

LPI 104.5. Administrar permisos i propietaris de fitxers

Wiki:

http://acacha.org/mediawiki/index.php/LPI_104.5

Objectius

104.5. Administrar permisos i propietaris de fitxers



- **Objectiu:** Els candidats han de ser capaços de controlar l'accés a fitxers mitjançant el correcte ús dels permisos i propietaris dels fitxers.
- **Pes:** 3



Àrees Clau de Coneixement:

- [Administrar els permisos d'accés a fitxers regulars, especials i directoris.](#)
- Utilitzar modes d'accés com [suid](#), [sgid](#) i [sticky bit](#) per a mantenir la seguretat.
- [Conèixer com canviar la màscara de creació de fitxers.](#)
- [Utilitzar el camp de grup per a donar accés als membres del grup.](#)



La següent és una llista parcial de fitxers, termes i utilitats utilitzades:

- [chmod](#)
- [umask](#)
- [chown](#)
- [chgrp](#)



Apunts: [LPI 104.5](#). Administrar permisos i propietaris de fitxers

Control d'accés als fitxers

- ♦ **Linux: sistema operatiu multiusuari real**
 - ♦ Múltiples usuaris treballant alhora al sistema.
 - ♦ Cada usuari disposa d'una carpeta pròpia amb fitxers propis. La carpeta, segons **FHS**, és **/home**.
- ♦ **Mecanisme de Control d'accés als fitxers**
 - ♦ Cal controlar l'accés als fitxers: **sistema permisos**
 - ♦ **Seguretat**: Alguns fitxers contenen informació confidencial o informació essencial pel correcte funcionament del sistema

♦ Sistema de permisos de fitxers Unix-like (POSIX)

- ♦ També conegut com sistema de permisos tradicional de Unix
- ♦ És el sistema utilitzat per defecte a Linux i altres sistemes operatius com MAC OS X
- ♦ Existeixen altres mecanismes com les ACL (l·listes de control d'accés). Es pot implementar també en Linux
- ♦ Els permisos són un mecanisme proporcionat pel sistema de fitxers (ext2, ext3, reiser, xfs...). Alguns sistemes de fitxers com FAT32 o NTFS no suporten aquest sistema

NOTA: Recordeu que a Linux, tot són fitxers (dispositius de maquinari, enllaços, endolls, conductes, etc...). Per tant, un sistema de control d'accés als fitxers és un sistema de control a tot.

♦ Concepte de propietari d'un fitxer

- ♦ Establir els permisos d'un fitxer permet determinar explícitament, **qui** té drets (i quin tipus de drets té) d'accés a aquest arxiu.
- ♦ La combinació de propietari i permisos d'un fitxer, estableix el que un usuari concret pot fer amb aquest fitxer.

♦ Hi han dos nivells de propietat d'un fitxer:

- ♦ El propietari en si (owner)
- ♦ El grup propietari (group o group owner)

- ♦ **Linux classifica als usuaris en tres tipus (entitats):**
 - ♦ **El propietari (owner):** usuari del sistema al que pertany el fitxer. Al crear un fitxer s'assigna com a usuari el nom d'usuari (*login name*) de l'usuari que l'ha creat.
 - ♦ **El grup al que pertany el fitxer (group o group owner):** els fitxers, ha més de pertànyer a un usuari, també pertanyen a un grup. Al crear un fitxer s'assigna com a grup el grup principal de l'usuari que l'ha creat.
 - ♦ **La resta d'usuaris i grups (others):** qualsevol usuari o grup que no sigui el propietari o el grup del fitxer. També es coneixen com els permisos per a la **resta del mon (world permissions)**.



Permisos tradicionals de Unix

◆ Definicions

- ◆ **Mode:** Els drets d'accés a un arxiu s'anomenen també “mode” en Unix.
- ◆ **Entitat:** cadascun dels tipus d'usuaris especificats a l'apartat anterior és conegut com entitat
- ◆ **Bit de permís:** Determina el tipus de permís

◆ Cada entitat pot tenir 3 tipus (bit) de permís:

- ◆ **Permís de lectura (read):** indicat pel caràcter **r**. Aquest permís permet obrir l'arxiu i veure el seu contingut.
- ◆ **Permís d'escriptura (write):** indicat per una **w**. Aquest permís permet modificar el fitxer
- ◆ **Permís d'execució (execution):** indicat per una **x**. Aquest permís indica si un fitxer es pot executar.



