

Apostila - Introdução ao Linux

Capacitação em Bioinformática

Laboratório de Engenharia Biológica - Jan/2020

Ministrante: Lucas Pompeu Neves - lucaspompeuneves@gmail.com

Introdução

Linux é um sistema operacional de código aberto, tendo seu código e versão utilizável disponível na internet. O Linux foi criado em 1991 por *Linus Torvald* na Universidade de Helsinki, na Finlândia.



GNU/Linux é o nome do sistema operacional que usualmente conhecemos como Linux.



GNU: conjunto de ferramentas como shell, compiladores, editores desenvolvidas como alternativa livre às ferramentas do Unix.



Linux: kernel (núcleo) do sistema operacional



LINUX + GNU = GNU/Linux ou somente Linux

Dentre as principais características do Linux estão:

- Tipo Unix, sistema operacional base do Linux e responsável pelo suporte a multitarefas, multiusuários e portabilidade.
- Interpretador Bash, terminal para interpretação de comandos entre usuário e sistema operacional.
- Utiliza estrutura de permissões para execução, leitura e manipulação de arquivos, pastas e programas.
- Modularização, estrutura que carrega para memória principal apenas o que é utilizado durante o processamento, liberando a memória totalmente após a execução, permitindo uma leveza e baixo consumo de recurso computacional.

O Linux possui uma série de distribuições, possuindo as mais diversas aplicações, seja para utilização em servidores ou em computadores pessoais. O ramo da Bioinformática se utiliza do Linux para diversas aplicações, os principais motivos são sua natureza de código aberto, permitindo modificações para uso aplicado na área biológica, e sua robustez em distribuir e executar softwares, tanto em ambiente Web quando em ambiente de Linha de Comando, da área biológica.

Shell

O Shell é uma interface de usuário para acessar os serviços do sistema operacional, podendo existir como linha de comando ou como interface gráfica. Este curso abordará o Shell em linha de comando, conhecido como Bash, o interpretador do Linux, chamado de Terminal.

Existem algumas maneiras de abrir o terminal:

- A maneira mais rápida de abrir o Terminal é apertando *Ctrl + Alt + T* no teclado.
- Uma segunda maneira é apertando *Alt + F2* e logo em seguida escrever *gnome-terminal*.
- A terceira maneira é apertar a tecla *Win* digitar *terminal* e apertar a tecla *Enter*.

Principais comandos

Os comandos do Terminal do Linux possuem alguns padrões que devem ser anotados para futura utilização, estes são:

- Estrutura do comando
\$ [comando] [opções] [argumentos]
- Comando de ajuda
\$ [comando] -h | --help
- Comando para o manual
\$ man [comando]
Comando para versão
\$ [comando] -v | -V | --version

Arquivos e diretórios

Listagem:

ls - lista conteúdo do diretório

- `$ ls` (lista conteúdo do diretório)
- `$ ls -a` (lista todo o conteúdo existente, até o oculto)
- `$ ls -l` (lista os arquivos e suas respectivas permissões)
- `$ ls /var/` (lista o conteúdo do diretório /var de qualquer parte do sistema)
- `$ ls -R` (lista os diretórios recursivamente)

Navegação e busca:

cd - troca de diretório

- `$ cd [caminho]` (acessa diretório inferior e superior em relação ao atual)
- `$ cd /bin/` (acessa o diretório /bin de qualquer parte do sistema)

mv - movendo arquivos entre diretórios

- `$ mv [arquivo/diretório] [destino]` (movendo arquivos ou diretório)
- `$ mv [nome atual do diretório/arquivo] [novo nome do diretório/arquivo]` (renomeando arquivo ou diretório)

cp - copiando arquivos

- `$ cp [arquivo] [destino]` (copiar arquivos)
- `$ cp -r [diretório] [destino]` (copiar diretórios)

find - procura por arquivos em uma hierarquia de diretórios

- `$ find [nome]`
- `$ find -name *sh` (busca todos os arquivos que possuem extensão *.sh)

