

1ª Parte: Conhecendo o Linux

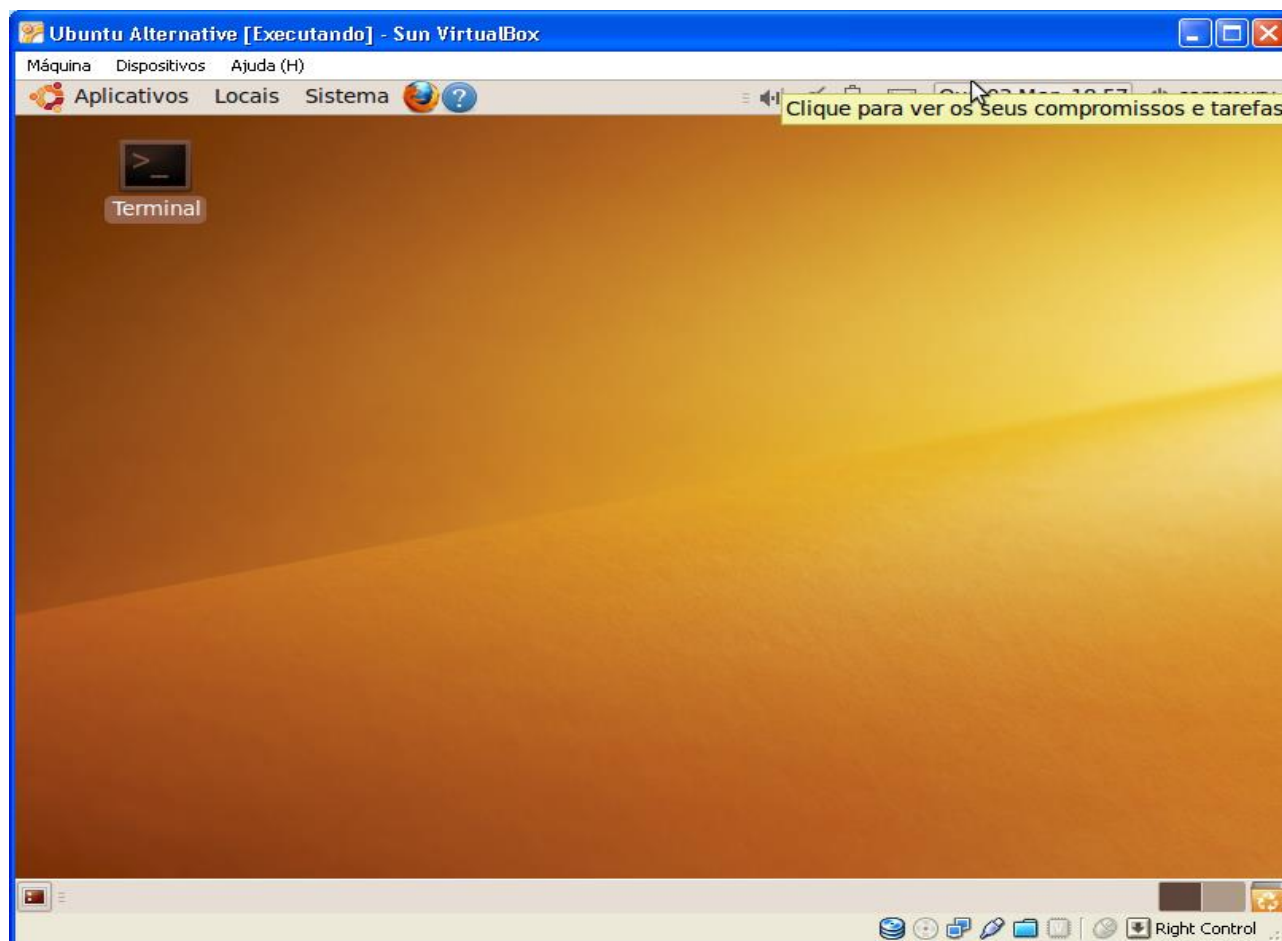
Para facilitar a utilização do Sistema Operacional (SO) Linux, esta primeira seção visa esclarecer alguns pontos na transição dos usuários Windows para o Linux. A maioria das aplicações no Linux segue o mesmo padrão do Windows, porém algumas tarefas corriqueiras como renomear um arquivo, criar um atalho no desktop, etc. podem ter ligeiras diferenças que serão demonstradas a seguir.

Estas aulas têm o intuito de familiarizar o usuário com o sistema de forma geral, não se preocupando em detalhar a história do SO ou mesmo de esgotar os assuntos expostos. As aulas serão divididas em partes sequenciais, ou seja, para que o aluno compreenda uma aula X, é necessário que o mesmo domine todas as aulas anteriores a X.

Inicialmente, navegue pelas opções do Ubuntu Alternate 9.10 – Karmic Koala. Veja os programas instalados, abra-os, compare-os com os do Windows, etc. Quando estiver um pouco mais familiarizado tente fazer as seguintes tarefas:

- a) Crie um atalho na área de trabalho para as seguintes aplicações:
 - a. Terminal
 - b. Editor de texto do Open Office ou AbiWord
 - c. Firefox

Estes aplicativos serão muito utilizados em nossas aulas e devem ficar em local de fácil acesso para facilitar. Obs: Tente descobrir sozinho como realizar esta tarefa. Caso não consiga, pesquise na internet. Se a dúvida ainda persistir, pergunte ao professor.



