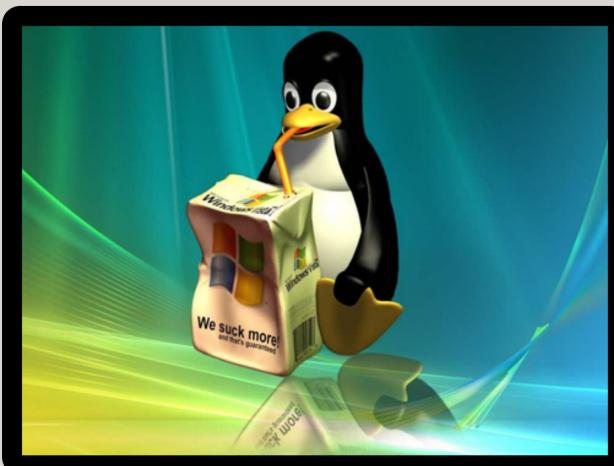




INSTITUTO POLITECNICO NACIONAL  
Centro de Estudios Cientificos Y Tecnológico No 3.

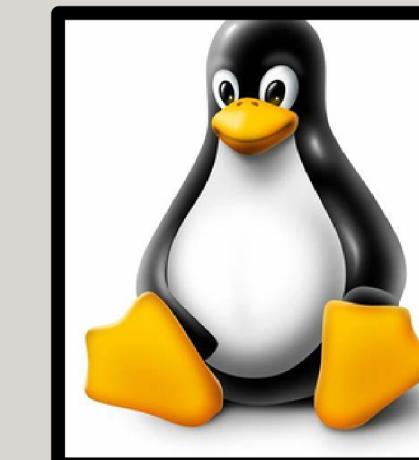


# FICHEROS



# Introducción

- El manejo de ficheros es una habilidad fundamental en Linux.
- Permite crear, modificar, eliminar y organizar archivos y directorios.
- Es esencial para navegar y administrar el sistema de archivos.





# ¿Qué es un fichero?

- Un fichero es una unidad de almacenamiento de datos en un sistema operativo.
- Puede contener texto, imágenes, audio, video u otro tipo de información.
- Cada fichero tiene un nombre único que lo identifica.

# Tipos de ficheros

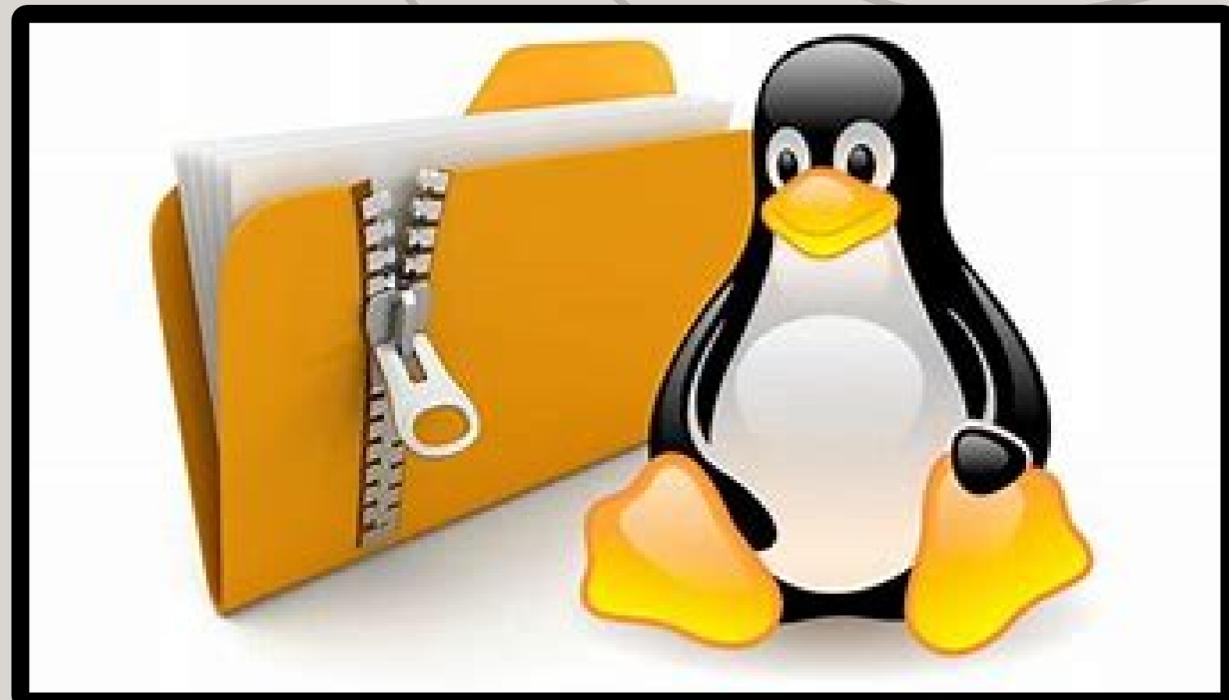
- Ficheros regulares: Contienen datos como texto, imágenes, etc.

