

# Linux Básico



Tulio Marcus R. Calixto  
**Téc. Informática**

## **Linux básico**

Este curso tem como objetivo apresentar o linux e seus diversos comandos e estrutura de funcionamento. O material de apoio contém:

- Introdução
- Características
- Estrutura básica
- Funcionamento
- Estrutura de diretório
- Comandos para manipulação de arquivos
- Alguns comando unix
- Permissões para acesso a arquivos
- Manipulação de diretorios
- Outros comandos unix
- Gerenciamento de usuários e grupos
- Processos e Deamons
- Mais comandos unix
- Montar e desmontar volumes
- Configuração da rede
- Configuração do apt
- Utilizando o dpkg
- Serviços de rede
- Compilando e instalando pacotes

## Introdução ao sistema operacional Linux

**Sistema operacional** é um conjunto de programas (softwares) responsável pelo gerenciamento do hardware (parte física do computador).

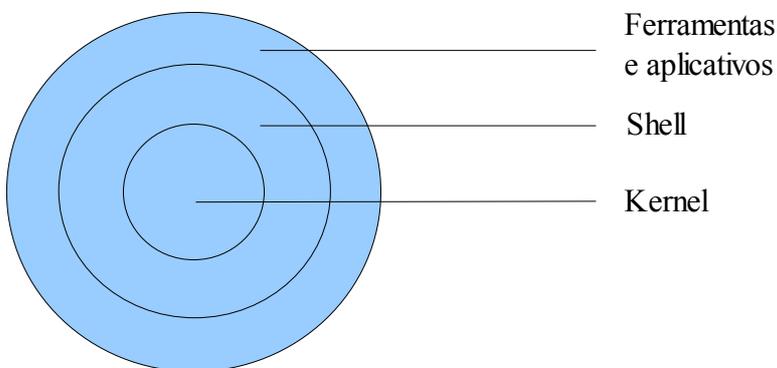
É dever do sistema operacional controlar a memória do computador para que um programa não invada a área de memória reservada para outro programa, por exemplo. Gerenciar os discos armazenando a informação de uma forma que ela possa ser recuperada posteriormente, controlar as placas de rede, som, etc também são funções do sistema operacional.

**UNIX** é um sistema operacional desenvolvido por Bell Laboratories, projetado em 1969 e tinha como objetivo propiciar um ambiente no qual os programadores pudessem criar programas.

### Características do UNIX

- Capacidade multitarefa: executa várias tarefas 'ao mesmo tempo';
- Capacidade multiusuário: vários usuários podem utilizar o computador ao mesmo tempo;
- Portabilidade: Compatível com vários modelos de computadores e com diferentes arquiteturas de hardware;
- Sistema modularizado;
- Possui uma vasta biblioteca de softwares aplicativos;
- Case sensitive.

