

Curso de Engenharia de Computação ***Sistemas Operacionais***

INSTITUTO MAUÁ DE TECNOLOGIA



O sistema de arquivos do Linux (UNIX)



O sistema de arquivos do Linux

■ Partições do sistema

- O **sistema de arquivos** do **Linux** está **armazenado** em **um ou mais tipos de dispositivos** (disco rígido, cartão SD, *pen drive* etc) que internamente podem estar formatados de vários modos (ext4, ext3, FAT32, NTFS etc);
- Os **dispositivos** de armazenamento podem ser primários ou **secundários; mestres ou escravos**;
- **Comando para listar as partições do sistema**
- **fdisk** – manipula a tabela de partições do sistema:

```
$ sudo fdisk -l
```

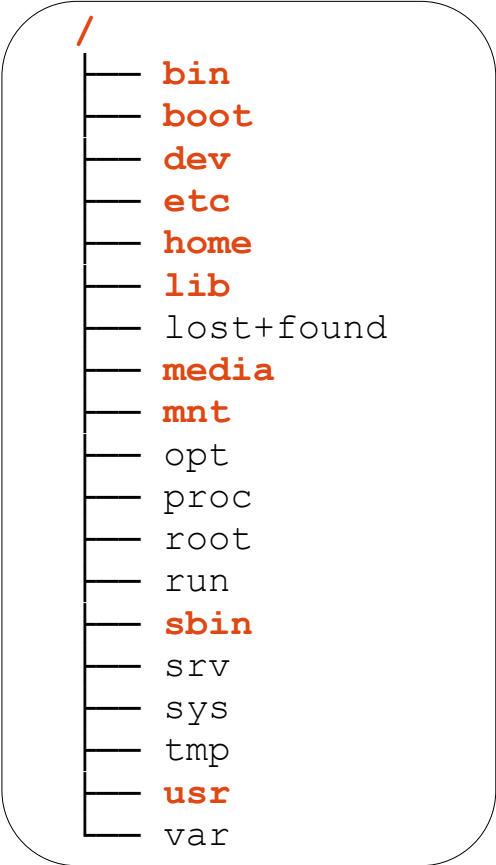
O sistema de arquivos do Linux

■ Montagem de dispositivos de armazenamento externo

- Ao se conectar um *pen drive*, por exemplo, o sistema pode (ou não) realizar uma **montagem automática** do dispositivo;
- Se for montado automaticamente, sua árvore de diretórios será anexada como um **subdiretório** de **/media**;
- Se não montado automaticamente, o dispositivo deve ser montado com o comando **mount**;
- Depois do uso, para **desmontar** um dispositivo utiliza-se o comando **umount**;
- Tanto **mount** quanto **umount** devem ser executados como superusuário (mas é possível alterar este comportamento);

O sistema de arquivos do Linux

■ Conhecendo o sistema de arquivos



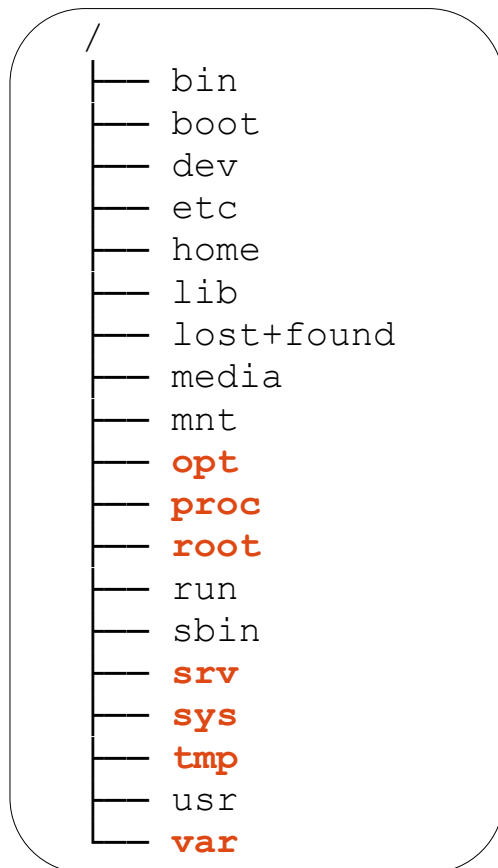
```
/
├── bin
├── boot
├── dev
├── etc
├── home
├── lib
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── srv
├── sys
├── tmp
├── usr
└── var
```

