

FIAP GRADUAÇÃO

TECNOLOGIA EM DESENVOLVIMENTO DE SISTEMAS

DevOps Tools & Cloud Computing
Virtualização: Comandos Básicos Linux

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

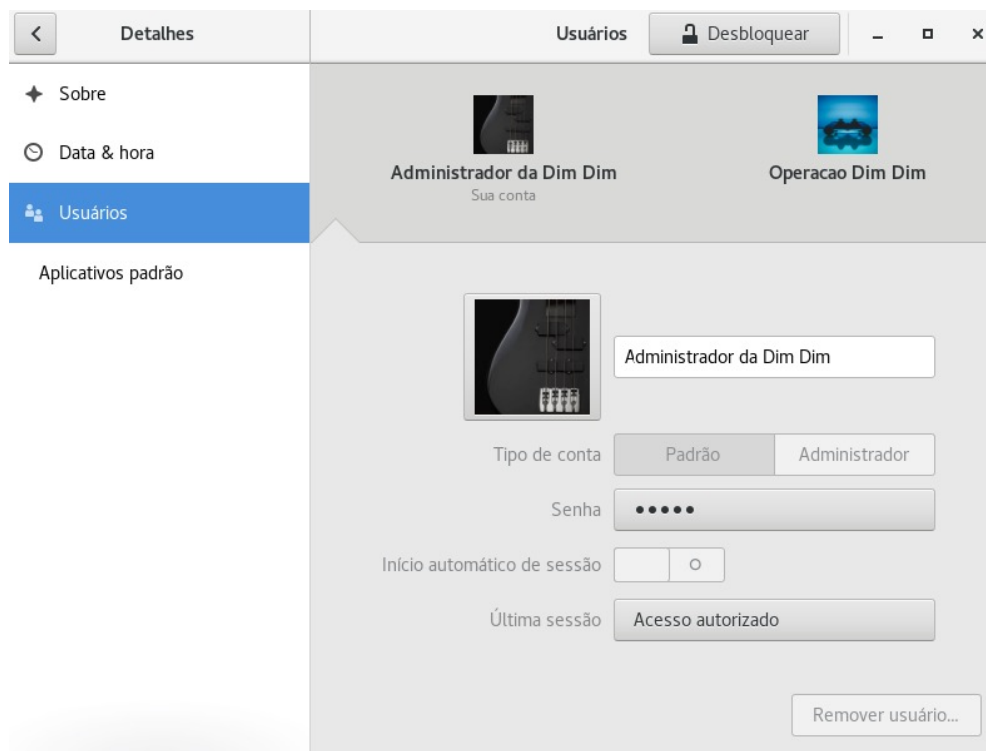
PROF. SALVIO PADLIPSKAS salvio@fiap.com.br

| Criação de um novo usuário

Vamos criar um novo usuário para podermos praticar os comandos de forma uniforme entre todos

Crie um usuário com o nome de **Login: admdimdim** e **Senha: admPa\$\$wOrd2021**

Após a criação desse usuário, realize um logoff e entre com o login criado



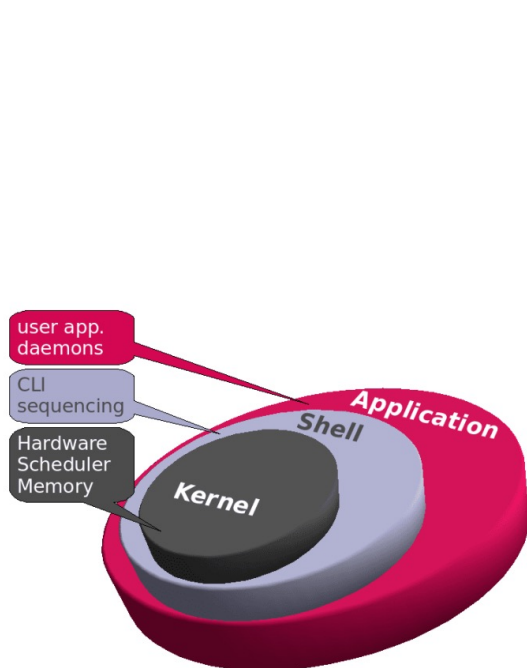
- Foi desenvolvido pelo finlandês Linus Torvalds, inspirado no sistema Minix
- O nome Linux surgiu da mistura de Linus + Unix
- Inicialmente desenvolvido e utilizado por grupos de entusiastas em computadores pessoais, o sistema Linux passou a ter a colaboração de grandes empresas, como a IBM, a Sun Microsystems,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

