

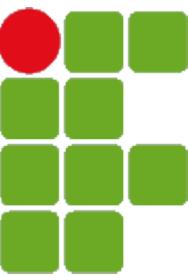
Instituto Federal de Santa Catarina
CST em Análise e Desenvolvimento de Sistemas
ICO7862 - Introdução à Computação

O Linux no modo Texto

Prof. Diego da Silva de Medeiros

São José, 21 de agosto de 2023

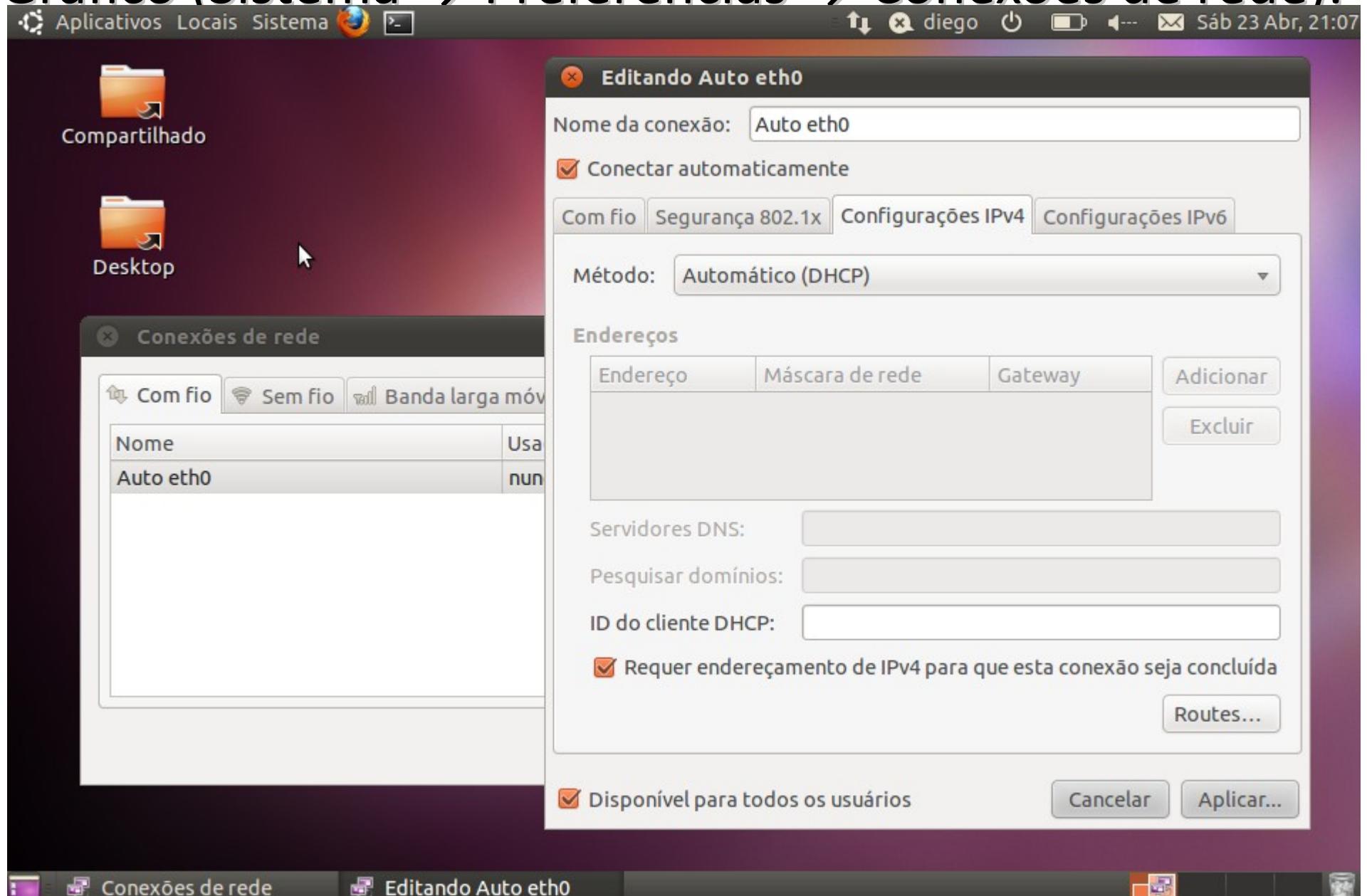
Vantagens



- Simplicidade
 - Configurações através de arquivo de configuração
- Estabilidade
- Baixo requerimento de recursos
- Automatização de processos com scripts
- Maior facilidade de elaboração de tutoriais
 - Maior quantidade de tutoriais na internet
- PODER!
 - Configuração no modo gráfico precisa ter disponível o parâmetro
 - Configuração no modo texto basta digitar o valor no arquivo .conf

Exemplo: configuração de rede

- Gráfico (Sistema → Preferências → Conexões de rede):



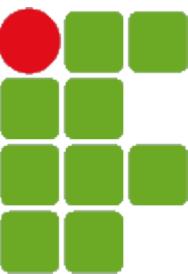
Exemplo: configuração de rede

- **Texto (Arquivo /etc/network/interfaces):**

```
# This file describes the network interfaces
# available on your system and how to activate them.
# For more information, see interfaces(5) .

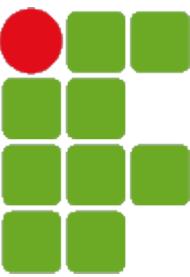
# The loopback network interface
auto lo
iface lo inet loopback
address 127.0.0.1
Netmask 255.0.0.0

# The Ethernet interface 1
iface eth1 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1
```



- **Programa** interpretador de comandos do Linux
- Analisa o texto digitado e produz um resultado
 - Texto é *case sensitive*
 - “cd”, “cD”, “Cd” e “CD” são coisas diferentes
- Exemplos de interpretadores: ksh, csh, bash, etc

Extensão de arquivos



- No Windows:
 - arquivo.txt → Bloco de notas
 - arquivo.bmp → Paint
 - arquivo.doc → Word
- No Linux, arquivos não precisam ter extensão
 - Informação do tipo do arquivo fica no cabeçalho
 - Extensões servem para informação ao usuário
 - Alguns ambientes gráficos ainda utilizam extensões

