , Adicionando um novo grupo a um usuário .

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.

21.2 Tipos de Permissões de acesso

Quanto aos tipos de permissões que se aplicam ao dono , grupo e outros usuários , temos 3 permissões básicas:

* **r** - Permissão de leitura para arquivos. Caso for um diretório, permite listar seu conteúdo (através do comando `ls` , por exemplo).

* **w** - Permissão de gravação para arquivos. Caso for um diretório, permite a gravação de arquivos ou outros diretórios dentro dele. Para que um arquivo/diretório possa ser apagado, é necessário o acesso a gravação.

* **x** - Permite executar um arquivo (caso seja um programa executável). Caso seja um diretório, permite que seja acessado através do comando `cd` . As permissões de acesso a um arquivo/diretório podem ser visualizadas com o uso do comando `ls -la` . As 3 letras (rwx) são agrupadas da seguinte forma:

```
-rwxrwxrwx  gleydson  users  teste
```

Virou uma bagunça não? Vou explicar cada parte para entender o que quer dizer as 10 letras acima (da esquerda para a direita):

* A primeira letra diz qual é o tipo do arquivo. Caso tiver um "d" é um diretório, um "l" um link a um arquivo no sistema , um "-" quer dizer que é um arquivo comum, etc.

* Da segunda a quarta letra (rwx) dizem qual é a permissão de acesso ao dono do arquivo. Neste caso gleydson ele tem a permissão de ler (r - read), gravar (w - write) e executar (x - execute) o arquivo teste .

* Da quinta a sétima letra (rwx) diz qual é a permissão de acesso ao grupo do arquivo. Neste caso todos os usuários que pertencem ao grupo users tem a permissão de ler (r), gravar (w), e também executar (x) o arquivo teste .

* Da oitava a décima letra (rwx) diz qual é a permissão de acesso para os outros usuários . Neste caso todos os usuários que não são donos do arquivo teste tem a permissão para ler, gravar e executar o programa.

21.3 Etapas para acesso a um arquivo/diretório

O acesso a um arquivo/diretório é feito verificando primeiro se o usuário que acessará o arquivo é o seu dono , caso seja, as permissões de dono do arquivo são aplicadas. Caso não seja o dono do arquivo/diretório, é verificado se ele pertence ao grupo correspondente, caso pertença, as permissões do grupo são aplicadas. Caso não pertença ao grupo , são verificadas as permissões de acesso para os outros usuários que não são donos e não pertencem ao grupo correspondente ao arquivo/diretório.

Após verificar aonde o usuário se encaixa nas permissões de acesso do arquivo (se ele é o dono , pertence ao grupo , ou outros usuários), é verificado se ele terá permissão acesso para o que deseja fazer (ler, gravar ou executar o arquivo), caso não tenha, o acesso é negado, mostrando uma mensagem do tipo: "Permission denied" (permissão negada).

O que isto quer dizer é que mesmo que você seja o dono do arquivo e definir o acesso do dono (através do comando `chmod`) como somente leitura (r) mas o acesso dos outros usuários como leitura e gravação, você somente poderá ler este arquivo mas os outros usuários poderão ler/grava-lo.

As permissões de acesso (leitura, gravação, execução) para donos, grupos e outros usuários são independentes, permitindo assim um nível de acesso diferenciado. Lembre-se: Somente o dono pode modificar um arquivo/diretório!

22 Exemplos práticos de permissões de acesso

22.1 Exemplo de acesso a um arquivo

Abaixo um exemplo e explicação das permissões de acesso a um arquivo no GNU/Linux (obtido com o comando `ls -la`, explicarei passo a passo cada parte:

```
-rwxr-xr-- 1 gleydson user 8192 nov 4 16:00 teste
```

-rwxr-xr--

Estas são as permissões de acesso ao arquivo `teste`. Um conjunto de 10 letras que especificam o tipo do arquivo, permissão do dono do arquivo, grupo do arquivo e outros usuários.

Veja a explicação detalhada sobre cada uma abaixo:

- rwxr-xr--

A primeira letra (do conjunto das 10 letras) determina o tipo do arquivos. Se a letra for um `d` é um diretório, e você poderá acessa-lo usando o comando `cd`. Caso for um `l` é um link simbólico para algum arquivo ou diretório no sistema. Um `-` significa que é um arquivo normal.

- rwx r-xr--

Estas 3 letras (da segunda a quarta do conjunto das 10 letras) são as permissões de acesso do dono do arquivo `teste`. O dono (neste caso `gleydson`) tem a permissão para ler (`r`), gravar (`w`) e executar (`x`) o arquivo `teste`.

-rwx r-x r--

Estas 3 letras (da quinta a sétima do conjunto das 10 letras) são as permissões de acesso dos usuários que pertencem ao grupo `user` do arquivo `teste`. Os usuários que pertencem ao grupo `user` tem a permissão somente para ler (`r`) e executar (`x`) o arquivo `teste` não podendo modifica-lo ou apaga-lo. **-rwxr-x r--**

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que não são donos do arquivo `teste` e que não pertencem ao grupo `user`. Neste caso, estas pessoas somente terão a permissão para ver o conteúdo do arquivo `teste`.

`gleydson`

Nome do dono do arquivo `teste`.

