Informe Lab2 Sistemes Operatius

Aran Roig i Biel Palomar

Març 2025

1 Introducció

En aquest document veurem l'implementació dels nostres scripts de la pràctica 2. Que hem fet per controlar errors, i quines comandes hem utilitzat per a fer certes operacions. Totes les comandes utilitzades son dins del tutorial o que s'han fet servir a classe.

2 Exercicis

2.1 Exercici 1

Hem escrit 3 diferents comprovacions per a cobrir tots els casos, si alguna de les coses no es compleix sortirem del programa immediatament amb codi 1. Primer de tot mirem que ens passin pels paràmetres un únic fitxer.

```
if  [ $# -ne 1 ]; then
    echo "Se requiere el siguiente formato: ./ejercicio.sh <archivo_calidad_agua>"
    exit 1
fi
```

Tot seguit comprovem que aquest fitxer existeixi.

```
if  [ ! -f "$1" ]; then
    echo "El archivo \'$1\' no existe."
    exit 1
fi
```

Finalment, comprovem que a la tercera columna del csv, separat per comes hi hagi la columna pH. Si és aixi seguim amb el codi.

```
if [[ $(head -n 1 "$1" | awk -F ',' '{print $3}') != "pH" ]]; then
    echo "El archivo tiene no tiene el formato correcto. Debe haber una columna de pH."
    exit 1
fi
```

Amb aquesta línia agafem totes les línies del fitxer, agafem nomes la 3a columana de pH i obviem les que tinguin lletres amb el grep.

```
linies=$( cat $1 | awk -F ',' '{print $3}' | grep -E '[0-9]+(\.[0-9]+)?' )
```

Llavors comparem els nivells amb mes o menys del límit i si es necessari incrementem el contador. Per acabar imprimim els resultats com es demana

2.2 Exercici 2

Primer comprovem que hi hagi com a mínim un paràmetre i guardem en una variable els predeterminats. Aleshores en cas que s'introdueixin els opcionals, els canviem segons correspongui.

Després, llegim el fitxer `archivo_consumo` i agrupem les línies que corresponen al mes indicat. Per fer això, agafem la primera columna del fitxer i mirem si coincideix amb l'expressió regular `[0-9]+-<"\$mes"-[0-9]+` on `$mes` es substitueix per la variable mes (ex. si mes=01 aleshores l'expressió regular és `[0-9]+-01-[0-9]+`) La propia comanda de `awk` ja fa la mitjana de tots els litres.

```
if [ ! -z "$3" ]; then
  mes=$3
  litres=$(cat $1 | awk -v r="[0-9]+-$mes-[0-9]+" -F '|' '{ if($1 ~ r)
  { a[$2]+=$3; n[$2]++; } } END {for(i in a){printf "%s|%.2f\n", i, a[i]/n[i];}}' | grep -E H )
  # years es una lista de todos los años distintos en las entradas.
  # Cogemos de la primera columna los años distinitos
  years=$(cat $1 | awk -F '|' '{print $1}' | awk -F '-' '{a[$1]="";}
  END {for(i in a){print i;}}' | tail -n +2)
else
  litres=$(cat $1 | awk -F '|' '{a[$2]+=$3; n[$2]++; } END
  { for(i in a){printf "%s|%.2f\n", i, a[i]/n[i];}}' | grep -E H )
  years=""
fi
```

La variable litres guarda separat per | el nom de la casa i la mitjana de litres consumits. Més endavant tornem a fer servir `awk` per recuperar aquestes dues dades. Finalment en cas que les dades continguessin dades de diferents anys pero que corresponguessin al mateix mes, agafem la mitjana dels mesos i mostrem tots els anys analitzats.

2.3 Exercici 3

A l'exercici 3 hem fet 2 controls d'errors. Primer de tot mirem que ens passin el nombre de paràmetres correcte com a tots els exercicis. Tot seguit mirem si el que ens han passat és un directori fent servir l'opció -d.

```
if [ ! -d "$1" ]; then
        echo "El directorio $1 no és un directorio o no existe"
        exit 1
fi
```

Si alguna de les dues condicions és errònia sortim de l'script amb codi 1. L'exercici l'hem dividit en 2 parts, primer les estacions amb sequía. Per a això fem un for de tots els fitxers i per cada fitxer llegim la segona columna que ens diu la quantitat de precipitació. Tot això ho sumeme a una variable global i si es menor que el mínim imprimim el resultat.

Per llegir les línies dels fitxers fem servir l'awk igual que al final del Exercici 1.

