



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Introdução a Sistemas Abertos

Ambiente shell

Conteúdo Programático

- Login e Logout
- Iniciando e Desligando o Sistema
- Tipos de arquivo
 - Texto
 - Binário
- Arquivos e Diretório
 - Manipulação de arquivos e diretório
- Interpretador de Comandos (SHELL)

Login e Logout

Depois de instalado, o Sistema Operacional Linux só pode ser acessado mediante fornecimento de **credencial** composta por **login** e **senha**

É de extrema importância zelar pela credencial e não deixá-la disponível para outras pessoas, deixando o computador logado, por exemplo, e saindo para um cafezinho.

```
Debian GNU/Linux 5.0 sor tty1  
sor login: _
```

Login e Logout

Login é a entrada no sistema quando você digita seu nome (usuário) e senha

```
Debian GNU/Linux 5.0 sor tty1
sor login: sor
Password:
```

Logout é a saída do sistema.

A saída do sistema é feita pelos comandos logout, exit, CTRL+D, ou quando o sistema é reiniciado ou desligado

```
sor@labsor:~$ logout_
```

Entendendo o Prompt linux

Ao efetuar login em um terminal linux receberemos na tela um interpretador de comandos linux, como o apresentado na imagem abaixo:

```
labrador@labrador: ~$
```

Nome do usuário

Nome do host

Diretório atual

O símbolo '#' será mostrado para o root e o '\$' para demais usuários

Prompt de comando

Interpretador de comandos

Também conhecido como "Shell". É o programa responsável por interpretar as instruções enviadas pelo usuário e seus programas ao kernel;

O Shell executa comandos lidos do dispositivo de entrada padrão (teclado) ou de um arquivo executável;

O Shell é a principal ligação entre o usuário, os programas e o kernel;

O GNU/Linux possui diversos interpretadores de comandos, entre os mais comuns estão o bash, sh e o ksh. O bash é o mais utilizado nas distribuições Linux

Interpretador de comandos

Os comandos podem ser enviados ao Shell de duas maneiras: interativa e não-interativa:

Interativa

Os comandos são digitados no aviso de comando (Prompt) e passados ao interpretador de comandos um a um. Neste modo, o computador depende do usuário para executar uma tarefa, ou próximo comando.

Interpretador de comandos

Não-interativa

São usados arquivos de comandos criados pelo usuário (scripts) para o computador executar os comandos na ordem encontrada no arquivo. Neste modo, o computador executa os comandos do arquivo um por um e dependendo do término do comando, o script pode checar qual será o próximo comando que será executado e dar continuidade ao processamento.

Este sistema é útil quando temos que digitar por várias vezes seguidas um mesmo comando ou para compilar algum programa complexo



Algumas teclas de atalho

Teclas	Ação
Ctrl + l	Limpa a tela, equivalente ao comando clear
Ctrl + d	Encerra entrada de dados pelo teclado fazendo logout
Ctrl + a	Para ir ao início da linha de comando
Ctrl + e	Para ir ao final da linha de comando
Ctrl + u	Limpa a linha de comando corrente
Ctrl + c	Termina a execução do comando corrente

Tecla TAB

O shell Bash possui ainda outra característica interessante: A completção dos nomes. Isto é feito pressionando-se a tecla TAB. Por exemplo, se digitar “ls tes” e pressionar <tab>, o Bash localizará todos os arquivos, daquele diretório, que iniciam com “tes” e completará o restante do nome.

Caso a completção de nomes encontre mais do que uma expressão que satisfaça a pesquisa, ou nenhuma, é emitido um beep. Se você apertar novamente a tecla TAB imediatamente depois do beep, o interpretador de comandos irá listar as diversas possibilidades que satisfazem a pesquisa, para que você possa escolher a que lhe interessa.

A completção de nomes também funciona para comandos internos.

Desligando o computador

Para desligar o computador, primeiro digite (como root): "**shutdown -h now**", "**halt**" ou "**poweroff**". O GNU/Linux finalizará os programas e gravará os dados em seu disco rígido, quando for mostrada a mensagem "power down", desligue o computador

```
root@labsor:~$ halt_
```

Desligando o computador

NUNCA desligue diretamente o computador sem usar o comando **shutdown**, **halt** ou **poweroff**, pois podem ocorrer perda de dados ou falhas no sistema de arquivos de seu disco rígido devido a programas abertos e dados ainda não gravados no disco.

