



# Comandos

## Explicações Básicas

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Prof. Edwar Saliba Júnior

Julho / 2017



## Leitura

- Textos para leitura:
  - GNU/Linux - Introdução
  - Dicas de Migração DOS/Windows para GNU/Linux



## Explicações Básicas

- Arquivos:
  - É onde gravamos nossos dados;
  - Cada arquivo deve ser identificado por um nome;
  - O Linux é *case-sensitive*, ou seja, diferencia letras maiúsculas de minúsculas;
  - Um arquivo oculto no GNU/Linux é identificado por um "." no início do nome (por exemplo, `.bashrc`). Arquivos ocultos não aparecem em listagens normais de diretórios, deve ser usado o comando `ls -a` para também listar arquivos ocultos.



## Extensão dos Arquivos

- A extensão serve para identificar o tipo do arquivo. A extensão são as letras após um "." no nome de um arquivo, explicando melhor:
  - relatório.txt - O .txt indica que o conteúdo é um arquivo texto;
  - script.sh - Arquivo de Script (interpretado por /bin/sh);
  - system.log - Registro de algum programa no sistema;
  - arquivo.gz - Arquivo compactado pelo utilitário gzip e
  - index.html - Página de Internet (formato Hypertexto).
- A extensão de um arquivo também ajuda a saber o que precisamos fazer para abri-lo. Por exemplo, o arquivo **relatório.txt** é um texto simples e podemos ver seu conteúdo através do comando cat, já o arquivo index.html contém uma página de Internet e precisaremos de um navegador para poder visualizá-lo (como o lynx, Firefox ou o Konqueror).
- A extensão, na maioria dos casos, não é requerida pelo sistema operacional GNU/Linux, mas é conveniente o seu uso para determinarmos facilmente o tipo de arquivo e que programa precisaremos usar para abri-lo.



## Tipos de Arquivos

- **Texto:**
  - seu conteúdo é compreendido pelas pessoas. Um arquivo texto pode ser uma carta, um script, um programa de computador escrito pelo programador, arquivo de configuração e etc.
- **Binário:**
  - seu conteúdo somente pode ser entendido por computadores. Contém caracteres incompreensíveis para pessoas normais.



## Diretórios

- Diretório é o local utilizado para armazenar conjuntos arquivos para melhor organização e localização. O diretório, como o arquivo, também é *case-sensitive* (diretório /teste é completamente diferente do diretório /Teste).



## Diretório Raiz

- Este é o diretório principal do sistema. Dentro dele estão todos os outros diretórios do sistema;
- O diretório Raiz é representado por uma "/", assim se você digitar o comando `cd /` você estará acessando este diretório.



## Estrutura de Subdiretórios

- A estrutura de diretórios e subdiretórios pode ser identificada da seguinte maneira:

```
/
/bin
/sbin
/usr
/usr/local
/mnt
/tmp
/var
/home
```





## Diretório Atual

- É o diretório em que nos encontramos no momento. Você pode digitar **pwd** para verificar qual é seu diretório atual;
- O diretório atual também é identificado por um "." (ponto).



## Diretório Home

- Também chamado de diretório de usuário. Em sistemas GNU/Linux cada usuário (inclusive o root) possui seu próprio diretório onde poderá armazenar seus programas e arquivos pessoais;
- Este diretório está localizado em `/home/[login]`, neste caso se o seu login for "joao" o seu diretório home será `/home/joao`. O diretório home também é identificado por um `~` (til), você pode digitar tanto o comando **`ls /home/joao`** como **`ls ~`** para listar os arquivos de seu diretório home;
- O diretório home do usuário root, na maioria das distribuições GNU/Linux, está localizado em `/root`.



## Diretório Superior

- O diretório superior (*Upper Directory*) é identificado por `..` (2 pontos);
- Caso estiver no diretório `/usr/local` e quiser listar os arquivos do diretório `/usr` você pode digitar **`ls ..`**
- Este recurso também pode ser usado para copiar, mover arquivos/diretórios, etc.



## Diretório Anterior

- O diretório anterior é identificado por "`-`" (sinal de subtração). É útil para retornar ao último diretório usado;
- Se estiver no diretório `/usr/local` e digitar **`cd /lib`**, você pode retornar facilmente para o diretório `/usr/local` usando **`cd -`**.



