

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

## OBJECTIUS DE LA PRÀCTICA

- Identificar i comprendre els permisos de fitxers i directoris en Linux.
- Convertir permisos simbòlics a decimals i viceversa.
- Manipulació avançada de propietaris i permisos amb `chown`, `chgrp` i `chmod`.
- Gestionar els permisos per defecte amb la màscara `umask`.
- Aplicar permisos especials com SUID, SGID i Sticky Bit.
- Treballar amb fitxers i directoris compartits en grups.

## DESENVOLUPAMENT DE LA PRÀCTICA

### Identificació de permisos

En un sistema Linux, cada fitxer i directori té associats permisos que determinen quines accions poden realitzar diferents tipus d'usuaris sobre ells. Els permisos es divideixen en tres categories:

- **Propietari (user):** Normalment, l'usuari que ha creat el fitxer o directori.
- **Grup (group):** Un conjunt d'usuaris amb permisos específics sobre el fitxer o directori.
- **Altres (others):** Qualsevol altre usuari del sistema.

Cada fitxer o directori pot tenir permisos de:

- **Lectura (r):** Permet veure el contingut del fitxer o llistar el directori.
- **Escriptura (w):** Permet modificar el contingut del fitxer o crear/eliminar fitxers en un directori.
- **Execució (x):** Permet executar un fitxer com a programa o accedir a un directori.

Per veure els permisos actuals utilitzem la comanda: `ls -l`

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

Aquesta comanda mostra els permisos amb un format com aquest:

```
-rwxr-xr-- 1 user1 grup1 5120 març 21 14:00 exemple.txt
```

La primera columna indica els permisos del fitxer:

- El primer caràcter indica el tipus (- per fitxers normals, d per directoris).
- Els següents 9 caràcters es divideixen en blocs de 3:
  - rwx: Permisos del propietari.
  - r-x: Permisos del grup.
  - r--: Permisos per a altres usuaris.

### Activitat 1:

- Crear un directori practica\_permisos i accedeix-hi.  
**mkdir ~/practica\_permisos && cd ~/practica\_permisos**
- Crear un fitxer anomenat exemple.txt amb contingut aleatori i comprova els permisos:

```
echo "Aquest és un fitxer d'exemple" > exemple.txt  
ls -l exemple.txt
```

- Interpreta els permisos mostrats i apunta'ls en un document apart.

```
-rw-r--r-- 1 user1 user1 28 març 21 14:00 exemple.txt
```

**El fitxer creat té permisos de lectura i escriptura per al propietari, i només lectura per al grup i altres.**

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

## *Equivalències de permisos en octal*

Els permisos es poden representar de manera numèrica utilitzant valors octals (de 0 a 7) que representen combinacions dels permisos bàsics (lectura, escriptura, execució).

- **Lectura (r)** = 4
- **Esriptura (w)** = 2
- **Execució (x)** = 1

Per tant, les combinacions possibles són:

- 7 (rwx) = 4 + 2 + 1 = Total permís
- 6 (rw-) = 4 + 2 = Lectura i escriptura
- 5 (r-x) = 4 + 1 = Lectura i execució
- 4 (r--) = Només lectura

Exemple d'ús amb chmod:

```
chmod 750 exemple.txt
```

Això assigna:

- **Propietari:** rwx (7)
- **Grup:** r-x (5)
- **Altres:** Cap permís (0)

També podem fer servir la notació simbòlica:

```
chmod u=rwx,g=rx,o= exemple.txt
```

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

### Activitat 2:

- Canvia els permisos de exemple.txt per a que:
  - El propietari tinga tots els permisos (lectura, escriptura, execució).
  - El grup tinga només lectura i execució.
  - Els altres no tinguen cap permís.

**chmod 750 exemple.txt**

**ls -l exemple.txt**

**-rwxr-x--- 1 user1 user1 28 març 21 14:00 exemple.txt**

**El propietari té permisos complets (rwx), el grup té lectura i execució (r-x), i altres no tenen cap permís (---).**

- Prova també de canviar els permisos amb la notació simbòlica:

**chmod a=r exemple.txt**

**ls -l exemple.txt**

**-r--r--r-- 1 user1 user1 28 març 21 14:00 exemple.txt**

- Comprova els permisos i compara amb els anteriors.