Arquivo texto e binário

Texto

Seu conteúdo é compreendido pelas pessoas. Um arquivo texto pode ser uma carta, um script, um programa de computador escrito pelo programador, arquivo de configuração, etc.

Binário

Seu conteúdo entendido apenas por computadores. Contém caracteres incompreensíveis para pessoas normais. Um arquivo binário é gerado a partir de um código fonte (formato texto) através de um processo chamado de compilação.

Compilação é basicamente a conversão de um programa em linguagem humana para a linguagem de máquina

Diretório

Diretório é o local utilizado para armazenar conjuntos de arquivos para melhor organização e localização. O diretório, como o arquivo, também é "Case Sensitive" (diretório /**teste** é completamente diferente do diretório /**Teste**).

Não podem existir dois arquivos com o mesmo nome em um diretório, ou um sub-diretório com um mesmo nome de um arquivo em um mesmo diretório.

Diretório Raíz “/”

Este é o diretório principal de um sistema GNU/Linux.

Dentro dele estão todos os diretórios do sistema. O diretório Raíz é representado por uma “/”, assim se você digitar o comando **cd /** você estará acessando este diretório.

```
sor@labsor:~$ cd /  
sor@labsor:/$ _
```

Diretório home

Também chamado de diretório de usuário. Em sistemas GNU/Linux cada usuário (inclusive o root) possui seu próprio diretório onde poderá armazenar seus programas e arquivos pessoais.

Este diretório está localizado em **/home/[login]**, neste caso se o seu login for "joao" o seu diretório home será **/home/joao**. O diretório home também é identificado por um **~(til)**, você pode digitar tanto o comando `ls /home/joao` como `ls ~` para listar os arquivos de seu diretório home.

O diretório home do usuário root (na maioria das distribuições GNU/Linux) está localizado em `/root`.

Diretório atual

É o diretório em que nos encontramos no momento. Você pode digitar **pwd** para verificar qual é seu diretório atual.

```
sor@labcor:~$ pwd  
/home/sor
```

O diretório atual também é identificado por um "." (ponto). O comando **ls .** pode ser usado para listar seus arquivos (é claro que isto é desnecessário porque se não digitar nenhum diretório, o comando **ls** listará o conteúdo do diretório atual)

Diretório Superior

O diretório superior (Upper Directory) é identificado por `..` (2 pontos).

```
sor@lab:~$ pwd
/home/sor
sor@lab:~$ ls ..
sor
```

Caso estiver no diretório **`/usr/local`** e quiser listar os arquivos do diretório **`/usr`** você pode digitar, **`ls ..`**. Este recurso também pode ser usado para copiar, mover arquivos/diretórios, etc.

Diretório Anterior

O diretório anterior é identificado por traço "-". É útil para retornar ao último diretório usado

Se estiver no diretório **/usr/local** e digitar **cd /lib**, você pode retornar facilmente para o diretório **/usr/local** usando **cd -**.

```
sor@labsor:/usr/local$ pwd
/usr/local
sor@labsor:/usr/local$ cd /lib
sor@labsor:/lib$ cd -
/usr/local
sor@labsor:/usr/local$ _
```

Caminho Completo e Caminho relativo

No Linux, é possível se referir a um caminho completo, ou seja, a partir do raiz, ou a partir de um ponto intermediário. Por exemplo:

Se o usuário estiver no **/usr/local** e quiser entrar em **/usr/local/share/man**, ele pode fazer das seguintes formas:

Completo: `cd /usr/local/share/man`

Relativo: `cd share/man`

Exemplos de utilização dos comandos

Comando **ls** para listar arquivos e diretórios no sistema de arquivos

```
sor@labsor:~/usr/local$ ls -l
total 32
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 bin
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 etc
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 games
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 include
drwxrwsr-x 3 root staff 4096 Abr  3 17:22 lib
lrwxrwxrwx 1 root staff    9 Abr  3 16:52 man -> share/man
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 sbin
drwxrwsr-x 3 root staff 4096 Abr  3 16:52 share
drwxrwsr-x 2 root staff 4096 Abr  3 16:52 src
```

Exemplos de utilização dos comandos

Comando **cd** para navegar no sistema de arquivos

```
sor@labsor:~/usr/local$ pwd
~/usr/local
sor@labsor:~/usr/local$ cd etc
sor@labsor:~/usr/local/etc$ ls
sor@labsor:~/usr/local/etc$ cd ../../
sor@labsor:~/usr$ pwd
~/usr
sor@labsor:~/usr$ cd /home/sor/
sor@labsor:~$
```

