



METCOMPLIC
F I S 0 1 0 6 9

Introdução ao terminal GNU/Linux

(uso do bash)

R. Pezzi

31 de agosto de 2020



Arquivo Editar Ver Pesquisar Terminal Ajuda

```
peessoa@meu_computador:~$ cat topicos.md
```

```
# Tópicos #
```

- * 0 terminal Bash
 - * Comandos
 - * Propriedades dos arquivos
 - * Caminhos
 - * Utilizando computadores remotos
 - * Operadores na linha de comando
 - * Manipulação de texto
- * Segurança
 - * Permissões
 - * Integridade de arquivos
 - * Dicas sobre senhas

```
peessoa@meu_computador:~$ █
```

O que é o terminal?

- O terminal é um programa do tipo interpretador de comandos. https://pt.wikipedia.org/wiki/Interpretador_de_comandos
 - Interpretadores de comandos são programas de computador responsáveis por tomar as ações efetivas conforme a orientação do usuário através de comunicação textual
- Bash é o interpretador de comandos mais utilizado no GNU/Linux.

<https://pt.wikipedia.org/wiki/Bash>

Elementos da linha de comando

Em inglês a arroba se lê “at”, que significa “em” na nossa língua vernácula

Tipo de terminal

- \$ → usuário
- # → root (administrador)

cursor

```
peSSoaa@meu_computador:~$ █
```

Nome do usuário

Nome do computador
(hostname)

Diretório de trabalho
(diretório atual)

Dica:
A tecla <TAB> completa comandos,
caminhos e nome de arquivos

- ~ é um atalho para o diretório do usuário.
- Neste exemplo /home/pessoa

man: manual dos comandos

- \$ man ls

```
NOME
    ls, dir, vdir - lista o conteúdo do diretório

SINOPSE
    ls [opções] [arquivo...]
    dir [arquivo...]
    vdir [arquivo...]

Opções POSIX: [-CFRacdilqrutl]
```

```
OPÇÕES POSIX
-C    Lista arquivos em colunas, ordenados verticalmente.
-F    Sufixa o nome de cada diretório com '/', cada nome de FIFO com '|', e cada nome de executável com '*'.
-R    Lista os diretórios encontrados, recursivamente.
-a    Inclue os arquivos com o nome iniciando com '.' na listagem.
-c    Usa o status do tempo de alteração ao invés do tempo de modificação para ordenar (com -t) ou listar (com -l).
-d    Lista nome de diretórios como arquivo, preferencialmente no lugar de seus conteúdos.
-i    Precede a saída para o arquivo pelo número serial do arquivo (número do i-node).
-l    Escreve (no formato de coluna simples) o modo do arquivo, o número de ligações para o arquivo, o nome do proprietário, o nome do grupo, o tamanho do arquivo (em bytes), o rótulo de tempo, e o nome do arquivo.
```

-a, --all

Lista todos os arquivos nos diretórios, incluindo todos os arquivos começados com '.'.

-h, --human-readable

Anexa a letra de tamanho, por exemplo M para binários de megabytes ('mebibytes'), para cada tamanho.

Estrutura de um comando

- Comando [opções] [argumentos]

```
pessoa@meu_computador:~$ ls
arquivo_1 arquivo_2
pessoa@meu_computador:~$ ls -l
total 4
-rw-rw-r-- 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1
-rw-rw-r-- 1 pessoa algum_grupo 4 Aug 30 15:26 arquivo_2
pessoa@meu_computador:~$ ls -lh
total 4.0K
-rw-rw-r-- 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1
-rw-rw-r-- 1 pessoa algum_grupo 4 Aug 30 15:26 arquivo_2
pessoa@meu_computador:~$ ls -lah
total 28K
drwxr-xr-x 2 pessoa pessoa 4.0K Aug 30 15:46 .
drwxr-xr-x 1 root root 4.0K Aug 30 15:09 ..
-rw----- 1 pessoa pessoa 339 Aug 30 15:48 .bash_history
-rw-r--r-- 1 pessoa pessoa 220 Aug 30 15:09 .bash_logout
-rw-r--r-- 1 pessoa pessoa 3.7K Aug 30 15:09 .bashrc
-rw-r--r-- 1 pessoa pessoa 807 Aug 30 15:09 .profile
-rw-rw-r-- 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1
-rw-rw-r-- 1 pessoa algum_grupo 4 Aug 30 15:26 arquivo_2
pessoa@meu_computador:~$ █
```

