

LXT0210-Criar, mover, copiar e apagar arquivos

Introdução

Esta lição aborda a gestão de arquivos e directórios no Linux usando ferramentas de linha de comando.

Um arquivo é uma colecção de dados com um nome e um conjunto de atributos. Se, por exemplo, você transferisse algumas fotos de seu telefone para um computador e lhes desse nomes descritivos, você teria uma porção de arquivos de imagem no computador. Esses arquivos têm atributos, como a data e hora em que o arquivo foi acessado ou modificado pela última vez.

Um directório é um tipo especial de arquivo usado para organizar arquivos. Uma boa maneira de pensar em directórios é compará-los às pastas suspensas usadas para organizar papéis em uma gaveta metálica. Porém, ao contrário das pastas de documentos em papel, podemos facilmente colocar directórios dentro de outros directórios.

Maiúsculas e minúsculas

Ao contrário do Microsoft Windows, os nomes de arquivos e directórios nos sistemas Linux diferenciam maiúsculas de minúsculas isso por o Linux ser **Case Sensivity**. Isso significa que os nomes `/etc/` e `/ETC/` designam directórios diferentes. Experimente os seguintes comandos:

```
idaseg@debian:/$ cd /
idaseg@debian:/$ ls
bin  etc      initrd.img.old  lib64      media  proc  sbin  tmp  vmlinuz
boot home    lib             libx32     mnt    root  srv   usr  vmlinuz.old
dev  initrd.img lib32          lost+found opt     run   sys   var
idaseg@debian:/$ cd ETC
-bash: cd: ETC: Ficheiro ou pasta inexistente
idaseg@debian:/$ pwd
/
idaseg@debian:/$ cd etc/
idaseg@debian:/etc$ pwd
/etc
```

O `pwd` mostra o directório em que você está no momento. Como vemos, tentar mudar para `/ETC` não funcionou, pois, esse directório não existe. Mas quando mudamos para o directório `/etc`, que existe, a operação foi bem-sucedida.

Criando `mkdir` (make directory – criar directório)

Como mostra a tradução, o comando `mkdir` é usado para criar directórios. Para exemplificar como se usa esse comando, criaremos um directório dentro do nosso directório pessoal:

```
idaseg@debian:/$ cd ~
idaseg@debian:~$ pwd
/home/idaseg
idaseg@debian:~$ mkdir idaleciosilvadocs
idaseg@debian:~$ ls
Desktop  Documentos  idaleciosilvadocs  Imagens  Músicas  Transferências
idaseg@debian:~$ cd idaleciosilvadocs/
idaseg@debian:~/idaleciosilvadocs$ pwd
/home/idaseg/idaleciosilvadocs
```

Comando **touch**

Sintaxe do comando: `$ touch + nome-do-arquivo-a-ser-criado`

```
idaseg@debian:~$ touch ida.txt
idaseg@debian:~$ ls
Desktop  Documentos  idaleciosilvadocs  ida.txt  Imagens  Músicas  Transferências
```

O comando **touch** cria comandos vazios. É importante perceber que os arquivos são criados pelos programas que trabalham com os dados armazenados nos arquivos. Se executarmos o **touch** em um arquivo existente, o conteúdo do arquivo não será alterado, mas o registo de data e hora de modificação do arquivo será actualizado.

Comando **mv** (Moving Directory – movendo directório)

O comando **mv** possui duas aplicações básicas — renomear e mover arquivos. Mover arquivos é retirar um arquivo de um directório para outro. Renomear é mudar o nome de um arquivo.

Renomeado um arquivo

Para entender a sintaxe do comando **mv** para renomear um arquivo, usemos o arquivo criado anteriormente, o arquivo **ida.txt**.

`$ mv nome-actual novo-nome`

Ao renomear um arquivo, o primeiro argumento é o nome actual, o segundo é o novo nome como mostra a imagem:

```
idaseg@debian:~/idaleciosilvadocs$ ls
cibersegurança  gestão-de-projectos  ida.txt  linux
idaseg@debian:~/idaleciosilvadocs$ mv ida.txt idaleciosilva.txt
idaseg@debian:~/idaleciosilvadocs$ ls
cibersegurança  gestão-de-projectos  idaleciosilva.txt  linux
```

Como se pode ver, após listar o conteúdo do directório vemos um arquivo **ida.txt**. Após executar o comando **mv** o arquivo deixou de ser **ida.txt** e passou a ser **idaleciosilva.txt**

OBS: deve-se ter cuidado ao usar o comando **mv**. Se o novo nome já estiver atribuído a um arquivo existente, ele será sobrescrito.

Movendo arquivos

Os arquivos são movidos de um directório para outro com o comando **mv**, como dito anteriormente e como se observa na sintaxe do comando.