Extensão de arquivos

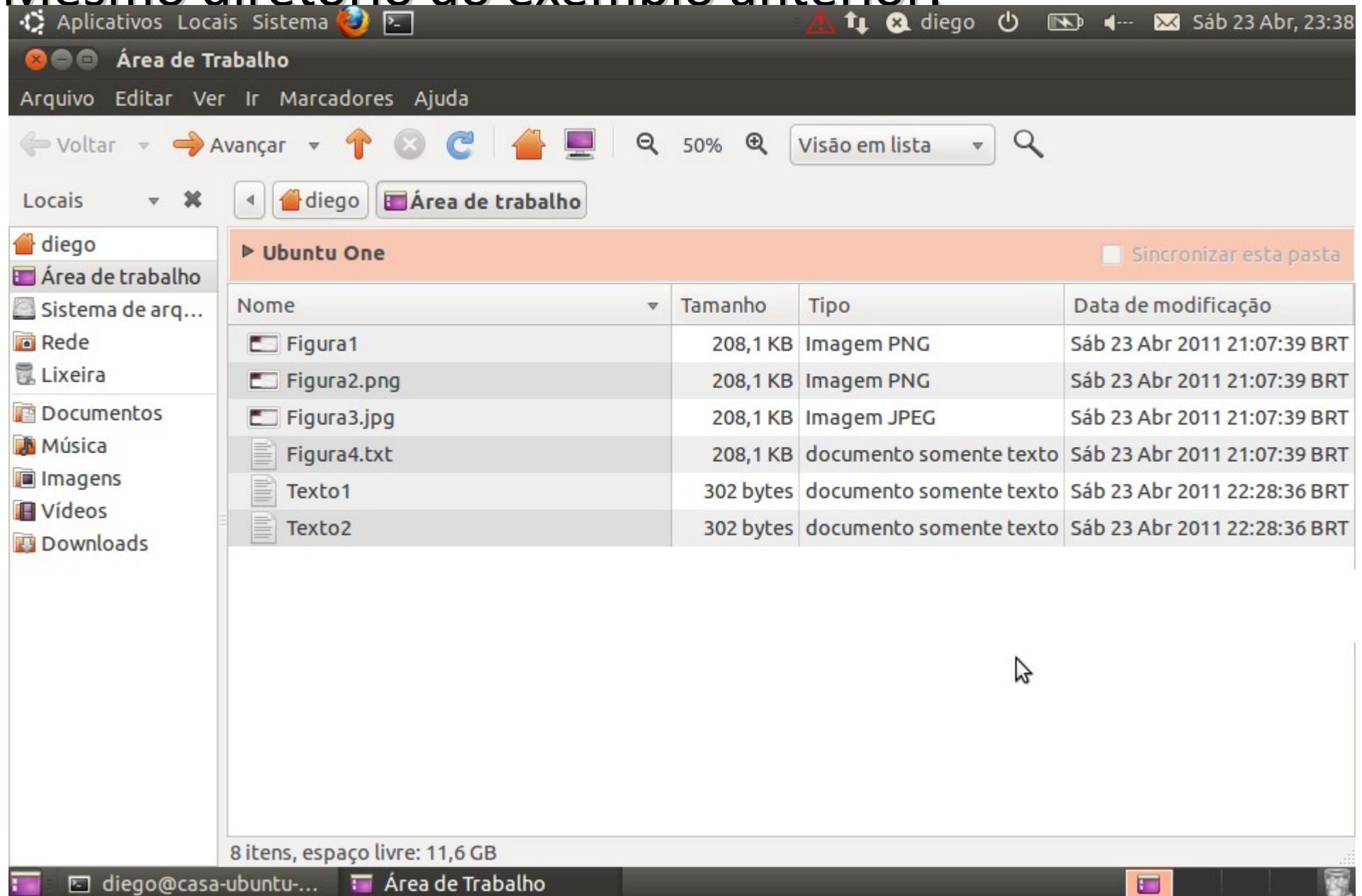
- Exemplo: o comando *file*

```
user@maquina:~$ ls -l
total 856
-rw-r--r-- 1 user user 213072 2011-04-23 21:07 Figura1
-rw-r--r-- 1 user user 213072 2011-04-23 21:07 Figura2.png
-rw-r--r-- 1 user user 213072 2011-04-23 21:07 Figura3.jpg
-rw-r--r-- 1 user user 213072 2011-04-23 21:07 Figura4.txt
-rw-r--r-- 1 user user      302 2011-04-23 22:28 Texto1
-rw-r--r-- 1 user user      302 2011-04-23 22:28 Texto2.mp3
```

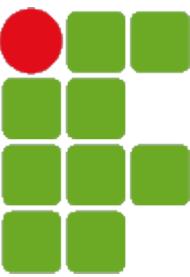
```
user@maquina:~$ file *
Figura1:          PNG image, 950x632, 8-bit/color RGB
Figura2.png:      PNG image, 950 x 632, 8-bit/color RGB
Figura3.jpg:      PNG image, 950 x 632, 8-bit/color RGB
Figura4.txt:      PNG image, 950 x 632, 8-bit/color RGB
Texto1:           ASCII text
Texto2:           ASCII text
```

Extensão de arquivos

- Mesmo diretório do exemplo anterior:



