

# Gestión de archivos y directorios

Desde siempre que existió Linux se ha tenido que interactuar con el Shell. Es una herramienta que nos permite administrar archivos, texto, programas, procesos entre otras.

Y una forma fácil podríamos decir que todo en Linux es un archivo. Inclusive el Flash Drive que conectemos en la computadora es un archivo. Es cierto que profundiza más allá de eso pero por el momento podemos quedarnos con ese concepto.

Hay miles de comandos que podemos utilizar y la combinación con cada programa incrementa exponencialmente la cantidad de variaciones que podemos hacer. Pero para efectos de nuestro trabajo diario como administradores de un sistema podríamos reducirlo a unos pocos cientos.

## [Administrando archivos y directorios]

Los archivos en un sistema Linux son guardados como una única estructura.

Normalmente lo primero que se aprende cuando trabajamos con Linux es navegar entre los directorios, y manejar archivos.

Así que para efectos de este video podemos comenzar por ahí:

```
pwd

ls

ls --help

[]- sintaxis=
  []ls [OPTION]... [FILE]...

      entre corchetes se llaman argumentos. [OPTION] significa que es opcional
      [FILE] significa que podemos un directorio al comando o al archivo para ver la metadata.

ls -l (organizado ascendentemente)
ls -lr (organizado descendientemente)

ls -lh vs ll (is an alias)

man ls
```

Un alias es una representación corta de un comando que usamos muy seguido, por lo que tiene sentido para simplificar las cosas al máximo.

## [vista grafica]

Si estamos trabajando con directorio los archivos ayuda mucho sobre las cosas gráficamente. Si no tenemos una interfase gráfica, todavía tenemos la oportunidad de ver las cosas en una forma pseudo gráfica utilizando el comando tree (tres es la traducción en inglés para árbol)

Eso significa que de una forma ramificada podemos ver los directorios y archivos dentro de ellos.

```
tree
```

## [espacios en los nombres de un directorio]

En Linux el uso de la barra invertida se debe a que Shell utiliza los espacios como separadores entre argumentos de un comando.

Para que Shell interprete correctamente el nombre de un archivo o directorio con espacios como un solo argumento, es necesario identificar esos espacios con el símbolo `(\)`

```
cd Directorio\ con\ ejemplos
```

Para que no tengan que escribir manualmente algo tan confuso, Pueden utilizar la tecla tab `cd`  
`Dire<Tab>`

Alternativamente los directorios se pueden manejar con 'comillas' simples o "comillas" dobles

```
cd "Directorio con ejemplos"
```

```
cd 'Directorio con ejemplos'
```

## [change directory]

Si estamos cambiando de directorio y anteriormente no habíamos corrido el comando LS, y no recordamos cuál es el siguiente directorio al que queremos viajar, mientras corremos el comando CD podemos apretar la tecla tab para que nos sugiere cuál es el siguiente directorio de esa lista

```
→ ~ cd 'directorio con ejemplos'<tab>
documentos/  informacion/  libros/
```

## [touch]

Usando el comando Touch se puede crear un archivo o múltiples archivos.

Para crear múltiples archivos lo único que tenemos que hacer es listarlos: `→ ~ touch archivo1.txt`  
`archivo2.txt archivo3.txt`

```
demo@ubuntu20:~$ mkdir directorioejemplo && touch archivocreado.txt
demo@ubuntu20:~$ ls
archivocreado.txt  'directorio con ejemplos'  directorioejemplo

demo@ubuntu20:~$ rm -r directorioejemplo/ && rm archivocreado.txt

demo@ubuntu20:~$ mkdir directorioejemplo && touch directorioejemplo/archivocreado.txt
demo@ubuntu20:~$ ls
'directorio con ejemplos'  directorioejemplo

demo@ubuntu20:~$ tree
```

## [rm and rmdir]

---

Revisión #1

Creado 18 noviembre 2024 02:34:54 por Greivin

Actualizado 18 noviembre 2024 02:35:50 por Greivin