`user`

Nome do grupo que o arquivo `teste` pertence.

`teste`

Nome do arquivo.

22.2 Exemplo de acesso a um diretório

Abaixo um exemplo com explicações das permissões de acesso a um diretório no GNU/Linux :

```
drwxr-x--- 2 gleydson user 1024 nov 4 17:55 exemplo
```

drwxr-x---

Permissões de acesso ao diretório `exemplo`. É um conjunto de 10 letras que especificam o tipo de arquivo, permissão do dono do diretório, grupo que o diretório pertence e permissão de acesso a outros usuários. Veja as explicações abaixo:

d rwxr-x---

A primeira letra (do conjunto das 10) determina o tipo do arquivo. Neste caso é um diretório porque tem a letra `d`.

d rwx r-x —

Estas 3 letras (da segunda a quarta) são as permissões de acesso do dono do diretório exemplo . O dono do diretório (neste caso gleydson) tem a permissão para listar arquivos do diretório (`r`), gravar arquivos no diretório (`w`) e entrar no diretório (`x`). **drwx r-x** —

Estas 3 letras (da quinta a sétima) são as permissões de acesso dos usuários que pertencem ao grupo `user` . Os usuários que pertencem ao grupo `user` tem a permissão somente para listar arquivos do diretório (`r`) e entrar no diretório (`x`) exemplo .

drwxr-x —

Estas 3 letras (da oitava a décima) são as permissões de acesso para usuários que não são donos do diretório exemplo e que não pertencem ao grupo `user` . Com as permissões acima, nenhum usuário que se encaixe nas condições de dono e grupo do diretório tem a permissão de acessa-lo.

gleydson

Nome do dono do diretório exemplo .

user

Nome do grupo que diretório exemplo pertence.

exemplo

Nome do diretório. Para detalhes de como alterar o dono/grupo de um arquivo/diretório, veja os comandos `chmod` , `chgrp` e `chown` .

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.

23 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.

23.1 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.

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 .

24 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.

24.1 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, imagens, textos.

24.2 Tipos de compactação

Existem basicamente dois tipos de compactação, a compactação sem perdas e a compactação com perdas .

Os exemplos a seguir tentam explicar de forma simples os conceitos envolvidos.

A compactação sem perdas, como o próprio nome diz não causa nenhuma perda nas informações contidas no arquivo. Quando você compacta e descompacta um arquivo, o conteúdo é o mesmo do original.

A compactação com perdas é um tipo específico de compactação desenvolvido para atingir altas taxas, porém com perdas parciais dos dados. É aplicada a tipos de arquivos especiais, como músicas e imagens ou arquivos que envolvam a percepção humana.

Sabe-se que o ouvido humano não é tão sensível a determinados sons e frequências, então a compactação de um arquivo de música poderia deixar de gravar os sons que seriam pouco percebidos, resultando em um arquivo menor. Uma compactação do tipo ogg ou mp3 utiliza-se destes recursos. O arquivo resultante é muito menor que o original, porém alguns dados sonoros são perdidos. Você só notaria se estivesse reproduzindo a música em um equipamento de alta qualidade e se tivesse um ouvido bem aguçado. Para efeitos práticos, você está ouvindo a mesma música e economizando muito espaço em disco.

Outro exemplo de compactação com perdas são as imagens jpg. Imagine que você tem uma imagem com 60000 tons de cor diferentes, mas alguns tons são muito próximos de outros, então o compactador resume para 20000 tons de cor e a imagem terá 1/3 do tamanho original e o nosso olho conseguirá entender a imagem sem problemas e quase não perceberá a diferença. Exemplos de extensões utilizadas em imagens compactadas são jpg, png, gif.

Apesar das vantagens da grande taxa de compactação conseguida nos processos com perdas, nem sempre podemos utilizá-lo. Quando compactamos um texto ou um programa, não podemos ter perdas, senão o nosso texto sofre alterações ou o programa não executa. Nem mesmo podemos ter perdas quando compactamos imagens ou músicas que serão utilizadas em processos posteriores de masterização, mixagem ou impressão em alta qualidade.

24.3 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.
- * .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 .

24.4 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. Curingas 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.

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 .

24.5 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 curingas 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.

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.

24.6 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 curingas 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 curingas 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 .

24.7 rar

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 armazená-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 o tipo de compactador usado durante a geração do arquivo `.tar` pois será necessário especificar a opção apropriada para descompactá-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`.

24.8 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 consequência 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. Curingas 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 `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 equilíbrio 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 .

24.9 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 decidi coloca-lo aqui porque 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 curingas 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-extráctil. Arquivos auto-extrácteis 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-extrácteis. Arquivos auto-extrácteis 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 curingas

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).

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.

25 A distribuição Debian GNU/Linux

Este capítulo traz algumas características sobre a distribuição Debian GNU/Linux , programas de configuração e particularidades. A maioria dos trechos aqui descritos, também se aplicam a distribuições baseadas na Debian , como a Corel Debian GNU/Linux e a Libranet .

