

Pràctica 2. Comandes per manipular fitxers

Sistemes Operatius 1

Març 2022

En aquest document es mostren els exercicis a programar per a la pràctica 2. Per tal d'intentar seguir la “filosofia” dels scripts a Unix els arguments es passen per línia de comandes i la sortida s'imprimeix per pantalla.

A continuació hi ha dues seccions: la primera secció explica els scripts a programar i la segona indica l'informe que s'ha d'entregar amb els scripts. La puntuació dels scripts és d'un 80% i la de l'informe un 20%.

1 Scripts a programar

A continuació es presenten 6 exercicis a resoldre. Tots els scripts necessiten arguments per línia de comandes. En cas que no s'especifiquin el nombre necessari d'arguments l'script haurà de mostrar un missatge d'error per pantalla i sortir amb un codi d'error, `exit 1`. En cas que s'especifiquin tots els arguments, podeu suposar que s'han especificat de forma correcta. Si l'script s'executa correctament, caldrà que aquest surti amb un codi d'execució correcte, `exit 0`.

Per exemple, suposeu un script en què s'han d'especificar dos arguments: el primer argument ha de ser un número i el segon una cadena. Si no s'especifiquen els dos arguments, caldrà donar un missatge d'error per pantalla i sortir de l'script amb el codi d'error. Si s'especifiquen els dos arguments, podeu suposar que el primer argument és un número i el segon una cadena.

Tots els noms de fitxers, directoris i cadenes a tractar no tindran espais ni caràcters especials i, a més, podeu suposar que tots els fitxers a tractar són fitxers de text pla. En alguns d'aquests exercicis es passa com a paràmetre un directori: l'script ha de funcionar independentment del directori que es passi com a paràmetre. També podeu suposar que no hi ha enllaços simbòlics en els fitxers analitzats. Al campus teniu disponible un fitxer comprimit que conté un gran nombre de fitxers de text amb els quals podeu fer proves (gutenberg).

Cada script té la mateixa puntuació. **Per fer aquests exercicis està permès fer servir les instruccions que hi ha al document del primer i segon tutorial llevat que s'indiqui el contrari. Podeu fer servir qualsevol opció de la comanda encara que no estigui indicada al tutorial.** En cas que es faci servir una instrucció no permesa la puntuació de l'exercici serà de 0 punts.

Es recomana fer els exercicis junts, en parella, ja que hi ha exercicis que són difícils de resoldre sense haver resolt l'anterior. No us repartiu els exercicis per treballar de forma independent!

Exercici 1

Es demana implementar un script `exercici1.sh` que rep dos arguments, un directori i una extensió. L'script buscarà **de forma recursiva** tots els fitxers amb l'extensió especificada i els ordenarà segons la seva mida (en bytes) de menor a major. Finalment escriurà per pantalla només el nom dels 10 primers fitxers.

En cas que es faci servir un bucle per implementar l'script, la puntuació màxima del problema serà de 1,5 punts. La puntuació màxima també serà de 1,5 en cas que s'utilitzin fitxers (temporals) per emmagatzemar resultats intermedis. En altres paraules, feu servir canonades (i no pas redirecció) per implementar aquest exercici!

Exemple d'execució:

```
$ ./exercici1.sh gutenber/ txt
gutenberg/copyright.txt
gutenberg/donate-howto.txt
gutenberg/etext90/bill11.txt
gutenberg/etext98/2ws2710.txt
gutenberg/etext05/bib0010h/Readme.txt
gutenberg/etext90/liber11.txt
...
```

Exercici 2

Fer un script anomenat `exercici2.sh` que rep dos arguments: un directori i un valor numèric. L'objectiu de l'script és analitzar la mida en bytes dels fitxers que hi ha dintre del directori, ignorant tots els subdirectoris. L'script haurà de mostrar per pantalla el nom i la mida en bytes de tots els fitxers amb mida igual o superior al nombre indicat per paràmetre. Aquest exercici vol replicar l'exercici 4 de la pràctica 1, però sense fer servir bucles ni fitxers temporals. La implementació NO es realitzarà de manera recursiva.

