

MOD369 – Implantación de sistemas

UD8- Administración de la propiedad y de permisos

Propiedad y Permisos

Introducción

- El modelo de seguridad de Linux está basado en el de Unix, como sistema multiusuario. Dicho modelo de seguridad asume, por tanto, la presencia de múltiples usuarios en un solo ordenador y proporciona los medios para asociar los archivos individuales con los usuarios que los hayan creado, es decir, en los archivos “tienen propietarios”.
- Es imprescindible comprender a fondo este concepto para poder administrar bien la propiedad de los archivos tanto desde una herramienta gui como desde la línea de comandos del shell. El Concepto de propiedad se aplica igualmente a los programas en ejecución, es decir, a los “procesos”.

Propiedad y Permisos

Introducción

- La mayoría de los programas que ejecutamos quedan vinculados a la cuenta desde la que se abrieron.
- Esta identidad, Junto con la propiedad y los permisos asociados a los archivos, determina en último término si un programa puede, o no, modificar un archivo.
- El acceso al archivo se controla mediante una serie de permisos que se pueden establecer independientemente para el propietario del archivo, para el grupo del archivo y para el resto de usuarios del ordenador.

Propiedad y Permisos

Propiedad - Concepto de propiedad

- Las cuentas de usuario son la base para el sistema de propiedad de archivos. En Concreto: todo archivo tiene un propietario, es decir, una cuenta a la que está vinculado. Dicha asociación se establece mediante el número UID (User ID, ID de usuario) de la cuenta. Asimismo, cada archivo está asociado a un grupo mediante el número GID (Group ID, ID de grupo).
- Como root, se tiene la capacidad de modificar el propietario y el grupo de cualquier archivo.

```
[xabier@Centos Escritorio]$ cd tres  
[xabier@Centos tres]$ touch hola.txt  
[xabier@Centos tres]$ ls -l  
total 0  
-rw-rw-r--. 1 xabier xabier 0 mar 20 09:53 hola.txt  
[xabier@Centos tres]$
```

Propietario Grupo

Demostración del Contenido Conceptual

Propiedad y Permisos

Propiedad - Concepto de propiedad

- El propietario del archivo podrá también cambiar el grupo al que dicho archivo pertenezca, pero solo a alguno de los grupos de los que él mismo sea miembro.

Los mismos principios de propiedad de los archivos son de aplicación a los directorios, es decir, los directorios tienen propietarios y grupos.

- El root también tiene la capacidad de cambiarlos, así como, de forma más limitada, el usuario propietario del directorio.

Propiedad y Permisos

Propiedad - Establecer la propiedad

- Desde un intérprete de comandos el comando que permite cambiar la propiedad de un archivo es el **chown**, en su forma más básica, basta con pasarle como parámetro el nombre del archivo y un nombre de usuario:
`# chown bob file.ext`
- Es posible cambiar el propietario principal y su grupo con un solo comando separando propietario y grupo con dos puntos (:):
`# chown bob:users file.ext`
- Para cambiar de grupo sin cambiar de propietario basta con omitir este último, dejando los dos puntos (:) y el nombre del grupo:
`# chown :users file.ext`

```
xabier@Centos:/home/xabier/Escritorio/tres
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[xabier@Centos tres]$ ls -l
total 0
-rw-rw-r--. 1 xabier xabier 0 mar 20 09:53 hola.txt
[xabier@Centos tres]$ su
Contraseña:
[root@Centos tres]# cd /home/xabier/Escritorio/tres/
[root@Centos tres]# ls -l
total 0
-rw-rw-r--. 1 xabier xabier 0 mar 20 09:53 hola.txt
[root@Centos tres]# chown mikel hola.txt
[root@Centos tres]# ls -l
total 0
-rw-rw-r--. 1 mikel xabier 0 mar 20 09:53 hola.txt
[root@Centos tres]# chown xabier hola.txt
[root@Centos tres]# ls -l
total 0
-rw-rw-r--. 1 xabier xabier 0 mar 20 09:53 hola.txt
[root@Centos tres]# chown mikel:alumnos hola.txt
[root@Centos tres]# ls -l
total 0
-rw-rw-r--. 1 mikel alumnos 0 mar 20 09:53 hola.txt
[root@Centos tres]#
```

