

Estrutura de diretórios no Linux

Gustavo Sverzut Barbieri (gsbarbieri@yahoo.com.br)

10 de novembro de 2002

1 Introdução

Algumas informações úteis:

- O separador de diretórios é o caractere `/`. Por exemplo, usa-se `cd /` e não `cd \`.
- O Unix é “case sensitive”, isto é, diferencia maiúscula de minúscula.
- O Unix não utiliza extensões de arquivos. Para arquivos executáveis, o que importa é a permissão do arquivo, para os demais, o conteúdo somente. Geralmente quem requer extensões são alguns programas como o “gzip”, mas é específico de programas, não do sistema. Algumas extensões são usadas para o humano diferenciar com mais facilidade um arquivo de outra. Alguns window managers se utilizam delas para simular funções de atribuições de “tipos de arquivos” a programas, como no Windows
- O símbolo “`~`” indica casa do usuário, por exemplo: `cd ~/doc` vai para o diretório “doc” dentro do diretório do usuário.
- Arquivos começados com “`.`” são considerados escondidos. Eles não são listados com o comando “ls” ao menos que a opção “`-a`” for utilizada.
- Sempre que quiser executar um programa que está no diretório local e que não está em diretórios do PATH (`/bin`, `/usr/bin`, ...) usa-se o ponto e barra (`./`). Por exemplo, usa-se `./meuprograma` para executar o arquivo “meuprograma” que está no diretório atual.
- No Linux usa-se o espaço entre o nome do programa e seus argumentos. Por exemplo se você digitar `cd/` ou `cd..` dará erro.

A estrutura de diretrios é a seguinte:

Diretório	Descrição
/bin	arquivos executáveis essenciais para o sistema.
/boot	arquivos necessários ao boot (inicialização) do sistema.
/dev	componentes físicos do computador ¹ .
/etc	arquivos de configuração.
/home	diretórios de usuários.
/initrd	arquivos necessários ao boot (em algumas distribuições).
/lib	bibliotecas essenciais ao sistema.
/mnt	pontos de montagem de outros sistemas de arquivos ² .
/opt	alguns programas(geralmente comerciais) utilizam este diretório.
/proc	informações sobre seu sistema ¹ .
/root	diretório do super-usuário(administrador).
/sbin	arquivos executáveis essenciais, somente administrativo (sbin = superuser binaries).
/tmp	arquivos temporários.
/usr	replicação do '/' para utilização comum (Ex: instalação de programas em geral).
/var	diretórios subsequentes são usados pelos servidores.
/usr/X11R6	diretório contém toda a estrutura do serviço de janelas (modo gráfico) X11.
/usr/doc	documentação de programas. (vai ser mudado para /usr/share/doc)
/usr/man	manuals de programas. (vai ser mudado para /usr/share/man)
/usr/share	arquivos auxiliares dos programas.
/usr/src	fontes (sources) de aplicativos.

2 Alguns diretórios especiais

2.1 O /dev

O diretório `/dev` contém o mapa de todos os componentes do seu computador. Ele não foi feito para ser usado por usuários comuns, mas é sempre bom entender alguma coisa sobre ele.

Dentre os componentes mapeados alguns nós conhecemos, veja a tabela:

Componente	Descrição
/dev/ttyS0	1ª porta serial. (COM1 no Windows)
/dev/ttyS1	2ª porta serial. (COM2 no Windows)
/dev/ttyS2	3ª porta serial. (COM3 no Windows)
/dev/ttyS3	4ª porta serial. (COM4 no Windows)
/dev/hda	Disco conectado ao IDE 0 como master.
/dev/hdb	Disco conectado ao IDE 0 como slave.
/dev/hdc	Disco conectado ao IDE 1 como master.
/dev/hdd	Disco conectado ao IDE 1 como slave.
/dev/hda1	1ª partição do disco conectado ao IDE 0 como master.
/dev/fd0	1ª disquete (Drive A:)
/dev/fd1	2ª disquete (Drive B:)
/dev/dsp ³	Acesso à sua placa de som pelo canal digital.
/dev/audio ³	Acesso à sua placa de som pelo canal analógico.
/dev/mixer ³	Acesso ao mixer da sua placa de som.

Os componentes são ou de blocos ou de caracter. A diferença é que os componentes de blocos podem ser lidos caracter por caracter e os de blocos, somente em blocos.