Per a fer la comparació de dos números decimals farem servir el awk de la següent manera. Si $1 és més petit que $2 retornara 1.

```
if [ "$(echo "$total 30" | awk '{print ($1 < $2)}')" -eq 1 ];then
```

Seguidament mirem les estacions amb risc d'inundació. Simplement, anem línia a línia de cada fitxer i si és per sobre del límit, imprimim el dia i quantitat d'aigua.

Les comandes són iguals que a la primera part de l'exercici.

2.4 Exercici 4

Obtenim la llista de processos utilitzant la comanda

```
proc=$( ps -Ao pid,ni,time,cmd | head -n 6)
```

Després creem una variable de nom option que guarda l'opció seleccionada que inicialitzem a -1. Per fer el menú simplement hem fet un `case` dins d'un `while`. Mentres l'opció llegida no sigui un 5 el `while` no s'atura, i depenent de l'opció seleccionada (o si és invàlida) fem una acció o una altra.

- Per matar un procés hem fet servir la comanda `kill`
- Per canviar la prioritat d'un procés hem utlitzat

  ```
  renice $newNice $procId > /dev/null
  ```

  on `newNice` i `procId` són la nova prioritat i la id del procés respectivament.
- Per mostrar les estadístiques hem fet servir les comandes corresponents. Per mostrar l'ús del primer disc hem fet

  ```
  echo $(df -h | head -n 2 | tail -n 1)
  ```

  a part, per veure els primers 5 processos que consumeixen més CPU hem fet servir

  ```
  echo "$(top -b -o +%CPU | head -n 12 | tail -n 6)"
  ```

En tots els casos en cas que pugui donar error comprovem amb la variable `$?` si la comanda anterior ha sortit amb un codi d'error, i aleshores avisem si és el cas

2.5 Exercici 5

Per començar l'script de l'exercici 5 hem comprovat que se'ns passés un paràmetre i que aquest fos un directori exactament igual que a l'exercici3. Si no és així sortim del script amb un codi 1.

Un cop comprovat això ens definim el nom del directori com l'última part de la ruta, per a evitar si ens havien passat una ruta absoluta.

Primer de tot iterem per cada fitxer del directori, i creem una carpeta per cada categoria (primera paraula del fitxer separades per _), excepte per la paraula informe, ja que simultàniament estem creant un, i sinó quedaria malament. Per a fer-ho fem servir aquestes dues comandes, la primera agafa la part final de la ruta amb $NF i la 2a agafa la primera part del nom.

```
paraula=$(echo "$file" | awk -F '/' '{print $NF}')
paraula=$(echo "$paraula" | awk -F '_' '{print $1 }')
```

Seguidament, tornem a iterar pel directori per aquest cop movent cada fitxer a la seva carpeta.

Un cop acabat el bucle imprimim el resultat dels comptadors que teníem a dins dels bucles per a donar un resum a l'usuari.

2.6 Exercici 6

Primer ens hem assegurat que hi hagin els tres paràmetres obligatoris. Un cop presents, hem fet un grep recursiu per obtenir les linies que contenen el paràmetre que ens interessa. Separem per ':'. La primera columna és la ruta del fitxer. La segona el paràmetre i la tercera el valor amb un espai davant. Tallem amb el awk un altre cop per treure la segona columna (el paràmetre) i escribim la primera i tercera columna amb un | entremig. Després, passem un altre cop per awk i separem per ' '. Ens queden al final el nom del fitxer i el valor del paràmetre separats per |:

```
res=$(grep -r $parametro $directorio | awk -F ':' '{print $1  "|" $3}'
| awk -F ' ' '{print $1 $2}')
```

Finalment de totes les linies guardades a $res, ens quedem amb les que l'umbral és més petit que la segona columna (separades per |). Utilitzem awk per fer la comparació. I després l'unic que quedaria és mostrar els paràmetres:

```
for line in $res; do
  archivo=$(echo "$line" | awk -F '|' '{print $1}')
  numero=$(echo "$line" | awk -F '|' '{print $2}')
  if [ $(echo "" | awk "{ print $umbral < $numero }") -eq 1 ]; then
    echo "----------------------------------------"
    echo "Archivo: $archivo"
    nombre=$(cat $archivo | grep 'Estación:' | cut -d ' ' -f2-)
    echo "Estación: $nombre"
    echo "Valor de $parametro: $numero"
  fi
done
echo "----------------------------------------"
exit 0
```

3 Conclusions i valoracions personals

En conclusió, aquesta pràctica ens ha permès fer servir els scripts bàsics de bash per a l'anàlisi de dades i gestió de fitxers.

Personalment, opinem que és una bona pràctica, ja que té diverses casuístiques i permet trobar molts tipus de dades, i organitzades de dades diferents, com també fitxers.

4 Contribució

Aran Roig Serra 50%
Biel Palomar Gonzàlez 50%