

FIAP GRADUAÇÃO

TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

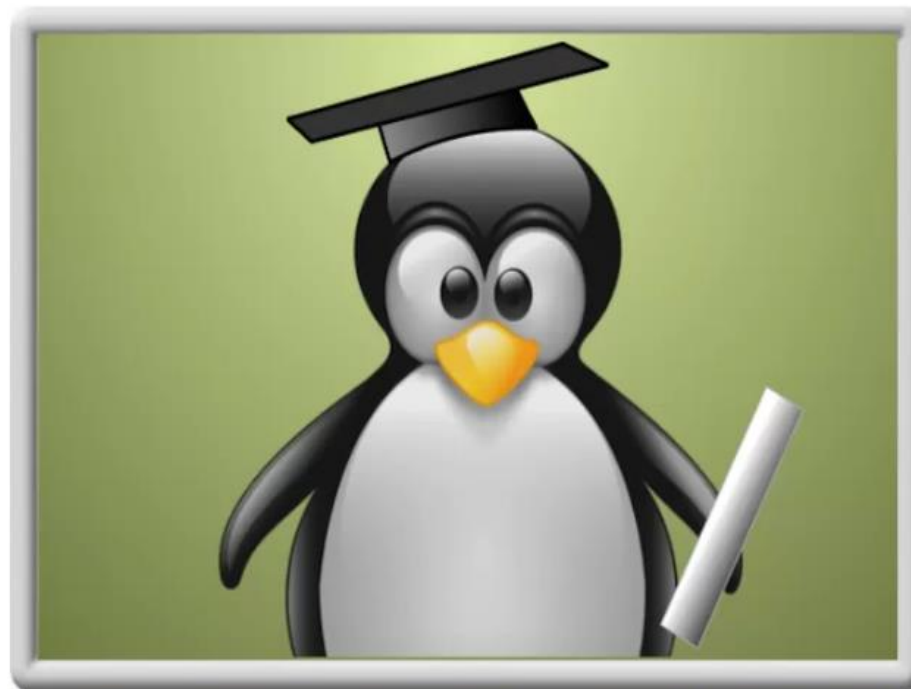
DevOps Tools & Cloud Computing
Virtualização: Comandos Básicos Linux

PROF. JOÃO MENK profjoao.menk@fiap.com.br

PROF. SALVIO PADLIPSKAS salvio@fiap.com.br

Linux

- Foi desenvolvido pelo finlandês Linus Torvalds, inspirado no sistema Minix
- O nome Linux surgiu da mistura de Linus + Unix
- Inicialmente desenvolvido e utilizado por grupos de entusiastas em computadores pessoais, o sistema Linux passou a ter a colaboração de grandes empresas, como a IBM, a Sun Microsystems, HP, Red Hat, Oracle, Google e a Canonical



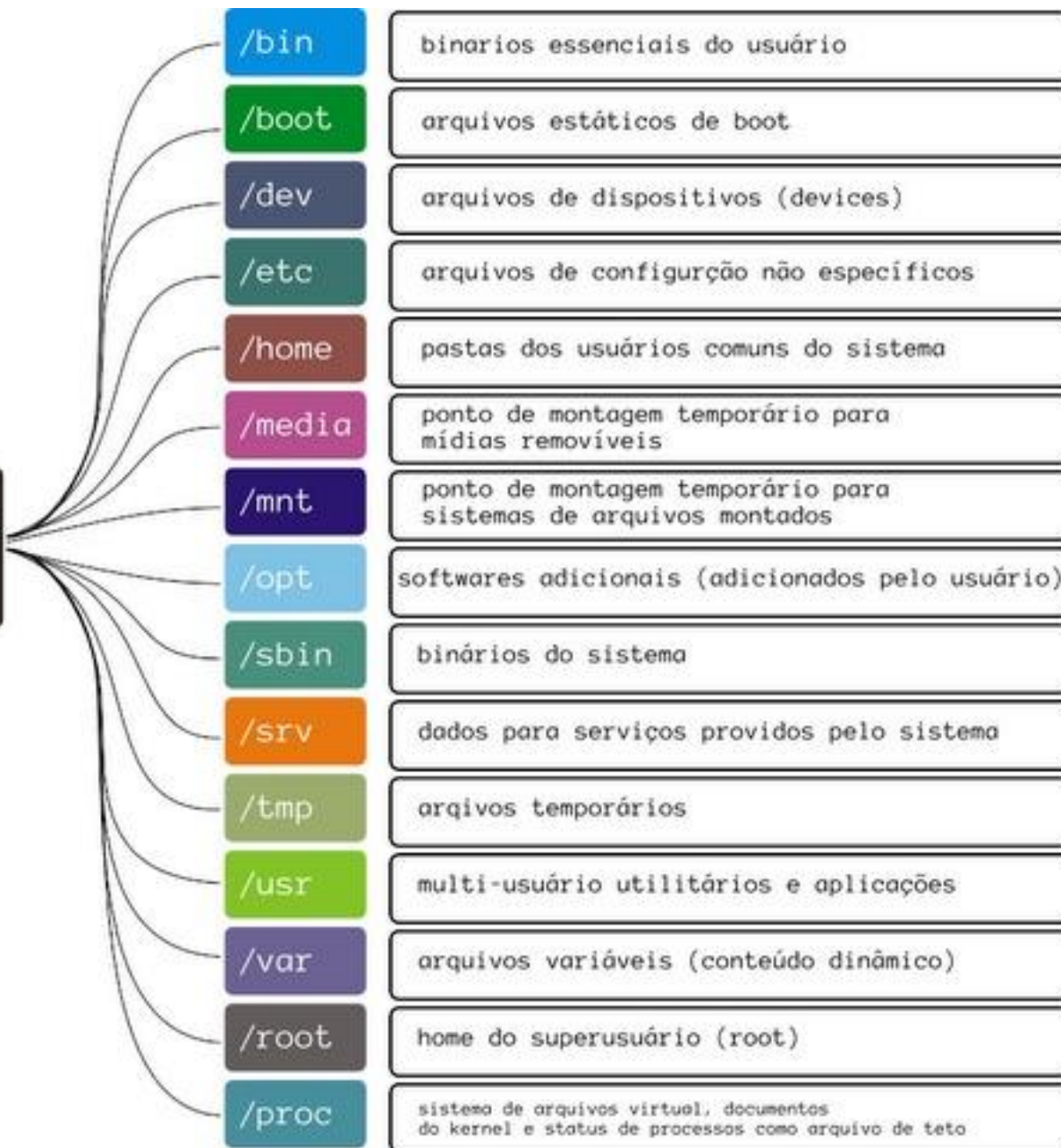
O pinguim Tux é a mascote do sistema operacional Linux criado por Larry Ewing em 1996 (Foto: Reprodução/Linux Foundation)

