Los permisos de acceso

1. Los permisos básicos

a. Permisos y usuarios

El papel de un sistema operativo es también el de asegurar la integridad y el acceso a los datos, lo que es posible gracias a un sistema de permisos. A cada archivo o directorio se le asignan unos privilegios que le son propios, así como autorizaciones de acceso individuales. Al intentar acceder, el sistema comprueba si está autorizado.

Cuando el administrador crea un usuario, le asigna un **UID** (*User Identification*) único. Los usuarios quedan definidos en el archivo **`/etc/passwd`** . Del mismo modo, cada usuario se integra en, al menos, un grupo (grupo primario). Todos éstos tienen un identificador único, el **GID** (*Group Identification*) y están definidos en el archivo **`/etc/group`** .

El comando **id** permite obtener esta información. A nivel interno, el sistema trabaja únicamente con los UID y GID, y no con los propios nombres.

```
$ id
uid=1000(seb) gid=100(users) grupos=7(lp),16(dialout),33(video),
100(users)
```

Se asocian un UID y un GID a cada archivo (inodo) que define su propietario y su grupo con privilegios. Usted asigna permisos al propietario, al grupo con privilegios y al resto de la gente. Se distinguen tres casos:

- UID del usuario idéntico al UID definido para el archivo. Este usuario es propietario del archivo.
- Los UID son diferentes: el sistema comprueba si el GID del usuario es idéntico al GID del archivo. Si es el caso, el usuario pertenece al grupo con privilegios del archivo.
- En los otros casos (ninguna correspondencia): se trata del resto de la gente (others), ni es el propietario, ni un miembro del grupo con privilegios.

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

d	rwxr-xr-x	29	seb	users	4096	Mar 15 22:13	Documentos

En esta línea de la tabla, el directorio Documentos pertenece al usuario seb y al grupo users, y posee los permisos rwxr-xr-x.

b. Significado

Permiso	Significado
General	
r	Readable (lectura).
w	Writable (escritura).
x	Executable (ejecutable como programa).
Archivo normal	
r	Se puede leer el contenido del archivo, cargarlo en memoria, listarlo y copiarlo.
w	Se puede modificar el contenido del archivo. Se puede escribir dentro. Modificar el contenido no significa poder eliminar el archivo (ver permisos en directorio).
x	Se puede ejecutar el archivo desde la línea de comandos si se trata de un programa binario (compilado) o de un script (shell, perl...).

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

Directorio	
r	Se pueden listar (leer) los elementos del directorio (catálogo). Sin esta autorización, ls y los criterios de filtro en el directorio y su contenido no serían posibles. No obstante, puede seguir accediendo a un archivo si conoce su ruta de acceso.
w	Se pueden modificar los elementos del directorio (catálogo), y es posible crear, volver a nombrar y suprimir archivos en este directorio.**Es este permiso el que controla el permiso de eliminación de un archivo**.
x	Se puede acceder al catálogo por CD y se puede listar. Sin esta autorización, es imposible acceder al directorio y actuar en su contenido, que pasa a estar cerrado.

Así, para un archivo:

rwx	**r-x**	**r--**
Permisos para el propietario de lectura, escritura y ejecución.	Permiso para los miembros del grupo de lectura y ejecución.	Permisos para el resto del mundo de lectura únicamente.

2. Modificación de los permisos

Cuando se crea, un archivo o un directorio dispone de permisos por defecto. Utilice el comando **chmod** (*change mode*) para modificar los permisos en un archivo o un directorio. Existen dos métodos para modificar estos permisos: mediante símbolos o mediante un sistema octal de representación de permisos. Sólo el propietario de un archivo puede

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

modificar sus permisos (además del administrador del sistema). El parámetro `-R` cambia los permisos de manera recursiva.

a. Mediante símbolos

La sintaxis es la siguiente:

```
chmod modificaciones Fic1 [Fic2...]
```

Si hay que modificar los permisos del propietario, utilice el carácter **u**; para los permisos del grupo con permisos, el carácter **g**; para el resto, el carácter **o**, y para todos, el carácter **a**.

Para añadir permisos, se utiliza el carácter **+**; para retirarlos, el carácter **-**, y para no tener en cuenta los parámetros anteriores, el carácter **=**.

Finalmente, ponga el permiso cuyos símbolos son: **r**, **w** o **x**.

