

# Manipulació de fitxers

Albert Ruiz

Departament de Matemàtiques

28 de febrer de 2008



**Universitat Autònoma  
de Barcelona**

# Continguts

- 1 Veure fitxers
  - more i less
- 2 Algunes comandes per a manipular fitxers
  - Convertir latin1-utf8
  - Comprimir fitxers
  - a2ps, pdf2ps, ps2pdf
- 3 Canvis sortida i entrada (pipes)
  - Sortida
  - Entrada
  - Més pipes
- 4 Expressions regulars
  - grep
- 5 Ordenar el contingut d'un fitxer

# cat, tac, nl

## cat, tac, nl

`cat` fitxer: ens ensenya el fitxer per pantalla, tot de cop.

`tac` fitxer: ens ensenya el fitxer, tot de cop, però començant per la última línia i acabant a la primera.

`nl` fitxer: ens ensenya el fitxer amb les línies numerades.

## Exemples

Agafem el fitxer `linux_u8.txt` de `mat.uab.cat/~albert/curs1`:

```
$ wget http://mat.uab.cat/~albert/curs1/linux_u8.txt
```

```
$ cat linux_u8.txt
```

```
$ tac linux_u8.txt
```

```
$ nl linux_u8.txt
```

# fmt, pr, wc

## fmt, pr, wc

`fmt [opcions] fitxer`: permet formatar la sortida.

`pr [opcions] fitxer`: permet veure'l en més d'una columna (si l'amplada de les línees és massa gran, no es veu bé).

`wc fitxer`: ens diu quantes línees, paraules i caràcters hi ha al fitxer (word count).

## Exemples i primeres pipes

```
$ fmt -40 linux_u8.txt
```

```
$ pr -2 linux_u8.txt
```

```
$ wc linux_u8.txt
```

# more i less

## more i less

La comanda `more` permet veure un fitxer de manera paginada. Cada cop que s'omple la pantalla s'atura i espera per continuar: `espai` o `z` per avançar una pàgina.

`return` per a avançar un línia.

La comanda `less` permet veure un fitxer i moure's endavant i enrera amb les fletxes. Per a moure's s'utilitzen les mateixes tecles que quan consultem el manual amb `man`.

## Exemples

```
$ more linux_u8.txt
```

```
$ less linux_u8.txt
```

Mentre l'estem executant podem buscar una paraula amb la comanda `/paraula`.

# Continguts

- 1 Veure fitxers
  - more i less
- 2 Algunes comandes per a manipular fitxers
  - Convertir latin1-utf8
  - Comprimir fitxers
  - a2ps, pdf2ps, ps2pdf
- 3 Canvis sortida i entrada (pipes)
  - Sortida
  - Entrada
  - Més pipes
- 4 Expressions regulars
  - grep
- 5 Ordenar el contingut d'un fitxer

# iconv

## iconv

La comanda `iconv` permet convertir fitxers de caràcters de/a `utf8` a/de `latin1`. Les opcions bàsiques són:

- f: codificació inicial,
- t: codificació final i
- o `nom_fitxer`: fitxer de sortida.

## Exemples

Per a convertir el fitxer `linux_u8.txt` que està en format `utf8` a `latin1` en un fitxer que es digui `linux_l1.txt` fem:

```
$ iconv -f utf8 -t latin1 linux_u8.txt -o linux_l1.txt
```

Per a convertir el fitxer `linux_l1.txt` que està en format `latin1` a `utf8` en un fitxer que es digui `l_u8.txt` fem:

```
$ iconv -f latin1 -t utf8 linux_l1.txt -o l_u8.txt
```

## tar

## tar

La comanda `tar` permet (des)agrupar i (des)comprimir diferents fitxers en un de sol. Les opcions bàsiques són:

- c crear,
- x extreure,
- r afegir,
- t llista el contingut d'un arxiu,
- f `nom_fitxer` fitxer que volem crear/extreure,
- z comprimir amb el format `gzip`.



