

Usuarios y permisos en GNU/Linux

Al igual que Unix, GNU/Linux es un sistema operativo multiusuario desde su concepción.

Todo archivo (y directorio) pertenece a un *usuario* y a un *grupo de usuarios*.

Por lo tanto, respecto de un determinado archivo, hay **3 tipos de usuarios: dueño (u); perteneciente al grupo (g), y otros (o)**.

A cada uno de estos tipos de usuarios, se le pueden asignar (o no), tres permisos: 1) permiso de **lectura (r)**; 2) permiso de **escritura (w)**; 3) permiso de **ejecución (x)**.

Permisos en los archivos

Para el caso de los archivos, el **permiso de lectura** significa la posibilidad de *acceder* al contenido del archivo (leerlo si es un texto, escucharlo si es un audio, etc). El **permiso de escritura** significa la posibilidad de *modificar* el contenido del archivo. El **permiso de ejecución (x)** significa la posibilidad de *ejecutar* el archivo como un programa.

Permisos en los directorios

Para el caso de las carpetas, el **permiso x** es el que da la posibilidad de *usar* la carpeta. Es decir: un usuario que no tenga permiso **x** sobre una carpeta, no podrá ejercer ninguno de los permisos que se le otorgaran dentro de esa carpeta.

En los directorios, el **permiso r** significa que el usuario tiene la posibilidad de ver una lista del contenido de la carpeta. Un usuario al que se le otorguen los permisos **w** y **x**, puede eliminar archivos y subdirectorios dentro de la carpeta, así como crear nuevos archivos y subcarpetas.

El comando ls -l

El comando `ls` con la opción `-l` (guion y “ele” minúscula), nos permite ver, entre otros datos, el usuario y grupo al que pertenece cada archivo, así como los permisos otorgados. Un ejemplo de la salida del comando es:

```
drwxr-xr-- raul alumnos 4096 2008-06-15 16:30 Cuentos
drwxrwxr-- raul alumnos 4096 2008-06-15 18:26 Musica
-rw-r--r-- raul alumnos 16852 2008-06-12 17:15 leerEsto.txt
-rw-rw-r-- raul alumnos 11329 2008-06-12 18:39 poema.odt
-rwxr-x--x raul alumnos 358 2008-06-16 11:05 programa.pl
```

La segunda columna indica el dueño del archivo (raul, en este caso); y la tercera columna indica el grupo del archivo (alumnos, en este caso).

La primera columna es un conjunto de 10 caracteres en donde se describen los permisos que posee cada usuario.

El primer carácter es un guion cuando se trata de un archivo, y una “d” cuando se trata de una carpeta.

Del segundo al cuarto carácter son los permisos del **dueño**. Por ejemplo, **rwX** significa que el dueño tiene todos los permisos. **r--** significa que el dueño tiene solamente permiso de lectura. Del quinto al séptimo carácter se representan los permisos del **grupo**. Los últimos tres caracteres representan los permisos de los **otros** usuarios.

Por ejemplo, esta línea de la salida:

```
-rwxr-x--x raul alumnos 358 2008-06-16 11:05 programa.pl
```

indica que: a) es un archivo (primer carácter); b) el dueño (raul) tiene todos los permisos (2º al 4º carácter, **rwX**); c) los usuarios que forman parte del grupo alumnos tienen permiso de lectura y ejecución (5º al 7º carácter, **r-X**); y d) los otros usuarios tienen solamente permiso de ejecución (últimos 3 caracteres, **--X**).

Notación binaria

Los permisos de un archivo o directorio, pueden definirse con notación binaria, con el siguiente método. Supongamos que los permisos de un determinado archivo son: **rwXr-xr--** Entonces:

- se separan los caracteres de tres en tres (**rwX r-x r--**), correspondiendo a dueño, grupo y otros.
- se reemplazan los guiones por ceros, y los otros caracteres por unos: **111 101 100**
- se “traduce” de binario a decimal (**111 → 7; 101 → 5; 100 → 4**). Se dice que el archivo *tiene permisos 754*.

El comando chmod

Sirve para cambiar los permisos de un archivo. Su sintaxis es: **chmod permisos nombreDeArchivo**

La parte de “**permisos**”, puede escribirse de dos maneras.

1) Especificando el tipo de usuario y el permiso. Por ejemplo, si se desea **darle(+)** permiso de **lectura(r)** a los “**otros(o)**” para el archivo `a.txt`: **chmod o+r a.txt**

Otro ejemplo: si se desea **quitarle(-)** permiso de **ejecución(x)** al **dueño(u)** sobre el mismo archivo: **chmod u-r a.txt**

Otro más: si se desea quitarle permiso de escritura y ejecución tanto a los usuarios del grupo como a los otros: **chmod go-wx a.txt**

2) Escribiéndolo con la notación binaria: **chmod 744 a.txt** (da todos los permisos al dueño y solamente permiso de lectura a los otros usuarios). O bien: **chmod 400 a.txt** (otorga permisos de lectura al dueño, y ningún permiso a los otros usuarios).

Los comandos chown y chgrp

El primero de ellos permite cambiar el dueño de un archivo o directorio. Su sintaxis es

chown nombreDeUsuario nombreDeArchivo

El otro comando permite cambiar el grupo al que pertenece un archivo. Su sintaxis es

chgrp nombreDelGrupo nombreDeArchivo

El usuario root, o superusuario

Es un usuario que **tiene, en todo momento, todos los permisos sobre todos los archivos**. Por este motivo debe ser utilizado con precaución, y solamente para realizar tareas administrativas. En algunas distribuciones hay que acceder al sistema con el usuario llamado **root**. O bien, se puede usar el comando **su**, para adquirir temporalmente la identidad de **root**. En otras distribuciones, se antepone **sudo** a ciertos comandos, para indicar que los mismos se ejecutarán como superusuario. Ej: **sudo mkdir carpeta**