Puede separar las modificaciones con comas y acumular varios permisos en un mismo comando.

```
$ ls -l
total 0
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic3
$ chmod g+w fic1
$ ls -l fic1
-rw-rw-r-- 1 seb users 0 mar 21 22:03 fic1
$ chmod u=rwx,g=x,o=rw fic2
$ ls -l fic2
-rwx--xrw- 1 seb users 0 mar 21 22:03 fic2
$ chmod o-r fic3
$ ls -l fic3
-rw-r----- 1 seb users 0 mar 21 22:03 fic3
```

Si quiere suprimir todos los permisos, no especifique nada después del signo =:

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

```
$chmod o=fic2
$ ls -l fic2
-rwx--x--- 1 seb users 0 mar 21 22:03 fic2
```

b. Sistema octal

La sintaxis es idéntica a la de los símbolos. A cada permiso le corresponde un valor octal, posicional y acumulable. Para codificar tres permisos rwx, hacen falta tres bits: cada uno tomaría el valor 0 o 1 según la presencia o ausencia del permiso. 2^3= 8, de ahí la notación octal.

- r vale 4.
- w vale 2.
- x vale 1.

La tabla siguiente servirá de ayuda:

Propietario			Grupo			Resto de la gente		
r	**w**	**x**	**r**	**w**	**x**	**r**	**w**	**x**
400	200	100	40	20	10	4	2	1

Para obtener el permiso final, basta sumar los valores. Por ejemplo, si quiere rwxrw-rw-, entonces obtiene 400+200+100+40+10+4+1=755, y para rw-r--r-- 400+200+40+4=644.

```
$ chmod 755 fic1
$ chmod 644 fic2
$ ls -l fic1 fic2
-rwxr-xr-x 1 seb users 0 mar 21 22:03 fic1
-rw-r--r-- 1 seb users 0 mar 21 22:03 fic2
```

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

La notación octal de los permisos no es sutil y no permite modificar un solo permiso. Es la totalidad de los permisos lo que se ha modificado de una sola vez.

3. Máscara de permisos

a. Restringir permisos de manera automática

En el momento de la creación de un archivo o de un directorio, se les asignan unos permisos automáticamente. Suele ser rw-r--r-- (644) para un archivo y rwxr-xr-x (755) para un directorio. Una máscara de permisos controla estos valores. Se puede modificar con el comando **umask**. El comando **umask** coge como parámetro un valor octal cuyo permiso individual se suprimirá de los permisos de acceso máximo del archivo o del directorio.

- Por defecto, se crean todos los archivos con los permisos 666 (rw-rw-rw-).
- Por defecto, se crean todos los directorios con los permisos 777 (rwxrwxrwx).
- Luego se aplica la máscara.
- La máscara es la misma para el conjunto de los archivos.
- Una máscara no modifica los permisos de los archivos existentes, sino solamente los de los archivos creados a partir de este momento.

Los permisos por defecto (máximo) de los archivos y de los directorios no son idénticos. Es lógico: como el permiso x permite entrar en un directorio, es normal que éste disponga de él por defecto. Este mismo permiso es inútil por defecto en los archivos: sólo una pequeña minoría de los archivos son scripts o binarios.

La máscara por defecto es 022, o sea `----w--w-`. Para obtener este valor, inserte **umask** sin parámetro.

```
$ umask
```

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

```
0022
```

b. Cálculo de máscara

Para un archivo

```
Predeterminado rw-rw-rw- (666)
Retirar      ----w--w- (022)
Resta        rw-r--r-- (644)
```

Para un directorio

```
Predeterminado rwxrwxrwx (777)
Retirar      ----w--w- (022)
Resta        rwxr-xr-x (755)
```

Observe que aplicar una máscara no es sustraer, sino suprimir permisos de los establecidos por defecto, permiso a permiso. Por ejemplo:

```
Predeterminado rw-rw-rw- (666)
Retirar      ----wxrwx (037)
Resta        rw-r----- (640)
```

Y no 629, lo que es imposible en sistema octal...

4. Cambiar de propietario y de grupo

Es posible cambiar el propietario y el grupo de un archivo gracias a los comandos **chown** (*change owner*) y **chgrp** (*change group*). El parámetro `-R` cambia la propiedad de manera recursiva.

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

```
chown usuario fic1 [Fic2...]
chgrp grupo fic1 [Fic2...]
```

Al especificar el nombre de usuario (o de grupo), el sistema comprueba primero su existencia. Usted puede especificar un UID o un GID. En este caso, el sistema no efectuará comprobación alguna.

Para los dos comandos, no se modifican los permisos anteriores ni la ubicación del archivo. Con un solo comando se puede modificar el propietario y el grupo a la vez.

```
chown usuario[:grupo] fic1 [fic2...]
chown usuario[.grupo] fic1 [fic2...]
```

Sólo root tiene permiso para cambiar el propietario de un archivo. Pero un usuario puede cambiar el grupo de un archivo si forma parte del nuevo grupo.

```
$ chgrp video fic1
$ ls -l fic1
-rwxr-xr-x 1 seb video 0 mar 21 22:03 fic1
```

5. Permisos de acceso extendidos

a. SUID y SGID

Es posible establecer **permisos de acceso especiales** para archivos ejecutables. Estos permisos de acceso extendidos aplicados a un comando permiten sustituir los permisos otrogados al usuario que lo inició por los permisos del propietario o del grupo a los que pertenece el comando.

