



## **GNU / Linux**

Prof. Edwar Saliba Júnior  
Março de 2009



## Conceitos Básico

- O S.O. é *case sensitive*, ou seja, sensível à caixa;
  - **Exemplo:** o arquivo “apple.txt” é diferente do arquivo “Apple.txt” e podem conviver juntos no mesmo diretório;
  - Esta regra também é válida para comandos e diretórios.



## Dispositivos

- Os dispositivos são identificados de uma forma diferente da usada no DOS. Por exemplo:

DOS/Windows	Linux
A:	/dev/fd0
B:	/dev/fd1
C:	/dev/hda1 ou /dev/sda1
LPT1	/dev/lp0
LPT2	/dev/lp1
LPT3	/dev/lp2
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3



## Arquivos

- Um arquivo pode ser do tipo binário ou texto;
- Deve conter um nome;
- Um arquivo oculto é identificado por “.” no início do nome (por exemplo: .bashrc);
- Arquivos ocultos não aparecem em listagens normais de diretórios.



## Arquivos - Algumas Extensões

- A extensão serve para identificar o tipo do arquivo:
  - Relatório.**txt** → Arquivo texto;
  - Script.**sh** → Arquivo de *script* interpretado por “/bin/sh”;
  - System.**log** → Registro de algum programa do sistema;
  - Arquivo.**gz** → Arquivo compactado pelo utilitário gzip;
  - Index.**html** → Página da Internet (formato



## Diretórios

- Local utilizado para armazenar conjuntos de arquivos;
- No GNU / Linux o diretório é especificado por uma “/”, ao contrário do DOS onde usa-se a barra invertida “\”;
- Diretório Raiz: Este é o principal diretório do sistema. Dentro dele estão todos os demais diretórios. O diretório Raiz é representado por “/”, ou seja, se você digitar o comando: “cd /”, você estará acessando o diretório raiz.



## Árvore de Diretórios

- Alguns dos principais diretórios do GNU/Linux:
  - /
  - /bin - Contém arquivos de programas do sistema que são usados com frequência pelos usuários;
  - /boot - Contém arquivos necessários para a inicialização do sistema;
  - /usr - Contém maior parte de seus programas. Normalmente acessível somente como leitura;
  - /mnt - Ponto de montagem temporário;
  - /tmp - Diretório para armazenamento de arquivos temporários criados por programas;
  - /var - Contém a maior parte dos arquivos que são gravados com frequência pelos programas do sistema, *e-mails*, *pool* de impressora, *cache*, etc.;
  - /home - Diretórios contendo os arquivos dos usuários.



## 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 “joel” o seu diretório *home* será `/home/joel`. O diretório *home* também é identificado por um `~` (til), você pode digitar tanto o comando `ls /home/joel` como `ls ~` para listar os arquivos de seu diretório *home*;





## Diretório Superior e Anterior

- 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.;
- O diretório anterior é identificado por `-`. É útil para retornar ao último diretório usado;
- Se estiver no diretório `/usr/local` e digitar `cd /lib`, você acessará o diretório `/lib`. E poderá retornar facilmente para o diretório `/usr/local` usando `cd -`.



## Nomeando Arquivos e Diretórios

- No GNU/Linux, os arquivos e diretórios podem ter o tamanho máximo de 255 letras;
- Você pode identificá-lo com 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.



## Alguns Comandos

- **clear** - Limpa a tela;
- **cd** - Mudança de diretório;
- **ls** - Listagem de conteúdo de diretório;
- **pwd** - Mostra em qual diretório você está;
- **chmod** - Mudança de permissão de acesso;
- **date** - Permite ver a data e hora do sistema;
- **df** - Mostra o espaço livre/ocupado em uma partição;
- **du** - Mostra o espaço ocupado por arquivos do diretório e dos arquivos dos subdiretórios do diretório atual.



## Alguns Comandos

- **find** – Procura arquivos/diretórios no disco;
- **free** – Mostra detalhes sobre a utilização da memória RAM do sistema;
- **grep** – Procura por um texto dentro de um arquivo;
- **head** – Mostra as linhas iniciais de um arquivo de texto;
- **nl** – Mostra o número de linhas e o conteúdo de um arquivo;
- **touch** – Muda a data e hora que um arquivo foi criado. Também pode ser usado para criação de arquivos vazios;
- **uptime** – Mostra o tempo de execução do sistema, desde o momento que o computador foi ligado;
- **uname** – Retorna o nome da versão do kernel atual;
- **reboot** – Reinicia o computador;
- **shutdown** – Desliga o computador.