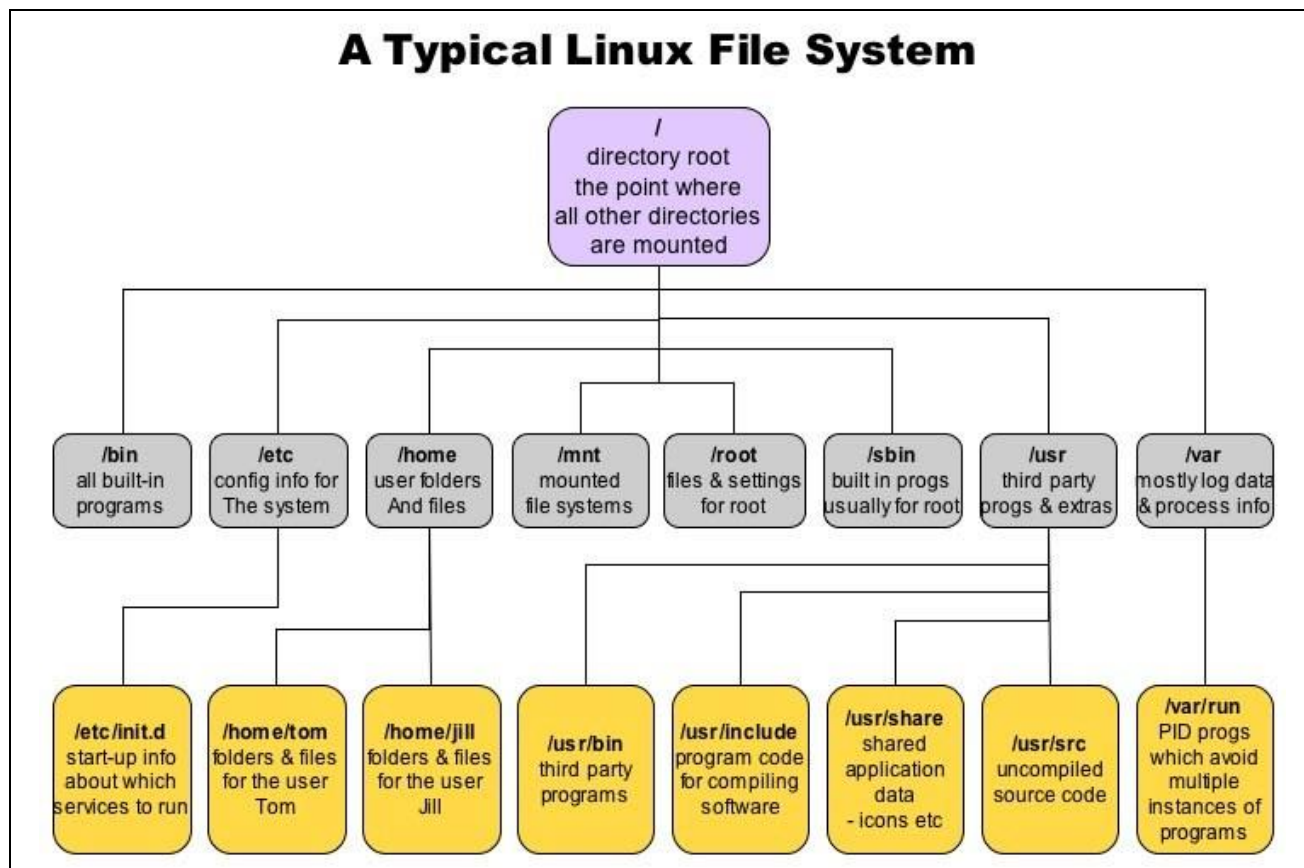
O Linux, em sua forma mais simples, não possui interface gráfica. O sistema operacional pode ser utilizado em sua plenitude através de linhas de comando em terminais. As belas telas e gráficos que você vê quando inicia o seu Linux, não são partes obrigatórias do SO em si, mas sim um adendo, uma interface gráfica, chamada de **X**.

Esta interface gráfica visa facilitar o uso do SO pelos usuários, contudo demanda mais recursos computacionais tais como memória, espaço em disco, processamento, etc. Além disso, o X acaba sendo uma abstração dos terminais Linux e, como toda abstração, tende a oferecer menos recursos que o SO oferece quando operado diretamente pela linha de comando. Por tudo isto, o uso da interface gráfica é facultativo e o número de usuários adeptos a utilização do SO somente pelos terminais é grande.

Como a utilização do Linux através da interface gráfica é parecida com a do Windows vamos, inicialmente, aprender a utilizá-lo sem o X. O acesso aos terminais pode ser feito pela própria interface gráfica; ou “matando” o X, caso o usuário esteja em modo gráfico e queira sair; ou mesmo abrindo um novo terminal em paralelo a interface gráfica, pressionando as tecla CTRL+ALT+FN (N=1,2,3,4,5,6) onde FN representa as teclas F1, F2, F3, etc. Geralmente, o X é carregado no ambiente 7, ou seja, se o usuário deseja voltar para a interface gráfica basta apertar CTRL+ALT+F7. Tudo isto é possível graças ao fato do Linux ser um SO multiusuário/multitarefa.

A ESTRUTURA DE DIRETÓRIOS DO LINUX.

Os diretórios do Linux são organizados em forma de árvore hierarquicamente a partir do diretório raiz '/'. Esta árvore contém os diretórios de usuários, as bibliotecas do sistema, diretórios de dispositivos, etc. De forma geral, uma disposição simplificada dos diretórios das distribuições Linux é:



