

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Grado en Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Universidad de Cádiz

Práctica 3 Permisos (Sesión 1)

Sistemas
Operativos

Contenido

Introducción

Permisos de
ficheros y
directorios

Cambio de
permisos

La orden
umask

- 1 Introducción
- 2 Permisos de ficheros y directorios
- 3 Cambio de permisos
- 4 La orden umask

- **Permisos** Todo fichero lleva asociado un conjunto de permisos que definen quién puede acceder a él y qué operaciones puede realizar.
- **Superusuario (root)** Usuario especial que puede hacer cualquier operación sobre un fichero, independientemente de sus permisos. También puede cambiar los permisos de cualquier fichero.
- **UID** Número que identifica a cada usuario dentro del sistema.
- **GID** Número que identifica a cada grupo dentro del sistema
- **Grupo principal** Todo usuario pertenece a un grupo principal (su GID aparece en el fichero `/etc/passwd`).
- **Grupos secundarios** Los usuarios pueden pertenecer a uno o varios grupos secundarios (`/etc/group`).

Los ficheros /etc/passwd y /etc/group

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Contenido

Introducción

Permisos de ficheros y directorios

Cambio de permisos

La orden umask

- **/etc/passwd** Fichero de usuarios. Para cada usuario del sistema mantiene:
login_id:contraseña_cod:UID:GID:varios:dir_entrada:shell
- **/etc/group** Ficheros de grupos. Para cada grupo mantiene:
nombre_grupo:x:GID:lista_usuarios_grupo_secundario

Permisos de ficheros y directorios

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Contenido

Introducción

Permisos de ficheros y directorios

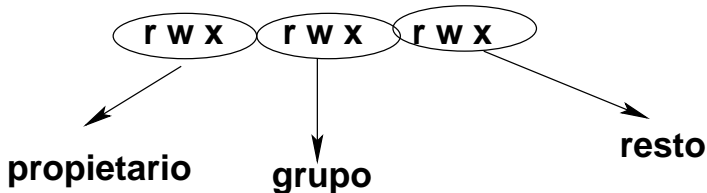
Cambio de permisos

La orden umask

- Los ficheros regulares y los directorios pueden llevar asociados 3 tipos de permisos: lectura (**r**), escritura (**w**) y ejecución (**x**).

```
ls -l /etc/passwd
```

```
-rw-r--r-- 1 root root .... /etc/passwd
```



Ficheros regulares

- **r** Nos permite examinar su contenido
- **w** Nos permite modificar su contenido
- **x** Nos permite ejecutarlo, si se trata de un programa

Directorios

- **r** Nos permite ver el contenido del directorio (**ls**)
- **w** Nos permite crear nuevos ficheros, borrar ficheros existentes, modificar el identificador de un fichero.
- **x** **Permiso de búsqueda** Nos permite buscar en el directorio el número de nodo-i que le corresponde a un fichero a partir de su nombre

Búsqueda de ficheros

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Contenido

Introducción

Permisos de ficheros y directorios

Cambio de permisos

La orden umask

Directorio raíz

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

La búsqueda de /usr
produce el nodo-i 6

Nodo-i 6 de /usr

tamaño tiempos modos
132

El nodo-i 6 indica
que /usr está en
el bloque 132

El bloque 132 es
el directorio de /usr

6	.
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

/usr/ast es el
nodo-i 26

El nodo-i 26
es de /usr/ast

tamaño tiempos modos
406

El nodo-i 26 indica
que /usr/ast es el
bloque 406

El bloque 406 es
el directorio /usr/ast

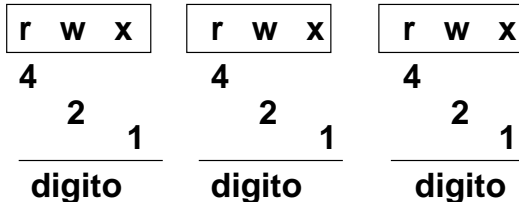
26	.
6	..
64	gran
92	libros
60	buzon
81	linux
17	mtp-ii

/usr/ast/buzon es
el nodo-i 60

- La orden `chmod` permite cambiar los permisos de un fichero

`chmod modo fichero ...`

- Sólo puede cambiar los permisos de un fichero el superusuario (`root`) o el propietario del fichero
- El modo se puede especificar de 2 formas:
 - Números octales** `chmod 750 miprograma`



- Descripción simbólica

- Descripción simbólica

```
chmod ug-x, o+r miprograma
```

[quién] op permiso

<i>quién</i>	<i>op</i>	<i>permiso</i>
u (prop.)	+ (añade permiso)	r
g (grupo)	- (quita permiso)	w
o (otros)	= (asigna abs.)	x
a (todos)		s, t

Ejercicios de ejemplo

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Contenido

Introducción

Permisos de ficheros y directorios

Cambio de permisos

La orden umask

- 1 Suponga que da la orden `ls -l mifichero` y obtiene:

```
-rw--w-r-- user1 grupol ... mifichero
```

¿Qué línea de órdenes debería dar para que el fichero quedara con la máscara de permisos `rw-r---`? El modo debe especificarlo mediante números octales y en forma simbólica.

- 2 Suponga que da la siguiente secuencia de órdenes. Indique de forma razonada qué permisos tendría el fichero tras la ejecución de cada orden.
 - 1 `chmod 351` listado
 - 2 `chmod u+w` listado
 - 3 `chmod g-r` listado
 - 4 `chmod o=x` listado

- Permite especificar los permisos que tendrán los ficheros y directorios en el momento de su creación
`umask [modo]`
- El administrador del sistema establece una máscara por omisión para todos los ficheros del sistema.
- Los usuarios pueden establecer su propia máscara.
- El modo se puede dar de forma octal (indica los permisos que no se van a establecer) y en forma simbólica (se indican los permisos que se van a establecer)
- Hay que tener en cuenta que en el caso de los ficheros regulares el permiso de ejecución nunca se activa al crearlo

Ejercicios de ejemplo

Práctica 3 Permisos (Sesión 1)

Sistemas Operativos

Contenido

Introducción

Permisos de ficheros y directorios

Cambio de permisos

La orden umask

- 1 Supongamos que da la orden `umask 046` ¿Qué máscara de permisos tendrán activados los ficheros ordinarios y directorios que se creen a partir de ese momento?

- 2 Dados los ficheros:

```
drwxr-xr-x  root  alum  /home/alum
drwxr-xr-x  juan  alum  /home/alum/juan
drwxr--r--  juan  alum  /home/alum/juan/docs
-rw-rw-rw-  juan  alum  /home/alum/juan/docs/apuntes
```

¿Podría el usuario `pepe` perteneciente al grupo `alum` dar las órdenes siguientes? ¿Y el usuario `manual` perteneciente al grupo `profs`?

- 1 `ls ~juan`
- 2 `ls ~juan/docs`
- 3 `rm ~juan/docs/apuntes`