Arquivos ocultos

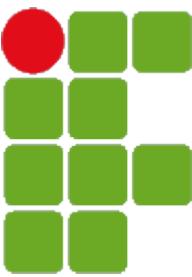


- Arquivo oculto começa com o caracter “.”
- Exemplo .bashrc, .cache, etc

```
user@maquina:~$ ls -l
total 216
-rw-r--r-- 1 user user 213072 2011-04-23 23:46 Figural
-rw-r--r-- 1 user user      302 2011-04-23 23:46 Texto1
```

```
user@maquina:~$ ls -la
total 864
drwxr-xr-x  2 user user    4096 2011-04-23 23:48 .
drwxr-xr-x 43 user user    4096 2011-04-23 23:46 ..
-rw-r--r--  1 user user 213072 2011-04-23 23:46 Figural
-rw-r--r--  1 user user 213072 2011-04-23 23:46 .Figura2.png
-rw-r--r--  1 user user 213072 2011-04-23 23:46 .Figura3.jpg
-rw-r--r--  1 user user 213072 2011-04-23 23:46 .Figura4.txt
-rw-r--r--  1 user user     302 2011-04-23 23:46 Texto1
-rw-r--r--  1 user user     302 2011-04-23 23:46 .Texto2
```

Tab completion



- Recurso que completa o texto digitado após um TAB
- O shell vê as opções disponíveis e completa o texto

```
user@maquina:~$ ls  
Pasta_com_nome_grande_e_dificil_de_digitar Musicas  
  
user@maquina:~$ cd Pas (TAB)  
user@maquina:~$ cd Pasta_com_nome_grande_e_dificil_de_digitar
```

Tab completion

- Tendo mais de um arquivo com nome parecido, o shell completa até o ponto onde os nomes ficam diferentes
- Outro TAB mostra todas as opções

```
user@maquina:~$ ls -l
total 856
drw-r--r-- 1 user user 213072 2011-04-23 21:07 Pasta1
drw-r--r-- 1 user user 213072 2011-04-23 21:07 Pasta2
drw-r--r-- 1 user user 213072 2011-04-23 21:07 Pasta3
drw-r--r-- 1 user user 213072 2011-04-23 21:07 Pasta4
-rw-r--r-- 1 user user      302 2011-04-23 22:28 Texto1
-rw-r--r-- 1 user user      302 2011-04-23 22:28 Texto2.mp3
```

```
user@maquina:~$ cd Pas (TAB)
user@maquina:~$ cd Pasta (TAB)
Pasta1 Pasta2 Pasta3 Pasta4
```

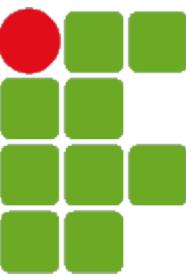
```
user@maquina:~$ cd Pasta
```

Terminals

- Ambiente semelhante à estações de trabalho
- Acesso através de **CTRL+ALT+Fn** → $n = 1,2,\dots,6$
- Além disso há o terminal virtual do modo gráfico
 - Aplicativos → Acessórios → Terminal



Login e Logout



- Processo de iniciar e encerrar uma sessão
- Terminal virtual:
 - login é realizado automaticamente
 - Usuário já logou-se no ambiente gráfico
- Num terminal real, é preciso passar usuário e senha
- Nos dois casos, para sair usa-se o comando *exit*

man: O comando mais importante!

- Manual do sistema, geralmente em inglês
- Geralmente, todas as dúvidas sobre um comando podem ser sanadas com a leitura do *man*
- Sintaxe: *man [página] <comando>*
- Controle:
 - Navegação pelo manual setas acima e abaixo
 - Sair do manual q
 - Primeira linha g
 - Última linha G
 - Procurar texto /texto

man: O comando mais importante!

- Seções mais importantes do manual:

- **NAME** – nome do comando
 - ls - list directory contents
- **SYNOPSIS** – expressão que resume todo o comando pesquisado
 - ls [OPTION]... [FILE]...
- **DESCRIPTION** – descrição detalhada do comando e da sua sinopse
 - -a, --all do not ignore entries starting with .
 - -A, --almost-all do not list implied . and ..
 - --author with -l, print the author of each file
- **NOTES** – alguns pontos importantes no comando
- **AUTHOR** – quem desenvolveu o programa
- **SEE ALSO** – comandos relacionados e mais informações

man: O comando mais importante!