Diretórios importantes e comuns a todo Linux

- **/**: é a raiz do sistema de arquivos;
- **/bin**: arquivos de programas essenciais ao sistema;
- **/sbin**: arquivos de programas para o administrador;
- **/boot**: arquivos de sistema que inicializam o Linux;
- **/dev**: arquivos de dispositivo de hardware;
- **/etc**: arquivos de configurações de programas;
- **/home**: diretórios e arquivos dos usuários;
- **/lib**: códigos de bibliotecas e módulos do sistema;
- **/media**: arquivos de dispositivos externos (ex.: *pen drive*);
- **/mnt**: arquivos de dispositivos externos (não habituais);
- **/usr**: arquivos acessíveis pelo usuário – programas que podem ser executados pelo usuário. É o maior diretório do sistema.

O sistema de arquivos do Linux

■ Conhecendo o sistema de arquivos



Diretórios importantes e comuns a todo Linux

- **/opt**: pode armazenar diversos aplicativos, gerenciadores de bancos de dados, etc;
- **/proc**: é um diretório especial – é uma interface para a memória do computador, utilizada para ajustar o seu desempenho e também obter informações do seu estado atual;
- **/root**: é o diretório do superusuário **root**;
- **/srv**: utilizado para armazenar configurações de serviços (WEB, por exemplo);
- **/sys**: armazena informação sobre o estado do sistema;
- **/tmp**: armazenam-se aqui arquivos e diretórios temporários e descartáveis – limpo a cada reinicialização;
- **/var**: armazena dados que se alteram com frequência – *caches*, e-mail, *logs*, WEB, etc.

O sistema de arquivos do Linux

■ Trabalhando com o sistema de arquivos

– Trabalhando com **diretórios**

- **pwd** – exibe o diretório atual

```
$ pwd
```

- **ls** – lista o conteúdo de um diretório

```
$ ls -Al
```

```
$ ls -Al /usr/bin
```

- **cd** – mudar para outro diretório

```
$ cd /usr/bin
```

Abreviações importantes de nomes de diretórios

- ~ (til) – diretório *home* do usuário;
- . (ponto) – diretório atual;
- .. (ponto ponto) – diretório pai.

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Trabalhando com **diretórios**

- **mkdir** – criar um diretório

```
$ mkdir testes
```

- Criar um diretório com subdiretórios

```
$ mkdir -p testes/a/p
```

- **rmdir** – remove um diretório vazio

```
$ rmdir testes
```

- Para remover um diretório não vazio (CUIDADO!)

```
$ rm -rf testes
```

Nota: aqui, o diretório manipulado é **relativo** a aquele que se está no momento. Pode-se manipular um diretório fornecendo um **caminho absoluto**, também, por exemplo, **/home/marco/testes**

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**
 - **Tipos de arquivos do Linux**
 - **Arquivos regulares:** arquivos de dados ou de programa;
 - **Diretórios** (um diretório é um arquivo especial);
 - **Link simbólico:** um “apelido” para o arquivo;
 - **Arquivos de dispositivo orientado a blocos:** representar dispositivos de hardware de acesso aleatório;
 - **Arquivos de dispositivo orientado a caractere:** representar dispositivos de hardware seriais;
 - **Pipes:** arquivos especiais que sincronizam a saída de um programa com a entrada de outro;
 - **Sockets:** arquivos especiais que representam a conexão de rede entre duas máquinas.

O sistema de arquivos do Linux

■ Trabalhando com o sistema de arquivos

– Trabalhando com arquivos

- **ls** – Listar o conteúdo de um diretório

```
$ ls -Al /usr/bin
```

- Das opções possíveis, estas duas são muito úteis:
 - **A**: inclui arquivos ocultos;
 - **l**: exibe as propriedades dos arquivos.

Nota: arquivos cujo nome se inicia com “.” (ponto) são arquivos **ocultos** – listados apenas com a opção **-l** de **ls**.