- Directorios: Contienen otros ficheros y directorios.

- Enlaces simbólicos: Apuntan a otros ficheros o directorios.

- Dispositivos: Representan dispositivos de hardware.

- Tuberías y sockets: permiten la comunicación entre procesos.



# Directorios

01

Un directorio es un contenedor para ficheros y otros directorios.

02

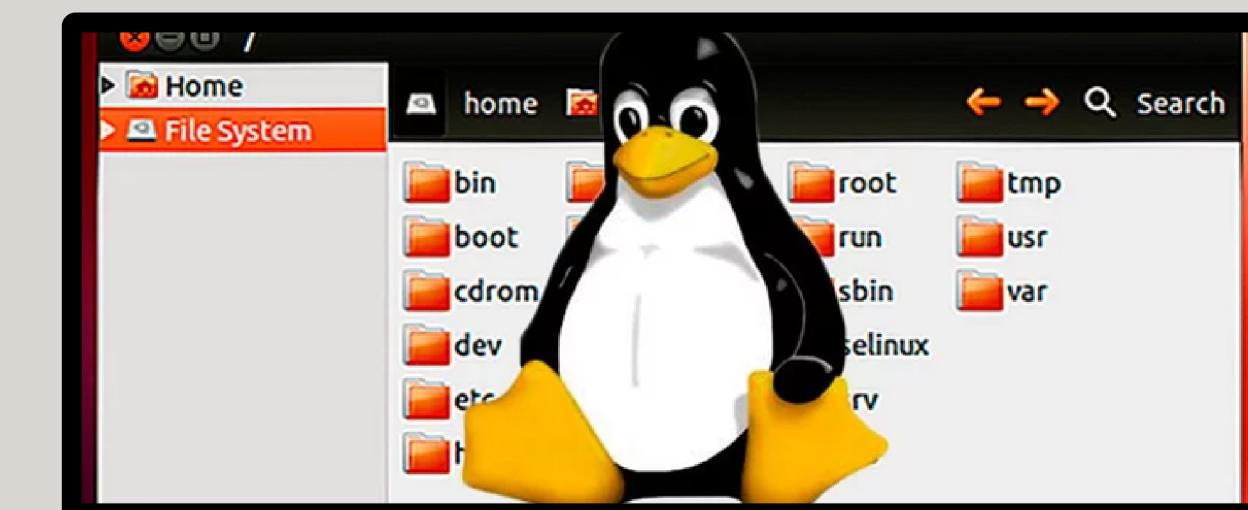
Permiten organizar y estructurar el sistema de archivos.

03

Cada directorio tiene un nombre único

- Existen directorios especiales como:

``/home``  
``/etc``  
``/var``



# Sistema de Ficheros

- El sistema de ficheros es la forma en que se organizan y almacenan los ficheros en el disco duro.