Propriedades dos arquivos

Tipo de arquivo

- - → arquivo comum
- d → diretório
- l → link (ligação)

```
-rw-rw-r-- 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1  
-rw-rw-r-- 1 pessoa algum_grupo 4 Aug 30 15:26 arquivo_2
```

permissões

Usuário
proprietário

Grupo
proprietário

tamanho

Data e hora
da última
modificação

nome

Trabalhando com arquivos no bash

- ls → lista arquivos
- touch → cria arquivos
- cp → copia arquivos
- rm → apaga arquivos
- nano → editor de texto simples
- cat → exibe o conteúdo de um arquivo

Dica:
A tecla <TAB> completa comandos,
caminhos e nome de arquivos

```
-rw-rw-r-- 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1  
-rw-rw-r-- 1 pessoa algum_grupo 4 Aug 30 15:26 arquivo_2
```


Trabalhando com arquivos no bash

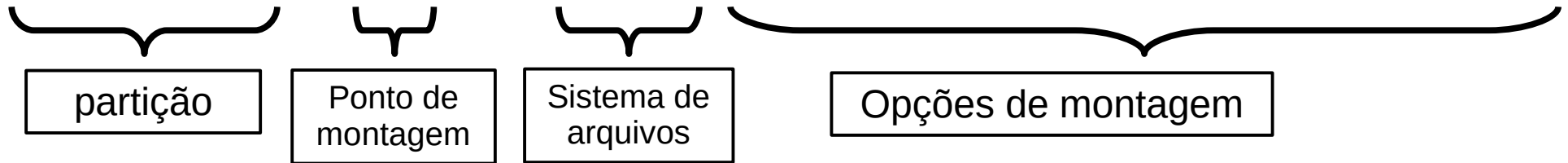
- head → exibe as primeiras linhas de um arquivo
- tail → exibe as últimas linhas de um arquivo
- less → exibe arquivo com possibilidade de navegação
 - q para sair

Caminhos no linux

No Linux todos os arquivos (e muitos recursos do sistema) são acessíveis a partir do diretório raiz (/)

- `mount` → anexa e desanexa partições físicas ao sistema; sem argumentos mostra lista de partições montadas

```
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
```



```
/dev/sdb1 on /media/pendrive type ext4 (rw,relatime,errors=remount-ro)
```

```
# umount /media/pendrive  
$ sudo umount /media/pendrive
```



Desmonta um sistema de arquivos
(vulgo ejetar / remover com segurança)

Garante que todos os dados foram gravados no disco antes da remoção

Caminhos no linux

- `pwd` → mostra o diretório atual
- `cd` → muda de diretório
- `mkdir` → cria um diretório
- `rmdir` → apaga um diretório

Acessando e copiando arquivos para computadores remotos

- `ssh` → permite acessar recursos em computadores remotos de forma segura
 - `$ ssh meu_username@lief.if.ufrgs.br`
- `scp` → copia arquivos através do protocolo `ssh`
 - `$ scp arquivo_1 meu_username@lief.if.ufrgs.br:dir/`
- `rsync` → sincroniza arquivos copiando apenas as diferenças (incremental)
 - `$ rsync -av ~/dir1_local/ meu_user@lief.if.ufrgs.br:dir1_remoto/`

Atenção: alguns comandos requerem que o diretório do destino já exista

`rsync` é uma ferramenta ideal para realizar backups e sincronizar pastas entre diferentes computadores