### Estrutura básica do UNIX



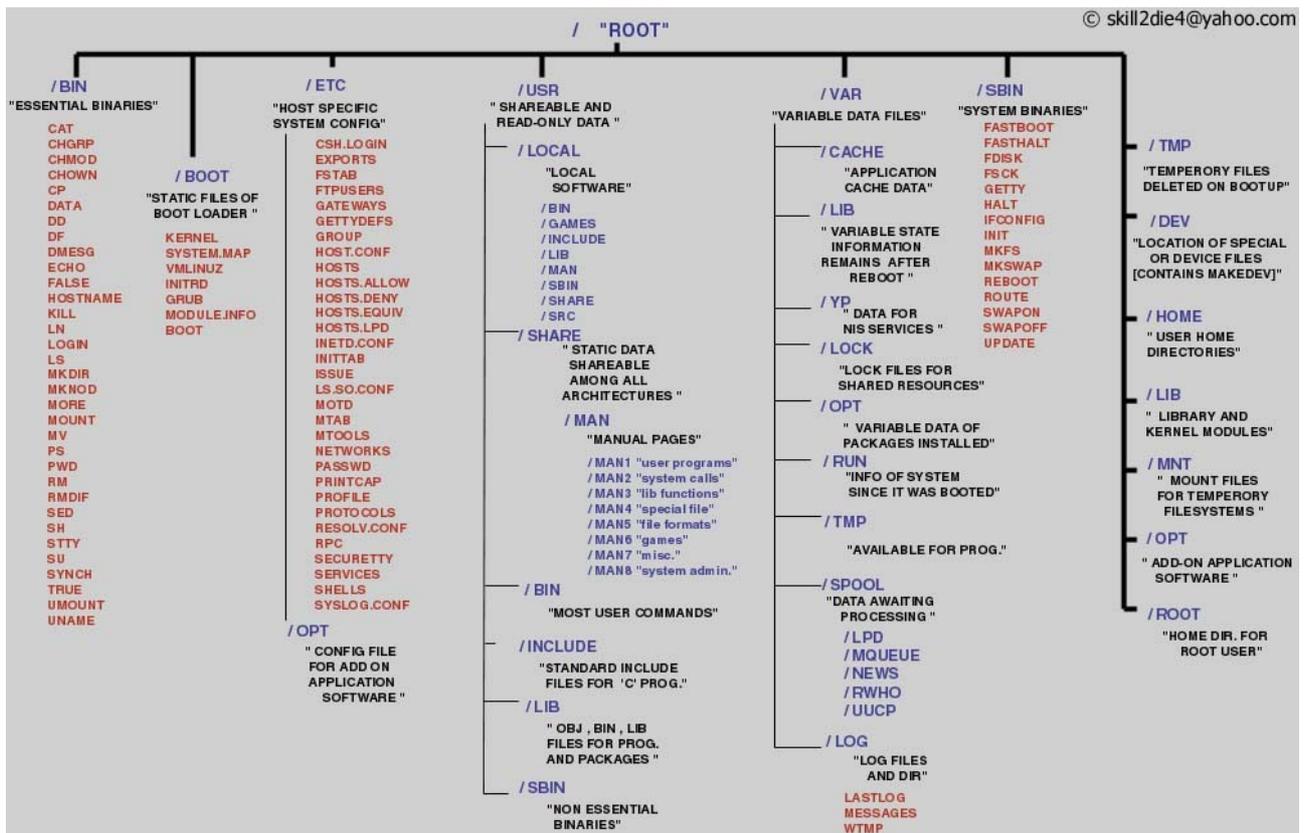
### Funcionamento:

Para realizar tarefas específicas, utilizamos os softwares, como processadores de texto, calculadoras matemáticas, etc.

O Kernel é o núcleo do sistema UNIX. O Kernel se comunica diretamente com o sistema de hardware, e portanto, deve ser adaptado à arquitetura específica de cada plataforma de hardware. É o kernel que isola o usuário das diferenças entre os vários tipos de hardwares.

O shell é um interpretador de comandos. Quando um comando é digitado, o shell traduz o nome daquele comando no conjunto de chamadas de sistema, em linguagem de máquina, que são instruções necessárias para que o kernel realize aquela tarefa.

## Estrutura de diretório



## Comandos para manipulação de arquivos

**ls:** Lista o conteúdo de um diretório. O comando ls pode vir acompanhado de alguns parâmetros.

**-l** – Lista em formato longo. Os arquivos são ordenados pelo nome. Exibe as permissões, o proprietário e a data e hora da última modificação do arquivo.

**-a** - Lista todo conteúdo do diretório, incluindo os arquivos cujos nome comecem com um (.)

Os parâmetros podem ser usados juntamente no mesmo comando ls.

Para mais informações sobre o comando ls vide **man ls**.

**cat:** Concatena e imprime um arquivo. Concatenar significa colocar junto. Este utilitário é usado para exibir o conteúdo de um arquivo e de vários arquivos em sucessão. Exemplo:

**cat teste** – exibe o conteúdo do arquivo chamado teste.

**cat teste teste1 teste2** – exibe o conteúdo do arquivo teste, teste1 e teste2 sucessivamente.

**cat > texto** – direciona a entrada do teclado para o arquivo chamado texto. Se este arquivo não existir ele será criado. **Ctrl + c** finaliza a entrada dos dados.

**cat teste teste1 teste2 > prova** – direciona a saída dos arquivos teste, teste1, teste2 para um único arquivo chamado prova.