Você deve estar se perguntando mas porque um capítulo falando sobre a distribuição Debian se eu uso outra? . Bem, a partir da versão Intermediário do Foca Linux existem algumas partes que são específicas de algumas distribuições Linux e que não se aplicam a outras, como a localização dos arquivos de configuração, nomes dos programas de configuração e outros detalhes específicos e esta versão é a baseada na Debian . Pegue na página do Foca Linux (<http://focalinux.cipsga.org.br>) uma versão Intermediário do guia específico para sua distribuição.

25.1 Porque usar a Debian?

A Debian é a distribuição que mais cresce no mundo, cada versão é somente lançada após rigorosos testes de segurança e correção de falhas fazendo desta a mais segura e confiável dentre todas as outras distribuições Linux. É reconhecida como a mais segura, maior e atualizada mais freqüentemente entre as outras distribuições Linux , além de ser a única sem fins comerciais.

É a única que adota o estilo de desenvolvimento aberto e não é mantida por uma empresa comercial (note que o endereço do WebSite da Debian termina com .org), ao invés disso é mantida por programadores, hackers e especialistas de segurança espalhados ao redor do mundo, seguindo o estilo de desenvolvimento do Linux. Possui suporte a mais de 10 arquiteturas e 15 sub-arquiteturas (entre elas, Intel x86, Alpha, VMS, Sparc, Macintosh (m68k), Power Pc, ARM, etc).

Suas atualizações são constantes e não é necessário adquirir um novo CD para fazer upgrades. Meu sistema é atualizado semanalmente e de forma segura através de 2 simples comandos.

Cada pacote da distribuição é mantida por uma pessoa, o que garante uma boa qualidade, implementações de novos recursos e rápida correção de falhas. Qualquer pessoa com bons conhecimentos no sistema e inglês pode se tornar um Debian Developer , para detalhes consulte a lista de discussão `debian-user-portuguese` ou veja a página oficial da Debian : <http://www.debian.org/>.

A distribuição apresenta compatibilidade com outros sistemas a partir da instalação até a seleção de programas e execução do sistema, sua instalação está até mesmo disponível desde computadores 386 que utilizam unidades de disquetes de 5 1/4 polegadas até para computadores UDMA66 e com CD-ROM inicializável.

É a distribuição mais indicada para uso em servidores devido ao seu desempenho, segurança e programas úteis de gerenciamento e monitoração da rede, recomendados por especialistas que participam de seu desenvolvimento.

Não existem versões separadas da Debian para servidores, uso pessoal, etc, ao invés disso a distribuição usa perfis de usuário (dependendo da função do usuário) e perfis de computador (dependendo do que deseja fazer), podendo ser selecionado mais de um perfil de usuário/computador.

Os perfis selecionam automaticamente os pacotes mais úteis para a instalação. Os pacotes existentes em cada perfil foram escolhidos através de debates entre usuários que trabalham ativamente naquela área, resultando em uma seleção de pacotes de alta produtividade.

Para os usuários avançados e exigentes, também é possível selecionar os pacotes individualmente via `dselect` , o que resultará em uma instalação somente com pacotes úteis e melhor configurada.

25.2 Pacotes existentes na Debian

O número de pacotes existentes na distribuição atual da Debian (Woody - 3.0) é de 8710.

A Debian (como a Red Hat) usa um formato próprio para armazenar os programas: o formato `.deb` . Este formato permite a declaração, resolução e checagem automática de dependências, pacotes sugeridos, opcionais e outras características que o torna atraente para o desenvolvimento, gerenciamento e manutenção do sistema.

Estes pacotes são gerenciados através do programa `dpkg` (Debian Package) ou através de front-ends como o `dselect` ou `apt`.

25.3 que é sid/testing/frozen/stable?

Para o lançamento de uma nova distribuição Debian , o seguinte processo ocorre: `sid => testing => stable` (sendo a `stable` sempre o lançamento oficial e sem bugs da distribuição).

Sid Durante o desenvolvimento de uma nova distribuição Debian , ela é chamada de `sid` . A `sid` é a versão Unstable , isto não significa instabilidade, mas sim que a distribuição esta sofrendo modificações para se tornar uma versão estável, recebendo novos pacotes, etc.

Quando os pacotes não são modificados após um determinado período, os scripts da Debian copiam estes pacotes (novos ou atualizados) para a `testing` .

Não use a distribuição `sid` (unstable) ao menos que tenha experiência no Linux para corrigir problemas, que certamente aparecerão.

Testing A `testing` recebe os pacotes que não são modificados durante algum tempo da `unstable` , isto significa que eles possuem alguma estabilidade.

A `testing` é uma espécie de congelamento permanente (freeze) durante o desenvolvimento da Unstable .

Os novos pacotes que entram na unstable também caem na testing após certo tempo.

Mesmo assim, podem existir falhas graves na testing , se você precisa de um ambiente realmente livre de falhas, use a stable .

Frozen (congelada) Na data programada pela equipe de lançamento da Debian , a distribuição testing é congelada: nenhum pacote novo da unstable cai na testing e começa a procura de falhas na distribuição testing . Nenhuma nova característica é implementada nos pacotes (a não ser que seja extremamente necessário) e os developers se dedicam a correção de erros nos pacotes.

A distribuição testing congelada se tornará a futura stable após todas as falhas serem corrigidas. É considerado seguro usar a frozen após 1 mês de "congelamento".

