

201 - Permisos de directoris

Curs 2020 - 2021

ASIX M01-ISO UF1-A01-03 Permisos de directoris

Permisos	2
Descripció	2
Permisos de directoris	2
Permisos de directoris	2
Estructura de dades d'un directori	3
Diferència entre els permisos de directori i de fitxer	6
Exercicis d'exemple	8

Permisos

Descripció

Conceptes clau:

- ☐ Elements d'una entrada de directori
- ☐ Diferència entre fitxers i directoris.
- ☐ i-nodes
- ☐ diferència entre el permís del directori i del fitxer

Ordres a treballar:

- ☐ ls -l
- ☐ stat
- ☐ chmod

Permisos de directoris

Permisos de directoris

Els permisos bàsics en sistemes GNU/Linux són fàcils d'entendre si els tractem a nivell individual. Un fitxer amb permisos `rw-` és clar que significa que es té permís de lectura i modificació. Un fitxer amb permisos `r--` només de lectura i amb permisos `---` no es pot accedir. El mateix passa amb directoris, amb `rwX` es té control total, amb `r-X` només d'accés/l·listar i amb `---` no es pot fer res.

Ara bé, que passa amb la combinació de permisos entre directoris i fitxers?

Sovint un aspecte difícil d'entendre per l'alumne és a què s'apliquen els permisos de directoris i de fitxers i com s'interrelacionen. Sovint es confón el permís del fitxer amb el del directori... però anem a veure-ho.

Un fitxer: imaginem que un fitxer és una caixa tancada que dins seu té contingut. Si es té el permís `r` sobre el fitxer es pot obrir la caixa i llegir-ne el contingut, però no es pot modificar. Si es té el permís `w` llavors a la caixa s'hi pot afegir, treure i modificar contingut. La `w` en el fitxer indica que es pot modificar el contingut. En CAP cas la `w` en el fitxer indica si es pot esborrar o canviar de nom el fitxer.

Un directori: un directori és com un armari ple de caixes (fitxers). Cada caixa dins de l'armari té els seus permisos que determinen si es pot obrir per veure què conté i si se'n pot modificar el contingut. Els permisos de directori (de l'armari) indiquen si es pot saber (mirar

o llistar) quines caixes hi ha i si es poden afegir o eliminar caixes. Així en un directori el permís r-x proporciona el dret d'entrar a l'armari (deu ser gran!) i mirar quines caixes hi ha (no diu res de què podem fer dins de cada caixa perquè això depèn dels permisos de la caixa). El permís w del directori determina si podem modificar-lo o no, si podem tocar les caixes que hi ha dins de l'armari o no. Amb la w podem treure caixes de l'armari, posar-ne de noves i canviar el nom de les caixes. Sense la w (amb r-x) només podem entrar a l'armari i mirar les caixes que hi ha, cada caixa té els seus permisos per indicar si la podem mirar per dins o modificar el seu contingut.

Fàcil oi? Easy peasy...

Estructura de dades d'un directori

A part de l'exemple de caixes i armaris segurament la menera més encertada de pensar amb els directoris i fitxers és amb l'estructura d'un llibre. cada capítol és un fitxer i al principi de tot hi ha un índex amb el llistat dels capítols, el directori.

De fet això és el que significa la paraula *Directory* en anglès, un llistat de noms o d'adreces, una llista amb un nom i una direcció.

Igual que un índex d'un llibre és una llista amb el nom del capítol i la pàgina del llibre on es troba, de manera similar un directori és una llista amb el nom de l'element i una d'adreça de disc.

Estructura d'una entrada de directori:

nom-element i-node

nom-element correspon al nom del fitxer o directori (en el símil amb un índex d'un llibre seria el nom del capítol).

i-node és una adreça de disc on es troba el fitxer (en realitat informació del fitxer). En el símil del llibre seria la pàgina on comença el capítol.

És a dir, un directori és un llistat amb dues columnes, un nom i un número (que és una adreça de disc) igual que un directori telefònic és el nom de cada persona i el seu número de telèfon.

```
$ ls -l
154080 carta.txt
154094 dades.pdf
155473 dossier.odt
180861 newcarta.txt
151318 projecte.odt
152278 treball.txt
```

- podeu observar amb l'ordre *ls -l* el número d'*i-node* de cada element.