Demostración del Contenido Conceptual

Propiedad y Permisos

Propiedad - Establecer la propiedad

- Otra posibilidad es utilizar el comando **chgrp**, que funciona igual solo cambia el grupo y no necesita los dos puntos (:) antes de su nombre:
\$ chgrp users file.txt
Los comandos utilizados para cambiar la propiedad requieren privilegios de administrador mientras que se puede cambiar el grupo como usuario estándar siempre que el archivo sea nuestro y pertenezcamos a dicho grupo.
- Tanto **chown** como **chgrp** soportan una serie de opciones que modifican su comportamiento. La más útil de todas es -R (o - - recursive) porque permite cambiar la propiedad todos los archivos en un árbol completo de directorios.
- Chown —R bob /home/mary
Este comando otorga a bob la propiedad del directorio /home /mary, de todos los archivos que contenga, de todos sus subdirectorios y de todos los subdirectorios y archivos.

```
xabier@Centos:/home/xabier/Escritorio/tres
Archivo Editar Ver Buscar Terminal Ayuda
total 4
drwxr-xr-x. 2 xabier xabier 4096 mar 20 10:08 cuatro
-rw-r--r--. 1 xabier xabier   0 mar 20 10:07 hola2.txt
-rw-rw-r--. 1 xabier xabier   0 mar 20 09:53 hola.txt
[root@Centos tres]# chgrp alumnos hola.txt
[root@Centos tres]# ls -l
total 4
drwxr-xr-x. 2 xabier xabier 4096 mar 20 10:08 cuatro
-rw-r--r--. 1 xabier xabier   0 mar 20 10:07 hola2.txt
-rw-rw-r--. 1 xabier alumnos 0 mar 20 09:53 hola.txt
[root@Centos tres]# chgrp -r *
chgrp: opción inválida -- r
Pruebe `chgrp --help' para más información.
[root@Centos tres]# chgrp -R alumno *
chgrp: invalid group: «alumno»
[root@Centos tres]# chgrp -R alumnos *
[root@Centos tres]# ls -l
total 4
drwxr-xr-x. 2 xabier alumnos 4096 mar 20 10:08 cuatro
-rw-r--r--. 1 xabier alumnos   0 mar 20 10:07 hola2.txt
-rw-rw-r--. 1 xabier alumnos   0 mar 20 09:53 hola.txt
[root@Centos tres]#
```

Demostración del Contenido Conceptual

Propiedad y Permisos

Permisos - Entender los permisos de Linux

La salida en pantalla del comando `ls` presenta una lista de archivos contenidos en un directorio.

- Si lo usamos con la opción `-l`, crea una lista que incluye los permisos de los archivos.
- Esta línea contiene varias secciones, que proporcionan datos ordenados sobre el archivo:
 - Permisos (Permissions): La primera columna, en este caso `-rwxr-xr--x`, muestra los permisos del archivo, que nos interesan en este momento.
 - Número de enlaces o vínculos (links): La columna siguiente, muestra el número de enlaces duros (hard links) del archivo, es decir, el número de nombres de archivo únicos que se pueden utilizar para acceder al archivo.
 - Nombre de usuario (username): La columna siguiente, identifica al propietario del archivo mediante su nombre de usuario.
 - Nombre de grupo (Group name): El grupo de archivo.
 - Tamaño de archivo (Size): El tamaño del archivo.
 - Marca de tiempo (Time stamp): La marca de tiempo, identifica el momento en que el archivo se modificó por última vez.
 - Nombre de archivo (Filename):