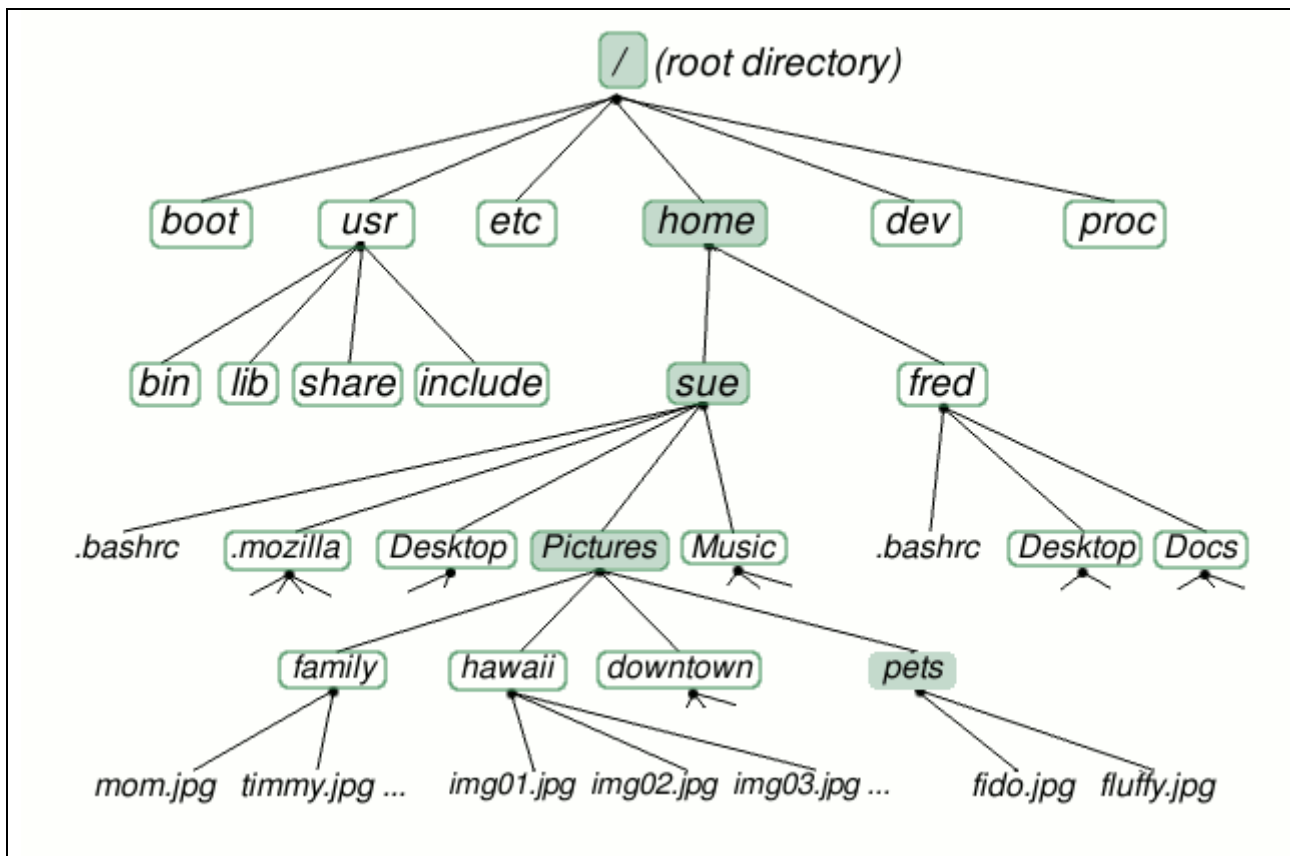


Figura 1: Árvore de diretórios do Linux (simplificada)

/

Diretório raiz do sistema. É abaixo dele que os demais são organizados.

/boot

Contém tudo o que é necessário para o processo de boot. Neste diretório se encontra o kernel e os arquivos de configuração do lilo ou grub, dependendo do gerenciador de boot que você estiver usando.

/home

Os diretórios de trabalho dos usuários do sistema ficam nesse diretório, se você não for utilizar quotas ou se tiver um número muito grande de usuários é aconselhável você criar uma partição para esse diretório.

/root

Este é o diretório de trabalho do root (super-usuário).

/bin

Neste diretório ficam os aplicativos e utilitários do sistema.

/usr/bin

Outros comandos que usuários podem acessar, ele foi introduzido para dividir comandos nos primeiros sistemas Unix que tinham pouco espaço em disco.

/sbin

Este diretório contém as ferramentas para administração e configuração do sistema, somente o super-usuário tem acesso a esse diretório.

/etc

Nesse diretório ficam os arquivos de configuração do sistema, configuração de serviços e outras aplicações.

/dev

Diretório onde estão localizados os arquivos de dispositivo. Os arquivos dentro de /dev representam os dispositivos que são suportados pela versão do seu GNU/Linux.

/lib

Nesse diretório estão as bibliotecas utilizadas para inicialização do sistema e bibliotecas necessárias para rodar outras aplicações além de módulos do kernel.

/mnt

Diretório padrão para a montagem de discos.

/tmp

Nesse diretório estão os temporários de programas.

/proc

Nesse diretório estão os processos que estão rodando no sistema, nele também se encontra informações como: IRQ's, portas de entrada e saída utilizadas, partição swap, etc.

/usr

Nesse diretório é onde fica a grande parte de sua distribuição GNU/Linux, nele estão programas, o sistema de janelas X, o kernel (GNU/Linux), etc.

/usr/src/linux

Nesse diretório contém o seu Kernel (GNU/Linux) do seu sistema operacional, a documentação, e os códigos abertos em linguagem C.

/opt

Nesse diretório devem ser instalados pacotes adicionais.

/var

Nesse diretório contém dados variáveis do sistema.

/root - Diretório HOME do Superusuário.

/home - Diretório HOME dos usuários comuns.

/bin - Comandos utilizados durante o boot e por usuários comuns.

/sbin - Como os comandos do /bin só que não são utilizados pelos usuários comuns.

/proc - Sistema de arquivos virtual (na memória) com dados do Kernel.

/boot - Arquivos utilizados durante a inicialização do sistema e o Kernel.

/dev - Dispositivos (modem, mouse, teclado, etc..).

/etc - Arquivos de configurações do sistema.

/etc/skel - Padrão de arquivos para o diretório HOME de novos usuários.

/etc/sysconfig - Arquivos de configuração do sistema para os dispositivos.

/mnt - Local onde são montados discos e volumes temporários (disquete, outros HDs, CD-ROM, etc..).

/tmp - Arquivos temporários do sistema utilizados antes da inicialização do sistema ter sido concluída.

/var - Contém arquivos que são modificados com o decorrer do uso do sistema (e-mail, temporários, filas de impressão, manuais).

/var/lib - Bibliotecas que mudam enquanto o sistema está rodando.

/var/local - Arquivos variáveis de programas que estão rodando.

/var/lock - Travas para indicar que um programa está utilizando um determinado dispositivo.

/var/log - Arquivos de log do sistema (erros, logins, etc..)

/var/run - Arquivos importantes ao sistema úteis até o próximo boot (atualizações de softwares e kernel).