**cp (copy):** Copia um arquivo.

**cp agenda agenda.bkp** – cria uma cópia do arquivo agenda chamada agenda.bkp

Alguns parâmetros que podem ser usados com o comando cp:

- R – Copia o conteúdo recursivamente.
- f – Força a cópia de um arquivo.
- v – Verbose, exibe na tela o conteúdo que está sendo copiado.
- p – Preserva as permissões do arquivo que está sendo copiado.

Para maiores informações verifique o manual do comando (**man cp**).

**rm (remove):** Remove arquivos. O comando pode ser utilizado para remover um ou mais arquivos.

**rm arquivo1 arquivo2 arquivo3**

**mv (move):** Move ou renomeia um arquivo. Exemplo.

**mv arqVelho arqNovo** – altera o nome do arquivo de arqVelho para arqNovo.

**mv /home/aluno/notas /home/professor/** – move o arquivo de /home/aluno para /home/professor.

### Alguns comandos UNIX

**df** – disk free, exibe os dispositivos montados no sistema, monto de montagem, tamanho ocupado, tamanho disponível e a porcentagem em uso.

**pwd** – exibe o caminho completo do diretório onde estamos trabalhando.

**free** - exibe informações sobre a memória e área de troca (swap).

**whoami** – mostra quem é o usuário logado no sistema.

**date** – exibe a data do sistema.

**uname -a** – exibe informações sobre o sistema.

**cal** – exibe um calendário.

**id** – exibe informações sobre o usuário logado no sistema.

**whereis** – localiza um arquivo binário (comando).

**wget** – faz o download de arquivos e diretórios a partir da web. Caso o servidor dos arquivos permita, continua um download interrompido por falha na conexão sem necessidade de intervenção do usuário.

**lynx** – browser em modo texto.

### Permissões para acesso a arquivos

Para um sistema multiusuário a restrição ao acesso aos arquivos é fundamental, pois você pode querer que outros usuários do sistema nem mesmo leiam seus arquivos. O sistema de permissão de acesso a arquivos permite que você controle o destino de seus arquivos.

Existem três classes de usuários no sistema:

**Proprietário** – usuário que criou o arquivo.

**Grupo** – o grupo é formado por um ou mais usuários que tem características semelhantes.

**Outros** – refere-se a qualquer outro usuário do sistema.

<i>Modo de acesso</i>	<i>Arquivo</i>	<i>Diretório</i>
Leitura (read)	Ler o conteúdo do arquivo	Listar os arquivos
Escrita (write)	Alterar o conteúdo do arquivo	Criar, remover e renomear arquivos
Execução	Executar o arquivo como comando	Pesquisar no diretório

Cada arquivo/diretório é acompanhado do conjunto de permissões que ele pertence. A saber:

r x x	r w x	r w x	arquivo
proprietário	grupo	outros	nome do arquivo

A presença de uma permissão é indicada pela letra apropriada (r,w,x), a ausência é indicada por um traço (-).

Exmplos:

**r - - r - - r - -**

Este arquivo pode ser lido por todas as três classes. Um arquivo com essas características é conhecido como read-only (somente leitura).

**r w x r - x r - -**

Este arquivo pode ser lido, alterado e executado pelo proprietário do arquivo. Lido e executado pelos usuários que fazem parte do grupo ao qual o arquivo pertence, e lido pelos outros usuários do sistema.

Para cada letra é associado um valor numérico para que possamos atribuir as permissões aos arquivos/diretórios. São elas:

x -> 1

w -> 2

r -> 4

A soma dos valores definem o conjunto de permissões. Exemplo:

7 5 1 = r w x r - x - - x

4 3 0 = r - - - w x - - -

O comando **chmod** permite que se altere os modos de permissões de um ou mais arquivos/diretórios. O formato da linha de comando é: **chmod operação de permissão arquivo**. Exemplo: `chmod 644 teste.doc`

`rw- r- - r- - teste.doc`

### Manipulação de diretórios

**mkdir (make directory)** – cria um diretório.

**cd (change directory)** – troca de diretório.

**rmdir (remove directory)** – remove um diretório, mas este precisa estar necessariamente vazio. Para remover um diretório não vazio e tudo que encontra-se nele utilize o comando: **rm -Rfv diretório**.

