



1/36



Para Todos os Cursos

Linux: linha de  
comando

Grupo Colméia  
Claudio Cesar de Sá  
\$ : linha de comando



Back

Close



2/36

## Direitos Autorais:

- ⇒ Claudio Cesar de Sá (claudio.sa@udesc.br)
- ⇒ Livre para usar e melhorar
- ⇒ Me retorne com as melhorias!
- ⇒ Atualizado em <https://github.com/claudiosa/CCS/> ⇒ linux



Back

Close

# Contents

1	Direitos Autorais:	2
2	O que é o Linux?	5
3	Linux em qualquer máquina	6
4	Uma Interface Linux	7
5	Linux x Hardware	8
6	Como acessar o <i>kernel</i> Linux?	9
7	Acesso via Terminal ou Consoles	10
8	Um Terminal	11



3/36



Back

Close

9	Comandos no Linux	12
10	As Teclas	13
11	Estendendo estes atalhos de teclado	14
12	Atalhos para mover o cursor	15
13	Comandos Introdutórios	16



4/36



Back

Close



5/36

# O que é o Linux?

- Um Sistema Operacional, como Windows
- Código aberto – sem proprietários
- Suas versões iniciais datam dos anos 199x
- Nome vem do idealizador: Linus Torvalds
- Muitos cursos *on-line* tais como:

<https://br.udacity.com/course/linux-command-line-basics--ud59>



Back

Close

# Linux em qualquer máquina



Figure 1: Linux a U\$ 249,00



6/36



Back

Close

# Uma Interface Linux



Figure 2: Uma Interface Linux



7/36



Back

Close

# Linux x Hardware

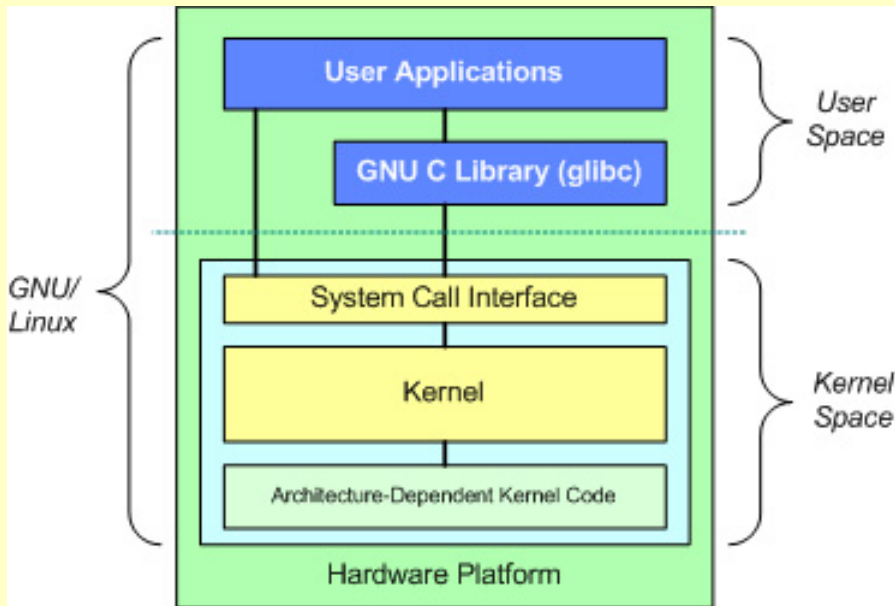


Figure 3: Linux x Hardware





# Como acessar o *kernel* Linux?



9/36

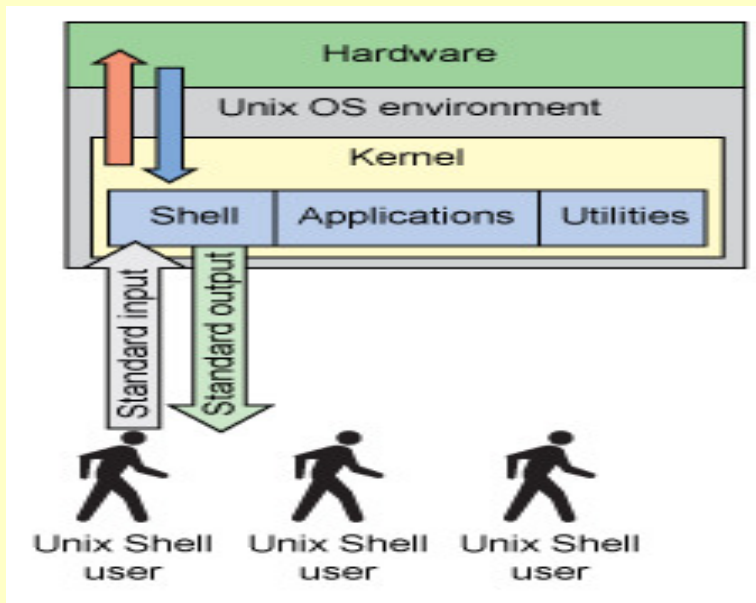
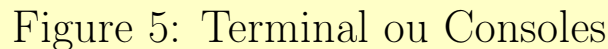