## Estrutura de Diretórios do Linux

- `/bin`
  - Contém arquivos programas do sistema que são usados com frequência pelos usuários.
- `/boot`
  - Contém arquivos necessários para a inicialização do sistema.
- `/cdrom`
  - Ponto de montagem da unidade de CD-ROM.
- `/media`
  - Ponto de montagem de dispositivos diversos do sistema (rede, pendrives, CD-ROM em distribuições mais novas).
- `/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.
- `/lost+found`
  - Local para a gravação de arquivos/diretórios recuperados pelo utilitário `fsck.ext2`. Cada partição possui seu próprio diretório `lost+found`.



## Estrutura de Diretórios do Linux

- `/mnt`
  - Ponto de montagem temporário.
- `/proc`
  - Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.
- `/sys`
  - Sistema de arquivos do kernel. Este diretório não existe em seu disco rígido, ele é colocado lá pelo kernel e usado por diversos programas que fazem sua leitura, verificam configurações do sistema ou modificar o funcionamento de dispositivos do sistema através da alteração em seus arquivos.
- `/root`
  - Diretório do usuário root.
- `/sbin`
  - Diretório de programas usados pelo superusuário (root) para administração e controle do funcionamento do sistema.
- `/tmp`
  - Diretório para armazenamento de arquivos temporários criados por programas.
- `/usr`
  - Contém maior parte de seus programas. Normalmente acessível somente como leitura.
- `/var`
  - Contém maior parte dos arquivos que são gravados com frequência pelos programas do sistema, e-mails, spool de impressora, cache, etc.



## Nomeando Arquivos e Diretórios

- No GNU/Linux, os arquivos e diretórios podem ter o tamanho de até 255 letras;
- Podem possuir uma extensão (um conjunto de letras separadas do nome do arquivo por um ".");
- Os programas executáveis do GNU/Linux, ao contrário dos programas de DOS e Windows, não são executados a partir de extensões .exe, .com ou .bat. O GNU/Linux (como todos os sistemas POSIX) usa a permissão de execução de arquivo para identificar se um arquivo pode ou não ser executado.



## Comandos

- Comandos são ordens que passamos ao sistema operacional para executar uma determinada tarefa;
- Cada comando tem uma função específica, devemos saber a função de cada comando e escolher o mais adequado para fazer o que desejamos, por exemplo:

**ls** - para mostrar arquivos de diretórios e

**cd** - para mudar de diretório.





## Comandos

- Um comando pode receber opções e parâmetros:
- Opções:
  - as opções são usadas para controlar como o comando será executado, por exemplo, para fazer uma listagem mostrando o dono, grupo, tamanho dos arquivos você deve digitar **ls -l**.
- Opções podem ser passadas ao comando através de um "-" ou "--":
  - "-" opção identificada por uma letra. Podem ser usadas mais de uma opção com um único hífen. O comando `ls -l -a` é a mesma coisa de `ls -la`
  - "--" opção identificada por um nome. Também chamado de opção extensa. O comando `ls --all` é equivalente a `ls -a`;
- Pode ser usado tanto "-" como "--", mas há casos em que somente "-" ou "--" esta disponível.



## Comandos

- Parâmetros
  - Um parâmetro identifica o caminho, origem, destino, entrada padrão ou saída padrão que será passada ao comando.
- Se você digitar: **ls /usr/share/doc/copyright**, então `/usr/share/doc/copyright` será o parâmetro passado ao comando `ls`, neste caso queremos que se liste os arquivos do diretório `/usr/share/doc/copyright`;
- Se o nome de um comando é digitado errado, então é mostrada a mensagem `command not found` (comando não encontrado).
- Por exemplo: "`ls -la /usr/share/doc`", `ls` é o comando, `-la` é a opção passada ao comando e `/usr/share/doc` é o diretório passado como parâmetro ao comando `ls`.



## Terminal Virtual

- Terminal (ou console) é o teclado e tela conectados em seu computador;
- No GNU/Linux, em modo texto, você pode acessar outros terminais virtuais segurando a tecla `ALT` e pressionando as teclas de `F1` a `F6`. Cada tecla de função corresponde a um número de terminal do 1 ao 6 (o sétimo é usado por padrão pelo ambiente gráfico X);
- O GNU/Linux possui mais de 63 terminais virtuais, mas apenas 6 estão disponíveis inicialmente por motivos de economia de memória RAM.