## Outros comandos UNIX

**more** – mostra na tela o conteúdo de um arquivo.

**less** – mostra na tela o conteúdo de um arquivo.

**clear** – limpa a tela – Ctrl + l.

**du** – mostra a quantidade de disco utilizada por um determinado diretório – du -sh.

**find** – procura por arquivos. Sintaxe: find localParaPesquisa [argumento] paramentro

**hostname** – exibe o nome do computador.

**passwd** – altera a senha de um determinado usuário.

**fdisk** – manipula partições de disco no linux.

**cdisk** – manipula partições de disco no linux.

**top** – exibe, em tempo real, os processos que estão sendo executados no sistema.

**halt** – pára o sistema.

**reboot** – reinicia o sistema.

**shutdown** – desliga o sistema – shutdown -h now.

## Gerenciamento de usuários e grupos

Cada usuário em um sistema UNIX possui um login e uma senha que são usados para identificar o usuário no sistema. Essa identificação consiste em verificar quem é o usuário que está utilizando o sistema, quais são seus privilégios, que grupo ele faz parte, etc.

Gerenciar usuários é uma tarefa do administrador do sistema, chamado **root**.

É necessário ter muito cuidado ao se identificar no sistema como root, pois o root pode fazer qualquer tipo de modificação no sistema.

Alguns comando utilizados para gerenciar usuários e grupos:

**groupadd nomeDoGrupo** -> Adiciona um novo grupo ao sistema.

**useradd nomeDoUsuario** -> Adiciona um novo usuário ao sistema.

**Groupdel nomeDoGrupo** -> Remove um grupo.

**userdel** -> Remove um usuário.

Exemplo:

**useradd aluno -d /home/convidado -g temporario**

A linha de comando acima insere um novo usuário no sistema chamado aluno, define o diretório HOME em /home/convidado e o usuário fará parte do grupo chamado temporario.

O arquivo de configuração **/etc/shadow** é utilizado para armazenar criptografadas, as senhas dos usuários.

O arquivo de configuração **/etc/passwd** é o arquivo mais cobiçado por **hackers** porque contém os dados pessoais do usuário como o login, senha e uid.

## Processos e Daemons

Um Daemon é um processo que executa em segundo plano e que realiza uma função específica ou uma tarefa relacionada ao sistema. Muitos daemons são ativados na inicialização e continuam a ser executados enquanto o sistema estiver ativo e funcionando.

Para cada comando que o usuário digita um processo é iniciado e um identificador é

atribuído a ele.

O comando **ps** exibe uma lista dos processos do sistema. Se alguma processo apresenta algum mal comportamento, este pode ser finalizado, 'morto', utilizando-se o comando **kill**.

Os daemons encontrados no **/etc/init.d** são facilmente controlados através dos argumentos **start, stop, restart, reload e status**.

Os daemons que são iniciados juntos com o sistema estão no diretório **/etc/rc2.d**. Neste diretório há apenas links cujos nomes que iniciam com a letra **S** são os daemons iniciados no boot do sistema. Os que iniciam com a letra **K** são os que não iniciam junto com o sistema. Os números que acompanham o nome dos arquivos indicam a prioridade do daemon.

Exemplos:

```
/etc/init.d/apache2 stop  
/etc/init.d/mysql start  
cd /etc/rc2.d && mv S99postgresql K99postgresql
```

**Mais comandos unix** - (tar, gzip, unzip, bzip)

**tar** -> Armazena ou extrai arquivos em format tar. Normalmente é utilizado para fazer backup, armazenar arquivos em tapes. O tar é utilizado para reunir vários arquivos num único arquivo de modo a facilitar sua distribuição. Utilização:

```
tar -cpvf nome_do_arquivo.tar arquivo1 arquivo2 arquivo3 arquivoN
```

O comando acima cria um arquivo do tipo tar, contendo os arquivos 1, 2, 3 ... N. O parametro p indica ao comando para preservar as permissões dos arquivos. v (verbose) exibe na tela informações sobre os arquivos que estão sendo processados.