Així doncs en un directori:

- per a cada element hi ha una entrada amb el **nom de l'element** i el número d'**i-node** (adreça de disc) amb la resta d'informació i xixa (contingut) de l'element.
- en el directori **NO** hi ha desada la informació dels atributs de l'elementjnt (propietari, grup, enllaços, permisos, mida i data). Aquesta informació es troba en el i-node.
- el **permís r** en un directori proporciona el dret de llistar el directori, és a dir, obtenir els noms i el número d'i-node. Res més.
- el **permís x** en un directori proporciona drets administratius sobre el directori. El dret d'entrar-hi i fer-lo directori actiu amb cd. I també el dret d'accedir a l'inode de cada element.

```
$ ls -ld apunts/
drwxrwxr-- 2 ecanet ecanet 180 18 des 18:50 apunts/

[guest@a36 operatius]$ ls -l apunts/
ls: cannot access 'apunts/newcarta.txt': Permission denied
ls: cannot access 'apunts/dades.pdf': Permission denied
ls: cannot access 'apunts/dossier.odt': Permission denied
ls: cannot access 'apunts/projecte.odt': Permission denied
ls: cannot access 'apunts/informe.pdf': Permission denied
ls: cannot access 'apunts/treball.txt': Permission denied
ls: cannot access 'apunts/carta.txt': Permission denied
total 0
-???????? ? ? ? ?      ? carta.txt
-???????? ? ? ? ?      ? dades.pdf
-???????? ? ? ? ?      ? dossier.odt
-???????? ? ? ? ?      ? informe.pdf
-???????? ? ? ? ?      ? newcarta.txt
-???????? ? ? ? ?      ? projecte.odt
-???????? ? ? ? ?      ? treball.txt

[guest@a36 operatius]$ cd apunts
```

- l'usuari guest amb permisos others r-- només pot llistar els noms dels elements però no té drets (la x) per accedir al i-node, per això no es poden mostrar els atributs de cada entrada.
- només la combinació mínima r-x permet poder llistar els noms i accedir als atributs desats a cada i-node.

i-node

On es desen realment els atributs de cada element? El propietari, grup, permisos, enllaços, mida i data? I on es desa la 'xixa', el contingut de l'element?

Per cada element del sistema de fitxers es genera un **i-node**, una mena de fitxa resum o fitxa tècnica amb informació de l'element. Conté informació descriptiva però no el contingut (les dades o xixa). Si pensem en una biblioteca un i-node seria la fitxa d'un llibre on es descriu el títol, autor, gènere, ISBN, etc però no el propi llibre.

Es pot observar el contingut d'un i-node amb l'ordre **stat**:

```
$ stat apunts/carta.txt
File: apunts/carta.txt
Size: 183      Blocks: 8      IO Block: 4096  regular file
Device: 2ah/42d Inode: 154080 Links: 1
Access: (0400/-r-----)  Uid: ( 1001/  ecanet)  Gid: ( 1001/  ecanet)
Access: 2020-12-18 18:52:58.134493235 +0100
Modify: 2020-12-18 18:41:23.362441239 +0100
```

```
Change: 2020-12-18 18:52:24.977395303 +0100
Birth: -
```

```
$ stat apunts/
File: apunts/
Size: 160          Blocks: 0          IO Block: 4096   directory
Device: 2ah/42d   Inode: 155463    Links: 2
Access: (0774/drwxrwxr--)  Uid: ( 1001/  ecanet)  Gid: ( 1001/  ecanet)
Access: 2020-12-18 20:55:54.043178533 +0100
Modify: 2020-12-18 19:53:18.389441201 +0100
Change: 2020-12-18 20:47:27.915591448 +0100
Birth: -
```

- el primer és el llistat d'informació dels atributs del fitxer carta.txt.
- el segon el llistat d'informació dels atributs del directori apunts
- observeu que internament es desen tres dates diferents, la de l'últim accés [Access](#), la de modificació del contingut [Modify](#) i la de modificació dels atributs (com per exemple els permisos) [Change](#).

A part dels atributs que llista *stat* un i-node conté altra informació com per exemple la llista de [blocks](#) de disc on hi ha emmagatzemades les dades de l'element (la xixa).

El [permís x](#) és el permís necessari per poder accedir a la informació dels i-nodes.

Tot el funcionament en conjunt

Anem a veure què passa en realitat què passa quan per exemple es llista el contingut d'un directori (versió simplificada):

[ls -l](#).

