

Comandos Básicos

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

15 de Agosto de 2019

Básico do Sistema de Arquivos

```
/
/etc
  /etc/passwd
  /etc/rc.d
  ...
/home
  /home/joaomarclo
  /home/aluno01
  ...
/usr
  /usr/local
  /usr/lib
  ...
```

- ▶ A partição de inicialização é montada na raiz (/).
- ▶ Partindo da raiz, outros diretórios são organizados de acordo com cada distribuição, mas há um padrão geral:
 - ▶ /etc: arquivos de configuração específicos do computador.
 - ▶ /boot: imagem do *kernel*.
 - ▶ /dev: dispositivos.
 - ▶ /home: diretórios dos usuários.
 - ▶ /lib: bibliotecas básicas.
 - ▶ /bin: executáveis acessíveis por todos os usuários.
 - ▶ ...
- ▶ https://pt.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Comandos no Linux

Formato geral:

```
$ comando [-opções...] [argumentos...]
```

Para obter ajuda:

```
$ comando --help (ou -help)
```

```
$ man comando
```

- ▶ Pressionar a tecla *seta para cima* repete os últimos comandos digitados.
- ▶ Você pode apenas digitar o início do comando e apertar TAB, o *shell* completa o comando (ou caminho do diretório).
- ▶ Caso o *shell* não complete, TAB duas vezes mostra as opções disponíveis.
- ▶ CTRL+R permite buscar nos últimos comandos digitados.

pwd - Informa nome do diretório corrente

```
$ pwd  
/home/joaomarclo
```

- ▶ Retorna o diretório atual.
- ▶ Não parece muito útil a primeira vista.
- ▶ Em *scripts* vai permitir que verifiquemos se estamos no diretório correto.

cd - Navegando entre diretórios

```
$ cd /usr/local
```

```
$ pwd
```

```
/usr/local
```

Os seguintes atalhos podem ser usados tanto no *cd* quanto em outros comandos:

.	(ponto)	Diretório atual
..	(dois pontos)	Diretório anterior
~	(acento til)	Diretório <i>home</i> do usuário
/	(barra)	Diretório Raiz
-	(hífen)	Último diretório

ls - Lista Arquivos

```
$ ls [opções] [arquivo/diretório]
```

-l	Lista detalhada
-a	Lista arquivos ocultos
-r	Ordem reversa
-h	Formato legível
-R	Listar subdiretórios

cp - Cópia de arquivos e diretórios

```
$ cp [opções] <origem> <destino>
```

-i	Modo interativo
-v	Exibe cada arquivo copiado
-r	Copia diretórios e subdiretórios
-p	Tenta preservar o máximo de atributos

Em *scripts*, talvez a opção mais importante seja `-p`.

mv - Move arquivos e diretórios

```
$ mv [opções] <origem> <destino>
```

- ▶ Mover é uma maneira de renomear arquivos!
- ▶ Mover costuma ser mais lento, principalmente se for entre partições diferentes.
- ▶ Pode ser aplicado tanto para arquivos quanto diretórios.
- ▶ A opção `-i` pergunta antes de sobrescrever arquivos existentes. Não confie que ela está ativa!

ln - Ligações entre arquivos

```
$ ln [-s] <origem> <ligação>
```

- ▶ *Links* são atalhos.
- ▶ *Hard Link*: o arquivo de ligação tem o mesmo **inode** do arquivo de origem.
- ▶ *Soft Link*: o arquivo de ligação contém apenas o caminho do arquivo de origem.
- ▶ A *origem* pode ser tanto arquivos quanto diretórios.

mkdir - Cria um diretório

```
$ mkdir [opções] <nomedodiretorio>
```

- ▶ Você pode usar todas as abreviações do comando *cd*.
- ▶ Subdiretórios não são criados automaticamente.

rmmdir - Remove diretório

```
$ rmmdir <nomedodiretorio>
```

- ▶ Para usar esse comando, o diretório tem que estar vazio!
- ▶ Pode parecer uma limitação, mas é bom usar em um *script* quando deseja ter certeza que o diretório excluído está vazio.

rm - Deleta arquivos e diretórios

```
$ rm [opções] <arquivos>
```

-f	Modo força bruta
-I	Modo Confirmação
-r	Remove diretórios

A opção `-r` remove mesmo se o diretório contenha arquivo.

file - Indica o tipo de arquivo

```
$ file <arquivo>
```

- ▶ Extensões de arquivos não tem significado preciso no Linux.
- ▶ O *file* abre a porção inicial do arquivo e extrai informações sobre sua natureza.