Permisos de lectura

♦ Propietats

- ♦ Els fitxers amb permisos de lectura **poden ser copiats** amb l'ordre **cp** a un lloc on l'usuari tingui permisos d'escriptura. Les còpies si es poden modificar, però mai l'original.
- ♦ **En cap cas**, el fet que un fitxer estigui marcat com a **fitxer sense permisos de lectura**, **impedeix** que el nom del fitxer **sigui vist** per altres usuaris. Es pot fer amb els permisos dels directoris.
- ♦ No permet fer **cap tipus de modificació**, ni del nom del fitxer, ni del PATH (ordre mv) ni dels continguts del fitxer.



Permisos de lectura

- ♦ El permís de lectura **no determina si podem canviar els permisos o el propietari** d'un fitxer. Només el propietari d'un fitxer o el superusuari poden canviar els permisos d'un fitxer.

Cal tenir en compte que el propietari (no superusuari) no pot assignar el fitxer a un altre usuari ni a cap grup al que no pertanyi.

- ♦ Es poden fer enllaços simbòlics i durs a un fitxer amb permisos de lectura però això no ens permet modificar el fitxer. Si que podem però eliminar l'enllaç (no elimina el fitxer original).
- ♦ **Directoris:** permet llegir **només els noms** dels fitxers d'aquest directori. No podem però consultar cap altre informació (mida, propietari, permisos...). El directori en si és visible



Permisos de lectura

♦ Exemple

```
$ cd
$ mkdir directori
$ touch directori/a
$ touch directori/b
$ chmod u=r directori
$ ls -l directori/
ls: no s'ha pogut accedir a directori/b: Permission denied
ls: no s'ha pogut accedir a directori/a: Permission denied
total 0
-????????? ? ? ? ?           ? a
-????????? ? ? ? ?           ? b
$ cd directori/
bash: cd: directori/: Permission denied
$ touch directori/c
touch: no s'han pogut canviar les dates de «director/c»:
Permission denied
```



Permisos d'escriptura

♦ Propietats

- ♦ Permeten fer operacions de modificació un fitxer:
 - Eliminar el fitxer amb l'ordre `rm`
 - Modificar els continguts del fitxer
 - Canviar el nom o moure'l a un altre carpeta amb l'ordre `mv`.
 - Tingueu en compte l'sticky bit i el comportament per defecte amb directoris amb permisos d'escriptura.

♦ Metadades

- Es poden modificar les marques de temps
- No es poden modificar els permisos (excepte propietari o root)
- No es pot modificar el propietari (excepte usuari root).
- El propietari pot modificar el grup, si el grup escollit és un dels grups als que pertany l'usuari



Permisos d'escriptura

♦ Directoris

- ♦ Pot ser una mica confús! Els directoris són un tipus especial de fitxers
- ♦ L'usuari amb accessos d'escriptura en un directori pot:
 - Crear nous fitxers dins del directori
 - Esborrar fitxers (rm) o canviar el nom/moure (mv) els fitxers, encara que no sigui el propietari del fitxers a modificar i/o no tingui permisos d'escriptura.
 - Es pot modificar el sticky bit per tal de modificar aquest comportament.

Tenir permisos per eliminar un fitxer no implica que puguem modificar els continguts. Sinó tenim permisos per modificar un fitxer, però esta en una carpeta en la que si que tenim permisos de modificació, podem eliminar o moure el fitxer però no modificar els seus continguts

Permisos d'escriptura

♦ Seguim amb l'exemple anterior

```
$ cd
$ chmod u=rw directori
$ cd directori/
bash: cd: directori/: Permission denied
      Atenció com no podem entrar (permisos)!:
$ mkdir directori/c
mkdir: no s'ha pogut crear el directori «directori/c»:
Permission denied
$ touch directori/c
touch: no s'han pogut canviar les dates de «directori/c»:
Permission denied
```

!!!!??????

