

## Practica Sistema de Ficheros

### INODO

Un INODO es una estructura de datos, por así decirlo una tabla que contiene información sobre un fichero. Cada fichero se identifica por un número de inodo. Este número es único dentro de todo el sistema de ficheros.

Dentro de cada inodo existe la siguiente información:

- \* Número de inodo
- \* Tipo de fichero
- \* Propietario de dicho fichero
- \* Permisos del fichero
- \* Fecha de creación del mismo

Puedes ver toda esta información si ejecutas un `ls -li` en cualquier directorio de tu sistema.

Ejemplo:

1. Creamos un archivo llamado `penlace` y le adicionamos `Hola Mundo`.

```
touch penlace
```

```
pico penlace
```

2. Identificamos su inodo

```
ls -li penlace
```

3. Creamos un enlace duro a `penlace` que se llame `pruebaln`.

```
ln penlace pruebaln
```

4. Con la orden `ls -li` vemos que ambos tienen el mismo inodo, y que este inodo tiene dos enlaces.

```
ls -li pruebaln penlace
```

5. Creamos un enlace simbólico a `penlace` que se llame `penlacesim`.

```
ln -s penlace penlacesim
```

6. Con la orden `ls -li` vemos que tienen distinto número de inodo. Además, el inodo de `penlacesim` sólo tiene un enlace, y el inodo de `penlace` sigue teniendo dos. En la línea correspondiente a `penlacesim` vemos que aparece el fichero al que apunta, y la letra `l` ("ele") al inicio.

*ls -li pruebaIn penlace penlacesim*

7. Cambiamos penlace y comprobamos si cambia penalcesim.

```
cat >>penlace  
otra vez adios  
CTRL+ D  
$cat penlacesim
```

8. Por último borramos penlace. Comprobamos que pruebaIn permanece y que no podemos ver el contenido de penlacesim, el sistema nos dirá que no existe. Para que desaparezca totalmente tenemos que borrarlo.

```
rm penlace  
cat pruebaIn  
cat penlacesim  
rm penlacesim  
rm pruebaIn
```

## Permiso de archivos

Octal	Decimal	Permission	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwx

Creamos las carpetas necesarias

```
mkdir /tmppermisos  
cd /tmp/permisos
```

Crear archivo con permisos específicos

```
echo "Secreto" > secreto.txt
```

Ninguno (000)

```
chmod 000 secreto.txt
```

```
cat secreto.txt # Error: Permiso denegado
```

Solo conocimiento (listar directorio, no leer)

```
chmod 100 secreto.txt # --x (ejecución en archivo = permiso de buscar)
ls -l secreto.txt    # Puedes ver que existe
cat secreto.txt      # Error
```

Lectura (400)

```
chmod 400 secreto.txt
cat secreto.txt    # OK
```

Adición (200 + 400 = 600)

```
chmod 600 secreto.txt
echo "Nueva línea" >> secreto.txt # OK
rm secreto.txt                # Error (necesita escritura en directorio)
```

Actualización (200)

```
chmod 200 secreto.txt
echo "Sobrescribir" > secreto.txt # OK
```

Borrado (permiso del directorio, no del archivo)

```
chmod 700 /tmp/permisos # Control total sobre el directorio
rm /tmp/permisos/secreto.txt # Ahora sí
```

## Grupos

Crear grupo y usuarios simulados

```
sudo groupadd biblioteca
sudo useradd -m -G biblioteca estudiante1
sudo useradd -m -G biblioteca profesor1
```

Crear archivo compartido

```
sudo mkdir /tmp/compartido
sudo chown :biblioteca /tmp/compartido
sudo chmod 775 /tmp/compartido
```

Archivo con permisos diferenciados

```
sudo -u profesor1 bash -c 'echo "Catálogo" > /tmp/compartido/catalogo.txt'
```

```
sudo chmod 664 /tmp/compartido/catalogo.txt # rw-rw-r-- (grupo puede escribir)
```

Ver permisos

```
ls -la /tmp/compartido/
```