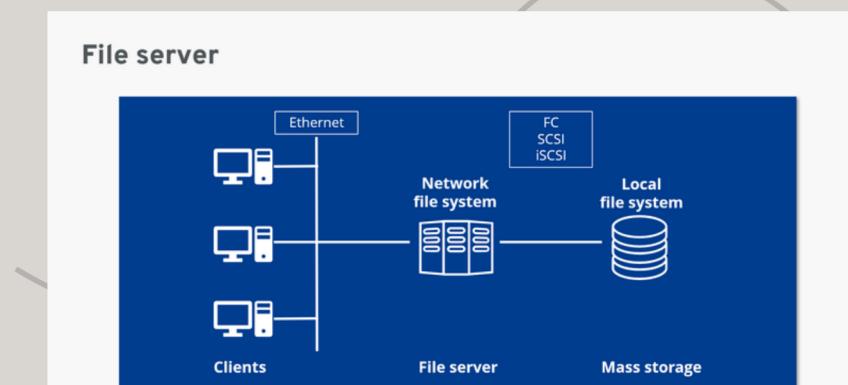
- En Linux, el sistema de ficheros se organiza en una estructura de árbol, con el directorio raíz `/` en la cima.
- Cada fichero y directorio tiene una ruta única que lo identifica dentro del sistema de archivos.
- Algunos sistemas de ficheros comunes en Linux son **ext4**, **XFS**, **Btrfs**, etc.

# Servidor de Ficheros

El servidor de ficheros es la capa de software del sistema operativo entre los dispositivos y los usuarios.

Sus objetivos son:

- Visión Lógica Simplificada: Presentar una vista lógica de los ficheros en los dispositivos de almacenamiento secundario.
- Esquema de Nombrado Lógico: Facilitar la identificación de los ficheros por parte de los usuarios.
- Visión Uniforme de Dispositivos: Presentar otros dispositivos y mecanismos de comunicación como ficheros especiales.
- Primitivas de Acceso: Ofrecer accesos cómodos e independientes de los detalles físicos y la estructura del sistema de ficheros.
- Mecanismos de Protección: Garantizar que los usuarios solo accedan a los ficheros para los que tienen permiso.

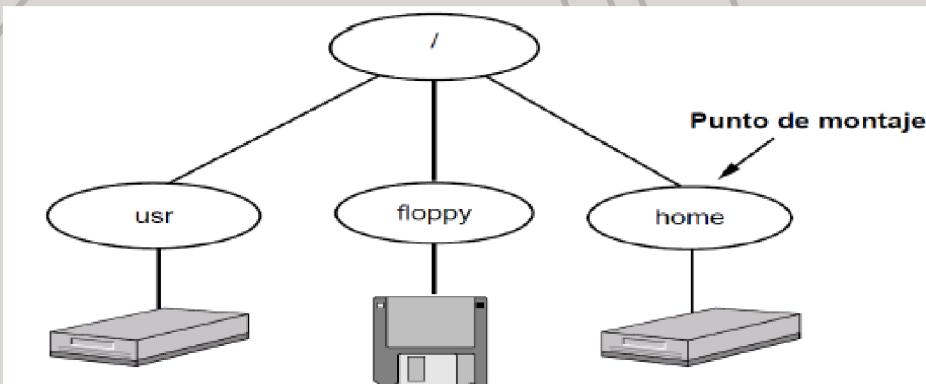


# Montado de sistema de ficheros

La operación de montaje integra la estructura jerárquica de un sistema de ficheros en un directorio (punto de montaje) del árbol de directorios. Esta operación está reservada al administrador o superusuario.

## Aspectos a Considerar:

- El nombre raíz del sistema de ficheros montado (SF2) se convierte en el punto de montaje (/usr), afectando a todos los nombres.
- El objeto /usr/ab queda oculto, pero no se borra. Si se desmonta SF2, vuelve a ser accesible.
- Al analizar una ruta que pasa por el nodo de montaje, se verifican los permisos del raíz montado, no del nodo de montaje original.
- Se pueden restringir los permisos de los ficheros montados, como conceder solo permiso de lectura o inhibir permisos de ejecución.