/var/spool - Diretório de filas de impressão, e-mail e outros
/var/tmp - Arquivos temporários dos programas
/var/catman - Um cache para manuais que são formatados na hora de serem utilizados.
/lib - Bibliotecas compartilhadas necessárias pelos programas do sistema.
/lib/modules - Módulos externos do kernel para dispositivos e funções.
/usr - Contém arquivos de todos os programas e bibliotecas para o uso dos usuários do Linux.
/usr/bin - Executáveis em geral.
/usr/sbin - Executáveis de administração do sistema não necessários pelo kernel, como por exemplo servidores.
/usr/include - Arquivos para serem utilizados em linguagens de programação.
/usr/lib - Bibliotecas dos executáveis encontrados no /usr/bin
/usr/local - Arquivos de programas instalados localmente (apenas para alguns usuários).
/usr/man - Manuais
/usr/info - Documentos de Informações
/usr/X11R6 - Arquivos do X Window System e seus aplicativos.

O AMBIENTE DO TERMINAL / INTERPRETADOR DE COMANDOS

O terminal ou Shell é um interpretador de comandos que faz a interação do usuário com o sistema operacional. Ele atua lendo os comandos digitados pelo usuário e os executa através de seus serviços ou outros programas. Existem diversos tipos de Shell, tais como: bash, sh, csh, etc. Cada um possui suas peculiaridades, que não serão abordadas aqui, ficando a cargo do usuário pesquisar mais informações se julgar necessário.

Para facilitar a consulta a materiais digitais tais como esta aula ou páginas da internet durante a utilização do terminal, fica a sugestão de não desabilitar o X. Abra um novo ambiente (ex: CTRL+ALT+F1) ou use o terminal do próprio X para seguir as aulas.

Para acessar o ambiente do shell fora do X o usuário tem de ser autenticado, ou seja deve efetuar o login digitando seu nome de usuário e sua senha - criados durante a instalação.

Para verificar qual Shell você está utilizando, basta digitar o seguinte comando:

```
echo $SHELL <ENTER>
```

Utilização Genérica de Comandos no Terminal

A sintaxe de uso dos comandos nos terminais Linux é variada, porém apresenta algumas características comuns. De forma geral, tende a seguir o seguinte padrão:

comando [-opções . . .] [argumentos]

Onde:

comando: função/comando que se deseja executar;

opções: descreve particularidades na execução do comando;

argumentos: argumentos de entrada ou saída para serem processados pelo comando.

Atenção! O Linux é “case sensitive”, ou seja, diferencia letras minúsculas de maiúsculas. Logo, um arquivo de nome: “arquivo” é diferente de “Arquivo”.

Finalmente, feita esta introdução mínima, porém extremamente necessária ao Linux, podemos iniciar as aulas de fato!

Abra o terminal e siga digitando os comandos de acordo com as instruções cedidas.

Para listar o conteúdo de um diretório use o comando: ls (listar)

ls <ENTER>

O comando ls faz a listagem dos arquivos e diretórios. Porém, sem o uso de nenhuma opção esta listagem é feita de forma simplificada (apenas o nome), ou seja, não relata maiores detalhes tais como se está ou não oculto, tamanho, se é ou não executável, proprietário, etc. As opções mais comuns do comando ls são:

- a (all - Lista todos os arquivos (inclusive os ocultos) de um diretório).
- F (Formats - Insere um caractere identificador após cada tipo listado: arquivos executáveis ('*'), diretórios ('/'), soquete ('='), link simbólico ('@') e pipe ('|').
- d (directory - Se o argumento for um diretório somente listar seu nome e não seu conteúdo).
- l (long - Usa o formato longo para listagem de arquivos).
- t (time - listar os arquivos por ordem de data de modificação. Arquivos que foram modificados por último, mais recentemente, são exibidos primeiro).

Ainda assim, estas são apenas algumas das diversas opções do comando ls. Para ver todas, pode-se chamar a ajuda do comando através da sintaxe --help. Ex:

ls --help <ENTER>

Com o uso da ajuda do comando, vemos que a sintaxe completa do ls é:

ls [opções] [caminho/arquivo]

Ou seja, se quisermos listar todos os arquivos, inclusive os ocultos, do diretório /home/caramuru deve-se usar o comando:

ls -a /home/caramuru <ENTER>

Quando não é digitado o caminho, o comando **ls** assume que a listagem deverá ser feita para o diretório corrente. É um comando recursivo. Ou seja, para listar os subdiretórios do diretório corrente, pode-se utilizar o comando:

ls -d */ <ENTER>

Faz uma listagem, filtra para o diretório atual e lista os que terminam com o caractere '/');

Às vezes é necessário filtrar a listagem para facilitar a leitura, por isso, o uso dos curingas é de grande valia. Se for necessário listar todos os

Uso do curinga

ls a* <ENTER> - lista os arquivos do diretório atual que iniciam com a letra 'a' e os arquivos e subdiretórios dos diretórios que começam com a letra 'a'.

Obs: Caso a listagem retorne outros arquivos/diretórios além dos que iniciam com a letra 'a', o comando alias deve estar ativado. Futuramente, abordaremos este comando e suas opções. Por enquanto, vamos apenas desabilitá-lo: unalias ls <ENTER>

Caso queria voltar para a formatação antiga na saída do terminal, escreva o comando: `alias ls='ls --color=auto'`

`ls -la *` <ENTER> - o mesmo de cima, incluindo os arquivos e diretórios ocultos.

Por fim, a opção `-l` faz a listagem no formato longo. No Linux, as extensões de arquivos não são obrigatórias, servem apenas para facilitar a associação dos arquivos aos programas necessários para abri-los. O SO aceita nomes de arquivos de até 255 caracteres.

