

Introducció a l'entorn UNIX per a científics (01)

Pere Quintana Seguí



Observatori de l'Ebre (URL-CSIC)

dj 2011.12.02

Introducció a
l'entorn
UNIX per a
científics
(01)

Pere
Quintana
Seguí

Introducció

Scripts

Material

Algunes
comandes
útils

Exemple

1 Introducció

2 Scripts

3 Material

4 Algunes comandes útils

5 Exemple

Les interfícies gràfiques disponibles als entorns GNU/Linux estan molt bé, però no és això el que fa especial aquest sistema. El que el fa especial és el nombre d'eines que s'han desenvolupat al llarg de 40 anys d'història d'UNIX i que permeten fer meravelles tot combinant comandes.

- Tot el que veurem avui és vàlid per a entorns UNIX com per exemple GNU/Linux o Mac OS X.

- El terminal és el programa que fa córrer un *shell*. Per exemple, ja deveu conèixer el Gnome Terminal.
- El *shell* és el programa que processa les comandes i el seus resultats. De fet, és una mena de llenguatge interpretat.
- *Bash* és el *shell* més utilitzat ara, però no és l'únic. N'hi ha de més antics, com el *tsch* o de més nous, com el *zsh*.
- El *shell* ens permet executar comandes interactivament o mitjançant *scripts*.

- Els editors de text són l'eina que utilitzem per a escriure els nostres programes, fitxers de dades, etc.
- **Val MOLT la pena dominar un editor de text**
- Aprendre bé un editor de text costa molt! Anys!

- Hi ha infinits editors de text.
- Vim i Emacs són els clàssics. Són com el rock, mai moren.
- Jo vaig començar amb Vim.
- Després em vaig passar a l'Emacs. Cosa que és com passar del Budisme a l'Islam. Algun dia algú m'ho farà pagar.
- El Gedit és el que ve per defecte amb el Gnome. Activant els seus *plugins* pot arribar a ser molt potent.

L'objectiu d'avui no és fer un curs d'editors, però pot ser estaria bé fer-ne un un dia.

- Un script és un fitxer que conté una sèrie de comandes que s'executen una rere l'altre.
- Els scripts de *shell* es desen en fitxers acabats en `.sh`.
- A la primera línia hi posarem l'interpret (o *shell*) que hem d'utilitzar:
 - `#!/bin/bash`
 - `#!/usr/bin/python`
 - ...

- Per saber on es troba el meu intèrpret: `which`

```
which bash
```

```
which python
```

```
which perl
```

```
which gnuplot
```

```
#!/bin/bash  
text='hola món'  
echo $text
```

- per recuperar el valor d'una variable hem de posar el \$ davant del nom: \$text.
- en casos on el nom tingui caràcters especials podem emprar \${text}.
- Creeu l'*script* hola.sh i executeu-lo.

- Els arguments ens permeten fer programes generals que agafen els valors de les variables de la línia de comandes.
- Podem encadenar tants arguments com vulguem.

Exemple:

```
#!/bin/bash  
nom=$1  
cognom=$2  
echo "El meu nom és $nom $cognom"
```

Per a fer els exercicis heu de baixar-vos el material. Obriu les vostres consoles.

- 1 `wget`
`http://pere.quintanasegui.com/coses/quintana-curs-`
- 2 `tar xvfz quintana-curs-linux-01.tgz`

Abans de veure cap comanda, heu de tenir sempre present que el manual és el vostre amic!

- `man echo`

La filosofia de UNIX és fer eines que fan una sola cosa, però la fan molt bé, i que es puguin combinar mitjançant scripts o a la mateixa línia de comandes.

- Ara veurem unes quantes d'aquestes comandes.

Per exemple, el GMT és hereu d'aquesta filosofia.

- El GMT proveeix comandes al sistema, que es poden executar de manera autònoma o dins d'un *shell* script.
- `psxy`, `xyz2grd`, `pscontour`, `minmax` ...

Veure el contingut dels fitxers

Introducció a l'entorn UNIX per a científics (01)

Pere
Quintana
Seguí

Introducció

Scripts

Material

Algunes
comandes
útils

Exemple

- 1 `cat`: imprimeix en pantalla el contingut del fitxer.
- 2 `more`: com `cat`, però fent una pausa si el fitxer és llarg.
- 3 `less`: com `more`, però més sofisticat (cercar, envant, enrere, etc.).
- 4 `head`: mostra les `n` primeres línies d'un fitxer.
- 5 `tail`: mostra les `n` darreres línies d'un fitxer.

Proveu les comandes amb el fitxer `dades.csv`

- `wc` compta les línies i les paraules d'un fitxer.
- `wc -l` compta només les línies
- `wc -w` compta només les paraules

Quantes estacions hi ha al fitxer `dades.csv`?

- `grep` troba les línies que contenen un text o una “expressió regular”.
- `grep -v` troba les línies que **no** contenen ...

Quantes estacions no tenen dades? Busqueu `nan`.

- > desa la sortida en un fitxer nou, reescrivint-lo si ja existeix.
- » desa la sortida al final d'un fitxer ja existent, o nou si no existeix.

Proveu, per aquest ordre:

- 1 `echo 'hola' > hola.txt`
- 2 `cat hola.txt`
- 3 `echo 'adéu' » hola.txt`
- 4 `cat hola.txt`
- 5 `echo 'hola' > hola.txt`

**Introducció a
l'entorn
UNIX per a
científics
(01)**

Pere
Quintana
Seguí

Introducció

Scripts

Material

**Algunes
comandes
útils**

Exemple

- Creeu un fitxer nou sense les línies que contenen `nan`.

- Creeu un fitxer nou sense les línies que contenen `nan`.

```
grep -v nan dades.csv > nou.csv
```

Permeten trobar *patterns* enlloc de cadenes.

1 `grep ^- dades.csv`

2 `grep 2$ dades.csv`

3 `grep ^2\.[0-9][0-9]\, dades.csv`

Les expressions regulars són potents i complicades.

|
Les *pipes* (canonades) ens permeten connectar comandes.
`grep -v nan dades.csv | tail -n +2 | wc -l`

- `sort` permet ordenar fitxers.
- té l'avantatge que pot tenir en compte les columnes.

```
sort -k 2 -t , dades.csv | less
```

- `-k`: columna
- `-t ,`: delimitador

Deures:

- `man uniq`.

cut ens permet extreure columnes.

Si volem extreure la latitud, la longitud i el valor observat del nostre fitxer:

```
cut -f 1,2,4 -d , dades.csv |grep -v nan > obs.csv
```

- Deures: man paste.

- Hem vist que `sort` i `cut` entenen les columnes, però és un pal dir-li que cada vegada que tingui en compte que el separador és una coma.
- El separador per defecte és el tabulador.
- Substituïm totes les comes per tabuladors.

```
sed "s/,/\t/g"dades.csv
```

- `sed` és **molt** potent. Sap parlar expressions regulars i sap fer moltes més coses que simples substitucions. De fet, és un editor de text.

He preparat dos *scripts* d'exemple:

- 1 `gmt2.sh` és un petit script que permet fer plots de punts. Agafa moltes dades de la línia de comandes, així és molt reutilitzable.
- 2 `plot.sh` prepara les dades per a fer el plot. Aquest és el que hem d'executar per fer els plots.

Exercici:

- 1 Exploreu-los.
- 2 Mireu el que no enteneu.
- 3 A base de `man` trobeu respostes a les vostres preguntes.
- 4 Proveu-los.
- 5 Adapteu-los a les vostres necessitats.