O importante é lembrar que este diretório `/dev` contém o mapa dos componentes de seu computador.

¹O Unix mapeia tudo do computador para a estrutura de diretórios/arquivos. Vide seções 2.1 e 2.2

²Vide seção 4

³Os componetes da placa de som podem estar localizados em locais diferentes, como em `/dev/snd`, caso usarmos o driver de som do projeto ALSA. Se sua placa de som possuir mais de 1 componente listado, eles estarão nomeados com um número apartir de zero após o nome, ex: o 2º mixer se encontra em `/dev/mixer1`

2.2 O /proc

O diretório `/proc` fornece informações sobre o seu computador. Dentro dele você encontra informações sobre o processador (`/proc/cpuinfo`), memória (`/proc/meminfo`) e outros.

Segue uma tabela com a descrição de alguns arquivos do `/proc`:

Arquivo	Descrição
<code>/proc/cmdline</code>	comandos passados ao kernel do linux na inicialização.
<code>/proc/cpuinfo</code>	informações sobre o seu CPU.
<code>/proc/devices</code>	lista os componentes de sistema funcionando.
<code>/proc/filesystems</code>	informações sobre os sistemas de arquivos.
<code>/proc/ide/</code>	informações sobre os discos ide.
<code>/proc/interrupts</code>	informações sobre interrupções geradas.
<code>/proc/iomem</code>	informações sobre as áreas de memória de entrada/saída.
<code>/proc/ioports</code>	informações sobre as portas de entrada/saída.
<code>/proc/kcore</code>	o kernel core, nunca dê um <code>cat /proc/kcore</code> !!!.
<code>/proc/meminfo</code>	informações sobre a memória.
<code>/proc/modules</code>	informações sobre os módulos lidos.
<code>/proc/partitions</code>	informações sobre partições.
<code>/proc/mounts</code>	informações sobre sistemas montados.
<code>/proc/pci</code>	informações sobre componentes PCI.
<code>/proc/self</code>	informações sobre o próprio programa.
<code>/proc/sys/</code>	arquivos que controlam ⁵ o sistema.
<code>/proc/uptime</code>	tempo que o sistema está ativo.
<code>/proc/version</code>	versão do sistema.

3 Criando e apagando diretórios e arquivos

Para criar um diretório, usamos o comando “`mkdir`” seguido do diretório que queremos criar. O nome do diretório a ser criado pode ter qualquer caracter exeto o separador de diretório (`/`), mas caracteres diferentes de letras, números e underscore, precisam ter um “`\`” na frente ou então o nome deve estar entre aspas. Exemplo:

Comando	Descrição
<code>mkdir teste</code>	cria diretório “teste” no diretório local.
<code>mkdir /teste</code>	cria diretório “teste” no diretório raiz.
<code>mkdir ./teste</code>	cria diretório “teste” no diretório local.
<code>mkdir ../teste</code>	cria diretório “teste” no diretório anterior ao local.
<code>mkdir teste1 teste2</code>	cria os diretórios “teste1” e “teste2” no diretório local.
<code>mkdir teste1\ teste2</code>	cria o diretório “teste1 teste2” no diretório local.
<code>mkdir "teste1 teste2"</code>	cria o diretório “teste1 teste2” no diretório local.

Para apagar um arquivo ou diretório usamos o comando “`rm`” seguido do arquivo que queremos apagar. O nome do arquivo segue as mesmas características citadas acima para criar um diretório. O comando “`rm`” tem algumas opções úteis que são comumente utilizadas, elas são:

- **-f** força apagar o arquivo e não lhe pergunta se pela confirmação. Isso é utilizado no caso de você ser o dono dele mais colocou permissões de somente leitura.
- **-r** vai recursivamente dentro dos diretórios conseguintes (como o “`deltree`” do ms-dos).

Exemplo:

⁵Alguns arquivos no diretório `/proc/sys` servem para receber comandos. Por exemplo, se você fizer “`echo 1 > /proc/sys/net/ipv4/ip_forward`” você estará abilitando o repassamento de pacotes de rede.