basename e dirname - Retornam nome do arquivo e nome do diretório

```
$ basename <arquivo>
```

```
$ dirname <diretório>
```

Novamente, não são muito úteis fora de *scripts*.

grep - Pesquisa arquivos por conteúdo

```
$ grep [-opções] [expressão] [arquivo1] [arquivo2] [arquivo3] ...
```

-c	Apenas informar quantas linhas contém a expressão
-i	Não diferenciar maiúsculas e minúsculas
-l	Apenas informar quais dos arquivos contém a expressão
-v	Busca reversa
-n	Exibe o número da linha

A *expressão* é uma E.R. Por enquanto, vamos considerar como uma simples cadeia de caracteres.

find - Procurar arquivos

```
$ find [caminho] [expressão] [ação]
```

caminho : diretório a partir do qual a busca começa.

expressão :

- name *nome*
- user *usuário*
- group *grupo*
- type *tipo*
- size \pm *tamanho*
- atime \pm *dias*
- ctime \pm *dias*
- mtime \pm *dias*

ação :

- print
- exec *comando* {} \;
- ok *comando* {} \;
- printf *formato*

cat - Exibe conteúdo de arquivos

```
$ cat [-opções] [arquivo]
```

-v	Mostra caracteres sem representação na tela
-e	Mostra a alimentação de linha
-t	Mostra os TAB
-n	Enumera as linhas

wc - Contar

```
$ wc [-lwc] [arquivo]
```

-l	Linhas
-w	Palavras
-c	Caracteres

- ▶ A maneira mais fácil de saber se dois arquivos são diferentes.
- ▶ Muito útil para saber se *log* contém novas informações.

head - Exibe linhas a partir do início

```
$ head [-número] [-cn] [arquivo]
```

-número	Linhas a partir do início que são mostradas
-c	Bytes a partir do início que são mostrados
-n	igual à <i>-numero</i>

tail - Exibe linhas em relação ao final

```
$ tail [-número] [+número] [-cnf] [arquivo]
```

+número	Mostra a partir da linha número até o fim.
-número	Mostra as número linhas do arquivo.
-c	Mostra os últimos <i>bytes</i>
-n	Igual à -número
-f	Exibe o final do arquivo a medida que ele cresce.

sort - Ordenar arquivos

```
$ sort [opções] [arquivo]
```

-k	Define a chave de ordenação no formato $n_1.m_1, n_2.m_2$
-t	Define o caractere de separação
-m	Intercala arquivos
-n	Classificação numérica
-r	Classificação inversa

chown - Trocando o Dono do Arquivo

```
$ chown [-f] [-R] dono arquivo
```

arquivo	objeto a ter as permissões alteradas
dono	novo dono do arquivo
-f	não reportar erros
-R	diretórios e subdiretórios

chgrp - Trocando o Dono do Arquivo

```
$ chgrp [-f] [-R] grupo arquivo
```

arquivo	objeto a ter as permissões alteradas
dono	novo dono do arquivo
-f	não reportar erros
-R	diretórios e subdiretórios

Tipos de Acesso

dono (u)	usuário dono do arquivo
grupo (g)	o grupo a que pertence o arquivo
outros (o)	outros usuários

Leitura(r), escrita(w) e execução(r).

chmod - Ajustando as Permissões de Arquivo

```
$ chmod string-de-acesso arquivo
```

string-de-acesso	formato padrão de permissões
arquivo	nome dos arquivos ou diretórios

who - Usuários Ativos

```
$ who [-mH] [am i]
```

am i	informações do próprio usuário
-H	mostra um cabeçalho
-a	informações detalhadas

id - Identificadores do Usuário

```
$ id [-ngu]
```

-n	informações do próprio usuário
-g	mostra apenas o grupo
-G	mostra todos os grupos
-u	apenas o identificador do usuário

chfn - Mudar as informações do usuário

```
$ chfn [-frpho]
```

-f	modifica o nome do usuário
-r	modifica a localização do usuário
-p	modifica o telefone de trabalho do usuário
-h	modifica o telefone residencial do usuário
-o	informações adicionais (somente root)

date - Exibe altera a data

```
$ date [MMDDhhmm] [+formato]
```

[MMDDhhmm]	atualiza a data
+formato	indica como exibir a data

ssh - Login Remoto Criptografado

```
$ ssh [-p PORTA] [usuario]@IP
```

-p	Tentar se conectar na PORTA no lugar da 22 (padrão)
usuario	usuário na máquina remota

scp - Copiar Arquivo Através do ssh

```
$ scp [-P PORTA] [-r] arquivo [usuario]@IP:[diretorio]
```

