

Programa académico CAMPUS



Trainer
Ing. Carlos H. Rueda C.





INTRODUCCION A LINUX



[Link del video](https://youtu.be/Q9DLleO7ulk?si=YQZ6CSjwellIV-9l4)

<https://youtu.be/Q9DLleO7ulk?si=YQZ6CSjwellIV-9l4>

Características de Linux

- **64 bits**
- **Multitarea**
- **Multiusuario**
- **POSIX** : es un estándar de la industria, que asegura una calidad mínima en ciertas partes del S.O. y asegura su compatibilidad, a nivel de código, es decir, programas POSIX que funcionan en otros Unix.

Características de Linux

- **Estabilidad**
- **Es libre**
- **Soporte**
- **Adaptación**

Organizacion de directorios

- **/** : El directorio raíz, de este cuelgan todos los demás.
- **/sbin** : Contiene programas de administración indispensables
- **/bin** : Contiene los programas básicos para el uso normal
- **/root** : El home del administrador del ordenador
- **/tmp** : Es un directorio que todo el mundo puede usar como temporal, normalmente se borra su contenido cada cierto tiempo de manera automática
- **/lib** : Contiene las librerías necesarias para los programas

Organizacion de directorios

- **/etc** : Contiene toda la configuración del ordenador, es similar al registro de Windows, cualquier programa que este instalado tendrá un archivo en modo texto (esto es, que lo podemos editar con un editor de textos común como puede ser vi) en algún lugar de este directorio.
- **/proc** : Contiene información del funcionamiento del ordenador, prueba a entrar en el y ver con cat el contenido de los ficheros que tiene.

Organizacion de directorios

- **/lost+found** : Contiene los ficheros que se encuentran después de realizar una comprobación del disco, suelen aparecer después de fallos del PC.
- **/dev** : Contiene los dispositivos que tiene el ordenador disponibles.

Organizacion de directorios

- **/var** : Contiene datos sobre programas que son comunes a todos los usuarios, records de los juegos, la cola de impresión, los buzones de correo y los log (registros de sucesos en programas) de los demonios (programas que siempre están funcionando aunque no lo veamos ni los notemos).
- **/mnt /floppy /cdrom /dos** : Son directorios donde se suelen montar los sistemas de almacenamiento, depende de cada distribución o administrador.
- **/home** : De aquí cuelgan todos los home de los usuarios del sistema.

Organizacion de directorios

- **/usr** : Dentro de este directorio hay varios directorios no imprescindibles.
- **/usr/sbin** : Programas de administración indispensables
- **/usr/bin** : Programas de usuarios básicos
- **/usr/games** : Juegos y programas que pueden comprometer la seguridad, el superusuario no suele poder usarlos.
- **/usr/lib** : Librerías de los programas de /usr/xxx
- **/usr/man** : Páginas de los manuales

Organizacion de directorios

- **/usr/doc** : Documentación de todos los programas instalados
- **/usr/src** : Fuentes de programas para compilar (normalmente el kernel)
- **/usr/X11R6** : Es el sistema de ventanas X Windows system, dentro de él encontrarás man, bin, lib, etc...

Organizacion de directorios

- **/usr/local** : Este directorio es como /usr (tiene su bin, su lib, su man), pero se refiere a programas o datos propios de ese ordenador, es decir aquellos que no serán visibles en otros ordenadores conectados en red (es posible compartir el mismo disco duro entre varios ordenadores), de esta manera por ejemplo, da igual en cual te pusieses, siempre tendrías tu home y tus aplicaciones. Tambien es donde se instalan las cosas que se compilan en el ordenador, sin utilizar un sistema de paquetes y se quiere que esten disponibles para los usuarios.

¿Qué es bash?

Bash (abreviación de Bourne Again SHell) es una shell de Unix, y un intérprete de lenguaje de comandos. Una shell es simplemente un macroprocesador que ejecuta comandos. Es la shell más utilizada de forma predeterminada para la mayoría de distribuciones de Linux y es un sucesor de la shell Korn (ksh) y shell C (csh).

Muchas cosas que se pueden hacer en el sistema operativo Linux se pueden hacer a través de la línea de comandos.

Algunos ejemplos son...

- Editar archivos
- Ajustar el volumen del sistema operativo
- Obtención de páginas web
- Automatizar trabajo que realizas a diario

Usar bash en la línea de comandos (Linux, OS X)

Puedes comenzar a usar bash en la mayoría de los sistemas operativos Linux y OS X abriendo una terminal. Consideremos un ejemplo sencillo de "Hola Mundo!". Abre una terminal y escribe la siguiente línea (todo después del signo \$):

```
zach@marigold:~$ echo "Hola Mundo!"  
Hola Mundo!
```

Como puedes ver, usamos el comando echo para imprimir la cadena "Hola Mundo!" en la terminal.