**El primer canvi (chmod 750) assigna permisos molt restrictius, donant tot el control al propietari i limitant l'accés del grup a només lectura i execució.**

**El segon canvi (chmod a=r) assigna el permís mínim possible, que és la lectura per a tothom, sense cap opció d'escriptura o execució.**

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

## Manipulació de propietaris i permisos

La propietat i el grup d'un fitxer o directori es poden canviar amb les comandes `chown` i `chgrp`. Normalment, només el propietari d'un fitxer o l'usuari `root` pot modificar els permisos i propietats.

- `chown`: Modifica el propietari d'un fitxer o directori.
- `chgrp`: Modifica el grup d'un fitxer o directori.

Exemple d'ús:

```
sudo chown usuari1 exemple.txt  
sudo chgrp projecte exemple.txt
```

Comprovar els permisos amb: `ls -l exemple.txt`

### Activitat 3:

- Crea dos usuaris nous (`usuari1` i `usuari2`) i un grup (`projecte`).

```
sudo useradd -m usuari1
```

```
sudo useradd -m usuari2
```

L'opció `-m` assegura que es crea un directori personal (`/home/usuari1` i `/home/usuari2`).

```
sudo groupadd projecte
```

- Assigna `usuari1` i `usuari2` al grup `projecte`.

```
sudo usermod -aG projecte usuari1
```

```
sudo usermod -aG projecte usuari2
```

- Canvia el grup propietari de `exemple.txt` a `projecte`.

```
sudo chgrp projecte exemple.txt
```

- Permet que l'usuari1 tinga permisos complets sobre el fitxer i usuari2 només permisos de lectura.

## Pràctica 2. Gestió de permisos en Linux

MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS

CFGS DAM

UD6. Usuaris, grups i permisos

Nom alumne:

### Usuari

### Permisos

usuari1: rw- (lectura i escriptura)

usuari2 (projecte): r-- (només lectura)

Altres: --- (cap permís)

`sudo chown usuari1 exemple.txt`

`sudo chmod 640 exemple.txt` o `sudo chmod 740 exemple.txt`

- Comprova i documenta els resultats.

`ls -l exemple.txt`

`-rw-r----- 1 usuari1 projecte 28 març 21 14:00 exemple.txt`

## Permisos per defecte (umask)

Umask és una configuració que defineix quins permisos es posen per defecte quan es creen fitxers o directoris nous.

### Com funciona realment umask?

El sistema de permisos en Linux utilitza bits per a definir els permisos. Anem a veure-ho en detall:

#### 1. Permisos per defecte:

- *Fitxers*: 666 (rw-rw-rw-) (lectura i escriptura per a tots) → Binari: 110 110 110
- *Directoris*: 777 (rwxrwxrwx) (lectura, escriptura i execució per a tots) → Binari: 111 111 111

#### 2. Valor de umask:

Suposem que umask és 027 → Binari: 000 010 111

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

### 3. Complement a bits de la umask:

- umask es converteix en l'oposat dels seus bits (0 es transforma en 1 i viceversa).
- 027 → 000 010 111 → **Complement a bits: 111 101 000**

### 4. AND lògic:

- Ara, es fa un **AND lògic entre els permisos per defecte i el complement de la umask.**

### 5. Exemple pràctic per a un fitxer:

- Permisos per defecte:** 666 → 110 110 110
- Umask (027) en binari:** 000 010 111
- Complement a bits:** 111 101 000
- AND lògic:**

```

110 110 110 (Permisos per defecte)
111 101 000 (Complement de la umask)
-----
110 100 000 (640 en decimal) → -rw-r----
```

### 6. Exemple pràctic per a un directori:

- Permisos per defecte:** 777 → 111 111 111
- Umask (027) en binari:** 000 010 111
- Complement a bits:** 111 101 000
- AND lògic:**

```

111 111 111 (Permisos per defecte)
111 101 000 (Complement de la umask)
-----
111 101 000 (750 en decimal) → drwxr-x---
```

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

### Resum important:

- **No és una resta aritmètica.**
- És un **AND lògic** entre els permisos per defecte i el complement del umask.