Exemple d'execució (l'ordre amb què apareixen els resultats per pantalla pot ser diferent del que se us mostra aquí):

```
$ ./exercici2.sh gutenber/etext00 500000
gutenberg/etext00/00ws110.txt : 4516616 bytes
gutenberg/etext00/1cahe10.txt : 1736316 bytes
gutenberg/etext00/1vkip11.txt : 1696860 bytes
gutenberg/etext00/2cahe10.txt : 1734379 bytes
gutenberg/etext00/2yb4m10.txt : 928034 bytes
...
```

Exercici 3

Fer un script anomenat `exercici3.sh` que rep tres arguments: un directori, una extensió i una cadena. L'script ha d'analitzar, de forma recursiva, els fitxers dintre del directori amb la extensió corresponent. Per a cada fitxer trobat, cal imprimir el nom complet del fitxer (path) i analitzar quants cops hi apareix la cadena dins del fitxer, ja sigui com a paraula individual o com a part d'una altra.

La puntuació màxima serà de 1,5 en cas que s'utilitzin fitxers (temporals) per emmagatzemar resultats intermedis. Es pot fer servir com a màxim un bucle per implementar l'script.

Exemple d'execució:

```
$ ./exercici3.sh gutenbergetext00/ txt science
gutenbergetext00/00ws110.txt
la cadena science apareix 126 vegades
gutenbergetext00/1cahe10.txt
la cadena science apareix 59 vegades
gutenbergetext00/1vkip11.txt
la cadena science apareix 16 vegades
gutenbergetext00/2cahe10.txt
la cadena science apareix 113 vegades
gutenbergetext00/2yb4m10.txt
la cadena science apareix 4 vegades
...
```

La funcionalitat a implementar es pot fer amb una línia de comandes encadenades. No es permeten fer servir bucles ni redirecció cap a (o des de) fitxer per implementar aquest exercici.

Tingueu en compte que la comanda **grep**, per defecte, imprimeix la línia sencera on apareix la paraula trobada. La paraula pot aparèixer diverses vegades en una mateixa línia. Busqueu doncs l'opció de **grep** que permeti comptar-les correctament. Els resultats poden variar dels mostrats aquí.

Exercici 4

Fer un script anomenat **exercici4.sh** que rep un argument, un directori, que podeu suposar que existeix. L'script ha de calcular **de forma recursiva** el nombre total fitxers que conté així com el seu pes total, és a dir, la suma de la mida de tots els fitxers en bytes. Cal evitar que se sumin la mida dels directoris i podeu suposar que no hi ha cap altre tipus d'element (com enllaços simbòlics).

En cas que es faci servir un bucle per implementar l'script, la puntuació màxima del problema serà de 1,5 punts. La puntuació màxima també serà de 1,5 en cas que s'utilitzin fitxers (temporals) per emmagatzemar resultats intermedis.

Exemple d'execució:

```
$ ./exercici4.sh gutenbergetext00/
47 files
33317565 bytes
```

Exercici 5

Es demana fer un script anomenat **exercici5.sh** que accepti un argument que serà un usuari del sistema. Per exemple, l'usuari **oslab**. Mitjançant la comanda **ps aux** haureu de trobar els processos d'aquest usuari i calcular el total de memòria que està ocupada per aquests processos. La memòria ocupada per un procés es pot saber fent servir la columna **VSZ**, que està associada a la memòria (en bytes) que ocupa el procés a l'espai de memòria virtual, així com la columna **RSS**, que indica quants bytes està fent servir en aquell moment el procés a memòria física RAM.

En cas que es faci servir un bucle per implementar l'script, la puntuació màxima del problema serà de 1,5 punts. La puntuació màxima també serà de 1,5 en cas que s'utilitzin fitxers (temporals) per emmagatzemar resultats intermedis.