- Notação utilizada pela sinopse:

- **texto_negrito** digitar exatamente como está escrito
- texto sublinhado substituir pelo argumento apropriado
- [-abc] argumentos a,b,c opcionais
- -a|-b argumentos a,b não podem ser usados juntos
- argumento ... argumento pode ser repetido
- [expressão] ... toda a expressão entre [] pode ser repetida

- Olhar o man dos comandos pwd e history

man: O comando mais importante!

- Comandos semelhantes:

- Comando --help → Exemplo: `ls --help`

- Comandos úteis para conhecer programas:

- apropos palavra Procura manuais do sistema que tenham a palavra
 - whatis comando Imprime uma descrição curta do comando

- E se tudo isso não for suficiente...

- Google → “Linux comando ação a ser feita”

O superusuário

- Usuário administrador do sistema
- Conhecido como usuário *Root*
- Acesso a todos os arquivos e configurações do sistema
- **CUIDADO! Com o usuário root você pode danificar irreversivelmente o sistema com apenas alguns comandos**
- Login: root Senha: root
- Comando para se logar como root: *su -*

O superusuário

- Em versões recentes do Ubuntu
 - Usuário root desabilitado por padrão
 - Para habilitar: *sudo passwd*
- O comando *sudo*:
 - Forma de executar um comando com um usuário diferente do root

```
user@maquina:~$ ls /root
ls: não foi possível abrir o diretório /root: Permissão negada
```

```
user@maquina:~$ sudo ls /root
Área de Trabalho  Desktop  Downloads  Modelos  Público
Compartilhado    Documentos  Imagens  Música  Vídeos
```

Comandos - PWD

pwd - lista o diretório atual

```
user@maquina:~$ pwd  
/home/user
```

Comandos - HISTORY

history [opções] - lista o histórico dos comandos utilizados

- **Opções interessantes:**

- -c apaga o histórico

```
user@maquina:~$ history
1 history
2 ls
3 cd Pasta4
4 cd ~
5 ifconfig
6 less Área\ de\ Trabalho/Pasta/arquivo.txt
7 history
```

Comandos - CLEAR

clear - apaga os dados presentes na tela

```
user@maquina:~$ ls
arquivo.log  arquivo.txt  Pasta1  Pasta2  Pasta3  Pasta4
user@maquina:~$ cd Pasta1
user@maquina:~$ ls
imagem.jpg      soma.txt      PastaA          PastaB
user@maquina:~$ clear

user@maquina:~$
```

Comandos - RESET

reset - reinicia o terminal

- Útil quando o terminal trava

```
user@maquina:~$ cat imagem.jpg
?dl?d??&` ? ?Ad?B9?S??yf???
??gvo??$ac??G??t??Tuu?t?qP?"?hsL6?N%=?Tw?Tq
?L7?Y?~?HE*R?T?%r~?H?J?|?\?
u?#9?keg?"X?t?~??T?c??S??HE*R??T?"?o????O"ik?
?QB?K?zU) v?? ??.?N?R???
v?? ??.?N? g%$?r
user@maquina:~$
```

Comandos - ECHO

echo [msg] - imprime na tela a mensagem *[msg]*

- *[msg] pode ser escrita entre aspas*
- Muito utilizado em scripts

```
user@maquina:~$ echo Olá amiguinhos!  
Olá amiguinhos!
```

```
user@maquina:~$
```

Comandos - CD

cd diretorio - change directory

- Muda o diretório corrente para o diretorio
 - diretorio pode ser um endereço completo ou relativo
 - diretorio pode ser um diretório especial (/., .., ~, -)

```
user@maquina:~$ pwd  
/home/user
```

```
user@maquina:~$ cd /etc  
user@maquina:~$ cd init.d
```

```
user@maquina:/etc$ pwd  
/etc/init.d
```