Comando	Descrição
<code>rm teste</code>	apaga arquivo “teste”.
<code>rm -f teste</code>	apaga arquivo “teste” sem lhe pedir confirmação, mesmo que o arquivo teste seja somente de leitura.
<code>rm -f teste</code>	apaga arquivo “teste” sem lhe pedir confirmação, mesmo que o arquivo teste seja somente de leitura.
<code>rm -r teste_dir</code>	apaga todos os arquivos do diretório “teste_dir” e o próprio diretório.
<code>rm -fr /</code>	faça isso como root e você fodeu teu sistema :)

4 Montando outros sistemas de arquivos

No Unix não existe c:, d:, a:... os sistemas de arquivos são montados (mapeados) para a estrutura de diretórios do sistema. Por exemplo, se você tem um cdrom, ele geralmente é montado em `/mnt/cdrom` ou `/cdrom`. Um diskete geralmente é montado em `/mnt/floppy`. Mas não é obrigatória a montagem em `/mnt`, por exemplo, muitas pessoas que tem 2 HDs (ou instalaram o sistema em várias partições) gostam de deixar o sistema em uma partição/disco, a parte de boot em outra e os diretórios de usuários em outras, mas a estrutura de diretórios fica a mesma. Ex:

Ponto de Montagem	Origem
<code>/</code>	<code>/dev/hda1</code> (1a partição do hd em ide0, master)
<code>/home</code>	<code>/dev/hda2</code> (2a partição do hd em ide0, master)
<code>/boot</code>	<code>/dev/hda3</code> (3a partição do hd em ide0, master)
<code>/usr</code>	<code>/dev/hdb1</code> (1a partição do hd em ide0, slave)
<code>/mnt/cdrom</code>	<code>/dev/hdc</code> (disco em ide1, master)
<code>/mnt/dvd</code>	<code>/dev/hdd</code> (disco em ide1, slave)
<code>/mnt/windows</code>	<code>/dev/hda4</code> (4a partição do hd em ide0, master)
<code>/mnt/redewin</code>	algum computador de rede windows
<code>/mnt/redelin</code>	algum computador de rede linux

Valhe notar que geralmente o Unix não monta/desmonta automaticamente os pontos de montagem, por exemplo, se você inserir um CD-ROM e não montá-lo, em `/mnt/cdrom` não terá nada. Por outro lado, se você não desmontar o CD-ROM e tirá-lo do drive, você continuará a ver os arquivos em `/mnt/cdrom` e quando tentar copiar, ele vai esperar até que o drive esteja pronto. Por isso sempre se lembre de desmontar os drives de CD e disquete.

OBS: Em algumas distribuições como Conectiva e Mandrake, vem com o supermount habilitado. O supermount monta para você as o CD-ROM e disquete automaticamente.

4.1 CD-Rom

Para montar o CD-ROM, escolheremos um ponto de montagem, por exemplo `/mnt/cdrom`. Sabendo que seu CD-ROM é o `/dev/hdc` (Vide seção 2.1). Usamos:

```
mount -t auto /dev/hdc /mnt/cdrom
```

Se isso não funcionar, tente com outro dispositivo (Ex.: `/dev/hdd`). Se mesmo assim não funcionar, tente ler o manual do “mount” e procurar ajuda na internet. Se isso funcionar, edite o arquivo `/etc/fstab` e adicione a linha:

```
/dev/hdc /mnt/cdrom auto user,noauto 0 0
```

Agora, você pode digitar somente:

```
mount /mnt/cdrom
```

4.2 Disquete

Para montar o disquete, escolheremos um ponto de montagem, por exemplo `/mnt/floppy`. Sabendo que seu disquete é o `/dev/fd0` (Vide seção 2.1). Usamos:

```
mount -t auto /dev/fd0 /mnt/floppy
```

Se isso não funcionar, tente com outro dispositivo (Ex.: `/dev/fd1`). Se mesmo assim não funcionar, tente ler o manual do “mount” e procurar ajuda na internet. Se isso funcionar, edite o arquivo `/etc/fstab` e adicione

a linha:
`/dev/fd0 /mnt/floppy auto user,noauto 0 0`
Agora, você pode digitar somente:
`mount /mnt/floppy`

4.3 Desmontando os sistemas montados

Para desmontar um sistema montado, use o comando “`umount`” seguido do lugar onde o sistema está montado.

