



1/33



Para Todos os Cursos

Fast Linux

Grupo Colméia
\$: linha de comando



Back

Close



2/33

Contents

1	O que é o Linux?	4
2	Linux em qualquer máquina	5
3	Uma Interface Linux	6
4	Linux x Hardware	7
5	Como acessar o <i>kernel</i> Linux?	8
6	Acesso via Terminal ou Consoles	9
7	Um Terminal	10
8	Comandos no Linux	11



Back

Close

9 As Teclas

12

10 Comandos Introdutórios

13



3/33



Back

Close



4/33

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



Back

Close

Linux em qualquer máquina



Figure 1: Linux a U\$ 249,00



5/33



Back

Close

Uma Interface Linux



Figure 2: Uma Interface Linux



6/33



Back

Close

Linux x Hardware

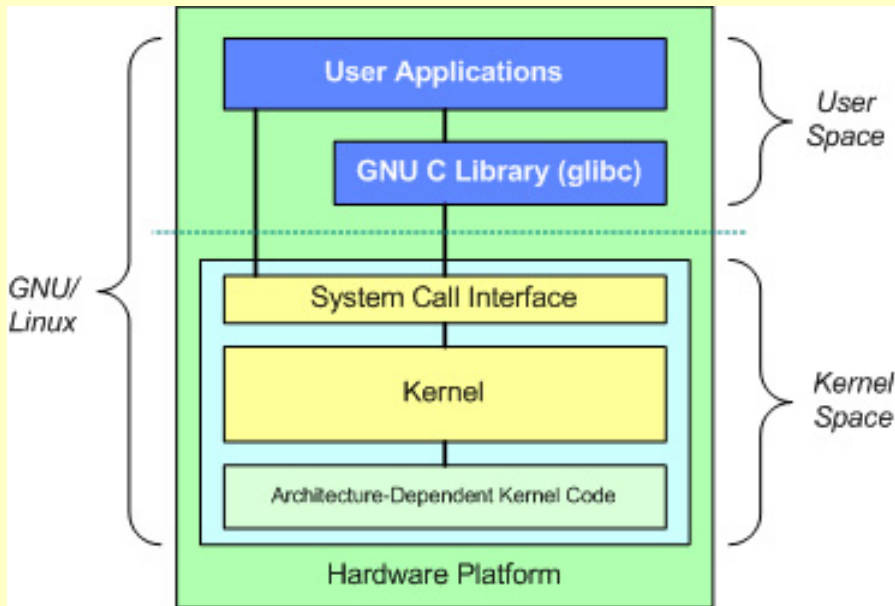


Figure 3: Linux x Hardware



Como acessar o *kernel* Linux?



8/33

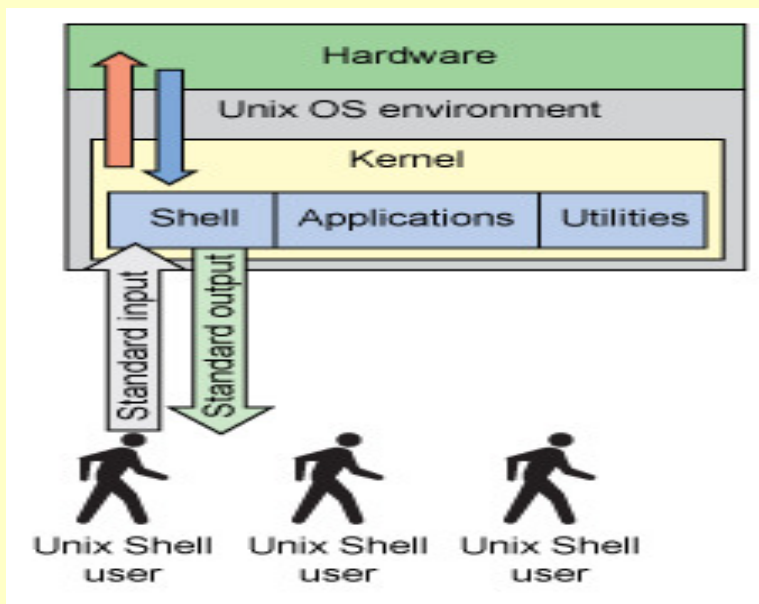


Figure 4: Linux x Hardware





Back

Close

Figure 5: Terminal ou Consoles



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:~$ ls
bin  cdrom  dev  etc  home  lib  lib64  media  mnt  opt  proc  run  sbin  sys  tmp  usr  var
dan@dan-virtual-machine:~$ cd /
dan@dan-virtual-machine:/ $ ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  run  sbin  sys  tmp  usr  var
dan@dan-virtual-machine:/ $ updatedb: can't open /etc/passwd: No such file or directory
dan@dan-virtual-machine:/ $ [sudo] password dan:
dan@dan-virtual-machine:/ $ cd /home/dan/Documents
dan@dan-virtual-machine:/home/dan/Documents $ mv new-name-essay1.txt
dan@dan-virtual-machine:/home/dan/Documents $
```

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!



Back

Close



As Teclas

No linux as teclas ocupam o lugar do *mouse*, estas são **aceleradoras**

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





13/33

Comandos Introdutórios

➡ Vendo o diretório onde estou:

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



Back

Close

➡ Listando o conteúdo do diretório:

```
$ ls
```

```
append.txt      lab_inic.pl      rapidas_linux.out  
aula-15-08a.pl  lab_recurso.c    rapidas_linux.pdf  
aula-15-08.pl   lab_recurso.pl   rapidas_linux.tex
```

```
.....
```

```
$
```

```
TESTE ESTE
```

```
$ ls .*
```



14/33



Back

Close

➡ Cria um diretório

```
$ mkdir seu_diretorio
```

```
$ ls
```

```
TEM QUE APARECER seu_diretorio LAH
```



15/33



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
$
```



10/33



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
```



17/33



Back

Close



➤ Passos para os laboratórios da turma de ALP:

- `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`
- `g++ 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:

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



19/33



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
```



20/33



Back

Close

➡ Processos na memória e seu estado:

```
$ gedit nome_arquivo.txt &
```

```
[1] 4801
```

```
$ ps aux | grep gedit
```

```
udesc      4809  3.1  0.4  58132 16668 pts/2    Sl   19:54   0
```

```
udesc      4814  0.0  0.0   3060   816 pts/2    S+   19:54   0
```

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

```
^Z
```

```
[1]+  Parado
```

```
gedit nome_arquivo.txt
```

PARADO

```
$
```

```
$ bg 1
```

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



21/33



Back

Close

➤➔ Apagar arquivo e diretório:

```
$ touch x
```

```
$ rm x
```

```
$ mkdir cria_diretorio
```

```
$ rmdir cria_diretorio
```



22/33



Back

Close

➡ Copiar um arquivo:

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

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



23/33



Back

Close

➤ Copiando recursivamente um diretório:

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



24/33



Back

Close

➤➔ Renomear um arquivo:

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

```
$ rename atual novo
```



25/33



Back

Close

➡ Limpar tela:

clear

reset



26/33



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%)
```



27/33



Back

Close

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



28/33



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$
```



29/33



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.



➡ 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
```



31/33



Back

Close

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

```
$ unlink link_simbolico_criado
```

O original ficou intacto!



32/33



Back

Close

Contato:

UDESC/CCT/DCC

Grupo de Hardware e Software Livre – Colméia

Sítio de Referência:

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



33/33



Back

Close