Quando a testing é congelada, o ciclo de desenvolvimento da unstable continua para que a próxima distribuição da Debian seja lançada.

Stable Quando todos os bugs da testing congelada são eliminados, ela é lançada como stable , a nova versão Oficial da Debian .

A stable é o resultado final do desenvolvimento, das correção de falhas/segurança e que passou por todos os ciclos de testes para ser lançada. Resumindo é a distribuição pronta para ser usada com toda a segurança.

25.4 Como obter a Debian

A instalação da distribuição pode ser obtida através do site oficial do debian - www.debian.org

25.5 Programas de configuração

* dselect - Seleciona pacote para instalação/desinstalação

* pppconfig - Configura o computador para se conectar a Internet facilmente. Após isto, use pon para se conectar a Internet, poff para se desconectar e plog para monitorar a conexão.

* modconf - Permite selecionar os módulos que serão automaticamente carregados na inicialização do sistema. Se requerido pelos módulos os parâmetros I/O, IRQ e DMA também podem ser especificados.

* kbdconfig - Permite selecionar o tipo de teclado que utiliza

* shadowconfig - Permite ativar ou desativar o suporte a senhas ocultas (shadow password). Com as senhas ocultas ativadas, as senhas criptografadas dos usuários e grupos são armazenadas nos arquivos shadow e gshadow respectivamente, que somente podem ser acessadas pelo usuário root.

Isto aumenta consideravelmente a segurança do sistema pois os arquivos passwd e group contém dados de usuários que devem ter permissão de leitura de todos os usuários do sistema.

* tasksel - Permite selecionar/modificar de forma fácil a instalação de pacotes em seu sistema através da função que sua máquina terá ou do seu perfil de usuário.

* tzconfig - Permite modificar/selecionar o fuso-horário usado na distribuição.

Além destes, a Debian 3.0 (Woody) conta com o sistema de configuração baseado no dpkg-reconfigure que permite configurar de forma fácil e rápida aspecto de pacotes: dpkg-reconfigure xserver-*svga* .

25.6 Sistema de gerenciamento de pacotes

Este capítulo ensina a operação básica do programa de manipulação de pacotes Debian , a instalação, remoção, consulta e checagem de arquivos .deb .

26 dpkg

O dpkg (Debian Package) é o programa responsável pelo gerenciamento de pacotes em sistemas Debian . Sua operação é feita em modo texto e funciona através de comandos, assim caso deseje uma ferramenta mais amigável para a seleção e instalação de pacotes, prefira o dselect (que é um front-end para o dpkg) ou o apt , dpkg é muito usado por usuários avançados da Debian e desenvolvedores para fins de instalação, manutenção e construção de pacotes.

26.1 Pacotes

Pacotes Debian são programas colocados dentro de um arquivo identificados pela extensão .deb incluindo arquivos necessários para a instalação do programa, um sistemas de listagem/checagem de dependências, scripts de automatização para remoção parcial/total do pacote, listagem de arquivos, etc.

Um nome de pacote tem a forma nome-versão revisão.deb

26.2 Instalar pacotes

Use o comando: `dpkg -i [NomeDoPacote]` para instalar um pacote em seu sistema. Talvez ele peça que seja instalado algum pacote que depende para seu funcionamento. Para detalhes sobre dependências veja Seção

26.3 Dependências

Dependências são pacotes requeridos para a instalação de outro pacote. Na Debian cada pacote contém um programa com uma certa função. Por exemplo, se você tentar instalar o pacote de edição de textos supertext que usa o programa sed , você precisará verificar se o pacote sed está instalado em seu sistema antes de tentar instalar o supertext , caso contrário, o pacote supertext pedirá o sed e não funcionará corretamente. Note que o pacote supertext é apenas um exemplo e não existe (pelo menos até agora :-). O programa dselect faz o trabalho de checagem de dependências automaticamente durante a instalação dos pacotes.

A colocação de cada programa em seu próprio pacote parece ser uma dificuldade a mais para a instalação manual de um certo programa. Mas para os desenvolvedores que mantém os mais de 4300 pacotes existentes na distribuição Debian , é um ponto fundamental, porque não é preciso esperar uma nova versão do supertext ser lançada para instalar a versão mais nova do pacote sed . Por este motivo também é uma vantagem para o usuário.

26.4 Listar pacotes existentes no sistema

Use o comando: `dpkg -l [pacote]` para isto.

Na listagem de pacotes também será mostrado o "status" de cada um na coluna da esquerda, acompanhado do nome do pacote, versão e descrição básica. Caso o nome do [pacote] seja omitido, todos os pacotes serão listados.

É recomendado usar "dpkg -l—less" para ter um melhor controle da listagem (pode ser longa dependendo da quantidade de programas instalados).

26.5 Removendo pacotes do sistema

Use o comando: `dpkg -r NomedoPacote` para remover um pacote do sistema completamente. Somente é necessário digitar o nome e versão do pacote que deseja remover, não sendo necessário a revisão do pacote.

O comando `dpkg -r` não remove os arquivos de configuração criados pelo programa.

26.6 Removendo completamente um pacote

Use o comando: `dpkg --purge NomedoPacote` para remover um pacote e todos os diretórios e arquivos de configuração criados. Não é necessário especificar a revisão do pacote. O comando `dpkg--purge` pode ser usado após uma remoção normal do pacote (usando `dpkg -r`).