Use o comando `ls -l` e veja a saída:

```
drwxr-xr-x 2 caramuru caramuru 4096 2009-04-08 01:15 Área de Trabalho/
lrwxrwxrwx 1 root      root      9 2009-04-08 01:37 bash -> /bin/bash*
-rw-r--r-- 1 root      root      58 2009-04-07 12:31 contato.txt
drwxr-xr-x 2 caramuru caramuru 4096 2009-04-07 22:41 Documentos/
drwxr-xr-x 2 caramuru caramuru 4096 2009-04-07 02:01 Download/
drwxr-xr-x 2 caramuru caramuru 4096 2009-03-29 18:01 Imagens/
drwxr-xr-x 2 caramuru caramuru 4096 2009-03-29 18:01 Modelos/
drwxr-xr-x 2 caramuru caramuru 4096 2009-03-29 18:01 Música/
-rw-r--r-- 1 root      root      13 2009-04-07 12:57 teste
-rw-r--r-- 1 caramuru caramuru  0 2009-04-12 17:31 teste2
drwx----- 8 caramuru caramuru 4096 2009-04-12 14:29 tmp/
drwxr-xr-x 2 caramuru caramuru 4096 2009-03-29 18:01 vídeos/
```

Observe que a listagem em formato longo possui na primeira coluna a seguinte estrutura:

-rw-r--r-- 1 root root 13 2009-04-07 12:57 teste

Separando por colunas, temos as seguintes informações:

1ª Coluna: -rw-r--r-- / São as permissões de acesso ao arquivo teste.

A primeira letra (da esquerda) identifica o tipo do arquivo, se tiver um “d” é um diretório, se tiver um “-” é um arquivo normal, se for um “l” é um link para um arquivo.

Os próximos 9 caracteres são divididos em 3 grupos de 3 letras e representam, consecutivamente, as permissões de acesso ao arquivo para o dono, grupo e outros usuários. As letras representam:

Letra	Aplicado aos arquivos	Aplicado aos diretórios
r – (Read)	Permite ler o conteúdo do arquivo	Permite listar o conteúdo do diretório
w – (Write)	Permite alterar o conteúdo do arquivo	Permite alterar o conteúdo do diretório
x – (eXecute)	Permite executar um arquivo (programa)	Permite pesquisar o diretório

O conjunto das três letras pode ser associado a um conjunto de 3 bits para atribuição das permissões. Convertendo em binário as posições consecutivas (rwx em 000), (r) seria o bit mais significativo valendo 4 (100 em binário), (w) o do meio igual a 2 (010 em binário) e (x) o menos significativo igual a 1 (001 em binário).

Portanto, quando queremos dar permissão total de acesso a um arquivo, devemos ligar todos os bits (4+2+1=7) ou 111 em binário, ou seja, usamos um 7 decimal que significa atribuir 1 para Read, 1 para Write e 1 para Execute.

Na primeira coluna, a partir da primeira letra, cada sequência de 3 letras representa as permissões de acesso do Usuário, Grupo do usuário e Outros, respectivamente.

2ª Coluna: 1 / Caso seja um arquivo, será mostrado o número 1 (ou o número de *hardlinks* para o arquivo). Se for um diretório, mostra a quantidade de sub-diretórios existentes dentro dele.

3ª Coluna: root / Nome do dono do arquivo teste.

4ª Coluna: root / Nome do grupo que o dono do arquivo teste pertence.

5ª Coluna: 13 / Tamanho do arquivo (em bytes).

6ª Coluna: 2009-04-07 / Data da última modificação do arquivo.

7ª Coluna: 12:57 / Data/Hora da última modificação do arquivo.

8ª Coluna: teste Nome do arquivo.

Agora que você sabe listar arquivos e diretórios vamos tentar trocar de diretórios. O comando `cd` (Change Directory) faz a troca de diretórios, mas você precisa ter a permissão de execução para entrar no diretório. Ele funciona da seguinte forma:

cd [diretório]

Onde:

diretório - diretório que deseja entrar.

Exemplos:

<code>cd /</code>	irá para o diretório raiz.
<code>cd -</code>	retornará ao diretório anteriormente acessado.
<code>cd ..</code>	sobe um diretório.

Exercício: Utilize o comando `cd` e passeie pelos diretórios diversos de sua distribuição. Tente acessar diretórios ocultos (começam com um ponto ".") e descubra diretórios que não tenha acesso.

O comando **pwd** (Print Working Directory) mostra o nome e caminho do diretório atual. Escreva `pwd` e veja o caminho completo até o diretório atual ser mostrado.

Através do comando `mkdir` (Make Directory) é possível criar diretórios, a sintaxe correta é:

mkdir [opções] [caminho/diretório] [caminho1/diretório1]

Opções:

`[-p]` Permite a criação de nome_do_diretório sob um diretório pai que não existe, que foi especificado no caminho para o nome_diretório. Cria diretórios recursivamente.

`[-v]` Mostra mensagem para cada diretório criado.

Por exemplo, para criar um único diretório "teste" dentro do diretório já existente /tmp, utiliza-se o comando:

mkdir/tmp/teste

Para criar vários diretórios e subdiretórios recursivamente utiliza-se:

mkdir -p /home/caramuru/novodir1/novodir2/novodir3

Onde somente o diretório /home/caramuru já existia, e outros foram criados de uma só vez.

cp – (Copy) Copia arquivos.

Sintaxe: cp [opções] [origem] [destino]

Opções:

- [-i] Execução no modo interativo. Pede confirmação caso o arquivo destino exista;
- [-p] Duplica também as permissões juntamente com data/hora;

Exemplos:

cp teste.txt teste1.txt	Copia o arquivo teste.txt para teste1.txt.
cp teste.txt /tmp	Copia o arquivo teste.txt para dentro do diretório /tmp.

rm – (Remove) Apaga arquivos e diretórios.

Sintaxe: rm [opções][caminho][arquivo/diretório] [caminho1][arquivo1/diretório1]

Opções:

- [-i] Pede confirmação antes de apagar cada arquivo/diretório;
- [-f] Não pede confirmação para remover um arquivo protegido contra escrita, nem mostra uma mensagem se o arquivo não existe;
- [-r] Esta opção é usada para apagar um diretório, e remove todos os arquivos e sub-diretórios que ele porventura contenha;
- [-R] Equivalente à opção -r;
- [-e] Imprime uma mensagem após a remoção de cada arquivo.

rm teste.txt - Apaga o arquivo teste.txt no diretório atual.

rmdir – (Remove Directory) Remove um diretório vazio.