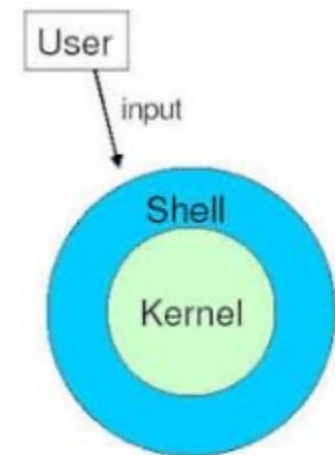
Distribuições Linux

- 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



Exibe a distribuição Linux RH76

```
cat /etc/os-release
```



Linux: Estrutura de Diretórios



/bin	binários essenciais do usuário
/boot	arquivos estáticos de boot
/dev	arquivos de dispositivos (devices)
/etc	arquivos de configuração não específicos
/home	pastas dos usuários comuns do sistema
/media	ponto de montagem temporário para mídias removíveis
/mnt	ponto de montagem temporário para sistemas de arquivos montados
/opt	softwares adicionais (adicionados pelo usuário)
/sbin	binários do sistema
/srv	dados para serviços providos pelo sistema
/tmp	arquivos temporários
/usr	multi-usuário utilitários e aplicações
/var	arquivos variáveis (conteúdo dinâmico)
/root	home do superusuário (root)
/proc	sistema de arquivos virtual, documentos do kernel e status de processos como arquivo de texto

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

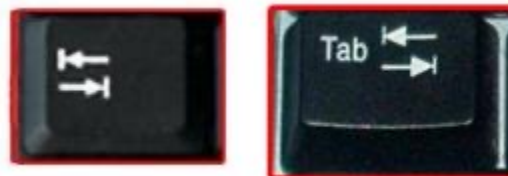
Um órgão que define os principais diretórios e o seu conteúdo em sistema Linux

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

- Mostrando data e hora

Para mostrar a data e a hora corrente, usamos o comando: **date**

Calendário: **cal**

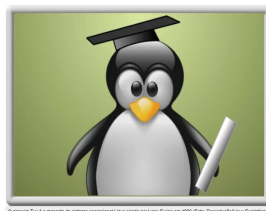
Exemplo: **cal 01 2021** (*exibe o mês de janeiro de 2021)

- Limpando a tela

Para limpar a tela, usamos o comando: **clear** (**ctrl+l**)

Comandos Básicos Linux: Diretórios e Subdiretórios

FIAP



- Dentro do novo desenho da arquitetura de software da DimDim, será necessário receber e enviar centenas de arquivos nos mais diversos formatos. E para obter uma melhor organização, precisamos armazenar esses arquivos em **diretórios** e **subdiretórios**.
- Nossa meta será explorar como realizar essa tarefa dentro do SO Oracle Linux RH76



Criando diretórios e subdiretórios com o comando **mkdir**

- Vamos criar 3 diretórios como exemplo:
 - **mkdir ex1**
 - **mkdir ex2**
 - **mkdir ex3**

- Mudando de diretório
 - Mude de diretório com o comando: **cd**
 - Vamos mudar para o diretório que acabamos de criar: **cd ex1**
 - Crie um subdiretório dentro do diretório **ex1** chamado **subex1**