Leitura:

more - leitura de arquivos

- `$ more [opção] [arquivo]`
- `$ more arquivo.txt` (exemplo de leitura simples de arquivo)

Escrita:

mkdir - cria diretórios

- `$ mkdir [opção] [diretório]`
- `$ mkdir teste` (exemplo de criação de diretório nomeado como "teste")
- `$ mkdir teste/teste2`

touch - cria arquivos

- `$ touch [arquivo]`
- `$ touch arquivo.txt` (exemplo de criação de arquivo nomeado como “arquivo.txt”)
- `$ touch teste/teste2/arquivo.txt` (exemplo de criação de arquivo nomeado como “arquivo.txt” no diretório /teste/teste2/)

saída de outros programas

- `$ [comando] > [nome do arquivo]`
- `$ echo “bio” > arquivo.txt` (exemplo de escrita da palavra “bio” no arquivo “arquivo.txt”)
- `$ cat arquivo.txt > arquivo2.txt` (escrita no “arquivo2.txt” da leitura do arquivo “arquivo.txt”)
- `$ ls / > arquivo.txt`
- `$ ls /bin > arquivo.txt` (escrita no “arquivo.txt” da lista de arquivos no diretório “/bin”)

Exclusão:

rm - remove arquivos ou diretórios

- `$ rm [opção] [arquivo]`

remoção de arquivos

- `$ rm [nome do arquivo | padrão]`
- `$ rm arquivo.txt $ rm *txt`

remoção de diretórios

- `$ rm [-r | -R | --recursive] [diretório | padrão]`
- `$ rm -r teste`

Pacotes

Os pacotes são as peças que formam todas as distribuições Linux e podem conter programas, bibliotecas de sistema ou mesmo coisas como papéis de parede e ícones. Alguns programas grandes (como o KDE por exemplo) são divididos em vários pacotes para que você possa instalar apenas as partes que lhe interessam, ficando com um sistema mais enxuto. Alguns pacotes dependem de outros (um certo programa pode precisar de uma biblioteca que faz parte de outro pacote por exemplo), as chamadas dependências. Para evitar que você fique com coisas sem funcionar ou com pacotes desnecessários, o instalador automaticamente verifica as dependências de cada

pacote, adicionando ou removendo pacotes relacionados a ele. É por isso que às vezes ao marcar um determinado pacote alguns outros são marcados juntos.

Busca

- *\$ apt-cache search [Padrão]*

Instalação

- *\$ sudo apt-get install [nome do pacote]*

Remoção

- *\$ sudo apt-get remove [nome do pacote]*

Atualiza índices de origem dos pacotes

- *\$ sudo apt-get update*

Para atualizar as versões dos pacotes instalados (requer update)

- *\$ sudo apt-get upgrade*

Arquivos compactados

Arquivos ZIP

- *\$ unzip [parâmetro] [nome_do_arquivo_zipado]*
- *\$ unzip [nome_do_arquivo_zipado]*

Arquivos TAR

- *\$ tar -xvzf [nome_do_arquivo_zipado]*
onde,
x - extrai os arquivos
v - exibe o nome dos arquivos um a um
z - o formato é “gzip” (.tar.gz)
f - usa o seguinte arquivo tar para operação

Permissões Linux

No Linux, quando um arquivo ou diretório é criado, algumas permissões são atribuídas a ele automaticamente.

Essas permissões são divididas em três grupos: O usuário que criou o arquivo (usuário dono), o grupo dono do arquivo (que pode conter vários usuários), e as dos demais usuários que não pertencem ao grupo dono.