1. el directori actiu conté una llista de noms i i-nodes (imaginem capítols i pàgines).
2. per a cada el nom d'element obté el número d'i-node associat (s'ha obtingut el número de pàgina on hi ha el capítol).
3. s'accedeix a la informació del i-node que conté els atributs permisos, enllaços propietari, grup, mida i data.
4. es mostra aquesta informació (els atributs) i el nom per a cada element.

Anem a veure què passa en realitat què passa quan per exemple es mostra el contingut d'un fitxer:

[cat carta.txt](#).

1. primerament es mira en el directori actiu si hi ha un fitxer anomenat carta. per poder-ho fer cal almenys els permisos r-x sobre el directori actiu.
2. el directori actiu conté una llista de noms i i-nodes (imaginem capítols i pàgines).
3. si localitza el nom carta.txt obté el número d'i-node associat (s'ha obtingut el número de pàgina on hi ha el capítol).

4. s'accedeix a la informació del i-node de carta per obtenir el propietari, el grup i els permisos.
5. es contrasten amb els de l'usuari que vol mostrar el contingut de carta.
6. si l'usuari no té permisos suficients per mostrar-ne el contingut es finalitza amb un missatge d'error.
7. si l'usuari té dret a mostrar el contingut de carta.txt es mira al i-node a quins blocs de disc hi a les dades (la xixa) de carta.txt, s'hi accedeix i es mostra.

Anem a veure què passa en realitat què passa quan per exemple es llista el contingut d'un directori (versió completa):

`'ls -l'`.

1. el directori actiu conté una llista de noms i i-nodes, es consulta l'entrada corresponent al directori actiu, l'entrada punt ".".
2. un cop localitzada l'entrada "." s'obté el seu número d'i-node.
3. s'accedeix al inode i s'obtenen els atributs propietari, grup i permisos.
4. es contrasta aquests atributs amb els de l'usuari que vol llistar el directori.
5. si l'usuari no té suficients permisos finalitza el llistat indicant un error de falta de permisos.
6. si l'usuari té suficients permisos per fer el llistat procedeix a fer-lo:
 - a. el directori actiu conté una llista de noms i i-nodes (imaginem capítols i pàgines).
 - b. per a cada el nom d'element obté el número d'i-node associat (s'ha obtingut el número de pàgina on hi ha el capítol).
 - c. s'accedeix a la informació del i-node que conté els atributs permisos, enllaços propietari, grup, mida i data.
 - d. es mostra aquesta informació (els atributs) i el nom per a cada element.

En resum:

- ☐ Un directori és una llista de noms i números d'i-node.
- ☐ Donat un nom d'element s'accedeix al i-node per obtenir-ne els atributs.
- ☐ Donat l'inode s'obté la llista de blocs de disc on està desada realment la xixa, les dades, de l'element.

Diferència entre els permisos de directori i de fitxer

Sovint un dels problemes en la comprensió dels permisos és barrejar a què afecten els permisos de directori i els permisos de fitxer.

Tornem-hi:

- els de fitxer afecten a la caixa, al capítol, al 'contingut' del fitxer. tenir r vol dir que es pot veure què conté. Tenir w vol dir que es pot modificar el contingut del fitxer.
- en CAP cas, **CAP**, la w del fitxer significa res sobre si es pot esborrar o canviar de nom el fitxer. No, això depèn dels permisos del directori on es troba.

- en un directori tenir w significa que es poden modificar les entrades del directori. Si l'entendem com un índex d'un llibre vol dir que podem afegir nous capítols (afegir fitxers), esborrar capítols (eliminar fitxers) i canviar de nom capítols (renombrar fitxers).
- en CAP cas, **CAP**, el permís w en el directori indica res de si es pot o no modificar el contingut, l'interior, d'un fitxer.

Els errors més comuns en el tractament de permisos són:

- **ERROR**: creure que si un fitxer no té permís w no es pot esborrar. Insistim, la w del fitxer només té a veure amb el seu contingut, no en si es pot esborrar el fitxer del directori o no.
- **ERROR**: creure que perquè un fitxer no té permís w no es pot esborrar. Altre cop indicar que qui marca que es pot fer o no en un directori són els permisos del directori i no els individuals del fitxer.

