

Linux: Comandos Básicos

Sistemas Operacionais

Luiz Fernando T. de Farias

Conteúdo

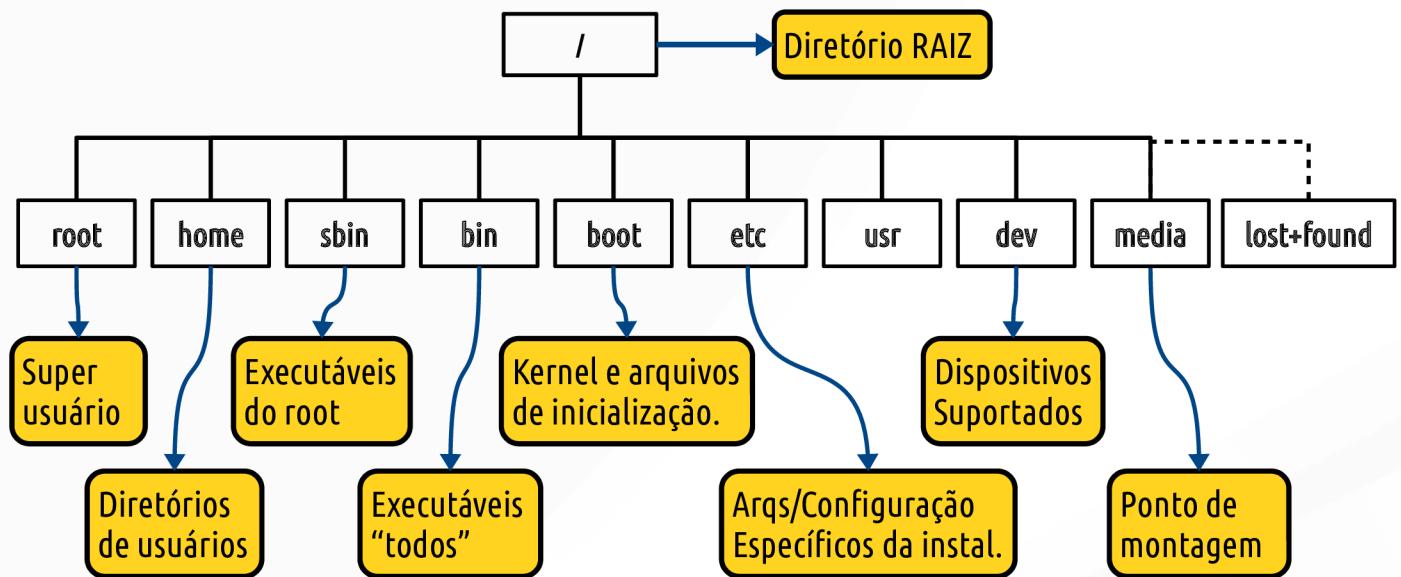
- ✓ Estrutura de Diretórios
- ✓ Manipulação de Diretório
- ✓ Arquivos e Diretórios
- ✓ Nó-Índice
- ✓ Trabalhando com Arquivos
- ✓ Usando Metacaracteres
- ✓ Copiando, movendo e removendo
- ✓ Ligações entre arquivos

Estrutura de Diretórios

- ◆ Arquivos e diretório são organizados hierarquicamente em forma de uma árvore cuja função é dividir o espaço de armazenamento, visando a organização.
- ◆ Não são atribuídas letras para identificar unidades de armazenamento. Todos os arquivos e diretórios estão dispostos abaixo do diretório “/”.
- ◆ **FHS (*Filesystem Hierarchy Standard*)** é o padrão para sistema de arquivos hierárquico. Define a organização dos diretórios em sistemas compatíveis com o Unix.

Estrutura de Diretórios

Estrutura de diretórios



Slide Vazio

Manipulação de Diretório

pwd

Mostra o nome do diretório atual.

```
[~] $ pwd  
/home/luftf
```

Diretórios:
/ → RAIZ
~ → “do usuário”
.. → acima (“pai”)

Manipulação de Diretório

cd

Altera o diretório de trabalho.

```
[~] $ cd /home  
[~] $ pwd  
/home  
[~] $ cd /home/luftf
```

Experimente:

```
$ cd /  
$ cd ~  
$ cd -
```

Caminhos ABSOLUTOS

- ✓ São aqueles que começam na raiz (/)
- ✓ Define o caminho a partir do topo da árvore de diretórios.

Exemplo: /var/spool/mail

Caminhos RELATIVOS

- ✓ Começa a partir do diretório atual.
- ✓ Pode ser referenciado com o diretório corrente (.) ou
- ✓ como o diretório hierarquicamente acima, o diretório pai (..)

