

Propietarios y permisos

Por Paco Aldarias Raya

25 de noviembre de 2003

Email: pacosecun@inicia.es

Web: <http://pagina.de/pacodebian>

Este documento es de libre reproducción siempre que se cite su fuente.

Realizado con: \LaTeX

Índice

1. Introducción	1
2. Propietarios y permisos de los ficheros	1
3. Permisos	2
4. Modificar permisos: <code>chmod</code>	3
5. Cambiar grupo: <code>chgrp</code>	4
6. Cambiar propietario y el grupo de un fichero: <code>chown</code>	5
7. Máscara de permisos: <code>chmod</code>	6
8. Permisos por defecto: <code>umask</code>	7
9. Enlaces a ficheros: <code>ln</code>	8
10. Bibliografía	9

1. Introducción

Un fichero es el lugar donde se guarda la información. Un sistema de ficheros es un conjunto de ficheros y directorios. Linux es el sistema con mayor seguridad debido a su sistema de ficheros.

El linux (Unix) considera un fichero como cualquier cosa instalada, aquí se incluye teclado, terminal, cdrom, disquetera, carpetas remotas,etc.

2. Propietarios y permisos de los ficheros

Esta característica refleja el carácter multiusuario.

Permite proteger la información de usuario del resto.

Un usuario sólo puede acceder a los ficheros q el es propietario y según los permisos q posea.

Desde consola hacemos:

```
$ ls -l
```

Nos aparecen cosas asi:

```
drwxr-x-x 5 jose alumnos 5000 Oct 30 13:47 cartas
```

Veamos q es numerádolo:

```
drwxr-x-x(1) 5(2) jose(3) alumnos(4) 5000(5) Oct 30 13:47(6) cartas(7)
```

Indican:

1. Indica si es fichero o directorio y sus permisos.
Si al principio hay :
-Una d es un directorio.
-Una - es un fichero.
2. Si es un fichero es el número de enlaces.
Si es un directorio el número de subdirectorios.
3. Propietario
4. Grupo
5. Tamaño en bytes.
6. Fecha de creación o última actualización.
7. Nombre del fichero o directorio.

3. Permisos

Permisos para ficheros normales:

- Escritura (w). Es posible modificarlo o borrarlo.
- Lectura (r). Es posible leer el contenido.
- Ejecución (x). Se puede ejecutar.

Permisos para directorios:

- Escritura (w). Es posible crear ficheros y directorios en su interior. También es posible borrar los ficheros contenidos en él. El propietario de un directorio, siempre podrá borrar los archivos contenidos en él, aunque sean de otro usuario.
- Lectura (r). Permite listar su contenido pero no permite entrar.
- Ejecución (x). Permite el acceso a directorios.

En general, se da permiso de ejecución y lectura, cuando se quiera autorizar el acceso, y ninguno cuando se quiera denegar.

Si no se tiene el permiso necesario para realizar alguna acción, sale el mensaje de error:

”Permission denied”

Que permisos representa esto?

- r w x r - x r - x

Son 10 caracteres:

- r w x r - x r - x

0 1 2 3 4 5 6 7 8 9

Veamos q significan:

0: Fichero/Directorio

123: Propietario tiene rwx

456: Grupo tiene r - x

789: Otros tiene r - x

Cuando se deniega un permiso aparece un gui3n.

Ejercicio. Indicar su significado:

1. `d r w x r - x - - x` 1 scout staff 5000 Oct 30 13:47 scout
2. `- r w x r - - - - -` 2 brigitte webs 3000 Oct 30 13:50 util.c
3. `- r w x - - x - - -` 1 root admin 2000 Oct 30 13:51 conecta

4. Modificar permisos: chmod

La forma general es:

chmod ugoa+-=rwxugo fichero

Siendo:

- u : propietario
- g : grupo
- o : otros (ni propietario, ni el grupo)
- a : todos
- + : a1ade permisos
- - : quita permisos
- = : iguala el nivel de permisos a los del grupo
- r : permiso de lectura
- w : permiso de escritura
- x : permiso de ejecuci3n

El atributo +x equivale a a+x

Ejemplos:

1. `chmod og-r fichero.c`
Quitamos el permiso de lectura a otros y grupo.
2. `chmod +x prog`
Damos permiso de ejecuci3n a todos los usuarios.

Ejercicios:

1. Crear una carpeta llamada prueba. Y hacer q puedan guardar cosas todos los usuarios del grupo, Y que no puedan guardar el resto de los usuarios.
2. Cambiar los permisos de la carpeta prueba, q puedan guardar todos los usuarios.
3. Crear una carpeta q el grupos de usuarios pueda entrar dentro de la carpeta, y el resto de usuarios solo puedan listar su contenido.

5. Cambiar grupo: chgrp

Un usuario puede pertenecer a más de un grupo de usuarios. De este modo los usuarios de un grupo pueden tener acceso a ciertos ficheros.

Su forma general es:

chgrp grupo fichero

Esto permite cambiar el grupo de usuarios a un fichero.

Ejemplo:

```
$ chgrp prog tabla.c (cambiamos el grupo a prog, del fichero tabla.c)
```

```
$ chmod g+r tabla.c (cambiamos el los permisos del grupo para q sea r)
```

```
$ ls -l tabla.c (vemos como ha quedado)
```

Sólo podremos cambiar el grupo de un fichero si somos el propietario de un fichero y miembros del grupo al que queremos asignarlo.