Exemple d'execució:

```
$ ./exercici5.sh oslab
VSZ: 50009244 bytes
RSS: 4036016 bytes
```

~~La funcionalitat a implementar es pot fer amb una línia de comandes encadenades. No es permeten fer servir bucles ni redirecció cap a (o des de) fitxer per implementar aquest exercici.~~

~~Tingueu en compte que la comanda **grep**, per defecte, imprimeix la línia sencera on apareix la paraula trobada. La paraula pot aparèixer diverses vegades en una mateixa línia. Busqueu doncs l'opció de **grep** que permeti comptar-les correctament. Els resultats poden variar dels mostrats aquí.~~

Exercici 6

En l'entorn de la informàtica és habitual treballar amb fitxer del tipus CSV (comma-separated values) o TSV (tab-separated values). En aquest cas haureu de crear un script anomenat **exercici6.sh** que donat el fitxer **alumnes.csv**, de tipus CSV, hi permeti buscar un alumne i permeti modificar el fitxer per tal de canviar el grup pertinent del alumne, sigui del grup de problemes o de pràctiques. En concret, el primer paràmetre de l'script serà el niub de l'estudiant, el segon paràmetre serà el paràmetre "problemes" o "practiques" depenent de si es vol modificar el grup de problemes o de pràctiques de l'estudiant, el tercer paràmetre serà el nou grup assignat a l'estudiant. Podeu suposar que el niub especificat és correcte i existeix al fitxer CSV.

En cas que es faci servir un bucle per implementar l'script, la puntuació màxima del problema serà de 1,5 punts. La puntuació màxima també serà de 1,5 en cas que s'utilitzin fitxers (temporals) per emmagatzemar resultats intermedis.

Per separar els elements del fitxer CSV entre sí es pot fer servir l'eina **awk**. Mitjançant l'opció -F es pot establir un separador diferent que el de per defecte. En el següent cas indiquem que el separador és un punt i coma ";": `awk -F ";" '{print $X}'`, on \$X és la columna que volem extreure. A l'hora de canviar l'alumne de grup de pràctiques o de problemes caldrà mantenir el separador ";" com a separador entre columnes, ja que el punt i coma és un caràcter que el sistema utilitza habitualment per separar columnes en un fitxer CSV.

Exemple d'execució:

```
$ ./exercici6.sh niub18076796 practiques F00
Grup de pràctiques modificat.
$ ./exercici6.sh niub18076796 problemes ABO
Grup de problemes modificat.
```

No es permeten fer servir bucles ni redirecció cap a (o des de) fitxer per implementar aquest exercici. En cas de que el niub no existeixi al fitxer **alumnes.csv**, es donarà un missatge per pantalla.

2 Informe

A més dels scripts caldrà entregar un informe sobre la feina feta. **L'informe a entregar ha d'estar en format PDF o equivalent (no s'admeten formats com odt, docx, ...).** Aquest

informe ha de mostrar les proves que s'han realitzat per assegurar el bon funcionament del codi. No inclogueu una descripció de l'algorisme implementat llevat que cregueu rellevant comentar algun detall. Inclogueu, preferentment en format text (tal com es fa en aquest document), la comanda que heu executat així com la sortida obtinguda i algun comentari breu descrivint el resultat si ho creieu necessari. En cas que no us funcioni el codi indiqueu també el problema detectat així com possibles sospites de quin pot ser la font del problema. Sigueu breus i clars a les vostres respostes, no fa falta que us esteneu en el text escrit.

En cas que preferiu incloure captures de pantalla en comptes d'incloure el resultat en format text, assegureu-vos que el text de la captura es pot llegir bé (és a dir, que tingui una mida similar a la resta del text del document) i que totes les captures siguin uniformes (és a dir, que totes les captures tinguin la mateixa mida de text).

Cada exercici té la mateixa puntuació (80% de la qualificació) i l'informe té un pes del 20% de la qualificació. L'informe ha de tenir **una llargada màxima de 3 pàgines** (sense incloure la portada). El document s'avaluarà amb els següents pesos: proves realitzades i comentaris associats, un 60%; escriptura sense faltes d'ortografia i/o expressió, un 20%; paginació del document feta de forma neta i uniforme, un 20%.

Cada grup ha de pujar un únic arxiu ZIP que contingui els scripts i l'informe.