Observem el següent exemple:

```
[user01@a36 apunts]$ ls -la
drwxr-xr-x 2 ecanet ecanet 160 Dec 18 19:53 .
drwxrwxr-x 4 ecanet ecanet 80 Dec 18 18:21 ..
-r----- 1 ecanet ecanet 183 Dec 18 18:41 carta.txt
-rw-rw-r-- 1 ecanet ecanet 28 Dec 18 18:21 dades.pdf
-rw-rw-r-- 1 ecanet ecanet 21448 Dec 18 18:21 dossier.odt
-rw-r----- 1 user01 ecanet 134 Dec 18 19:03 newcarta.txt
-rw-rw-r-- 1 ecanet ecanet 134 Dec 18 18:21 projecte.odt
-rw-rw-r-- 1 ecanet ecanet 1159 Dec 18 18:21 treball.txt
```

- saber si un usuari pot crear fitxers nous, esborrar-ne o canviar el nom dels que hi ha dins d'aquest directori apunts NO té res a veure amb els permisos individuals dels fitxers.
- té a veure amb els permisos del directori, llistat a la primera entrada punt "." que són rwxr-xr-x.
- així doncs l'usuari user01 no pot afegir, esborrar ni canviar de nom cap dels elements del directori, perquè se li apliquen els permisos de grup r-x i no té la w.
- no pot ni tan sols esborrar ni canviar de nom el fitxer newcarta.txt que és seu!.
- la w en el fitxer newcarta.txt (n'és el propietari i per tant té permisos rw) indica que en pot modificar el contingut. Si vol pot esborrar totes les línies de dins el fitxer, però el fitxer NO el pot eliminar.
- perquè no? perquè això voldria dir modificar la llista del directori i per fer això li cal el permís w del directori.
- és el mateix motiu pel qual no pot tampoc canviar-li el nom.

```
[user01@a36 apunts]$ rm newcarta.txt
rm: cannot remove 'newcarta.txt': Permission denied

[user01@a36 apunts]$ mv newcarta.txt newname.txt
mv: cannot move 'newcarta.txt' to 'newname.txt': Permission denied
```

Recordeu: els permisos de fitxers no afecten en res, en res, en res a l'hora de decidir si es pot afegir, eliminar o canviar de noms elements en un directori, només afecta els permisos del directori.

Exercicis d'exemple

Primerament heu de fer l'exercici **00-Creació_de_estructura** per generar els directoris i els fitxers necessaris per fer aquest exercici.

1. Fer actiu el directori `/tmp/mp1/operatiu/apunts`.

Des d'aquest directori realitzarem tots els exercicis, tots. No és permès de canviar de directori amb l'ordre `cd`. Totes les ordres del sistema es poden realitzar des de qualsevol directori actiu.

Després de fer aquest directori actiu verificar-ho amb l'ordre `pwd`.

```
$ cd /tmp/m01/operatiu/apunts
```

```
$ pwd
/tmp/m01/operatiu/apunts
```

Recordeu quina és l'estructura de directoris i fitxers que estem utilitzant:

```
$ tree /tmp/m01
/tmp/m01
├── operatiu
│   ├── apunts
│   │   ├── carta.txt
│   │   ├── dades.pdf
│   │   ├── dossier.odt
│   │   ├── informe.pdf
│   │   ├── projecte.odt
│   │   └── treball.txt
│   └── exercicis
└── xarxes
```

```
$ tree /var/tmp/prova/
/var/tmp/prova/
```

2. Llistar el directori actiu i **observar els permisos**

```
$ ls -l
-rw-rw-r-- 1 ecanet ecanet 113 18 des 18:21 carta.txt
-rw-rw-r-- 1 ecanet ecanet 28 18 des 18:21 dades.pdf
-rw-rw-r-- 1 ecanet ecanet 21448 18 des 18:21 dossier.odt
-rw-rw-r-- 1 ecanet ecanet 89 18 des 18:21 informe.pdf
-rw-rw-r-- 1 ecanet ecanet 134 18 des 18:21 projecte.odt
-rw-rw-r-- 1 ecanet ecanet 1159 18 des 18:21 treball.txt
```

3. Com a **root** crear l'usuari **user01** que pertany al mateix grup que el vostre usuari i validar que existeix l'usuari **guest** i que pertany a un altre grup.

```
[ecanet@a36 m01]$ su -
Password:
```

```
[root@a36 ~]# id ecanet
uid=1001(ecanet) gid=1001(ecanet) groups=1001(ecanet),10(wheel),975(docker),971(vagrant)
```

```
[root@a36 ~]# useradd -g ecanet user01
[root@a36 ~]# passwd user01
```



```
Changing password for user user01.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.
```

```
[root@a36 ~]# id user01  
uid=1029(user01) gid=1001(ecanet) groups=1001(ecanet)
```

```
[root@a36 ~]# id guest  
uid=1000(guest) gid=1000(guest) groups=1000(guest)
```