O sistema de arquivos do Linux

- Trabalhando com o sistema de arquivos
 - Atributos listados

-	rw-r--r--	1	pi	pi	5781	Feb	3	2013	teste.txt
---	-----------	---	----	----	------	-----	---	------	-----------

tipo

- = regular
d = diretório
p = pipe
l = link
p = pipe
s = socket
c = disp. carac.
b = disp. bloco

permissões

número de *hardlinks*

para um diretório é o número de diretórios internos mais o diretório pai e ele próprio; para arquivo é o número de *hardlinks* existentes..

dono do arquivo

grupo do arquivo

data da última modificação

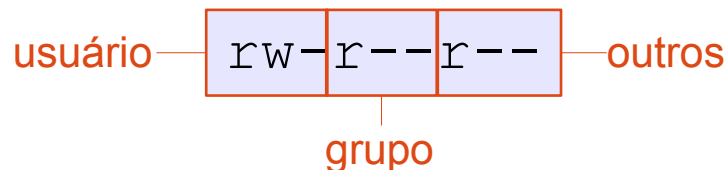
tamanho em bytes
modificado pela opção -h

nome do arquivo

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Sobre as **permissões** de arquivos e diretórios



- **Partes**

- As permissões são especificadas em três partes: para o **usuário**, para o **grupo** e para **outros**.

- **Tipos de permissão**

- **r**: permite leitura;
 - **w**: permite escrita;
 - **x**: permite execução;
 - **-**: não permite.

- Diretórios sempre tem a permissão de execução **x** – senão não é possível acessar seus conteúdos.

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Trabalhando com **arquivos**

- Utilização de “coringas” para selecionar mais de um arquivo

- *: substitui zero ou mais caracteres

- Exemplo

- Listar os arquivos que possuem **maua** em alguma parte do nome:

```
$ ls *maua*
```

- []: aceita qualquer um dos caracteres presentes na lista

- Exemplo

- Listar os arquivos que possuem o nome **Maua** ou **maua** em alguma parte do nome:

```
$ ls *[Mm]aua*
```

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Trabalhando com **arquivos**

- Nomes de diretórios ou arquivos com **espaços** em **branco**

- Para o *shell*, um espaço em branco representa a separação de argumentos em sua linha de comando;
 - Quando se manipula diretórios ou arquivos que contêm espaços em branco no nome, pode-se proceder de duas formas:
 - Proteger o nome com aspas duplas:

```
$ mkdir "Diretório com espaços"
```

- Utilizar “barra-espaço” para indicar que se trata de um espaço:

```
$ mkdir Diretório\ com\ espaços
```

O sistema de arquivos do Linux

▪ Trabalhando com o sistema de arquivos

– Trabalhando com arquivos

- **rm** – apagar arquivos ou diretórios

- Apagar arquivos específicos

```
$ rm teste.txt
```

- Apagar um diretório e todo conteúdo

```
$ rm -rf MeuDiretorio
```

- Pedir a confirmação do usuário antes de apagar

```
$ rm -i medo.txt
```

- Receber uma confirmação da operação

```
$ rm -iv medo.txt
```

O sistema de arquivos do Linux

▪ Trabalhando com o sistema de arquivos

– Trabalhando com arquivos

- **cp** – copia arquivos e diretórios

- Copiar um arquivo de **dir1** para **dir2**

```
$ cp dir1/teste.txt dir2
```

- Copiar um arquivo e alterar seu nome

```
$ cp teste.txt teste.txt.bak
```

- Copiar um arquivo, perguntando se deseja sobrescrever

```
$ cp -i teste.txt teste.txt.bak
```

- Copiar recursivamente o diretório **dir1** para **dir2**

```
$ cp -r dir1 dir2
```

O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Trabalhando com **arquivos**

- **mv** – mover arquivos e diretórios

- Mover um arquivo de **dir1** para **dir2**

```
$ mv dir1/teste.txt dir2
```

- Renomear um arquivo

```
$ mv teste1.txt teste2.txt
```

- Renomear um diretório

```
$ mv dir1 dir3
```


O sistema de arquivos do Linux

- **Trabalhando com o sistema de arquivos**

- Determinação do **espaço livre e ocupado**

- **df** – exibe o espaço livre (**-h** é mais compreensível)

```
$ df -h
```

- **du** – exibe o espaço ocupado pelo diretório atual

```
$ du -summarize -h
```

O sistema de arquivos do Linux

▪ Trabalhando com arquivos-texto

- **cat** – concatena e exibe o conteúdo de arquivos

```
$ cat /etc/hosts
```

- Se não for informado o nome de um arquivo, **cat** utiliza a entrada-padrão;
- Exemplos

- O menor editor de texto do Linux – termine com CTRL+D

```
$ cat > teste.txt
```

- Concatenando dois arquivos em um terceiro

```
$ cat a.txt b.txt > c.txt
```

- **touch** – altera o *timestamp* de um arquivo ou cria um arquivo vazio.