- Trata-se de um conjunto de softwares elaborados em torno do Kernel do Linux, tornando-o operacional para usos distintos
- Nesse processo, as distribuições adicionam seus próprios toques finais, como temas e softwares personalizados, além de escolher o ambiente desktop, o navegador web e outros programas padrões que irão rodar junto ao sistema operacional



Linux: Estrutura de Diretórios

FIAP



A estrutura de diretórios do Linux segue o FHS: File Hierarchy Standard

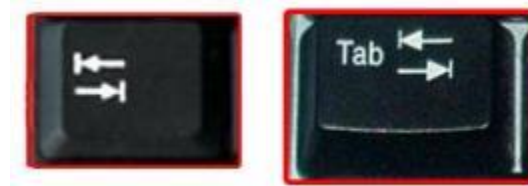
Um órgão que define os principais diretórios e o seu conteúdo em sistema Linux

Linux: Estrutura de Diretórios



- A tecla TAB pode ser usada para completar um comando. Por exemplo:

sql<TAB> : produz sqllite3



- Se a parte do comando digitado não for suficientemente único para completar o comando, digitar TAB duas vezes faz com que uma lista de comandos com a parte do texto digitado. Exemplo:

wh<TAB> produz

```
[oracle@oralinux76 ~]$ wh
whatis      whereis  which      while      whiptail   who         whoami
[oracle@oralinux76 ~]$ wh
```

- A seta para cima exibe os últimos comandos executados no Terminal



- O comando **history** lista os últimos comandos executados no Terminal

```
[oracle@orainux76 ~]$ history
```

```
26  ls
27  cd ~
28  pwd
29  celar
30  claer
31  clear
32  ;
33  cal
34  more /etc/passwd
35  ls -l um.sh
36  touch arq1
37  ls -l
38  ls -l arq1
39  chmod g+w arq1
40  ls -l arq1
41  chmod g-w arq1
42  ls -l arq1
43  chmod a+x arq1
```

1 - Apagar o Histórico:

history -c

2 - Com o atalho de teclado '**Ctrl+R**', é possível também fazer uma busca por uma String qualquer

3 - Se você quiser executar novamente o comando de número 35, execute o comando:
!35

ls: Lista Arquivos e Diretórios

-l

Mostra arquivo ou diretório, tamanho, data e hora modificada, nome de arquivo ou pasta e proprietário do arquivo e também as permissões de tudo o que for listado (Lista longa)

-a

Inclui arquivos ocultos

-ls

Lista os arquivos pelo tamanho, no caso começará pelo o arquivo de maior tamanho

-lt

Lista os arquivos pela data de alteração, no caso começará pela data mais nova

-r

Utilizando a opção **r** a ordenação será de forma reversa (Exemplos: **-lsr** / **-ltr**)

-R

Recursivo

Indicar um caminho para listar arquivos e diretórios: **ls <caminho>**

Exemplo: **ls /usr/bin**

- **Para conseguir ajuda com as opções dos comandos**
 - Digite qualquer comando seguido de “--help” (Dois traços e a palavra help) para ver a uma descrição detalhada do comando
- Outra forma de conseguir documentação oficial dos comandos Linux é o comando man (manual).

Digite man seguido do nome do comando que você precisa de informação.

Ex.: `man ls`

- Curingas
 - * Atua como um substituto para qualquer sequência de caracteres
 - ? Atua como substituto para qualquer caractere único

Vá para o diretório: [/etc](#)

1. `ls *.conf`
2. `ls -l r??.*`
3. `ls -l r*.*`
4. `ls [u-x]*.*`

- Mostrando data e hora
Para mostrar a data e a hora corrente, usamos o comando: **date**
Calendário: **cal**
- Limpando a tela
Para limpar a tela, usamos o comando: **clear** (**ctr+l**)
- Comandos de diretório
Mostre o diretório corrente com o comando: **pwd**
- Crie um diretório com o comando: **mkdir**
Vamos criar um diretório chamado ex1: **mkdir ex1**
- Mudando de diretório
Mude de diretório com o comando: **cd**
Vamos mudar para o diretório que acabamos de criar: **cd ex1**

Vamos agora mudar para o diretório home/oracle: **cd ..**

Os dois pontos que seguem o comando **cd** indica que estamos saindo do diretório corrente para um diretório um nível acima, nesse caso do exemplo, estamos indo para o diretório home/oracle

Se quisermos ir direto para um diretório, em qualquer lugar do diretório-raiz (/), basta digitar o comando **cd** seguido do nome do diretório. Por exemplo, se queremos ir para o subdiretório bin que está dentro de usr, basta digitar: **cd /usr/bin**

Vamos agora ao diretório-raiz com o comando: **cd /**

Para ir ao diretório home do usuário:

