

# Introducao ao Sistema Operacional Linux

**Eduardo Augusto Bezerra**

## **Exercicios:**

- Realizar o login no sistema digitando o solicitado (username e senha)
- Um dos comandos mais uteis do Linux e' o "man". Esse comando e' utilizado para mostrar paginas de ajuda no Linux. Para exercitar digitar o comando abaixo para obter ajuda sobre o funcionamento do proprio "man" (para sair do man basta pressionar a tecla q):

**man man**

- Para cada um dos comandos mostrados a seguir, digitar o "man" antes de utilizar o comando.
- Criar um diretorio (pasta) para a execucao dos exercicios propostos. No prompt digitar:

**mkdir *sobrenome***

onde *sobrenome* e' o seu nome de familia

- No prompt digitar:

**cd *sobrenome***

- Todos os exercicios dessa lista deverao ser executados dentro deste diretorio. Nesse exercicio foram utilizados dois comandos basicos do Linux:

cd == change directory

mkdir == make directory

- A seguir utilizar o comando "ls" para listar o conteudo de diretorios e o comando "touch" para mudar os dados relativos a criacao de um arquivo. Uma observacao importante sobre o comando "touch" e que quando utilizado com um nome de arquivo que nao existe, o arquivo sera' criado.
- No prompt executar a seguinte sequencia:

**ls**

(o diretorio devera' estar vazio)

```
touch teste1.txt
touch teste2.text
touch learnlatex.tex
touch fubar.extension
```

- Utilizar novamente o "ls" para listar o conteudo do diretorio apos as operacoes realizadas. A saida devera' ser a seguinte:

```
learnlatex.tex  teste1.txt  teste2.text  fubar.extension
```

- Para obter a listagem do diretorio em colunas, e com mais detalhes sobre cada arquivo, digitar o seguinte:

```
ls -l
```

- A saida devera' ser a seguinte:

```
-rw-rw-r--  1 user group      0 Jun  4 12:30 learnlatex.tex
-rw-rw-r--  1 user group      0 Jun  4 12:30 teste1.txt
-rw-rw-r--  1 user group      0 Jun  4 12:30 teste2.text
-rw-rw-r--  1 user group      0 Jun  4 12:30 fubar.extension
```

onde "user" e' o nome do usuario (dono do arquivo), e "group" e' o group no qual o usuario faz parte.

- Listar o conteudo de um diretorio, ordenando os arquivos por ordem de criacao:

```
ls -lt
```

Executar o comando "ls" e comentar o resultado da obtido.

- Executar:

```
touch teste1.tex
```

- Executar novamente "ls -lt" e comparar o fornecido com a listagem anterior. Alguma diferenca?
- Utilizar o comando "cp" para copiar arquivos, e o comando "mv" para mover arquivos (ou renomear). Executar:

```
cp teste1.tex teste2.tex
```

Executar o comando "ls" e comentar o resultado da obtido.

```
mv teste1.tex teste1.tex.old
```

Executar o comando "ls" e comentar o resultado da obtido.

- Utilizar o comando "mkdir" para criar diretorios. Executar:

```
mkdir olds
mv teste1.tex.old olds/
```

Executar o comando "ls" e comentar o resultado da obtido.

- Executar a seguinte sequencia de comandos, mesmo que ja' tenham sido executados anteriormente, e comentar o

resultado obtido:

```
mkdir olds
mkdir olds
```

- Mais copia de arquivos. Executar:

```
cp testel.tex ahan.tex
cp naosei.txt aiaiai.txt
```

Executar o comando "ls" e comentar o resultado da obtido.

- Executar:

```
touch moveral
touch movera2
mv moveral exempl.exp
mv exempl.exp olds/
mv movera2 newfile.tmp
```

Executar o comando "ls" e comentar o resultado da obtido.

- **Desafio:** utilizar o comando "ls" para listar todos os arquivos com extensao .tex
- Comando "grep" para pesquisa de palavras em arquivos. Criar um arquivo com algumas linhas de texto. O nome do arquivo sera' fubar.test. Digitar a seguinte sequencia:

```
cat > fubar.test
digite seu nome
idade
nome dos pais
parte inicial do programa da semana academica (procure repetir
algumas palavras em mais de uma linha)
pressionar Ctrl-D para encerrar o arquivo.
```

Digitar:

```
cat fubar.test | grep ???
```

Onde ??? e' a palavra (ou parte da palavra) a ser encontrada no arquivo fubar.test.

- Utilizar o comando acima para mostrar que existem as seguintes palavras no arquivo criado: "nome", "sobrenome", "nome do pai", "para", "que"
- Utilizar o comando "rmdir" para remover diretorios VAZIOS. O comando "rm" com a opcao "recursive" pode ser utilizado para remover diretorios que nao estejam vazios (use o man para ver como fazer isso).

```
mkdir maisum
rmdir maisum
rmdir testes
```

- Remocao de diretorios e arquivos. Utilizar o comando "rm" para remover os arquivos "testel.tex" e "teste2.tex". Remover tambem o diretorio "teste2".