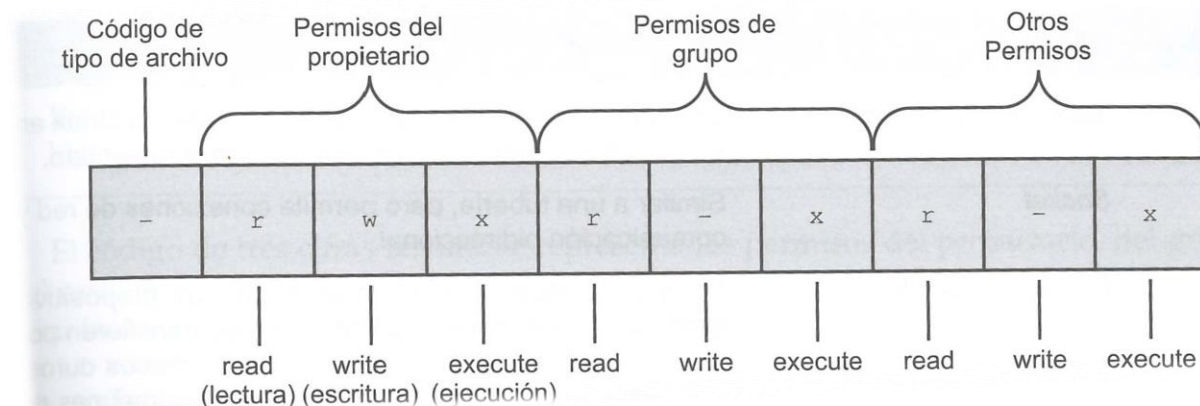
```
[root@Centos tres]# ls -l
total 4
drwxr-xr-x. 2 xabier xabier 4096 mar 20 10:08 cuatro
-rw-r--r--. 1 xabier xabier   0 mar 20 10:07 hola2.txt
-rw-rw-r--. 1 xabier alumnos  0 mar 20 09:53 hola.txt
```

Demostración del Contenido Conceptual

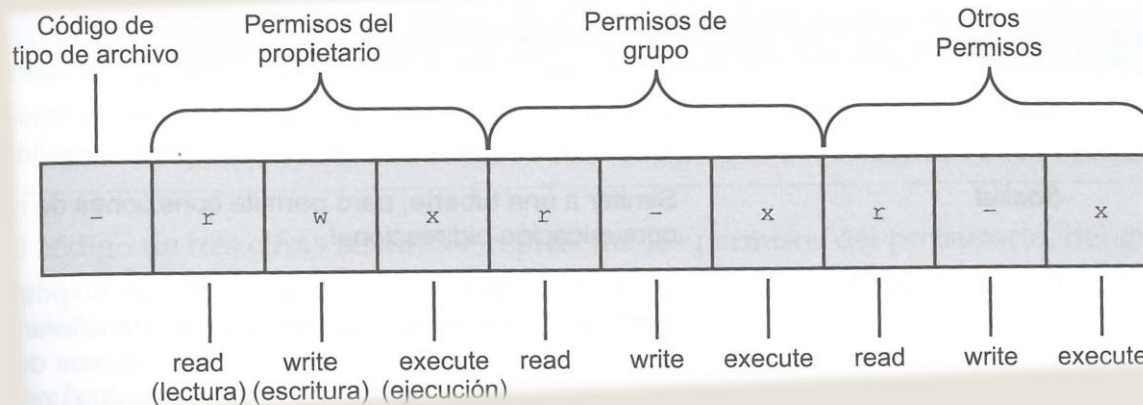
Propiedad y Permisos

Permisos - Entender los permisos de Linux

- Código de tipo de archivo (File Type Code): El primer carácter es el código de tipo de archivo, que representa el tipo de archivo.
- Permisos del propietario (Owner permissions): Estos permisos determinan lo que el propietario puede hacer con el archivo.
- Permisos de grupo (Group permissions): Estos permisos determinan lo que el resto de los miembros del grupo, salvo el propietario, pueden hacer con el archivo.
- Permisos globales, otros permisos o permisos para “el resto del mundo” (Other o World Permissions): Estos permisos determinan lo que el resto de usuarios, que ni son propietario ni pertenecen a su grupo, pueden hacer con el archivo.



```
[root@Centos tres]# ls -l
total 4
drwxr-xr-x. 2 xabier xabier 4096 mar 20 10:08 cuatro
-rw-r--r--. 1 xabier xabier   0 mar 20 10:07 hola2.txt
-rw-rw-r--. 1 xabier alumnos   0 mar 20 09:53 hola.txt
```