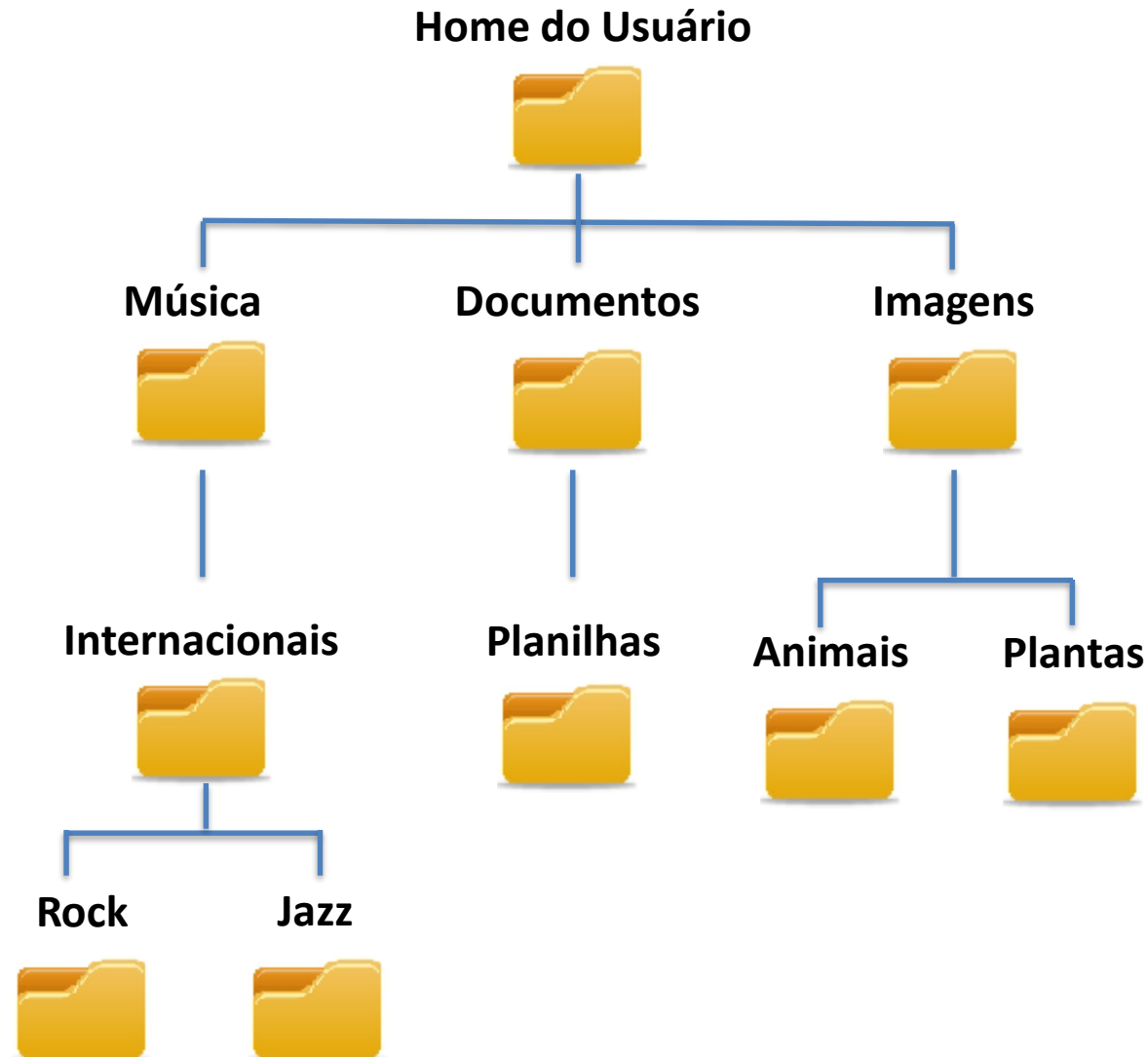
cd ~ ou simplesmente **cd**

Exercício Prático Comandos Linux: Parte I



Crie a seguinte estrutura de diretórios no Servidor Linux

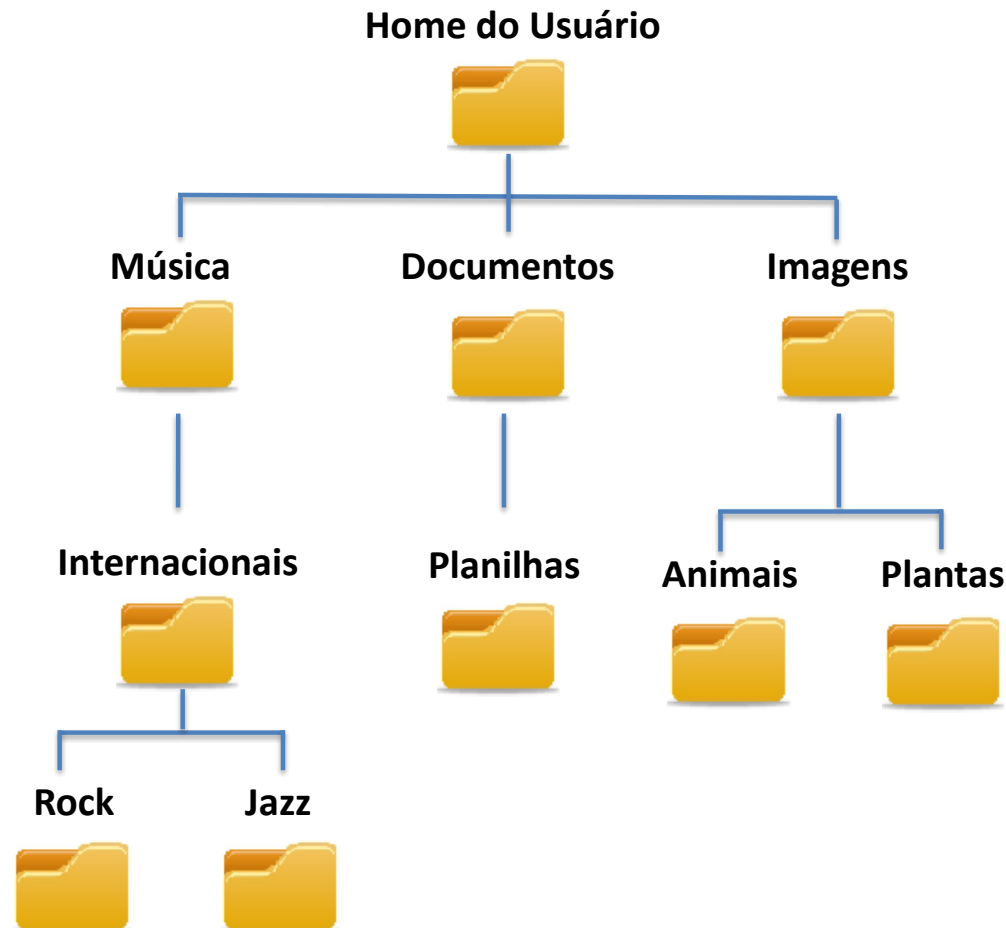
Dica: para entrar no diretório Música digite: `cd *sica` (teclado em inglês)



Exercício Prático Comandos Linux: Parte II



1. Copie ao menos um arquivo com conteúdo significativo para dentro de cada novo diretório criado utilizando o Cliente de FTP
2. Depois execute esses arquivos dentro da Máquina Virtual