- primerament inicia sessió com a administrador del sistema amb l'usuari **root** amb l'ordre **su -**.
- mostrar la informació de l'usuari **ecanet**, en especial observar que pertany al grup **1001(ecanet)**.
- crear l'usuari **user01** assignant-li el grup **ecanet** (l'opció **-g**) i assignar-li un password.
- observar que l'usuari pertany al grup **ecanet** igual que l'usuari **ecanet**.
- observar que l'usuari **guest** pertany a un altre grup, al grup **1000(guest)**.

4. Mostrar la informació amb l'ordre **stat** del directori **operatiu**.

```
$ stat /tmp/m01/operatiu/  
File: /tmp/m01/operatiu/  
Size: 80      Blocks: 0      IO Block: 4096  directory  
Device: 2ah/42d Inode: 155462 Links: 4  
Access: (0775/drwxrwxr-x) Uid: ( 1001/ ecanet) Gid: ( 1001/ ecanet)  
Access: 2020-12-18 21:45:35.715536083 +0100  
Modify: 2020-12-18 18:21:06.965848615 +0100  
Change: 2020-12-18 18:21:06.965848615 +0100  
Birth: -
```

5. Mostrar la informació amb l'ordre **stat** del fitxer **dades.pdf**.

```
$ stat dades.pdf  
File: dades.pdf  
Size: 28      Blocks: 8      IO Block: 4096  regular file  
Device: 2ah/42d Inode: 154094 Links: 1  
Access: (0664/-rw-rw-r--) Uid: ( 1001/ ecanet) Gid: ( 1001/ ecanet)  
Access: 2020-12-18 18:21:09.128855004 +0100  
Modify: 2020-12-18 18:21:09.129855007 +0100  
Change: 2020-12-18 18:21:09.129855007 +0100  
Birth: -
```

6. Assignar al directori **apunts** els permisos **775**.

```
$ pwd  
/tmp/m01/operatiu  
  
$ chmod 775 apunts/  
  
$ ls -ld apunts/  
drwxrwxr-x 2 ecanet ecanet 160 18 des 19:53 apunts/
```

7. Pot l'usuari **user01** crear dins del directori **apunts** un fitxer anomenat **user01file.pdf**, canviar-li el nom per **user01file.odt** i finalment **esborrar-lo**?

```
[user01@a36 apunts]$ echo "sóc user01 i creo un fitxer" >> user01file.pdf
[user01@a36 apunts]$ ls -l user01file.pdf
-rw-r--r-- 1 user01 ecanet 29 Dec 19 18:20 user01file.pdf

[user01@a36 apunts]$ mv user01file.pdf user01file.odt
[user01@a36 apunts]$ ls -l user01file.odt
-rw-r--r-- 1 user01 ecanet 29 Dec 19 18:20 user01file.odt

[user01@a36 apunts]$ rm user01file.odt
[user01@a36 apunts]$ ls -la
drwxrwxr-x 2 ecanet ecanet 160 Dec 19 18:21 .
drwxrwxr-x 4 ecanet ecanet 80 Dec 18 18:21 ..
-r----- 1 ecanet ecanet 183 Dec 18 18:41 carta.txt
-rw-rw-r-- 1 ecanet ecanet 28 Dec 18 18:21 dades.pdf
-rw-rw-r-- 1 ecanet ecanet 21448 Dec 18 18:21 dossier.odt
-rw-r----- 1 user01 ecanet 134 Dec 18 19:03 newcarta.txt
-rw-rw-r-- 1 ecanet ecanet 134 Dec 18 18:21 projecte.odt
-rw-rw-r-- 1 ecanet ecanet 1159 Dec 18 18:21 treball.txt
```

- si pot fer les tres accions perquè l'usuari user01 pertany al mateix grup que el directori apunts i se li apliquen els permisos rwx. Per tant té dret a crear, eliminar i modificar les entrades de directori.

