

03 - Mostrar i crear fitxers: cat

Curs 2020 - 2021

ASIX M01-ISO UF1-A01-02 ordres de fitxers

Mostrar i crear fitxers: cat	1
Descripció	1
L'ordre cat	1
Creació de fitxers	2
Exercicis d'exemple	4

Mostrar i crear fitxers: cat

Descripció

L'ordre cat

L'ordre cat permet mostrar el contingut d'un fitxer de text. Veure què conté. Si intentem mostrar el contingut de fitxers que no siguin de text la consola d'omplirà de caràcters estranys.

***Atenció*:** en cap cas confonguem les expressions llistar el contingut d'un directori (veure la llista d'elements que hi ha, que conté un directori), amb l'expressió mostrar el contingut d'un fitxer (veure les dades, la xixa, que conté un fitxer).

- ❑ **Llistar:** llistar o mostrar el contingut d'un directori es fa amb l'ordre *ls*.
- ❑ **Mostrar el contingut d'un fitxer:** es fa amb l'ordre *cat*.

L'ordre conceptualment té el format bàsic:

cat file[...]

file és el fitxer de text del qual volem poder mostrar el contingut. Es poden indicar múltiples fitxers separats per espai. Cada element ha de contenir la seva ruta (absoluta o relativa).

Exemple de llistat simple:

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
```

- Mostra el contingut del fitxer dels comptes d'usuari, cada línia descriu un usuari del sistema.

Creació de fitxers

Hi ha diversos mecanismes per crear fitxers, els més lògics són usar editors de text com el vim, el gedit, emacs, nano, etc. Però també hi ha 'trucs' per fer-los més ràpid o si no disposem d'editors.

Quan volem crear fitxers 'tontos' sense que ens importi el contingut, crear-los per crear-los podem usar també alguns 'trucs'. Aquests mecanismes són molt útils per fer proves, exemples i exercicis de tractament de fitxers perquè en lloc d'utilitzar fitxers reals del sistema ens podem crear fitxers amb contingut intrascendent i practicar copiant-los, movent-los i esborrant-los.

Amb l'ordre cat:

cat > [ruta]file

Posa dins de *file* tot allò que s'escriu per el teclat fins a prémer la combinació de tecles *control+d* (les dues al mateix temps). File pot ser només el nom del fitxer a crear o la ruta (relativa o absoluta) i el nom del fitxer (tot junt).

Exemple de creació d'un fitxer intrascendent amb cat > file:

```
$ cat > /tmp/file1
ara escrivim pel teclat
prem enter per canviar de línia
escrivim rant com volem
prem ^d (control d) per finalitzar
per cert, un cop entrada una línia ja no es pot rectificar
no podem editar cap amunt,
bye!

$ cat /tmp/file1
ara escrivim pel teclat
prem enter per canviar de línia
escrivim rant com volem
prem ^d (control d) per finalitzar
per cert, un cop entrada una línia ja no es pot rectificar
no podem editar cap amunt,
bye!
```

- ☐ La primera comanda permet crear el fitxer, en escriure-la el prompt es queda esperant allò que escrivim.
- ☐ Cada línia la finalitzem amb enter.
- ☐ Un cop apretat enter ja no es poden editar les línies anteriors.
- ☐ Per acabar l'edició es prem control+d que en general s'escriu ^d.

- ❑ El significat és que li diem que mostri però no li diem el què i que ho posi en un fitxer, com que no li hem dit que ha de mostrar es queda esperant a l'entrada de dades pel teclat, que finalitza en prémer ^d.

Amb qualsevol ordre i el redireccionament de sortida:

ordre > [ruta]file

Posa a file el contingut que es mostraria per pantalla en executar l'ordre indicada. Quan s'executa una ordre aquesta (normalment) genera informació (sortida) que va a parar a la pantalla. Amb el caràcter de redireccionament > se li indica que en lloc de a la pantalla la sortida ha d'anar a dins del fitxer.

Exemple de creació de fitxer usant una ordre i el redireccionament de sortida:

```
$ id > identity.txt  
  
$ cat identity.txt  
uid=1001(ecanet) gid=1001(ecanet) groups=1001(ecanet),10(wheel),971(vagrant),975(docker)
```