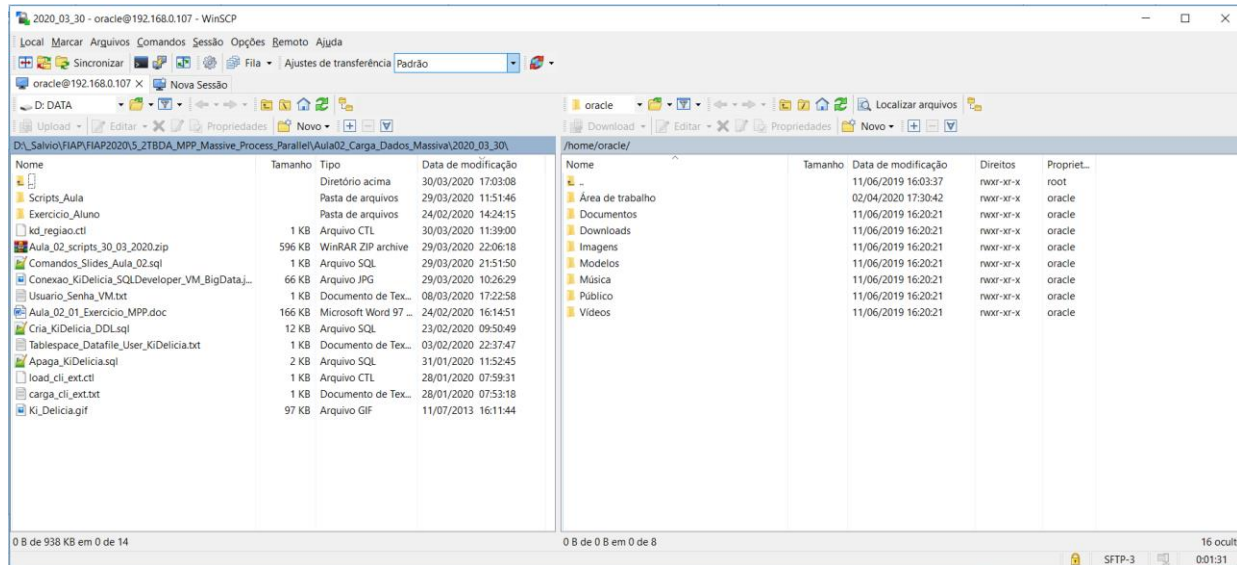


Solução: Exercício Prático Comandos Linux: Parte II



FIAP

1. Utilizando o **WinSCP** procure o arquivo no seu Host (lado esquerdo da tela), posicione o diretório desejado do Servidor Linux (lado direito da tela) e arraste o arquivo desejado



2. Para executar via linha comando:

Imagens: **eog <nome imagem e extensão>**

Músicas/Vídeos: **totem <nome imagem e extensão>**

3. Ou Clique em



navegue até o arquivo e dê uma duplo clique

- Listando o conteúdo de arquivos

Podemos listar o conteúdo de arquivos de duas formas: usando o comando **cat** ou o comando **more**

A diferença entre eles é que **more** permite a paginação do conteúdo, se ele for muito extenso

Vamos listar o conteúdo do arquivo passwd. Para isso, vá para o diretório **/etc** e digite: **cat passwd**

Com **more**, aparece uma indicação na parte inferior esquerda da tela, indicando o percentual do conteúdo mostrado. Tecle <enter> para continuar a paginação e pressione **Q/q** para interrompê-la

■ Criando **ALIAS**

O ALIAS funciona como um apelido para um comando

alias seuAtalho='Comando --OPCOES'

Exemplos:

```
alias c='clear'
```

```
alias ls='ls -la'
```

1. Criar um ALIAS para o comando rm: **alias rm='rm -i'**
2. Criar um ALIAS para listar o conteúdo de um diretório: **alias dir='ls -la'**

Para deixar o ALIAS atribuído de forma permanente:

Para todos os usuários (somente com o usuário root)

```
gedit /etc/bash.bashrc
```

Para o usuário corrente

```
gedit $HOME/.bashrc (*para dar refresh... source ~/.bashrc)
```

Linux: Comandos Básicos

■ Copiando arquivos e diretórios

Usamos o comando **cp** para esse propósito.

Sintaxe: **cp** (arquivo ou diretório origem) (arquivo ou diretório destino)

Vamos criar um arquivo dentro de **/home/oracle** e copiar esse arquivo para dentro de um subdiretório:

1. Vá ao diretório: **/home/oracle**
2. Crie o arquivo texto1 a partir do arquivo passwd em **/etc**
cat /etc/passwd > texto1
3. Crie o subdiretório ex1 e copie o arquivo criado para esse subdiretório
Crie o diretório com o comando **mkdir ex1**
cp texto1 ex1
4. Isso feito, insira o seguinte comando
ls -l ex1

- Alterando a data e hora de modificação ou criando arquivos novos vazios

Utilizamos o comando **touch** para realizar essas operações

Vamos criar alguns arquivos:

1. vá ao diretório **/home/oracle**
2. Crie um diretório **ex2**
3. Crie quatro arquivos dentro desse novo diretório: texto2, texto3, doc1 e doc2

touch texto2 (*insira um texto dentro desse arquivo)

touch texto3 (*insira um texto dentro desse arquivo)

touch doc1 (*insira um texto dentro desse arquivo)

touch doc2 (*insira um texto dentro desse arquivo)

■ Copiando o conteúdo de um diretório para outro

Vamos copiar todo o conteúdo do diretório `/home/oracle` para dentro do subdiretório `ex3`, que criaremos agora:

1. Vá ao diretório `/home/oracle`
2. Crie o diretório `ex3`
3. Entre no subdiretório criado e digite: `cp /home/oracle/* .`
4. Agora, escreva: `ls -l`
5. Veja o conteúdo do diretório original: `ls -l /home/oracle/`

Observe que os subdiretórios de `/home/oracle` não foram copiados, somente os arquivos

Para copiar os subdiretórios, use a opção "r" do comando

Vamos, agora, fazer o mesmo, só que, desta vez, copiando os subdiretórios e arquivos de `/home/oracle`

1. Primeiro, digite: `cp -r /home/oracle/* .`
2. Depois, digite o seguinte comando: `ls -l`

- Apagando arquivos e diretórios

Para apagar arquivos e diretórios utilizamos o comando: **rm**

Exemplos:

Apaga o arquivo especificado: **rm texto2**

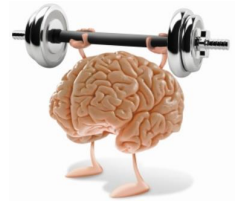
Apaga o arquivo especificado forçadamente (-f de force): **rm -f texto3**

Apaga o diretório especificado: **rm -r ex1**

Apaga o diretório especificado forçadamente: **rm -rf ex2**

1. Vá ao diretório: **/home/oracle/ex2**
2. Apague os arquivos: **texto*** (confirmando)
3. Apague os arquivos: **doc*** (sem solicitar confirmação)
4. Apague o diretório: **ex1** (confirmando)
5. Apague o diretório: **ex2** (sem solicitar confirmação)