8. Amb el vostre usuari (ecanet en l'exemple) canviar els permisos del fitxer [treball.txt](#) posant únicament permís de lectura per el propietari i tots els altres permisos tancats. Ara que el fitxer està sense permisos per a grup i altres pot l'usuari [user01](#) esborrar-lo?

```
$ chmod 600 apunts/treball.txt
$ ls -l apunts/treball.txt
-rw----- 1 ecanet ecanet 1159 18 des 18:21 apunts/treball.txt

[user01@a36 apunts]$ rm treball.txt
rm: remove write-protected regular file 'treball.txt'? y
```

- Si pot, la pregunta té la trampa de intentar confondre entre els permisos de fitxer i de directori. Que el fitxer tingui permisos (600 r-- --- ---) no afecta en res (en res!) a si l'usuari user01 el pot eliminar o no.
- L'usuari user01 pot eliminar el fitxer perquè els permisos de directori són 775 i pertany al mateix grup que el directori. Per tant se li apliquen els permisos rwx i pot modificar el contingut del directori.

9. Establir al fitxer [carta.txt](#) permisos perquè només el propietari pugui veure'n el contingut i tots els altres permisos eliminats (feu-ho amb el vostre usuari). Pot l'usuari [user01](#) modificar el contingut del fitxer [carta.txt](#)?

```
$ chmod 400 apunts/carta.txt
$ ls -l apunts/carta.txt
-r----- 1 ecanet ecanet 183 18 des 18:41 apunts/carta.txt

[user01@a36 apunts]$ echo "sóc user01 i vull modificar carta" >> carta.txt
-bash: carta.txt: Permission denied
```

- L'usuari no pot modificar el contingut de carta.txt perquè no té el permís w del fitxer. De fet no pot ni veure el seu contingut.

10. Què passa si en lloc d'usar un redireccionament d'afegir l'usuari `user01` 'intenta modificar `carta.txt` amb l'ordre `vim`?

- primerament el vim s'obre però no es mostra el contingut de `carta.txt` perquè l'usuari `user01` no té permisos (---).
- l'usuari `user01` pot escriure contingut però en prémer w per desar el contingut del vim se l'avisava de que no pot desar el contingut perquè el fitxer és **read only**.
- si tot i això l'usuari `user01` insisteix en desar el contingut amb l'opció **w!** de vim, el fitxer es desa sobreescrivint `carta.txt`.
- Perquè? Què ha passat?
 - L'usuari `user01` no pot ni veure ni modificar el contingut de `carta.txt` (permisos del fitxer) però sí que el pot destruir i generar de nou (sobreescriure'l perquè els permisos de directori `rwx` li permeten).

```
[user01@a36 apunts]$ vim carta.txt
[user01@a36 apunts]$ cat carta.txt

[user01@a36 apunts]$ ll
-rw----- 1 user01 ecanet  0 Dec 19 18:35 carta.txt
-rw-rw-r-- 1 ecanet ecanet 28 Dec 18 18:21 dades.pdf
-rw-rw-r-- 1 ecanet ecanet 21448 Dec 18 18:21 dossier.odt
-rw-r----- 1 user01 ecanet 134 Dec 18 19:03 newcarta.txt
-rw-rw-r-- 1 ecanet ecanet 134 Dec 18 18:21 projecte.odt
```

- observeu que ara el fitxer és propietat de l'usuari `user01` i té els permisos `rw-----`.
- observeu també que el fitxer queda destruït, és a dir, queda amb contingut zero.

11. Amb el vostre usuari assigneu al fitxer `projecte.odt` permisos totals al propietari i cap per a grup i altres. Pot l'usuari `user01` sobreescriure el fitxer `projecte.odt`, generar un nou fitxer `projecte.odt` seu en lloc de ser propietat del vostre usuari?

```
$ chmod 600 apunts/projecte.odt
$ ls -l apunts/projecte.odt
-rw----- 1 ecanet ecanet 134 19 des 18:50 apunts/projecte.odt
```

```
[user01@a36 apunts]$ date > projecte.odt
-bash: projecte.odt: Permission denied
```

```
[user01@a36 apunts]$ date >> projecte.odt
-bash: projecte.odt: Permission denied
```

```
[user01@a36 apunts]$ rm projecte.odt
[user01@a36 apunts]$ date >> projecte.odt

[user01@a36 apunts]$ ls -l projecte.odt
-rw-r--r-- 1 user01 ecanet 29 Dec 19 18:56 projecte.odt
```

- en intentar sobreescriure el fitxer el sistema operatiu no el deixa, ni afegir ni sobreescriure.
- però com que l'usuari `user01` té permisos `rwx` en el directori sí que pot eliminar el fitxer i generar-lo de nou.