Estem pendants de conèixer els detalls dels permisos
d'execució en directoris



Permisos d'execució

♦ Propietats

- ♦ Només tenen sentit en fitxers ordinaris o directoris. S'ignoren en altre tipus de fitxers com per exemple els dispositius.
- ♦ No implica que sigui un executable! p. ex. es pot tenir un fitxer de configuració en text pla amb permisos d'execució, però cal evitar-ho per no dur a confusions.

♦ Directoris:

- ♦ Els directoris NO s'executen. S'utilitza el permís amb altres objectius



Permisos d'execució

♦ Directoris

- ♦ Permet establir el directori com a directori de treball, és a dir, podem entrar-hi amb l'ordre **cd**
- ♦ Es necessiten permisos d'execució per accedir a la informació dels **inodes** dels fitxers del directori.
- ♦ Es necessari per cercar els fitxers i per aquesta raó sovint també s'anomena permís de cerca

Els directoris són uns **fitxers de dades especials** que tenen dos informacions per cada fitxer del directori:

- **Nom del fitxer:** si accedeix amb els permisos de lectura
- **Inode del fitxer:** si accedeix amb els permisos de lectura (aka cerca)



Permisos d'execució

♦ Seguim amb l'exemple anterior

- ♦ El permís d'execució (poder accedir a l'inode) és necessari per a moltes operacions
- ♦ Cal permisos de lectura a foo `$ cat /home/user/foo`
- ♦ També cal permisos d'execució a /home i a /home/user
- ♦ Ordres com touch, mv, rm... necessiten accés a l'inode

Els permisos no determinen exactament el que poden fer les ordres! Determinen el que poden fer les **crides del sistema** operatiu. Cada crida del sistema indica quins permisos necessita als manuals (man 2)

```
$ sudo apt-get install manpages-dev
$ man 2 write | man 2 read | man 2 open | man 2 lseek ...
```

Exemples

♦ No tenim cap permís en una carpeta

```
$ cd
$ sudo mkdir -p carpeta/carpeta_oculta
$ sudo chmod 700 carpeta
$ ls carpeta
ls: no s'ha pogut obrir el directori carpeta: Permission denied
$ sudo ls carpeta
carpeta_oculta
$ cd carpeta/
bash: cd: carpeta/: Permission denied
$ rm -r carpeta/
rm: voleu descendir al directori protegit contra escriptura «carpeta/»?
La carpeta no s'ha esborrat!

$ ls -la
total 24
drwxr-xr-x  4 sergi sergi  4096 2009-11-14 10:06 .
drwx----- 44 sergi sergi 12288 2009-11-14 10:04 ..
drwx-----  3 root  root   4096 2009-11-14 10:06 carpeta
```



El superusuari i els permisos

♦ Superpoders (poders de supervaca ;-))

- ♦ Les restriccions d'accés no s'apliquen al usuari root. Aquest usuari pot llegir o modificar qualsevol fitxer, fins i tot aquells als que no té permisos, per exemple un fitxer amb permisos 000.

♦ “Excepcions”

- ♦ Es possible que el superusuari no pugui modificar un sistema de fitxers muntat com a només lectura. Cal però, fer notar que root podria tornar a muntar el sistema amb uns altres permisos
- ♦ Fins i tot el superusuari necessita també el bit d'execució per executar un programa (el pot assignar si cal)



El superusuari i els permisos

♦ Exemple

```
$ su
# whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
   /usr/share/man/man1/ls.1posix.gz
# ls -l /bin/ls
-rwxr-xr-x 1 root root 96216 2008-06-27
02:31 /bin/ls
# chmod 644 /bin/ls
# ls
bash: /bin/ls: Permission denied
# chmod 755 /bin/ls
```

♦ Què és supervaca? Un ou de pasqua (broma)

```
$ apt-get
....
      This APT has Super Cow Powers.
```

```
$ apt-get moo
          (__)
          (oo)
        /-----\
       /  |      ||
      *  /\---/\
         ~~      ~~
...."Have you mooed
today?"...
```

➤ **ls amb l'opció long list (-l)**

```
$ ls -l
total 4948900
-rw-r--r-- 1 vlino vlino 644666 2009-09-11 08:03 des-Per.pdf
```