Para descompactar um arquivo .tar basta digitar o comando:

```
tar -xvf nome_do_arquivo.tar
```

Podemos utilizar o comando **tar** em conjunto com os compactadores de arquivo **bzip** e **gzip**. Para isso basta acrescentar o parâmetro z (para gzip) ou j (bzip).

Há também o compactador de arquivos **zip**, compatível com compactadores de arquivos em windows, por exemplo, winzip, winrar, brazilip...

A sintaxe básica desse comando é:

```
zip arquivo.zip arquivo1 arquivo2 arquivo3 arquivoN
```

Para descompactar use o seguinte comando:

```
unzip arquivo.zip
```

## **Serviços de rede**

**scp**: security copy, é utilizado quando queremos copiar arquivos para um computador remoto. A cópia é feita de maneira criptografada. O comando possui a seguinte sintaxe:

```
scp /home/aluno/notas.final aluno@10.10.10.1:/home/aluno/notas.final
```

aluno deverá ser um usuário do computador que possui o IP 10.10.10.1. A senha do usuário será solicitada para que a cópia dos arquivos possa ser concluída.

**ssh:** security shell, é utilizado para que possamos gerenciar um computador remotamente, como se estivéssemos na frente dele. Exemplo:

**ssh aluno@192.168.0.234**

O comando acima permite que administremos remotamente o computador que possui o IP 192.168.0.234. Após autenticado no sistema, se tivermos a senha de root é possível fazer qualquer alteração no sistema.

**NFS** (Network File System): criado em 1985 pela sun, o nfs é um sistema de arquivos em rede que possibilita o compartilhamento de arquivos sobre uma rede.

Recursos necessários:

- Programa **portmap**: mapeador de portas para números de programas **RPC**
- Daemons **nfsd** e **mountd**

Para verificar se os serviços estão rodando execute o comando **rpcinfo -p**

Pacotes necessários:

nfs-user-server e nfs-common

Arquivo de configuração:

/etc/exports

A configuração do arquivo **exports** é feita da seguinte forma:

**/home/aluno/ ip\_do\_computador\_remoto(rw, no\_root\_squash)**

**/home/aluno** -> Este é o diretório que será compartilhado

**ip\_do\_computador\_remoto** -> Este é o ip do computador que terá acesso ao diretório compartilhado. Pode-se usar \* para liberar para qualquer computador.

**rw** -> Dá permissão de leitura e gravação para os ips especificados.

**no\_root\_squash** -> Dá permissão de acesso ao root remoto.

Para montar um sistema de arquivos nfs utilize o seguinte comando.

**mount -t nfs ip\_do\_computador:/caminho\_do\_compartilhamento /ponto\_montagem**

## **samba**

samba é um pacote de Software Livre, que permite que computadores Windows, ligados em rede, utilizem o Linux, ou Unix, como servidor de arquivos e impressão, como se eles estivessem acessando um servidor Windows.

O samba permite que clientes Windows compartilhem arquivos através do **NetBios**.

Para instalar o samba é necessário instalar os pacotes: **samba, smbfs e smbclient**.

O arquivo de configuração do samba é **smb.conf**, geralmente localizado no diretório **/etc**.

É comum encontrar ferramentas web para fazer a configuração do samba. O SWAT (samba

web administration tool) e o WEBMIN são dois exemplos. Para instalar o swat basta dar o comando **apt-get install swat**.

Para acessar um compartilhamento feito em um Windows através do protocolo NetBios utilize a seguinte linha de comando:

```
mount -t smbfs -o username=nomeDoUsuario password = //nomeDoCompartilhamento
pontoDeMontagem
```

## Montar e desmontar volumes

Um dispositivo seja ele um hd, uma unidade de cdrom, disquete, etc precisa estar preparado para poder ser lido ou para poder receber informações. Essa preparação é conhecida como “montar uma unidade”. Para montar uma unidade utilizamos o comando **mount** com a seguinte sintaxe:

```
mount -t [tipo do sistema de arquivos] [dispositivo a ser montado] ponto de montagem
```