- ara el fitxer ja no és propietat del vostre usuari (ecanet en l'exemple) sinó de l'usuari user01. Els permisos que té també han canviat, s'han establert els permisos per defecte de l'usuari.

12. Pot l'usuari user01 crear dins del directori apunts un directori anomenat processos? De qui és el directori i amb quins permisos?

```
[user01@a36 apunts]$ mkdir processos
[user01@a36 apunts]$ ls -ld processos/
drwxr-xr-x 2 user01 ecanet 40 Dec 19 19:00 processos/
```

- Pot crear el directori perfectament perquè té permisos rwx en el directori apunts.
- El directori és propietat seva i del grup al que pertany. Els permisos assignats són els que s'apliquen per defecte a aquest usuari (poden ser diferents en cada cas).

13. Pot l'usuari *guest* llistar el contingut del directori *apunts*?

```
[guest@a36 apunts]$ ls -la
total 36
drwxrwxr-x 3 ecanet ecanet 160 Dec 19 19:00 .
drwxrwxr-x 4 ecanet ecanet 80 Dec 18 18:21 ..
-rw----- 1 user01 ecanet 0 Dec 19 18:35 carta.txt
-rw-rw-r-- 1 ecanet ecanet 28 Dec 18 18:21 dades.pdf
-rw-rw-r-- 1 ecanet ecanet 21448 Dec 18 18:21 dossier.odt
-rw-r----- 1 user01 ecanet 134 Dec 18 19:03 newcarta.txt
drwxr-xr-x 2 user01 ecanet 40 Dec 19 19:00 processos
-rw-r--r-- 1 user01 ecanet 29 Dec 19 18:56 projecte.odt
```

- si pot perquè el directori apunts té permisos r-x per a altres.

14. Pot l'usuari *guest* veure el contingut del fitxer *dades.pdf*?

Pot l'usuari *guest* modificar el contingut del fitxer *dades.pdf*?

```
[guest@a36 apunts]$ cat dades.pdf
dv des 18 18:21:09 CET 2020
```

```
[guest@a36 apunts]$ echo "guest intenta modificar dades.pdf" >> dades.pdf
-bash: dades.pdf: Permission denied
```

- per determinar si *guest* pot o no veure el contingut de *dades.pdf* cal mirar els permisos de altres del fitxer, en aquest cas són r-- i per tant *guest* en pot llistar el contingut.
- com que els permisos del fitxer a altres no inclouen la w *guest* no pot modificar el contingut de *dades.pdf*.

15. Amb el vostre usuari modificar els permisos de *dades.pdf* establint-los a *666*. Pot ara sí l'usuari *guest* modificar-ne el contingut?

```
$ chmod 666 apunts/dades.pdf
```

```
$ ls -l apunts/dades.pdf
-rw-rw-rw- 1 ecanet ecanet 28 18 des 18:21 apunts/dades.pdf
```

```
[guest@a36 apunts]$ echo "guest intenda modificar dades.pdf" >> dades.pdf
[guest@a36 apunts]$ ls -l dades
-rw-rw-rw- 1 ecanet ecanet 62 Dec 19 19:11 dades.pdf
```

- si pot perquè amb el permís rw- a others en el fitxer l'usuari guest pot modificar-ne el contingut.

16. Ara que hem vist que en pot modificar el contingut pot l'usuari *guest esborrar* el fitxer *dades.pdf*?

I canviar-li el nom?

I crear un nou fitxer anomenat dades-guest.odt?

```
[guest@a36 apunts]$ rm dades.pdf
rm: cannot remove 'dades.pdf': Permission denied

[guest@a36 apunts]$ mv dades.pdf dades.odt
mv: cannot move 'dades.pdf' to 'dades.odt': Permission denied

[guest@a36 apunts]$ cat > dades-guest.odt
-bash: dades-guest.odt: Permission denied
```

- l'usuari guest no pot ni esborrar ni canviar el nom al fitxer dades.pdf perquè per fer-ho es necessiten permisos al directori apunts i no els té. Al directori apunts els permisos de others són r-x i per tant a guest li falta el permís w per modificar el directori.
- tampoc pot crear cap nou fitxer en el directori per el mateix motiu, no té permís per modificar el directori.

17. Amb l'usuari *user01* crear dins del directori *processos* un fitxer anomenat *cron.txt* amb permís de lectura per tothom. Crear també al mateix lloc un fitxer anomenat *at.txt* amb permís de lectura només per al propietari.

