



# SISTEMES OPERATIUS



## Pràctica Laboratori 2

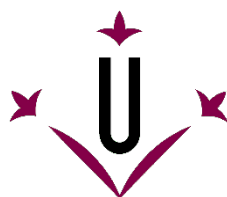
Grau en Enginyeria  
Informàtica



Jordi Lazo  
Florensa



Alejandro Clavera  
Poza



**Universitat de Lleida**  
Escola Politècnica Superior

### 3.1. Comanda history:

#### 1. Cerqueu informació sobre aquesta comanda, expliqueu què fa i doneu exemples d'execució.

La comanda *history* es pot utilitzar per a mostrar l'historial d'instruccions executades en ordre descendent mostrant el número corresponent a cada instrucció executada.

*history n* mostra les últimes *n* comandes realitzades.

*history -c* borra la llista del historial.

*history -d n* borra la comanda que es troba en la posició *n* de la llista de l'historial.

```
lazo@jlazo:~/Desktop$ history 12
1318 Dec/01 - 21:56:54 :wq
1319 Dec/01 - 21:56:54 lz
1320 Dec/01 - 18:18:29 $ export HISTTIMEFORMAT="%h/%d - %H:%M:%S "
1321 Dec/01 - 18:18:59 export HISTTIMEFORMAT="%h/%d - %H:%M:%S "
1322 Dec/01 - 18:19:16 history | tail -5
1323 Dec/01 - 18:19:25 history | tail 10
1324 Dec/01 - 18:19:29 history | tail -10
1325 Dec/01 - 18:19:38 history 12
1326 Dec/01 - 21:56:56 history
1327 Dec/01 - 21:57:15 export HISTTIMEFORMAT="%h/%d - %H:%M:%S "
1328 Dec/01 - 21:57:20 history -21
1329 Dec/01 - 21:57:26 history 12
```

#### 2. Cerqueu informació sobre la variable HISTTIMEFORMAT. De quin tipus és? Per a què serveix?

Es una variable d'entorn de tipus *string* de caràcter temporal ja que al tancar la terminal les característiques d'aquesta variable no es guarden.

Serveix per a mostrar el temps en que es van executar les comandes en la terminal i es mostra en la sortida de la comanda *history*.

```
lazo@jlazo:~/Desktop$ HISTTIMEFORMAT="%d/%m/%y - %H:%M:%S "
lazo@jlazo:~/Desktop$ history 12
1328 01/12/19 - 21:57:20 history -21
1329 01/12/19 - 21:57:26 history 12
1330* 01/12/19 - 22:17:18 HISTTIMEFORMAT="%h/%d - %H:%M:%S "
1331 01/12/19 - 22:17:27 history 12
1332 01/12/19 - 22:21:07 HISTTIMEFORMAT="%h/%d - %H:%M:%S "
1333 01/12/19 - 22:21:10 history 12
1334 01/12/19 - 22:21:42 HISTTIMEFORMAT="%h/%d/%y - %H:%M:%S "
1335 01/12/19 - 22:21:43 history 12
1336 01/12/19 - 22:22:05 HISTTIMEFORMAT="%h/%y/%d - %H:%M:%S "
1337 01/12/19 - 22:22:07 history 12
1338 01/12/19 - 22:22:16 HISTTIMEFORMAT="%d/%m/%y - %H:%M:%S "
1339 01/12/19 - 22:22:17 history 12
```

**3. Modifiqueu un dels fitxers d'arrancada del *shell* per establir-la a tots els vostres intèrprets de comandes. Quin fitxer heu modificat? Amb quin format de data-hora heu establert que surti la data de l'històric? Mostreu la comanda usada.**

Hem modificat el fitxer *.profile* i hem afegit la següent comanda al final del arxiu:

```
export HISTTIMEFORMAT="%d/%m/%y"
```

Per a fer-ho directament hem executat la següent comanda en la terminal de linux:

```
echo "export HISTTIMEFORMAT='%d/%m/%y : "' >> $HOME/.profile
```

Amb aquesta modificació en el *.profile* al utilitzar la comanda *history* ens mostrarà el dia, el mes i l'any de totes les instruccions realitzades.