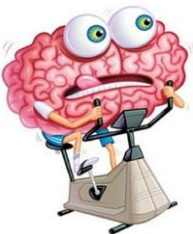
1. Vá ao diretório: `/home/oracle/ex2` `cd /home/oracle/ex2`
2. Apague os arquivos: `texto*` (confirmando) `rm texto*`
3. Apague os arquivos: `doc*` (sem solicitar confirmação) `rm -f doc*`
4. Apague o diretório: `ex1` (confirmando)
`cd ~`
`rm -r ex1`
5. Apague o diretório: `ex2` (sem solicitar confirmação) `rm -rf ex2`



EXERCÍCIO



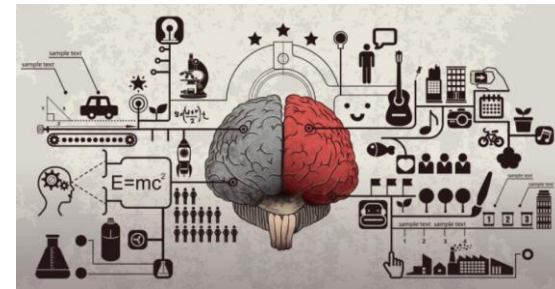
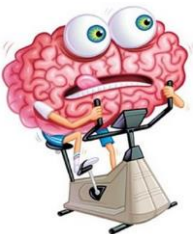
- 1) Entre no seu diretório HOME
- 2) Crie um diretório com o nome "atividade1"
- 3) Entre no diretório criado
- 4) Crie dois arquivos: "arq1.txt" e "arq2.txt" com base no arquivo "/etc/yum.conf". Utilize o comando cat



EXERCÍCIO



- 5) Liste os arquivos utilizando a opção de "lista longa"
- 6) Limpe o terminal
- 7) Mostre a data atual do sistema
- 8) Elimine os arquivos e a estrutura de diretório criada nos exercícios acima

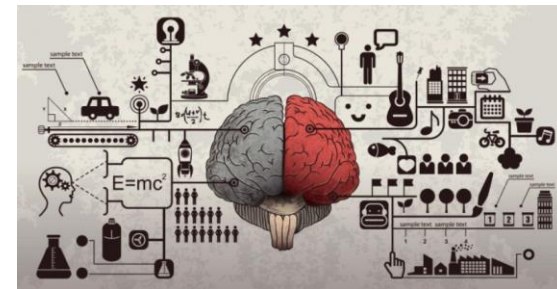
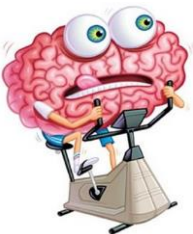




EXERCÍCIO



- 9) Mostre no terminal as possíveis opções do comando "mv" através de um comando
- 10) Em seu diretório HOME, crie um novo arquivo com o nome "arq1.txt", e insira algumas linhas nesse arquivo
- 11) Com esse novo comando aprendido (mv), renomeie o arquivo "arq1.txt" para "arquivo.txt"



- **rmdir** (remove directory):
 - Remove um ou mais diretórios vazios
- Exemplo

Vá para o diretório Home e crie dois diretórios:

```
mkdir apaga1
```

```
mkdir apaga2
```

Apague os diretórios criados:

```
rmdir apaga1
```

```
rmdir apaga2
```

- **head**

- Exibe as primeiras linhas de um ou mais arquivos

Sintaxe: **head [opções] <arquivo>**

Exemplo:

Para listar as cinco primeiras linhas do arquivo `/etc/group`

`head -5 /etc/group`

onde: -n número de linhas a serem listadas

- **tail**

- Exibe as últimas linhas de um ou mais arquivos

Sintaxe: **tail [opções] <arquivo>**

Exemplo:

Para listar últimas cinco linhas do arquivo /etc/group

```
tail -5 /etc/group
```

onde: -n número de linhas a serem listadas

*Obs: Comando muito utilizado para ver arquivos de logs.

- **stat**

- Usado para descobrir os atributos estendidos de um arquivo ou diretório

Sintaxe: **stat** *arquivo/diretório*

Exemplos:

`stat /etc/hosts.deny`

```
[oracle@oralex76 ~]$ stat /etc/hosts.deny
  File: "/etc/hosts.deny"
  Size: 460          Blocks: 8          IO Block: 4096   arquivo comum
Device: fc00h/64512d Inode: 781845       Links: 1
Access: (0644/-rw-r--r--)  Uid: (   0/   root)   Gid: (   0/   root)
Context: system_u:object_r:net_conf_t:s0
Access: 2019-09-20 11:31:31.282097117 -0300
Modify: 2013-06-07 11:31:32.000000000 -0300
Change: 2019-06-11 15:44:23.116000000 -0300
 Birth: -
[oracle@oralex76 ~]$
```

- **stat**

1. touch altera.txt
2. gedit altera.txt (inclua duas linhas no arquivo e salve)
3. stat altera.txt
4. cat altera.txt
5. stat altera.txt
6. rm altera.txt

- **file**
 - Usado para descobrir o tipo um arquivo

Sintaxe: **file** *arquivo*

Exemplos:

`file /etc/passwd`

`file /usr/lib/jvm/jre-1.8.0/lib/rt.jar`

- **find**

- Usado para pesquisar arquivos em uma hierarquia de diretórios

Exemplos:

```
cd /usr
```

```
find . -name "file*"
```

```
find . -iname "file*" (ignora o case sensitive)
```

```
find . -name "file*" -type f (somente arquivos (f=file))
```

```
find . -name "file*" -type d (somente diretórios (d=directory))
```

- **which**
 - Usado para descobrir onde os executáveis mencionados estão armazenados no PATH do usuário

Exemplos:

`which java`

`which rm`

- **free**
 - O comando free apresenta a utilização da memória do sistema
 - Mem: utilização da memória física
 - Swap: utilização da área de swap do sistema
- Exemplo
 - free** (padrão kilobytes)

free -m: Os valores são exibidos em Megabytes
free -g: Os valores são exibidos em Gigabytes

```
[oracle@linux76 ~]$ free
              total        used          free      shared  buff/cache   available
Mem:      3766172    770964    1398896        73328    1596312    2605204
Swap:      1257468           0    1257468
[oracle@linux76 ~]$
```