Els permisos del directori *processos* són 755.

Pot l'usuari *guest* fer actiu el directori *processos* i mostrar el contingut dels fitxers *cron.txt* i *at.txt*?

```
[user01@a36 apunts]$ cat > processos/cron.txt

[user01@a36 apunts]$ date > processos/at.txt

[user01@a36 apunts]$ chmod 444 processos/cron.txt

[user01@a36 apunts]$ chmod 400 processos/at.txt

[user01@a36 apunts]$ ls -la processos/
drwxr-xr-x 2 user01 ecanet 80 Dec 19 19:26 .
drwxrwxr-x 3 ecanet ecanet 160 Dec 19 19:00 ..
-r----- 1 user01 ecanet 29 Dec 19 19:26 at.txt
-r--r--r-- 1 user01 ecanet 174 Dec 19 19:25 cron.txt
```

```
[guest@a36 apunts]$ cd processos/

[guest@a36 processos]$ pwd
/tmp/m01/operatiu/apunts/processos
```

```
[guest@a36 processos]$ cat cron.txt
```

```
December 2020  
Su Mo Tu We Th Fr Sa  
1 2 3 4 5  
6 7 8 9 10 11 12  
13 14 15 16 17 18 19  
20 21 22 23 24 25 26  
27 28 29 30 31
```

```
[guest@a36 processos]$ cat at.txt
```

```
cat: at.txt: Permission denied
```

- l'usuari guest si pot entrar el directori processos perquè se li apliquen els permisos de other que són r-x. També en pot llistar el contingut.
- que pugui o no veure el contingut del fitxer cron.txt depèn exclusivament del permís r de others del fitxer cron.txt (sempre que hagi pogut accedir al directori). En aquest cas com que els permisos de other de cron.txt són r-- l'usuari guest en pot mostrar el contingut.
- del fitxer at.txt no en pot mostrar el contingut perquè els permisos de others són ---.

18. Amb l'usuari *user01* assignar al directori processos els permisos *750*.

Pot ara l'usuari *guest* entrar al directori (si cal sortiu i torneu a entrar)?

Pot llistar el contingut de *cron.txt* i *at.txt*?

```
[user01@a36 apunts]$ chmod 750 processos/
```

```
[user01@a36 apunts]$ ls -ld processos/
```

```
drwxr-x--- 2 user01 ecanet 80 Dec 19 19:26 processos/
```

```
[guest@a36 processos]$ ls
```

```
ls: cannot open directory '.': Permission denied
```

```
[guest@a36 processos]$ cd ..
```

```
[guest@a36 apunts]$ cd processos/
```

```
-bash: cd: processos/: Permission denied
```

```
[guest@a36 apunts]$ cat processos/cron.txt
```

```
cat: processos/cron.txt: Permission denied
```

```
[guest@a36 apunts]$ cat processos/at.txt
```

```
cat: processos/at.txt: Permission denied
```

- primerament l'usuari guest ja estava al directori processos i en intentar fer un ls no pot.
- surt del directori i intenta tornar-hi entrar però no pot. Els permisos del directori per a others són --- i per tant no té cap permís.
- tot i que el fitxer cron.txt té el permís r per a other l'usuari guest no el pot llistar perquè ni tan sols pot accedir a l'interior del directori.
- el fitxer at.txt tampoc el pot llistar, primerament perquè no té accés al directori, però si en tingués tampoc podria perquè no té permís per al fitxer.

19. Pot el vostre usuari (ecanet en l'exemple) esborrar el directori processos i tot el què conté?

```
$ rm -rf apunts/processos/
```

```
rm: cannot remove 'apunts/processos/at.txt': Permission denied
rm: cannot remove 'apunts/processos/cron.txt': Permission denied
```

- el vostre usuari no pot esborrar el directori processos i tot el què conté perquè dins del directori hi ha elements que no té permís per esborrar.
- el directori és propietat de user01 i té permisos 755. El vostre usuari és del mateix grup que l'usuari user01 i per tant se li apliquen els permisos de grup (5 o r-x). Això vol dir que no pot eliminar contingut.
- **punyeta!** el vostre usuari ha deixat el directori apunts obert (775) i un altre usuari ho ha aprofitat per posar-hi elements dins i ara el vostre usuari ja no els pot eliminar.

20. elimineu tot l'estructura de directoris de pràctiques abans de fer el següent conjunt d'exercicis.