Figure 4: Linux x Hardware



Back

Close





# Um Terminal

```
dan@dan-virtual-machine:~/Documents/essays$ ls
essay1.txt essay2.txt essay3.txt essay4
dan@dan-virtual-machine:~/Documents/essays$ cd ..
dan@dan-virtual-machine:/$
dan@dan-virtual-machine:/$ bin cdrom
dan@dan-virtual-machine:/$ boot dev
dan@dan-virtual-machine:/$ updatedb: can
dan@dan-virtual-machine:/$ [sudo] passwor
dan@dan-virtual-machine:/$ /home/dan/Doc
dan@dan-virtual-machine:/$ /home/dan/
dan@dan-virtual-machine:/$ /home/dan/Documents/mv-new-name-essay1.txt
```

Figure 6: Isto é uma console ou terminal (*o fim do mundo!*)





# Comandos no Linux

1. Não ocorrer nesta linha de comando
2. Você terá que aprender e decorar uns comandos (traga um *dataset*)
3. Estes comandos serão interpretados pelo programa *bash*, o qual acionam o *kernel* linux
4. e só!





# As Teclas

- No linux as teclas também ocupam o lugar do *mouse*, estas são **aceleradoras**
- Quando tiveres prática, o *rato* fica em segundo momento!

Começando:

**xterm ou lxterminal:** aplicativo terminal (aqui faz tudo!)

**Cursor + botão do meio:** *copy-paste* ao ter um texto marcado

**Alt + Tab:** alterna entre as aplicações ativas

**↑ e ↓:** repetem os comandos digitados neste terminal

**^:** é o ENTER o qual deve ser pressionado após os comandos que se seguem



Back

Close



# Estendendo estes atalhos de teclado

## Sobre processos:

Ctrl+C: Mata um processo sob o aplicativo terminal

Ctrl+Z: Depois

Ctrl+D: Fecha a janela terminal (não use), apenas em desespero de causa!

## Sobre a tela:

Ctrl+L: Limpa a tela no terminal (uso bastante)

Ctrl+S: Faz um **stop** de uma saída no terminal

Ctrl+Q: Retoma esta saída pausada na janela terminal





# Atalhos para mover o cursor

## Atalhos na linha de comando:

Ctrl+A ou Home: move para o início de linha

Ctrl+E ou End: move para o fim de linha

Ctrl+B: move o cursor para palavra **antes** dele (isto é: à esquerda)

Alt+F: move o cursor para palavra **depois** dele (isto é: à direita)

Ctrl+F: move o cursor um carácter à **direita** dele (muito útil)

Ctrl+D ou Delete: exclui o carácter e puxa-o da direita dele o próximo carácter

Alt+D: exclui todos caracteres à direita dele

Ctrl+H ou Backspace: exclui o carácter antes dele

Falta ainda o *copy-paste* usando teclas!





16/36

# Comandos Introdutórios

➡ Vendo o diretório onde estou:

```
$ pwd  
/home/udesc
```



Back

Close



➡➡ Listando o conteúdo do diretório:

```
$ ls
```

```
.....
```

```
$
```

```
TESTE ESTE
```

```
$ ls *.txt
```

```
.....
```

```
$
```

```
TESTE ESTE
```

```
$ ls .*
```

```
.....
```



17/36



Back

Close

➡ Cria um diretório

```
$ mkdir seu_diretorio
```

```
$ ls
```

```
TEM QUE APARECER seu_diretorio LAH
```



18/36



Back

Close

➡ Entrando dentro de uma pasta/diretório:

```
$ cd seu_diretorio/      VAI para seu diretorio
$ cd ..                  SOBE um nivel acima
$ cd pgms_prolog/        VAI PARA BAIXO ou um dado diretorio
$ cd ~ ATEH RAIZ HOME
$ pwd
/home/udesc
$
```



19/36



Back

Close

➡ Cria um arquivo e lista o conteúdo:

```
$ touch nome_arquivo.txt
```

```
$ ls -al *.txt
```

```
-rwxr-xr-x 1 udesc udesc 435 2011-08-29 15:34 append.txt
```

```
-rw-r--r-- 1 udesc udesc 0 2011-08-29 19:41 nome_arquivo.txt
```



20/36



Back

Close



➔ Passos para os laboratórios da turma de ALP (os demais cursos, mais adiante):

- `gterminal`: é o terminal já mencionado
- `mkdir SEU_NOME ^`: cria diretório
- `cd SEU_NOME ^`: vai para o seu diretório
- `touch programa.c ^`: criou um arquivo chamado `programa.c`
- `geany programa.c & ^`: edita o `programa.c` com o `geany`
- `gcc programa.c ^`: compila o arquivo o `programa.c`
- `ls a* ^`: verifica se o programa executável foi gerado, é o `a.out`
- `./a.out ^`: executa o `a.out`
- `ls ^`: lista diretório corrente



