

Comandos de Criação e Gestão de Directórios e Arquivos

Cadeira: Administração de Sistemas Linux

Introdução

- ✓ Um diretório é um local de memória onde armazenamos arquivos para melhor organização e localização e um arquivo é onde armazenamos nossos dados, seu conteúdo pode ser um texto, uma música, programa, planilha, etc, que são os arquivos ou ficheiros.
- ✓ Cada diretório ou arquivo deve ser identificado por um nome, para facilmente ser encontrado quando se deseja usá-lo. E os nomes são “case sensitive” (diretório *teste* é completamente diferente do diretório *Teste*).
- ✓ Não podem existir num diretório dois arquivos com o mesmo, ou um sub-diretório com um mesmo nome de um arquivo num mesmo diretório.

Introdução

Extensão de Arquivos

- ✓ A extensão serve para identificar o tipo do arquivo. A extensão são um conjunto de letras após um "." no nome de um arquivo, explicando melhor:
 - relatório.txt - O **txt** é a extensão e 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.
 - index.html - Página de Internet (formato Hypertexto).
- ✓ A extensão de um arquivo ajuda a identificar o programa que precisamos “rodar” para executar ou ver o seu conteúdo.

Introdução

Caminhos Absolutos e Relativos

- ✓ Para navegar entre directórios numa interface de linha de comando é necessário entender dois tipos de caminhos: os caminhos absolutos e relativos.
- ✓ Caminhos absolutos são a localização de directórios e arquivos a partir da raiz do sistema “/” (barra, root). Por exemplo: */home/professor/disciplina/pdfs*
- ✓ Caminhos relativos dependem do directório corrente. Por exemplo: Assumindo a cadeia de directórios (*/home/professor/disciplina/pdfs*) e assumindo que o utilizador esteja no directório */home/professor*, para ele chegar ao directório */pdfs* basta se mover para a seguinte sequência de caminhos: *disciplina/pdfs*. Este caminho foi determinado em relação a posição do utilizador.

Introdução

Caminhos Absolutos e Relativos

- ✓ Comparado ao caminho absoluto, o caminho relativo é bem mais compacto; contudo, depende necessariamente do ponto de partida do usuário.
- ✓ Já o caminho absoluto, pode ser utilizado independente de onde o usuário esteja, pois sua referência é global, enquanto que o caminho relativo a sua referência é local.

1. CRIAÇÃO E GESTÃO DE DIRECTÓRIOS

1.1 Criação de Directórios

COMANDO mkdir (make directory)

✓ O comando mkdir é utilizado para criar um directório no sistema. Um directório é como uma pasta onde guardamos arquivos e subpastas.

✓ Sintaxe:

```
mkdir [opções] [caminho1/diretorio1] [caminho2/diretorio2] ...
```

✓ Exemplos:

- \$ mkdir /tmp/pasta1 - Cria as directório “pasta1” dentro do directório /tmp
- # mkdir -p asl/alunos - A opção -p permite criar o directório “asl” e o subdirectório “alunos”, caso não existam.

1.2 Explorando Diretórios

COMANDO pwd (print working directory)

- ✓ Este comando mostra o nome e caminho do diretório actual. Um comando bastante útil quando navegamos pelo sistema e não lembramos em que diretório estamos.

```
sousa@ubuntuVM:~$ pwd  
/home/sousa
```


1.2 Explorando Directórios

COMANDO cd (change directory)

- ✓ Muda do directório corrente para o directório especificado como argumento.
- ✓ Sintaxe:

cd directório

✓ Exemplos

- \$ cd / – entra no directório root
- \$ cd /tmp – entra no directório /tmp

✓ Atalhos importantes

- \$ cd ~ → vai para o directório home do usuário logado
- \$ cd .. → sobe para um directório de nível superior
- \$ cd – → retorna ao directório anterior

1.2 Explorando Directórios

COMANDO ls (list)

- ✓ Lista conteúdo de um directório.
- ✓ Sintaxe:

`ls [opções] [directorio1] [directorio2] ...`

- ✓ Exemplos:
 - `$ ls` - Lista o conteúdo do directório actual;
 - `$ ls /` - Lista os directórios e arquivos do directorio raiz;
 - `$ ls /etc` - Lista os directórios e arquivos do /etc;
 - `$ ls / /etc` - Listar o conteúdo do / e do /etc, num único comando.