**4. Aproveiteu aquesta modificació per establir el directori actual de treball (.) al vostre PATH. Com ho heu fet? Indiqueu la comanda i el fitxer modificat.**

Si el que volem es modificar es el *PATH* del directori actual de treball tan sols hem de modificar el arxius *.bashrc* o *.bash\_profile* (en el cas que només vulguem afegir en el intèrpret *bash*) ja que cada usuari té la seva pròpia còpia dels arxius. (Aquests arxius es troben en el directori personal de cada usuari (~)).

En el nostre cas hem modificat *.profile* afegint:

```
$PATH = $PATH:$PWD
```

### 3.2. Un primer script:

**1. Copieu-lo al vostre entorn de treball, anomeu-lo *prac2\_2.sh*. Aquest guió té error/s de sintaxi, error/s de funcionament i/o error/s de compliment de requeriments. Cerqueu-los, quins són? Esmeneu-los fins que pugueu executar correctament l'script. Incloeu el script corregit en la resposta.**

```
1  #!/dev/bash
2  if ( $# -ne 3 )
3  then
4      echo "$0 suma els dos nombres passats com a parametres"
5      echo "Ús: $0 <nombre1> <nombre2>"
6      exit 1
7  end fi
8  echo "$1 + $2 = `expr $1 - $2`"
```

Aquest script conté el següents errors:

- 1- Les condicions dintre del *if* han s'han amb les claus [ ] o amb la condició *test* a més a més s'ha de canviar la condició del numero de paràmetres introduïts a 2 ja que només es sumen 2 arguments.
- 2- S'ha de tabular correctament el *then* i les línies de codi dintre d'aquest.
- 3- El condicional *if* acaba amb la sentència *fi*.
- 4- S'ha de canviar en la primera línia del codi el */dev/* per */bin/*.

Script correcte:

```
1  #!/bin/bash
2  if [ $# -ne 2 ]
3  then
4      echo "$0 suma els dos nombres passats com a parametres"
5      echo "Ús: $0 <nombre1> <nombre2>"
6      exit 1
7  fi
8  echo "$1 + $2 = `expr $1 + $2`"
```

**2. A part de la identificació i correcció dels errors, indiqueu els passos que heu hagut de fer per a poder executar-lo (els passos més importants i fonamentals per a poder executar-lo).**

Per a poder executar aquest script hem hagut que donar permisos d'usuari amb la comanda:  
*chmod +rx nom\_arxiu*

Amb aquesta instrucció donem permisos a l'usuari de lectura i execució del arxiu.

**3. Doneu un llistat de les variables que usa, indicant-ne el tipus al qual pertanyen i quina és la seva funcionalitat.**

En aquest script es troben les següents variables:

- \$# es el número total d'arguments que es passen per el script
- \$1 fa referencia al primer paràmetre passat per el script.
- \$2 referencia al segon paràmetre passat per el script.
- \$0 nom del script.

**4. Expliqueu raonadament què fa.**

Aquest script llegeix 2 números introduïts com a arguments, en el cas de que no s'hagin passat 2 arguments el script mostrarà el funcionament d'aquest. Per el contrari mostrarà la suma del 2 valors introduïts com arguments.

**5. Afegiu 2 solucions alternatives a la darrera comanda del script mantenint la mateixa funcionalitat. És a dir, emprant altres mètodes de fer operacions aritmètiques. Entregueu aquesta versió del script.**

Solució 1:

```
1  #!/bin/bash
2  if test $# -ne 2
3  then
4      echo "$0 suma els dos nombres passats com a parametres"
5      echo "Ús: $0 <nombre1> <nombre2>"
6      exit 1
7  fi
8  echo "$1 + $2 = $(( $1 + $2 ))"
```

Solució 2:

```
1  #!/bin/bash
2  if [ $# -ne 2 ]
3  then
4      echo "$0 suma els dos nombres passats com a parametres"
5      echo "Ús: $0 <nombre1> <nombre2>"
6      exit 1
7  fi
8  let "n1 = $1" "n2 = $2"
9  let "sol = n1 + n2"
10 echo "$1 + $2 = $sol"
```