### Tipo de sistema de arquivos

O sistema de arquivo varia de acordo com o hardware e com o sistema operacional que está operando sobre um determinado hardware. O sistema de arquivo diz como as informações estão gravadas no dispositivo. Tipos de sistema arquivos:

**fat** – utilizado no windows 95

**fat32** – utilizado no windows 95/98

**ntfs** – utilizado no windows NT, 2000, 2003, XP e VISTA

**ufs** – utilizado pelo FreeBSD

**xfs, ext2, ext3, raiserfs** – são alguns dos sistemas de arquivos utilizados pelo linux.

O sistema de arquivo contido nas unidades de cdrom é iso9660.

Os dispositivos a serem montados geralmente estão no diretório `/dev/` que é onde ficam os dispositivos do sistema. A saber:

`/dev/fd0` – Dispositivo de disquete 1.44.

`/dev/cdrom0` – Dispositivo do cdrom.

`/dev/sda` – Dispositivos USB, HD SATA, SCSI e outros.

`/dev/hda` – HD primário.

`/dev/hdb` – HD secundário.

O ponto de montagem define um diretório onde o dispositivo será montado. O dispositivo será representado pelo diretório definido como ponto de montagem. Normalmente montamos os dispositivos no diretório `/mnt/...`

A linha de comando a seguir monta um disquete 1.44

```
mount -t vfat /dev/fd0 /mnt/disquete
```

disquete é um diretório que foi criado dentro do diretório mnt

Para desmontar um volume utiliza-se o comando **umount**. Quando desmontamos um

volume todas as alterações serão gravadas. Exemplo: **umount /mnt/cdrom0**, onde /mnt/cdrom0 é o ponto de montagem desse volume.

O arquivo de configuração /etc/fstab contém informações sobre os pontos de montagem.

### Configuração da rede

Para se ter acesso aos serviços de rede, é necessário configurar a rede no seu sistema. A configuração da rede envolve: endereço ip, máscara de rede, gateway de rede e servidor dns.

A configuração da rede pode ser feita editando alguns arquivos.

**/etc/hostname** – neste arquivo define-se o nome do computador.

**/etc/resolv.conf** – neste arquivo coloca-se o endereço dos servidores dns. A configuração é feita da seguinte forma: **nameserver ipDoServidor**

**/etc/network/interfaces** – neste arquivo configuramos o ip do computador, máscara de rede e gateway que serão utilizados.

Outra maneira de configurarmos a rede é através do comando **ifconfig**.

**ifconfig** – exibe as interfaces de rede detectadas pelo sistema.

A atribuição do endereço ip utilizando o comando ifconfig é feita da seguinte forma:

**ifconfig eth0 10.0.0.1 netmask 255.0.0.0**, onde eth0 é a interface de rede que está sendo configurada. 10.0.0.1 é o endereço ip atribuído à interface eth0 e 255.0.0.0 é a máscara de rede.

A configuração do gateway (roteador) é feita da seguinte forma:

**route add default enderecoDoGateway**

### Manipulação de pacotes

O **apt** é o sistema de gerenciamento de pacotes de programas que possui resolução automática de dependência entre pacotes. Método fácil para instalação de pacotes, facilidade de operação, permite atualizar facilmente sua distribuição.

Além do **apt** existe também o **dpkg**.

### Configuração do apt

Configurar o apt consiste em definir o repositório fonte dos pacotes e a versão do S.O. Essas fontes podem ser **mirrors** ou o **CD-ROM**. O arquivo que será configurado é o /etc/apt/sources.list. Exemplo:

**deb cdrom:/cdrom**

**deb http://www.us.debian.org/debian woody main contrib non-free**

**deb ftp://ftp.debian.org/debian**

Feita a configuração do arquivo o próximo passo é fazer as instalações dos pacotes usando o **apt-get**.

## Instalação dos pacotes via apt-get

Inicialmente é necessário fazer uma atualização das informações dos pacotes disponíveis, para isso use o comando: **apt-get update**.

Depois atualize os pacotes instalados, usando o seguinte comando: **apt-get upgrade**.

