



Introdução ao Linux

Por: *Leandro Dalceró*

Índice

1. Introdução	Pág. 4
1.1. Sistema Operacional.....	Pág. 4
1.2. O que é Linux?	Pág. 4
2. Primeiros Passos	Pág. 6
2.1. Usuários	Pág. 6
2.2. Grupos	Pág. 6
2.3. Superusuário	Pág. 6
2.4. Entrando e saindo do sistema	Pág. 6
2.5. Desligando o sistema	Pág. 7
3. Gerenciamento de Arquivos e Diretórios	Pág. 8
3.1. Listando Arquivos	Pág. 8
3.2. Metacaracteres	Pág. 9
3.3. Estrutura de diretórios	Pág. 10
3.4. Criando e Removendo diretórios	Pág. 11
3.4.1. Criando diretórios	Pág. 11
3.4.2. Removendo diretórios	Pág. 11
3.5. Copiando Arquivos	Pág. 12
3.6. Mover ou Renomear	Pág. 13
3.7. Removendo Arquivos	Pág. 14
3.8. Links	Pág. 14
3.8.1. Hard Links	Pág. 14
3.8.2. Links Simbólicos	Pág. 15
4. Permissões de Arquivos	Pág. 16
4.1. Conceitos	Pág. 16
4.2. Utilizando o chmod	Pág. 17
4.3. Permissões padrão (umask)	Pág. 18
4.4. Utilizando chown	Pág. 18
5. Gerenciamento de contas de usuários	Pág. 19
5.1. Contas de usuários	Pág. 19
5.2. O comando useradd ou adduser	Pág. 19
5.3. O comando passwd	Pág. 20
5.4. O comando userdel	Pág. 20
5.5. O comando groupadd ou addgroup	Pág. 20
5.6. O comando gpasswd	Pág. 21
5.7. Verificando informações do usuário	Pág. 21
6. Gerenciamento de Processos	Pág. 22
6.1. Criar, Monitorar e Eliminar processos	Pág. 22
6.2. O comando ps	Pág. 22
6.3. O comando kill	Pág. 23
6.4. O comando top	Pág. 23
6.5. O comando jobs	Pág. 23

6.6. O comando bg	Pág. 24
6.7. O comando fg	Pág. 24
7. Compactação e Empacotamento de Arquivos	Pág. 24
7.1. Compactadores de Arquivos	Pág. 24
7.1.1. GZIP	Pág. 24
7.1.2. BZIP2	Pág. 25
7.1.3. Compress	Pág. 25
7.1.4. ZIP	Pág. 26
7.2. O comando tar	Pág. 26
8. Instalação de Aplicativos	Pág. 27
8.1. Gerenciamento de Pacotes RPM	Pág. 27
8.1.1. Conceitos básicos	Pág. 27
8.1.2. Vantagens da utilização do RPM	Pág. 27
8.1.3. Instalando / Atualizando	Pág. 28
8.1.4. Desinstalando	Pág. 28
8.1.5. Consultando / Verificando	Pág. 29
8.2. Instalação de Aplicativos em Código Binário	Pág. 29
8.2.1. O comando configure	Pág. 29
8.2.2. O comando make e make install	Pág. 29
9. Uso e Configuração do Ambiente Gráfico	Pág. 30
9.1. Gerenciadores de Janelas	Pág. 30
9.2. Usando a Área de Trabalho	Pág. 30
9.3. Gerenciador de Arquivos	Pág. 30
9.4. Adicionando ícones na área de trabalho	Pág. 31
9.5. Usando a Lixeira	Pág. 31
9.6. Configurando a Área de Trabalho	Pág. 31
9.7. Trocando a configuração do seu teclado	Pág. 31
10. Navegação na Internet	Pág. 32
10.1. Conhecendo o Mozilla	Pág. 32
10.2. Navegando com o Mozilla	Pág. 32
11. Envio e Recebimento de E-Mails	Pág. 33
11.1. Usando o Evolution	Pág. 33
11.2. Configurando Contas	Pág. 34
11.3. Enviando e Recebendo E-Mails	Pág. 34
12. Mensagens Instantâneas	Pág. 34
12.1. LICQ	Pág. 34
12.2. GAIM	Pág. 35
13. Utilização do IRC (CHAT)	Pág. 36
13.1. Como utilizar programas baseados em IRC	Pág. 36
14. Impressão de Arquivos	Pág. 36
14.1. Imprimindo Arquivos	Pág. 36
14.2. Filas de Impressão	Pág. 37

1. Introdução

O capítulo *Introdução* contém explicações teóricas sobre o sistema operacional, você pode pular este capítulo caso já conheça estas explicações.

Para quem está começando, muita teoria pode atrapalhar o aprendizado, é mais produtivo ver na prática o que o computador faz e depois porque ele faz isto. Mesmo assim, é recomendado ler estes capítulos pois seu conteúdo pode ser útil.

Seguem abaixo algumas dicas para um bom começo:

- Faça a leitura desta apostila e pratique o que aprendeu imediatamente. Isto facilita o entendimento do programa/comando.
- É preciso ter interesse em aprender, se você tiver vontade em aprender algo, você terá menos dificuldade do que em algo que não gosta e está se obrigando a aprender.
- Decorar não adianta, pelo contrário, só atrapalha no aprendizado. Você precisa entender o que o comando faz, deste modo você estará também usando e desenvolvendo sua interpretação, e entenderá melhor o assunto.
- Curiosidade também é importante. Você talvez possa estar procurando um comando que mostre os arquivos que contém um certo texto, e isto fará você chegar até o comando `grep`, depois você conhecerá suas opções, etc.
- Não desanime vendo outras pessoas que sabem mais que você, lembre-se que ninguém nasce sabendo. Uma pessoa pode ter mais experiência em um assunto no sistema como compilação de programas, configuração, etc, e você pode ter mais interesse em redes.
- Ninguém pode saber tudo da noite para o dia, não procure saber tudo sobre o sistema de uma só vez senão não entenderá NADA. Caso tenha dúvidas sobre o sistema, procure ler novamente esta apostila, e caso ainda não tenha entendido procure ajuda nas milhares de páginas web disponíveis com documentação para o Linux.
- A interface gráfica existe, mas os melhores recursos e flexibilidade estão na linha de comando. Você pode ter certeza que o aprendizado no Linux ajudará a ter sucesso e menos dificuldades em usar qualquer outro sistema operacional.

Boa Sorte e bem vindo ao Linux!

1.1 Sistema Operacional

O *Sistema Operacional* é a interface ao usuário e seus programas com o computador. Ele é responsável pelo gerenciamento de recursos e periféricos (como memória, discos, arquivos, impressoras, CD-ROMs, etc) e a execução de programas.

No Linux o Kernel é o Sistema Operacional. Você poderá construí-lo de acordo com a configuração de seu computador e os periféricos que possui.

1.2 O que é Linux?

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

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

Algumas características do sistema Linux:

- É de graça e desenvolvido voluntariamente por programadores experientes, hackers, e contribuidores espalhados ao redor do mundo que tem como objetivo a contribuição para a melhoria e crescimento deste sistema operacional.
- Convivem sem nenhum tipo de conflito com outros sistemas operacionais (com o DOS, Windows, OS/2) no mesmo computador.
- Multitarefa real
- Multiusuário
- Suporte a nomes extensos de arquivos e diretórios (255 caracteres)
- Conectividade com outros tipos de plataformas como *Apple, Sun, Macintosh, Sparc, Alpha, PowerPc, ARM, Unix, Windows, DOS, etc.*
- Utiliza permissões de acesso a arquivos, diretórios e programas em execução na memória RAM.
- **NÃO EXISTEM VIRUS NO LINUX!** Em 9 anos de existência, nunca foi registrado NENHUM tipo de vírus para este sistema. Isto tudo devido a grande segurança oferecida pelas permissões de acesso do sistema que funcionam inclusive durante a execução de programas.
- O sistema de arquivos usado pelo Linux (Ext3) organiza os arquivos de forma inteligente evitando a fragmentação e fazendo-o um poderoso sistema para aplicações multi-usuárias exigentes e gravações intensivas.
- Suporte a diversos dispositivos e periféricos disponíveis no mercado, tanto os novos como obsoletos.
- Entre muitas outras características que você descobrirá durante o uso do sistema.

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

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

2. Primeiros Passos

Este capítulo apresenta o Linux ao aluno. Serão vistos os conceitos de:

- Usuário
- Grupo
- Superusuário
- Login (entrando no sistema)
- Logout (saindo do sistema)
- Shutdown/halt (desligando corretamente o sistema)

2.1 Usuário

Como o Linux foi concebido para que várias pessoas pudessem utilizar os mesmos recursos presentes em uma única máquina, surgiu o conceito de usuário para diferenciar o que cada pessoa estivesse fazendo e quais recursos ela estivesse ocupando. O usuário é a identificação da pessoa que irá utilizar o sistema.