- Comandos de diretório
 - Mostre o diretório corrente com o comando: **pwd**

Removendo diretórios e subdiretórios com o comando **rmdir**

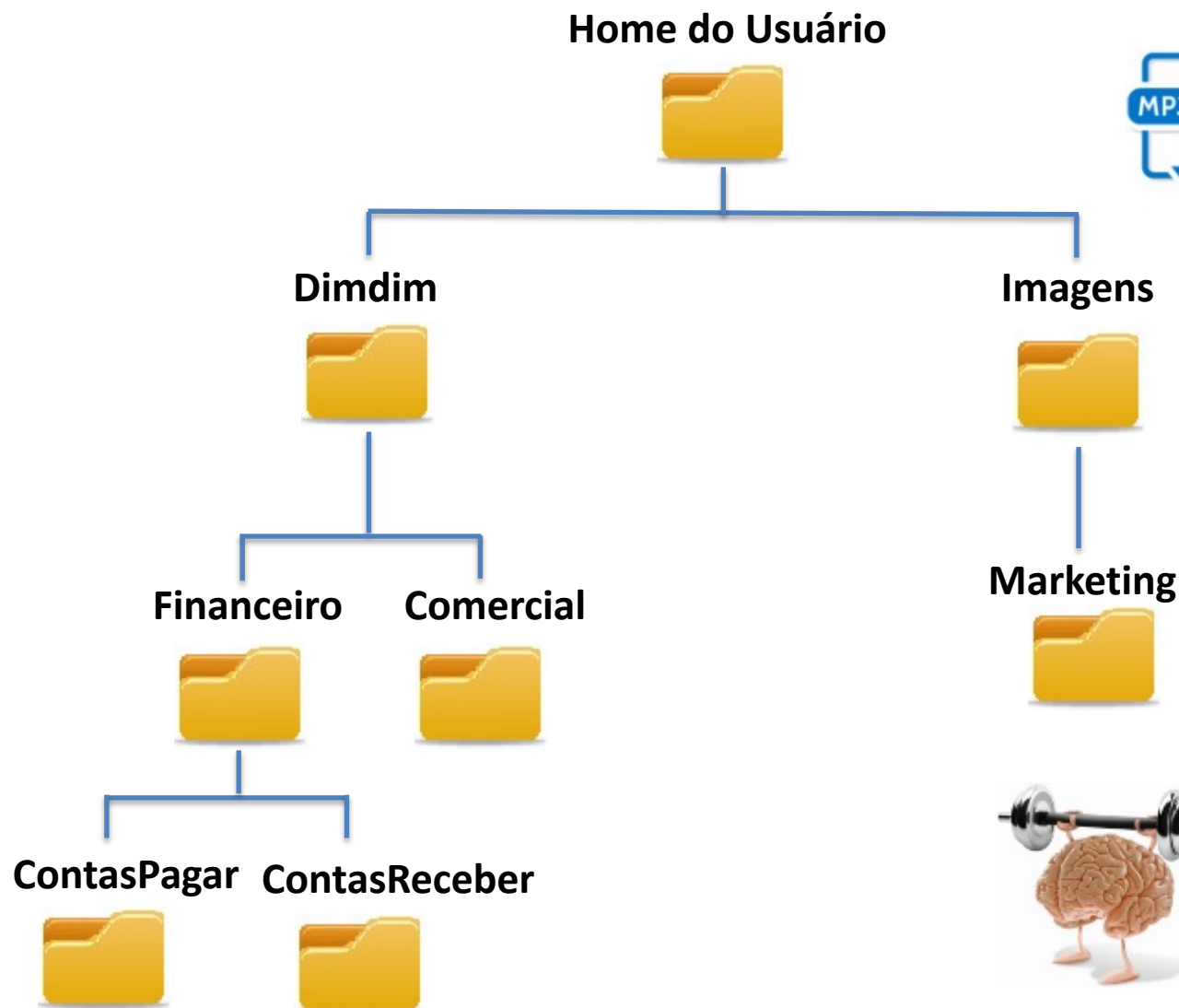
- Remove um ou mais diretórios vazios
- Elimine os diretórios abaixo:
 - **rmdir ex2**
 - **rmdir ex3**
- Mudando de diretório
Mude de diretório com o comando: **cd**
Vamos mudar para o diretório que acabamos de criar: **cd ex1**
Elimine o subdiretório dentro do diretório **ex1** chamado **subex1**
- Verifique pelo comando **ls** se os diretórios foram eliminados