Todos os pacotes que necessitam de atualização serão atualizados. Este processo pode levar tempo. Depende das conexões de rede.

Após atualizar o sistema, podemos fazer a instalação de outros pacotes com segurança. O comando utilizado para instalar novos pacotes no sistema é: **apt-get install nomeDoPacote**. Por exemplo para instalar um utilitário que permite a gravação de cd/dvd (k3b) digite:

```
apt-get install k3b
```

Quando não sabemos o nome correto do pacote que queremos instalar, podemos procurar os pacotes utilizando o comando: **apt-cache search nomeDoPacote**. É aconselhável utilizar esse comando em conjunto com os comandos **grep**, **more** ou **less** pois a quantidade de resultados é muito grande. Exemplo.

```
apt-cache search apache | grep apache2  
apt-cache search dvd | more  
apt-cache search player | less
```

## Remoção dos pacotes

A remoção dos pacotes é simples de se fazer. Basta digitar o seguinte comando: **apt-get remove nomeDoPacote**.

## Utilização do comando dpkg

Outro comando utilizado para manipular pacotes no Debian é o **dpkg**.

Para podermos instalar um pacote utilizando o **dpkg** é necessário fazermos o download do pacote. Esse pacote deve estar preparado para ser instalado em um sistema debian. Estar preparado, significa estar compilado e empacotado para o sistema debian. Esses pacotes tem **.deb** como extensão. Por exemplo, para podermos instalar um pacote chamado mplayer.deb no debian, fazemos da seguinte forma: **dpkg -i mplayer.deb**.

Para removermos um pacote utilizando o comando **dpkg** basta digitar:

```
dpkg --remove pacote  
dpkg --purge pacote
```

A primeira instrução remove o pacote em questão e a segunda remove os arquivos de utilizados para configuração do pacote.

Uma lista dos pacotes instalados no sistema pode ser obtida através do comando: **dpkg -l**.

Para verificarmos os pacotes instalados que se referem ao servidor web apache, digitamos o seguinte comando: **dpkg -l | grep apache**.

## Compilado e instalando pacotes

Os comando **apt** e **dpkg** utilizam pacotes prontos para serem instalados no sistema. Isso facilita muito a instalação de pacotes, porém perdemos um pouco a capacidade de personalizar a instalação.

Uma das vantagens de se usar um sistema linux é a grande capacidade de personalização do sistema e pacotes.

Compilar um pacote significa pegar um programa que alguém escreveu, geralmente em C/C++, e transformá-lo em uma linguagem que o computador possa entender. Essa conversão é feita utilizando um programa chamado compilador. Para cada linguagem de programação tem-se vários compiladores específicos. No caso do linux é o C/C++.

A compilação e instalação de pacotes no linux consiste dos seguintes passos:

- 1) Fazer o download do pacote que se deseja instalar. Esse pacote é um conjunto de arquivos e normalmente vem compactado.
- 2) Fazer a descompactação do pacote.
- 3) Entrar no diretório descompactado e executar o comando **./configure**. O comando **configure** irá verificar se seu sistema possui todos os recursos necessários para instalar o pacote. É também no comando **configure** que determinamos onde iremos instalar o pacote e quais recursos o pacote irá prover. Terminado a configuração um arquivo chamado **Makefile** é criado. Esse arquivo indicará ao compilador as regras a serem seguidas para compilar o pacote. O próximo passo é compilar o pacote.
- 4) Para compilarmos o pacote basta digitar o comando **make**.
- 5) Após compilado utilizamos o comando **make install** para instalarmos os arquivos nos seus respectivos lugares.

## Referencias:

Devido a grande diversidade de tarefas e maneiras de executar essas tarefas em um ambiente linux, frequentemente precisamos pesquisar sobre tais assuntos na Web. Um bom link para referencia é: <http://focalinux.cipsga.org.br/>

A maioria dos comando linux possui um manual que diz tudo a respeito do comando. Aprender a analisar o manual do comando auxilia bastante na execução de tais comandos.

## Outras Referencias:

<http://www.linorg.cirp.usp.br/>

<http://www.debian.com/>

<http://pt.wikipedia.org/wiki/Linux>