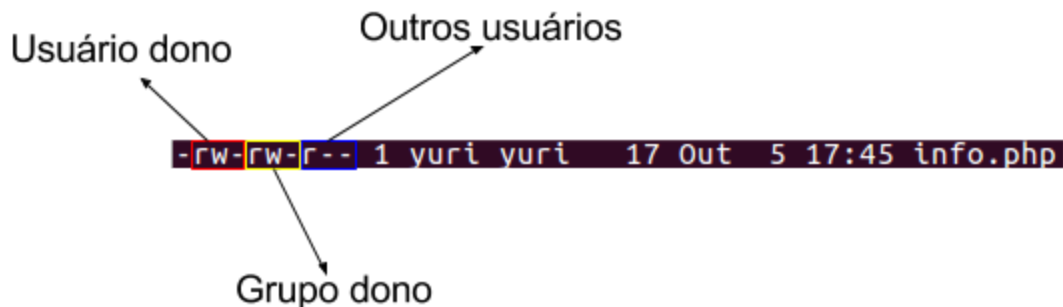
Essas permissões podem ser de leitura (r, read), escrita (w, write) e execução (x, execute).

A permissão de leitura (r) permite visualizar o conteúdo de um arquivo ou diretório, já a permissão de escrita (w) serve para alterar o conteúdo de um arquivo e diretório. A opção de execução serve para tornar o arquivo um executável, ou, em caso de diretórios, para poder acessá-lo e usar comandos, como o cd.

Visualizando permissões:

- `$ ls -l`

```
yuri@yuri-laptop:/var/www/html$ ls -l
total 8
-rw-rw-r-- 1 yuri yuri  17 Out  5 17:45 info.php
drwx----- 5 yuri yuri 4096 Feb 15 2017 loja
yuri@yuri-laptop:/var/www/html$
```



Alterando permissões:

- `$ chmod o=r bioinfo/` (dando permissão de leitura da pasta “bioinfo” a categoria de “outros usuários”)
- `$ chmod o+x bioinfo/` (adicionando permissão de execução da pasta “bioinfo” a categoria de “outros usuários”)

Executando um arquivo binário

Arquivos binários são arquivos já prontos em linguagem de máquina, onde basta executar no navegador.

Primeiro se deve dar a permissão de execução ao arquivo:

- `$ chmod +x [arquivo]`

Em seguida basta executar:

- `$./[arquivo]` ou `$ sudo ./[arquivo]`

Conectando a um servidor

De acordo com especialistas, SSH é “um protocolo para realizar login remoto e outros serviços na rede de forma segura em uma rede insegura”. Em outras palavras, nada mais é do que uma forma de entrar em um computador e acessar os serviços que ele tem a oferecer.

Instalando SSH

- `$ sudo apt-get install openssh-server`

Acessando SSH

- `$ ssh usuario@ip-alvo -p numerodaporta`

Saindo SSH

- `$ exit`

Copiando dados de servidores

O SCP (Secure Copy) é um protocolo de rede para transferências de arquivos. Com ele, você pode transferir arquivos de forma fácil e segura entre um local remoto e um host ou entre dois locais remotos. É uma maneira bem acessível de prevenir que seus dados fiquem expostos para outros usuários ao mesmo tempo em que preserva a confidencialidade. Na essência, o SCP é uma mistura de RCP e SSH (Secure Shell). Ele se baseia no primeiro para realizar ações de cópia. Enquanto que, no segundo, ele usa toda a parte de encriptação de informação para autenticar sistemas remotos.

Comando padrão:

- `$ scp [opções] [origem usuario@IP]:/[diretório e arquivo] [destino usuario@IP]:/[diretório e arquivo]`

Vamos supor que você está na máquina 200.239.92.999 usando o usuário “tony” no diretório “/home/Documents/” e um arquivo “stark.txt”, agora você quer passar esse arquivo para a máquina 200.239.92.333, na porta 22 usando o usuário “steve” na pasta /home/rogers”, o comando deverá ser assim:

```
$ scp -P 22 stark.txt steve@200.239.92.333:/home/rogers
```

E então digita-se a senha do usuário “steve”.

Agora você continua na máquina do “tony” e quer copiar o arquivo da máquina de “steve” para você. O comando deve ser:

```
$ scp -P 22 steve@200.239.92.333:/home/rogers/arquivo.txt .
```