Caso você usar diretamente o comando `dpkg --purge`, `dpkg` primeiro removerá o pacote normalmente e após removido apagará todos os arquivos de configuração.

Note que o `dpkg --purge` somente remove arquivos de configuração conhecidos pelo pacote. Em especial, os arquivos de configuração criados para cada usuário do sistema devem ser removidos manualmente. Seria pedir demais que o `dpkg` também conhecesse os usuários de nosso sistema ;-).

26.7 Mostrar descrição do pacote

Use o comando: `dpkg -I NomedoPacote` para mostrar a descrição do pacote. Entre a descrição são mostradas as dependências do pacote, pacotes sugeridos, recomendados, descrição do que o pacote faz, tamanho e número de arquivos que contém.

26.8 Procura de pacotes através do nome de um arquivo

Use o comando: `dpkg -S arquivo` para saber de qual pacote existente no sistema o arquivo pertence.

26.9 status do pacote

Use o comando: `dpkg -s pacote` para verificar o status de um pacote em seu sistema, se está ou não instalado, configurado, tamanho, dependências, maintainer, etc.

Se o pacote estiver instalado no sistema, o resultado será parecido com o do comando `dpkg -c [pacote]`. Este comando é equivalente ao `dpkg --status`.

26.10 Procurando pacotes com problemas de instalação

A checagem de pacotes com este tipo de problema pode ser feita através do comando:
`dpkg -C`

Será listado todos os pacotes com algum tipo de problema, verifique os detalhes do pacote com `dpkg -s` para decidir como corrigir o problema.

26.11 Mostrando a lista de pacotes do sistema

Use o comando:

```
dpkg --get-selections
```

para obter uma lista de seleção dos pacotes em seu sistema. A listagem é mostrada na saída padrão, que pode ser facilmente redirecionada para um arquivo usando `dpkg --get-selections >dpkg.lista`.

A listagem obtida com este comando é muito útil para repetir os pacotes usados no sistema usando o `dpkg --set-selections`.

26.12 Obtendo uma lista de pacotes para instalar no sistema

Use o comando:

```
dpkg --set-selections <arquivo>
```

para obter a lista de pacotes que serão instalados no sistema. O uso do `dpkg --get-selections` e `dpkg --set-selections` é muito útil durante uma necessidade de reinstalação do sistema GNU/Linux ou repetir a instalação em várias máquinas sem precisar selecionar algumas dezenas entre os milhares de pacotes no `dselect`.

Após obter a lista com `dpkg --get-selections`, use `dpkg --set-selections |arquivo` e então entre no `dselect` e escolha a opção `INSTALL`, todos os pacotes obtidos via `dpkg --set-selections` serão automaticamente instalados.

26.13 Configurando pacotes desconfigurados

Pacotes estão desconfigurados quando, por algum motivo, a instalação do mesmo não foi concluída com sucesso. Pode ter faltado alguma dependência, acontecido algum erro de leitura do arquivo de pacote, etc. Quando um erro deste tipo acontece, os arquivos necessários pelo pacote podem ter sido instalados, mas os scripts de configuração pós-instalação não são executados.

Use o comando:

```
dpkg --configure [ NomeDoPacote ]
```

Para configurar um pacote. O `NomeDoPacote` não precisa conter a revisão do pacote e extensão.

26.14 Listando arquivos de um pacote

Use o comando: `dpkg -c arquivo` para obter a listagem dos arquivos contidos no pacote. É necessário digitar o nome completo do pacote. O comando `dpkg -c` é útil para listarmos arquivos de pacotes que não estão instalados no sistema.

Para obter a listagem de arquivos de pacotes já instalados no sistema, use o comando: `dpkg -L arquivo`. É necessário digitar somente o nome do pacote (sem a revisão e extensão).

27 apt

O apt é sistema de gerenciamento de pacotes de programas que possui resolução automática de dependências entre pacotes, método fácil de instalação de pacotes, facilidade de operação, permite atualizar facilmente sua distribuição, etc. Ele funciona através de linha de comando e ainda não existe nenhuma interface amigável para uso deste programa, mesmo assim sua operação é muito fácil.

O apt pode utilizar tanto com arquivos locais como remotos na instalação ou atualização, desta maneira é possível atualizar toda a sua distribuição Debian via ftp ou http com apenas 2 simples comandos!

É recomendável o uso do método apt no programa dselect pois ele permite a ordem correta de instalação de pacotes e checagem e resolução de dependências, etc. Devido a sua facilidade de operação, o apt é o método preferido para os usuários manipularem pacotes da Debian .

O apt é exclusivo da distribuição Debian e distribuições baseadas nela e tem por objetivo tornar a manipulação de pacotes poderosa por qualquer pessoa e tem dezenas de opções que podem ser usadas em sua execução ou configuradas no arquivo `/etc/apt/apt.conf` . Explicarei aqui como fazer as ações básicas com o apt , portanto se desejar maiores detalhes sobre suas opções, veja a página de manual apt-get .

27.1 O arquivo `/etc/apt/sources.list`

Este arquivo contém os locais onde o apt encontrará os pacotes, a distribuição que será verificada (stable, testing, unstable, Woody, Sarge) e a que será copiada (main, non-free, contrib, non-US).

