# Sistema de permisos en Unix/Linux

Los permisos en Linux funcionan según el mismo esquema que en Unix. Básicamente se aplican sobre archivos y sobre directorios, van asociados a usuarios o grupos, y pueden ser de lectura, de escritura o de ejecución.

Dicho de otro modo, todos los archivos y directorios en Linux tienen asociado un conjunto de permisos que debe definir las posibilidades de lectura, escritura y ejecución que se aplican al usuario propietario del archivo, al grupo de usuarios al que pertenece, y al resto del mundo.

## Estructura de Usuarios y Grupos

### *Usuarios*

GNU/Linux es un sistema multiusuario en el que conviven simultáneamente diferentes cuentas, desempeñando roles muy diversos. Cada usuario se identifica por un User ID o UID. Los principales usuarios que podemos encontrar en un sistema Linux son:

- **Root o superusuario.** Por un lado tenemos a *Root* o superusuario (UID = 0), que es el único que tiene privilegios sobre todo el sistema, y el responsable de las tareas de administración del sistema, tales como la instalación y desinstalación de software, entre otras muchas. Para cualquier acción que necesite permisos de superusuario, el sistema requerirá las credenciales de root.
- **Usuarios de sistema.** Aparte, están los que podríamos denominar usuarios de sistema o especiales. Estos son los que van vinculados a ciertos servicios, y que pueden asumir ciertos privilegios relativos a este servicio. Se crean automáticamente en la instalación del sistema o de ciertas aplicaciones, como es el caso del antivirus ClamAV. Algunos ejemplos de estos usuarios son *bin*, *mail*, *apache*, *clamav*, *pulse*, *avahi*, *syslog*, *colord*, etc., pero realmente hay muchos más en cualquier instalación estándar.
- **Usuarios estándar.** Por último tenemos a las cuentas de usuarios individuales. De estas, puede haber tantas como se requiera. Cada usuario estándar posee su directorio personal dentro de la /home. Allí se almacenan todos sus archivos personales, además de las preferencias de varias aplicaciones, archivos temporales, etc.

Como pequeño paréntesis, solo decirte que en el post que te enlazo a continuación te cuento más en detalle sobre el usuario root y la gestión de privilegios que hacen la mayoría de distribuciones. Dicho esto, seguimos con el tema.

Todos los usuarios en Linux suelen estar registrados en el archivo /etc/passwd. Puedes visualizar el contenido del fichero con el comando que tienes a continuación (como ves, para hacerlo necesitas permisos de superusuario).

```
sudo cat /etc/passwd
```

## *Grupos*

Para simplificar la gestión de permisos entre tantas cuentas, Linux agrupa todos los usuarios dentro de grupos. De este modo, a la hora de especificar ciertos permisos sobre un conjunto de cuentas, se pueden establecer los permisos directamente sobre el grupo. Cada grupo se identifica por un Group ID o GUID.

Cada usuario pertenece por defecto a un grupo con el mismo nombre, que se conoce como grupo principal de usuario o grupo primario. Además, un mismo usuario puede estar dentro de otros grupos, que serían los secundarios.

Los grupos en Linux quedan registrados en el archivo /etc/group, y puedes visualizar su contenido con el siguiente comando:

```
sudo cat /etc/group
```

Todos y cada uno de los ficheros en Linux son propiedad de un usuario y de un grupo, y sobre cada una de ellos recaen unos privilegios relativos al propietario, al grupo, y al resto del mundo. Y con esto enlazamos directamente con las siguientes lineas…

# Tipos de Permisos

## *Privilegios de Usuario, Grupo y Otros*

De entrada, en cualquier fichero o carpeta hay tres tipos o niveles de privilegio, los que se aplican al propietario del archivo o carpeta, los que se aplican a todos los miembros del grupo al que pertenece el propietario, y los que se aplican a todos los demás. Para verlo con un poco más de detalle:

- **Permisos del Usuario.** Es el primer nivel de privilegios. Básicamente representa los permisos que se aplican al propietario de un determinado fichero o directorio.
- **Permisos del Grupo.** Estamos ante el segundo nivel de privilegios, que define los derechos de lectura, escritura y ejecución que se aplican solo a aquellos usuarios que pertenecen al mismo grupo que el propietario del fichero.
- **Permisos de Otros.** Este es el último nivel. Representan los privilegios de lectura, escritura y ejecución por parte del resto de usuarios que no entran en ninguno de los niveles anteriores.

## *Permisos de Lectura, Escritura y Ejecución*

Como hemos dicho, sobre un archivo podemos definir básicamente tres tipos de permisos (lectura, escritura y ejecución). Estos se establecen tanto sobre archivos como sobre carpetas, pero existen algunas diferencias de matiz entre unos y otros.

- **Lectura.** Representan la posibilidad de poder acceder a un archivo o carpeta y leer su contenido. En un directorio hacen referencia a la posibilidad visualizar su contenido.

- **Escritura.** Definen la posibilidad de acceder y modificar el contenido de un archivo, o en caso de un directorio, la capacidad de borrar o añadir archivos dentro de él.
- **Ejecución.** Estos son los más críticos. Indican la posibilidad de ejecutar un determinado archivo en el sistema. En el caso de un directorio, los derechos de ejecución representan la capacidad de entrar dentro del directorio.

# Como se Representan los Permisos

## *Representación Estándar*

Los permisos asociados a un archivo o directorio se suelen representar en línea. El primer carácter se reserva para los denominados permisos especiales (básicamente hace referencia al tipo de archivo), seguido de tres caracteres para los privilegios del propietario del archivo, otros tres para los del grupo, y los tres últimos para los del resto de usuarios.