5 Tipos de arquivos peculiares ao Unix

O Unix tem tipos de arquivos que não existem no MS-DOS e que são muito úteis, são eles:

- **FIFO:** é uma fila (Firs In, Firs Out) na qual um programa lê enquanto o outro escreve. É utilizada para simular um controle remoto por alguns programas como o Licq. Por exemplo, se você estiver com o seu Licq aberto, experimente “`echo “message NICK_DO_SEU_AMIGO ‘mensagem’ ” > ~/.licq/licq_fifo`”. Um FIFO é criado com o comando “`mkfifo`” seguido do nome do arquivo.
- **Link simbólico:** é um arquivo que na verdade se referencia a outro arquivo, um “link” como em HTML. Este é muito utilizado no meio Unix porque ele pode referenciar um outro arquivo por meio de um caminho absoluto (Ex.: `/home/gustavo/teste`) ou por meio relativo (Ex.: `../teste`). Um link simbólico é criado com o comando “`ln`” com a opção “`-s`” seguida do caminho ao arquivo referenciado e o nome do link. Ex.: `ln -s /bin/ls ~/meu_ls`
- **Link:** é uma peculiaridade do Unix bem útil, pois permite dar vários nomes a um mesmo arquivo. A diferença entre o link normal e o simbólico é que o simbólico apenas aponta o caminho para o outro arquivo, como em HTML, se você apagar ou mover o arquivo a qual o link simbólico se referencia, o link fica quebrado. Já com o link normal não, pois eles apontam para o lugar no disco (fisicamente) e quando você move o arquivo original, você não muda a posição no disco, portanto ele nunca se quebra. Para se apagar um arquivo com vários links (normais, não simbólicos), é necessário apagar todos os links, senão o arquivo permanece em disco. O problema deste tipo de link é que se você for, por exemplo, compactar um diretório que tem um arquivo que possui 2 nomes (links) ele vai compactar 2 arquivos em separado, como se fossem 2 arquivos diferentes. Por isso o simbólico, apesar de dar mais problemas, é mais utilizado

6 Permissões de Arquivos

No Unix pode-se dar permissões para um arquivo. As permissões são divididas em 3 blocos: permissões do usuário, permissões do grupo e permissões para outros.

Primeiramente temos que falar, então, como se escolhe o dono e grupo do arquivo. Para escolher o dono do arquivo, você precisa ser o usuário “`root`” e usar o comando “`chown`” seguido do nome do usuário a ser dono do arquivo e em seguida. Para escolher o grupo do arquivo, precisa-se estar dentro do grupo a escolher (cria-se um grupo com o usuário `root`, editando o arquivo `/etc/groups`) e usa-se o comando “`chown`” seguido do nome do grupo e o nome do arquivo.

Cada bloco de permissões especifica as permissões de leitura(indicada pela letra “`r`”, do inglês *Read*), escrita(“`w`”, do inglês *Write*) e execução(“`x`”, do inglês *eXecute*) do arquivo. Para se mudar as permissões de um arquivo usa-se o comando “`chmod`” seguido das permissões e do arquivo. Existem 2 jeitos de se especificar as permissões de um arquivo, um literal e outro número:

- **Modo literal:** é uma sequência separada por vírgula de comandos literais. Um comando literal é dividido em 3 blocos: “para quem”, “fazer o que” e “que permissões”. O bloco “para quem” pode ser as letras “`u`” que indica o usuário dono do arquivo, “`g`” que indica o grupo e “`a`” que indica

todos. O bloco “fazer o que” pode ser o sinal de “+” para se adicionar as permissões ou “-” para retirar. O bloco “que permissões” pode ser “r”, “w” e “x” como indicado anteriormente. Ex.: `chmod u+rx,g+rx,g-w,a-rwx arquivo` vai dar ao dono permissões de ler, escrever e executar, ao grupo somente de ler e executar e aos outros nenhuma.

- **Modo numérico:** é um número de 3 algarismos, sendo que o algarismo da centena especifica permissões do usuário, o algarismo da dezena especifica as permissões do grupo e o da unidade as permissões dos outros. Cada algarismo é composto pela soma de 3 números: 1 para executar, 2 para escrever e 4 para ler. Por exemplo, o exemplo anterior do modo literal pode ser reescrito como `chmod 750 arquivo`, pois o 7 significa ler, escrever e executar, o 5 indicar ler e executar e o 0 indica nenhuma permissão.