El ejemplo más sencillo es el programa **passwd**, que permite cambiar la contraseña. Si se ejecutara el comando con los permisos de un usuario clásico, **passwd** no podría abrir y modificar los archivos `/etc/passwd` y `/etc/shadow` :

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

```
$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1440 feb 24 10:35 /etc/passwd
```

Puede observar que este archivo pertenece a root, y que sólo root puede escribir en él. Un usuario normal no puede leer su contenido sin interactuar. El comando **passwd** no debería, por lo tanto, poder modificar los archivos. Vea los permisos del comando **passwd** (/bin/passwd o /usr/bin/passwd):

```
> ls -l /usr/bin/passwd
-rwsr-xr-x 1 root shadow 78208 sep 21 23:06 /usr/bin/passwd
```

Lleva asociado un nuevo permiso: **s** para los permisos del usuario root. Este nuevo atributo permite la ejecución del comando con permisos de acceso extendidos. Durante el tratamiento, se ejecuta el programa con los permisos del propietario del archivo o del grupo al que pertenece. En el caso de passwd, se inicia con los permisos de root y no del usuario que lo lanzó.

El permiso s sobre el usuario se llama **SUID-Bit** (*Set User ID Bit*), y sobre el grupo, **GUID-Bit** (*Set Group ID Bit*).

El comando **chmod** permite ubicar SUID-Bit y GUID-Bit.

```
chmod u+s comando
chmod g+s comando
```

Los valores octales son 4000 para SUID-Bit y 2000 para GUID-Bit.

```
chmod 4755 comando
chmod 2755 comando
```

Sólo el propietario o el administrador puede activar esta propiedad. Posicionar SUID-bit o SGID-Bit tiene sentido únicamente si se han establecido los permisos de ejecución previamente (atributo x en el propietario o el grupo). Si éstos no están presentes; se sustituye la **s** por una **S**.

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

b. Real / efectivo

En los datos de identificación del proceso, ha podido observar la presencia de **UID** y **GID reales y efectivos**. Cuando se inicia un comando con un SUID-Bit o un SGID-Bit posicionado, los permisos se modifican. El sistema conserva los UID y GID de origen del usuario que inició el comando (UID y GID reales) transmitidos por el padre, los números UID y GID efectivos son los del propietario o del grupo de pertenencia del programa.

P. ej.: pepito (UID=100, GID=100) envía passwd, que pertenece a root (UID=1, GID=1) con SUID-Bit activado.

```
UID real: 100
GID real: 100
UID efectivo: 1
GID efectivo: 100
```

Si se posiciona sólo SGID-Bit:

```
UID real: 100
GID real: 100
UID efectivo: 100
GID efectivo: 1
```

Hay que subrayar que no se transmiten los SUID-Bit y SGID-bit a los hijos de un proceso. En este caso, se ejecutarán los hijos con los permisos del usuario que inició el comando básico, los UID reales.

c. Sticky bit

El **sticky bit** (*bit pegajoso*) permite asignar un criterio protector contra el borrado del contenido de un directorio. Imagine un directorio /tmp donde todos los usuarios tienen permiso para leer y escribir archivos.

```
$ ls -ld /tmp
drwxrwxrwx 6 root system 16384 Ago 14 13:22 tmp
```

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema

En este directorio todo el mundo puede suprimir archivos, incluidos los que no le pertenecen (permiso w presente en todas partes y para todos). Si el usuario pepito crea un archivo, el usuario titi puede suprimirlo incluso aunque no le pertenezca.

El sticky bit aplicado a un directorio, aquí /tmp, impide esta operación. Sí, pepito aún puede visualizar y modificar el archivo, pero sólo su propietario (o el administrador) podrá suprimirlo.

```
$ chmod u+t /tmp
ls -ld /tmp
drwxrwxrwt 35 root root 77824 mar 21 22:30 /tmp
```

En octal, se utilizará el valor 1000 (`chmod 1777 /tmp`).

Aunque aplicado al usuario, el sticky bit, representado por una **t**, aparece en el grupo de permisos de "others".

d. Permisos y directorios

Si da el permiso s al grupo en un directorio, todos los archivos creados dentro de este directorio serán del mismo grupo que este directorio, sea cual sea el grupo de la persona que crea este archivo.

```
$ mkdir dir
$ chmod 770 dir
$ ls -ld dir
drwxrwx--- 2 seb users 4096 mar 21 22:36 dir
$ chgrp video dir
$ chmod g+s dir
$ ls -ld dir
drwxrws--- 2 seb video 4096 mar 21 22:37 dir
$ cd dir
$ touch pepito
$ ls -l pepito
-rw-r--r-- 1 seb video 0 mar 21 22:37 pepito
```

© Éditions ENI - Todos los derechos reservados - Copia personal de Diego Armando Simbaña Quisilema