El primer bit indica el tipo de archivo. Generalmente pueden darse los siguientes casos:

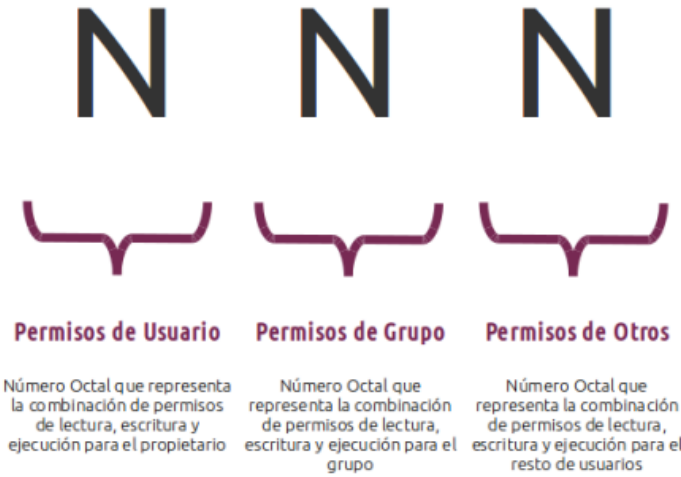
- `-` | Se trata de un archivo regular
- `d` | Se trata de un directorio
- `l` | Se trata de un enlace simbólico
- `b` | Se trata de un archivo especial de bloques

El resto de los 9 bits que vienen a continuación representan los permisos del propietario del fichero o directorio, los del grupo al que pertenece este usuario, y los del resto de usuarios. Aquí tienes el significado:

- `r` | Se dan permisos de lectura
- `w` | Se dan permisos de escritura
- `x` | Se dan permisos de ejecución
- `-` | No se da el permiso de lectura, escritura o ejecución

## *Representación Numérica o en Octal*

También hay la representación numérica, que parte básicamente de la representación anterior, pero sustituye cada grupo rwx por un número. Ese número se obtiene al sustituir cada tipo de permiso (r, w y x) por un 1 o 0, dependiendo de si se da o no el permiso).




Al final, del número en binario obtenido, el número decimal que le corresponde es el que se utiliza para representar las diferentes combinaciones rwx en cada grupo o nivel de permisos (usuario, grupo y otros).

```
r--  |  100  |  4
-w-  |  010  |  2
--x  |  001  |  1
```

Teniendo en cuenta esto, por cada nivel de privilegio (usuario, grupo u otros) podemos sacar un número que es la suma de todos los anteriores, dependiendo de los privilegios que se tengan. Con esto podemos llegar a tener las siguientes combinaciones:

```
rwx  |  111  |  7
rw-  |  110  |  6
r-x  |  101  |  5
r--  |  100  |  4
-wx  |  011  |  3
-w-  |  010  |  2
```

```
--x  |  001  |  1
---  |  000  |  0
```

Teniendo esto en cuenta, la representación es igual que en el caso anterior, pero para cada uno de los niveles (usuario, grupo y otros) se sustituye la combinación de los tres tipos de permisos por un número del 1 al 7, en función de las combinaciones anteriores.

De este modo, tenemos el 777 como la combinación que representa el máximo nivel de permisos que se puede dar a un fichero (¡ojo con el 777!)

# Cambiar permisos

```
chmod [opciones] permisos fichero
```

Donde permisos podrá ser una combinación de [u|g|o|a] +/- [r|w|x] o la codificación en octal de permisos para el usuario, el grupo y el resto de usuarios.

- u: usuario
- g: grupo
- o: resto de usuarios
- a: todos. Equivale a “ugo”
- r: lectura
- w: escritura
- x: ejecución

```
chmod ug+wx presentacion.txt
```

Otorgaría permisos de escritura y ejecución al fichero presentacion.txt.

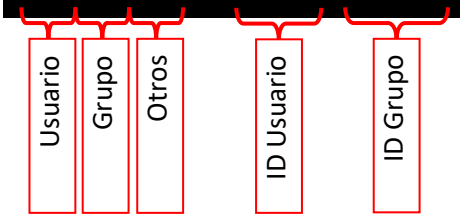
```
chmod 755 micarpeta/
```

Otorgaría permisos de lectura escritura y ejecución al usuario (el primer 7 equivale a rwx que en binario lo podemos codificar como 111), y lectura y ejecución al grupo y al resto de usuarios (el 5 equivale a r-x que en binario lo podemos codificar como 101).

# Visualizar permisos

```
ls -l
```

```
admin@DESKTOP-9F55G5J MINGW64 ~/Documents/Virtual Machines
$ ls -la
total 80
drwxr-xr-x 1 admin 197121 0 oct. 13 22:19  ./
drwxr-xr-x 1 admin 197121 0 oct.  9 14:43  ../
drwxr-xr-x 1 admin 197121 0 jun. 18  2019 'Debian 9.x 64-bit'/
-rw-r--r-- 1 admin 197121 0 oct. 13 22:19  fichero.txt
drwxr-xr-x 1 admin 197121 0 jun. 18  2019  ovirt/
drwxr-xr-x 1 admin 197121 0 jun. 18  2019  proxmox/
```




Adaptado de:

https://computernewage.com/2015/06/27/conoce-la-estructura-de-permisos-de-linux-al-detalle/