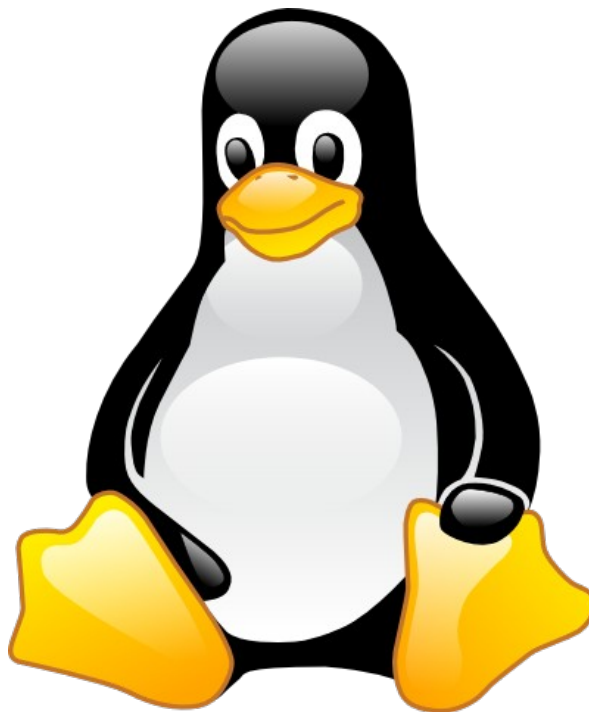


Linux System Administration 455



Aula 2 - 455 Intensivo



Aula 1 - 455 Intensivo

Uma coisa importante antes de mexer no Linux é que ele é Case Sensitive. Ou seja, ele diferencia letras maiúsculas de letras minúsculas.

Então, se eu tenho um arquivo chamado:

teste.txt

Ele é diferente do arquivo que chama:

Teste.txt

Aula 1 - 455 Intensivo

O Linux tem 2 tipos de comandos, os internos e os externos.

Os comandos internos são comandos que estão dentro do próprio Shell, que é o interpretador de comandos.

São os comandos essenciais do sistema.

A grande vantagem dos comandos internos é a velocidade, pois não precisam ser procurados no disco rígido, nem criam processos.

Aula 1 - 455 Intensivo

Comandos externos estão localizados em diretórios específicos no disco rígido, como /bin e /sbin.

Para um comando externo funcionar o Linux precisa consultar o disco rígido. A maioria dos comandos Linux é externa.

Exemplos:ls, cp, rm, mv, mkdir, rmdir.

Dica para descobrir a localização de um comando no disco rígido:

\$ which comando

Exemplo:

\$ which ls

/bin/ls

Aula 1 - 455 Intensivo

Quando o prompt que anteceder um comando for um \$ (cifrão), o comando poderá ser executado por qualquer usuário.

Quando o prompt que anteceder um comando for uma # (cerquilha, joga da velha etc...), o comando poderá ser executado somente pelo superusuário, conhecido como root.

Aula 1 - 455 Intensivo

Crie um diretório chamado "teste" em seu /home.
Por exemplo /home/maria:

```
$ cd /home/maria
```

```
$ pwd
```

```
/home/maria
```

```
$ mkdir teste
```

Aula 1 - 455 Intensivo

Eu também posso criar um diretório se eu estiver em outro diretório, mas eu precisarei indicar o caminho completo onde ficará esse novo diretório.

Exemplo:

```
$ pwd
```

```
/tmp
```

```
$ mkdir /home/leo/teste
```


Aula 1 - 455 Intensivo

Dica: Você pode usar a tecla TAB do teclado para auto-completar um comando, assim você não precisa ter que fica digitando ele todo!

```
$ mkd<TAB>
```

Dica: Se estou no diretório /home/leo e vou para o diretório /tmp com:

```
$ cd /tmp
```

Se eu digitar o comando abaixo eu volto para o diretório que eu estava antes:

```
$ cd -
```

Como posso usar o comando cd para ir direto para o home do usuário corrente sem colocar o caminho completo?

```
$ cd ~
```

A vantagem é que você não vai precisar mais digitar todo o caminho /home/maria para ir até seu diretório pessoal.