Exemplos: home/luftf/arquivo.tar, ./arquivo.tar ou
..../home/alunolista.txt

Arquivos e Diretórios

ls [-Ralh...][diretório]

Descrição: Lista o conteúdo dos diretórios.

{
-R Recursivo (subdiretórios)
-a Lista arquivos ocultos
-l Formato longo
-h Formato humano

```
[~] $ ls  
[~] $ ls -a  
[~] $ ls -R /home  
[~] $ ls -l /boot  
[~] $ ls -lh /boot
```

...

```
rw-r--r-- 1 root root 1,8M Abr 2 16:28 /boot/vmlinuz-2.6.18-194.el5
```

→ *Tipo do arquivo* {
- Arquivo Regular
d Arquivo do tipo diretório
l Ligação
c/b Especial: Caractere ou bloco

Arquivos e Diretórios

ls [-Ral...][diretório]

Descrição: Lista o conteúdo dos diretórios.

{
-R Recursivo (subdiretórios)
-a Lista arquivos ocultos
-l Formato longo
-h Formato humano

```
[~] $ ls  
[~] $ ls -a  
[~] $ ls -R /home  
[~] $ ls -l /boot  
[~] $ ls -lh /boot
```

-rw-r--r--	1	root	root	1,8M	Abr 2 16:28	/boot/vmlinuz-2.6.18-194.el5
------------	---	------	------	------	-------------	------------------------------

Segurança
Num. ligações
Dono | Grupo
Tamanho
Hora/Data modificação
Nome do Arquivo

Tipo do arquivo

mkdir [-opções] diretório

{
-p Recursivo (cria subdiretórios)

Descrição: Cria diretórios

```
[~] $ cd ~  
[~] $ mkdir dir1  
[~] $ ls  
dir1  
[~] $ mkdir dir2 dir3  
[~] $ ls  
dir1 dir2 dir3
```

Experimente:

```
$ mkdir /home/dir4/teste1  
$ mkdir -p /home/dir4/teste1  
$ mkdir -p /home/dir5/teste2
```

Crie o diretório luiz fernando

Arquivos e Diretórios

rmkdir [-opções] diretório

{ -p Recursivo (cria subdiretórios)

Descrição: Remove diretórios

```
[~] $ cd ~  
[~] $ rmkdir luiz*  
[~] $ rmkdir dir1  
[~] $ ls  
dir2 dir3 dir4 dir5  
[~] $ rmkdir dir2 dir3 dir5
```

Remova o diretório dir4 usando a opção recursiva.

Slide Vazio

Nó índice

I-NODE

- Estrutura de dados usada nos sistemas de arquivos.
- Descreve e armazena informações sobre os arquivos.

Armazena as seguintes informações:

- ✓ Tamanho do arquivo em bytes.
 - ✓ Endereço físico do arquivo.
 - ✓ Dono e Grupo: UID e GID
 - ✓ Registro de horário:
 - ✓ Tipo de arquivo:
 - ✓ Permissões de acesso:
 - ✓ Contador de referências.
-
- The diagram illustrates the structure of an I-node by mapping its various fields to their corresponding meanings. Arrows point from each field to a bracketed list of details:
- Registro de horário: atime: *access time* (último acesso); ctime: *change* (última alteração); mtime: *modification* (última alteração).
 - Tipo de arquivo: Regular (comum), Diretório, Bloco/Caractere, FIFO ou soquete.
 - Permissões de acesso: Quem pode ler, Gravar ou executar.

Nó índice

stat

```
[~] $ ls -l arquivo
-rw-rw-r-- 1 luftf luftf 0 Ago 18 14:50 arquivo
[~] $ ls -i arquivo
327787
[~] $ stat arquivo
  File: `arquivo'
  Size: 0          Blocks: 0      IO Block: 4096 arquivo comum vazio
Device: fd00h/64768d  Inode: 327787  Links: 1
Access: (0664/-rw-rw-r--)  UID: (500/luftf)  Gid: (500/luftf)
Access: 2010-08-18 15:07:47.0000000 -0300
Modify: 2010-08-18 14:50:47.0000000 -0300
Change: 2010-08-18 14:50:47.0000000 -0300
```

Slide Vazio

Trabalhando com Arquivos

touch [-op] arquivo

{
-a altera hora: último acesso
-m altera hora: última modificação

Descrição: Altera data e hora no arquivo

```
[~] $ ls -l arquivo
.... Ago 18 14:02 arquivo
[~] $ touch -m arquivo
.... Ago 18 15:07 arquivo
[~] $ touch -a arquivo
.... Ago 18 15:07 arquivo
```

Não houve modificações porque o ls não mostra detalhes de acesso.
É necessário usar o comando stat.