```
$ uname -a > /var/tmp/sistema.txt  
  
$ cat /var/tmp/sistema.txt  
Linux a36.informatica.escoladeltreball.org 4.18.19-100.fc27.x86_64 #1 SMP Wed Nov 14 22:04:34 UTC 2018  
x86_64 x86_64 x86_64 GNU/Linux
```

- En el primer exemple s'executa la ordre *id* que mostra per pantalla la informació de identificació de l'usuari. Aquesta informació en lloc d'anar a la pantalla es redirecciona a un fitxer del directori actiu anomenat identity.txt que tot seguit es mostra amb cat.
- El segon exemple executa l'ordre *uname* que mostra el tipus de sistema operatiu que s'està utilitzant. Aquesta informació en lloc d'anar a la pantalla es desa en un fitxer de nom sistema.txt dins de /var/tmp.
- El resultat és que s'han creat dos fitxers un anomenat identity.txt i un altre anomenat sistema.txt.

Crear fitxers amb l'ordre echo

echo "missatge" > [ruta]file

Aquesta és una variant de l'ordre anterior per posar contingut concret dins d'un fitxer. Amb l'ordre *echo* es genera un missatge (allò que vulguem) que via el redireccionament de sortida es desa dins del fitxer indicat.

```
$ echo "aquest és el contingut del fitxer" > /tmp/hola.txt  
  
$ cat /tmp/hola.txt  
aquest és el contingut del fitxer
```

Crear fitxers amb l'editor vim

vim [ruta]fitxer

Amb l'editor vim (com amb qualsevol altre editor) podem editar fitxers, modificar-los i crear-ne de nous.

```
$ vim /var/tmp/exemple.txt
```

- ❑ Dins de l'editor vim hi ha el mode *insert* que és quan podem escriure contingut dins del text. Per entrar a mode insert cal prémer la *lletra i*, observar que a la part inferior esquerra indica *insert*.
- ❑ Un altre mode és el mode *comanda*. S'hi accedeix prement la tecla *escape*, en fer-ho es deixa d'estar en mode edició i es poden fer comandes.
- ❑ Per desar el fitxer cal prémer *escape : w* (escape, la lletra dos punts (:) i escriure una w i enter). Això significa entrar en mode comanda i executar l'opció *write*.
- ❑ Per sortir del programa cal anar a mode comanda i fer l'opció *quit*, és a dir, prémer *escape : q*.
- ❑ Si s'ha modificat el contingut però es vol abortar l'edició es pot *sortir sense desar* amb *escape : q!*, que significa forçar (!) el quit.
- ❑ En mode comanda es poden *esborrar línies* premement les *lletres dd*, que esborren la línia actual. Si es prem per exemple 3dd s'esborren tres línies.
- ❑ La *lletra u* de *undo* anula la última acció i la lletra *U* de *redo* la torna a realitzar.

nota l'editor vim és una eina fundamental en l'administració de sistemes informàtics i la utilitzarem constantment. És per això que es recomana que en sigueu hàbils. Trobareu documentació, tutorials, vídeos i jocs pel vim a internet.

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*.

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/
```

```
$ pwd
/tmp/m01
```

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 fitxer dels comptes d'usuari del sistema (en sortiran molts però ja és correcte).

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
```

- Mostra el contingut del fitxer dels comptes d'usuari, cada línia descriu un usuari del sistema.

Podem limitar el llistat als deu primer usant el filtre **head** (ordre avançada).

```
$ cat /etc/passwd | head
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/bin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

També podem comptar quants n'hi ha amb el filtre **word count** (wc) (ordre avançada).

```
$ cat /etc/passwd | wc -l
70
```

- S'ha utilitzat l'opció -l del word count per comptar les línies.

3. Mostrar el contingut del fitxer dels grups del sistema.

```
$ cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
...
```

Podem per exemple paginar (pàgina a pàgina) el llistat anterior usant el filtre *less* (ordre avançada).

```
$ cat /etc/passwd | less
```

- ☐ Per sortir del llistat prémer la *lletra q* de *quit*.
- ☐ Per avançar pàgina a pàgina prémer *l'espai* o les tecles *pgup* i *pgdwn*.
- ☐ Per anar al principi o final prémer la *lletra g* o la *lletra G*. O bé prémer les tecles *inici* o *final*.

