

## Aula 3

### 1 - Introdução

Para que um usuário possa operar um sistema Linux, lhe são concedidos terminais, pois como sabemos, esses sistemas possuem interpretadores de comando para que os usuários possam interagir indiretamente com a máquina, nos moldes de sistemas operacionais clássicos. Como era de se esperar, esses interpretadores só podem ser executados por aqueles usuários que obtiverem o credenciamento e autenticação do sistema para tal.

O sistema permite o uso de interfaces (para o interpretador de comando) através desses terminais que podem ser remotos ou locais. São os antigos “teletypes” ou *tty* que representam os terminais disponíveis locais, podendo ser alternados entre as teclas CTRL+ALT+Fx (onde x é número do terminal, em geral de 1 até 7 ou até 10).

O importante é saber que existem dois modos básicos de operação e interação com o sistema. Estes podem ser grosseiramente definidos como modo de operação texto e modo de operação gráfico. Os aspectos mais importantes quando se tem o primeiro contato com um sistema Linux podem ser descritos por login e modo de trabalho:

- **O processo de login:** O programa *getty* é executado em todo processo de inicialização do sistema e este chama o programa *login* para todo usuário que se deseja autenticar. Além de checar a identidade, o processo de *login*, pode antes de mais nada, verificar a existência de alguns arquivos que determinam o comportamento ou acesso ao sistema.

Os arquivos mais comuns são: */etc/nologin* (proíbe o login de qualquer usuário) ou o */etc/securetty* (não permite o login de usuário *root* em um determinado terminal)

- **O ambiente misto (gráfico/texto):** Pode ser operado através de interface de janelas com a ajuda de um servidor de ambiente gráfico ou então ser inteiramente executado em modo de linha de comando. O servidor de janelas mais popular é o X11 (*X Window System*), que atualmente é mantido pelo *X.Org Foundation*. O arquivo de configuração mais importante para esse software, em seu modelo clássico, pode ser encontrado em */etc/X11/xorg.conf*.

### 2 - Os sistemas de arquivos e sua árvore no Linux

Assim como em outros sistemas operacionais, o Linux tem suporte a uma vasta gama de sistemas de arquivos. Além de trabalhar com os sistemas de arquivos de outras plataformas como por exemplo FAT, FAT 32, NTFS, o Linux opera também com os sistemas chamados de nativos como ReiserFS, EXT2, EXT4, XFS entre outros.

Os sistemas de arquivos nativos do Linux possuem algumas características comuns e tem como ponto forte a segurança e acesso aos dados. Em verdade, os sistemas de arquivos mais recentes, como por exemplo o NTFS também possuem essas características de segurança e controle. Como o Linux descende de sistemas POSIX, desde sua origem, todos os sistemas nativos construídos utilizaram como premissas a características implícitas desses sistemas como concorrência, sincronismo e segurança.

Em relação à formatação e nomes, os arquivos podem ser escritos com até 255 caracteres e não existe a necessidade de se manter uma extensão compulsória para arquivos. Dessa forma, os nomes podem ser livremente escritos desde que não utilizem caracteres especiais (como !@#\$\$%^ entre outros). Existe diferenciação entre letras maiúsculas e minúsculas, por isso deve-se atentar ao criar nomes de arquivos ou diretórios para não correr o risco de não encontrá-los posteriormente.

Em relação aos tipos de arquivos no Linux podemos encontrar os seguintes tipos:

- **Arquivos de dados** : podem ser arquivos de texto ASCII ou arquivos binários. São arquivos de texto, arquivos com linhas de comando, arquivos executáveis e outros.

- **Arquivos diretórios** : arquivos especiais que organizam coleções de arquivos comuns. No Linux existem diretórios que tem fins e usos específicos. Muitas vezes esses padrões podem variar um pouco de uma distribuição para outra, mas de qualquer forma sempre encontraremos diretórios bem definidos e conhecidos nos sistemas POSIX.

- **Links** : são na verdade atalhos. Assim como em outros sistemas, o Linux possui atalhos de arquivos para outros arquivos dentro do mesmo sistema de arquivos.