Exercício Prático Comandos Linux



FIAP

- 1) Crie a seguinte estrutura de diretórios no Servidor Linux
- 2) Transfira os arquivos conforme orientação do professor



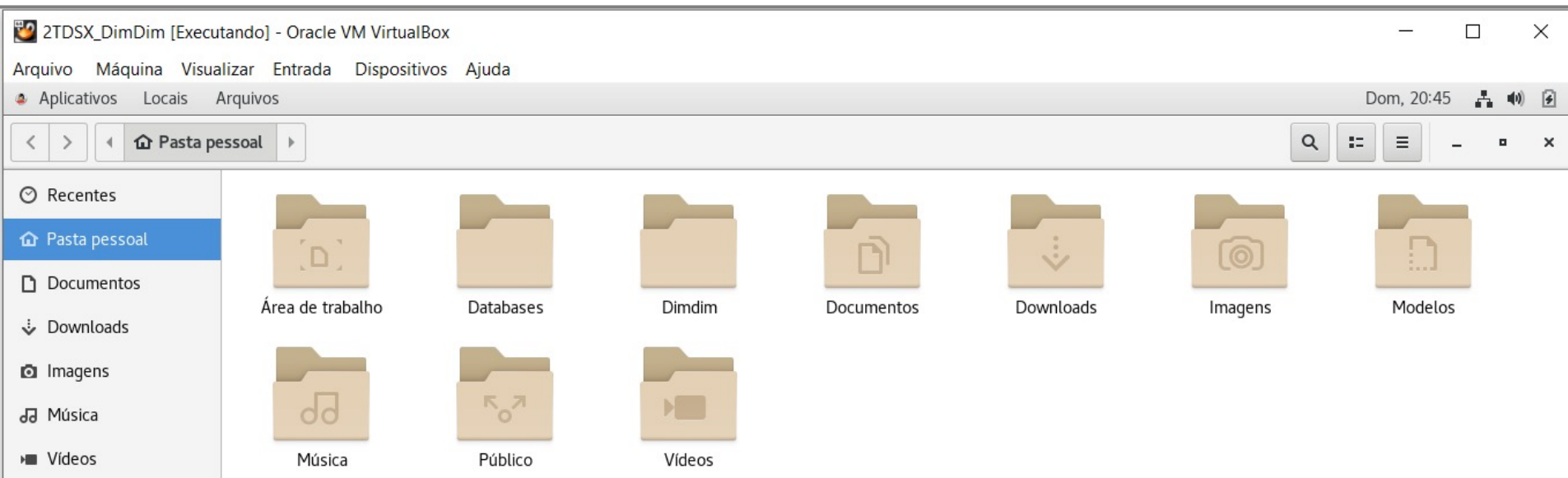
- Comercial01.jpg
- Comercial02.xlsx
- ContasPagar01.pdf
- ContasPagar02.mp3
- ContasPagar03.mp3
- ContasReceber01.mp3
- ContasReceber02.txt
- Marketing01.JPG
- Marketing02.jpg





Na Área de Trabalho clique duas vezes no Ícone **Pasta pessoal**

Navegue visualmente nas pastas criadas



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

Ex.: `ls --help`

- 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]*.*`



Listar Arquivos e Diretórios

- 1) Identifique a localização exata por meio do comando **ls** dos arquivos do projeto DimDim
- 2) Tente identificar a data e horário de carga

-  Comercial01.jpg
-  Comercial02.xlsx
-  ContasPagar01.pdf
-  ContasPagar02.mp3
-  ContasPagar03.mp3
-  ContasReceber01.mp3
-  ContasReceber02.txt
-  Marketing01.JPG
-  Marketing02.jpg