4. Què fa la següent ordre?

```
$ cat /usr/bin/date
```

- Mostra el contingut del fitxer date, però aquest és un fitxer *executable binari*. Per tant omple la pantalla de caràcters estranys!.
- Podem usar l'ordre cat per mostrar qualsevol contingut però no té sentit fer-ho si el contingut no és *text ascii* (text pur).

Podem verificar si un fitxer és o no un fitxer de text amb l'ordre *file* (ordre avançada).

```
$ file /usr/bin/date
/usr/bin/date: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, interpreter
/lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=086e8b634e66a2b2529ffaf7a802ea375c0d2dc2,
stripped

$ file /etc/passwd
/etc/passwd: ASCII text
```

- L'ordre file informa que /usr/bin/date és un binari executable en format ELF-64-bit.

5. Mostrar el contingut del fitxer *carta* del directori *apunts*.

```
$ cat operatiu/apunts/carta.txt
```

- Usem una ruta relativa perquè des del directori actiu /tmp/m01 fins a carta hi ha un camí descendent.

6. Mostrar el contingut dels fitxers d'extensió *txt* del directori *apunts*.

```
$ cat operatiu/apunts/*.txt
```

- Utilitzem una plantilla de nom de fitxer, un Pathname Expansion amb el metacaràcter *** que significa que volem veure el contingut dels fitxers de qualsevol nom amb extensió *txt* de dins del directori *apunts* que està dins del directori *operatiu* que està dins del directori actiu.

7. Mostrar els fitxers *treball.txt* *projecte.odt* i tots els *pdf* del directori *apunts*.

```
$ cat operatiu/apunts/treball.txt operatiu/apunts/projecte.odt operatiu/apunts/*.pdf
```

- En aquest exemple l'ordre *cat* rep tres arguments.
- A cada argument cal construir correctament la ruta.
- Totes tres són rutes relatives del directori actual fins a dins del directori *apunts*.
- Encara que ttes tres són la mateixa cal escriure-les en cada cas ja que els arguments són independents els uns dels altres.
- L'últim argument conté el metacaràcter *** que indica qualsevol nom de fitxer amb extensió *pdf*.
- ***nota*** aquests fitxers d'exemple són de mentida, no són realment *pdf* ni *odt* sinó que són de text *ascii* (hem fet trampa en crear-los!). És per això que els podem mostrar amb *cat*.

8. Crear un fitxer dins de */tmp* anomenat *text.txt* amb el contingut que vulguis escriure interactivament per teclat usant un *cat* >.

```
$ cat > /tmp/text.txt
aquest és un text d'exemple
s'escriu línia a línia
a cada línia es prem return
i per acabar
s'apreta ^d
```

```
$ cat /tmp/text.txt
aquest és un text d'exemple
s'escriu línia a línia
a cada línia es prem return
i per acabar
s'apreta ^d
```

9. Crear un fitxer al *directori actual* anomenat *engegat.txt* amb la informació que genera l'execució de l'ordre *uptime*.

```
$ uptime > engegat.txt
```

```
$ cat engegat.txt
17:17:02 up 7:52, 1 user, load average: 1,22, 1,03, 0,74
```

10. Crear un fitxer al directori */var/tmp* anomenat *processos.txt* amb el resultat de l'execució de la ordre *ps*.

```
$ ps > /var/tmp/processos.txt  
  
$ cat /var/tmp/processos.txt  
PID TTY      TIME CMD  
8236 pts/0    00:00:00 bash  
12232 pts/0    00:00:00 ps
```

11. Generar un fitxer anomenat *benvinguda.txt* al directori *xarxes* amb el resultat de fer una ordre *echo* amb un missatge de benvinguda.

```
$ echo "Benvinguts a l'escola" > xarxes/benvinguda.txt  
  
$ cat xarxes/benvinguda.txt  
Benvinguts a l'escola
```

12. Usant l'editor *vim* edita un fitxer anomenat *fitxa* en el *directori actiu* i omple'l amb el nom, cognom, dni, edat i adreça, cada concepte en una línia.

```
$ vim fitxa  
  
$ cat fitxa  
pere pou prat  
99911555x  
urgel 136
```