A identificação do usuário é feita por um “número de identificação” ou id, que é atribuído ao usuário durante a criação de sua conta no sistema.

Com a finalidade de garantir a integridade do trabalho de cada usuário, impedindo que um usuário altere o trabalho de outro, no momento de entrada no sistema, você deve informar a senha do seu usuário. O nome de usuário associado à senha é a sua “chave de entrada” no sistema, portanto deve ser guardada com cuidado.

2.2 Grupos

O Linux também possui o conceito de “grupo”. Um grupo é, como o próprio nome diz, um agrupamento de vários usuários que devem compartilhar algumas características em comum como, por exemplo, permissões de acessos a arquivos e dispositivos.

2.3 Superusuário

O superusuário é aquele que tem plenos poderes dentro do Linux. É o superusuário quem pode criar novos usuários, alterar direitos, configurar e fazer a atualização do sistema. Somente ele tem direito de executar essas atividades.

É recomendado utilizar a conta de superusuário somente quando for necessário configurar algo no sistema. Ainda assim é recomendado utilizá-la o mínimo possível para evitar que algum erro danifique o sistema.

2.4 Entrando e Saindo do Sistema

Ao iniciar o Linux, um prompt semelhante ao ilustrado a seguir será mostrado:

```
Mandrake Linux (tty1)
ldalcer0 login:
```

Informe o seu login/nome de usuário. A seguir será solicitada a senha (Password) do usuário. Digite a senha do seu usuário.

Após informar o nome de usuário e a senha corretamente, você será levado ao prompt do sistema:

```
[aluno@ldalcero aluno]$
```

OBS.: O linux tem terminais virtuais. Você pode alterar entre eles utilizando as teclas **Alt-Fn**, onde n pode variar de 1 até 6 na configuração padrão.

Pode-se utilizar o comando **logout** na linha de comando para se desconectar do sistema:

```
[aluno@ldalcero aluno]$ logout
```

2.5 Desligando o Sistema

A fim de evitar danos ao sistema de arquivos, é necessário que o superusuário pare o sistema antes de desligar o computador. Um dos comandos que podem ser utilizados é o comando **shutdown**. Este comando permite tanto desligar quanto reiniciar o computador.

```
[aluno@ldalcero aluno]$ shutdown -h now
```

O comando acima permite desligar o computador imediatamente, enviando uma mensagem a todos os usuários que estão utilizando o sistema.

```
[aluno@ldalcero aluno]$ shutdown -h -t 30 "Atenção: O sistema será desligado dentro de 30 segundos"
```

O comando acima finaliza todos os processos e desliga o computador dentro de 30 segundos, enviando a mensagem de aviso a todos os usuários logados no sistema.

O comando **halt** diz ao sistema que ele deverá desligar imediatamente.

```
[aluno@ldalcero aluno]$ halt
```

Para reinicializar o sistema, pode-se utilizar, além do comando **shutdown**, o comando **reboot**:

```
[aluno@ldalcero aluno]$ shutdown -r -t 30 "Atenção: O sistema será reiniciado dentro de 30 segundos"
```

Esta opção finaliza todos os processos e **reinicia** o computador após 30 segundos.

```
[aluno@ldalcero aluno]$ reboot
```

O comando **reboot** chama o comando **shutdown** e ao final deste reinicia o sistema. Após executar os comandos deve-se aguardar até que o sistema esteja parado (com a mensagem o sistema está parado ou Power down) para então poder desligar seu computador ou esperar que ele reinicie.

3. Gerenciamento de Arquivos e Diretórios

Será visto nesse tópico como criar, renomear, excluir e mover arquivos e diretórios, bem como criar links.

3.1 Listando Arquivos

O comando **ls** mostra o conteúdo de um diretório.

O formato do comando é o seguinte:

```
ls [-l] [-a] [-F] [dir]
```

Onde [-l] é o formato longo, e [-a] serve para mostrar todos os arquivos, incluindo arquivos ocultos (os quais têm seu nome indicado por um ponto). Existem várias outras opções, embora estas sejam mais usadas. Finalmente, [-F] coloca no final dos nomes de arquivo um símbolo indicando o seu tipo.

Um exemplo do uso do **ls** é mostrado a seguir:

```
[aluno@ldalcero X11]$ ls
LessTif bin doc etc fonts include lib man share

[aluno@ldalcero X11]$
```

Um exemplo do uso do **ls** usando parâmetros:

```
[aluno@ldalcero X11]$ ls -laF
drwxr-xr-x 11 root root 4096 Ago 27 2002 ./
drwxr-xr-x 22 root root 4096 Mar 22 2003 ../
drwxr-xr-x 3 root root 4096 Mar 22 2003 LessTif/
drwxr-xr-x 2 root root 8192 Jul 24 08:45 bin/
drwxr-xr-x 2 root root 4096 Jul 24 2003 doc/
drwxr-xr-x 4 root root 4096 Ago 27 2002 etc/
drwxr-xr-x 2 root root 4096 Mar 22 2003 fonts/
drwxr-xr-x 4 root root 4096 Mar 22 2003 include/
drwxr-xr-x 9 root root 4096 Jul 24 2003 lib/
drwxr-xr-x 7 root root 4096 Jul 24 2003 man/
drwxr-xr-x 15 root root 4096 Jul 24 08:45 share/

[aluno@ldalcero X11]$
```

No exemplo acima, como são nomes de diretórios; o parâmetro [-F] adiciona uma barra indicando nome de diretório. O parâmetro [-l] coloca várias informações sobre o arquivo (permissões, links, dono, grupo, tamanho, data, hora, nome do arquivo).

3.2 Metacaracteres

Existem sinais, chamados metacaracteres, usados para facilitar a utilização de comandos no Linux.

Quando se trabalha com os comandos de manipulação de arquivos, freqüentemente é útil empregarmos metacaracteres. Estes símbolos – como *, ?, [], {} - são úteis para se referenciar arquivos que possuam características em comum.

Para os exemplos dados nesta seção, será usada a seguinte lista de arquivos:

```
[aluno@ldalcero aluno]$ ls
12arquivo  1arquivo  2arquivo  arquivo  arquivo3  arquivo34
arquivo5arquivo
```

- O asterisco “*“:

O asterisco é usado para representar “qualquer quantidade de qualquer caractere”. Por exemplo, **arquivo*** retornaria todos os arquivos em que o nome iniciasse com “arquivo”. Veja o efeito da utilização prática deste metacaractere.

```
[aluno@ldalcero aluno]$ ls arquivo*
arquivo  arquivo3  arquivo34  arquivo5arquivo
```

- O Ponto de Interrogação “?“:

O ponto de interrogação é utilizado para representar “um único e qualquer caractere”. Ao digitar `ls arquivo?`, o usuário estará pedindo a lista de arquivos cujos nomes são indicados por “arquivo” e terminal com um único caractere qualquer. Como no exemplo que segue:

```
[aluno@ldalcero aluno]$ ls arquivo?
arquivo3
```

Agora digitando `ls ??arquivo` o resultado seria:

```
12arquivo
```

- Os colchetes “[“]:

Os colchetes são utilizados para indicar uma lista de caracteres. Para entender melhor; verifique os exemplos.

```
[aluno@ldalcero aluno]$ ls arquivo[123]
arquivo3
```

```
[aluno@ldalcero aluno]$ ls [123]arquivo
1arquivo  2arquivo
```

- As chaves “{}”:

As chaves têm sua utilização muito assemelhada a dos colchetes. A diferença está na possibilidade de referenciar seqüências de caracteres separadas por vírgulas, conforme o exemplo a seguir:

```
[aluno@ldalcero aluno]$ ls arquivo{1,34}
arquivo34
```

```
[aluno@ldalcero aluno]$ ls arquivo{1,2,3,34}
arquivo3  arquivo34
```

3.3 Estrutura de Diretórios

O sistema Linux possui a seguinte estrutura básica de diretórios:

- /bin
Contém arquivos, programas do sistema que são usados com freqüência pelos usuários.
- /boot
Contém arquivos necessários para a inicialização do sistema.
- /cdrom
Ponto de montagem da unidade de CD-ROM.
- /dev
Contém arquivos usados para acessar dispositivos (periféricos) existentes no computador.
- /etc
Arquivos de configuração de seu computador local.
- /floppy
Ponto de montagem de unidade de disquetes.
- /home
Diretórios contendo os arquivos dos usuários.
- /lib
Bibliotecas compartilhadas pelos programas do sistema e módulos do kernel.
- /mnt
Ponto de montagem temporário.
- /usr
Contém maior parte de seus programas. Normalmente acessível somente como leitura.
- /var
Contém maior parte dos arquivos que são gravados com freqüência pelos programas do sistema, e-mails, spool de impressora, cache, etc.