➡ Erro recorrente da turma:

```
$ geany nome_arquivo.txt ^ // processo PARADO  
^Z  
[1]+  Parado                //geany xxxx  PARADO  
$
```

```
$ bg 1 ^                // processo 1 PARADO  
[1]+ geany nome_arquivo.txt & ^ // bg ATIVANDO-O
```

o correto eh:

```
$ geany nome_arquivo.txt & ^  
  
// NAO ESQUECA O & comercial ao final  
// ^ eh o ENTER ....
```



22/36



Back

Close

➡➡ Listando processos na memória:

```
$ ps
  PID TTY          TIME CMD
 4685 pts/2    00:00:00 bash
 4790 pts/2    00:00:00 ps
```

```
$ ps -aux | grep udesc
.....
```

```
$ ps -aux | grep udesc | more
.....
lista os processos por pagina
```

```
TESTE
$ htop
```



23/36



Back

Close

## ➡ Processos na memória e seu estado:

```
$ ps aux | grep NOME_DO_PROGRAMA
```

SE PARADO, por exemplo:

```
$ gedit nome_arquivo.txt    // processo em modo parado  
^Z
```

```
[1]+  Parado
```

```
$
```

```
$ bg 1
```

```
[1]+ gedit nome_arquivo.txt &    // posto em background
```



24/36



Back

Close



➡ Criar e apagar arquivos e diretórios:

```
$ touch x
```

```
$ rm x
```

```
$ mkdir cria_diretorio
```

```
$ rmdir cria_diretorio
```

- Diretórios só apagam se estiverem vazios!
- **NUNCA:** `rm *` muito menos `rm -r *`



25/36



Back

Close

➡ Copiar um arquivo:

```
$ cp origem.txt destino.txt
```

```
$ cp casa.pdf /media/arch_linux/pgms_prolog/
```



26/36



Back

Close

➤ Copiando recursivamente um diretório:

```
$ cp -R haskell/ /media/arch_linux/
```



27/36



Back

Close

➤➔ Renomear um arquivo:

```
$ mv casa.pdf /media/arch_linux/pgms_prolog/
```

```
$ rename atual novo
```



28/36



Back

Close

➡ Limpar tela:

clear

reset



29/36



Back

Close

➡ Mostrar o conteúdo de um arquivo texto use o comando *more*:

```
$ more append.txt
more 001_tipos_dados_C.c
#include <stdio.h>
int main()
{    // INICIO { ... comentado
    int ANO                ;
    float PI = 3.141519141519141519141519 ;
    char  sexo = 'M'       ;
    char  nome[20] = "Isto eh uma string" ;
    bool  luz = true       ;
    // ISTO EH UMA ATRIBUICAO
    ANO = 2017              ;
// ESCRITA DE VALORES
/*
f : eh file = arquivo
stdin: standard input = teclado
stdout: standard output = tela
--More--(44%)
```



*Obs: o comando cat também funciona para qualquer tipo de arquivo!*



31/36



Back

Close

➡ Pesquisar um arquivo com um dado específico:

```
$ grep "Y" *.pl
aula-15-08a.pl:      p(Y) ,
aula-15-08a.pl:      X \== Y,
aula-15-08a.pl:      Z is (X + Y) ,
.....
udesc@matrizubuntu9:~/pgms_prolog$
```



32/36



Back

Close



- ➔ Criar um link simbólico ou atalho (em geral se cria este atalho em /usr/bin ou /usr/local/bin).

EXEMPLO:

```
$ ln -s caminho/minizinc minimizinc
```

PERMISSAO DE EXECUCAO:

```
$ chmod +x minimizinc (em /usr/bin)
```

caminho = onde foi instalado o ORIGINAL

*A soft link, or more common, a symlink, is link a shortcut to the targeted file or directory. So when is removed the original target stays present. This is the opposite of a hard link which is a reference to the target and so, if the hard link is removed, so is the target.*



33/36



Back

Close

➡➡ Remover um link simbólico ou atalho (não link físico)

APENAS para o SIMBOLICO ....

```
$ ln -s caminho/nota_Minizinc_IDE.txt TESTE.TXT
```

```
$ rm TESTE.TXT
```

```
$ ls caminho/*.txt
```

```
$ nota_Minizinc_IDE.txt
```



34/36



Back

Close

➤➤ Remover um link simbólico ou atalho (não link físico) com segurança

```
$ unlink link_simbolico_criado
```

O original ficou intacto!



35/36



Back

Close

## Contato:

UDESC/CCT/DCC

Grupo de Hardware e Software Livre – Colméia

## Sítio de Referência:

<http://www.colmeia.udesc.br/>



36/36



Back

Close