## Terminal Virtual

- Se estiver usando o modo gráfico, você deve segurar **CTRL + ALT** enquanto pressiona uma tecla de **<F1>** a **<F6>** para entrar em um terminal;
- Para voltar ao modo gráfico, pressione **CTRL + ALT + <F7>**
- O comando **CTRL + D** encerra o terminal.



## Curingas

- Curingas (ou referência global) é um recurso usado para especificar um ou mais arquivos ou diretórios do sistema de uma só vez;
- Este é um recurso que permite que você faça a filtragem do que será listado, copiado, apagado, etc.
- São usados 4 tipos de curingas no GNU/Linux:
  - "\*" - faz referência a um nome completo/restante de um arquivo/diretório;
  - "?" - faz referência a uma letra naquela posição.



## Curingas

- [padrão] - Faz referência a uma faixa de caracteres de um arquivo/diretório. Padrão pode ser:
  - [a-z][0-9] - faz referência a caracteres de **a** até **z** seguido de um caractere de **0** até **9**;
  - [a,z][1,0] - faz referência aos caracteres **a** e **z** seguido de um caractere **1** ou **0** naquela posição;
  - [a-z,1,0] - faz referência a um intervalo de caracteres de **a** até **z** ou **1** ou **0** naquela posição.
- A procura de caracteres é *case-sensitive* assim se você deseja que sejam localizados todos os caracteres alfabéticos você deve usar [a-zA-Z].
- Caso a expressão seja precedida por um **^**, faz referência a qualquer caractere exceto o da expressão.
  - Por exemplo: [^abc] faz referência a qualquer caractere exceto **a**, **b** e **c**.



## Curingas

- {padrões} - expande e gera *strings* para pesquisa de padrões de um arquivo/diretório;
  - X{ab,01} - faz referência a sequência de caracteres **Xab** ou X01
  - X{a-z,10} - faz referência a sequência de caracteres **Xa-z** e X10.
- O que diferencia este método de expansão dos demais é que a existência do arquivo/diretório é opcional para geração do resultado. Isto é útil para a criação de diretórios;
- Lembrando que os 4 tipos de curingas ("\*", "?", "[ ]", "{ }") podem ser usados juntos.



## Curingas

- Exemplos:
  - Vamos dizer que existam 5 arquivos no diretório `/usr/teste`: `teste1.txt`, `teste2.txt`, `teste3.txt`, `teste4.new` e `teste5.new`.
  - Caso deseje listar todos os arquivos do diretório `/usr/teste` você pode usar o coringa `"*"` para especificar todos os arquivos do diretório:  
`cd /usr/teste e ls *` ou `ls /usr/teste/*`.
  - Não tem muito sentido usar o comando `ls` com `"*"` porque todos os arquivos serão listados se o `ls` for usado sem nenhum Coringa;
  - Agora para listar todos os arquivos `teste1.txt`, `teste2.txt` e `teste3.txt` com exceção de `teste4.new` e `teste5.new`, podemos usar inicialmente 3 métodos:
    - Usando o comando `ls *.txt` que pega todos os arquivos que começam com qualquer nome e terminam com `.txt`.
    - Usando o comando `ls teste?.txt`, que pega todos os arquivos que começam com o nome `teste`, tenham qualquer caractere no lugar do coringa `?` e terminem com `.txt`. Com o exemplo acima `teste*.txt` também faria o mesmo efeito, mas se também tivéssemos um arquivo chamado `teste10.txt` este também seria listado.





## Curingas

- Continuando o exemplo anterior:
  - usando o comando `ls teste[1-3].txt`, que pega todos os arquivos que começam com o nome teste, tenham qualquer caractere entre o número 1-3 no lugar da 6a letra e terminem com `.txt`. Neste caso se obtém uma filtragem mais exata, pois o coringa `?` especifica qualquer caractere naquela posição e `[]` especifica números, letras ou intervalo que será usado.
- Agora para listar somente `teste4.new` e `teste5.new` podemos usar os seguintes métodos:
  - `ls *.new` que lista todos os arquivos que terminam com `.new`
  - `ls teste?.new` que lista todos os arquivos que começam com teste, contenham qualquer caractere na posição do coringa `?` e terminem com `.new`.
- `ls teste[4,5].*` que lista todos os arquivos que começam com teste contenham os números de 4 ou 5 naquela posição e terminem com qualquer extensão.



## Referências

- GUIA FOCA GNU/Linux. Iniciante. Disponível em: <http://www.guiafoca.org/cgs/guia/iniciante/ch-intro.html>. Acesso em: 27 jul. 2017.