Sintaxe: rmdir [caminho/diretório] [caminho1/diretório1]

Opções:

- [-p] remove diretório e seus ancestrais.
- [-v] Mostra mensagem para cada diretório processado.

Exemplo: Para excluir o subdiretório teste do diretório corrente utiliza-se:

rmdir teste

Para excluir os diretórios encadeados (todos vazios):

```
rmdir -p aluno/novodir1/novodir2/novodir3
```

Porém, o comando `rmdir` não remove diretórios que não estejam vazios. Para remover diretórios que não estejam vazios, utilize o comando `rm` com a opção `-r`:

```
rm -r /home/caramuru/novodir1
```

Desta forma, o diretório “novodir1” e todos os seus subdiretórios são excluídos de uma só vez, porém exigindo confirmação para cada exclusão.

Exercícios:

Tente entrar em cada uma das pastas do diretório raiz. Anote o nome dos diretórios em que o acesso foi negado.

No ambiente gráfico, crie 5 arquivos dentro do diretório `/home/$USER`: `a.txt`, `b.txt`, `c.txt`, `d.txt`, `e.txt`.

Através dos comandos ensinados, crie um diretório “testes” e os subdiretórios “a, b, c, d, e” e copie os respectivos arquivos para cada diretório de mesma inicial. Entre no diretório “testes”.

Anote os comandos necessários para realizar as tarefas.

Interprete o resultado de cada comando abaixo:

- a) `ls -a b*`
- b) `ls a*`
- c) `ls *`
- d) `ls /`
- e) `ls /home/$USER`
- f) `ls ?`

Anote os comandos necessários para realizar as tarefas.

No shell, crie os diretórios “vogais” e “consoantes” e copie os arquivos de iniciais correspondentes para o diretório adequado.

Anote os comandos necessários para realizar as tarefas.

Crie o diretório “.oculto” e copie todos os arquivos para dentro dele.

Anote os comandos necessários para realizar as tarefas.

mv – (Move) Move ou renomeia arquivos e diretórios.

Sintaxe: mv [opções] [origem] [destino]

Opções:

- `[-f]` Suprime todas as mensagens (force);
- `[-i]` Modo interativo (confirma mudanças).

[-u] Move somente quando o arquivo destino é mais novo que o destino, ou quando o arquivo não existir no destino.

Exemplos:

<code>mv teste.txt teste1.txt</code>	Renomeia o arquivo teste.txt para teste1.txt.
<code>mv teste.txt /tmp</code>	Move o arquivo teste.txt para /tmp. Lembre-se que o arquivo de origem é apagado após ser movido.

Agora que você já está um pouco mais ambientado com a forma de utilização do Shell, os textos serão mais simples, focados apenas em passar a sintaxe de uso dos comandos e suas funções. Isto permitirá que seu desenvolvimento seja mais rápido e tornará as aulas mais interessantes.

echo - Mostra mensagens. Este comando é útil na construção de scripts para mostrar mensagens na tela para o usuário acompanhar sua execução.

Sintaxe: echo [mensagem]

A opção `-n` pode ser usada para que não ocorra o salto de linha após a mensagem ser mostrada.

Exemplo:

```
echo CEFET-MG <ENTER>
```

cat - Lê dados e escreve dados, permitindo uma grande flexibilidade em sua utilização. Pode-se utilizá-lo para criar, listar e estender o conteúdo de arquivos.

Sintaxe: cat [-n -v] [arquivo ...]

Opções:

`[-n]` Numera linhas de saída.

`[-v]` Apresenta caracteres não imprimíveis usando a notação `^X`.

Exemplo:

Para criar um arquivo de texto chamado “arq.dat” digita-se:

```
cat > arq.dat <ENTER>
```

O arquivo “arq.dat” será criado e aberto para edição, logo digite o seguinte texto:

```
CEFET-MG <ENTER>
```

```
Divinópolis<ENTER>
```

```
<CTRL+D >
```

Para visualizar o arquivo criado digite:

```
cat arq.dat
```

Para enumerar as linhas do arquivo digite:

```
cat -n arq.dat
```

O re-direcionador ">" é utilizado para dar a saída para um arquivo, exemplo:

```
echo teste > arquivo <ENTER>
```

Irá criar um arquivo de nome "arquivo" com o conteúdo "teste".

Veja o conteúdo do arquivo listagem.txt quando se usa:

```
ls -a > listagem.txt
```

find - Procura por arquivos/diretórios no disco.

Sintaxe: find [diretório] [opções/expressão]

find / -name nomearq - Procura no diretório raiz e sub-diretórios um arquivo/diretório chamado nomearq.

find . -type d -name nomeDir - Procura no diretório atual e seus subdiretórios um diretório chamado nomeDir.

Obs: /run/user/1000/gvfs é um sistema de arquivos virtual que pertence ao Gnome e o permite rodar recursos como FTP, servidores SAMBA, etc. Só é acessível ao usuário rodando o Gnome e mesmo o root não consegue sobrepujar esta restrição.

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

Sintaxe: clear

date - Permite ver/modificar a Data e Hora do Sistema. Você precisa estar como usuário root para modificar a data e hora.

Sintaxe: date MesDiaHoraMinuto[AnoSegundos]

Se quiser mudar a Data para 25/12 e a hora para 08:15 digite:

```
date 12250815
```

Opções:

[%H] Hora em formato de 24h;

[%M] Minutos;

[%S] Segundos;

[%d] Dia do mês

[%m] Mês;

[%Y] Ano;

```
$ date +"%Y-%m-%d %H:%M"
```

```
2013-09-16 17:30
```

Para trocar a hora e data:

```
$ date --set HH:MM:SS  
$ date --set YYYY-MM-DD
```

cal - **cal** apresenta o calendário do ano e mês especificados. Na falta dos argumentos, são assumidos o mês e ano corrente.

Sintaxe: cal [mês [ano]]

cal march 2010

cal 2010

sort - **Organiza as linhas de um arquivo texto ou da entrada padrão.**

Sintaxe: sort [opções] [arquivo]

Onde:

arquivo é o nome do arquivo que será organizado. Caso não for especificado, será usado o dispositivo de entrada padrão (teclado).