- /sbin

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

3.4 Criando e Removendo Diretórios

3.4.1 Criando Diretórios

O comando **mkdir** é usado para criar diretórios. A sintaxe do comando será mostrado a seguir:

```
mkdir [parâmetros] nome_dir
```

A linha de comando a seguir cria um diretório:

```
[aluno@ldalcero aluno]$ mkdir meu_diretorio
```

O comando também pode ser usado para criar uma árvore de diretórios, como será mostrado a seguir:

```
[aluno@ldalcero aluno]$ mkdir -p meu_dir/meu_sub_dir/sub_sub_dir
```

O comando anterior cria um diretório chamado `meu_dir`, e dentro dele cria um subdiretório chamado `meu_sub_dir` e dentro deste, um subdiretório chamado `sub_sub_dir`.

Também é possível criar vários diretórios em simultâneo; simplesmente colocando vários nomes de diretórios junto com o comando, como será mostrado a seguir:

```
[aluno@ldalcero aluno]$ mkdir dir_1 dir_2 dir_3
```

O comando anterior criará os diretórios `dir_1`, `dir_2` e `dir_3` dentro do diretório atual.

3.4.2 Removendo Diretórios

O comando **rmdir** é usado para remover diretórios. Por exemplo; para remover o diretório `meu_dir` basta digitar o seguinte comando:

```
[aluno@ldalcero aluno]$ rmdir meu_dir
```

O comando também pode remover árvores de diretórios. Para tal, utiliza-se o parâmetro `[-p]`, como será mostrado a seguir:

```
[aluno@ldalcero aluno]$ rmdir -p temp/sub_dir/sub_dir_2
```

O comando anterior apagou o subdiretório `sub_dir_2`, depois apagou o subdiretório `sub_dir` e finalmente apagou o `temp`.

Porém, o comando remove diretórios e não arquivos; se existir algum arquivo dentro do diretório este não será removido.

Para conseguir remover diretórios com arquivos deve-se utilizar em conjunto o comando **rm**, que será visto mais adiante.

3.5 Copiando Arquivos

O comando **cp** é utilizado para efetuar a cópia de arquivos no Linux, Sua sintaxe é mostrada a seguir:

```
cp [parametros] arquivo_original [destino]
```

Observações importantes relativas à cópia de arquivos:

- Copiar um arquivo para outro diretório onde já existe outro arquivo com mesmo nome: o arquivo será sobrescrito.
- Copiar um arquivo para outro diretório que, por sua vez, possui um diretório com mesmo nome do arquivo a ser copiado: não é permitido, pois no Linux um diretório também é um arquivo.
- Copiar um arquivo, especificando como `arquivo_destino` outro nome: o arquivo será renomeado durante a cópia.

<i>Opção</i>	<i>Descrição</i>
-a	Preserva as permissões do arquivo_original quando possível.
-b	Faz backup de arquivos que serão sobrescritos.
-i	Solicita confirmação antes de sobrescrever arquivos.
-R	Copia diretórios recursivamente, ou seja, toda a árvore abaixo do diretório de origem. O destino sempre será um diretório.

Iniciando pela forma mais simples do comando, ou seja, copiar um arquivo para um novo arquivo. O comando pode ser visto a seguir:

```
[aluno@ldalcero aluno]$ cp doc.txt documento.txt
```

Neste caso ocorre a criação do arquivo documento.txt a partir do arquivo doc.txt. Também é possível copiar para outro local como será mostrado a seguir:

```
[aluno@ldalcero aluno]$ cp doc.txt /tmp
```

Como não foi mencionado o nome do arquivo de destino, será criado um arquivo com o mesmo nome do atual, É sempre bom ter um pouco de cuidado no uso do comando **cp**; para tal será usado o parâmetro **-i**. A menos que se utilize essa opção, o comando **cp** irá sobrescrever os arquivos existentes, como pode ser visto a seguir:

```
[aluno@ldalcero aluno]$ cp -i doc.txt /tmp
cp: sobrescrever `/tmp/doc/txt'? y
```

Um método mais seguro seria usar o parâmetro **-b** (backup); então, quando o **cp** encontra um arquivo com o mesmo nome cria uma cópia acrescentando um “~” ao nome do arquivo. Como pode-se observar a seguir:

```
[aluno@ldalcero aluno]$ cp -b doc.txt /tmp
[aluno@ldalcero aluno]$ ls /tmp/doc.txt*
doc.txt  doc.txt~
```

Também é possível copiar vários arquivos simultaneamente; para tal, basta colocar os nomes dos arquivos a copiar logo depois do comando, como mostrado a seguir:

```
[aluno@ldalcero aluno]$ cp arq_1 arq_2 arq_3 arq_4 dir_1
```

Vale lembrar que o último nome na cadeia é o destino, ou seja, os arquivos arq_1, arq_2, arq_3 e arq_4 são copiados para o diretório dir_1.

3.6 Mover ou Renomear

As habilidades para mover e renomear arquivos no Linux são básicas para organizar informações no sistema. A seguir, será apresentado como fazê-lo utilizando o comando **mv**.

O formato básico do comando é mostrado a seguir:

```
mv arquivo destino
```

O comando **mv** é basicamente usado pra mover um arquivo dentro do sistema de arquivos do Linux.

```
[aluno@ldalcero aluno]$ mv documento.txt /tmp
```

O comando anterior move o arquivo documento.txt para o diretório /tmp. É possível também usar o comando para renomear arquivos, como mostrado no exemplo a seguir:

```
[aluno@ldalcero aluno]$ mv doc.txt documento.txt
```

Para não sobrescrever arquivos deve-se utilizar o parâmetro **-i**, como mostrado a seguir:

```
[aluno@ldalcero aluno]$ mv -i doc.txt documento.txt
mv: sobrescrever `documento.txt'? y
```

```
[aluno@ldalcero aluno]$ ls documento*
documento.txt  documento.txt~
```

Também é possível renomear um arquivo durante a movimentação do mesmo:

```
[aluno@ldalcero aluno]$ mv documento.txt /tmp/documento-2.txt
```

No exemplo a seguir será movido um diretório:

```
[aluno@ldalcero aluno]$ mv dir_1 dir_2
```

O comando acima move toda a árvore do dir_1 para dentro do dir_2. Caso o dir_2 não exista, o dir_1 será renomeado para dir_2.

3.7 Removendo Arquivos

O comando **rm** (remove) é usado para remover arquivos e diretórios. É possível remover vários arquivos simultaneamente, bastando para tal colocar o nome dos arquivos a remover, logo depois do comando. O formato básico do comando é mostrado a seguir:

```
rm [parametros] arquivo
```

Como primeiro exemplo será emitido um comando para apagar o arquivo documento.txt.

```
[aluno@ldalcero aluno]$ rm documento.txt
```

É também possível remover vários arquivos listados logo após o comando. Por exemplo:

```
[aluno@ldalcero aluno]$ rm documento.txt doc.txt documento-2.txt
```

A maneira mais segura de se usar o comando **rm** é com o parâmetro **-i**, ou seja, é solicitada uma confirmação para cada arquivo a apagar.

```
[aluno@ldalcero aluno]$ rm -i /tmp
```

No comando anterior, além de se usar o parâmetro **-i** foi também usado o parâmetro **-r** (recursivo), isto remove todos os arquivos do diretório **/tmp** de forma recursiva. Já o parâmetro **-i** irá pedir uma confirmação para cada arquivo a apagar.

Caso não seja necessária uma confirmação, pode-se forçar a remoção de toda a árvore de diretórios; para tal utiliza-se o parâmetro **-f**.

```
[aluno@ldalcero aluno]$ rm -rf /tmp
```

O que ocorre é a remoção total do diretório e de todos os seus subdiretórios.

Lembre-se de evitar o uso do comando **rm** desnecessariamente quando estiver trabalhando como root, ou seja, superusuário, para prevenir que arquivos necessários ao sistema sejam apagados acidentalmente.

É possível utilizar o comando com metacaracteres, conforme mostrado a seguir:

```
[aluno@ldalcero aluno]$ rm *.txt *.doc
```

3.8 Links

Links, são referências, atalhos ou conexões lógicas entre arquivos ou diretórios. Estas referências podem ser de dois tipos: Hard Links (diretas) ou Symbolic Links (simbólicas).

3.8.1 Hard Links

Ocupam apenas um inode na área de inodes. E são usados quando estas referências estiverem no mesmo sistema de arquivos.

A quantidade de links fazendo referência ao mesmo arquivo pode ser vista usando o comando **ls -l**. O valor que aparece após as permissões é o número de conexões lógicas.

Exemplo:

```
[aluno@ldalcer0 aluno]$ ls -l doc*  
  
-rwxr-xr-x 4 aluno grupo01 36720 Jun 2 14:25 documento.txt
```

Nesse caso, o arquivo documento.txt possui 4 links associados a ele. Quando qualquer um dos links é alterado, o original também é modificado; visto que são o mesmo arquivo, apenas com nomes diferentes. O original só será eliminado quando todos os seus links também forem. O formato do comando para criar um hard link é mostrado a seguir:

ln arquivo link

OBS.: Não é possível criar hard links para diretórios, e também é impossível criar links diretos entre sistemas de arquivos.