- **top**
 - Utilização da CPU, processos e memória

- Exemplo

top

ctrl + c para sair

```
14:06:32 up 4 days, 21:20, 4 users, load average: 0.00, 0.00, 0.00
77 processes: 76 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  cpu      user      nice      system    irq      softirq  iowait    idle
              total    19.6%     0.0%      0.0%      0.0%      0.0%      0.0%     180.2%
              cpu00    0.0%      0.0%      0.0%      0.0%      0.0%      0.0%     100.0%
              cpu01    19.6%     0.0%      0.0%      0.0%      0.0%      0.0%      80.3%
Mem:  1028548k av, 716604k used, 311944k free, 0k shrd, 131056k buff
      324996k actv, 108692k in_d, 13988k in_c
Swap: 1020116k av, 5276k used, 1014840k free 382228k cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT  %CPU %MEM    TIME CPU  COMMAND
 17578 root        15   0 13456 13M  9020 S    18.5  1.3   26:35  1  rhn-applet-gu
 19154 root        20   0  1176 1176   892 R     0.9  0.1    0:00  1  top
      1 root        15   0   168  160   108 S     0.0  0.0    0:09  0  init
      2 root        RT   0    0    0     0 SW     0.0  0.0    0:00  0  migration/0
      3 root        RT   0    0    0     0 SW     0.0  0.0    0:00  1  migration/1
      4 root        15   0    0    0     0 SW     0.0  0.0    0:00  0  keventd
      5 root        34  19    0    0     0 SWN    0.0  0.0    0:00  0  ksoftirqd/0
      6 root        35  19    0    0     0 SWN    0.0  0.0    0:00  1  ksoftirqd/1
      9 root        15   0    0    0     0 SW     0.0  0.0    0:07  1  bdflush
      7 root        15   0    0    0     0 SW     0.0  0.0    1:19  0  kswapd
      8 root        15   0    0    0     0 SW     0.0  0.0    0:14  1  kscand
     10 root        15   0    0    0     0 SW     0.0  0.0    0:03  1  kupdated
     11 root        25   0    0    0     0 SW     0.0  0.0    0:00  0  mdrecoveryd
```

top -u oracle

- **watch**
 - Reexecuta um comando periodicamente. CTRL+C interrompe a execução
- Exemplo
`watch free`

Por padrão o tempo de execução é a cada 2 segundos. Para alterar esse valor utilize o parâmetro **-n**

```
watch -n 5 free
```

- **ps**

O padrão é mostrar somente os processos no shell atual

Adicione a opção **a** para visualizar todos os processos

Para exibir o usuário associado a um processo, inclua a opção **u**

ps au

```
[student@serverX ~]$ ps au
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	428	0.0	0.7	152768	14400	tty1	Ss+	Feb03	0:04	/usr/bin/Xorg
root	511	0.0	0.0	110012	812	ttyS0	Ss+	Feb03	0:00	/sbin/agetty

O comando **df** (*disk filesystem*) mostra o espaço livre e ocupado de cada partição existente

Pode ser utilizado junto com várias opções, se for utilizado sozinho, mostrará o espaço usado e disponível de todos os sistemas de arquivos atualmente montados

Exemplo:

```
df -a
```

```
df -Th
```

Opções:

- help** exibe as opções do comando
- a** inclui sistema de arquivos com 0 blocos
- h** mostra o espaço livre/ocupado em MB, KB, GB em vez de bloco
- k** lista em Kbytes
- l** somente lista sistema de arquivos locais
- m** lista em Mbytes
- T** lista o tipo de sistema de arquivos de cada partição

I Incluir um usuário para utilizar o comando SUDO

O **sudo** permite que um usuário permitido execute um comando como outro usuário, de acordo com as especificações no arquivo **/etc/sudoers**

Incluir um usuário no SUDOERS

su - <enter>

<digite a senha do ROOT>

gedit /etc/sudoers

Inclua a seguinte linha:

oracle ALL=(ALL) ALL

Conforme a figura abaixo

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
oracle  ALL=(ALL)        ALL
```


- **grep**
 - Procura por linhas que contém um padrão de busca (expressão regular simples ou estendida)

Sintaxe: **grep [opções] padrão arquivos**

Exemplo:

Para procurar a palavra 'root' no arquivo /etc/passwd

```
grep root /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
operator:x:11:0:operator:/root:/sbin/nologin
```

- Dentro do diretório "/etc", procure todos os arquivos com extensão ".conf" que possuam o texto "limit" :

```
grep limit /etc/*.conf
```

- Procurar o texto "localhost" em todos os arquivos e subdiretórios de "/usr/bin"

```
grep -R localhost /usr/bin
```

- **Âncoras**

Meta	Mnemônico	Função
^	Circunflexo	Início da linha
\$	Cifrão	Fim da linha

Para procurar a palavra 'root' no arquivo /etc/passwd:

```
grep root /etc/passwd
```

Para procurar a palavra 'root' no início da linha, no arquivo /etc/passwd:

```
grep ^root /etc/passwd
```

Para procurar a palavra 'bash' no final da linha, no arquivo /etc/passwd:

```
grep bash$ /etc/passwd
```

- **Representantes**

Meta	Mnemônico	Função
.	Ponto	Um caractere qualquer
[...]	Lista	Lista de caracteres permitidos
[^...]	Lista negada	Lista de caracteres proibidos

Preparar arquivos para a atividade:

```
cd ~
```

```
gedit teste.txt
```

Digite:

sonha

sonho

sonhe

soninho

-> salve o arquivo

- Procurar no arquivo "teste.txt" por todas as linhas que possuem uma palavra "sonh" independente do final

```
grep sonh. teste.txt
```

- Procurar no arquivo "teste.txt" por todas as linhas que possuem uma palavra "son" independente do final

```
grep son. teste.txt
```

- Procurar no arquivo "teste.txt" por todas as linhas que possuem uma palavra "sonh" e que termine com "a" ou "o"

```
grep sonh[ao] teste.txt
```

Comando: | (pipe)

- O pipe é um das maneiras que o Linux pode utilizar para a comunicação entre processos
- De uma maneira simples poderíamos dizer que o pipe nada mais é do que o encadeamento de processos
- Dessa forma é possível enviar a saída de um comando para a entrada de outro



Exemplos:

```
cat /etc/passwd | more
```

O comando "cat" lista o conteúdo do arquivo "passwd" oferecendo juntamente a opção de paginação

```
ls -l /etc | grep alias | sort -d
```

1 - O comando "ls -l" lista o conteúdo do diretório "/etc" em formato longo

2 - A seguir o comando pipe faz com que a saída do primeiro comando entre no segundo comando: "grep alias", que procura na listagem do diretório qualquer arquivo que possua a palavra: "alias"