Aula 1 - 455 Intensivo

Crie os arquivos no diretório teste, cada um chamado assim: arq1, arq2, arq3, sessao1, sessao2 sessao3 sapo satisfacao

```
$ cd /home/maria/teste
```

```
$ pwd
```

```
/home/maria/teste
```

```
$ touch arq1
```

```
$ touch arq2
```

```
$ touch arq3
```

```
$ touch sessao1 sessao2 sessao3 sapo satisfacao (Posso criar tudo numa linha de comando só)
```

Aula 1 - 455 Intensivo

Para alterarmos a data e hora que esse diretório foi criado, também usamos o comando touch. Por exemplo:

```
$ touch aniversario
```

```
$ touch -t 200809161940 aniversario
```

A opção -t é para escolher o tempo que vamos alterar

200809161940 tem o seguinte formato AAAAMMDDHHHH

A - ano, M - mes, D - dia, H - hora

Aula 1 - 455 Intensivo

Como eu posso criar um arquivo oculto no linux?

```
$ touch .arq4
```

E como poderei visualizá-lo? Será que o comando ls sozinho consegue listar arquivos ocultos? Ou devo passar algum parâmetro?

```
$ ls -a
```

```
.arq4
```

Aula 1 - 455 Intensivo

Ok, agora mudei de idéia e quero que o arquivo não seja mais oculto. O que posso fazer para que isso aconteça?

```
$ mv .arq4 arq4 (basta renomear o arquivo)
```

Para remover o arquivo:

```
$ rm arq4
```

Aula 1 - 455 Intensivo

Para copiar um diretório e seu conteúdo precisamos usar o parâmetro R com comando cp:

```
$ cd /home/leo
```

```
$ pwd
```

```
/home/leo
```

```
$ cp -R teste /tmp
```

```
$ cd /tmp
```

```
$ ls
```

```
teste
```

```
$ cd teste
```

```
$ ls
```

```
arq1 arq2 arq3 sessao1 sessao2 sessao3 sapo satisfacao
```

Aula 1 - 455 Intensivo

Metacacteres são caracteres que possuem significado especial para o Shell.

Os principais metacaracteres são:

*

?

[] (listas)

[a-z] (listas)

Aula 1 - 455 Intensivo

```
$ cd /home/leo/teste
```

```
$ ls
```

```
arq1 arq2 arq3 sessao1 sessao2 sessao3 sapo satisfacao
```

```
$ ls arq*
```

```
arq1 arq2 arq3
```

```
$ ls sess*
```

```
sessao1 sessao2 sessao3
```

```
$ ls *1
```

```
arq1 sessao1
```


Aula 1 - 455 Intensivo

Lista:

```
$ ls sessao[13]
```

```
sessao1 sessao3
```

```
$ ls sessao[1-3]
```

```
sessao1 sessao2 sessao3
```

```
[1-3] = [123]
```

Aula 1 - 455 Intensivo

Missão Online

A missão é a seguinte:

Lembrando que os arquivos que temos no diretório são:

arq1 arq2 arq3 sessao1 sessao2 sessao3 sapo satisfacao

Liste apenas os arquivos que começam com a letra "s", a segunda letra pode ser de "a" até "e" e tem que terminar com a letra "o".

Conversem entre vocês e coloquem aqui no chat, como ficaria sua linha de comando.

5 minutos!

Aula 1 - 455 Intensivo

Resposta:

```
$ ls s[a-e]*o
```

Aula 2 - 455 Intensivo

Crie um diretório chamado arquivos em /tmp:

```
$ cd /tmp
```

```
$ mkdir arquivos
```

```
$ cd arquivos
```

```
$ touch não nao
```

```
$ ls n[ãa]o
```

```
não nao
```

Aula 2 - 455 Intensivo

```
$ touch teste Teste
```

```
$ ls [Tt]este
```

```
Teste teste
```

```
$ touch barata batata
```

```
$ ls ba?ata
```

```
barata batata
```

Asterisco corresponde a todos os caracteres;

O ponto de interrogação corresponde a um único carácter;

Aula 2 - 455 Intensivo

O que o comando `ls` faz quando usamos junto à ele o parâmetro `-l`?

Mostra detalhes de um arquivo. Exemplo:

```
$ cd /home/maria
```

```
$ ls -l
```

```
drwxr-xr-x 2 linux linux 1024 2007-07-11 15:37 teste
```

```
-rw-r--r-- 1 linux linux 1569 2007-07-09 11:09 instrucoes_iniciais
```

```
-rw-r--r-- 1 linux linux 112669 2007-07-09 12:52 intro_firewall.pdf
```

Aula 2 - 455 Intensivo

O primeiro caractere da saída indica o tipo de um arquivo:

- arquivo texto comum

d diretório

l link

Aula 2 - 455 Intensivo

Para remover o diretório teste:

```
$ cd /home/leo
```

```
$ pwd
```

```
/home/leo
```

```
$ rm -rf teste
```


Aula 2 - 455 Intensivo

Dica: Você pode usar vários comandos numa linha só ao invés de digitar ENTER a cada comando.