3.8.2 Links Simbólicos

O link simbólico é como um “atalho” para um arquivo. O ato de se apagar um link simbólico não faz com que o arquivo original desapareça do sistema, somente o vínculo será apagado.

Existem vários motivos para se criar um link simbólico, dentre eles pode-se destacar:

Quando se deseja criar nomes mais significativos para chamadas a comandos. Existe um exemplo prático na chamada dos shells tais como **csch** e **sh**, que na realidade são links simbólicos para os shells **tsch**, **bash**.

Um outro exemplo bastante significativo é com relação aos comandos **mtools**, tais como o **mformat** e vários outros, que nada mais são do que links simbólicos para o comando **mtools**.

Um outro uso para o comando é como atalho para diretórios com nomes complicados; o que não pode ser feito com hard links.

O formato do comando para criação de links simbólicos é:

ln -s arquivo link

Observe o exemplo a seguir:

```
[aluno@ldalcer0 aluno]$ ln -s /usr/X11R6/ /usr/X11
```

O comando anterior cria um link **/usr/X11** que aponta para o diretório real **/usr/X11R6**. Desta forma, é possível acessar este diretório por qualquer caminho.

Como se pode observar, a única diferença nos comandos entre a ligação simbólica e a ligação direta é o parâmetro **-s**.

4. Permissões de Arquivos

Este tópico trata do sistema de direitos de acesso a arquivos do Linux, incluindo também informações de como alterar estes direitos.

4.1 Conceitos

O sistema de arquivos do Linux possibilita que sejam atribuídos direitos de acesso diferenciados para os usuários do sistema. A cada arquivo ou diretório do sistema é associado um proprietário, um grupo e seus respectivos direitos de acesso, ou permissões. O método mais simples e comum de verificar estes atributos de um arquivo é através do comando **ls**, como exemplificado a seguir:

```
[aluno@ldalceru aluno]$ ls -l documento.txt
-rw-r--r-- 4 aluno grupo01 36720 Jun 2 14:25 documento.txt
```

O dono do arquivo no exemplo citado é o usuário aluno, e o grupo a que está relacionado o arquivo é o grupo01. A primeira informação retornada por esta listagem é um conjunto de caracteres, o qual indica o tipo do arquivo e as permissões de acesso ao mesmo. O caractere inicial indica o tipo de arquivo, e a tabela abaixo mostra os tipos de arquivos existentes:

Caractere	Tipo de arquivo
-	Arquivo regular
d	Diretório
l	Link simbólico
b	Dispositivos orientados a blocos (HD)
c	Dispositivos orientados a caracteres (modem, porta serial)
s	Socket (comunicação entre processos)
p	Pipe (comunicação entre processos)

Os demais nove caracteres, divididos em três grupos de três caracteres cada, definem as permissões do dono do arquivo, dos membros do grupo a que está relacionado o arquivo e de outros usuários, respectivamente. As permissões de acesso aos arquivos são representadas pela letras r, w e x, conforme detalhado na tabela abaixo:

Modo de acesso	Arquivo regular/especial	Diretório
Leitura (r)	Examinar conteúdo de arquivo	Listar arquivos do diretório
Escrita (w)	Modificar o conteúdo de arquivo	Alterar diretório
Execução (x)	Executar arquivo	Pesquisar no diretório

Observe novamente os atributos do arquivo anteriormente citado:

```
-rw-r--r-- 4 aluno grupo01 36720 Jun 2 14:25 documento.txt
```


Para este arquivo, o usuário aluno possui permissões de leitura e escrita, os membros do grupo grupo01, assim como os demais usuários possuem apenas permissões de leitura.

4.2 Utilizando o chmod

O comando **chmod** permite que se altere as permissões de um ou mais arquivos. Existem duas notações para se aplicar o comando: o modo simbólico e o octal. Somente o superusuário ou o dono do arquivo podem executar esta operação. Veja a sintaxe do comando abaixo:

```
chmod [opções] arquivos
```

Uma das opções mais usadas no **chmod** é a opção **-R** que permite que se altere recursivamente as permissões de arquivos e diretórios.

No modo simbólico, deve ser indicado quem será afetado (u, g, o, a) e qual, ou quais, permissões serão concedidas ou suprimidas conforme tabelas abaixo.

Notação simbólica do chmod

Símbolo	Descrição
u	Usuário, ou dono do arquivo
g	Grupo do arquivo
o	Outros usuários que não são os donos, nem estão cadastrados no grupo especificado
a	Afeta todos os anteriores (u, g, o)

Operadores

Operador	Descrição
+	Concede permissão(ões) especificada(s)
-	Remove permissão(ões)
=	Atribui somente estas permissão(ões) ao arquivo, retirando as que não se encontram explícitas

A segunda forma de alterar permissões consiste em definir uma seqüência de três algarismos octais. Ela é mais utilizada quando se deseja alterar permissões. Cada algarismo se refere a um grupo de permissões (u, g, o).

Para facilitar o seu entendimento, associamos um valor decimal com cada permissão, conforme tabela abaixo:

Permissões octais

Decimal Associado	Permissão
4	Leitura (read)
2	Escrita (write)
1	Execução (execute)

Para se obter o octal referente às permissões selecionadas, se deve executar uma operação de soma entre elas. Veja abaixo um exemplo de definição simbólica de permissões:

```
[aluno@ldalcero aluno]$ chmod u+rw, g+x documento.txt
```

Neste caso, são concedidas permissões de leitura e gravação ao dono, e execução ao grupo para o arquivo documento.txt.

Exemplo de definição octal de permissões:

```
[aluno@ldalcero aluno]$ chmod 651 documento.txt
```

Neste exemplo, será concedida permissão de leitura e gravação ao dono (rw- =4+2 conforme tabela de octais), leitura e execução para o grupo (r-x = 4+1), e execução para qualquer outro (--x = 1).

4.3 Permissões Padrão (umask)

O comando **umask** é o comando que define as permissões padrão dos arquivos quando são criados pelo usuário. Veja a sintaxe abaixo:

```
umask [opções] modo
```

O parâmetro modo informa as permissões que serão dadas ao usuário/grupo/outros. Ele pode ser informado de duas maneiras:

- Como um número octal (022)
- Como uma máscara semelhante à utilizada pelo comando **chmod** (u=rwx, g=rx, o=rx)

Desta maneira é possível controlar automaticamente as permissões dos arquivos que são criados pelo usuário.

4.4 Utilizando chown

Veja a sintaxe do comando abaixo:

```
chown [novo_proprietário][:novo_grupo] arquivos
```

O comando **chown** permite ao root a alteração do dono e do grupo relacionado ao arquivo, ou arquivos, selecionados. Em casos especiais o usuário pode alterar o grupo caso ele pertença tanto ao grupo de origem como ao grupo de destino.

Exemplos:

```
[aluno@ldalcero aluno]$ chown :grupo02 documento.txt
```

Altera o grupo do arquivo documento.txt para grupo02.

```
[aluno@ldalcero aluno]$ chown aluno:grupo02 documento.txt
```

Altera o dono do arquivo documento.txt para aluno e o grupo do mesmo para grupo02.

5. Gerenciamento de Contas de Usuários

Um bom gerenciamento do sistema, com o uso das ferramentas administrativas adequadas, torna-o estável e seguro, mantendo-o dentro dos padrões esperados em qualquer área de atuação.

5.1 Contas de Usuário

O Linux é um sistema operacional multiusuário, portanto é necessário que todos os usuários sejam cadastrados e tenham permissões de acesso diferenciados, É possível também cadastrá-los em grupos para facilitar o gerenciamento. Neste tópico serão abordados justamente estes aspectos do Linux e os comandos necessários para a administração do sistema.

Primeiramente será visto como é possível criar os usuários.

5.2 O comando `useradd` ou `adduser`

O comando `useradd` ou `adduser` permite que se crie usuários especificados em opções. Somente o superusuário poderá utilizar este comando. Veja abaixo a sua sintaxe:

```
useradd [opções] [usuário]
```

Este comando altera os seguintes arquivos:

- `/etc/passwd` – informações de contas de usuários
- `/etc/shadow` – informações de contas de usuários e senhas criptografadas
- `/etc/group` – informações de grupos

O comando `useradd`

Opção	Descrição
<code>-d dir</code>	Define em “dir” qual será o diretório pessoal do usuário. Este será, também, o diretório inicial daquele usuário.
<code>-g grupo</code>	Grupo ou número do grupo inicial, ao qual pertencerá. Na maioria das distribuições Linux é criado por padrão um grupo como mesmo nome do usuário.
<code>-s shell</code>	Informa qual o shell a ser usado no login.

Exemplo:

```
[aluno@ldalcero aluno]$ useradd -d /home/aluno01 aluno1
```

Cria o usuário `aluno1` e designa o diretório `/home/aluno01` como diretório pessoal deste.

```
[aluno@ldalcero aluno]$ useradd -g turma -d /home/aluno02 aluno2
```

Cria o usuário `aluno 2`, pertecendo ao grupo `turma` e designa `/home/aluno 02` como diretório pessoal.