3 - Por final, a saída do comando "grep alias" é utilizada no comando "sort -d" que faz a classificação por data

- Use o comando **ls -l** para exibir permissões:

```
ls -l teste.txt
```

r	permissão para leitura (read)
w	permissão para escrever (write)
x	permissão para executar (execute)
-	substitui r, w, x se a permissão é negada



- As permissões são divididas em quatro partes para indicar: tipo, proprietário, grupo e outras permissões

Temos 10 caracteres definindo as permissões:

O tipo do objeto é representado por meio do primeiro caractere e as permissões são listadas por meio dos 9 caracteres na sequência

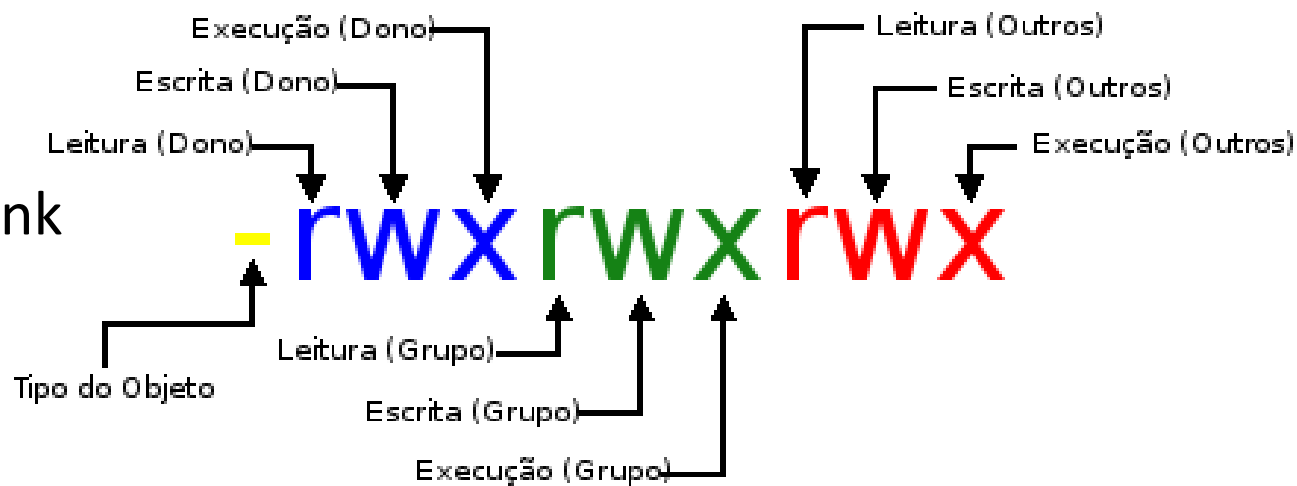
O primeiro caractere indica o tipo do arquivo:

Os mais comuns são:

d = Diretório

l = Indica que é um link

- = Arquivo comum



- Em seguida temos 3 grupos formados por 3 letras (que definem os privilégios): r, w, x

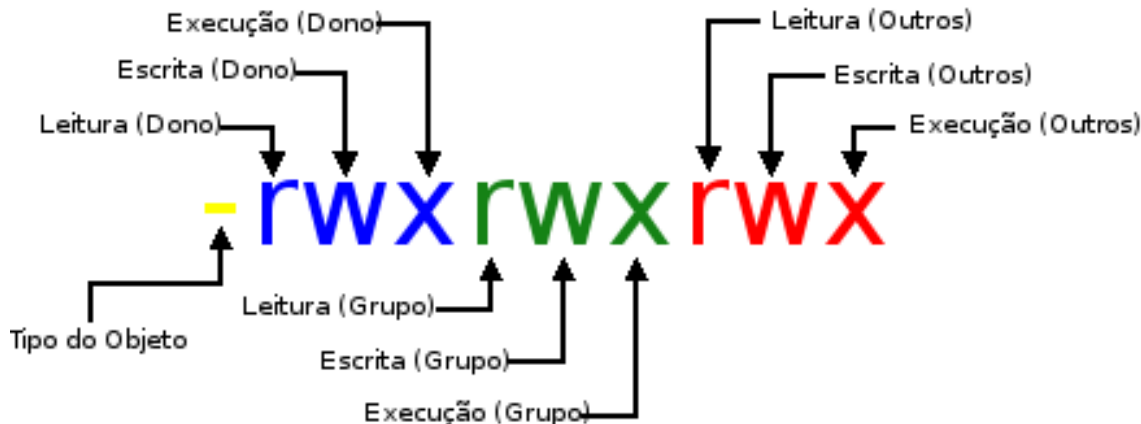
r = Read (Permitido ler o arquivo ou diretório)

w = Write (Permitido editar um arquivo ou modificar o conteúdo de um diretório)

x = Execute (Permitido executar um arquivo ou acessar um diretório)

- (hífen) = Não é permitido a escrita, leitura ou execução

- O Primeiro grupo de 3 letras se referem ao dono do arquivo (o usuário que o criou)
- As próximas 3 letras se referem ao grupo do usuário
- As últimas 3 letras aos demais usuários do computador ou da rede



```
ls -l teste1
```

```
-rwxr-xr-- 1 usuario_1 grupo_1 227 Jun 8 21:22 teste1
```

-rwxr-xr--

No exemplo acima **teste1** é um arquivo (-) que pertence ao **usuario_1** do **grupo_1**

- ✓ O **dono** pode ler (r), escrever (w) e executar (x)
- ✓ **Usuários do grupo** podem ler e executar (r-x) mas não modificar o arquivo
- ✓ **Os demais usuários** podem apenas ler (r--)

Comando chmod

Altera a permissões de acesso aos arquivos e diretórios (mode of access)

Sintaxe:

chmod [opções] [permissões] [diretório/arquivo]

Ha duas formas de usar o comando chmod:

- 1 - Modo literal ou simbólico (atribuição direta)
- 2 - Modo Octal

Modo literal ou simbólico (atribuição direta)

Opções:

u = Representa o dono (user)

g = Representa o grupo (group)

o = Representa os demais usuários (others)

a = Representa todos os utilizadores (all)

Permissões:

r = Leitura (read)

w = Gravação (write)

x = Execução (Execution)

Operações:

"+" Acrescenta uma permissão

"-" Retira uma permissão

"=" Define a permissão exatamente como especificado

Para preparar o exercício, execute os comandos:

```
touch arq1  
chmod 700 arq1  
ls -l arq1
```

01 - Acrescentar permissão de escrita ao grupo para o arquivo arq1:

`chmod g+w arq1`

02 - Retirar a permissão de escrita para o grupo: `chmod g-w arq1`

03 - Acrescentar a permissão de execução a todos os usuários: `chmod a+x arq1`

04 - Demais usuários não tenham permissão de leitura, mas tenham permissão de escrita e execução: `chmod o=wx arq1`

05 - Adicionar permissão de gravação no arquivo arq1 para o usuário dono: `chmod u+w arq1`

06 - Permissões de leitura e execução ao seu grupo: `chmod g+rx arq1`

07 - O arquivo arq1 deverá estar com todas as permissões disponíveis para o grupo: `chmod g=rwx arq1`

08 - Permissão de execução adicionado a todos os usuários para o arquivo teste1: `chmod a+x arq1`

09 - Permissão de execução adicionada ao proprietário, e somente leitura e gravação ao grupo: `chmod u+x,g=rw arq1`

10 - Permissão do proprietário e grupo para somente leitura e escrita: `chmod ug=rw arq1`

Exercícios:

01 - Crie um arquivo com o nome "arquivo1"

02 - Conceda somente o acesso de leitura nesse novo arquivo para os outros usuários

03 - Adicione o privilégio de execução a esse arquivo para os outros usuários

04 - Adicione o privilégio de execução a esse arquivo para o dono do arquivo

05 - Retire a permissão de escrita do grupo

06 - Remover a permissão de execução do usuário dono do arquivo, acrescentar as opções de escrita e execução para o grupo e deixar os demais usuários apenas com a permissão de execução

Modo Octal. Para usa-lo a seguinte tabela deverá ser utilizada:

0 => Nenhuma permissão de acesso

4 => Permissão de leitura (Read) -> r = 4

2 => Permissão de escrita (Write) -> w = 2

1 => Permissão de execução (Execute) -> x = 1

A partir disso, podemos obter qualquer permissão, somando os números correspondentes as permissões desejadas

3 => Permissão de execução e escrita (1 + 2)

5 => Permissão de execução e leitura (1 + 4)

6 => Permissão de escrita e leitura (2 + 4)

7 => Todas as permissões: execução, escrita e leitura (1 + 2 + 4)

Permissão	Binário	Decimal
---	000	0
--X	001	1
-W-	010	2
-WX	011	3
r--	100	4
r-X	101	5
rW-	110	6
rWX	111	7

Com esses algarismos, construímos números com três dígitos:

XYZ, onde:

X representa a permissão que será definida para o dono

Y a permissão do grupo

Z é a permissão para outros usuários

Exemplo:

`chmod 762 exemplo.txt` ou `chmod 762 /home/cursos/exemplo.txt`

- Estamos concedendo permissão 7 ao dono do arquivo exemplo.txt. Isso significa que estamos dando permissão de leitura, escrita e execução ao dono do arquivo
- Para o grupo, damos permissão 6 (escrita e leitura)
- Aos demais, damos apenas permissão de escrita (permissão 2)

Exercícios:

01 - Crie um arquivo com o nome "arquivo2"

`touch arquivo2`

02 - Conceda a permissão de leitura para o dono, de escrita para o grupo e a de execução para os demais usuários

`chmod 421 arquivo2` -> Equivale ao comando: `chmod u=r,g=w,o=x arquivo2`

03 - O dono ter todas as permissões no arquivo, o grupo ter as permissões de leitura e escrita e os demais usuários não tenham nenhuma permissão

`chmod 760 arquivo2`

04 - Deixe o arquivo apenas para leitura do usuário proprietário

```
chmod 400 arquivo2
```

05 - Deixe o arquivo como leitura para o usuário proprietário, grupo e outros usuários

```
chmod 444 arquivo2
```

06 - Permissão total no arquivo para o usuário proprietário, grupo e outros

```
chmod 777 arquivo2
```

chown

Muda o dono de arquivos, diretórios e links

- Sintaxe

chown [opções] [novo_proprietário] [:novo_grupo] nomes_arquivos

- Onde :

novo_proprietário: nome do novo usuário

novo_grupo: nome do novo grupo

nomes_arquivos: nome(s) do(s) arquivo(s)

* Esse comando precisa ser executado com o ROOT

Exemplos (entrar com o ROOT antes):

Alterando o proprietário do arquivo arquivo2 para um usuário chamado adm (já existe no sistema):

```
chown adm arquivo2
```

Alterando o usuário do arquivo arquivo2 para oracle e seu grupo para games ao mesmo tempo:

```
chown oracle:games arquivo2
```

Se apenas o grupo deve ser mudado, então é só deixar o campo do usuário em branco. Por exemplo:

```
chown :mail arquivo2
```

chgrp

Altera o grupo de arquivos

- Sintaxe

chgrp [opções] novo_grupo arquivos

- Onde :

novo_grupo: nome do novo grupo

arquivos: nome(s) do(s) arquivo(s)

Exemplo:

Alterando o grupo do arquivo arquivo2 para o grupo oracle:

chgrp oracle arquivo2

* Esse comando precisa ser executado com o ROOT

01 - Com o root crie um arquivo com o nome: lista.txt

02 - Verifique o usuário e o grupo do arquivo criado

03 - Altere o proprietário do arquivo lista.txt para o usuário: oracle

04 - Verifique a situação atual do usuário e do grupo do arquivo

05 - Altere o grupo do arquivo lista.txt para o grupo: oracle

01 - Com o root crie um arquivo com o nome: lista.txt

```
touch lista.txt
```

02 - Verifique o usuário e o grupo do arquivo criado

```
ls -l lista.txt
```

03 - Altere o proprietário do arquivo lista.txt para o usuário: oracle

```
chown oracle lista.txt
```

04 - Verifique a situação atual do usuário e do grupo do arquivo

```
ls -l lista.txt
```

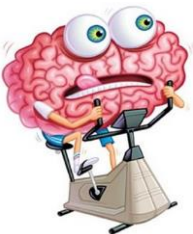
05 - Altere o grupo do arquivo lista.txt para o grupo: oracle

```
chgrp oracle lista.txt    ou    chown :oracle lista.txt
```



DESAFIO FINAL

- Dentro de sua máquina virtual, chegou a hora de instalar o banco de dados MySQL8, um dos mais renomados SGBDR
- Para realizar essa tarefa com sucesso, peça ajuda ao professor e utilize como guia o arquivo **HowTo_Instalacao_MySQL80_RH7.docx**
- Bom desafio!



Copyright © 2020 Prof. João Menk e Prof. Salvio Padlipskas

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).