Woody (Debian 3.0) e Sarge (Debian 3.1) são os nomes das versões enquanto stable e unstable são links para as versões estável e testing respectivamente. Se desejar usar sempre uma distribuição estável (como a Woody), modifique o arquivo `sources.list` e coloque Woody como distribuição. Caso você desejar estar sempre atualizado mas é uma pessoa cuidadosa e deseja ter sempre a última distribuição estável da Debian , coloque stable como versão. Assim que a nova versão for lançada, os links que apontam de stable para Woody serão alterados apontando para Sarge e você terá seu sistema atualizado.

Abaixo um exemplo simples de arquivo `/etc/apt/sources.list` com explicação das seções:

```
deb http://www.debian.org/debian stable main contrib non-free
deb http://nonus.debian.org/debian-non-US stable non-US
```

Você pode interpretar cada parte da seguinte maneira:

- * deb - Identifica um pacote da Debian. A palavra deb-src identifica o código fonte.
- * http://www.debian.org/debian - Método de acesso aos arquivos da Debian , site e diretório principal. O caminho pode ser http:// , ftp:// , file/ .
- * stable - Local onde serão procurados arquivos para atualização. Você pode tanto usar o nome de sua distribuição (Woody , Sarge) ou sua classificação (stable , testing ou unstable) . Note que unstable é recomendada somente para desenvolvedores, máquinas de testes e se você tem conhecimentos para corrigir problemas. Nunca utilize unstable em ambientes de produção ou servidores críticos, use a stable .
- * main contrib non-us - Seções que serão verificadas no site remoto.

Note que tudo especificado após o nome da distribuição será interpretado como sendo as seções dos arquivos (main, non-free, contrib, non-US). As linhas são processadas na ordem que estão no arquivo, então é recomendável colocar as linhas que fazem referência a pacotes locais primeiro e mirrors mais perto de você para ter um melhor aproveitamento de banda. O caminho percorrido pelo apt para chegar aos arquivos será o seguinte:

```
http://www.debian.org/debian/dists/stable/main/binary-i386
http://www.debian.org/debian/dists/stable/non-free/binary-i386
http://www.debian.org/debian/dists/stable/contrib/binary-i386
```

Você notou que o diretório dists foi adicionado entre `http://www.debian.org/debian` e `stable`, enquanto as seções `main`, `non-free` e `contrib` são processadas separadamente e finalizando com o caminho `binary-[arquitetura]`, onde `[arquitetura]` pode ser `i386`, `alpha`, `sparc`, `powerpc`, `arm`, etc. dependendo do seu sistema. Entendendo isto, você poderá manipular o arquivo `sources.list` facilmente.

OBS: Caso tenha mais de uma linha em seu arquivo `sources.list` de onde um pacote pode ser instalado, ele será baixado da primeira encontrada no arquivo. É recomendável colocar primeiro repositórios locais ou mais perto de você, como recomendado nesta seção.

27.2 Endereços de servidores e mirrors nacionais da Debian

Segue abaixo uma relação de servidores que podem ser colocados em seu arquivo `sources.list`:

Endereço	Diretório Principal
-----	-----
<code>ftp://ftp.debian.org.br</code>	<code>/debian</code>
<code>ftp://ftp.br.debian.org</code>	<code>/debian</code>
<code>ftp://ftp.debian.org</code>	<code>/debian</code>
<code>ftp://download.sourceforge.net</code>	<code>/debian</code>
<code>ftp://ftp.quimica.ufpr.br</code>	<code>/debian</code>
<code>ftp://download.unesp.br</code>	<code>/linux/debian</code>

27.3 Um modelo de arquivo `sources.list`

Você pode copiar o modelo do `sources.list` abaixo para ser usado em sua distribuição Stable ou personaliza-lo modificando a distribuição utilizada e servidores:

```
# Arquivos principais da stable
deb ftp://ftp.debian.org.br/debian stable main non-free contrib

# Non-US da Stable
deb ftp://ftp.debian.org.br/debian-non-US stable/non-US main non-free contrib

# Atualizações propostas para Stable main e non-US
deb ftp://ftp.debian.org.br/debian dists/proposed-updates/
```

```

deb ftp://ftp.debian.org.br/debian-non-US dists/proposed-updates/

# Atualizações de segurança da Stable
deb ftp://nonus.debian.org/debian-security stable/updates main

# Ximian é um conjunto de pacotes atualizados freqüentemente e compatíveis
# com a distribuição Debian. Entre estes programas estão o Gimp 1.2 e outros
# mais atuais e compatíveis com a Debian. Para usa-los inclua a seguinte linha no
# seu sources.list
# deb ftp://spidermonkey.ximian.com/pub/red-carpet/binary/debian-22-i386/ ./

# Kde 1 e 2
# deb ftp://kde.tdyc.com/pub/kde/debian woody main crypto optional qt1apps

```

27.4 O arquivo /etc/apt/apt.conf

Você pode especificar opções neste arquivo que modificarão o comportamento do programa apt durante a manipulação de pacotes (ao invés de especificar na linha de comando). Se estiver satisfeito com o funcionamento do programa apt, não é necessário modifica-lo. Para detalhes sobre o formato do arquivo, veja a página de manual do apt.conf. Na página de manual do apt-get são feitas referências a parâmetros que podem ser especificados neste arquivo ao invés da linha de comando.