5.3 O comando **passwd**

O comando **passwd** permite que se troque a senha de determinado usuário. O superusuário pode trocar a senha de qualquer outro. O usuário comum, porém, pode trocar somente a sua senha. As senhas são armazenadas no arquivo `/etc/shadow`, e as demais informações dos usuários são armazenadas no arquivo `/etc/passwd`.

Após a criação do usuário será necessário criar uma senha para este, caso contrário, não será permitido que este usuário faça login no sistema. Para tal, deve-se utilizar o comando **passwd**. Veja abaixo a sua sintaxe:

```
passwd [usuário]
```

Exemplos:

```
[root@ldalcero root]$ passwd aluno1
```

O superusuário irá alterar a senha do usuário `aluno1`.

```
[aluno@ldalcero aluno]$ passwd
```

O usuário `aluno1` irá alterar a sua senha.

5.4 O comando **userdel**

O comando **userdel** permite que se elimine usuários do sistema. Somente o superusuário poderá utilizar este comando. Veja abaixo sua sintaxe:

```
userdel [opções] [usuário]
```

Exemplo:

```
[aluno@ldalcero aluno]$ userdel aluno1
```

Elimina a conta do usuário `aluno1`.

5.5 O comando **groupadd** ou **addgroup**

Para facilitar a administração do sistema, pode-se usar o conceito de grupos de usuários com perfis semelhantes. Por exemplo, definir grupos conforme os departamentos de uma empresa. Para isto, precisa-se criar estes grupos através do comando **groupadd** ou **addgroup**.

Sintaxe:

```
groupadd [opções] grupo
```

Exemplos:

```
[root@ldalcero root]$ groupadd alunos
```

Cria o grupo `alunos`.

5.6 O comando gpasswd

O **gpasswd** é utilizado para administrar o arquivo `/etc/group` (e o arquivo `/etc/gshadow`, caso seja compilado com **SHADOWGRP**). Todos os grupos podem ter administradores, membros e uma senha. O administrador do sistema pode usar a opção `-A` para definir o administrador do grupo e `-M` para definir os membros e todos os seus direitos, assim como os do administrador. Veja a sintaxe abaixo:

```
gpasswd grupo
```

```
gpasswd -a usuário grupo
```

```
gpasswd -d usuário grupo
```

```
gpasswd -R grupo
```

```
gpasswd -r grupo
```

```
gpasswd [-A usuário,...] [-M usuário,...] grupo
```

Administradores de grupos podem adicionar e excluir usuários utilizando as opções `-a` e `-d` respectivamente. Administradores podem usar a opção `-r` para remover a senha de um grupo. Quando não há senhas definidas para um grupo, somente os membros do grupo podem usar o comando **newgrp** para adicionar novos usuários ao grupo. A opção `-R` desabilita o acesso ao grupo através do comando **newgrp**.

O **gpasswd** executado pelo administrador do grupo, seguido pelo nome, solicitará a senha do grupo. Caso o comando **newgrp** esteja habilitado para outros usuários do grupo sem o uso de senha, não-membros do grupo podem também executar o comando, informando, entretanto, a senha do grupo.

5.7 Verificando informações do usuário

Uma vez criados os usuários e grupos no sistema utilizamos o comando **id** para verificar informações sobre os usuários do sistema. Ele nos fornece dados como a identificação do usuário no sistema (UID) e os grupos aos quais o usuário está associado (GID). Veja a sintaxe abaixo:

```
id [opções] [nome]
```

Utilizado sem nenhuma opção o comando **id** retorna os dados do usuário corrente.

```
[aluno@ldalcero aluno]$ id
uid = 790(aluno) gid = 790(aluno) grupos = 790(aluno)
```

Quando informamos o nome de um usuário como opção, ele nos retorna as informações do usuário indicado.

```
[aluno@ldalcero aluno]$ id root
uid = 0(root) gid = 0(root) grupos = 0(root), 1(bin), 2(daemon), 3(sys), 4
(adm), 6(disk), 10(wheel)
```

6. Gerenciamento de Processos

O Linux, por ser um sistema operacional multitarefa, executa diversos processos simultaneamente. De um modo geral, um processo é um programa em execução, embora possa se apenas parte de um programa mais complexo.

6.1 Criar, Monitorar e Eliminar Processos

Para cada processo, o sistema fornece um código (PID) que o identificará. Há também uma hierarquia de processos, ou seja, um processo pode chamar outro, que por sua vez chama um terceiro, e assim por diante.

O processo que executa um segundo processo, é chamado de processo “pai”, e o novo processo, chamado de “filho”.

Como já foi dito, o Linux é um sistema multitarefa, o que lhe permite executar mais de um processo ou sistema. Portanto, é possível ter processos em primeiro plano (foreground), bem como em segundo plano (background). Os comandos para se trabalhar com processos serão vistos a seguir.

6.2 O comando ps

O comando **ps** mostra os processos ativos no sistema. Veja abaixo a sintaxe do comando:

ps [opções]

Opção	Descrição
-a	Mostra todos os processos.
-f	Mostra os processos em formato de “árvore”, ou seja, com seus caminhos completos.
-u	Inclui na lista os usuários e hora do início do processo.
-x	Inclui processos não associados a um terminal.

Exemplos:

```
[aluno@ldalcero aluno]$ ps
```

Exibe os processos ativos daquele usuário.

```
[aluno@ldalcero aluno]$ ps -xf
```

Mostra todos os processos do usuário, incluindo processos sem controle do terminal, no formato “árvore”.

6.3 O comando kill

O comando **kill** permite que o superusuário ou os donos dos processos possam eliminar processos ativos. A sintaxe desse comando é apresentada a seguir:

```
kill [opções] [-sinal] [processo]
```

Exemplos:

```
[aluno@ldalcero aluno]$ kill -l
```

Este comando mostra os sinais que se pode enviar para um processo.

```
[aluno@ldalcero aluno]$ kill -9 907
```

Este comando acima elimina o processo (PID) 907 enviando um sinal 9.

Todos os sinais passam pelo kernel. Quando é enviado qualquer sinal para um processo este sinal vai para o kernel que o envia ao processo de destino.

O sinal 9(SIGKILL), quando enviado para um processo, termina a sua execução. Na realidade, o que acontece é que o kernel, quando recebe o sinal, não o repassa para o processo, e sim, remove o processo diretamente.

O sinal 20(SIGTSTP) suspende a execução de um programa. Quando existe um programa em execução em foreground e deseja-se suspender o seu processamento basta enviar esse sinal; pode-se usar as teclas **Ctrl-z**.

Se for utilizado o comando sem o parâmetro -sinal, será enviado ao processo o sinal SIGTERM ou terminate possivelmente terminando sua execução.

6.4 O comando top

O comando **top** mostra uma lista (atualizada periodicamente) dos processos ativos no sistema. Esta lista é ordenada por consumo de recursos de CPU, veja a sintaxe abaixo:

```
top [opções]
```

Exemplos:

```
[aluno@ldalcero aluno]$ top -d1 -i
```

Este comando exhibe a lista dos processos, por ordem de consumo de recursos de CPU, exceto os ociosos ou “zombies”, e atualiza a lista a cada segundo.

6.5 O comando jobs

O comando **jobs** exhibe os processos parados ou em execução que se encontram em segundo plano. Veja abaixo a sintaxe:

```
jobs [opções]
```

Exemplos:

```
[aluno@ldalcero aluno]$ jobs -l
```

O comando acima exhibe os trabalhos em segundo plano, com seus respectivos PIDs.

6.6 O comando **bg**

Veja a sintaxe do comando:

```
bg [jobId]
```

Como foi visto anteriormente é possível parar um processo digitando **CTRL-Z**. Para reativar este processo e o colocar em segundo plano use o comando **bg %JobId**.

Exemplos:

```
[aluno@ldalcer0 aluno]$ bg %3
```

O comando acima coloca em segundo plano o terceiro **job**.

6.7 O comando **fg**

Ao contrário do comando **bg**, o **fg** (foreground) coloca o job em primeiro plano. Sintaxe:

```
fg [jobId]
```

Exemplos:

```
[aluno@ldalcer0 aluno]$ fg %2
```

Coloca em primeiro plano o **job** número 2.

7. Compactação e Empacotamento de Arquivos

Este tópico irá apresentar os comandos para compactação de arquivos mais utilizados, assim como o comando **tar**, o qual pode ser utilizado para efetuar backup do sistema.

7.1 Compactadores de Arquivos

O Linux tem diversos aplicativos para compressão de arquivos, sendo que cada um deles utiliza um algoritmo de compressão diferente. Serão abordados, neste tópico, os mais usados.

7.1.1 GZIP

O comando **gzip** é usado para a compactação/descompactação de arquivos.

O arquivo original é substituído por um arquivo compactado com a extensão **.gz**, mantendo o dono, permissões e datas de modificação.