Exemplos de utilização dos comandos

Comando **mkdir** para criar diretórios (pastas)

```
sor@labsor:~$ mkdir dir_001
sor@labsor:~$ ls
dir_001
sor@labsor:~$ mkdir dir_002
sor@labsor:~$ ls
dir_001  dir_002
sor@labsor:~$ mkdir dir_002/teste
sor@labsor:~$ ls
dir_001  dir_002
sor@labsor:~$ mkdir -p dir_003/teste3
sor@labsor:~$ ls dir_003/
teste3
sor@labsor:~$
```

Exemplos de utilização dos comandos

Comando **cp** para copiar arquivos e diretórios

```
sor@labsor:~$ ls -l
total 16
-rw-r--r-- 1 sor sor  38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 3 sor sor 4096 Abr  4 19:13 dir_002
drwxr-xr-x 3 sor sor 4096 Abr  4 19:14 dir_003
sor@labsor:~$ cp arquivo1.txt arquivo2.txt
sor@labsor:~$ ls -l
total 20
-rw-r--r-- 1 sor sor  38 Abr  4 19:18 arquivo1.txt
-rw-r--r-- 1 sor sor  38 Abr  4 19:18 arquivo2.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 3 sor sor 4096 Abr  4 19:13 dir_002
drwxr-xr-x 3 sor sor 4096 Abr  4 19:14 dir_003
sor@labsor:~$
```


Exemplos de utilização dos comandos

Comando **mv** para mover ou renomear arquivos e diretórios

```
sor@labsor:~$ ls
arquivo1.txt  dir_001  dir_002  dir_003  teste.txt
sor@labsor:~$ ls -l
total 20
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 3 sor sor 4096 Abr  4 19:13 dir_002
drwxr-xr-x 3 sor sor 4096 Abr  4 19:14 dir_003
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 teste.txt
sor@labsor:~$ mv dir_003 dir_002/
sor@labsor:~$ ls -l
total 16
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 4 sor sor 4096 Abr  4 19:39 dir_002
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 teste.txt
sor@labsor:~$ _
```

Exemplos de utilização dos comandos

Comando **rm** para remover (deletar) arquivos e diretórios

```
sor@labsor:~$ ls -l
total 16
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
drwxr-xr-x 4 sor sor 4096 Abr  4 19:39 dir_002
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 teste.txt
sor@labsor:~$ rm teste.txt
sor@labsor:~$ ls
arquivo1.txt  dir_001  dir_002
sor@labsor:~$ rm -rf dir_002/
sor@labsor:~$ ls -l
total 8
-rw-r--r-- 1 sor sor   38 Abr  4 19:18 arquivo1.txt
drwxr-xr-x 2 sor sor 4096 Abr  4 19:13 dir_001
sor@labsor:~$ _
```

Obtendo ajuda sobre os comandos

Ao surgir alguma dúvida sobre a utilização de qualquer comando, podemos recorrer a algumas ferramentas, entre elas:

- As Páginas de Manual (*man pages*)
- Help on line dos comandos

Páginas de manual

As páginas de manual acompanham quase todos os programas GNU/Linux. Elas trazem uma descrição básica do comando/programa e detalhes sobre o funcionamento das opções.

A utilização da página de manual é simples, digite:

man [comando]

A navegação dentro das páginas de manual é feita usando-se as teclas:

- q - Sai da página de manual
- PageDown ou f - Rola 25 linhas abaixo
- PageUP ou w - Rola 25 linhas acima
- SetaAcima ou k - Rola 1 linha acima
- SetaAbaixo ou e - Rola 1 linha abaixo

Help on-line

Quase todos os comandos/programas GNU/Linux oferecem este recurso que é útil para consultas rápidas (e quando não precisamos dos detalhes das páginas de manual). É útil quando se sabe o nome do programa mas deseja saber quais são as opções disponíveis e para o que cada uma serve. Para acionar o help on line, digite:

[comando] --help

exemplo:

ls --help

Atividade

Pesquise, pratique e explique para que serve os comandos abaixo:

- **ls -la**
- **mkdir -p**
- **cp -r**
- **rm -rf**
- **touch**
- **cat**
- **more**
- **less**
- **head**
- **tail**