27.5 Copiando a lista de pacotes disponíveis

O apt utiliza uma lista de pacotes para verificar se os pacotes existentes no sistema precisam ou não ser atualizados. A lista mais nova de pacotes é copiada através do comando apt-get update.

Este comando pode ser usado com alguma freqüência se estiver usando a distribuição stable e sempre se estiver usando a unstable (os pacotes são modificados com muita freqüência). Sempre utilize o apt-get update antes de atualizar toda a distribuição.

27.6 Utilizando CDs oficiais/não-oficiais/terceiros com o apt

Para usar CDs da Debian ou de programas de terceiros, use o seguinte comando com cada um dos CDs que possui:

```
apt-cdrom add
```

Este comando adicionará automaticamente uma linha para cada CD no arquivo /etc/apt/sources.list e atualizará a lista de pacotes em /var/state/apt/lists. Por padrão, a unidade acessada através de /cdrom é usada. Use a opção -d /dev/scd? para especificar um outra unidade de CDs.

Durante a instalação de um novo programa, o apt pede que o CD correspondente seja inserido na unidade e pressionado `Enter` para continuar. O método acesso do apt através de CDs é inteligente o bastante para instalar todos os pacotes necessários daquele CD, instalar os pacotes do próximo CD e iniciar a configuração após instalar todos os pacotes necessários.

Observação: - CDs de terceiros ou contendo programas adicionais também podem ser usados com o comando "apt-cdrom add".

27.7 Instalando novos pacotes

Use o comando `apt-get install [pacotes]` para instalar novos pacotes em sua distribuição. Podem ser instalados mais de um pacotes ao mesmo tempo separando os nomes por espaços. Somente é preciso especificar o nome do pacote (sem a versão e revisão).

Se preciso, o `apt` instalará automaticamente as dependências necessárias para o funcionamento correto do pacote. Quando pacotes além do solicitado pelo usuário são requeridos para a instalação, o `apt` mostrará o espaço total que será usado no disco e perguntará ao usuário se ele deseja continuar. Após a instalação, o pacote será automaticamente configurado pelo `dpkg` para ser executado corretamente em seu sistema.

27.8 Removendo pacotes instalado

Use o comando `apt-get remove [pacotes]` para remover completamente um pacote do sistema. Podem ser removidos mais de um pacote ao mesmo tempo separando os nomes dos pacotes com espaços. O `apt-get remove` remove completamente o pacote mas mantém os arquivos de configuração, exceto se for adicionada a opção `-purge`.

É preciso especificar somente o nome do pacote (sem a versão e revisão).

27.9 Atualizando sua distribuição

O `apt` tem uma grande característica: Atualizar toda a sua distribuição de uma forma inteligente e segura. O `apt` lê a listagem de pacotes disponíveis no servidor remoto, verifica quais estão instalados e suas versões, caso a versão do pacote seja mais nova que a já instalada em seu sistema, o pacote será imediatamente atualizado.

A cópia dos arquivos pelo `apt` pode ser feita via `FTP`, `HTTP` ou através de uma cópia local dos arquivos no disco rígido (um `mirror` local). Em nenhuma circunstância os pacotes existentes em seu sistema serão removidos ou sua configuração apagada durante um `upgrade` na distribuição.

Os arquivos de configuração em `/etc` que foram modificados são identificados e podem ser mantidos ou substituídos por versões existentes nos pacotes que estão sendo instalado, esta escolha é feita por você. Se estiver atualizando a `Debian Potato (2.2)` para `Woody (3.0)` (ou versão superior), execute os seguintes comandos antes de iniciar a atualização:

```
export LANG=C
export LC_ALL=C
export LC_MESSAGES=C
```

para retornar as variáveis de localização ao valor padrão (inglês). Isto é necessário por causa de modificações no sistema de locais, e o excesso de mensagens de erro do `perl` causaram alguns problemas em meus testes.

Após isto, a atualização da distribuição `Debian` pode ser feita através de dois simples comandos:

```
apt-get update           #Para atualizar a lista de pacotes (obrigatório)
apt-get -f dist-upgrade #Para atualizar a distribuição
```

A opção `-f` faz com que o `apt` verifique e corrija automaticamente problemas de dependências entre pacotes. Recomendo executar o comando `apt-get -f -dry-run dist-upgrade--less`

para ver o que vai acontecer sem atualizar a distribuição, se tudo ocorrer bem, retire o `-dry-run` e vá em frente.

A distribuição usada na atualização pode ser:

- * Para a mesma versão que utiliza - Para quem deseja manter os pacotes sempre atualizados entre revisões, copiar pacotes que contém correções para falhas de segurança (veja a página web em <http://www.br.debian.org> para acompanhar o boletim de segurança).

- * Para uma distribuição stable - Mesmo que o acima, mas quando uma nova distribuição for lançada, o link simbólico de stable será apontado para próxima distribuição, atualizando instantaneamente seu sistema.

- * Para a distribuição testing - Atualiza para a futura distribuição Debian que será lançada, é como a unstable, mas seus pacotes passam por um período de testes de 2 semanas na unstable antes de serem copiados para esta.