Caso nenhum arquivo seja especificado, ou se o nome do arquivo for "-", a entrada padrão será compactada na saída padrão. O **gzip** somente tentará compactar arquivos normais; em particular, ele ignorará links simbólicos.

O formato básico deste comando é mostrado a seguir:

```
gzip [opções] arquivos  
gunzip [opções] arquivos
```


Como primeiro exemplo, o arquivo será compactado:

```
[aluno@ldalcero aluno]$ gzip arquivo
```

No exemplo a seguir, o arquivo será descompactado:

```
[aluno@ldalcero aluno]$ gunzip arquivo.gz
```

7.1.2 BZIP2

O comando **bzip2** é usado para a compactação/descompactação de arquivos.

O arquivo original é substituído por um arquivo compactado com a extensão `.bz2`, mantendo o dono, permissões e datas de modificação.

Caso nenhum arquivo seja especificado, ou se o nome do arquivo for "-", a entrada padrão será compactada na saída padrão. O formato básico deste comando é indicado a seguir:

```
bzip2 [opções] arquivos  
bunzip2 [opções] arquivos.bz2
```

Como primeiro exemplo, o arquivo será compactado:

```
[aluno@ldalcero aluno]$ bzip2 arquivo
```

No exemplo a seguir, o arquivo será descompactado:

```
[aluno@ldalcero aluno]$ bunzip2 arquivo.bz2
```

7.1.3 Compress

O comando **compress** é usado para a compactação/descompactação de arquivos. O arquivo original é substituído por um arquivo compactado com a extensão `.Z`, mantendo o dono, permissões e datas de modificação.

Caso nenhum arquivo seja especificado, ou se o nome do arquivo for "-", a entrada padrão será compactada na saída padrão. O formato básico deste comando é indicado a seguir:

```
compress [opções] arquivos  
uncompress [opções] arquivos.Z
```

Como primeiro exemplo, o arquivo será compactado:

```
[aluno@ldalcero aluno]$ compress arquivo
```

No exemplo a seguir, o arquivo será descompactado:

```
[aluno@ldalcero aluno]$ uncompress arquivo.Z
```

7.1.4 ZIP

O comando **zip** é usado para a compactação e empacotamento de arquivos. Ele é compatível com o **pkzip** e similares.

O programa é útil para empacotamento de uma série de arquivos para distribuição, para arquivamento e para economizar espaço em disco temporariamente, compactando arquivos e diretórios sem uso. Veja a seguir a sintaxe do comando:

```
zip [opções] arquivo_zip arquivos
```

Como primeiro exemplo, os arquivos do diretório corrente serão compactados e empacotados e colocados no arquivo:

```
[aluno@ldalcero aluno]$ zip arquivo *
```

No exemplo a seguir, o arquivo será descompactado e desempacotado:

```
[aluno@ldalcero aluno]$ unzip arquivo
```

7.2 O comando tar

Permite copiar arquivos e depois restaurá-los, para efeito de backup de segurança e/ou transporte de dados através de um meio físico. Pode ser utilizado em máquinas diferentes, pois é padrão no ambiente UNIX. Também permite realizar cópias multivolume. Veja a sintaxe:

```
tar [opções] arquivos
```

O comando **tar** permite que se crie, atualize ou recupere backups do sistema segundo as opções.

A tabela a seguir mostra as principais opções do comando **tar**.

Opção	Descrição
-c	Cria um arquivo tar.
-r	Acrescenta novos arquivos no arquivo tar.
-x	Extraí conteúdo de um arquivo tar.
-t	Gera uma lista dos arquivos.
-p	Restaura os arquivos no seu modo de permissão original.
-m	Usa a hora de extração do arquivo como a hora da última alteração.
-v	Modo detalhado.
-T	Obtém o nome dos arquivos a incluir ou extrair a partir de “arquivo-lista”.
-f	Especifica o arquivo a ser usado como entrada ou saída.
-C	Extraí os arquivos para o diretório especificado.
-z	Compacta/descompacta os arquivos através do programa gzip .
-j	Compacta/descompacta os arquivos através do programa bzip .

Essas opções podem ser usadas conjuntamente, alguns exemplos são mostrados a seguir.

Exemplos:

```
[aluno@ldalcero aluno]$ tar -cvf /dev/fd0 /home
```

Grava os arquivos do diretório /home para um disquete.

```
[aluno@ldalcero aluno]$ tar -xvpf arquivo.tar
```

Extrai todos os arquivos do arquivo arquivo.tar, mantendo as permissões originais.

8. Instalação de Aplicativos

Neste tópico serão mostrados meios de atualizar o sistema, instalar novos pacotes e gerenciar os pacotes que já estão instalados no sistema.

8.1 Gerenciamento de Pacotes RPM

Esta seção aborda o RPM, que é um poderoso gerenciador de pacotes. Após sua conclusão, o aluno estará apto a instalar, verificar, atualizar e desinstalar pacotes de softwares.

8.1.1 Conceitos Básicos

O RPM disponibiliza uma série de implementações que facilitam a manutenção do sistema. A instalação, desinstalação e atualização de um programa que esteja no formato de um pacote RPM podem ser feitas através de um único comando, sendo que o gerenciador cuidará de todos os detalhes necessários ao processo.

Para desenvolvedores, o RPM permite manter fontes e binários e suas atualizações separadamente, empacotando-os de forma configurável para os usuários.

O gerenciador mantém uma base de dados com os pacotes instalados e seus arquivos, o que permite executar pesquisas complexas e verificações de maneira ágil e segura. Durante atualizações de programas, ele administra, por exemplo, arquivos de configuração, mantendo as configurações já realizadas no sistema.

8.1.2 Vantagens da utilização do RPM

- Atualização de Softwares

Com o gerenciador de pacotes é possível atualizar componentes individuais do sistema, sem a necessidade de reinstalação total. Os arquivos de configuração são mantidos durante o processo, não se perdendo assim uma personalização já efetuada.

- Pesquisas

O RPM faz pesquisas sobre os pacotes já instalados e seus arquivos, É possível pesquisar a que pacote pertence determinado arquivo e qual a sua origem.

- Verificação do sistema

Caso algum arquivo importante de algum pacote tenha sido removido, pode-se inicialmente verificar se o pacote apresenta alguma anormalidade.

- Código fonte

O RPM possibilita a distribuição dos arquivos fonte, idênticos aos distribuídos pelos autores dos programas e as alterações separadamente.

8.1.3 Instalando / Atualizando

Os pacotes RPM têm nomes de arquivos com o seguinte formato:

```
foo-1.0-1.i386.rpm
```

Onde:

foo : nome do arquivo

1.0 : versão

1 : Release

i386 : plataforma

A instalação é feita através de uma única linha de comando, como por exemplo:

```
[root@ldalcero root]$ rpm -ivh foo-1.0-1.i386.rpm
Preparing... ##### [100%]
 1: foo      ##### [100%]
```

Como se pode observar, o RPM apresenta o nome do pacote (o qual não tem necessariamente o mesmo nome do programa) e apresenta uma sucessão de caracteres # atuando como uma barra de status no processo de instalação.

Para atualizar um pacote utilize o comando:

```
[root@ldalcero root]$ rpm -Uvh foo-1.0-1.i386.rpm
Preparing... ##### [100%]
 1: foo      ##### [100%]
```

Qualquer versão anterior do pacote será desinstalada e será feita uma nova instalação guardando as configurações anteriores do programa para um possível uso caso o formato dos arquivos de configuração não tenha sido alterado.

8.1.4 Desinstalando

Para desinstalar um pacote utilize o comando:

```
[root@ldalcero root]$ rpm -e foo
```

Onde foo é o nome do pacote e não do arquivo utilizado na instalação.

8.1.5 Consultando / Verificando

Para consultar a base de dados de pacotes instalados utilize o comando:

```
[root@ldalcero root]$ rpm -q nome_do_pacote
```

Com a sua utilização são apresentados o nome do pacote, versão e release. Em vez de especificar o nome do pacote, pode-se utilizar as seguintes opções após o parâmetro q, mostrados na tabela abaixo:

Opção	Descrição
a	Consulta todos os pacotes instalados.
f	Consulta o pacote do qual o arquivo faz parte
i	Apresenta as informações do pacote.
l	Apresenta a lista de arquivos pertencentes ao pacote.
s	Apresenta o status dos arquivos do pacote.
d	Apresenta um lista dos arquivos de documentação.
c	Apresenta uma lista dos arquivos de configuração.

A opção de verificação pode ser útil caso haja alguma suspeita de que a base de dados RPM esteja corrompida.

8.2 Instalação de Aplicativos em código binário

Na maioria das vezes os aplicativos em código binário são empacotados pelo comando tar e dentro dele é necessário fazer três tipos de comando, para que o aplicativo seja instalado no sistema, a seguir vamos ver detalhadamente esses comandos.