Demostración del Contenido Conceptual

Propiedad y Permisos

Cadena de permisos	Código octal	Significado
<code>rwxrwxrwx</code>	777	Todos los usuarios tienen garantizados los permisos de lectura, escritura y ejecución.
<code>rwxr-xr-x</code>	755	Todos los usuarios tienen garantizados los permisos de lectura y ejecución, pero solo el propietario del archivo tiene permiso de escritura.
<code>rwxr-x---</code>	750	Tanto el propietario como el grupo pueden leer y ejecutar el archivo. El propietario tiene también permiso de escritura. El resto de usuarios no tiene acceso al archivo.
<code>rwx-----</code>	700	Solo el propietario tiene permisos de lectura, escritura y ejecución; nadie más tiene acceso.
<code>rw-rw-rw-</code>	666	Todos los usuarios gozan de permisos de lectura y escritura en el archivo, pero no hay permiso de ejecución para nadie.
<code>rw-rw-r--</code>	664	El propietario y el grupo tienen permiso de lectura y escritura. El resto de usuarios solo tiene permiso de lectura.
<code>rw-rw----</code>	660	El propietario y el grupo tienen permiso de lectura y escritura. El resto de usuarios no tiene acceso.
<code>rw-r--r--</code>	644	El propietario goza de los permisos de lectura y escritura. El resto de usuarios solo tiene permiso de lectura.
<code>rw-r-----</code>	640	El propietario tiene los permisos de lectura y escritura. Los miembros del grupo tienen permiso de lectura. No hay más permisos para el resto.
<code>rw-----</code>	600	El propietario tiene los permisos de lectura y escritura. No hay más permisos para nadie.
<code>r-----</code>	400	El propietario tiene permiso de lectura únicamente. No hay más permisos para nadie.

Propiedad y Permisos

Permisos - Entender los permisos de Linux

- Hay varios casos particulares de aplicación al ámbito de los permisos:
 - Bit de ejecución en el caso de los directorios: Los directorios utilizan el bit de ejecución para otorgar permisos de búsqueda en su interior. Esta es una característica deseable en los directorios, así que prácticamente siempre veremos activa el bit de ejecución cuando esté activado el bit de lectura.


```
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorka1 gorka1    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorka1    0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-xr-x. 2 root   root    4096 mar 20 10:25 uno
[root@Centos compartido]#
```

```
[root@Centos compartido]# su mikel
[mikel@Centos compartido]$ cd /compartido/uno
[mikel@Centos uno]$
```



```
[mikel@Centos uno]$ cd /home/xabier/Escritorio
bash: cd: /home/xabier/Escritorio: Permiso denegado
[mikel@Centos uno]$
```

```
[mikel@Centos tres]$ cd /home
[mikel@Centos home]$ ls -l
total 52
drwx-----. 4 Alamcen1 Alamcen1 4096 mar 18 20:01 Alamcen1
drwx-----. 4 belen    belen    4096 mar 12 16:16 belen
drwx-----. 4 comercial1 comerciales 4096 feb 24 19:12 comerciales
drwx-----. 4 dani     dani     4096 mar 12 16:40 dani
drwx-----. 4 gorka    gorka    4096 mar 18 20:17 gorka
drwx-----. 4 gorka1   gorka1   4096 mar 18 20:17 gorka1
drwx-----. 4 ion      ion      4096 mar 10 18:56 ion
drwx-----. 4 ion1     ion1     4096 mar 10 20:09 ion1
drwx-----. 4 ion2     directores1 4096 mar 10 19:41 ion2
drwx-----. 4 jokin    jokin    4096 mar 19 12:12 jokin
drwx-----. 4 mikel    mikel    4096 mar 10 17:56 mikel
drwx-----. 4 pruebas  pruebas  4096 mar 10 11:44 pruebas
drwx-----. 27 xabier   xabier   4096 mar 20 10:13 xabier
```