- * unstable - Versão em desenvolvimento, recomendada somente para desenvolvedores ou usuários que conhecem a fundo o sistema GNU/Linux e saibam resolver eventuais problemas que apareçam. A unstable é uma distribuição em constante desenvolvimento e podem haver pacotes problemáticos ou com falhas de segurança. Após o período de desenvolvimento, a distribuição unstable se tornará frozen.

- * frozen - Versão congelada, nenhum pacote novo é aceito e somente são feitas correções de falhas. Após todas as falhas estarem corrigidas, a distribuição frozen se tornará stable

A distribuição que será usada na atualização pode ser especificada no arquivo `/etc/apt/sources.list` (veja a seção correspondente acima). Caso o método de atualização usado seja via HTTP ou FTP, será necessário usar o comando `apt-get clean` para remover os pacotes copiados para seu sistema (para detalhes veja a seção seguinte).

27.10 Removendo pacotes baixados pelo apt

Use o comando `apt-get clean` para apagar qualquer arquivo baixado durante uma atualização ou instalação de arquivos com o `apt`. Os arquivos baixados residem em `/var/cache/apt/archives` (download completo) e `/var/cache/apt/archives/partial` (arquivos sendo baixados - parciais).

Este local de armazenamento é especialmente usado com o método `http` e `ftp` para armazenamento de arquivos durante o download para instalação (todos os arquivos são primeiro copiados para serem instalados e configurados).

O `apt-get clean` é automaticamente executado caso seja usado o método de acesso `apt` do `dselect`.

27.11 Procurando por pacotes através da descrição

O utilitário `apt-cache` pode ser usado para esta função. Ele também possui outras utilidades interessante para a procura e manipulação da lista de pacotes.

Por exemplo, o comando `apt-cache search clock` mostrará todos os pacotes que possuem a palavra `clock` na descrição do pacote.

27.12 Procurando um pacote que contém determinado arquivo

Suponha que algum programa esteja lhe pedindo o arquivo `perlcc` e você não tem a mínima idéia de que pacote instalar no seu sistema. O utilitário `auto-apt` pode resolver esta situação. Primeiro instale o pacote `auto-apt` e execute o comando `auto-apt update` para que ele copie o arquivo `Contents-i386.gz` que será usado na busca desses dados.

Agora, basta executar o comando:

```
auto-apt search perlcc
```

para que ele retorne o resultado:

```
usr/bin/perlcc interpreters/perl
```

O pacote que contém este arquivo é o perl e se encontra na seção interpreters dos arquivos da Debian . Para uma pesquisa que mostra mais resultados (como auto-apt search a2ps), é interessante usar o grep para filtrar a saída:

```
auto-apt search a2ps|grep bin/

usr/bin/psmandup      text/a2ps
usr/bin/pdiff        text/a2ps
usr/bin/psset        text/a2ps
usr/bin/composeglyphs text/a2ps
usr/bin/a2psj        text/a2ps-perl-ja
usr/bin/a2ps         text/a2ps
usr/bin/fixps        text/a2ps
usr/bin/ogonkify     text/a2ps
usr/bin/fixnt        text/a2ps
usr/bin/card         text/a2ps
usr/bin/texti2dvi4a2ps text/a2ps
```

Serão mostrados somente os binários, diretórios de documentação, manpages, etc. não serão mostradas.

27.13 Modos eficazes de compilação do código fonte para a Debian

O Debian como qualquer distribuição de Linux, possui o diretório /usr/local que segundo a FHS é o local apropriado para colocação de programas que não fazem parte da distribuição, que seria no caso o de fontes compilados manualmente. Um dos grandes trabalhos de quem pega o código fonte para compilação é a instalação de bibliotecas de desenvolvimento para a compilação ocorrer com sucesso.

O auto-apt facilita magicamente o processo de compilação da seguinte forma: durante o passo ./configure no momento que é pedida uma bibliotecas, dependência, etc. o auto-apt para o processo, busca por pacotes no repositório da Debian , pergunta qual pacote será instalado (caso tenha mais de uma opção), instala e retorna o ./configure do ponto onde havia parado.

Para fazer isso, execute o comando:

```
auto-apt run ./configure
```

E ele se encarregará do resto :-)

27.14 Verificando pacotes corrompidos

Use o comando apt-get check para verificar arquivos corrompidos. A correção é feita automaticamente. A lista de pacotes também é atualizada quando utiliza este comando.

27.15 Corrigindo problemas de dependências e outros erros

Use o comando `apt-get -f install` (sem o nome do pacote) para que o `apt-get` verifique e corrija problemas com dependências de pacotes e outros problemas conhecidos.

Esta apostila é uma adaptação feita por Franklin Anderson de Oliveira Souza do tutorial Foca Linux mantido por **Gleydson Mazioli da Silva**, que pode ser encontrado no seguinte endereço:

<http://focalinux.cipsga.org.br/>

Franklin Anderson de Oliveira Souza <franklinbr@linuxmail.org>

Este material foi produzido com o uso exclusivo de Software Livre. \LaTeX

É garantida a permissão para copiar, distribuir e/ou modificar este documento sob os termos da GNU Free Documentation Licence.