# Ficheros de inicio sesión en Linux

Los ficheros de script que se ejecutan al iniciar una sesión en Linux dependen de la distribución y del tipo de shell. Para el shell Bash, los ficheros básicos son:

Ficheros Comunes a Todos los Usuarios:

*/etc/profile: Se ejecuta al iniciar sesión cualquier usuario.  
Ejemplo:*

```
#!/etc/profile
# No core files by default
ulimit -S -c 0 > /dev/null 2>&1
# HOSTNAME is the result of running the hostname command
declare -x HOSTNAME=`/bin/hostname`
# No more than 1000 lines of Bash command history
declare -x HISTSIZE=1000
# If PostgreSQL is installed, add the Postgres commands to the user's PATH
if test -r /usr/bin/pgsql/bin ; then
declare -x PATH="$PATH":/usr/bin/pgsql/bin"
fi
# end of general profile
```

*/etc/bashrc: Se ejecuta cada vez que un usuario lanza bash de forma interactiva.*

Ficheros Privados de Cada Usuario

*~/.bash\_profile, ~/.bash\_login y ~/.profile: El que se ejecuta depende del siguiente pseudocódigo:*

```
IF ~/ .bash_profile exists THEN
    execute ~/ .bash_profile
ELSE IF ~/ .bash_login exists THEN
    execute ~/ .bash_login
ELSE IF ~/ .profile exists THEN
    execute ~/ .profile
END IF
```

*~/.bash\_logout: Se ejecuta cuando el usuario cierra la sesión.*



# Protección de ficheros

Se refiere a técnicas y mecanismos para proteger archivos contra modificaciones indebidas, accesos no autorizados o cualquier otra amenaza

## Permisos

se dividen en 3: escribir, leer y ejecutar.

Además de hay tres categorías de usuario: propietario, grupo y otros.



## Control de acceso

Linux también soporta Listas de Control de Acceso (ACL) que permiten definir permisos más específicos para distintos usuarios y grupos.



## Propiedad de archivos

Cada archivo tiene un propietario y un grupo asociado. Estos se pueden cambiar usando los comandos chown y chgrp



# Consideraciones de diseño

Es un aspecto crucial en la administración de sistemas y aplicaciones, y requiere una consideración cuidadosa para asegurar eficiencia, seguridad y facilidad de uso.

## ► Estructura y organización

Se necesita de una jerarquía con una buena organización y nombres además de separarlos por tipos y extensiones

## ► Acceso y seguridad

Aquí se asignan permisos y propietarios, la ACL y la encriptación de datos sensibles

## ► Rendimiento y eficiencia

Controlar el tamaño de los archivos y su fragmentación así como estructurarlos de manera adecuada para facilitar la navegación

## ► Rendimiento y eficiencia

los nombres tienen que ser legibles, fáciles de entender y recordar y consistente usando un esquema predecible

## ► Respaldo

Crear copias de seguridad para proteger contra la pérdida de datos y usar sistemas de control de versiones para gestionar y recuperar datos del archivo



# Sistemas de ficheros UNIX

Los sistemas de ficheros en Unix son una parte fundamental de la arquitectura del sistema operativo, proporcionando la estructura y las capacidades necesarias para almacenar, organizar y acceder a los datos.