Exemplo

```
$ touch teste.txt
```

O sistema de arquivos do Linux

▪ Trabalhando com arquivos-texto

- **tail** – exibe as últimas linhas de um arquivo
 - Principal uso: monitorar em tempo real as modificações em um arquivo.

```
$ tail -f /var/log/messages
```

- **head** – exibe as primeiras linhas de um arquivo
 - Utilização similar ao **tail**.
- **less** e **more** – paginadores – permite “navegar” em um arquivo que não cabe na tela.

```
$ cat teste.txt | less
```

```
$ cat teste.txt | more
```

O sistema de arquivos do Linux

▪ Trabalhando com arquivos-texto

- **sort** – ordena linhas de um arquivo.

- Exemplo

```
$ sort teste.txt > ordenado.txt
```

- **diff** – exibe as diferenças entre dois arquivos.

- Exemplo

```
Um pequeno texto.  
Original.  
a.txt
```

```
Um pequeno texto.  
Com alteração.  
Original? b.txt
```

```
$ diff a.txt b.txt
```

O sistema de arquivos do Linux

■ Trabalhando com arquivos-texto

– Pesquisa com **expressões regulares**

- **grep** – é um programa que pesquisa expressões regulares em arquivos fornecidos na linha de comando ou na entrada padrão, caso contrário;
- Uma expressão regular é uma sequência de caracteres que representa um padrão a ser procurado.
- Exemplos

- Pesquisar em todos arquivos C a ocorrência da função 'func':

```
$ grep 'func' *.c
```

- Pesquisar em 'listagem.txt' todos os nomes de e-mails válidos:

```
$ grep -o -E "\b[a-zA-Z0-9.-]+@[a-zA-Z0-9.-]+\.[a-zA-Z0-9.-]+\b" listagem.txt
```

■ Gerenciando permissões

- Permissões garantem um **acesso seguro** aos **objetos** do **sistema** de arquivo;
- **Propriedade**
 - Todo arquivo e diretório do Linux possui um dono;
 - As informações de propriedade podem ser observadas com o comando **ls -l**, como já foi apresentado anteriormente;
 - Para procurar arquivos no sistema cujo proprietário é um certo usuário, utiliza-se o comando **find / -user nome**, como já foi apresentado anteriormente.
 - Para procurar arquivos no sistema cujo proprietário é um certo grupo, utiliza-se o comando **find / -group grupo**.

■ Gerenciando permissões

- **chown** – altera a propriedade de um arquivo ou diretório para um usuário. Com a opção **-R**, altera recursivamente a propriedade de um diretório e de seus conteúdos.

- Exemplos

```
$ sudo chown marco /tmp/teste.txt
```

```
$ sudo chown -R marco /tmp/dirs
```

- **chgrp** – altera a propriedade de um arquivo ou diretório para um grupo. Com a opção **-R**, altera recursivamente a propriedade de um diretório e de seus conteúdos. Similar ao anterior.

■ Gerenciando permissões

- Como já foi explicado, as **permissões** referem-se às operações de **leitura**, **escrita** e **execução**, atribuídas a **três categorias** que são o **dono**, os **grupos** do dono e os **demais usuários** do sistema;
- As permissões de cada categoria são armazenadas como **números inteiros**, cujo valor depende do valor de bits (0 – desligado, 1- marcado) em seu valor:
 - Bit 2: leitura – se marcado é o valor 4;
 - Bit 1: escrita – se marcado é o valor 2;
 - Bit 0: execução – se marcado é o valor 1.
- Exemplo
 - 777 – dono, grupo e outros podem ler, escrever e executar.

■ Gerenciando permissões

- **chmod** – permite alterar as permissões de um arquivo ou diretório.

Exemplos:

- O arquivo `teste.txt` para somente pode ser lido ou escrito pelo seu dono:

```
$ sudo chmod 600 teste.txt
```

- Alterar o acesso do arquivo `teste.txt` para que apenas o dono e o grupo possam ler e gravar nele:

```
$ chmod u+rw,g+rw,o-rw teste.txt
```

- **u**, **g**, **o** representam o usuário, grupo e outros. As permissões são **r**, **w** e **x**, respectivamente leitura, escrita e execução. **+** permite e **-** inibe a permissão.