Comandos - LS

ls [opcoes][diretorio] - lista os arquivos do diretório atual

- Caso o **[diretorio]** seja passado, lista os arquivos dele
- Opções interessantes:
 - -a lista também os arquivos ocultos
 - -l lista o conteúdo do diretório de forma completa

```
user@maquina:~$ ls
```

Área de Trabalho	Desktop	Downloads	Modelos	Público
Compartilhado	Documentos	Imagens	Música	Vídeos

```
user@maquina:~$ ls /
```

bin	etc	lib	opt	selinux	usr
boot	home	lost+found	proc	srv	var
cdrom	initrd.img	media	root	sys	vmlinuz
dev	initrd.img.old	mnt	sbin	tmp	vmlinuz.old

Comandos - LS - Forma completa

```
user@maquina:~$ ls -l
total 4
-rwxr-xr-x 1 user user 179 2011-04-23 17:53 Arquivo.txt
```

1	2	3	4	5	6	7	8
-	rwxr-xr-x	1	user	user	179	2011-04-23 17:53	Arquivo.txt

Saída:

- **Total 4** - total em kBytes da pasta
- Campos
 - **1** “-” para arquivo e “d” para diretório
 - **2** esquema de permissão do arquivo
 - **3** número de diretórios dentro do arquivo
 - **4 e 5** usuário e grupo donos do arquivo
 - **6** tamanho do arquivo em blocos de 1kBytes
 - **7** data de modificação do arquivo
 - **8** nome do arquivo

Comandos - CP

cp [opcoes][origem][destino] - copia origem para destino

- Opções interessantes:

- -r copia diretórios dentro de diretórios recursivamente

```
user@maquina:~$ ls  
arquivo1.txt  Pasta
```

```
user@maquina:~$ cp arquivo1.txt arquivo2.txt  
user@maquina:~$ ls  
arquivo1.txt  arquivo2.txt  Pasta
```

```
user@maquina:~$ cp arquivo1.txt Pasta/arquivo3.txt  
user@maquina:~$ ls Pasta  
arquivo3.txt
```

Comandos - MV

mv [opcoes][origem][destino] - move origem para destino

- Pode ser usado também para renomear arquivos

```
user@maquina:~$ ls  
arquivo.txt      Pasta
```

```
user@maquina:~$ mv arquivo.txt Pasta  
user@maquina:~$ ls  
Pasta
```

```
user@maquina:~$ cd Pasta  
user@maquina:~/Pasta$ ls  
arquivo.txt
```

```
user@maquina:~/Pasta$ mv arquivo.txt outro_nome.txt  
user@maquina:~/Pasta$ ls  
outro_nome.txt
```

Comandos - *Mkdir*

mkdir [opcoes][diretório] - cria a pasta *diretório*

- **Opções interessantes:**

- -p cria diretórios dentro de diretórios

```
user@maquina:~$ ls  
arquivo.txt
```

```
user@maquina:~$ mkdir Pasta1  
user@maquina:~$ ls  
arquivo.txt     Pasta1
```

```
user@maquina:~$ mkdir -p Pasta2/Pasta3  
user@maquina:~$ ls  
arquivo.txt     Pasta1     Pasta2
```

```
user@maquina:~$ ls Pasta2  
Pasta3
```

Comandos - RM

rm [opcoes][arquivo] - remove o arquivo

- **Opções interessantes:**

- -f força a remoção dos arquivos
- -r remove diretórios e seus conteúdos recursivamente

```
user@maquina:~$ ls  
arquivo1.txt  arquivo2.txt  Pasta
```

```
user@maquina:~$ rm rquivol.txt  
user@maquina:~$ ls  
arquivo2.txt  Pasta
```

```
user@maquina:~$ rm Pasta  
rm: não foi possível remover "Pasta": É um diretório
```

```
user@maquina:~$ rm -r Pasta  
user@maquina:~$  
arquivo2.txt
```