- ❖ Cadena de caràcters de la primera columna
- ❖ Notació simbòlica
- ❖ 4 seccions
 - Primer caràcter: tipus de fitxer
 - 3 grups de 3 caràcters:
 - Permisos de cada entitat
 - En notació simbòlica





Notació simbòlica

♦ Es basa en els caràcters:

- ♦ **r** (read), **w** (write) i **x** (execution) per indicar els permisos (**modes**).
- ♦ El símbol **-** indica que el permís d'aquella posició no està activat.
- ♦ **u** (user), **g** (group) i **o**(others) per indicar les **entitats** afectades.
- ♦ El caràcter **a** equival a **ugo** i representa les tres entitats.
- ♦ Veurem més exemples quan consultem l'ordre **chmod**

Notació binària i notació octal

♦ Notacions alternatives

- ♦ Utilitzades en comandes relacionades amb permisos

Permisos (simbòlic)	Notació binària	Notació octal
---	000	0
--X	001	1
-W-	010	2
-WX	011	3
r--	100	4
r-X	101	5
rW-	110	7
rWX	111	8



Notació binària i notació octal

♦ S'utilitza molt la notació octal

- ♦ Exemples habituals:
- ♦ 755 (rwxr-xr-x): El propietari ho pot fer tot i la resta llegir i executar. Permís habitual de les carpetes
- ♦ 644 (rw-r--r--): El propietari pot llegir i modificar i la resta només llegir. Permís habitual de les carpetes
- ♦ 777 (rwxrwxrwx): tots els permisos. No és segur
- ♦ 000 (-----): Cap permís
- ♦ 700 (rwx-----): Tots els permisos només per al propietari

Bits especials de permisos

♦ SUID (Set User ID)

- ♦ S'utilitza conjuntament amb **fitxers amb permisos d'execució**
- ♦ Indica al sistema operatiu que el programa **s'ha d'executar amb els permisos del propietari del fitxer** i no pas amb els permisos de l'usuari que executa el fitxer.
- ♦ Es pot utilitzar per tal d'**executar fitxers com a superusuari sense ser root**. El fitxer ha de pertànyer al superusuari, ser executable i tenir el bit especial d'execució SUID.
- ♦ Aquesta opció es força utilitzada en alguns serveis i programes
- ♦ Els programes que s'executen d'aquesta forma són anomenats SUID root.
- ♦ Els fitxers amb aquest permís s'indiquen amb una **s al bit d'execució del propietari**. (o S majúscula si no és executable)



Bits especials de permisos

♦ SUID exemple

- ♦ Per exemple, l'ordre ping en alguns sistemes com Ubuntu:

```
# ls -l /bin/ping  
-rwsr-xr-x 1 root root 30856 2007-12-10 18:33 /bin/ping
```

- ♦ La creació de paquets ICMP (protocol de ping) requereix accés de superusuari.
- ♦ Activar el SUID en un directori no té cap sentit ni utilitat (s'ignora)
- ♦ Per seguretat, només funciona amb fitxers executables binaris (no guions de llenguatges interpretats)



Bits especials de permisos

♦ **SGID (Set Group ID)**

- ♦ És similar a SUID però estableix que el grup del programa executable és el grup del fitxer i no pas el grup de l'usuari que executa el fitxer.
- ♦ S'indica amb una **s al bit d'execució del grup**.
- ♦ **Directoris:** Quan el bit SGID s'estableix en un directori, els fitxers nous o directoris creats en aquest directori heretaran el grup del directori i no pas el grup de l'usuari que crea el directori o fitxer.

IMPORTANT: Tant el SUID com el GUID són bits potencialment insegurs. Només s'utilitzen quan no hi ha un altre remei. Cal que els programes que els utilitzin estiguin ben programats per tal d'evitar escalades de privilegis

Bits especials de permisos

♦ L'Sticky bit (bit enganxifós)

- ♦ Ha tingut diferents significats durant la història de Unix.
- ♦ A la majoria de sistemes operatius moderns (com Linux) s'utilitza per **evitar que certs fitxers siguin esborrats per persones que no siguin propietàries del fitxer**.
- ♦ Quan aquest bit està present en un directori, els fitxers del directori només poden ser eliminats pels seus propietaris o pel superusuari.
- ♦ S'indica amb una **t al bit d'execució de l'entitat altres**.
- ♦ S'ignora en els fitxers. Històricament feia que l'executable seguís en memòria fen més ràpida a seva execució.