Per comprovar el umask actual executa: *umask*

Per canviar-la temporalment: *umask 027*

Per a canviar la umask permanentment (és a dir, que s'aplique sempre que l'usuari inicié sessió), has de modificar uns fitxers de configuració en Linux. Depenent de si vols aplicar-ho a un **usuari concret** o a **tots els usuaris del sistema**, s'ha d'editar un fitxer o un altre.

### Canviar la umask per a un usuari concret

Si vols que l'ajust afecte només a un usuari concret, has de modificar el fitxer de configuració del seu shell.

#### 1. Edita el fitxer de configuració del shell de l'usuari:

- Si l'usuari utilitza **Bash** (molt comú): `nano ~/.bashrc`

#### 2. Afig la línia següent al final del fitxer: `umask 027`

Açò estableix el umask a 027 per a aquest usuari,

#### 3. Guarda el fitxer i ix.

#### 4. Aplica els canvis sense tancar la sessió:

`source ~/.bashrc` # O `~/.zshrc` si utilitzes `zsh`

### Canviar la umask per a tots els usuaris

Si vols que aquest canvi afecte a **tots els usuaris del sistema**, has de modificar fitxers globals.



Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

**1. Edita un dels següents fitxers (segons la distribució de Linux):**

- /etc/profile (afecta tots els shells)
- /etc/bash.bashrc (afecta només Bash)
- /etc/login.defs (per a sistemes que utilitzen PAM i defineix valors per defecte com UMASK 027)

**2. Afig la línia següent al final del fitxer:** umask 027

**3. Guarda el fitxer i ix.**

**4. Reinicia la sessió o reinicia el sistema per a aplicar els canvis.**

També pots utilitzar: source /etc/profile

**Comprovar que el canvi s'ha aplicat**

Per assegurar-te que la umask s'ha configurat correctament, pots executar:  
umask

Hauria de mostrar 0027 o el valor que hages configurat.

**Activitat 4:**

- Comprova la umask actual i modifica-la per veure com afecta la creació de fitxers nous.

**Per defecte és normalment 022, que permet permisos 644 per fitxers i 755 per directoris.**

**umask 022**

**touch fitxer1.txt**

**ls -l fitxer1.txt**

**-rw-r--r-- 1 user1 user1 0 març 21 14:20 fitxer1.txt**

- Documenta les diferències entre una umask de 022 i una de 077.

**umask 077**

**touch fitxer\_privat.txt**

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

```
ls -l fitxer_privat.txt
```

```
-rw----- 1 user1 user1 0 març 21 14:20 fitxer_privat.txt
```

## Permisos especials (Ampliació)

A Linux existeixen permisos especials que proporcionen funcionalitats avançades i control més detallat sobre l'accés als fitxers i directoris.

### SUID (Set User ID)

- Quan un fitxer executable té activat el bit SUID, s'executa amb els privilegis del propietari del fitxer en comptes de l'usuari que l'executa.
- Exemple típic: /usr/bin/passwd, que permet a qualsevol usuari canviar la seua contrasenya però s'executa amb permisos de root.

Exemple d'ús:

```
sudo chmod u+s exemple_suid
```

```
ls -l exemple_suid
```

```
-rwsr-xr-x 1 usuari1 grup1 5120 març 21 14:00 exemple_suid
```

### SGID (Set Group ID)

- Aplicat a fitxers executables: L'execució es fa amb els permisos del grup propietari, no de l'usuari que l'executa.
- Aplicat a directoris: Els fitxers nous creats dins del directori hereten el grup propietari del directori, no el grup de l'usuari que els crea.

Exemple d'ús:

```
sudo chmod g+s projecte_dir
```

```
ls -ld projecte_dir
```

## Pràctica 2. Gestió de permisos en Linux

MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS

CFGS DAM

UD6. Usuaris, grups i permisos

Nom alumne:

```
drwxr-sr-x 2 usuari1 projecte 4096 març 21 14:10 projecte_dir
```

### Sticky Bit

- Generalment s'utilitza en directoris públics com /tmp per evitar que un usuari pugui eliminar fitxers d'un altre.
- Quan un directori té Sticky Bit activat, només el propietari del fitxer o l'administrador poden esborrar o modificar els fitxers dins d'ell.