Comandos - RMDIR

rmdir [opcoes][diretório] - remove o *diretório*

- **Opções interessantes:**

- -p remove o diretório e todos os acima dele, desde que não tenham mais nenhum arquivo dentro

```
user@maquina:~$ ls  
Pasta1 Pasta2
```

```
user@maquina:~$ ls Pasta1  
user@maquina:~$ ls Pasta2  
Pasta3
```

```
user@maquina:~$ rmdir Pasta1  
user@maquina:~$ rmdir Pasta2  
falhou em remover o diretório "Pasta2": Diretório não vazio
```

```
user@maquina:~$ rmdir -p Pasta2/Pasta3  
user@maquina:~$ ls  
Pasta1
```

Comandos - CAT

cat [arquivo] - lista o conteúdo do arquivo

```
user@maquina:~$ ls  
arquivo.txt      imagem.jpg
```

```
user@maquina:~$ cat arquivo.txt  
Olá amiguinhos!
```

```
user@maquina:~$ cat imagem.jpg  
?PNG
```

```
? ? s ? ? O ? tEXtSoftwaregnome-screenshot ? ? > IDATx ? ? ? w ? ? ? ?  
\ ? = ? ? ? ? SH ? ? A " ' ? ? ? dl ? d ? ? ? ? & ? ` ?  
? ? Ad ? B9 ? S ? ? yf ? ? ?  
? ? gvo ? ? $ac ? ? ? ? G ? ? ? t ? ? ? ? Tuu ? t ? qP ? " ? hsL6 ? N%=? Tw ? Tq  
? L7 ? Yl ? ? ? ~ ? ? ? HE * R ? ? T ? ? % r ~ ? H ? J ? | ? \ ?
```

```
u ? ? #9 ? keg ? " X ? t ? ~ ? ? % ? ? T ? C ? ? S ? ? HE * R ? ? T ? " ? o ? ? ? ? O " ik  
? ? QB ? K ? zU ) v ? ? ? ? N ? R ? ? ? ? ?
```

Comandos - LESS

less* [arquivo]** - lista o conteúdo do arquivo, com opções mais avançadas que o comando ***cat

- Opções:

- Navegação pelo arquivo setas acima e abaixo
- Primeira linha g
- Última linha G
- Destacar palavra :palavra
- Sair q

Comandos - HEAD

head [opções][arquivo] - lista as primeiras linhas do arquivo

- Opções interessantes:

- **-n [valor]** lista as primeiras [valor] linhas
- Por padrão, o comando **head** mostra 10 linhas

```
user@maquina:~$ head -n 3 arquivo
Olá amiguinhos!
Como vocês estão?
Eu tô bem obrigado
```

```
user@maquina:~$ head -n 5 arquivo
Olá amiguinhos!
Como vocês estão?
Eu tô bem obrigado
Ah que beleza então
Puxa vida
```

Comandos - TAIL

tail [opções][arquivo] - lista as últimas linhas do arquivo

- Opções interessantes:

- **-n [valor]** lista as últimas [valor] linhas
- Por padrão, o comando **tail** mostra 10 linhas

```
user@maquina:~$ tail -n 3 arquivo
Puxa vida
Sóóóóóóó
Manero
```

```
user@maquina:~$ tail -n 5 arquivo
Eu tô bem obrigado
Ah que beleza então
Puxa vida
Sóóóóóóó
Manero
```

Comandos - REBOOT

reboot - reinicia o computador

- Em alguns sistemas, só o *root* pode reiniciar o computador
 - O comando *sudo* pode ser usado como alternativa

```
user@maquina:~$ sudo reboot
!! O sistema será reiniciado agora !!
```

Comandos - ~~HALT~~

halt - desliga o computador

- Em alguns sistemas, só o *root* pode desligar o computador
 - O comando *sudo* pode ser usado como alternativa

```
user@maquina:~$ sudo halt
!! O sistema será desligado agora !!
```

Mais comandos

- wiki.ubuntu-br.org/ComandosBasicos
- focalinux.cipsga.org.br/guia/iniciante/ch-cmdv.htm
- www.vivaolinux.com.br
- www.dicas-l.com.br
- WWW.GOOGLE.COM