# Permissões - Significado

d rwx rwx rwx

- A 1ª letra – Mostra se o arquivo é ou não um diretório;
- 2 a 4 letra – Permissões para o usuário dono do arquivo;
- 5 a 7 letra – Permissões para o grupo dono do arquivo;
- 8 a 10 letra – Permissões para outros usuários.



## Permissões - Valores

- Permissão se dá através de **dono**, **grupo** e **outros**:

d rwx rwx rwx

- Onde:  $R = 4$                        $W = 2$                        $x = 1$



## Exemplos

- --- .....0.....Sem permissão
- --x .....1.....Somente executa
- -w- ....2.....Somente grava
- -wx ....3.....Grava e executa
- r-- .....4.....Somente leitura
- r-x .....5.....Lê e executa
- rw- ....6.....Lê e grava
- rwx ....7.....Todas as permissões (Lê, grava e executa)



## Outros Exemplos

- **Veja abaixo uma lista com as permissões mais utilizadas:**
- `----- ...000...` Ninguém tem permissões;
- `r----- ...400...` Dono Lê;
- `r--r--r-- ...444...` Todos Lêem;
- `rw----- ...600...` Dono lê e grava;
- `rw-r--r-- ...644...` Dono lê e grava e os outros lêem;
- `rw-rw-r-- ...664...` Dono e o grupo lêem e gravam e os outros lêem;
- `rwX----- ...700...` Somente o dono tem todas as permissões;
- `rwXr-x--- ...750...` Dono com todas as permissões, o grupo lê e executa;
- `rwXr-xr-x ...755...` Dono com todas as permissões, o outros lêem e executam;
- `rwXrwxrwx ...777...` Todos com permissões liberadas.





## Símbolos

- $u \Rightarrow$  usuário
- $g \Rightarrow$  grupo
- $o \Rightarrow$  outros
- $a \Rightarrow$  todos
  
- $r \Rightarrow$  leitura
- $w \Rightarrow$  gravação
- $x \Rightarrow$  execução.



## Adicionar e Remover Permissão

- Para poder combinar os símbolos destas duas listas anteriores, usamos os operadores:
- + (adição) => adicionar permissão
- - (subtração) => remover permissão
- = (igualdade) => definir permissão
- Exemplo:  
`chmod u+w nome_do_arquivo.txt`

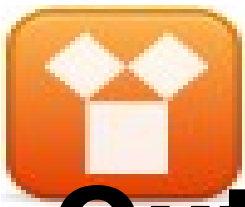


## Entendendo o Comando

• **chmod u+w nome\_do\_arquivo.txt**

- O “u” indica que a permissão será dada a um usuário;
- O sinal de adição (+) indica que está sendo adicionada a permissão;
- O “w” indica que a permissão que está sendo adicionada é a de

gravação.



# Outro Exemplo de 'chmod'

- Caso você queira dar permissão de leitura e execução ao seu grupo, o comando será:
- Exemplo:  
**chmod g+rx  
nome\_do\_arquivo.txt**



## Mais um Exemplo de 'chmod'

- Caso você queira dar permissão de leitura, gravação e execução ao seu grupo e ao dono do arquivo, o comando será:
- Exemplo:  
**chmod ug+rwx  
nome\_do\_arquivo.txt**



# Último Exemplo de 'chmod'

- Você pode copiar permissões de uma categoria para outra.
- Vamos copiar todas as permissões do “dono” do arquivo para a categoria “outros”;

- Exemplo:

```
chmod o=u
```

```
nome_do_arquivo.txt
```



## Exercício

- **Calcule os valores correspondentes as permissões abaixo:**

a) --x --x --x

b) -wx --x --x

c) --x -wx --x

d) --x --x -wx

e) rwx --x --x

f) rwx rwx --x

g) rwx rwx rwx

h) -wx -wx -wx

i) r-x r-x r-x

j) r-- r-- r--

l) rw- rw- rw-

m) r-x -wx -wx



## Exercício

**Descubra as letras correspondentes às permissões abaixo, preenchendo com '-' quando necessário:**

- a) 700
- b) 600
- c) 400
- d) 721
- e) 741
- f) 711
- g) 444
- h) 421
- i) 411
- j) 222
- l) 224
- m) 211
- n) 777





## Bibliografia

- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**, 3ª Ed., Rio de Janeiro: LTC Editora, 2002.
- MARTINS, Thiago M. *Slides da disciplina de Sistemas Operacionais de Arquitetura Aberta*. Faculdade Pitágoras, 2009.