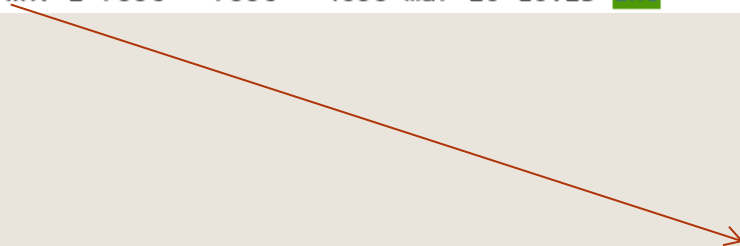

Demostración del Contenido Conceptual

Propiedad y Permisos

Permisos - Entender los permisos de Linux

- Hay varios casos particulares de aplicación al ámbito de los permisos:
 - Permisos de escritura en directorios: Los directorios son archivos que se interpretan de manera especial. Como tales, si un usuario puede escribir en un directorio, puede crear, borrar o cambiar el nombre a los archivos en aquel, incluso si dicho usuario no es el propietario de los archivos y carece de permiso de escritura sobre ellos.

```
-rw-rw-r--. 1 gorka1 gorka1    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorka1     0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-xr-x. 2 root   root    4096 mar 20 10:25 uno
[root@Centos compartido]# su mikel
[mikel@Centos compartido]$ touch uno/hola.txt
touch: no se puede efectuar `touch' sobre «uno/hola.txt»: Permiso denegado
[mikel@Centos compartido]$ exit
exit
[root@Centos compartido]# chmod 753 uno
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorka1 gorka1    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorka1     0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wx. 2 root   root    4096 mar 20 10:25 uno
[mikel@Centos tres]$ cd /compartido
[mikel@Centos compartido]$ ls -l
total 12
-rw-rw-r--. 1 gorka1 gorka1    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorka1     0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wx. 2 root   root    4096 mar 20 10:25 uno
[mikel@Centos compartido]$ touch uno/hola.txt
```




Demostración del Contenido Conceptual

Permiso de Escritura en directorio

Ha podido crear un fichero en el directorio

```
[mikel@Centos compartido]$ rm uno/hola2.txt  
rm: ¿borrar el fichero regular vacío «uno/hola2.txt» protegido contra escritura  
? (s/n) s  
[mikel@Centos compartido]$ █
```



```
[mikel@Centos uno]$ nano /compartido/uno/hola2.txt  
[mikel@Centos uno]$ █
```

Permiso de Escritura en directorio

**Ha podido eliminar un fichero sobre el que no
tenía ningún permiso en el directorio
No puede modificar su contenido**

Propiedad y Permisos

Permisos - Entender los permisos de Linux

- Uso de sticky bits
- Aunque el ejemplo anterior podría parecer un error, si no tenemos permisos de escritura sobre un archivo, no deberíamos ser capaces de borrarlo.
- Recordando que un directorio en linux no es más que un tipo de archivo especial, que guarda los nombres de otros archivos con punteros hacia las estructuras de datos de más bajo nivel, crear y borrar un archivo en un directorio necesita de permisos de escritura y lectura sobre el directorio.
- No sucede lo mismo con la escritura y lectura de un archivo pues será necesario tener permisos de escritura y lectura no sobre el directorio sino sobre el propio archivo.

Propiedad y Permisos

Permisos - Entender los permisos de Linux

- Uso de sticky bits
- La manera de crear resultados más coherentes es utilizar el Sticky bit, que es un marcador especial del sistema de archivos que modifica dicho comportamiento.
- Cuando se activa el sticky bit en un directorio, Linux solo nos permitirá borrar un archivo de su interior si somos o propietarios del archivo o del directorio contenedor.
- El sticky bit será reconocible ya que en su último dígito presentará una t.
- Este bit es especialmente importante para directorios compartidos por muchos usuarios.
- Es una característica estandar de /tmp y /var/tmp para que ningún usuario pudiera borrar archivos temporales de otros usuarios.

```
[root@Centos compartido]# chmod +t uno
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorkal gorkal    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorkal    0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wt. 2 root   root    4096 mar 20 14:18 uno
```

```
[root@Centos compartido]# chmod 1753 uno
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorkal gorkal    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorkal    0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wt. 2 root   root    4096 mar 20 14:51 uno
```

```
[root@Centos compartido]# chmod -t uno
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorkal gorkal    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorkal    0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wx. 2 root   root    4096 mar 20 14:18 uno
```

```
[root@Centos compartido]# chmod 0753 uno
[root@Centos compartido]# ls -l
total 12
-rw-rw-r--. 1 gorkal gorkal    5 mar 18 19:36 hola1.txt
-rw-rw-r--. 1 gorka  gorka     0 mar 18 19:37 hola3.txt
-rw-r--r--. 1 gorka  gorkal    0 mar 18 19:39 hola4.txt
-rw-rw-r--. 1 gorka  gorka     5 mar 18 19:30 hola.txt
drwxr-x-wx. 2 root   root    4096 mar 20 14:51 uno
```

