



























Para Todos os Cursos

Linux: |inha de comando

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







2/3

Direitos Autorais:

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







Contents

- **Direitos Autorais:**
- O que é o Linux?
- Linux em qualquer máquina
- Uma Interface Linux
- Linux x Hardware
- Acesso via Terminal ou Consoles
 - - Um Terminal
- Como acessar o kernel Linux?
- - - - - 9

2

5

6

8

11

10







9 Comandos no Linux	12	À
10 As Teclas	13	李季
11 Estendendo estes atalhos de teclado	14	4/37
12 Atalhos para mover o cursor	15	
13 Comandos Introdutórios	16	
		44
		1
		Back
		Close

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 Tosvalds
- Muitos cursos on-line tais como:

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









6/37

Linux em qualquer máquina



Figure 1: Linux a U\$ 249,00







Uma Interface Linux



Figure 2: Uma Interface Linux







Linux x Hardware

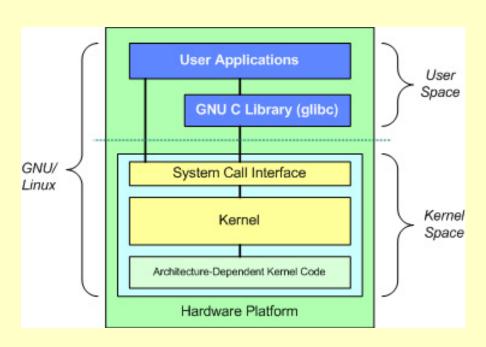


Figure 3: Linux x Hardware









Como acessar o kernel Linux?

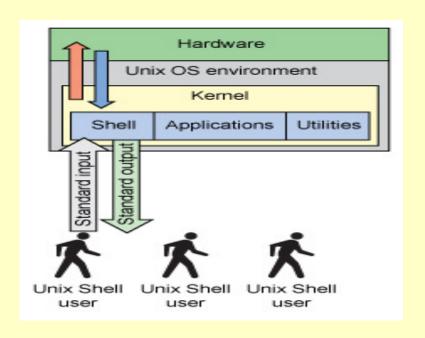


Figure 4: Linux x Hardware







Acesso via Terminal ou Consoles

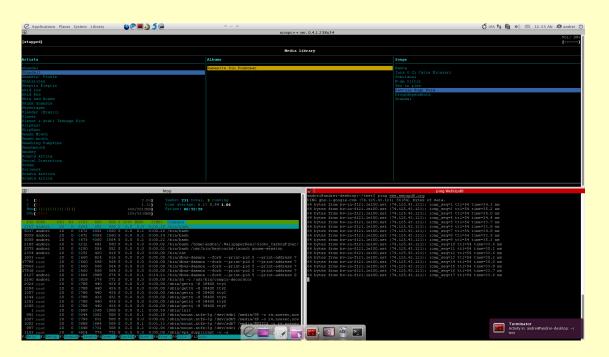


Figure 5: Terminal ou Consoles









Um Terminal

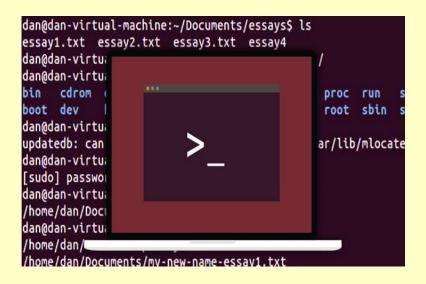


Figure 6: Isto é uma console ou terminal (o fim do mundo!)









- 1. Vão ocorrer nesta linha de comando
- 2. Voce terá que aprender e decorar uns comandos (traga um datashet)
- 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

- \uparrow e \downarrow : repetem os comandos digitados neste terminal
- **^:** é o ENTER o qual deve ser pressionado após os comandos que se seguem



Back



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





С



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)

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

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

Alt+D: exclui todos carácteres à direita dele

Ctrl+H ou Backspace: exclui o carácter antes dele

Falta ainda o *copy-paste* usando teclas!







Comandos Introdutórios

>> Vendo o diretório onde estou:

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









➤ Listando o conteúdo do diretório:	
\$ ls	季
\$ TESTE ESTE \$ ls *.txt	17/37
\$ TESTE ESTE \$ ls .*	
	44





Back

- → Cria um diretório
 - \$ mkdir seu_diretorio
 - \$ 1s

TEM QUE APARECER seu_diretorio LAH







Back

```
➤ Entrando dentro de uma pasta/diretório:
   $ cd seu_diretorio/
VAI para seu diretorio
```

```
SOBE um nivel acima
$ cd ..
```

- cd pgms_prolog/ VAI PARA BAIXO ou um dado diret
- \$ cd ~ ATEH RAIZ HOME
- \$ pwd
- /home/udesc
- \$









- ➤ 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.tct 20/37 -rw-r--r-- 1 udesc udesc 0 2011-08-29 19:41 nome_arquivo.tx





Back

Back

- → Passos para os laboratórios da turma de ALP (os demais cursos, mais adiante):
 - 1. terminal: é o terminal já mencionado
 - 2. mkdir SEU_NOME ^: cria diretório
 - 3. cd SEU_NOME ^: vai para o seu diretório
 - 4. touch programa.c ^: criou um arquivo chamado programa.c
 - 5. geany programa.c & ^: editar o programa.c no geany
 - 6. gcc programa.c ^: compila o arquivo o programa.c
 - 7. ls a* ^: verifica se o programa executável foi gerado, é o a.out
 - 8. ./a.out ^: executa o a.out
 - 9. ls ^: lista diretório corrente

➤ Erro recorrente das turmas:

```
$ geany nome_arquivo.txt ^ // processo PARADO
[1]+ Parado
                         //geany xxxx PARADO
$ bg 1 ^
                           // processo 1 PARADO
[1] + geany nome_arquivo.txt & ^ // bg ATIVANDO-0
o correto eh:
$ geany nome_arquivo.txt & ^
         // NAO ESQUECA O & comercial ao final
          // ^ eh o ENTER ....
```







Back

- ➤ Compilando os códigos em C (pessoal de EDA):
 - 1. gcc -Wall nome_programa.c -o saida.exe
 - 2. usem a opção -Wall mostra todos os *warnings* que o run.codes acusa
 - 3. ./saida.exe < entrada.txt









```
➤ Listando processos na memória:
```

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

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

lista os processos por pagina

```
TESTE $ htop
```







```
➤ 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
```





- ➤ 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 *







- ➤ Copiar um arquivo:
 - \$ cp origem.txt destino.txt
 - \$ cp casa.pdf /media/arch_linux/pgms_prolog/









- ➤ Copiando recursivamente um diretório:
 - \$ cp -R haskell/ /media/arch_linux/







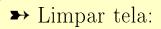
- ➤ Renomear um arquivo:
 - \$ mv casa.pdf /media/arch_linux/pgms_prolog/
 - \$ rename atual novo











clear reset









Back

```
→ 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!









```
➤ 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$
```







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

EXEMPLO:

- \$ ln -s caminho/minizinc minizinc
- PERMISSAO DE EXECUCAO:
- \$ chmod +x minizinc (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.



44



Back

- 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









- Remover um link simbólico ou atalho (não link físico) com segurança
 - \$ unlink link_simbolico_criado
 - O original ficou intacto!







Contato:

UDESC/CCT/DCC

Grupo de Hardware e Software Livre – Colméia

Sítio de Referência:

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









Back