6. Cambiar propietario y el grupo de un fichero: chown

Forma general:

chown usuario[.grupo] fichero

Podemos ver los grupos existentes siguiendo estos pasos:

```
su -  
cat /etc/group  
exit
```

Ejemplo. Hacemos q fichero.txt tenga como propietario alumno y como grupo 1bach:

```
chown alumno.1bach fichero.txt
```

Sería equivalente poner:

```
chgrp 1bach fichero.txt
```

a:

```
chown .1bach fichero.txt
```

IMPORTANTE:

Sólo el usuario **root** puede cambiar el propietario de un fichero

Ejemplo:

1. Creamos el fichero fichero1:
`touch fichero1.txt`
2. Comprobamos quien es el usuario y el grupo:
`ls -la fichero1.txt`
3. Pasamos a ser el usuario root:
`su -`
4. Cambiamos el fichero1, q tenga como propietario grupo2, y grupo grupo3.
`chown grupo2.grupo3 fichero1`
5. borrar fichero1:
`rm fichero1`
6. Para salir de root:
`exit` o (también CTRL+D)

Ejercicios:

1. Crear un fichero vacio llamado fichero2
2. Cambiar el usuario y grupo de fichero2 para q sea de otro grupo.
3. Borrar el fichero2

7. Máscara de permisos: chmod

Su forma general es:

```
chmod mascara-octal fichero
```

Siendo máscara tres números, correspondientes al propietario, grupo y otros.

Siendo:

```
usuario grupo otros
r w x r w x r w x
4 2 1 4 2 1 4 2 1
```

Ejemplo

```
chmod 763 fichero1
```

usuario = 7 = 4 + 2 + 1 = rwx

grupo = 6 = 4 + 2 = rw-

otros = 3 = 2 + 1 = -wx

Quedando los permisos como: r w x r w - w x

Ejercicios. Indicar q máscara tendrá:

1. 4 5 1
2. 2 3 0
3. 4 8 5
4. r - x - - - r w x
5. - w x - - - - w x
6. r w x - - - r - x

Se debe considerar, que chmod puede usarse sin máscara o con máscara. Con máscara es más rápido.

Ejemplo.

Para el fichero3,

con estos permisos: - r w x r - x r - x

queremos q tenga estos: - r w x - w - - - -

Quitar permisos de r y x a otros y al grupo.

Dar permisos de escritura al grupo.

Seria sin máscara:

```
chmod og-rx fichero3
chmod g+w fichero3
```

Seria con máscara:

```
chmod 720 fichero3
```

Ejercicios:

1. Crear un fichero con: touch fichero2.txt
Cambiarle los permisos para q tenga: r - - - - -
Borrar el fichero.
2. Crear el directorio: mkdir dir1
Cambiarle los permisos para q tengan: r w x r - - - - -
Borrar el directorio.

8. Permisos por defecto: umask

Su forma general es:

umask mascara

Para determinar q permisos se otorgan por defecto, se restara el número 6 de cada uno de los dígitos de la máscara para saber que permisos se dan a los ficheros normales y el 7 para los directorios.

Ejemplo.

Por defecto, se usa: umask 022 corresponde 644 a los ficheros, y 755 para los directorios.

La máscara indica qué permisos no se dan. Así el 0, indica q se otorgan todos los permisos al propietario del fichero (es decir, lectura y escritura, y de acceso y escritura para directorios).

Si se desea suprimir el acceso a nuestros ficheros al resto de usuarios, haríamos umask 077

Por defecto los ficheros se crean con la mascara 644 y los directorios con 755.

La cifra 7 es esas posiciones indica q ni los miembros del grupo ni el resto de usuarios tendr n ningun permiso sobre los ficheros q creemmos desde q ejecutemos el comando umask.

La máscara se respeta si utilizamos el comando chmod, sin asignar los permisos a ningún usuario. Es decir, el comando chmod +x , sólo otorgaria permisos de ejecución al propietario del fichero en el caso de q la mascara fuera 077.

Si se desea utilizar una máscara, se debera situar el comando umask en los ficheros q definen el entorno del usuario al iniciar la sesión (como .bashrc, .profile,..)

9. Enlaces a ficheros: ln

Los enlaces se utilizan para poder situar un mismo fichero en diferentes directorios o poder otorgarle diferentes nombres. sin q exista, en realidad, más

q un único fichero.

Los enlaces son entradas en un directorio q apuntan a otro fichero.

Un uso habitual de los enlaces es el mantener varias versiones de un mismo programa. Por ejemplo, supongamos q tenemos dos versiones del programa tostador, llamadas tostador-1.0 y tostador-1.3. Queremos utilizar tostador-1.0. Para q al escribir tostador, se ejecute la versión 1.0, haremos un enlace de forma q tostador apunte a tostador-1.0, y sea ese el programa q se ejecute.

Para crear enlaces se utiliza en comando ln, así:

ln -s fichero-origen fichero-enlace

Ejemplo: ln -s tostador-1.0 tostador

Veamos como queda el enlace:

```
$ls -l tostador*
```

```
lrwxrwxrwx 1 root 12 31 Oct 19:26 tostador ->tostador-1.0
-rwxrwxrwx 1 root 1000 15 Oct 19:26 tostador-1.0
-rwxrwxrwx 1 root 1000 30 Oct 19:26 tostador-1.3
```

Podemos ver q tostador apunta a tostador-1.0. También podemos ver q en el campo de permisos aparece una l (link)

Los permisos de enlace no significan nada, ya q siempre son los permisos del fichero q apunta.

Un enlace se borra de la misma forma q un fichero normal, es decir, con rm. Estos enlaces se llama simbólicos (symbolic links), de ahí q se crean q crearse con el parámetro -s. Los enlaces también pueden apuntar a directorios.

10. Bibliografía

1. Cesar Martín Pérex. Linux. Guia Práctica. Editorial Anaya. 1998.