Trabalhando com Arquivos

touch [-op] arquivo

$\begin{cases} -a & \text{altera hora: último acesso} \\ -m & \text{altera hora: última modificação} \end{cases}$

Descrição: Altera data e hora no arquivo

```
[~] $ ls -l arquivo  
.... Ago 18 14:02 arquivo  
[~] $ touch -m arquivo  
.... Ago 18 15:07 arquivo  
[~] $ touch -a arquivo  
.... Ago 18 15:07 arquivo  
[~] $ touch -t 09101130 arquivo  
.... Sep 10 11:30 arquivo
```

Não houve modificações porque o ls não mostra detalhes de acesso.
É necessário usar o comando stat.

Slide Vazio

Usando Metacaracteres

Metacaracteres

São caracteres que possuem significado especial para o shell.

Há dois tipos de metacaracteres:

- 1) “curingas (wildcards)” → Especifica arquivos por nome ou extensão
- 2) “editores de expressões regulares” → Especifica strings de texto em arquivos.

CURINCA	Correspondência
?	Um único caractere.
*	Todos os caracteres.
[]	Qualquer um dos caracteres entre colchetes
[a-z]	Uma faixa de caracteres.

Usando Metacaracteres

→ Utilize o comando TOUCH para criar os arquivos:

arquivo, arquivo1, arquivo2, arquivo3, arquivo34, arquivo4, arquivo3arq,
arquivo5arq e arquivo6arq.

```
[~] $ ls arquivo*  
[~] $ ls arquivo?  
[~] $ ls arquivo??  
[~] $ ls arquivo[124]  
[~] $ ls arquivo[1-3]  
[~] $ ls arquivo{1,34}  
[~] $ ls arquivo[1,3-5]*
```

Para cada linha digitada verifique os resultados encontrados.

Slide Vazio

Copiando, movendo e removendo

cp [-op] <origem> <destino>

Descrição: Copia Arquivos da Origem para o Destino.

```
[~] $ mkdir dir1  
[~] $ touch arq1  
[~] $ cp arq1 dir1  
[~] $ ls dir1  
arq1  
[~] $ cp -b arq1 dir1  
[~] $ ls dir1  
arq1 arq1~
```

~ → extensão usada para indicar o arquivo antigo, que foi mantido como backup.

Copiando, movendo e removendo

mv [-op] <origem> <destino>

Descrição: Move Arquivos da Origem para o Destino.

```
[~] $ mkdir testemv  
[~] $ touch arq1 arq2 arq5 arq52 arq55  
[~] $ ls  
arq2 arq5 arq52 arq55 arq57  
[~] $ mv arq2 arq57  
[~] $ ls  
arq5 arq52 arq55 arq57 arq57~
```

rm [-op] <arquivo>

Descrição: Remove arquivos.

```
[~] $ cd ~  
[~] $ touch arq1 arq2 arq3 arq31  
[~] $ rm -i arq1  
remove o arquivo comum vazio?  
[~] $ mkdir -p dir1/dir2/dir3  
[~] $ rm dir1  
rm: impossível remover: É um diretório  
[~] $ rm -r dir1
```

Ligações entre arquivos

Ligações entre arquivos

Links são referências, atalhos ou conexões lógicas entre arquivos e diretórios.

HARD LINKS

Ocupam apenas um i-node na tabela.

- Só pode ser usado quando estão no mesmo sistema de arquivos
- Quando um *hard link* é alterado o original também é modificado;
- Se o arquivo original for apagado, os *hard links* continuam.

SOFT LINKS

- É um atalho para um arquivo (similar ao do Windows).
- Podem ser usados em sistemas de arquivos diferentes (ocupam i-node diferentes.)
- Apagar o um link simbólico apaga APENAS o vínculo.
- Se o arquivo o arquivo original for apagado, os links ficam “quebrados”

Arquivos e Diretórios

ln [-op] <arquivo> <link>

{ -s Cria um link simbólico

Descrição: Cria ligação (link) para um arquivo.

```
[~] $ touch arq1  
[~] $ ls -i arq1  
327787 arq1  
  
[~] $ ls -l arq1  
-rw-rw-r- 1 luftf luftf 0 Ago 24 16:48 arq1  
  
[~] $ ln arq1 arq2  
[~] $ ls -i arq2  
327787 arq2  
  
[~] $ ls -l arq2  
-rw-rw-r- 2 luftf luftf 0 Ago 24 16:48 arq1
```

Observe:
A quantidade de links mudou!

```
[~] $ ln -s arq1 arq3  
[~] $ ls -i arq3  
327750 arq1  
  
[~] $ ls -l arq3  
-rw-rw-r- 1 luftf luftf 4 Ago 24 16:48 arq3  
[~] $ file arq2 arq3  
arq2: empty  
arq3: symbolic link to `arq1`
```

```
[~] $ rm arq1  
[~] $ file arq2 arq3  
arq2: empty  
arq3: broken symbolic link to `arq1`
```