Opções:

- [-b] Ignora linhas em branco.
- [-d] Somente usa letras, dígitos e espaços durante a organização.
- [-f] Ignora a diferença entre maiúsculas e minúsculas.
- [-r] Inverte o resultado da comparação.
- [-n] Caso estiver organizando um campo que contém números, os números serão organizados na ordem aritmética.
- [-c] Verifica se o arquivo já está organizado. Caso não estiver, retorna a mensagem “disorder on arquivo”.
- [-o] arquivo Grava a saída do comando sort no arquivo.
- [-t] delimitador de campo de coluna.
- [-k] indicador de coluna para o sort.

Exemplos:

sort texto.txt - Organiza o arquivo texto.txt em ordem crescente.

sort -r texto.txt - Organiza o conteúdo do arquivo texto.txt em ordem decrescente.

cat texto.txt | sort - Faz a mesma coisa que o primeiro exemplo, só que neste caso a saída do comando cat é redirecionado a entrada padrão do comando sort.

sort -f texto.txt - Ignora diferenças entre letras maiúsculas e minúsculas durante a organização.

Crie o arquivo “email.txt” com o seguinte conteúdo:

```
Jão!xjoao@gmail.com!CEFET-MG  
Maria!maria@hotmail.com!UFMG  
Aline!aline@yahoo.com!UOL
```

`sort -r -k3 -t! email.txt > email_ordenado.txt` – Ordena inversamente a terceira coluna delimitada por “!”.

grep – (Global Regular Expression Pattern) Procura por um texto dentro de um arquivo(s) ou no dispositivo de entrada padrão.

Sintaxe: grep [expressão] [arquivo] [opções]

Onde:

expressão - palavra ou frase que será procurada no texto. Se tiver mais de 1 palavra deve ficar entre aspas

arquivo - Arquivo onde será feita a procura.

Opções:

-A [número] Mostra o [número] de linhas após a linha encontrada pelo grep.

-B [número] Mostra o [número] de linhas antes da linha encontrada pelo grep.

-f [arquivo] Especifica que o texto que será localizado, esta no arquivo [arquivo].

-h, --no-filename Não mostra os nomes dos arquivos durante a procura.

-i, --ignore-case Ignora diferença entre maiúsculas e minúsculas no texto procurado e arquivo.

-n, --line-number Mostra o número de cada linha encontrada pelo grep.

-U, --binary Trata o arquivo que será procurado como binário.

-v, --invert-match Seleciona somente as linhas NÃO coincidentes

Se não for especificado o nome de um arquivo ou se for usado um hífen “-”, grep procurará a string no dispositivo de entrada padrão.

Exemplo:

`grep “capitulo” texto.txt`

`ps ax | grep inetd`

Para pesquisar as entradas que começam com a letra “K” no arquivo teste.txt, fazemos:

`grep ^K teste.txt`

Para pesquisar as entradas que terminam com a letra “a” no arquivo teste.txt, fazemos:

`grep a$ teste.txt`

`grep -i “palavra” livro.txt`

Procura por “palavra” no arquivo “livro.txt”, SEM case sensitive.

Vamos criar um arquivo contendo o nome de algumas pessoas:

`cat > arquivo <ENTER>`

`Joana <ENTER>`

`Carlos <ENTER>`

`Cassio <ENTER>`

`Assuncao <ENTER>`

Zeao <ENTER>

CTRL+D

Para “filtrar” os nomes que contém dois “s” seguidos, fazemos:

grep “ss” arquivo <ENTER>

Para saber em qual linha se encontram os valores encontrados (“ss”) utiliza-se:

grep -n “ss” arquivo <ENTER>

O comando grep também pode ser utilizado para “filtrar” a saída de outros comandos, por exemplo:

Para filtrar conteúdo de uma listagem ls para os arquivos que contenham a palavra trabalho, pode-se utilizar:

ls -a | grep trabalho <ENTER>

cut - Mostra seções de cada linha do arquivo dependendo das opções passadas ao programa.

Sintaxe: cut [opções] [arquivo]

Onde:

arquivo - Arquivo que será verificado pelo comando cut.

Opções

[-b] (bytes) Mostra somente a lista de [bytes] do arquivo.

[-c] (characters) [numero] Mostra somente o [número] de caracteres no arquivo.

[-f] (field) [campos] Mostra os [campos] correspondentes.

[-d] (delimite) Para uso com a opção -f, os campos são separados pelo primeiro caractere em [delimitador] ao invés de tabulações.

[-s] Para uso com a opção -f, somente mostra linhas que contém o caractere separador de campos. Devem ser especificadas opções para o funcionamento deste comando.

Exemplos:

cut -b 1,3 /etc/passwd - Pega a primeira e terceira letra (byte) de cada linha do arquivo /etc/passwd

cut -b 1,3-10 /etc/passwd - Pega a primeira letra (byte) e da terceira a décima letra de cada linha do arquivo /etc/passwd.

cut -c 1,3-10 /etc/passwd - Pega o primeiro caractere e terceiro ao décimo caractere de cada linha do arquivo /etc/passwd.

Crie as instruções abaixo e crie o arquivo colunas.dat:

cat > colunas.dat <ENTER>

carolina:100 <ENTER>

marcia:10 <ENTER>

aline:9 <ENTER>

CTRL+D

Veja o que acontece quando se usa o comando:

cut -f2 -d: colunas.dat

wc – Conta o número de linhas, palavras e caracteres de uma entrada especificada como parâmetro.

Sintaxe: wc [-lwc] [arquivo ...]

Opções:

[-c] Conta caracteres

[-l] Conta linhas

[-w] Conta palavras

Exemplo, para saber o número de linhas, palavras e caracteres do arquivo teste.txt:

wc teste.txt <ENTER>

Para contar apenas o número de linhas do arquivo teste.txt, fazemos:

wc -l teste.txt <ENTER>

Quase todos os programas GNU/Linux possuem manuais que ensinam a sintaxe correta de uso.