8.2.1 O comando configure

Esse comando é utilizado para que seja feita uma verificação no sistema e nas dependências dos pacotes necessários para o aplicativo ser instalado, se o comando obtiver algum erro de checagem do sistema ele retorna uma mensagem de erro e para a verificação.

Exemplo:

```
[root@ldalcero foo]$ ./configure
```

8.2.1 O comando make e make install

Esses dois comandos são responsáveis pela criação das configurações necessárias ao aplicativo no sistema, bem como sua instalação, eles só podem ser utilizados sem problemas, caso o comando configure tenha sido executado com sucesso.

Exemplos:

```
[root@ldalcero foo]$ make  
[root@ldalcero foo]$ make install
```

9. Uso e Configuração do Ambiente Gráfico

Neste tópico será mostrado como você interagir com o ambiente gráfico do Linux e também como fazer a sua configuração de acordo com sua necessidade e facilidade. Como padrão vamos utilizar o ambiente gráfico GNOME.

9.1 Gerenciadores de Janelas

A principal função de um gerenciador de janelas é, como o próprio nome diz, gerenciar a apresentação das janelas e fornecer métodos para controlar aplicações, criar e acessar menus. Além de fornecer meios para que o usuário possa personalizar o seu ambiente. Mas, como é feito o relacionamento com o sistema?

No Linux, este relacionamento é feito pelo Servidor de Janelas X. Seu objetivo é fornecer acesso aos dispositivos existentes em seu computador (mouse, teclado) e fornecer um ambiente agradável para a manipulação de aplicações, através de componentes chamados Janelas.

Mas, não confunda estes dois conceitos: o servidor de janelas possui recursos para implementar as aplicações em forma de janelas e formar um ambiente agradável para o usuário; já o gerenciador de janelas vai fornecer métodos para que o usuário possa modificar o tamanho das janelas, o papel de parede, enfim, o layout da interface gráfica.

9.2 Usando a Área de Trabalho

Usar a Área de Trabalho é tão simples como arrastar itens que se queira utilizar rotineiramente. A Área padrão inclui uma pasta do diretório pessoal (/home/nome_do_usuario).

Todos os itens armazenados na Área de Trabalho estão localizados no diretório:

```
/home/nome_do_usuario/.gnome-desktop/
```

Isto pode ser útil para lembrar quando se desejar que um item que apareça na Área de Trabalho, porém com as quais não se pode utilizar as funções de arrastar e soltar.

No painel inferior da Área de Trabalho, você encontra ícones, que na verdade são atalhos para acessar alguns aplicativos e para acessar o menu de programas.

9.3 Gerenciador de Arquivos

Se você der um clique no ícone que se encontra na Área de Trabalho com o nome de Pessoal ou Pasta de Início ou algo semelhante, irá abrir um programa que nada mais é do que o seu gerenciador de arquivos, no qual abrirá uma janela mostrando o conteúdo do seu diretório pessoal. Este é o diretório onde estão armazenados todos os seus documentos, e é permitido ser acessado somente por você. Uma janela se compõe de várias partes. Na parte superior está a Barra de Título, a qual mostra o título do programa em execução e o documento no qual você está trabalhando. Logo abaixo da Barra de Título está a Barra de Menu, onde se encontra as guias: Arquivo, Editar e assim sucessivamente, clicando em cada um delas aparecerá vários elementos de acesso com diversas funções diferentes. Logo abaixo da Barra de Título está a Barra de Ferramentas, que consiste de um ou mais ícones, cada um equivalente a um elemento

de menu, pode-se dizer que são atalhos para os elementos de menu mais usados no programa. A Barra de Status se encontra na parte inferior e mostra informações do arquivo que está sendo acessado no momento.

9.4 Adicionando Ícones na Área de Trabalho

Para adicionar ícones na sua Área de Trabalho você pode fazer de duas maneiras:

1. Escolhendo qual programa que deseja fazer o atalho, e arrastá-lo até a Área de Trabalho e soltar, será perguntando se deseja criar um atalho para o arquivo ali.
2. Clicando com o botão direito do mouse na Área de Trabalho e escolher a opção criar novo lançador, e adicionar as informações necessárias para o referente atalho do programa.

9.5 Usando a Lixeira

Em circunstâncias normais, apagar um arquivo no UNIX é uma operação sem retorno. No entanto, com o gerenciador de arquivos do GNOME, você pode escolher Mover para a Lixeira ao invés de Apagar. Isto irá mover o arquivo para a Pasta Lixeira, que por padrão é acessível via ícone em sua Área de Trabalho. Nesta pasta, sempre é possível recuperar arquivos apagados. Lembre-se de esvaziá-la regularmente clicando nela com o botão direito do mouse, e então escolhendo Esvaziar Lixeira, sob pena de ficar sem espaço em disco, pois os arquivos que estão na Lixeira continuam ocupando espaço. Note que, uma vez esvaziada a Lixeira, os arquivos que estavam lá estarão perdidos para sempre.

9.6 Configurando a Área de Trabalho

Existem várias maneiras de configurarmos a Área de Trabalho, essas configurações incluem:

- Tela de Fundo;
- Proteção de Tela;
- Tema da Área de Trabalho;
- Entre outras.

Podemos configurar nossa Área de Trabalho, através do Menu Preferências -> Centro de Controle, abrirá uma janela com várias ícones, correspondentes cada um a uma configuração de sua Área de Trabalho.

9.7 Trocando a Configuração do seu Teclado

Vista a grande quantidade de layouts (idiomas) de teclados que existe hoje no mercado de informática, iremos abordar neste tópico, como você pode configurar o layout do seu teclado.

Existem duas maneiras de configurar o seu teclado, pode ser via modo texto ou via modo gráfico.

- Modo Texto

Digite no terminal o seguinte comando:

```
kbdconfig
```

Observe que o comando pode variar de uma distribuição para outra, as outras variações do comando são: `keybconfig` ou ainda `keyboardconfig`.

- Modo Gráfico

Vá até o Menu de Configurações do Sistema, e em seguida na opção Teclado, na janela que será mostrada escolha o idioma do seu teclado.

10. Navegação na Internet

Neste tópico veremos o Mozilla que é uma suíte de acesso à Internet, composto por navegador, leitor de email e um livro de endereços.

10.1 Conhecendo o Mozilla

O Mozilla, sendo um programa gratuito e de código aberto, é uma alternativa viável de navegador ("browser" para acessar a Internet). Como outros programas conta, no seu desenvolvimento, com o auxílio de muitas pessoas, em todo o mundo, que contribuem para o controle de qualidade do navegador, que o copiam, testam as principais versões e sugerem melhorias.

O Mozilla pode ser usado sozinho, mas nada impede que seja usado simultaneamente com outro navegador, pois as suas configurações são independentes. Note-se que no caso de usar dois programas, a escolha de qual navegador deve ser o padrão do sistema fica a critério do usuário.

A instalação do Mozilla cria ícones novos: na tela, (um dinossauro vermelho) ao lado do "Botão Iniciar" e ao lado do relógio.

10.2 Navegando com o Mozilla

Para abrir o programa deve-se dar um duplo clique ícone do dinossauro que aparece ao lado do botão "Iniciar" ou no ícone que aparece na tela.

- Navegação com abas

O Mozilla possibilita abrir várias páginas na mesma janela, em diferentes abas. Ou seja, um "site", pode ficar, todo, dentro de uma única janela.

- Como abrir uma nova aba

Para abrir um link em uma nova aba:

- Clicar nele com o botão direito do mouse e, no menu que aparece, selecionar Abrir em nova aba.
- Clicar no link mantendo pressionada a tecla Ctrl. Ou Selecionar Aba no submenu Novo do menu "Arquivo" (ou pressionar as teclas CTRL-T. Veja opções de configurações no painel de preferências:"Editar" - "Preferências" - "Mozilla Navigator" – "Abas".

- Gerenciador de downloads ("Download Manager")

A janela do gerenciador de downloads fornece indicadores de progresso na janela para cada arquivo que está sendo copiado e, também, possibilita ver todos os arquivos que já foram copiados, sem necessitar abrir o gerenciados de arquivos (Windows Explorer). Notar que o Mozilla começa a cópia logo que o link é clicado, iniciando o download em segundo plano, já enquanto se está selecionando o local onde o arquivo será salvo.

- Aparência

Como escolher as fontes

(Em "Editar" - "Preferências" - "Aparência") clicar em "Fontes" Escolher as fontes que desejar.

Como escolher as cores

(Em "Editar" - "Preferências" - "Aparência") clicar em "Cores". Pode-se escolher entre ver as cores do texto e dos diversos tipos de links - como foi determinado pelo autor. Embaixo, em "Aplicar cores" selecionar. "Usar as cores especificadas pela página" sempre da mesma forma: Escolher as cores que desejar para cada um deles. Embaixo, em "Aplicar cores" selecionar "Sempre usar as cores padrão".

11. Envio e Recebimento de E-Mails

Neste tópico iremos ter uma introdução do uso de correio eletrônico no Linux.