-P	Tentar se conectar na PORTA no lugar da 22 (padrão)
-r	cópia recursiva
diretorio	destino onde será copiado o arquivo no servidor remoto

tar - Agrupar Arquivos em Um

```
$ tar [-cfprtuvx] [arquivo_tar] [arquivos]
```

-c	Cria novo arquivo
-f	Indica que o destino é o disco
-p	Preserva permissões
-r	Anexa ao final de um arquivo .tar existente
-t	Lista conteúdo de um arquivo .tar
-u	Adiciona apenas arquivos novos ou modificados
-v	Mostra o nome de cada arquivo processado
-x	Retira os arquivos do .tar

```
# Compacta pasta teste
```

```
$ tar -czvf teste.tar.gz teste/
```

```
# Descompacta arquivo
```

```
$ tar -xzvf teste.tar.gz
```


Atividade de Comandos Básicos - Parte I

Esta atividade deve ser feita **inicialmente** na sua máquina local.

- ▶ Crie um diretório chamado *ufc_quixada*.
- ▶ Crie um subdiretório chamado *redes_de_computadores*.
- ▶ Em *redes_de_computadores*, crie um subdiretório chamado *grade_curricular*.
- ▶ Acesse <http://rc.quixada.ufc.br/matriz-curricular/> e anote o nome cada disciplina do curso.
- ▶ Para cada disciplina:
 - ▶ Crie um subdiretório em *ufc_quixada/redes_de_computadores/grade_curricular*.
 - ▶ O nome do diretório deve ser em minúsculas e os espaços devem ser trocados por `_`.
 - ▶ Por exemplo: Internet e Arquitetura TCP/IP corresponde do diretório *internet_e_arquitetura_tcpip*

Atividade de Comandos Básicos - Parte II

Este exercício é continuação da Parte I.

- ▶ Acesse `www.quixada.ufc.br` e anote o nome e o primeiro sobrenome de cada professor que você já foi aluno.
- ▶ Crie um diretório `ufc_quixada/redes_de_computadores/professores`.
- ▶ Dentro do diretório criado, crie um diretório para cada professor. O nome do diretório de cada professor deve seguir o padrão `nome_sobrenome`.
- ▶ Exemplo: deve existir o diretório `ufc_quixada/rede_de_computadores/professores/joao_marcelo`.

Atividade de Comandos Básicos - Parte III

Este exercício é continuação da Parte II.

- ▶ Crie o diretório
ufc_quixada/redes_de_computadores/conquistas.
- ▶ Dentro de *conquistas*, você vai criar um diretório para cada disciplina que já foi aprovado(a).
- ▶ Dentro do diretório criado, defina dois *soft links*:
 - ▶ *programa*, que deve apontar para a pasta da disciplina em *ufc_quixada/redes_de_computadores/grade_curricular*.
 - ▶ *professor*, que deve apontar para a pasta do professor em *ufc_quixada/redes_de_computadores/professores*.

Atividade de Comandos Básicos - Parte IV

Este exercício é continuação da Parte III.

- ▶ Compacte sua pasta *ufc_quixada* no arquivo *ufc_quixada.tar.gz*.
- ▶ No servidor, dentro do seu diretório de usuário, crie a pasta *atividades*.
- ▶ Crie o subdiretório *atividade02*.
- ▶ Usando o comando *scp*, copie o arquivo *ufc_quixada.tar.gz* para o diretório remoto *atividades/atividade02*.
- ▶ Descompacte o arquivo *ufc_quixada.tar.gz* e verifique se os *links* estão corretos.
- ▶ Altere as permissões dos diretórios *grade_curricular* e *professores* para apenas leitura.

Resultado da Atividade

- ▶ Pela lógica, deve existir a pasta *atividades/atividade02/ufc_quixada* na sua pasta no servidor.
- ▶ Dentro dela, devem existir a pasta *redes_de_computadores* com os seguintes subdiretórios:
 - ▶ *grade_curricular* com subdiretórios para todas as disciplinas.
 - ▶ *professores* com subdiretórios para todos os professores.
 - ▶ *conquistas* com pastas para as disciplinas que foi aprovado e dentro delas ligações (*links*) para as pastas no diretório *grade_curricular* e *professores*.

Objetivo desse trabalho todo...

Garantir que o aluno tem prática no uso da linha de comando.