A utilização da página de manual é simples, digite:

Sintaxe: man [seção] [comando/arquivo]

Onde:

seção - É a seção de manual que será aberta, se omitido, mostra a primeira seção sobre o comando encontrada (em ordem crescente).

comando/arquivo - Comando/arquivo que deseja pesquisar.

A navegação dentro das páginas de manual é feita usando-se as teclas:

q - Sai da página de manual

PageDown ou f - Rola 25 linhas abaixo

PageUP ou w - Rola 25 linhas acima

SetaAcima ou k - Rola 1 linha acima

SetaAbaixo ou e - Rola 1 linha abaixo

r - Redesenha a tela (refresh)

p ou g - Início da página

h - Ajuda sobre as opções da página de manual

s - Salva a página de manual em formato texto no arquivo especificado (por exemplo: /tmp/ls).

Exemplo: man grep, man 5 hosts_access.

tail - Mostra as linhas a partir de um deslocamento em um arquivo texto.

Sintaxe: tail [opções]

Onde:

[-c] [numero] Mostra o [numero] de bytes/caracteres do final do arquivo.

[-n] [numero] Mostra o [numero] de linhas do final do arquivo.

Exemplos:

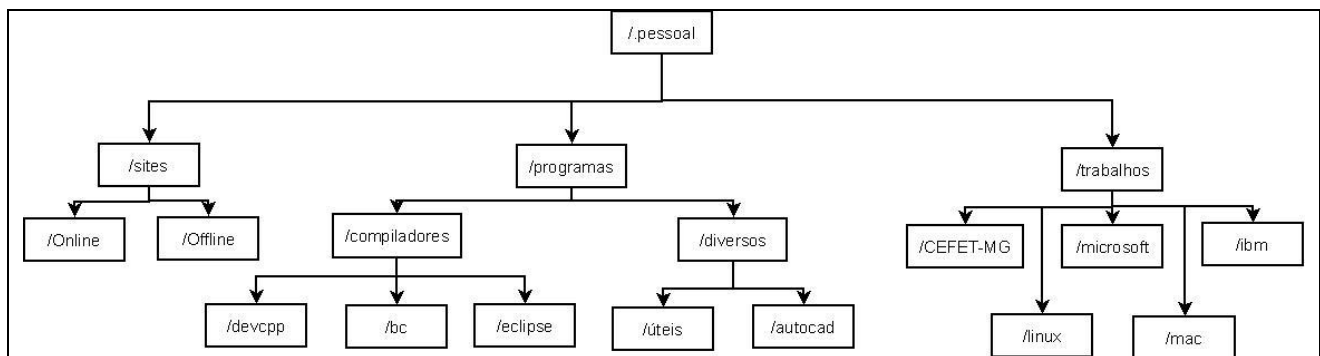
tail teste.txt

tail -n 20 teste.txt.

Lista de exercícios

Faça e entregue para o professor o resultado (passos necessários):

- Pesquisar os comandos de redirecionamento de fluxo (>, >>, |, etc.);
- O que faz cada um dos seguintes comandos?
 - ls *
 - ls *.c
 - ls ?.c
 - find /home -name 'trabalho.c'
 - find . -name '*.c'
 - grep -n main arquivo.c
- Qual a diferença entre utilizar o comando “ls -l” e “ls -la”?
- Crie um diretório oculto “.pessoal” em sua pasta de usuário e:
 - Crie a seguinte estrutura de subdiretórios:
 - Jogue esta estrutura dentro de um arquivo e envie para o professor.



- Crie um arquivo “agenda.txt” e adicione o conteúdo abaixo – através do comando “cat” ou usando o X. Imprima a saída na tela e veja o resultado. Ainda sobre este arquivo:
 - Ordenar o arquivo pelo campo nome e salvar o conteúdo no arquivo ordnome.txt
 - Ordenar o arquivo em ordem descendente de número de telefone e salvar em orddtel.txt
 - Mostrar apenas os endereços
 - Mostrar apenas os telefones
 - Mostrar a linha com a informação de contendo a palavra “Caramuru”

Aline Riscado#5432-8954#Grajaú

Daniela Cicarelli#3289-8967#Sion

Pamela Anderson#7856-2345#Palm Beach

Luize Altenhofen#3498-2367#Florianópolis

Angelina Jolie#3478-8967#Mui Tyau

Marcelo Caramuru#9999-9898#Divinópolis

Caroline Dieckman#6578-7890#Barra da Tijuca

Juliana Paes#2345-6789#Ipanema
Alinne Moraes#1265-6789#Morumbi
Gisele Bündchen#4576-8967#Florida
Tatiana Araújo#9256-8967#Copacabana
Sheila Carvalho#3456-6723#Salvador

- f) Crie um arquivo “musicas.txt” em sua máquina virtual e adicione o conteúdo do arquivo “ex1.txt” através do comando “cat” ou usando o X. Para permitir copiar/colar entre a máquina real e a virtual é necessário que os adicionais para convidado estejam instalados e configurados (aba Dispositivos-> Área de transferência compartilhada -> Bi-direcional). Gerar um arquivo de saída contendo a saída de cada item abaixo:
- a. Data; Hora; Tamanho; Nome da banda.
 - b. Todas as músicas do Guns and Roses
 - c. Ordenar o arquivo “musica.txt” por cada tipo abaixo salvando o resultado em outros arquivos:
 - i. Data (OrdDataMusica.txt); Hora(OrdHoraMusica.txt); Tamanho; Nome da banda.
 - d. Pesquise as 5 melhores bandas que achar no arquivo “musicas.txt” e salve todas as músicas das cinco bandas em um único arquivo chamado “5melhores.txt”. Dica: utilize o operador “>>”.
 - e. Descubra quantas músicas do Iron Maiden existem no arquivo.

Referências Bibliográficas

Marchetti, Edson. Apostila Linux – Módulo Básico. CEFET-MG. Divinópolis, 2007.

Silva, Gleydson Mazioli. Guia Foca GNU/Linux. Versão de 27/11/2007.

Webber, Celso Kopp. Apostila de Linux Básico. Extraído da Internet em jul/2007.

Bonan, Adilson R. Configurando e usando o sistema operacional Linux. São Paulo: Ed. Futura, 2ª Ed. 2002.