Escribir un script de bash

También puedes poner un conjunto de comandos de bash en un archivo .sh y ejecutarlos desde la línea de comandos.

Supongamos que tienes un script bash con el siguiente contenido:

```
#!/bin/bash  
echo "Hola Mundo!"
```

Vale la pena señalar que la primera línea del script comienza con #!. Esta es una directiva especial que Unix trata de manera diferente.

¿Por qué usamos **#!/bin/bash** al principio de nuestro script?

Esto se debe a que es necesario permitir que el shell interactivo sepa qué tipo de intérprete ejecutar para el programa que contiene.

La primera línea le dice a Unix que el archivo será ejecutado por `/bin/bash`. Esta es la ubicación estándar del shell Bourne en casi todos los sistemas Unix.

Al agregar `#!/bin/bash` como la primera línea de tu script, le indicas al sistema operativo que invoque el shell especificado para ejecutar los comandos que siguen en el script.

¿Por qué usamos **#!/bin/bash** al principio de nuestro script?

#! solo se ejecutará si llamas tu script como un ejecutable. Por ejemplo, cuando escribes `./nombre-del-script.extension`, el sistema buscará en la línea superior del archivo para determinar el intérprete, mientras que, al ejecutar el script como `bash nombre_del_script.sh`, la primera línea se ignorará.

```
zach@marigold:~$ chmod u+x miScriptBash.sh
```

```
zach@marigold:~$ ./miScriptBash.sh  
Hola Mundo!
```

Línea de Comandos de Linux: bash cat

Cat es uno de los comandos más usados frecuentemente en los sistemas operativos Unix.

Cat es usado para leer un archivo secuencial-mente e imprimirlo en la consola de manera estándar. Su nombre es derivado de su función para **concatenar** archivos.

Uso

```
cat [opciones] [nombres_de_los_archivos]
```

Opciones más usadas:

- **-b**, numera líneas con contenido en la salida del comando
- **-n**, numera todas las líneas en la salida del comando
- **-s**, reduce múltiples líneas adyacentes sin contenido
- **-v**, muestra caracteres no imprimibles, excepto las pestañas y el carácter de final de línea

Ejemplo:

Imprime en la terminal el contenido del archivo.txt:

```
cat archivo.txt
```

Concatena el contenido de dos archivos y muestra su resultado en la terminal:

```
cat archivo1.txt archivo2.txt
```

Línea de Comandos de Linux: bash cd

Cambia Directorio a la ruta especificada, por ejemplo

```
cd proyectos.
```

Hay unos pocos, pero realmente útiles argumentos que ayudan a su uso:

- `.` hace referencia al directorio actual, tal como `./proyectos`
- `..` puede ser usado para subir un directorio, usa `cd ..`, y puede ser combinado para subir múltiples niveles `../../mi_directorio`
- `/` es el root o "raíz" de tu sistema el cual contiene directorios de importancia, tales como `system`, `users`, etc.
- `~` es el directorio home, usualmente ubicado en la ruta `/users/nombre-de-usuario`. Regresa a determinado directorio contenido en home, incluyendo al comienzo del comando, por ejemplo `~/proyectos`.

Línea de Comandos de Linux: bash head

Head se utiliza para imprimir las primeras diez líneas (por defecto) o cualquier otra cantidad especificada de líneas de un archivo o archivos. Cat se utiliza para leer un archivo secuencialmente e imprimirlo de manera estándar.

Si se llama a más de un archivo, se muestran las primeras diez líneas de cada archivo, a menos que se especifique un número determinado de líneas. La elección de mostrar el encabezado del archivo es opcional usando la siguiente opción

Uso

```
head [opciones] [nombre-de-archivo(s)]
```

Opciones más utilizadas:

- **-n N**, imprime las primeras N líneas de un archivo(s)
- **-q**, no imprime los encabezados de los archivos
- **-v**, siempre imprime los encabezados de los archivos

Ejemplo:

```
head file.txt
```

Imprime en la terminal las primeras diez líneas del archivo file.txt (por defecto)

```
head -n 7 file.txt
```

Imprime en la terminal las primeras siete líneas del archivo file.txt

```
head -q -n 5 file1.txt file2.txt
```

Imprime en la terminal las primeras 5 líneas de archivo file1.txt, seguidas de las primeras 5 líneas del archivo file2.txt

Línea de Comandos de Linux: bash ls

ls es un comando en sistemas operativos Unix para listar el contenido de un directorio, por ejemplo, nombres de carpetas y archivos.

