

Capítulo 3 - Comandos para manipulação de diretório

Abaixo comandos úteis para a manipulação de diretórios.

3.1 ls

Lista os arquivos de um diretório.

```
ls [opções] [caminho/arquivo] [caminho1/arquivo1] ...
```

onde:

caminho/arquivo

Diretório/arquivo que será listado.

caminho1/arquivo1

Outro Diretório/arquivo que será listado. Podem ser feitas várias listagens de uma só vez.

opções

-a, --all

Lista todos os arquivos (inclusive os ocultos) de um diretório.

-A, --almost-all

Lista todos os arquivos (inclusive os ocultos) de um diretório, exceto o diretório atual e o de nível anterior.

-B, --ignore-backups

Não lista arquivos que terminam com ~ (Backup).

--color=PARAM

Mostra os arquivos em cores diferentes, conforme o tipo de arquivo. PARAM pode ser:

- *never* - Nunca lista em cores (mesma coisa de não usar o parâmetro --color).

- *always* - Sempre lista em cores conforme o tipo de arquivo.
- *auto* - Somente colore a listagem se estiver em um terminal.

-d, --directory

Lista os nomes dos diretórios ao invés do conteúdo.

-f

Não classifica a listagem.

-F

Insere um caracter após arquivos executáveis ('*'), diretórios ('/'), soquete ('='), link simbólico ('@') e pipe ('|'). Seu uso é útil para identificar de forma fácil tipos de arquivos nas listagens de diretórios.

-G, --no-group

Oculto a coluna de grupo do arquivo.

-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.

-n

Usa a identificação de usuário e grupo numérica ao invés dos nomes.

-L, --dereference

Lista o arquivo original e não o link referente ao arquivo.

-o

Usa a listagem longa sem os donos dos arquivos (mesma coisa que -lG).

-p

Mesma coisa que -F, mas não inclui o símbolo '*' em arquivos executáveis. Esta opção é típica de sistemas Linux.

-R

Lista diretórios e sub-diretórios recursivamente.

Uma listagem feita com o comando `ls -la` normalmente é mostrada da seguinte maneira:

```
-rwxr-xr--  1 claiton user      8192 nov  4 16:00 teste
```

Abaixo as explicações de cada parte:

`-rwxr-xr--`

São as permissões de acesso ao arquivo teste. A primeira letra (da esquerda) identifica o tipo do arquivo, se tiver um `d` é um diretório, se tiver um `-` é um arquivo normal.

`1`

Se for um diretório, mostra a quantidade de sub-diretórios existentes dentro dele. Caso for um arquivo, será 1.

`claiton`

Nome do dono do arquivo teste.

`user`

Nome do grupo que o arquivo teste pertence.

`8192`

Tamanho do arquivo (em bytes).

`nov`

Mês da criação/ última modificação do arquivo.

`4`

Dia que o arquivo foi criado.

16:00

Hora em que o arquivo foi criado/modificado. Se o arquivo foi criado há mais de um ano, em seu lugar é mostrado o ano da criação do arquivo.

teste

Nome do arquivo.

Exemplos do uso do comando `ls`:

- `ls` - Lista os arquivos do diretório atual.
- `ls /bin /sbin` - Lista os arquivos do diretório `/bin` e `/sbin`
- `ls -la /bin` - Listagem completa (vertical) dos arquivos do diretório `/bin` inclusive os ocultos.

3.2 cd

Entra em um diretório. Você precisa ter a permissão de execução para entrar no diretório.

`cd [diretório]`

onde:

diretório - diretório que deseja entrar.

Exemplos:

- Usando `cd` sem parâmetros ou `cd ~`, você retornará ao seu diretório de usuário (diretório home).
- `cd /`, retornará ao diretório raiz.
- `cd -`, retornará ao diretório anteriormente acessado.
- `cd ..`, sobe um diretório.
- `cd ../[diretório]`, sobe um diretório e entra imediatamente no próximo (por exemplo, quando você está em `/usr/sbin`, você digita `cd ../bin`, o comando `cd` retorna um diretório (`/usr`) e entra imediatamente no diretório `bin` (`/usr/bin`).

3.3 pwd

Mostra o nome e caminho do diretório atual.

Você pode usar o comando `pwd` para verificar em qual diretório se encontra (caso seu aviso de comandos não mostre isso).

3.4 mkdir

Cria um diretório no sistema. Um diretório é usado para armazenar arquivos de um determinado tipo. O diretório pode ser entendido como uma *pasta* onde você guarda seus papéis (arquivos). Como uma pessoa organizada, você utilizará uma pasta para guardar cada tipo de documento, da mesma forma você pode criar um diretório *vendas* para guardar seus arquivos relacionados com vendas naquele local.

```
mkdir [opções] [caminho/diretório] [caminho1/diretório1]
```

onde:

caminho

Caminho onde o diretório será criado.

diretório

Nome do diretório que será criado.

opções:

`--verbose`

Mostra uma mensagem para cada diretório criado. As mensagens de erro serão mostradas mesmo que esta opção não seja usada.

Para criar um novo diretório, você deve ter permissão de gravação. Por exemplo, para criar um diretório em `/tmp` com o nome de `teste` que será usado para gravar arquivos de teste, você deve usar o comando `"mkdir /tmp/teste"`.

Podem ser criados mais de um diretório com um único comando (`mkdir /tmp/teste /tmp/teste1 /tmp/teste2`).

3.5 rmdir

Remove um diretório do sistema. Este comando faz exatamente o contrário do `mkdir`. O diretório a ser removido deve estar vazio e você deve ter permissão de gravação para remove-lo.

```
rmdir [caminho/diretório] [caminho1/diretório1]
```

onde:

caminho

Caminho do diretório que será removido.

diretório

Nome do diretório que será removido.

É necessário que esteja um nível acima do diretório(s) que será(ão) removido(s). Para remover diretórios que contenham arquivos, use o comando `rm` com a opção `-r`.

Por exemplo, para remover o diretório `/tmp/teste` você deve estar no diretório `tmp` e executar o comando `rmdir teste`.