1.2 Explorando Directórios

✓ opções

- a, --all - Lista todos os arquivos (inclusive os ocultos) de um directório.
- d, --directory - Lista os nomes dos directórios ao invés do conteúdo.
- h, --human-readable - Mostra o tamanho dos arquivos em Kbytes, Mbytes, Gbytes.
- H - Faz o mesmo que -h, mas usa unidades de 1000 ao invés de 1024 para especificar Kbytes, Mbytes, Gbytes.
- l - Usa o formato longo para listagem de arquivos. Lista as permissões, data de modificação, donos, grupos, etc.
- R - Lista directórios e sub-directórios recursivamente.

1.2 Explorando Directórios

✓ Outros exemplos:

```
$ ls -l /
```

```
drwxr-xr-x 4 root root 1024 2007-01-15 23:17 boot
```

```
#ls -a /root
```

```
. .. .bash_history      .bashrc .profile
```

1.3 Remoção de Diretórios

COMANDO rmdir (remove directory)

- ✓ O comando rmdir é utilizado para remover diretórios vazios.
- ✓ Sintaxe:

\$ rmdir [opções] diretorio1 [diretorio2]...

- ✓ Algumas opções do comando:
 - p, --parents - remove uma hierarquia de diretórios.
 - v, --verbose: -exibe informações para cada diretório processado.
 - help - informa as opções do aplicativo.
- ✓ Exemplos:
 - \$ rmdir /tmp/pasta1 - Remove os directórios “pasta1”.
 - # rmdir -p asl/alunos - Remove o diretório “asl” e o subdiretório “alunos”.

2. CRIAÇÃO E GESTÃO DE ARQUIVOS

2.1 Criação de Arquivos

COMANDO touch

- ✓ Este comando muda a data e a hora de criação de arquivos existentes. Caso o *touch* seja usado com arquivos que não existam, por padrão ele cria estes arquivos.

- ✓ Sintaxe:

```
touch  [opções] caminho1/arquivo1  
       [caminho2/arquivo2] ...
```

- ✓ Opções:

-a, --time=atime - Faz o touch mudar somente a data e hora do acesso ao arquivo.

-t MMDDhhmm[ANO.segundos] - Usa Mês (MM), Dias (DD), Horas (hh), minutos (mm) e opcionalmente o ANO e segundos para modificação do(s) arquivos ao invés da data e hora atual.

2.1 Criação de Arquivos

- c, --no-create - Não cria arquivos vazios, caso os arquivos não existam.
- m, --time=mtime - Faz o touch mudar somente a data e hora da modificação.
- r [arquivo] - Usa as horas no [arquivo] como referência ao invés da hora actual,

✓ Exemplos:

- touch teste - Cria o arquivo teste no actual directório, caso este não exista.
- touch -t 10011230 teste - Altera da data e hora do arquivo para 01/10 e 12:30.

2.2 Visualização de Conteúdo

COMANDO cat

- ✓ Mostra o conteúdo de um ou mais arquivo binário ou texto.
- ✓ Sintaxe:

`cat [opções] caminho1/arquivo1] [caminho2/arquivo2]`

- ✓ Opções:
 - n, --number - Mostra o número das linhas enquanto o conteúdo do arquivo é mostrado.
 - b - Mostra o número das linhas que não são linhas em branco enquanto o conteúdo do arquivo é mostrado.

2.2 Visualização de Conteúdo

✓ Exemplos:

- `cat /etc/passwd` - Exibe o conteúdo do arquivo `passwd`.
- `cat -n /etc/passwd` - Exibe o conteúdo do arquivo `passwd` numerando suas linhas.
- `cat -n arquivo1.txt arquivo2.txt` - Exibe o conteúdo dos dois arquivos um após o outro concatenando-os e numerando suas linhas.

2.2 Visualização de Conteúdo

COMANDO tac

- ✓ Mostra o conteúdo de um arquivo binário ou texto (como o cat) só que em ordem inversa.
- ✓ Sintaxe:

`tac [opções] diretório1/arquivo1 [diretório2/arquivo2]...`

- ✓ Exemplo:
 - `tac /etc/passwd` - Exibe o conteúdo do arquivo passwd na ordem inversa, ou seja de traz para frente.
 - `tac arquivo1.txt arquivo2.txt` - Exibe o conteúdo dos arquivos um após o outro de traz para frente concatenando-os.