► **UFS (Unix File System)**

► **EXT (Extended File System)**

► **Btrfs (B-tree File System)**

► **ZFS (Zettabyte File System)**

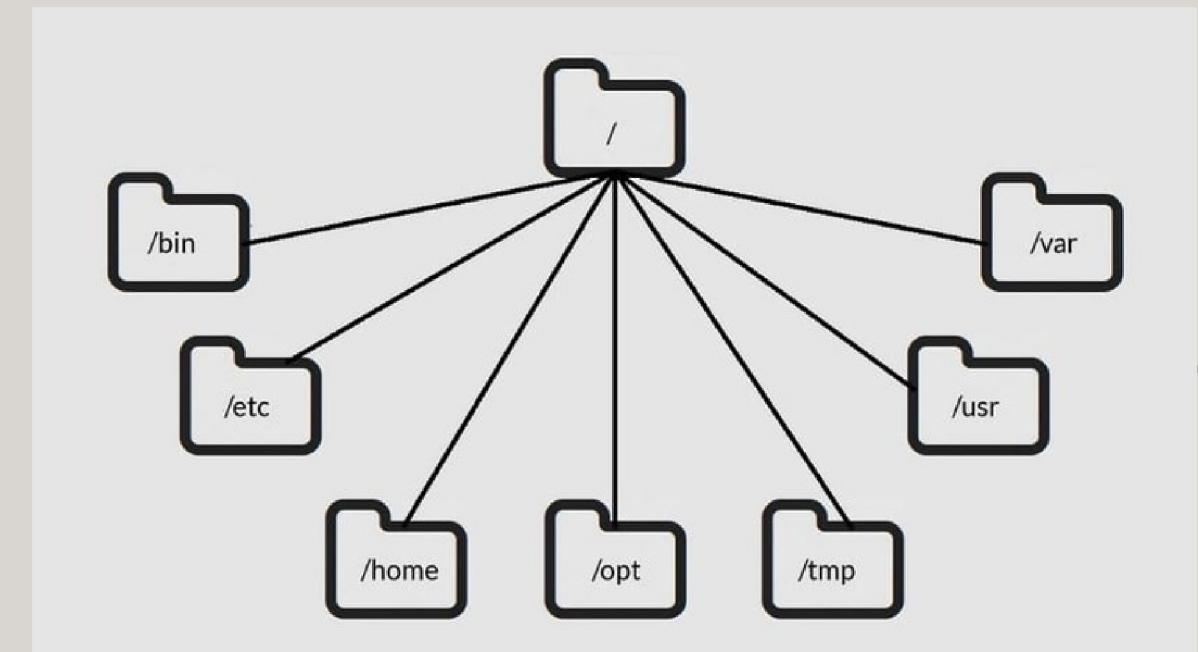
► **APFS (Apple File System)**

► **JFS (Journaled File System)**

► **XFS**

► **ReiserFS**

Cada sistema de ficheros tiene ventajas específicas y es adecuado para diferentes necesidades según el rendimiento, la integridad de datos y las características avanzadas requeridas.



# Servicios UNIX – ficheros

UNIX proporciona una secuencia lineal de bytes, gestionada mediante un puntero de posición que indica la ubicación actual en el fichero.

- `open` : abre un fichero y devuelve un `-1` si falla.
  - ruta (`path`)
  - apertura (`oflag`)
  - permisos (`mode`).
- Al crear se asignan el UID y el GID-
- `creat` = a `open` con las opciones:
  - `O\_WRO`
  - `NLY`
  - `O\_CRE`
  - `AT`
  - `O\_TRU`
  - `NC`
- fichero con permisos específicos.
- `read` y `write`: leer y escribir desde y hacia un fichero, posición actual del puntero de fichero ,puede leer menos datos si se alcanza el final y puede incrementar el puntero si se especifica
  - `O\_APPEND`.
- `close` libera un descriptor y decrementa (`nopens`).
- Libera al fichero y los recursos del s.o si `nopens` llega a 0.
- `lseek` cambia la posición del puntero
- Puede cambiar la posición a partir del inicio, la posición actual o el final del fichero.





# Servicios UNIX—directorios

Tiene la estructura lógica de una tabla, cada una de cuyas entradas contiene una estructura dirent que relaciona un nombre de fichero con su nodo-i

- opendir: Abre un directorio y devuelve un identificador.
- readdir: Lee la siguiente entrada de un directorio abierto y devuelve su información. Devuelve NULL cuando hay error o no hay más
- closedir: Cierra un directorio abierto. Devuelve 0 en caso de éxito y -1 en caso de error.
- rewinddir: Sitúa el puntero de posición en la primera entrada.
- rmdir: Borra un directorio vacío.
- mount: monta sistema de archivos en un directorio.
- mkdir: Crea un nuevo directorio con un nombre y permisos.
- link: Crea un nuevo enlace físico para un fichero existente.
- symlink: Crea un nuevo enlace
- umount: Desmonta un sistema de archivos.
- unlink: Elimina nombre y disminuye enlaces. Si los enlaces llegan a cero, se libera el espacio ocupado por el fichero.
- chdir: Cambia el directorio al especificado
- rename: Cambia el nombre
- getcwd: Nombre del directorio actual o de trabajo.