Comandos Básicos Linux: Acesso e mudança de diretório

Entre no diretório `home/admdimdim/Imagens` `cd /home/admdimdim/Imagens`

Vamos voltar um nível anterior, para o diretório `home/admdimdim` `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/admdimdim`

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`

- Mudando de diretório

Mude de diretório com o comando: **cd**

Vamos mudar para o diretório que acabamos de criar: **cd Dimdim**

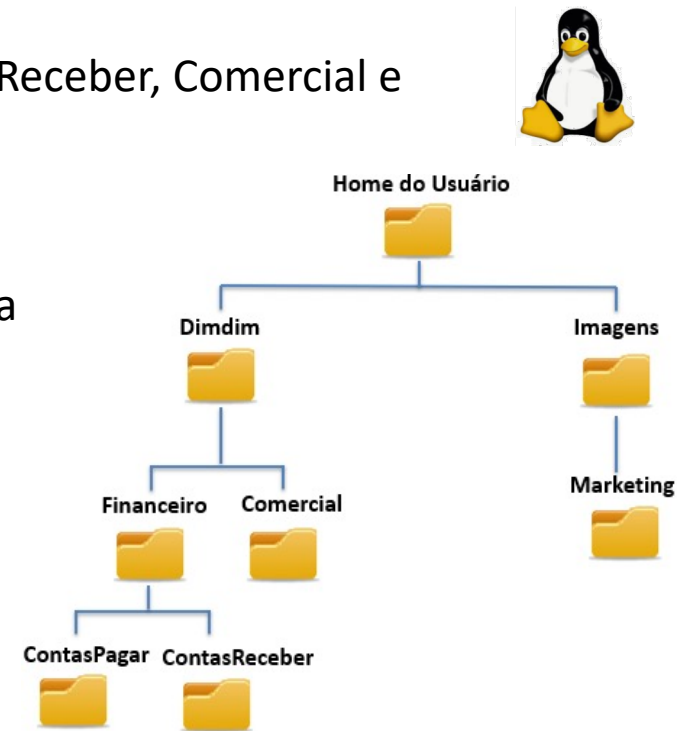
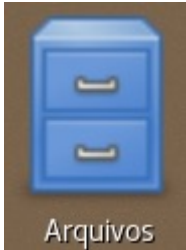
- Comandos de diretório

Mostre o diretório corrente com o comando: **pwd**

Exercício Prático: Acesso e mudança de diretório

1. Faça a mudança para os diretórios: ContasPagar, ContasReceber, Comercial e Marketing

2. Tente exibir alguns dos arquivos que se encontram nessa
(*caso queira utilizar a parte gráfica o linux, fique a vontade)



- Para exibir imagens em linha de comando: `eog <nome imagem e extensão>`
- Para exibir músicas em linha de comando: `totem <nome imagem e extensão>`

Vamos criar alguns arquivos:

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

```
touch texto2
```

```
touch texto3
```

```
touch doc1
```

```
touch doc2
```

❖ Insira um texto dentro desses arquivos

```
gedit nomeArquivo.ext
```

Comandos Básicos Linux: Exibindo o conteúdo de arquivos



- 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

Comandos Básicos Linux: Criando um ALIAS

O ALIAS funciona como um apelido para um comando no Linux

alias seuAtalho='Comando --OPCOES'

Exemplos:

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

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

1. Criar um ALIAS para o comando rm: **alias rm='rm -i'** * será utilizado depois
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 realizar um “Refresh”:
source .bashrc

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

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

Exemplo: vamos criar um arquivo dentro de **/home/admdimdim** e copiar esse arquivo para dentro de um subdiretório:

1. Vá ao diretório: **/home/admdimdim**
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

Comandos Básicos Linux: Copiando arquivos e diretórios

■ Copiando o conteúdo de um diretório para outro

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

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

Observe que os subdiretórios de `/home/admdimdim` 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/admdimdim`

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

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

Sintaxe: **mv** (arquivo ou diretório origem) (arquivo ou diretório destino).