# tar: exemples

## Exemples

Si volem agrupar el directori `/home/usuari` en el fitxer `usuari.tar.gz` fem:

```
$ tar -zcf usuari.tar.gz /home/usuari
```

Si volem extreure'n el contingut al directori on som:

```
$ tar -zxf usuari.tar.gz .
```

Si volem llistar-ne el contingut, i passar-lo per un `less`:

```
$ tar -tf usuari.tar.gz | less
```

# gzip (gunzip) i bzip2 (gunzip2)

## gzip, bzip2

L'ús més habitual del gzip i el bzip2 (resp. gunzip i bunzip2) és comprimir (resp. descomprimir) un fitxer. El mateix ordinador afegeix (resp. treu) una extensió al fitxer comprimit (resp. descomprimit) i esborra el que no (resp. sí) està comprimit.

## Exemples

```
$ gzip linux_u8.txt
$ gunzip linux_u8.txt.gz
$ bzip2 linux_u8.txt
$ bunzip2 linux_u8.txt.bz2
$ gzip * (comprimeix tots els fitxers del directori, un a un).
$ gunzip *.gz
```

# zip, unzip

## Exemples

Serveix per comprimir/descomprimir fitxers en format zip. Si volem descomprimir el fitxer.zip al directori on estem:

```
$ unzip fitxer.zip
```

Per afegir/reemplaçar el fitxer/directori dir de fitxer.zip:

```
$ zip -r fitxer.zip dir
```

Per esborrar tots els fitxers que comencen per aa que estan comprimits dins fitxer.zip:

```
$ zip -d fitxer.zip aa*
```

Si volem que només ens modifiqui els fitxers del directori dir dins fitxer.zip que tenen una data posterior als que ja hi ha comprimits:

```
$ zip -u fitxer.zip dir
```

Si volem veure el llistat de fitxers que hi ha dins fitxer.zip:

```
$ unzip -l fitxer.zip
```

# a2ps, pdf2ps, ps2pdf

## a2ps, pdf2ps, ps2pdf

Aquestes utilitats serveixen per a convertir diferents tipus de fitxers: a2ps serveix per a convertir arxius `ascii` a `ps`. Per defecte l'envia a la impressora.

pdf2ps serveix per a convertir arxius `pdf` a `ps`.

ps2pdf serveix per a convertir arxius `ps` a `pdf`.

## Exemples

Per a veure el resultat, executem el `kghostview` o bé el `kpdf`.

```
$ a2ps linux_l1.txt -o linux.ps
```

```
$ kghostview linux.ps
```

```
$ a2ps -1B --borders=no linux_l1.txt -o linux.ps
```

```
$ kghostview linux.ps
```

```
$ ps2pdf linux.ps
```

```
$ kpdf linux.pdf
```

# Continguts

- 1 Veure fitxers
  - more i less
- 2 Algunes comandes per a manipular fitxers
  - Convertir latin1-utf8
  - Comprimir fitxers
  - a2ps, pdf2ps, ps2pdf
- 3 Canvis sortida i entrada (pipes)
  - Sortida
  - Entrada
  - Més pipes
- 4 Expressions regulars
  - grep
- 5 Ordenar el contingut d'un fitxer



Habitualment quan executem una comanda esperem que la sortida sigui la pantalla. Per exemple, la sortida de la comanda `ls -l` és un llistat de fitxers per pantalla. Si volem que aquest llistat vagi a parar a un fitxer tenim l'opció d'utilitzar `> nom_fitxer`. Això crearà, o bé reemplaçarà, el fitxer `nom_fitxer` amb el contingut de la sortida de la comanda que executem.

Si volem afegir al final del fitxer el contingut de la sortida utilitzarem la comanda `>> nom_fitxer`.

### Exemple

```
$ cd Desktop
$ ls -l /usr/bin > llistat.txt
$ ls -l /usr/share >> llistat.txt
$ ls -l /usr/lib > llistat.txt
```

# Entrada

&lt;

Si volem passar el contingut d'un fitxer a una comanda podem utilitzar <.

## Exemple

```
$ cat < llistat.txt
```

# pipes

## pipes

Les pipes serveixen per a per passar la sortida per una comanda concreta.