Demostración del Contenido Conceptual

```
[root@Centos compartido]# chmod 1753 uno
[root@Centos compartido]# cd uno
[root@Centos uno]# touch hola2.txt
[root@Centos uno]# exit
exit
[xabier@Centos tres]$ su mikel
Contraseña:
[mikel@Centos tres]$ rm /compartido/uno/hola2.txt
rm: ¿borrar el fichero regular vacío «/compartido/uno/hola2.txt» protegido contra escritura? (s/n) s
rm: no se puede borrar «/compartido/uno/hola2.txt»: Operación no permitida
[mikel@Centos tres]$ █
```

Demostración del Contenido Conceptual

Propiedad y Permisos

Permisos - Establecer permisos desde el intérprete

- Shell en modo texto utilizaremos el comando **chmod** para modificar los permisos.
- Podemos configurar los permisos de como un número octal de tres cifras o de forma simbólica con un conjunto de la cadena de representación de los permisos.
- El nombre del comando **chmod** viene del inglés change mode (cambiar modo) donde mode es el nombre para referirse a los permisos.
- Modo octal:
- Chmod 456 file.txt
- El modo simbólico, por contra, consta de tres componentes:
- Un código que indica el conjunto de permisos que deseamos modificar: u para el usuario (o sea, el propietario), g para el grupo, o para otros usuarios y a para todos los permisos.
- Un símbolo para indicar si queremos añadir, +, eliminar, - o dejar igual, =, el modo respecto a su estado actual.
- Un código especificando cuál deberá ser el permiso. Este puede ser el habitual de r, w y x, para lectura, escritura y ejecución, respectivamente.

Propiedad y Permisos

Permisos - Establecer permisos desde el intérprete

- El modo simbólico, por contra, consta de tres componentes:
- Un código que indica el conjunto de permisos que deseamos modificar: u para el usuario (o sea, el propietario), g para el grupo, o para otros usuarios y a para todos los permisos.
- Un símbolo para indicar si queremos añadir, +, eliminar, - o dejar igual, =, el modo respecto a su estado actual.
- Un código especificando cuál deberá ser el permiso. Este puede ser el habitual de r, w y x, para lectura, escritura y ejecución, respectivamente.
- **Chmod** también incluye la opción de -r para hacerlo de forma recursiva.

Comando	Permisos iniciales	Permisos resultantes
chmod a+x bigprogram	rw-r--r--	rwxr-xr-x
chmod ug=rw report.tex	r-----	rw-rw----
chmod o-rwx bigprogram	rw-rwxr-x	rw-rwx---
chmod g-w,o-rw report.tex	rw-rw-rw-	rw-r-----

Propiedad y Permisos

Permisos - El bit de ejecución

El bit de permiso de ejecución nos permite identificar los programas ejecutables. Linux ejecuta dichos programas. Tales archivos se ejecutan con nuestras propias credenciales.

No obstante, hay ocasiones en las que los programas deben ejecutarse con privilegios más elevados.

Para realizar estas tareas, existen dos bits de permisos especiales:

SUID (Set user ID, establecer el ID de Usuario): Indica a Linux que debe ejecutar dicho programa con los privilegios del usuario dueño del archivo ejecutable en lugar de como la persona que ejecuta dicho archivo.

SGUID Define el grupo del programa en ejecución como el grupo del archivo.

Propiedad y Permisos

Permisos – Establecer el bit de ejecución

Para establecer el bit de ejecución, utilizaremos el comando `chmod`.

Código octal: Indicando un 4 en el primer bit de ejecución se activará el SUID. Estableciendo un 2 se activará el SGID estableciendo un 6 se activaran ambos valores.