Exemplo: vamos criar um arquivo dentro de **/home/admdimdim** e copiar esse arquivo para dentro de um subdiretório:

1. Vá ao diretório: **/home/admdimdim**
2. Crie o arquivo exemplo1.txt e digite algumas informações nele
gedit exemplo1.txt
3. Crie o subdiretório dir1 e mova o arquivo criado para esse subdiretório
Crie o diretório com o comando **mkdir dir1**
mv exemplo1.txt dir1
4. Isso feito, insira o seguinte comando
ls -l dir1

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**

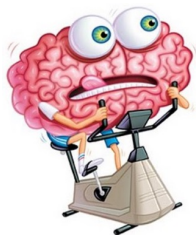
1. Vá ao diretório: `/home/admdimdim/ex2`
2. Apague os arquivos: `texto*` (confirmando) `rm texto*`
3. Apague os arquivos: `doc*` (sem solicitar confirmação) `rm -f doc*`
4. Apague o diretório: `ex1` (confirmando) `cd ~`
`rm -r ex1`
5. Apague o diretório: `ex2` (sem solicitar confirmação) `rm -rf ex2`

Exercício Prático: Manipulação de Arquivos



FIAP

- 1) Entre no seu diretório HOME
- 2) Crie um diretório com o nome `blog`
- 3) Entre no diretório criado
- 4) Crie dois arquivos: `arq1.txt` e `arq2.txt` com base no arquivo `Dimdim/Financeiro/ContasReceber/ContasReceber02.txt`
* Utilize o comando `cat`
- 5) Copie o arquivo `arq1.txt` para o diretório `Dimdim/Comercial`
- 6) Elimine o arquivo `arq2.txt`



Comandos Básicos Linux: Verificando Arquivos



- Vários comandos de visualização, informações e procura de arquivos são muito importantes para o dia a dia do Desenvolvedor que utiliza o Linux
- A DimDim precisa que você aprenda alguns desses comandos para ter mais desenvoltura na procura de informações sobre arquivos nesse Sistema Operacional



- **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

- **stat**

- Exibe os detalhes do arquivo a ser analisado. Última modificação, último acesso, contexto etc

Sintaxe: **stat** *arquivo/diretório*

Exemplos:

`stat /etc/hosts.deny`

`stat /home/admdimdim/Imagens/Marketing/Marketing01.JPG`

```
[oracle@oralinux76 ~]$ 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@oralinux76 ~]$
```

- **file**

- Usado para descobrir o tipo um arquivo

O comando file é ótimo para entender exatamente o que você tem em sua máquina

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

Permite ao usuário fazer buscas e localizar listas de arquivos/diretórios em uma única linha de comando

Exemplos:

```
cd /usr
```

```
find . -name *.txt
```

```
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

Procura o caminho para cada arquivo executável na Máquina

Exemplos:

`which java`

`which rm`

Comandos Básicos Linux: Monitoramento de Recursos



- Mais do que essencial, é saber como andam os recursos da Máquina Linux que está utilizando
- Para que você possa entender “o que se passa em sua máquina” vamos aprender alguns comandos utilizados para Monitorar os Recursos



- **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

Um dos comandos mais utilizados para monitorar o Linux

- 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  bdf flush
      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`

Pode ser utilizado para monitorar uma instalação, logs de saída etc

- **ps**

Obtem informações sobre o estado dos processos que estão sendo executados no momento

- ✓ 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

Comandos Básicos Linux: SUDO



- A DimDim precisa que o usuário **admdimdim** execute vários comandos que pertencem ao usuário **root** (Administrador)
- Para manter a segurança a senha do usuário **root** não pode ser revelada para qualquer pessoa que necessite executar alguma tarefa administrativa
- O Sudo é um comando utilizado no Sistema Operacional Linux, para que usuários comuns tenham privilégios de outro usuário e que consigam executar tarefas específicas dentro do Sistema
- Para executar essa solicitação sem expor a senha do **root**, vamos utilizar o commando **sudo** para incluir o usuário **admdimdim** em uma lista. Assim ele pode executar comandos como se fosse um Administrador (root), mesmo sendo um usuário padrão

| Comandos Básicos Linux: SUDO

As especificações dos usuários, privilégios etc está no arquivo **/etc/sudoers**

O “s” no comando significa “swich” e o “u” significa “user”, isto é, “switch user” (trocar de usuário)

Incluir um usuário no SUDOERS

su - <enter>

<digite a senha do ROOT>

gedit /etc/sudoers

Inclua a seguinte linha:

admdimdim ALL=(ALL) ALL

Conforme a figura abaixo

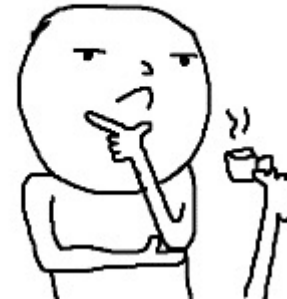
```
[vagrant@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[vagrant@localhost ~]$ sudo cat /etc/shadow
root:$1$QDyPlph/$oaAX/xNRf3aiW3l27NIUA/::0:99999:7:::
bin:!:17834:0:99999:7:::
daemon:!:17834:0:99999:7:::
adm:!:17834:0:99999:7:::
lp:!:17834:0:99999:7:::
nlp:!:17834:0:99999:7:::
```