2.3 Manipulação de Arquivos

COMANDO cp(copy)

- ✓ Serve para copiar arquivos e directórios
- ✓ Sintaxe:

`cp [opções] origem destino`

- ✓ O comando `cp` copia arquivos da ORIGEM para o DESTINO. Ambos origem e destino terão o mesmo conteúdo após a cópia.
- ✓ 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 directório `/tmp`.
 - `cp * /tmp` - Copia todos os arquivos do directório atual para `/tmp`.

2.3 Manipulação de Arquivos

✓ opções

- i, --interactive - Pergunta antes de substituir um arquivo existente.
- f, --force - Não pergunta, substitui todos os arquivos caso já exista.
- r - Copia arquivos dos diretórios e subdiretórios da origem para o destino. É recomendável usar -R ao invés de -r.
- R, --recursive - Copia arquivos e sub-diretórios (como a opção -r) e também os arquivos especiais FIFO e dispositivos.
- u, --update - Copia somente se o arquivo de origem é mais recente que o arquivo de destino ou quando o arquivo de destino não existe.

2.3 Manipulação de Arquivos

- v, --verbose - Mostra o processo de cópia dos arquivos.
- p, --preserve - Preserva atributos do arquivo, se possível.
- a - Copia todos os atributos (preservando proteção, dono e datas)
- preserve=timestamps - Preserva, dos atributos, apenas aqueles relativos à data (veremos mais abaixo quando usar)

✓ Exemplos:

- `cp -R /bin /tmp` - Cópia o diretório /bin e todos os arquivos/sub-diretórios existentes para o diretório /tmp.

2.3 Manipulação de Arquivos

COMANDO mv (move)

- ✓ Move e/ou renomeia arquivos e diretórios.
- ✓ Sintaxe:

`mv [opções] origem destino`

- ✓ O comando `mv` copia um arquivo da *ORIGEM* para o *DESTINO* (semelhante ao `cp`), mas após a cópia, o arquivo de *ORIGEM* é apagado.
- ✓ Exemplos:
 - `mv teste.txt teste1.txt` - Muda o nome do arquivo `teste.txt` para `teste1.txt`.
 - `mv teste.txt /tmp` - Move o arquivo `teste.txt` para `/tmp`. Lembre-se que o arquivo de origem é apagado após ser movido.

2.3 Manipulação de Arquivos

- `mv teste.txt teste.new` - (supondo que teste.new já exista)
-Cópia o conteúdo do arquivo teste.txt para teste.new e apaga teste.txt após terminar a cópia.

✓ Opções:

- f, --force - Substitui o arquivo de destino sem perguntar.
- i, --interactive - Pergunta antes de substituir. É o padrão.
- v, --verbose - Mostra os arquivos que estão sendo movidos.
- u, --update - Move somente arquivos antigos, ou novos arquivos.

2.3 Manipulação de Arquivos

COMANDO rm (remove)

✓ Apaga arquivos. Também usado para apagar diretórios e sub-diretórios que contenham ou não arquivos.

✓ Sintaxe:

```
rm  [opções]      caminho1/arquivo1  
    [caminho2/arquivo2] ...
```

✓ *opções*

-i, --interactive - Pergunta antes de remover, esta é ativada por padrão.

-v, --verbose - Mostra os arquivos na medida que são removidos.

2.3 Manipulação de Arquivos

- r, --recursive - Usado para remover arquivos em sub-diretórios. Esta opção também pode ser usada para remover sub-diretórios.
- f, --force - Remove os arquivos sem perguntar.

✓ Exemplos:

- `rm teste.txt` - Apaga o arquivo teste.txt no diretório actual.
- `rm -rf /tmp/teste/*` - Apaga todos os arquivos e sub-diretórios do diretório /tmp/teste mas mantém o sub-diretório /tmp/teste.
- `rm -rf /tmp/teste` - Apaga todos os arquivos e sub-diretórios do diretório /tmp/teste, inclusive /tmp/teste.
- `rm -f -- --arquivo--` - Remove o arquivo de nome --arquivo--.

2.3 Manipulação de Arquivos

COMANDO find

- ✓ Usado para procurar por directórios e arquivos no disco.
- ✓ Exemplos:
 - `$ find / -name documents` - Neste exemplo o comando find procura por um arquivo ou directório com o nome “documents” a partir do / (directório root).
 - `$ find ~ -name music` - Este outro procura por um arquivo ou directório com o nome “music” a partir do directório home do usuário.