## Exemples

```
$ cat llistat.txt | tac
$ nl llistat.txt | less
$ cat llistat.txt | nl | tac
$ cat llistat.txt | tac | nl
$ fmt -36 linux_u8.txt | pr -2
```



# Continguts

- 1 Veure fitxers
  - more i less
- 2 Algunes comandes per a manipular fitxers
  - Convertir latin1-utf8
  - Comprimir fitxers
  - a2ps, pdf2ps, ps2pdf
- 3 Canvis sortida i entrada (pipes)
  - Sortida
  - Entrada
  - Més pipes
- 4 Expressions regulars
  - grep
- 5 Ordenar el contingut d'un fitxer

# grep

## grep

L'ús bàsic de la comanda `grep` és `grep paraula fitxer` i ens ensenya les línies d'aquest fitxer que contene aquesta paraula. També és molt habitual utilitzar el `grep` com una pipa.

## Exemples

```
$ grep Linux linux_u8.txt --color=auto  
$ last | grep root
```

# Expressions regulars

## Expressions regulars

La comanda `grep` amb l'opció `-E` accepta buscar expressions per plantilles. Per exemple:

```
$ grep -E "Linu[xs]" linux_u8.txt --color=auto
```

busca tant les expressions `Linux` com `Linus`.

## Opcions: rangs

Cada caràcter es representa a ell mateix.

`[acds]` representa una d'aquestes 4 lletres.

`[b-j]` representa una lletra minúscula entre la `b` i la `j`.

`[A-La-l]` representa una lletra entre la `a` i la `l`.

`[^a-z]` el circumflex vol dir negar l'expressió següent. Per tant aquí demanem que no hi hagi una lletra minúscula entre la `a` i la `z`.

`[:space:]` (espai), `[:alpha:]` (`= [a-zA-Z]`), `[:print:]` i `[:punct:]`.

# Expressions regulars

## Opcions: especials

- ^ fòra d'un rang vol dir principi de línia.
- \$ vol dir final de línia.
- \< i \> vol dir principi i final de paraula.

## Opcions: repeticions

- ? l'objecte anterior (si cal, entre parèntesis) pot aparèixer o no, però un sol cop.
- + l'objecte anterior apareix una i més vegades.
- \* l'objecte anterior pot aparèixer o no, i més d'una vegada.
- {n} l'objecte anterior es repeteix exactament n vegades. Variants:
- {n,m} entre n i m vegades.
- {n,} almenys n vegades.

# Exemples

Si volem llistar els enllaços simbòlics del directori /usr/bin:

```
$ ls -l /usr/bin | grep -E "^l" --color=auto
```

Si volem llistar els enllaços simbòlics del directori /usr/bin que comencen per x:

```
$ ls -l /usr/bin | grep -E "^l[[:print:]]*[0-9] \<x"
--color=auto
```

Llistar les llibreries que hi ha a /usr/lib que no comencen per lib:

```
$ ls /usr/lib | grep -E "^[^l][^i][^b]" --color=auto
```

# Continguts

- 1 Veure fitxers
  - more i less
- 2 Algunes comandes per a manipular fitxers
  - Convertir latin1-utf8
  - Comprimir fitxers
  - a2ps, pdf2ps, ps2pdf
- 3 Canvis sortida i entrada (pipes)
  - Sortida
  - Entrada
  - Més pipes
- 4 Expressions regulars
  - grep
- 5 Ordenar el contingut d'un fitxer

## sort

## sort

La comanda `sort` permet ordenar les línees d'un fitxer respecte algun criteri. Les opcions bàsiques són:

`k` indica la columna,

`n` ordena de forma numèrica,

`r` ordena en ordre invers,

`b` no s'han de considerar espais en blanc i

`f` no s'han de considerar majúscules i minúscules com diferents.

## Exemples

```
$ sort -k5 llistat.txt |less
```

```
$ sort -k5,5n llistat.txt |less
```

```
$ sort -k5,5nr llistat.txt |less
```