

# >Linux Basic Commad Lines



# Linux

---

El termino Linux se refiere al Kernel creado por Linus Torvalds y a todo el software que es parte de este ecosistema.

El Shell es una interfaz de computadora hacia un SO, la Shell toma Nuestros comandos de la Terminal y se los da al SO para ejecutarlos, se llama Shell puesto a que es la capa exterior alrededor de SO

En el ecosistema de Linux, el Shell default es BASH este corriendo en todas las versiones de unix y sistemas como Unix.



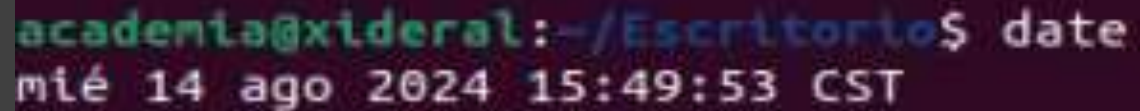
# Terminal

---

En la terminal hay varios comandos que nos permiten interactuar con Shell.

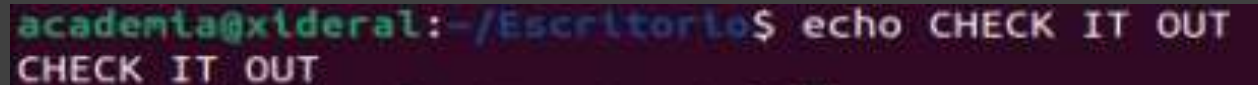
Algunos comandos simples son:

**date** nos regresa la fecha actual

A terminal window with a dark background. The prompt is 'academia@xideral:~/Escritorio\$'. The command 'date' has been entered, and the output is 'mié 14 ago 2024 15:49:53 CST'.

```
academia@xideral:~/Escritorio$ date  
mié 14 ago 2024 15:49:53 CST
```

**echo** el cual toma los argumentos que le demos y los imprime

A terminal window with a dark background. The prompt is 'academia@xideral:~/Escritorio\$'. The command 'echo CHECK IT OUT' has been entered, and the output is 'CHECK IT OUT'.

```
academia@xideral:~/Escritorio$ echo CHECK IT OUT  
CHECK IT OUT
```

No todos los comandos son sensibles a Caps pero muchos lo son y la manera mas rápido de usarlos es asumir que todos lo son.

# Opciones

---

Algunos comandos tienen una variedad de opciones que podemos usarlos para modificar su funcionamiento. En estos también es importante el CAPS pues son sensibles a esto.

Por ejemplo para el comando `date` podemos usar **`date -u`** para obtener el Coordinated Universal Time (UTC) pero esta opción también tiene la opción de usar el formato de palabra completo para que no sea tan confuso ósea que **`date --universal`** data el mismo resultado que `date -u`.

Esto también funciona cuando se quieren usar varias opciones en un comando `sort -ru` pero no se pueden combinar de la misma manera que la versión corta, `sort -reverse -unique` dará el mismo resultado que `sort -ru`.

```
/Escritorio
academia@xideral:~/Escritorio$ date --universal
mié 14 ago 2024 22:06:57 UTC
academia@xideral:~/Escritorio$ date -u
mié 14 ago 2024 22:07:08 UTC
academia@xideral:~/Escritorio$
```

# Man Pages

Las man pages son manuales cortos de documentación dentro de los sistemas operativos Unix y parecidos. Para leer estas se puede usar el **man command**. Que da informacion sobre cualquier comando que se necesite y una vez que se lea todo o se quiera salir solamente se tiene que escribir la tecla "q".

Tambien se mostrara toda la información referente a todas las opciones del comando elegido. El manual se separa en 8 secciones diferentes cada una cubriendo una parte a fondo y se pueden elegir en el mismo comando como por ejemplo.: **man 1 sort**

1. Comandos de usuario
2. Llamadas al sistema
3. Funciones de la librería C
4. Archivos especiales
5. Formas de archivo
6. Juegos
7. Misc
8. Comandos del admin del sistema

```
academia@xideral: ~/Escritorio
ECHO(1)                                User Commands                                ECHO(1)

NAME
    echo - display a line of text

SYNOPSIS
    echo [SHORT-OPTION]... [STRING]...
    echo LONG-OPTION

DESCRIPTION
    Echo the STRING(s) to standard output.

    -n      do not output the trailing newline
    -e      enable interpretation of backslash escapes
    -E      disable interpretation of backslash escapes (default)

    --help  display this help and exit
    --version
             output version information and exit

    If -e is in effect, the following sequences are recognized:

    \\      backslash
    \a      alert (BEL)
    \b      backspace
    \c      produce no further output
    \e      escape
    \f      form feed
    \n      new line

Manual page echo(1) line 1 (press h for help or q to quit)
```

# Man Pages



Para buscar el manual se puede usar la opción **-k**

**man -k date**

```
academia@xideral:~/Escritorio$ man -k date
update-alternatives (1) - Gestión de enlaces simbólicos que determinan ór...
update-passwd (8) - Actualiza /etc/passwd, /etc/shadow y /etc/group de fo...
aa-features-abi (1) - Extract, validate and manipulate AppArmor feature abis
apport-bug (1) - file a bug report using Apport, or update an existing...
```

El comando **type** nos dará información sobre el tipo de un comando.