Sintaxe do comando: `$ mv nome-do-arquivo directório-de-destino`

```
idaseg@debian:~$ ls
Desktop Documentos idaleciosilvadocs ida.text Imagens Músicas Transferências
idaseg@debian:~$ mv ida.text idaleciosilvadocs/
idaseg@debian:~$ ls idaleciosilvadocs/
ida.text
```

É importante referir que quando vamos mover arquivos para um directório externo, deve-se inserir o caminho absoluto do directório de destino como mostra a imagem abaixo.

```
idaseg@debian:~/idaleciosilvadocs$ ls
ida.text
idaseg@debian:~/idaleciosilvadocs$ mv ida.text /home/idaseg/Documentos/
```

Comando **rm** e **rmdir** (Remove, Remove directory – remover /Remover directório)

Sintaxe do comando: `$ rm + nome-do-arquivo`

```
idaseg@debian:~/Documentos$ ls
Analise-de-clientes.txt Auditoria ida.text
idaseg@debian:~/Documentos$ rm ida.text
idaseg@debian:~/Documentos$ ls
Analise-de-clientes.txt Auditoria
idaseg@debian:~/Documentos$ rm Auditoria/
rm: impossível remover 'Auditoria/': É pasta
```

Como se pode observar na sintaxe do comando, o **rm** serve para excluir arquivos. Após executar o comando **rm ida.txt** vimos que quando listamos o conteúdo do directório **Documentos**, o arquivo **ida.txt** não consta em **Documentos** isso significa que o comando **rm ida.txt** foi executado com êxito.

Mas, quando tentamos usar o comando **rm** para eliminar o directório **Auditoria**, vemos a seguinte informação: **rm: impossível remover 'Auditoria/': É pasta**.

Fica claro que o comando **rm** não elimina o directório. Para eliminar um directório usa-se o comando **rmdir**.

```
idaseg@debian:~/Documentos$ ls
Analise-de-clientes.txt Auditoria
idaseg@debian:~/Documentos$ rmdir Auditoria/
rmdir: falha ao remover 'Auditoria/': A pasta não está vazia
```

Como a imagem mostra, o **rmdir** não exclui um directório que não esteja vazio. Veja agora como o comando **rmdir** remove um dos directórios vazios do directório **home**.

```
idaseg@debian:~$ ls
Desktop Documentos idaleciosilvadocs Imagens Músicas Transferências
idaseg@debian:~$ rmdir Transferências/
idaseg@debian:~$ ls
Desktop Documentos idaleciosilvadocs Imagens Músicas
```

A imagem mostra que o directório Transferências que é um dos que subdirectórios do directório **home**.

Excluir um grande número de arquivos ou estruturas de directórios profundas, com muitos subdirectórios, pode parecer uma tarefa entediante, mas, na verdade, é fácil. Por padrão, o **rm** funciona apenas em arquivos regulares. A opção **-r** é usada para anular esse comportamento. Deve-se ter cuidado ao usar o **rm -r**. Quando usamos a opção **-r**, o **rm** não somente exclui qualquer directório, como também tudo o que estiver dentro desse directório, incluindo os subdirectórios e seu conteúdo.

OBS: para evitar que aconteçam erros o **rm** tem uma opção **-i** para exibir um aviso antes de executar qualquer acção.

```
idaseg@debian:~$ ls idaleciosilvadoes/  
cibersegurança gestão-de-projectos linux  
idaseg@debian:~$ rm -ri idaleciosilvadoes/  
rm: entrar na pasta 'idaleciosilvadoes/'? y  
rm: remover ficheiro normal vazio 'idaleciosilvadoes/gestão-de-projectos'? y  
rm: remover ficheiro normal vazio 'idaleciosilvadoes/linux'? y  
rm: remover ficheiro normal vazio 'idaleciosilvadoes/cibersegurança'? y  
rm: remover pasta 'idaleciosilvadoes/'? y
```

Comando **cp** (Copy – Copiar)

Sintaxe do comando: **\$ cp arquivo/directório-a-copiar directório-destino**

O comando **cp** é usado para copiar arquivos e directórios.

```
idaseg@debian:~/idaleciosilvadoes$ ls  
cibersegurança gestão-de-projectos linux  
idaseg@debian:~/idaleciosilvadoes$ cp linux /home/idaseg/Documentos/  
idaseg@debian:~/idaleciosilvadoes$ cd .. && ls Documentos/  
Analise-de-clientes.txt Auditoria linux
```

Se o último argumento for um directório, o **cp** criará uma cópia dos argumentos anteriores no directório. Como no caso de **mv**, vários arquivos podem ser especificados em simultâneo, desde que o destino seja um directório.

O comando **cp**, por padrão, só funciona em arquivos individuais. Para copiar um directório, usamos a opção **-r**. Lembre-se de que a opção **-r** fará com que **cp** também copie o conteúdo do directório que está sendo copiado:

```
idaseg@debian:~$ ls  
Desktop Documentos idaleciosilvadoes Imagens Músicas Transferências  
idaseg@debian:~$ cp -r idaleciosilvadoes/ Documentos/  
idaseg@debian:~$ ls Documentos/  
Analise-de-clientes.txt Auditoria idaleciosilvadoes linux
```

Conclusão

Os comandos apresentados nesse artigo são de grande importância para usuários Linux e até mesmo para quem administra redes e sistema Linux. Por isso, a observância a esses comandos ajudaram a quem aplica ou usa a automatizar processos para reduzir esforços e tempo.

Se gostou desses comandos, ainda há outros artigos que possam lhe interessar.