



UNIVERSITAT^{DE}
BARCELONA

Pràctica 2. Comandes per manipular fitxers

Noah Márquez Vara
Alejandro Guzman Requena

25 març 2022

EXERCICI 1

1. Proves realitzades

La comanda per provar el correcte funcionament de l'script ha estat la mateixa que en el document de l'enunciat de la pràctica:

```
$ ./exercici1.sh gutenber/ txt
```

A més, també ens retorna el resultat esperat:

```
gutenberg/copyright.txt
gutenberg/donate-howto.txt
gutenberg/etext90/bill111.txt
...
```

També hem provat l'script amb altres comandes i ens retornava la sortida esperada. Indiquem les comandes però no les sortides per tal de no fer massa llarg el document:

```
$ ./exercici1.sh gutenber/etext00 txt
$ ./exercici1.sh gutenber/etext99 txt
```

2. Comentaris

L'únic problema ha estat en descobrir que la comanda *find* té una opció per tal d'imprimir varies dades dels fitxers (en el nostre cas la mida i el *path*) i posteriorment aconseguir retallar la sortida per pantalla per tal de mostrar només el *path* amb la comanda *sed*.

EXERCICI 2

1. Proves realitzades

Per tal de comprovar el correcte funcionament de l'script primerament l'hem provat amb l'exemple de l'enunciat de la pràctica:

```
$ ./exercici2.sh gutenber/etext00 500000
gutenberg/etext00/poe2v10.txt : 567995 bytes
gutenberg/etext00/mbova10.txt : 663677 bytes
...
```

L'ordre d'aparició no és el mateix que l'enunciat de la pràctica.

A més, també hem probat l'execució de l'script en altres directoris:

```
$ ./exercici2.sh gutenber/etext99 500000
gutenberg/etext99/spzar10.txt : 653296 bytes
gutenberg/etext99/1onwr10.txt : 629725 bytes
...
```

2. Comentaris

L'únic problema que hem tingut amb aquest script ha estat que no trobàvem la manera per aconseguir que l'script tornés tots els fitxers amb *mida igual o superior* al nombre indicat per paràmetre. Finalment vam pensar que si indicàvem que els volíem de mida estrictament inferior (<) i negàvem la instrucció (amb *-not*) obteníem \geq , que era el que volíem.

EXERCICI 3

1. Proves realitzades

Després d'algunes modificacions finalment els resultats eren els esperats i coincideixen amb els de l'enunciat:

```
$ ./exercici3.sh gutenbergetext00/ txt science
gutenbergetext00/utopi10.txt
La cadena science apareix 2 vegades
gutenbergetext00/poe2v10.txt
La cadena science apareix 12 vegades
gutenbergetext00/dmsnd11.txt
La cadena science apareix 2 vegades
gutenbergetext00/mbova10.txt
La cadena science apareix 12 vegades
```

L'ordre d'aparició no és el mateix que l'enunciat de la pràctica.

2. Comentaris

El principal problema ha estat que no comptava les vegades que apareixia la cadena en una mateixa línia, ja que vam fer servir l'opció *-c* de la comanda *grep*, però després de comprovar el *man* vam veure que només comptava les línies i no totes les aparicions. Finalment es va solucionar al codi adjunt.

EXERCICI 4

1. Proves realitzades

La sortida de l'*script* programat per al directori utilitzat en l'enunciat era l'esperada, per tant hem provat l'*script* amb un altre directori:

```
$ ./exercici4.sh gutenbergetext99/
228 files
38822741 bytes
```

2. Comentaris

Per tal d'aconseguir el propòsit de l'*script* hem hagut d'enllaçar la comanda *find* que ens llista tots els fitxers del directori amb la comanda *awk*; això ha implicat conèixer correctament la sintaxi de les dues comandes mencionades anteriorment. Ens hem ajudat de la funcionalitat de la comanda *find* que permet executar (*exec*) una comanda per cada fitxer trobat ('*ls -l*' en el nostre cas), per tal d'obtenir la informació de cada fitxer i amb l'*awk* sumar la mida en bytes de cada fitxer.

EXERCICI 5

1. Proves realitzades

Per tal de comprovar el correcte funcionament de l'script, hem provat tant amb l'usuari del sistema que mostra l'enunciat, com amb el *root*:

```
$ ./exercici5.sh oslab
VSZ: 18549900 bytes
RSS: 976096 bytes
```

```
$ ./exercici5.sh root
VSZ: 6705496 bytes
RSS: 430864 bytes
```

2. Comentaris

L'única complicació que vam trobar amb aquest exercici va ser a l'hora d'aprendre la potència que ens ofereix la comanda *awk* amb el seu propi llenguatge de programació. Un cop entès com funcionava correctament, l'exercici va ser prou fàcil.

EXERCICI 6

1. Proves realitzades

L'output de l'*script* si donem com a paràmetres els indicats a l'enunciat no és el mateix, degut a que el NIUB de mostra no es troba dins del fitxer *alumnes.csv*:

```
$ ./exercici6.sh niub18076796 practiques F00
El NIUB no es troba al fitxer!
```

En canvi si provem amb un niub que si existeix al fitxer, l'*script* funciona correctament:

```
$ ./exercici6.sh niub18018000 practiques G22
Grup de practiques modificat.
```

Si s'introdueix un tipus de grup que no és ni "practiques" ni "problemes" surt un missatge d'error:

```
$ ./exercici6.sh niub18076796 teoria F00
No has introduït el tipus de grup correctament!
```

2. Comentaris

La programació d'aquest script ha estat la més llarga de tota la pràctica.

El primer problema que ens ha sorgit era que no sabíem com obtenir el nombre de línia on es trobava el niub indicat per paràmetre. Però ajudant-nos de l'opció *-n* de la comanda *grep* que ens indica el nombre de línia separat per ":" i amb una canonada a la comanda *awk* separant per aquest caràcter, obteníem la primera columna (nombre de línia).

Un altre problema que hem tingut és que a l'hora de fer servir la comanda *sed* per modificar el fitxer, ens havíem deixat l'opció *-i* per tal de modificar el fitxer i a més havíem indicat una *g* al final de la comanda, el que feia que es modifiquessin totes les coincidències.