Exemplo:

```
$ cd /home/leo; mkdir guarda_roupa; cd guarda_roupa; touch camiseta bermuda;
```

O ponto-e-vírgula nesse caso funciona como ENTER.

Temos os comandos para localizar arquivos.

Por exemplo, o comando locate e o comando find

Aula 2 - 455 Intensivo

O comando find é um comando que procura arquivos pelo nome ou pelas suas características.

Vamos iniciar uma pesquisa em um arquivo, mostrando um diretório específico.

Por exemplo:

```
$ find /etc -name passwd
```

Aula 2 - 455 Intensivo

Outro exemplo:

```
$ find / -name *.sh
```

Estou indicando que ele irá buscar a partir do / (barra), ou seja, no sistema inteiro qualquer arquivo * (asterisco) que tem a extensão.sh

Como isso faz parte do nome dos arquivos, usamos a opção -name também!

No caso do comandos find, quando usamos a expressão -name ele procura por nomes de arquivos que indicamos nos comandos.

Aula 2 - 455 Intensivo

```
$ find . -size -1024k
```

Com esse comando, estou indicando que ele tem que procurar (find), dentro do diretório que estou (.) arquivos e diretórios com tamanho (size) menor que 1024k.

Aula 2 - 455 Intensivo

```
$ find / -size +1024k
```

Aula 2 - 455 Intensivo

Para criar a base de dados:

```
# updatedb
```

Esse comando irá demorar um pouco, porque ele vai ler todo o seu sistema e atualizar a base de dados.

Aula 2 - 455 Intensivo

Depois que atualizamos a base de dados, podemos fazer a busca por um arquivo qualquer.

Por exemplo:

```
$ locate passwd
```

Saída do comando:

Ele mostrará todos os locais no sistema que tem algum arquivo ou diretório chamado passwd.

Aula 2 - 455 Intensivo

O man é dividido em 8 sub-seções.

Elas são:

- 1 - comandos de usuários - comandos que podem ser executados a partir de um shell;
- 2 - chamadas do sistema - funções executadas pelo kernel;
- 3 - bibliotecas de funções - funções da biblioteca libc;
- 4 - formato de arquivo especiais - drivers e hardware;
- 5 - arquivos de configuração - formato de arquivos e convenções;
- 6 - jogos e demonstrações;
- 7 - pacotes de macro e convenções - sistema de arquivos, protocolos de rede, códigos ASCII etc;
- 8 - comandos de administração do sistema - comandos que o root pode executar.

Aula 2 - 455 Intensivo

Exemplos:

```
$ man ls
```

Para sair do man, aperte q, de quit.

Como eu faria para saber informações sobre o comando passwd?

```
$ man 1 passwd
```

E se eu quero saber informações sobre o arquivo de configuração passwd:, como ficaria o comando?

```
$ man 5 passwd
```

Para saber mais sobre expressões regulares:

```
$ man regex
```

Aula 2 - 455 Intensivo

Um link interessante para vocês olharem depois:

<http://www.linuxmanpages.com/>

Temos também o comando apropos. O apropos trabalha com um banco de dados, buscando uma descrição do comando.

Ele é muito usado em situações que você quer executar um comando, mas não lembra o nome ou a funcionalidade desse comando.

Aula 2 - 455 Intensivo

\$ apropos copy

Saída do comando:

COPY (7) [copy]	- copy data between a file and a table
bf_copy (1)	- shell script to copy a bogofilter working directory
Copy (7)	- copy data between a file and a table
Cp (1)	- copy files and directories
Cpgr (8)	- copy with locking the given file to the password or group file
Cpio (1)	- copy files to and from archives
Cppw (8)	- copy with locking the given file to the password or group file
Dd (1)	- convert and copy a file
Debconf-copydb (1)	- copy a debconf database
Install (1)	- copy MSDOS files to/from Unix
Mysqldhotcopy (1)	- a database backup program
Objcopy (1)	- copy and translate object files
Rcp (1)	- secure copy (remote file copy program)
rsync (1)	- a fast, versatile, remote (and local) file-copying tool
scp (1)	- secure copy (remote file copy program)
ssh-copy-id (1)	- install your public key in a remote machine's authorized_keys

Aula 2 - 455 Intensivo

\$ whatis ls

\$ whatis touch

\$ whatis mkdir

Aula 2 - 455 Intensivo

Para abrir a ajuda de um comando, usamos a seguinte sintaxe:

```
$ ls --help
```

```
$ mkdir --help
```

```
$ cat --help
```