```
## Allow root to run any commands anywhere
```

```
root      ALL=(ALL)      ALL
admdimdim ALL=(ALL)      ALL
```

- O comando **grep** é um comando com uma função simples: ele procura por trechos de texto (strings) dentro de arquivos ou diretórios e retorna para você em quais arquivos a string foi encontrada, inclusive mostrando a linha em que isso ocorreu
- Uma das ferramentas mais úteis e versáteis disponível
- Utilizado com frequência para realizar pesquisas dentro dos diversos arquivos de configuração de um sistema, em arquivos de instalação, logs etc

Interessante...



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
```

```
grep -R Ola
```

* Procura pelo texto “Ola” dentro dos arquivos (recursivamente dentro dos diretórios)

- 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
```

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

Comandos Básicos Linux: Permissões

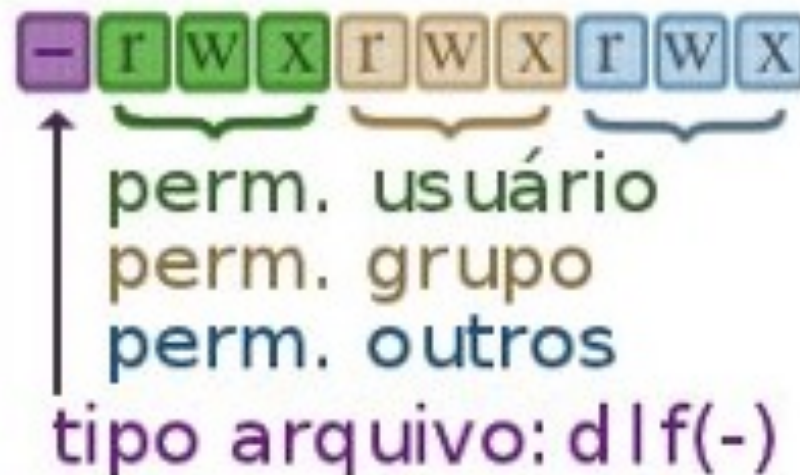


- Usuários finais, desenvolvedores e administradores utilizam o Linux como Sistema Operacional em seu dia a dia na DimDim
- Cada usuário tem um perfil de utilização, e para manter a proteção de dados precisamos trabalhar com Permissões de Usuários e Grupos com relação às pastas e arquivos existentes no computador, segurança é fundamental