11.1 Usando o Evolution

Iremos usar como programa de correio eletrônico o Ximian Evolution, um dos mais usados pelas distribuições Linux e também bastante similar ao Outlook da Microsoft. O uso do Evolution é muito simples, sua janela é bastante interativa e de fácil entendimento.

11.2 Configurando Contas

Vamos aprender como configuramos uma conta de correio POP no Evolution:

- 1 – Vá em “Ferramentas” - “Configurações de Correio”;
- 2 – Na tela que será apresentada clique no botão “Adicionar”;
- 3 – Siga as instruções das telas seguintes e preencha todos os campos.

OBS.: Lembre-se é necessário que você saiba quais são as configurações do seu servidor de E-Mail, por isso recomendo que antes de iniciar a configuração do Evolution, você entre em contato com seu provedor ou com o administrador do seu sistema.

11.3 Enviando e Recebendo E-Mails

Para você receber os seus e-mails, ou seja, fazer o download das mensagens que estão no seu servidor de e-mail, basta você clicar no botão “Enviar/Receber” ou na guia “Ações” e depois na opção “Enviar/Receber” ou ainda apertando a tecla F9 do seu teclado.

Para você enviar um E-mail, você deve antes de tudo compor este e-mail, para isto vá na guia “Arquivo” e depois na opção “Novo” - “Mensagem de correio” ou clique no botão “Nova mensagem” ou ainda aperte as teclas “Ctrl + N” . Aparecerá uma janela, preencha os campos obrigatórios dessa janela para se enviar um e-mail, que são eles:

Para: Endereço do destinatário, para quem você quer enviar a mensagem;

Assunto: Digite aqui o assunto do seu e-mail.

Você também pode anexar arquivos ao e-mail clicando no botão “Anexar”.

Para enviar a mensagem clique no botão “Enviar”, se você preferir pode salvar a mensagem para posteriormente a enviá-la, para tal clique na guia “Arquivo” e depois na opção “Salvar”.

12. Mensagens Instantâneas

Quem navega pela internet provavelmente já deve ter ouvido falar em salas de bate-papos, programas de irc(chat) e a última febre da internet os programas de mensagens instantâneas, com várias opções conversação, desde via teclado, até via web cam. Neste tópico iremos abordar dois dos mais usados programas de mensagens instantâneas para Linux.

12.1 LICQ

O LICQ é um clone do ICQ escrito na sua quase totalidade em C++. Ele foi uma das febres que acometeu a Internet nos últimos tempos, sendo usado por milhões de pessoas no mundo inteiro. Esse programa permite a transmissão de arquivos, envio de páginas web, envio de mensagens e o famoso chat, ou seja, a conversa entre duas ou mais pessoas.

Os nomes de usuários de sua lista aparece com cores e ícones diferentes. Há, ainda, a possibilidade de se usar skins, que nada mais são do que imagens e sons que mudam a aparência do seu LICQ.

Vejam sua utilização básica:

Começaremos criando um novo usuário. Quando você inicia o LICQ pela primeira vez, você tem a opção de criar um novo usuário ou utilizar um usuário já existente. Para criar um novo usuário você deve clicar em Próximo e a seguir digitar sua senha e confirmar sua senha. Caso você já tenha um UIN(Universal Internet Number), pode clicar em Registrar como Usuário Existente e informar seu UIN e senha. Continuando, clique em Próximo e em seguida em Finalizar. Agora que você já está cadastrado e faz parte da família ICQ, vejamos como cadastrar os seus contatos.

Existem diversas maneiras de se cadastrar um contato. A mais simples dela é conhecendo o UIN do seu contato. Caso você conheça o UIN de algum amigo, basta clicar em Sistema -> Funções do Usuário -> Adicionar usuário. Surgirá uma janela pedindo o número do UIN que você deseja acrescentar á sua lista. Informe o UIN e clique em OK que este usuário será adicionado à sua lista de contatos. Uma outra maneira de adicionar contatos à sua lista, é procurá-los por alguma das informações a seguir: Alias, Primeiro Nome, Último Nome, Email.

Vejam agora como mandar uma mensagem para o seu contato. Das diversas maneiras que existem de se mandar uma mensagem para um contato, a mais fácil é clicar duas vezes sobre o nick dele, a janela de conversação será aberta, daí é só digitar a mensagem e enviá-la.

12.2 GAIM

Uma segunda opção de programa para mensagens instantâneas é o GAIM, ele utiliza diversos protocolos de conexão com servidores de mensagens instantâneas, dentre eles o ICQ e a mais nova mania da internet o MSN Messenger, onde em pouco tempo já conquistou diversos usuários no mundo, através do GAIM podemos nos conectar tanto nos servidores do ICQ, quanto nos servidores do MSN.

Para registrar um usuário no GAIM, clique em Contas e na janela que aparecer, no botão Adicionar. Na janela que irá ser aberta será pedido os dados do usuários, assim como o tipo de servidor que deseja usar: ICQ, MSN, Yahoo, e etc. Insira as informações do seu usuário de acordo com o tipo de servidor escolhido e aproveite.

13. Utilização do IRC (Chat)

Muito usado por internautas do mundo inteiro os programas baseados em IRC, une amigos, colegas ou pessoas distintas em foruns de bate-papo. Existem vários softwares para este tipo de bate-papo, neste tópico iremos abordar um dois mais usados pelas distribuições Linux, o X-Chat.

13.1 Como utilizar programas baseados em IRC

O X-Chat é um aplicativo gráfico de IRC(Internet Relay Chat), usado para conexão com servidores de Chat ao redor do mundo. Esse aplicativo é utilizado para “conversar” com outros usuários. Para se utilizar o programa e entrar na rede de bate-papo, você precisará de algumas informações básicas, são elas:

- Nickname: Um apelido pelo qual você será reconhecido pelos outros usuários da rede;
- Servidor: Máquina remota, onde você se conectará para se comunicar com os usuários dessa rede. Por padrão o X-Chat já traz uma lista de servidores configurados para serem usados;
- Canal: Também chamado de “Sala”, será onde você entrará para conversar com os usuários que se encontram dentro deste canal.

Uma vez conectado em um servidor você pode entrar em vários canais simultaneamente e conversar com diversas pessoas de variados canais.

14. Impressão de Arquivos

Vamos abordar neste tópico a impressão de arquivos, levando-se em conta que a impressora já tenha sido configurada no computador pelo administrador do sistema e esta esteja funcionando corretamente.

14.1 Imprimindo Arquivos

O método mais usado pelos aplicativos do Linux para a impressão é o Ghost Script. O Ghost Script (chamado de gs) é um interpretador do formato Pos Script (arquivos .ps) e pode enviar o resultado de processamento tanto para a tela como impressora. Ele está disponível para diversas plataformas e sistemas operacionais além do Linux, inclusive o DOS, Windows, OS/2, etc. O formato .ps esta se tornando uma padronização para a impressão de gráficos em Linux devido a boa qualidade da impressão, liberdade de configuração, gerenciamento de impressão feito pelo gs e por ser um formato universal, compatíveis com outros sistemas operacionais.

A maioria das distribuições Linux já vem com o pacote do Ghost Script instalado e configurado, e quando você manda o comando de impressão para a sua impressora, através de qualquer software, desde editores de textos até editores de imagens, o Linux já faz todo o trabalho de chamar o Ghost Script e comandar para que sua impressão saia correta na impressora, por isso o único trabalho que você tem ao imprimir seus arquivos é clicar no botão “Imprimir” do software que você estiver utilizando.

14.2 Filas de Impressão

Impressoras são acessadas pelo Linux através de um mecanismo de armazenamento temporário, como é comum ocorrer em sistemas multitarefas, ou sejam, as tarefas de impressão ficam armazenadas temporariamente em um arquivo e são processadas posteriormente por um segundo programa de controle denominado servidor de impressão. Esta sistemática garante que muitos usuários podem enviar tarefas de impressão simultaneamente para uma única impressora sem o risco de conflitos. Após enviar a tarefa para impressão, o usuário pode continuar com as suas atividades normais, enquanto a tarefa ficará na fila ou será impressa.

Uma fila de impressão consiste em:

- uma entrada no arquivo `/etc/printcap`, onde a fila será criada;
- um diretório, normalmente sob `/var/spool/lpd`, onde ficarão armazenados arquivos de tarefas de impressão, arquivos de dados, arquivos de controle de configuração de impressoras e filas.

Cada fila é processada por somente uma impressora, porém é possível haver diversas filas para a mesma impressora.

O processamento de uma tarefa de impressão, enviada por uma estação remota ou local, é realizado em três passos:

- os arquivos de dados são copiados no diretório da fila de impressão e associado com um arquivo de controle que é criado naquele momento;
- os arquivos de dados passam por um programa de filtros que os converte para um formato específico de impressão (por exemplo Ghost Script). Esta conversão pode ser feita em diversos passos individuais;
- o arquivo específico para a impressora conectada àquela fila é impresso.