Uso

```
ls [opciones] [nombre_de_los_archivos]
```

Opciones más usadas:

- **-a**, despliega todos los archivos y carpetas, incluidos los que están ocultos y comienzan con un **.**
- **-l**, despliega los archivos en formato de lista detallada
- **-G**, en un listado extenso, no imprime los nombre de los grupos

Ejemplo:

Listar los archivos contenidos en la ruta `freeCodeCamp/guide/`

```
ls                                                                    ◦ master
CODE_OF_CONDUCT.md bin package.json utils
CONTRIBUTING.md gatsby-browser.js plugins yarn.lock
LICENSE.md gatsby-config.js src
README.md gatsby-node.js static
assets gatsby-ssr.js translations
```

Línea de Comandos de Linux: bash man

Man, o abreviación de **man**ual, es un comando de bash usado para desplegar referencias de manuales en línea de determinado comando.

Man, muestra la página relativa de man (abreviatura de **man**ual **page**) del comando dado.

Uso

```
man [opciones] [comando]
```

Opciones más utilizadas:

- **-f**, imprime una breve descripción del comando dado
- **-a**, muestra, en sucesión, todas las introducciones de páginas de manuales disponibles del comando especificado.

Ejemplo:

Muestra el manual de ls:

```
man ls
```

Línea de Comandos de Linux: bash mv

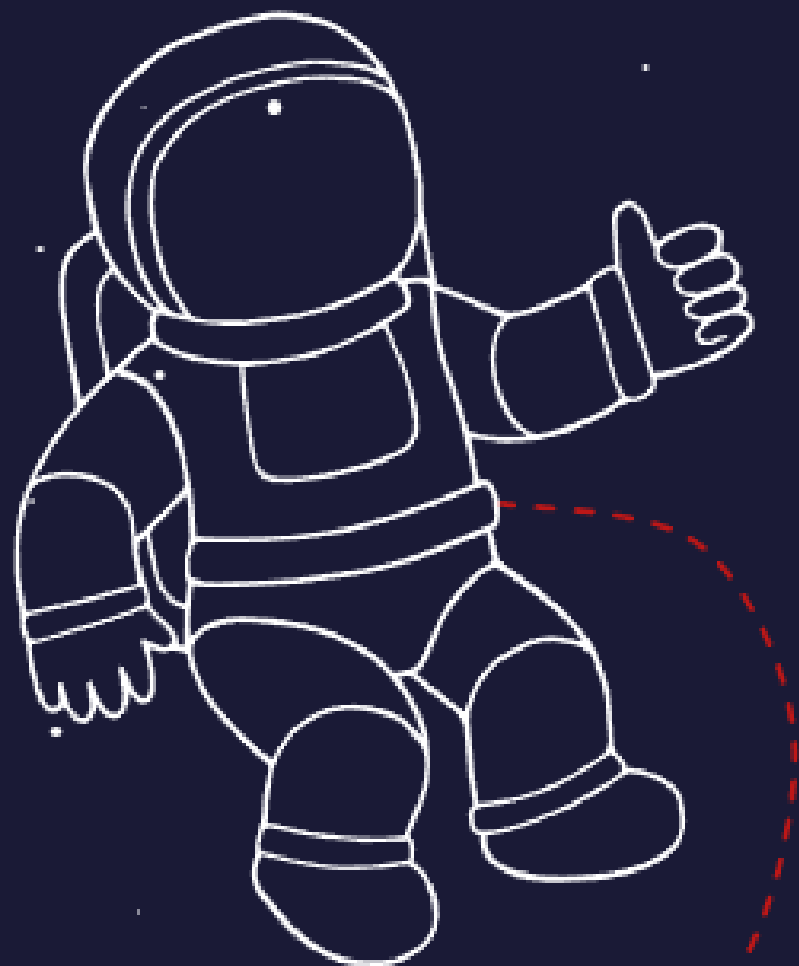
Mueve archivos y carpetas.

```
mv origen destino  
mv origen ... directorio
```

El primer argumento es el archivo el cual quieres mover, y el segundo es la ubicación a ser movido.

Opciones usadas comúnmente:

- `-f` para forzar mover y sobrescribir archivos sin confirmación del usuario.
- `-i` para solicitar confirmación antes de sobrescribir archivos.



Programa académico CAMPUS



Trainer
Ing. Carlos H. Rueda C.