**type date**

El comando **which** nos dará la localización exacta de un ejecutable, solamente funciona para ejecutables no para comandos o alias.

**which command**

El comando **help** nos ayuda con comandos que no tienen man pages porque están directo en el Shell para encontrar documentación de estos comandos podemos usarlo.

**help cd**

```
basin: type: date: no encontrado
academia@xideral:~/Escritorio$ type date
date está asociado (/usr/bin/date)
academia@xideral:~/Escritorio$ help cd
cd: cd [-L|[-P [-e]]] [dir]
Modifica el directorio de trabajo del shell.

Modifica el directorio actual a DIR. DIR por defecto es el valor de la
variable de shell HOME.
```

# Navegación



Todo empieza en el root directory para el sistema de archivos y su nombre actual es "/" después sigue el **/home** directory que contiene el folder para todos los usuarios del sistema como un ejemplo para mi el folder seria **/home/ever**.

**/ - root**

**~ - home**

En caso de que se quiera saber donde esta uno siempre se puede usar el siempre útil comando print working directory

**pwd**

```
academia@xideral:~/Escritorio$ pwd  
/home/academia/Escritorio
```



# Navegacion

---

El comando de listado **ls** dara una lista de todos los contenidos dentro de un directorio, ya sea usándolo sin contexto este imprimirá los contenidos del directorio actual o si se le otorga una ruta este dara el contenido del directorio otorgado como por ejemplo:



```
academia@xideral:~/Escritorio$ ls -a
.
..
numeros.txt
```

**ls** tiene un monton de opciones —a que es de las mas comunes muestra los archivos ocultos en el directorio.

Si se quiere mover de lugar se puede usar el comando **cd** por ejemplo si se quiere mover a un directorio en especifico se usa con una ruta **cd /home/ever** o si se quiere mover al directorio padre podemos usar **cd..** Para ir a /home puesto que los dos puntos representan el directorio padre.



# Creación de Archivos

---

Para simplemente crear un archivo o en caso de que ya exista, modificar la fecha de creación de una archiva a el tiempo actual podemos usar el **comando** touch Así:

**touch números.txt**

Si se quiere saber el tipo de archivo se puede usar el file command:

**file números.txt**

Para crear un directorio nuevo podemos usar el comando mkdir, en caso de querer crear varios folder nomas debemos poner los nombres en el comando, ósea para crear 2 folders llamados números y letras podemos usarlo así:

**Mkdir números letras**

# Alternado archivos

---

El comando **rm** sirve para remover archivos de nuestra maquina, no las recicla ni las oculta si se usa **rm** en un archivo este es totalmente borrado.

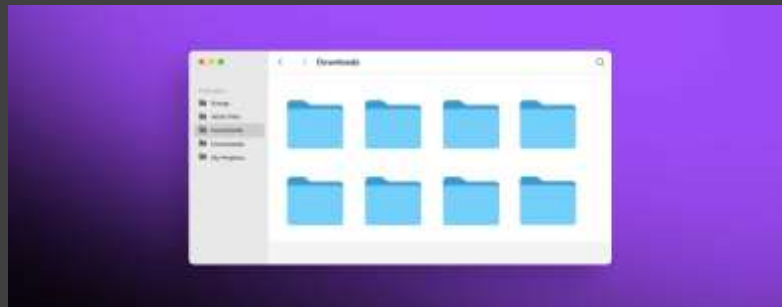
```
academia@xideral:~/Escritorio$ rm numeros.txt  
academia@xideral:~/Escritorio$ ls  
academia@xideral:~/Escritorio$
```

**rm números.txt**

Para remover directorios podemos usar el mismo comando pero para especificar podemos usar las banderas **-d** y **-r**:

**rm -d números** **-d** sirve para borrar un directorio Solamente si este se encuentra vacio.

**rm -r letras** **-r** sirve para borrar un directorio en caso de que tenga contenido o no.



# Alternado archivos

---

En caso de que se quiera mover un archivo o un directorio se puede usar el move command **mv**

**mv cinco.txt números/** Esto por ejemplo mover el archivo cinco.txt hacia el directorio números.

El mismo comando se puede usar para renombrar archivos o folders. Si quiero renombrar mi archivo cinco.txt a five.txt se puede usar así:

**mv cinco.txt five.txt**

Para crear una copia de un archivo se puede usar el copy command, por ejemplo si quiero crear una copia de mi numeros.txt y quiero llamar a la copia enteros.txt:

**cp números.txt enteros.txt**

O si quiero copiar estos dos archivos a mi directorio /math

**cp números.txt enteros.txt /math**

# Conclusiones

---

A un uso corto que le había dado a Linux, tenía un conocimiento básico de algunos comandos mas que nada para moverme entre carpetas pero nomas sabia su funcionamiento sin saber el porque de las cosas por ejemplo el cd command, no sabia que significaba change directory command, o que los dos puntos que usaba significaban el directorio padre.

Además de aprender comandos nuevos veo la importancia de saber la funcionalidad y el significado de estos mismos comandos, la facilidad del uso de Linux lo hace una herramienta poderosa a la hora de desarrollar.