Operadores em bash

- Operador “>”: direciona a saída de um programa para um arquivo
 - \$ ls > meus_arquivos.txt
 - \$ cat meus_arquivos.txt
- Operador “|” (pipe): direciona a saída de um programa para outro
 - \$ seq 1000
 - \$ seq 1000 | less
- Operador “;”: separa comandos na mesma linha de comando. Execução serial.
 - \$ date ; sleep 1 ; date ; sleep 2
- Operador “&”: separa comandos na mesma linha de comando. Execução paralela.
 - \$ date & sleep 1 & date & sleep 2

Manipulação de texto

- echo → imprime um texto
 - Ex.: `$ echo "Olá mundo!"`
- grep → imprime linhas que correspondem à expressão
 - Sintaxe 1: `grep "expressão" arquivo`
 - Ex.: `$ grep root /etc/passwd`
 - Sintaxe 2: grep recebendo saída de outro programa através do pipe ("|")
 - `$ seq 1000 | grep 00`
- sed → filtra e transforma texto (avançado)
 - Exemplo: busca e substitui: `'s/expressão/substituto/g'`
 - `$ seq 0 50 1000 | sed 's/00/ Centena/g'`
 - `'s/00/ Centena/g'` substitui "00" por " Centena"
 - Pode ser usado em arquivos. Exemplo de substituição de texto

```
echo abc > arquivo.txt
sed -i 's/b/ B /g' arquivo.txt # -i indica in-place (altera o arquivo - perigoso!)
cat arquivo.txt
```

Segurança

- Permissões dos arquivos
- Integridade dos dados
- Senhas seguras

Permissões

9 campos agrupados de 3 em 3 identificando as permissões para o proprietário, grupo e todos usuários

- r → leitura
- w → gravação
- x → execução

```
-rwxrwxrwx 1 pessoa algum_grupo 0 Aug 30 15:26 arquivo_1
```

Permissões do grupo
proprietário

- r w x r w x r w x

Permissões do usuário
proprietário

Permissões de todos
usuários do computador

Mudar permissões com chmod:

- \$ chmod o-w arquivo_1
-rwxrwxr-x
- \$ chmod o-rwx arquivo_1
-rwxrwx---
- \$ chmod g-x arquivo_1
-rwxrw-r--

Integridade dos dados

- Função hash https://pt.wikipedia.org/wiki/Função_hash
 - é um algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo. Os valores retornados por uma função hash são chamados valores hash, códigos hash, somas hash (hash sums), checksums ou simplesmente hashes.
- Exemplos:
 - MD5 <https://pt.wikipedia.org/wiki/md5sum>
 - Produz um valor de hash de 128 bits expresso em 32 caracteres. Seguro contra alterações não intencionais. Ex:
 - `$ echo a > a.txt ; md5sum a.txt`
`60b725f10c9c85c70d97880dfe8191b3 a.txt`
 - SHA-1 <https://pt.wikipedia.org/wiki/SHA-1>
 - Produz um valor de hash de 160 bits (20 bytes), expresso em 40 caracteres. Ex:
 - `$ echo a > a.txt ; sha1sum a.txt`
`3f786850e387550fdab836ed7e6dc881de23001b a.txt`

Senhas seguras

- Senhas devem ser únicas, longas e conter diversos tipos de caracteres
- Conveniente utilizar geradores e gerenciadores de senha
 - pwgen → gerador de senhas
 - \$ pwgen 15 # gera uma senha de 15 dígitos
 - \$ pwgen -s 15 # -s: caracteres aleatórios
 - \$ pwgen -y 15 # -y: inclui símbolos
 - \$ pwgen -ys 25 # sugestão para gerar senhas MUITO seguras (se você não vai nunca digitá-la)
 - Keepass2 → gerenciador de senhas
 - Banco de dados criptografado pelo método AES-256 a partir de uma chave mestre