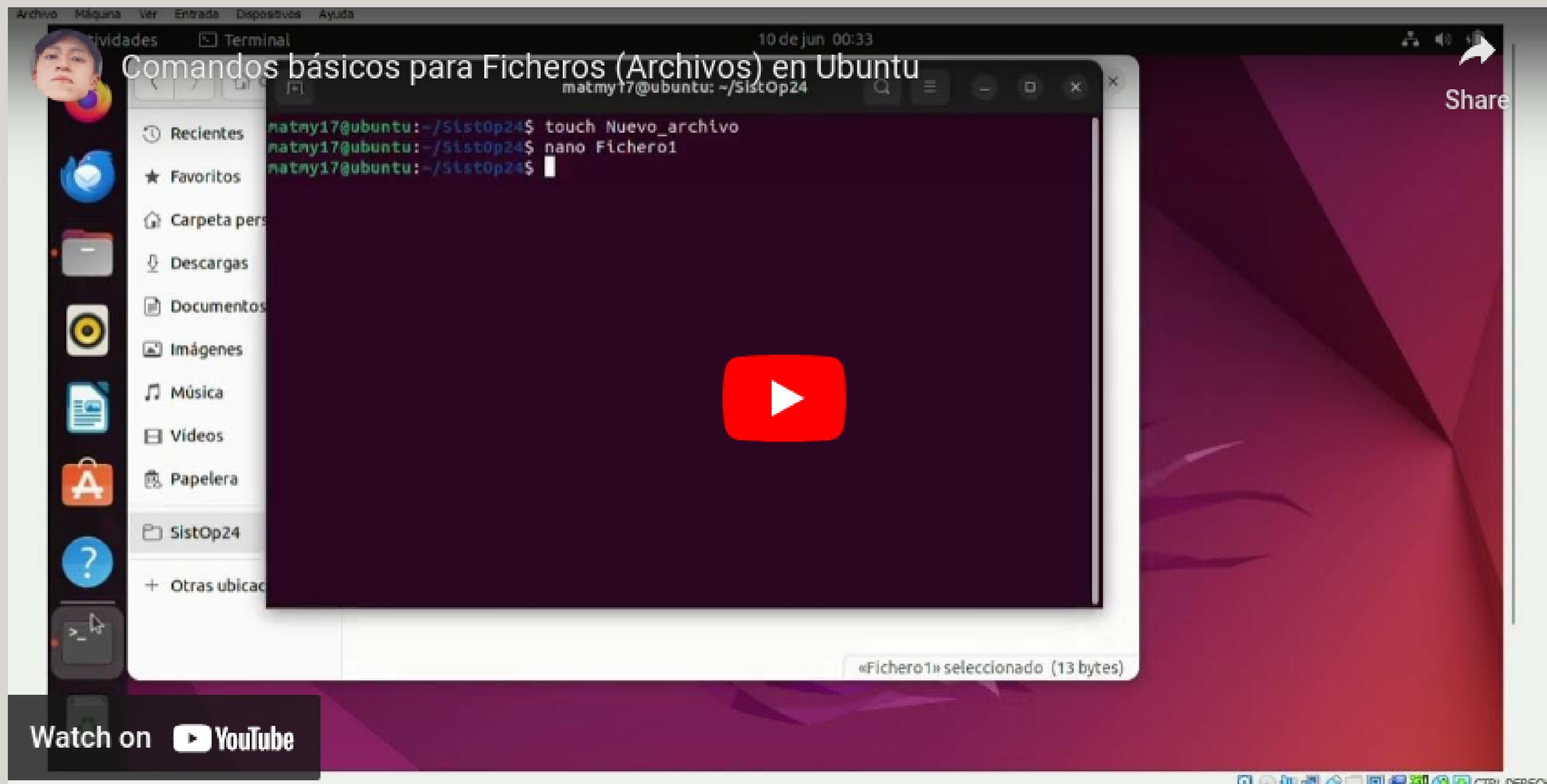
# Servicios de protección y seguridad

no existen servicios específicos para crear, destruir o abrir descriptores de protección. Estos se asocian a los objetos y se crean y se destruyen con dichos objetos.

- access: Comprueba si un fichero está accesible con ciertos privilegios de lectura, escritura o ejecución.
- umask: Define una máscara de protección que se aplica por defecto a los objetos creados por el proceso.
- chmod: Cambia los derechos de acceso a un archivo, modificando los bits de permiso y los bits SETUID y SETGID.
- chown: Cambia el propietario y el grupo de un archivo, permitiendo que un usuario "regale" un archivo a otro usuario.



# VIDEO:



**GRACIAS**