♦ Exemple de carpeta amb sticky bit

- ♦ El directori temporal /tmp permet a qualsevol usuari emmagatzemar els seus propis arxius, però per evitar que esborri els arxius d'altres, s'ha d'activar el permís Sticky:

```
$ ls -ld /tmp
drwxrwxrwt 5 root root 4096 jun 3 02:32 /tmp/
$ touch /tmp/arxiudewilly
$ ls -l /tmp
total 0
-rw-r--r-- 1 nicolas curs 0 jun 2 02:32 arxiudenicolas
-rw-r--r-- 1 willy      tecnic 0 jun 3 02:32 arxiudewilly
$ rm /tmp/arxiudenicolas
rm:borrar un archivo protegido en escritura archivo normal vacío
'/tmp/arxiudenicolas'? S
rm:no se puede borrar '/tmp/arxiudenicolas': Operation not permitted
```

Bits especials de permisos

♦ Notació

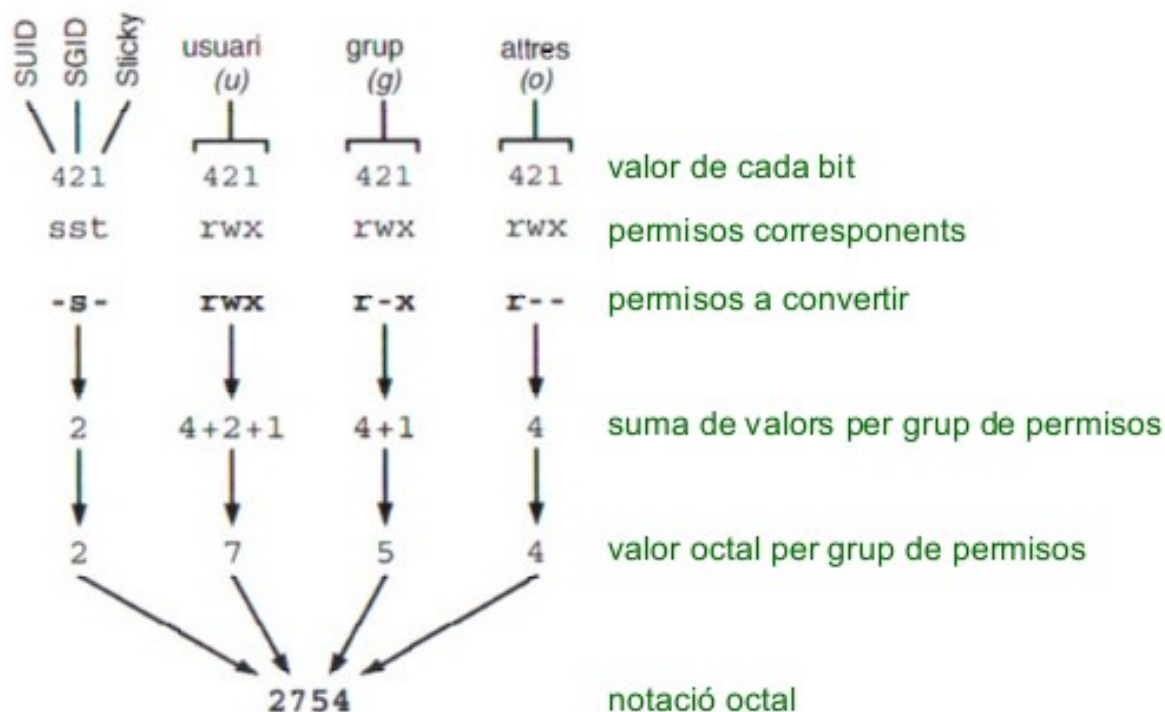
Permisos (simbòlic)	Notació binària	Notació octal
---	000	0
--t	001	1
-s-	010	2
-st	011	3
s--	100	4
s-t	101	5
st-	110	7
stt	111	8



Conversió entre notacions

◆ Exemple

- ◆ El bit dels permisos especials es pot obviar o posar un 0 (cap bit especial)



Taula resum i enllaços

♦ Taula

Permís	Símbol (en notació simbòlica)	Símbol (en notació octal)	Permisos sobre fitxers	Permisos sobre directoris
Read	r	4	Examinar els continguts, copiar el fitxer (ordre cp).	Llistar el contingut del directori
Write	w	2	Modificar els continguts, modificar el nom o el camí (ordre mv) o eliminar el fitxer (ordre rm).	Crear i esborrar fitxers dins del directori.
Execute	x	w	Executar fitxers	Llegir i escriure fitxers ja existents al directori.