- **Arquivos de dispositivos** : Todo hardware ou dispositivo reconhecido pelo Linux é mantido e operado na forma de arquivo. Portanto se uma placa de som estiver instalada em um computador com Linux, é possível passar comandos para esta placa através de um arquivo que a representa.

- **Sockets** : são arquivos especiais utilizados para a comunicação entre processos. Em geral, são utilizados para permitirem a comunicação de processos em rede.

- **Pipes** : são arquivos especiais também utilizados para a intercomunicação entre processos. Em geral, essa comunicação se restringe ao sistema local.

Dentre os sistemas de arquivos mais conhecidos estão o minix, ext4, ext3, JFS, XFS, ReiserFS e UFS. Entre esses sistemas de arquivos, o ReiserFS e o EXT4 são as mais encontradas nas distribuições disponíveis no mercado até o momento (fevereiro de 2010).

### 3 - Estrutura de diretórios do Linux

Como fora anteriormente citado, em todos os sistemas Linux encontramos diretórios com nomes padronizados e que possuem finalidades e utilizações bem definidas. Tudo isso ajuda a manter a padronização bem como a orientação de qualquer usuário que tenha familiaridade com sistema.

O grande diferencial entre o Linux e os sistemas mais utilizados na prática é de que a árvore de sistema de arquivos sempre vai descender de um ponto central. Essa é a raiz do sistema e vai abrigar todo e qualquer arquivo que tenha de ser manipulado. Diferentemente do MS-DOS, por exemplo, em Linux, é possível manter a unidade de CDROM abaixo do sistema de arquivo raiz, ao passo que no DOS, era necessário haver uma raiz ou volume separado para essa unidade.

A figura 1 abaixo exhibe de forma exageradamente resumida a estrutura de um sistema de arquivos tradicional.

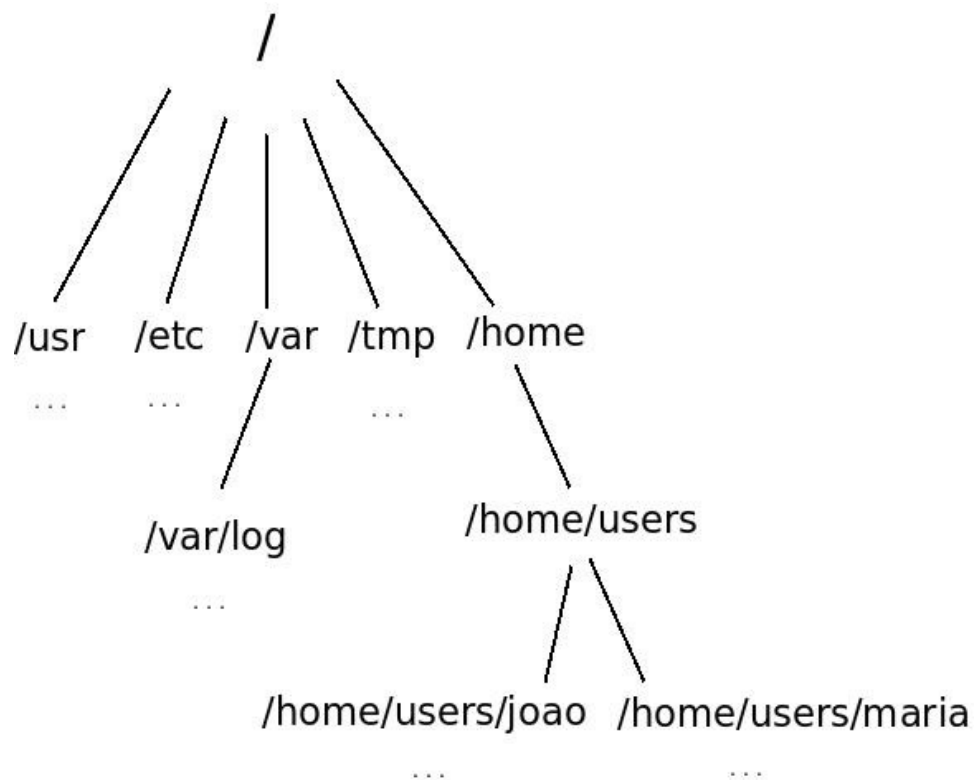


Figura 1 : Exemplo de árvore de sistema de arquivos de um GNU/Linux

Em verdade, há uma padronização formalizada pelo projeto *File System Hierarchy Standard* e que atualmente se encontra na versão 2.3.

