

# File permissions in Linux

## Project description

- Revisión y cambios de permisos sobre archivos y directorios de un sistema operativo Linux
- Referencia documento permisos facilitado por la empresa

## Check file and directory details

Hacemos uso del comando **ls** que sirve para listar el contenido de un archivo.

Las flag u opciones **-l -a -h** las podemos combinar en una **-lah**. Observa la salida de **tlldr**.

### Comando ls

```
researcher2@4ae8c11741fa:~/projects$ tree -L 2
.
|-- drafts
|-- project_k.txt
|-- project_m.txt
|-- project_r.txt
`-- project_t.txt

1 directory, 4 files
researcher2@4ae8c11741fa:~/projects$ ls -lah
total 32K
drwxr-xr-x 3 researcher2 research_team 4.0K Feb 16 09:16 .
drwxr-xr-x 4 researcher2 research_team 4.0K Feb 16 10:03 ..
-rw--w---- 1 researcher2 research_team  46 Feb 16 09:16 .project_x.txt
drwx--x--- 2 researcher2 research_team 4.0K Feb 16 09:16 drafts
-rw-rw-rw- 1 researcher2 research_team  46 Feb 16 09:16 project_k.txt
-rw-r----- 1 researcher2 research_team  46 Feb 16 09:16 project_m.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb 16 09:16 project_r.txt
-rw-rw-r-- 1 researcher2 research_team  46 Feb 16 09:16 project_t.txt
researcher2@4ae8c11741fa:~/projects$
```

## Comando *ls*

```
List directory contents. More information: https://www.gnu.org/software/coreutils/manual/html\_node/ls-invocation.html.

- List files one per line:
  ls -l

- List [a]ll files, including hidden files:
  ls -a

- List files with a trailing symbol to indicate file type (directory/, symbolic_link@, executable*, ...):
  ls -F

- List [a]ll files in [l]ong format (permissions, ownership, size, and modification date):
  ls -la

- List files in [l]ong format with size displayed using [h]uman-readable units (KiB, MiB, GiB):
  ls -lh

- List files in [l]ong format, sorted by [S]ize (descending) [R]ecursively:
  ls -lsR

- List files in [l]ong format, sorted by [t]ime the file was modified and in [r]everse order (oldest first):
  ls -ltr

- Only list [d]irectories:
  ls -d */
```

El comando **tree** nos muestra un árbol del contenido del directorio. La flag -L sirve para indicar la profundidad de nivel que quieres.

## Describe the permissions string

En la captura vamos a fijarnos en la línea marcada que nos muestra uno de los archivos.

```
researcher2@4ae8c11741fa:~/projects$ ls -lah
total 32K
drwxr-xr-x 3 researcher2 research_team 4.0K Feb 16 09:16 .
drwxr-xr-x 4 researcher2 research_team 4.0K Feb 16 10:03 ..
-rw--w---- 1 researcher2 research_team 46 Feb 16 09:16 .project_x.txt
drwx--x--- 2 researcher2 research_team 4.0K Feb 16 09:16 drafts
-rw-rw-rw- 1 researcher2 research_team 46 Feb 16 09:16 project_k.txt
-rw-r----- 1 researcher2 research_team 46 Feb 16 09:16 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 16 09:16 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 16 09:16 project_t.txt
researcher2@4ae8c11741fa:~/projects$
```

Nos muestra el archivo “**project\_r.txt**”, con los siguientes permisos:

- **-rw-rw-r--**

Esto nos dice que :

- El primer bit nos muestra un - que dice que es un archivo, si observas la línea que muestra el subdirectorio **/drafts** verás que el primer bit es una **d**
- El grupo de tres bits siguientes nos muestra los permisos del usuario propietario del archivo **projects\_r.txt** que es **researcher2**. Nos muestra que tiene los permisos **rw** - que quiere decir que tiene permiso de **read** (lectura), **w**rite (escritura) y no tiene **x** (permisos para ejecutar), que es el último bit de esta serie de tres, que se muestra con -
- El segundo grupo de tres bits nos muestra los permisos para el grupo al que pertenece el archivo. El grupo es un conjunto de usuarios del sistema. Se usa en Linux para facilitar la gestión de acceso a archivos. Estos tres bits nos muestra que el grupo tiene exactamente los mismos permisos que el usuario, es decir lectura y escritura y no tiene permisos de ejecución. Por lo tanto todos los usuarios que pertenezcan al grupo **research\_team** tienen esos permisos.
- Y para terminar tenemos los permisos para **Others** (otros) que son cualquier usuario que no pertenezca a ninguno de los anteriores. En este caso tenemos los tres bits **r - -**

## Change file permissions

Según la organización para ningún archivo en este directorio, el grupo others (otros) debe tener permisos de escritura. Podemos observar que tenemos un archivo que no cumple esa norma. Procedemos a cambiar los permisos del archivo con permisos de administrador mediante el comando **sudo** y usando el comando **chmod**.

```
drwx--x--- 2 researcher2 research_team 4.0K Feb 16 09:16 drafts
-rw-rw-rw- 1 researcher2 research_team 46 Feb 16 09:16 project_k.txt
-rw-r----- 1 researcher2 research_team 46 Feb 16 09:16 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 16 09:16 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 16 09:16 project_t.txt
researcher2@4ae8c11741fa:~/projects$
```

Puedes ver que los tres bits del grupo others muestra **r w** - por lo tanto tiene permisos de escritura. Lo corregimos con `chmod` como superusuario con **sudo**.

### **sudo chmod o-w project\_k.txt**

Como puedes ver en la siguiente captura el permiso se eliminó.

```
researcher2@fac9aac8376e:~/projects$ ls -lah
total 32K
drwxr-xr-x 3 researcher2 research_team 4.0K Feb 17 09:39 .
drwxr-xr-x 3 researcher2 research_team 4.0K Feb 17 10:14 ..
-rw--w---- 1 researcher2 research_team 46 Feb 17 09:39 .project_x.txt
drwx--x--- 2 researcher2 research_team 4.0K Feb 17 09:39 drafts
-rw-rw-r-- 1 researcher2 research_team 46 Feb 17 09:39 project_k.txt
-rw-r----- 1 researcher2 research_team 46 Feb 17 09:39 project_m.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 17 09:39 project_r.txt
-rw-rw-r-- 1 researcher2 research_team 46 Feb 17 09:39 project_t.txt
researcher2@fac9aac8376e:~/projects$
```

Ahora tenemos que corregir los permisos para el grupo del archivo `project_m.txt` y quitarle los permisos de escritura y lectura.

### **sudo chmod g-rw project\_m.txt**

## Change file permissions on a hidden file

Tenemos un file configurado como oculto. Según la empresa nadie debería poder escribir en él pero el usuario y el grupo deben poder leerlo.

```
sudo chmod ug-w .project_x.txt
```

```
sudo chmod ug+r .project_x.txt
```

## Change directory permissions

Revisamos permisos con `ls -lah` y confirmamos que el subdirectorio tiene permisos mal configurados ya que sólo el user `researcher2` debe poder acceder al directorio `drafts` y su contenido.

```
sudo chmod g-x ./drafts
```

## Summary

La correcta configuración de permisos en Linux es una piedra fundamental de la seguridad del sistema ya que evita accesos y modificaciones en archivos y directorios delicados. El uso de comandos de consola como **ls**, **cat**, **chmod** y **sudo** son básicos para un administrador de linux.