♦ Els enllaços simbòlics sempre apareixen a la sortida de ls amb tots els permisos:

- \$ ls -l /etc/rc3.d/S16ssh lrwxrwxrwx ...
- En cap cas un enllaç simbòlic ens permetrà accedir a un fitxer al que no teníem accessos.
- Sempre podem crear un enllaç simbòlic a un fitxer que ens sigui visible en una carpeta en la que tinguem permisos d'escriptura.
- Les operacions de lectura i de modificació de l'enllaç són operacions de lectura i modificació al fitxer enllaçat i s'apliquen les restriccions d'aquest
- Només pot eliminar un enllaç el propietari de l'enllaç



Canviar els permisos d'un fitxer. chmod

♦ **chmod** (change mode)

- ♦ Recordeu: mode és un sinònim de permís
- ♦ Sintaxi:

```
$ chmod [-R] <permisos> <arxiu ...>
```
- ♦ Permet modificar els permisos d'un fitxer
- ♦ Suporta tant la notació octal com la simbòlica
- ♦ Només el propietari de l'arxiu i l'administrador del sistema (root) poden utilitzar-lo
 - Cal tenir en compte que el propietari no podrà assignar el fitxer a un altre usuari (només pot canviar el grup i sempre que sigui a un grup al qual pertany).

Canviar els permisos d'un fitxer. chmod

♦ Sintaxi

```
$ chmod [-R] <permisos> <arxiu ...>
```

- ♦ **-R**: indica a la comanda que apliqui el canvi de forma recursiva (només aplicable a directoris).
- ♦ **permisos**: en notació simbòlica o en notació octal.
- ♦ **arxiu ...** : el camí al arxiu o arxius que es volen modificar.

♦ Notació simbòlica

- ♦ **r** (read), **w** (write) i **x** (execution) per indicar els permisos.
- ♦ El símbol **-** indica que el permís d'aquella posició no està activat.
- ♦ **u** (user), **g** (group) i **o**(others) per indicar les entitats afectades.
- ♦ El caràcter **a** equival a **ugo** i representa les tres entitats.

Canviar els permisos d'un fitxer. chmod

- ♦ **Afegir/treure permisos podem utilitzar els símbols:**
 - ♦ **+**: indica que s'ha d'activar el permís.
 - ♦ **-**: indica que s'ha de treure el permís.
 - ♦ **=**: permet definir els tres permisos en una sola etapa per a l'entitat indicada.
 - ♦ **Sintaxi:** <entitat(s)>[+ -=] <permís(os)>



Canviar els permisos d'un fitxer. chmod

♦ Notació octal

♦ Sintaxi:

```
$ sudo chmod xyz fitxer ...
```

- x: Són els permisos del propietari en octal
 - y: Són els permisos del grup en octal
 - z: Són els permisos d'altres en octal
- ### ♦ Es pot calcular fent una suma
- Lectura: val 4, escriptura: val 2 i execució: val 1



Canviar els permisos d'un fitxer. chmod

♦ Permisos especials

♦ S'activen amb:

- La lletra s i la entitat u per al SUID
- La lletra s i la entitat g per al SGID
- La lletra t i la entitat o per a l'Sticky bit.

♦ Exemples:

- u+s: afegeix (+) el bit SUID (s i u).
- g+s: afegeix (+) el bit SGID (s i g).
- o+t: afegeix (+) el bit Sticky (t i o).