Exemple d'ús:

```
sudo chmod +t /tmp/public_dir
```

```
ls -ld /tmp/public_dir
```

```
drwxrwxrwt 2 root root 4096 març 21 14:12 public_dir
```

### Activitat 5:

- Crea un fitxer exemple\_suid i aplica-li el SUID.

```
sudo chmod u+s exemple_suid
```

```
ls -l exemple_suid
```

```
-rwsr-xr-x 1 usuari1 grup1 5120 març 21 14:00 exemple_suid
```

El fitxer s'executarà amb els permisos del propietari en comptes del qui l'executa.

- Crea un directori projecte\_dir amb SGID i comprova com afecta als fitxers creats dins.

```
sudo chmod g+s projecte_dir
```

```
ls -ld projecte_dir
```

```
drwxr-sr-x 2 usuari1 projecte 4096 març 21 14:10 projecte_dir
```

Els fitxers creats dins hereten el grup propietari del directori.

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

- Crea un directori /tmp/public\_dir i aplica-li Sticky Bit. Prova a eliminar fitxers d'altres usuaris.

**chmod +t /tmp/public\_dir**

**ls -ld /tmp/public\_dir**

**drwxrwxrwt 2 root root 4096 març 21 14:12 public\_dir**

**Només el propietari dels fitxers dins public\_dir els pot eliminar.**

## Comprovacions finals

En aquesta secció comprovarem si els permisos aplicats s'han configurat correctament i es comporten com esperàvem.

### Activitat 6:

- Llista tots els fitxers i directoris creats amb ls -l i ls -ld.

**ls -l**

**ls -ld projecte\_dir /tmp/public\_dir**

- Documenta els permisos actuals de cada fitxer i explica si funcionen segons l'objectiu que es volia aconseguir.
- Reflexiona sobre la diferència entre permisos simbòlics i permisos octals.

**Permisos simbòlics: Són una representació textual dels permisos, com ara rwxr-xr--, on:**

**Permisos octals: Són una representació numèrica dels permisos. Cada permís (lectura, escriptura i execució) es representa amb un nombre.**

**Conclusió: Els permisos simbòlics ofereixen una visualització més fàcil de comprendre de qui té accés a què, mentre que els permisos octals són més compactes i utilitzats en configuracions ràpides.**

Pràctica 2. Gestió de permisos en Linux	
MÒDUL PROFESSIONAL: SISTEMES INFORMÀTICS	CFGS DAM
UD6. Usuaris, grups i permisos	
Nom alumne:	

- Considera possibles situacions on l'ús de SUID, SGID i Sticky Bit pot ser útil.

**SUID (Set User ID):** Quan un fitxer executat amb el bit SUID és executat per un usuari, aquest s'executa amb els permisos del propietari del fitxer, no amb els permisos de l'usuari que l'executa.

Útil per a: Programari que necessita permisos elevats per realitzar una tasca específica (per exemple, el comandament passwd per canviar contrasenyes, ja que necessita permisos de superusuari per modificar fitxers del sistema).

**SGID (Set Group ID):** Quan un fitxer executat amb SGID es llança, s'executa amb els permisos del grup del fitxer. Per als directoris, els fitxers creats dins del directori heretaran el grup del directori, no del creador del fitxer.

Útil per a: Compartir fitxers entre membres d'un grup. Per exemple, en un directori de treball en equip on els fitxers creats per un usuari es desitja que tinguin el mateix grup de treball.

**Sticky Bit:** S'aplica a directoris, evitant que els usuaris esborrin o moguin fitxers creats per altres usuaris en un directori compartit.

Útil per a: Directoris públics com /tmp on múltiples usuaris poden escriure, però no poden eliminar fitxers dels altres usuaris.

**En resum:** El SUID, SGID i Sticky Bit són útils per garantir que els usuaris tinguin els permisos adequats en situacions específiques com l'execució de programes amb permisos elevats o en directoris compartits.