Código simbólico: Se utiliza una `s` para junto con `u` para activar el SUID y `g` para activar el SGID.

```
Chmod u+s prog
```

```
Chmod g+s prog
```

```
Chmod ug-s prog
```

Propiedad y Permisos

Permisos – Ocultar archivos

Para ocultar archivos en Linux será suficiente comenzar el nombre del archivo o directorio con un ‘.’.

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**

Cuando creamos un archivo en nuestro sistema por defecto se le asignan permisos de lectura y escritura tanto para el propietario como para todos los demás, los permisos pueden ser modificados para que se asignen otros permisos distintos a los que vienen por defecto en la distribución, esto lo logramos mediante el comando umask.

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**

Hay dos formas de cambiar el valor de los permisos por defecto, una temporal y otra permanente.

que es ejecutando el comando desde la consola, y que durará mientras nos encontremos activos en esa consola y la otra es hacerla de manera permanente modificando el archivo `.bashrc` que se encuentra en nuestra carpeta home y que afectará solo a nuestro usuario y la otra es modificando el archivo `/etc/profile` que afectará a todos los usuarios del sistema, para poder modificar el segundo debemos tener permisos de root.

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**

Hay dos formas de cambiar el valor de los permisos por defecto, una temporal y otra permanente.

Temporal: ejecutando el comando `umask` desde la consola, y que durará mientras nos encontremos activos en esa consola.

Permanente: Para hacerlo de manera permanente, modificando el archivo `.bashrc` o el archivo `/etc/profile`.

`.bashrc`: se encuentra en la carpeta `home` y que afectará solo al usuario.

`/etc/profile` que afectará a todos los usuarios del sistema, para poder modificarlo debemos tener permisos de `root`.


```
[root@Centos home]# less /etc/profile | grep umask
# By default, we want umask to get set. This sets it for login shell
I  umask 002
   umask 022
```

```
umask 022
[root@Centos home]# less /home/ion/.bashrc | grep umask
[root@Centos home]# █
```

```
[root@Centos home]# umask
0022
[root@Centos home]# exit
exit
[xabier@Centos ~]$ umask
0002
[xabier@Centos ~]$ █
```

Demostración del Contenido Conceptual

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**
- Existen permisos base para la creación de archivos y directorios:
- Los permisos base de los archivos es 666 y los de los directorios es 777.
- Partiendo de ésta base de permisos la operación para determinar los permisos de creación de archivos y de directorios es que restemos el valor que le hemos dado a umask a los permisos base.
- Si creamos un archivo se lo restamos a los permisos base de los archivos que son 666
- En caso de ser un directorio lo que se hace es restárselo a los permisos base de los directorios que son 777.

DIRECTORIOS

Permisos base para directorios: 777

Permisos definidos en umask: 022

Operación: Restarle el umask a los permisos base de la siguiente manera:

$$777 - 022 = 755$$

ARCHIVOS

Permisos base para archivos: 666

Permisos definidos en umask: 022

Operación: Restarle el umask a los permisos base de la siguiente manera:

$$666 - 022 = 644$$

Demostración del Contenido Conceptual

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**
- Modificación temporal:
- En la consola el comando umask seguido del valor que queremos que obtenga, será sólo temporal es decir en el momento en que cerremos la consola se perderá ese valor y se volverá al que se encuentra almacenado en /etc/profile

```
[xabier@Centos ~]$ umask 044
[xabier@Centos ~]$ touch hola2.txt
[xabier@Centos ~]$ ls -l
total 36
drwxrwxr-x. 4 xabier xabier 4096 mar 21 17:04 cuatro
drwxr-xr-x. 2 xabier xabier 4096 feb  7 12:24 Descargas
drwxr-xr-x. 2 xabier xabier 4096 feb  7 12:24 Documentos
drwxr-xr-x. 8 xabier xabier 4096 mar 20 09:52 Escritorio
-rw--w--w-. 1 xabier xabier    0 mar 24 11:19 hola2.txt
```

666-044=722

```
[xabier@Centos ~]$ mkdir dhola
[xabier@Centos ~]$ ls -l
total 40
drwxrwxr-x. 4 xabier xabier 4096 mar 21 17:04 cuatro
drwxr-xr-x. 2 xabier xabier 4096 feb  7 12:24 Descargas
drwx-wx-wx. 2 xabier xabier 4096 mar 24 11:21 dhola
```