- Para podermos ajudar nessa tarefa precisamos entender como funcionam essas permissões no Linux

- 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



Comandos Básicos Linux: Permissões

- 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

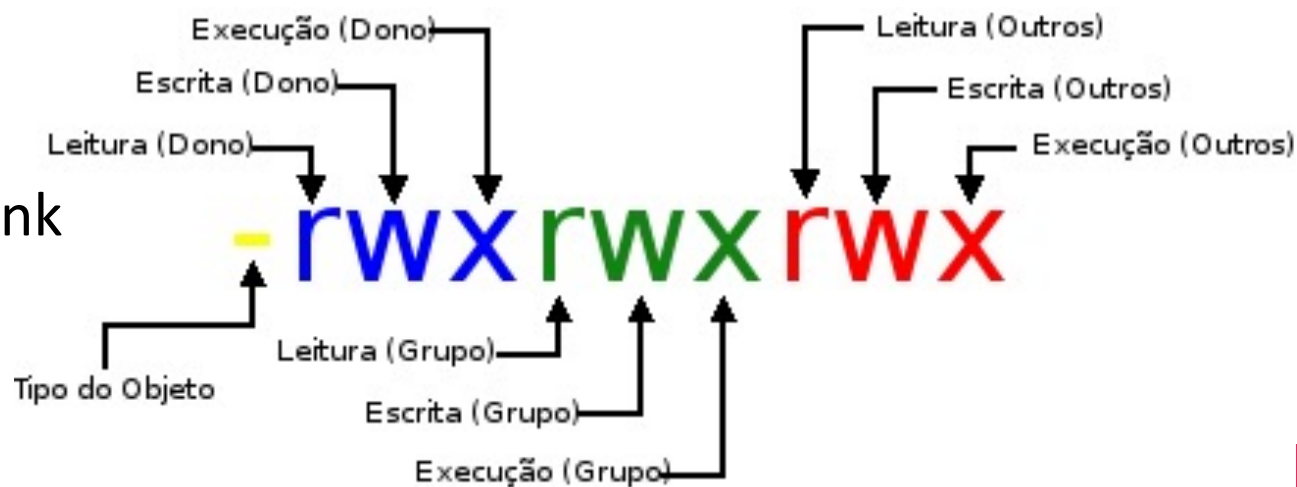
O primeiro caractere indica o tipo do arquivo:

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

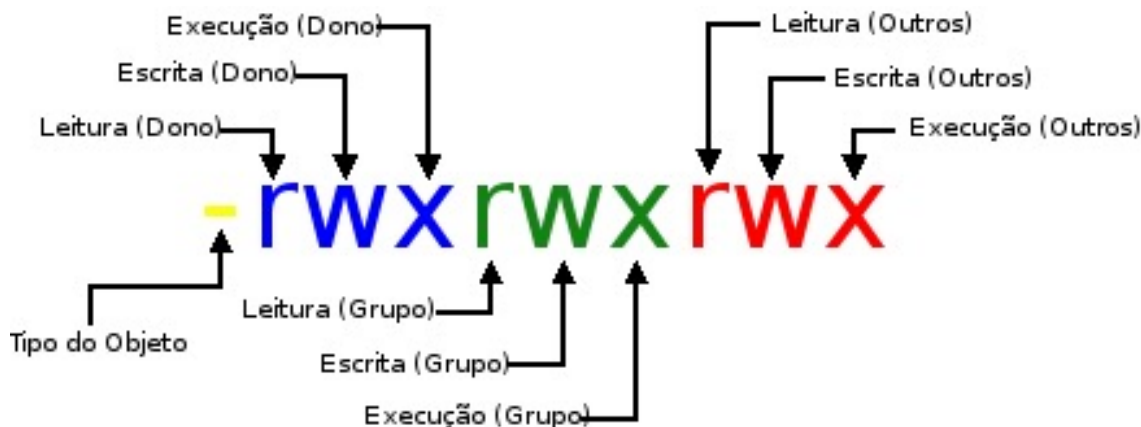
r = Read (Permitido ler o arquivo ou diretório)

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`

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/curso/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, demos permissão 6 (escrita e leitura)
- Aos demais, demos 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 **adm** e seu grupo para **games** ao mesmo tempo:

```
chown adm: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 games:

chgrp games arquivo2

* Esse comando precisa ser executado com o ROOT

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: adm

```
chown adm 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: games

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


Copyright © 2021 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).