Para o bom uso e operação do sistema é necessário conhecer e entender a finalidade de cada um destes diretórios. Nesse sentido, segue uma lista sucinta dos diretórios mais conhecidos e utilizados nos sistemas Linux:

/ - Diretório raiz do sistema. É o ponto inicial do sistema de diretórios. Todos outros diretórios estão dentro deste.

/boot – Diretório onde se colocam as imagens do sistema (kernels).

/proc – Diretório onde se encontram as informações relativas aos processos, kernel e dispositivos para controle e administração do próprio sistema.

/dev – Diretório onde se encontram os dispositivos de hardware.

/tmp – Diretório de arquivos temporários. Qualquer aplicação e usuário têm poder de escrita no mesmo.

/etc – Arquivos de configurações e scripts de configuração ou ativação de serviços.

/lib – Bibliotecas compartilhadas e módulos de kernel.

/mnt – Diretório específico para montagem de sistemas de arquivos externos (floppy, cdrom, pendrive, etc).

/opt – Softwares opcionais, não instaladas na formatação.

/root – Diretório pessoal do usuário administrador. Não confundir com o root (raiz) do sistema.

/home – Diretórios pessoais dos usuários.

/bin – Arquivos de programas e executáveis.

/sbin – Arquivos de programas e executáveis com privilégios apenas ao administrador

/usr – Arquivos Locais. São softwares, documentos, bibliotecas e todos os pacotes instalados

opcionalmente.

/var – Diretório de dados de entrada e saída. São as variáveis lidas ou escritas pelo sistema. Em geral possui filas de impressão, arquivos de log, etc.

Outra nomenclatura importante é em relação ao caminho de arquivos localizados dentro de um sistema de arquivos. De modo bastante simples, podemos classificar o caminho de um arquivo como absoluto ou relativo, sendo eles:

**a) caminho absoluto:** é o caminho completo de um arquivo (diretório,arquivo regular,etc) desde o ponto de origem do sistema, isto é, o diretório raiz “/”. Portanto, exemplos de caminhos absolutos seriam:

- /var/log/messages (caminho absoluto do arquivo messages)
- /etc/init.d/rc.local (caminho absoluto do arquivo rc.local)

**b) caminho relativo:** é o caminho que deve ser seguido na árvore de sistemas de arquivos em relação a um ponto referencial inicial. Por exemplo:

- log/messages (caminho relativo de messages em relação ao diretório /var/)
- rc.local (caminho relativo de rc.local em relação ao diretório /etc/init.d/)

De fato, essa nomenclatura é idêntica a qualquer outro sistema que a maioria das pessoas está acostumada a utilizar, portanto, não há grandes dificuldades para novatos. O importante é conseguir, a partir desta base, iniciar uma consulta mais aprofundada na literatura recomendada. O resultado será uma visão detalhada e específica sobre o sistema de arquivos e suas padronizações.

## 4 - Questões

- 1) Dentro de seu sistema, indique três arquivos regulares, e através destes mostre quais seriam exemplos de caminhos relativos e caminhos absolutos.
- 2) Descreva quais os tipos de arquivos do Linux. Através de seu sistema, escreva exemplos de ao menos dois desses tipos.
- 3) O comandos *ls,cat,more,whereis* e *who* são executáveis que vêm instalados em todos Linux. *Qual é o diretório que por padrão você deduziria poder encontrar esses executáveis?*
- 4) Qual a utilidade de cada um destes diretórios? Indique um arquivo existente em seu sistema que exemplifique bem a finalidade desses diretórios  
a) /etc b) /usr c) /var d) /root e) /boot f) /bin g) /sbin
- 5) Consulte o projeto FHS e indique 3 diretórios não citados no exercício 4 e descreva qual a orientação para uso desses 3 diretórios segundo esse projeto.

## 5 - Bibliografia

Ferreira, E. Rubem. Guia do Administrador Linux. 1 edição, 2003, Novatec Editora, São Paulo.