777-044=733

Demostración del Contenido Conceptual

Propiedad y Permisos

- **Permisos – Permisos De Archivo Por Defecto En Linux (umask)**
- Modificación permanente – Para el usuario
- Para hacerlo permanente editamos el archivo `.bashrc` que se encuentra en nuestro home, si están en la consola pueden ejecutar el siguiente comando “`gedit .bashrc`” y agregar al final del archivo la línea “`umask 022`” el 022 pueden modificarlo como ustedes gusten, es el valor que queremos sea restado para obtener los permisos deseados, guardamos el archivo cerramos la consola y la volvemos a abrir, de ahora en adelante `umask` tendrá el valor que especificamos en `.bashrc`, si queremos todo como antes pueden eliminar la línea que agregaron al archivo o darle el valor 022.

```
[root@Centos ion1]# cd /home/ion1
[root@Centos ion1]# ls -la
total 32
drwx-----. 4 ion1 ion1      4096 mar 10 20:09 .
drwxr-xr-x. 15 root root      4096 mar 19 10:38 ..
-rw-----. 1 ion1 ion1       204 mar 10 20:17 .bash_history
-rw-r--r--. 1 ion1 ion1        18 jul 18  2013 .bash_logout
-rw-r--r--. 1 ion1 ion1       176 jul 18  2013 .bash_profile
-rw-r--r--. 1 ion1 ion1       124 jul 18  2013 .bashrc
drwxr-xr-x. 2 ion1 ion1      4096 nov 12  2010 .gnome2
-rw-r--r--. 1 ion1 directores1    0 mar 10 19:52 hola2.txt
-rw-r--r--. 1 ion1 directores1    0 mar 10 19:58 hola3.txt
-rw-r--r--. 1 ion1 jefes1         0 mar 10 19:59 hola4.txt
-rw-rw-r--. 1 ion1 ion1          0 mar 10 19:49 hola.txt
drwxr-xr-x. 4 ion1 ion1      4096 feb  7 12:02 .mozilla
[root@Centos ion1]#
```

```
GNU nano 2.0.9          Fichero: .bashrc

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions
umask 044
```

Demostración del Contenido Conceptual

```
[root@Centos ion1]# cd /home/ion1
[root@Centos ion1]# ls -la
total 32
drwx-----. 4 ion1 ion1      4096 mar 10 20:09 .
drwxr-xr-x. 15 root root      4096 mar 19 10:38 ..
-rw-----. 1 ion1 ion1       204 mar 10 20:17 .bash_history
-rw-r--r--. 1 ion1 ion1        18 jul 18 2013 .bash_logout
-rw-r--r--. 1 ion1 ion1       176 jul 18 2013 .bash_profile
-rw-r--r--. 1 ion1 ion1       124 jul 18 2013 .bashrc
drwxr-xr-x. 2 ion1 ion1      4096 nov 12 2010 .gnome2
-rw-r--r--. 1 ion1 directores1    0 mar 10 19:52 hola2.txt
-rw-r--r--. 1 ion1 directores1    0 mar 10 19:58 hola3.txt
-rw-r--r--. 1 ion1 jefes1         0 mar 10 19:59 hola4.txt
-rw-rw-r--. 1 ion1 ion1          0 mar 10 19:49 hola.txt
drwxr-xr-x. 4 ion1 ion1      4096 feb  7 12:02 .mozilla
[root@Centos ion1]#
```

```
GNU nano 2.0.9          Fichero: .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions
umask 044
```

```
[root@Centos ion1]# su ion1
[ion1@Centos ~]$ umask
0044
[ion1@Centos ~]$
```

```
[ion1@Centos ~]$ touch hola3.txt
[ion1@Centos ~]$ ls -l
total 0
-rw-r--r--. 1 ion1 directores1 0 mar 10 19:52 hola2.txt
-rw-r--r--. 1 ion1 directores1 0 mar 24 11:41 hola3.txt
-rw-r--r--. 1 ion1 jefes1      0 mar 10 19:59 hola4.txt
-rw-rw-r--. 1 ion1 ion1       0 mar 10 19:49 hola.txt
```

Demostración del Contenido Conceptual