Canviar els permisos d'un fitxer. chmod

♦ Examples

- ♦ **u+x**: afegeix (+) el permís d'execució (x) per al propietari (u).
- ♦ **g-w**: suprimeix (-) el permís de modificació (w) per al grup (g).
- ♦ **o-rw**: suprimeix (-) els permisos de lectura (r) i de modificació (w) per als altres (o).
- ♦ **ug-x**: suprimeix (-) el permís d'execució (x) per al propietari (u) i el grup (g).
- ♦ **a+r**: afegeix (+) el permís de lectura (r) per a totes les entitats (a).
- ♦ **u=rw**: estableix (=) els permisos de lectura (r) i de modificació (w) i suprimeix el permís d'execució (x) per al propietari (u).

Canviar els permisos d'un fitxer. chmod

♦ Canviar permisos de forma massiva

```
$ sudo chmod -R 644 carpeta
```

- ♦ Atenció!: Executar aquesta comanda pot tenir repercussions imprevistes segons com i on es faci. Per exemple segurament no volem aplicar els mateixos permisos a carpetes que a fitxers!
- ♦ Consulteu la comanda **find** per canviar permisos només a fitxers o només a directoris:

Canviar els propietaris. chown

♦ Chown (change owner)

- ♦ Permet canviar el propietari d'un fitxer.
- ♦ Només pot canviar el propietari d'un fitxer el superusuari del sistema (però el propietari d'un fitxer pot canviar el grup d'un fitxer sempre que assigni un grup al qual pertany).

♦ Sintaxi: `$ chown [OPTION]... [OWNER] [:[GROUP]] FILE...`

♦ Exemples:

- Establir propietari i grup d'un fitxer:

```
$ sudo chown usuari:grup fitxer
```



Canviar els propietaris. chown

♦ Exemples

- ♦ Establir propietari d'un fitxer: `$ sudo chown owner fitxer`

♦ **\$ sudo chown usuari:grup fitxer**

- ♦ Establir només el grup d'un fitxer:

```
$ sudo chown :grup fitxer
```

- ♦ Establir propietari de forma recursiva:

```
$ sudo chown -R usuari:grup carpeta
```

- ♦ Hi han opcions com les habituals verbose (v) i com tractar els enllaços.



Canviar el grup

♦ Chgrp (change group)

♦ Permet canviar només el grup

- Sintaxi:

```
$ chgrp [-R] <grup> <arxiu ...>
```
- -R: aplicar l'ordre de forma recursiva
- El propietari pot cedir l'arxiu a qualsevol grup al qual pertanyi:

♦ El grup per defecte assignat als nous arxius és el grup principal de l'usuari que l'ha creat, a menys que estigui habilitat el permís SGID en el directori on es crea el fitxer.



id i groups

♦ Mostren informació del usuari i grups

```
$ groups  
sergi adm dialout cdrom plugdev lpadmin admin sambashare
```

```
$ id  
uid=506(alex) gid=502(vendes) grupos=502(vendes),503(compta)  
$ touch arxiu  
$ ls -l arxiu  
-rw-r--r-- 1 alex vendes 0 jun 3 03:58 arxiu  
$ chgrp comptabilitat arxiu  
-rw-r--r-- 1 alex compta 0 jun 3 03:58 arxiu
```

Gestió d'usuaris i grups

♦ No es veu a l'examen 101

- ♦ És tema de l'examen 102
- ♦ 107.1 Gestió d'usuaris i comptes de grup i els fitxers de sistema relacionats





Reconeixement 3.0 Unported

Sou lliure de:



copiar, distribuir i comunicar públicament l'obra



fer-ne obres derivades

Amb les condicions següents:



Reconeixement. Heu de reconèixer els crèdits de l'obra de la manera especificada per l'autor o el llicenciador (però no d'una manera que suggereixi que us donen suport o rebeu suport per l'ús que feu l'obra).

- Quan reutilitzeu o distribuïu l'obra, heu de deixar ben clar els termes de la llicència de l'obra.
- Alguna d'aquestes condicions pot no aplicar-se si obteniu el permís del titular dels drets d'autor.
- No hi ha res en aquesta llicència que menyscabi o restringeixi els drets morals de l'autor.

Advertiment

Els drets derivats d'usos legítims o altres limitacions reconegudes per llei no queden afectats per l'anterior
Això és un resum fàcilment llegible del text legal (la llicència completa).

<http://creativecommons.org/licenses/by/3.0/deed.ca>