- Utilizar o comando "tail" para listar o final de um arquivo. Esse comando mostra as ultimas 10 linhas (ler o man do comando). "tail -f" mostra as ultimas 10 linhas do arquivo e qualquer nova linha que seja adicionada a esse arquivo. Criar um arquivo texto (com cat ou touch) com pelo menos umas 20 linhas, e execute "tail nome\_do\_arquivo" sem opcoes nesse arquivo. A seguir execute "tail -f nome\_do\_arquivo", onde "nome\_do\_arquivo" e' o nome dado ao arquivo na criacao. Para sair do "tail -f" utilizar Ctrl-C.
- Utilizar o comando "cat nome\_do\_arquivo" para listar na tela o conteudo de um arquivo. Executar "cat" com o arquivo criado para o exemplo anterior e comentar a diferenca entre a saida fornecida pelo "cat" e pelo "tail".
- O comando "more" e' semelhante ao "cat", porem apresenta o conteudo do arquivo quadro-a-quadro. E' semelhante ao funcionamento do "man". Pressionar a barra de espaco para avancar uma pagina inteira e a tecla "enter" para avancar linha-a-linha.
- O comando e' semelhante ao "more", porem permite navegar pelo arquivo.
- Digitar a seguinte sequencia de comandos ("clear" limpa a tela):

```
cat /usr/include/time.h
clear
tail /usr/include/time.h
clear
tail -f /usr/include/time.h
Ctrl-D
clear
more /usr/include/time.h
q
clear
less /usr/include/time.h
```

Use as setas para navegar no texto.

Comentar as diferencas observadas entre os comandos executados.

- O comando "echo" e' bastante utilizado em scripts para mostrar informacoes na tela. Executar a seguinte sequencia de comandos:

```
echo "teste"
echo $SHELL
echo $PATH
echo "ahanahnahn" > teste.echo
```

Comentar as saidas fornecidas.

- O comando "history" mostra o historico de comandos digitados na sequencia e ordenadamente. Executar "history" no prompt e observar o resultado fornecido na tela.
- O comando "!" e' bastante importante na economia de tempo na digitacao de comandos. Ao se digitar "!" no prompt, o ultimo comando sera' executado novamente. Caso seja digitado "!ca" no prompt, o ultimo comando que iniciado por "ca" sera' executado novamente. Ao se executar !2 o segundo comando sera' executado novamente (provavelmente foi a criacao do

diretorio no inicio dessa bateria de exercicios). Executar no prompt:

## history

Observar o numero que precede o comando "cat /usr/include/time.h" executado anteriormente. Digitar o comando "!" seguido por esse numero:

**!n**

onde "n" e' o numero do comando "cat /usr/include/time.h".

- Pipes sao bastante importantes no dia-a-dia do usuario Linux. Por exemplo, para listar o conteudo de um diretorio mostrando todos os atributos dos arquivos e ordenados por data basta utilizar o comando "ls -lt". Porem a lista de arquivos e' muito longa e nao cabem todos na tela. Utilizar um pipe da seguinte maneira:

**ls -lt | more**

- Outra opcao para o problema acima seria redirecionar a saida da listagem de arquivos para um arquivo, e depois utilizar comandos tipo "more" ou "less" para ver o arquivo. Para redirecionar a saida do "ls", utilizar:

**ls -lt > log** ou  
**ls -lt >> log**

A diferenca entre o > e o >> e' que o primeiro escreve por cima apagando o arquivo caso exista. No segundo e' realizado um *append* das informacoes. Fazer um teste com arquivos criados anteriormente.

- O comando "chmod" e' utilizado para alterar as permissoes de um arquivo. Utilizar esse comando para alterar as permissoes do arquivo "fubar.test", de acordo com as dicas vistas na aula teorica. Para mais detalhes sobre o "chmod", utilizar o "man".
- O comando "chown" altera o dono do arquivo (change owner). Executar esse comando para alterar o dono do arquivo fubar.test para um colega da turma.
- O comando "date" mostra a data do sistema. Executar esse comando e observar o resultado.
- O comando "spell" verifica erros ortograficos em palavras ou textos. Digitar "spell fubar.test" e verificar a longa lista de erros de ortografia ("spell" utiliza dicionario de ingles). Digitar "spell" seguido de Enter, e forneça algumas palavras, intercaladas por Enter. Ao encerrar a digitacao da lista de palavras, pressionar Ctrl-D para finalizar a entrada de dados. Sera' apresentada a lista de palavras nao encontradas no dicionario do "spell".
- O comando "wc" informa o numero de palavras, linhas e caracteres existentes em um arquivo. Digitar:

**wc fubar.test**

Comentar o resultado obtido.

- O comando "df" mostra a porcentagem do disco que foi utilizada.
- O comando "du" mostra a ocupacao em Mega bytes de um arquivo, diretorio ou disco. Utilize "man du" para descobrir mais sobre esse comando.
- Utilizar o comando "locate" (ou "whereis") para localizar arquivos:

#### **locate mkdir**

- Utilizar o comando "find" para localizar arquivos a partir de um determinado diretorio:

#### **find /home -name fubar.test**

- Utilizar o comando "whoami" (quem sou eu) para saber quem esta' logado.
- O comando "which" pode ser utilizado para mostrar o caminho completo de um comando. Executar:

#### **which mkdir**

Comente o resultado.

- O comando "users" e' utilizado para mostrar os usuarios conectados na maquina.
- O comando "w" e' utilizado para mostrar os usuarios conectados na rede a qual a maquina esta conectada.
- O comando "ps" mostra os processos em execucao na maquina. Os comandos "ps -elf" e "ps -gaux" mostram mais detalhes desses processos.
- O comando "kill" e' utilizado para matar processos na maquina. Utilizar primeiro o comando "ps" para descobrir a ID do processo a ser eliminado.
- O comando "talk" e' utilizado para chat com outro usuario conectado na mesma maquina. Ver o man





[linux.org](https://linux.org)