



Asignatura: Sistemas Operativos
Grado: Ingeniería Informática

Practica Shell Script

Integrantes del equipo de prácticas:
Elías Leonel García Sáenz 1492201

20/11/2018

ÍNDICE

1. Objetivos de la práctica.....	Pág. 2
2. Descripción y planteamiento de la práctica.....	Pág. 2
3. Descripción de los procedimientos utilizados a nivel funcional.....	Pág. 4
4. Descripción de los problemas surgidos durante la realización de la práctica y las soluciones halladas.....	Pág. 4
5. Conclusiones.....	Pág. 5

Objetivos de la práctica

El objetivo principal de esta práctica es adquirir los suficientes conocimientos de la terminal de Linux para poder hacer con total dominio un Shell script. Para conseguir esto se nos propone hacer un Shell script para ver que restaurante es mejor de entre varios que existen, aparte también asignar unas puntuaciones a estos y ver cuál es mejor. Para finalizar con éxito esta práctica nos tendremos que poner manos a la obra buscando información y ejemplos para saber cómo funciona cada uno de los tantos comandos que existen y conseguir que nuestro programa funcione de manera correcta y rápida.

Descripción y planteamiento de la práctica

Opción 1

En esta opción lo que hacemos para empezar es guardar las líneas del archivo a leer en una variable, acto seguido pasamos a leer todas las líneas de este y guardad en arrays dinámicos, tanto las ID's como las puntuaciones (sin que se repitan). Cada vez que se guarda un nuevo ID, se procede a hacer la media de las puntuaciones para tener el rating de este restaurante (cada ID es un restaurante). Una vez hecho esto simplemente se pasa a comprobar los valores de "price", "dresscode", "smokingarea", "alcohol" y "accessibility" y finalmente se muestra el restaurante recomendado con sus respectivos valores.

```
Restaurante recomendado: Paniroles: 1.00; .8 ( 0.4 de price + 0.1 de dress_code + 0.2 de alcohol + 0.1 de smoking_area + )
Pulsa cualquier tecla para continuar
```

Opción 2

En la opción 2 se procede a realizar una búsqueda más precisa de restaurantes, para ello primero leemos que valores quiere que tenga el/los restaurantes (Price, alcohol, dresscode, smokingarea y accessibility).

Cuando tenemos esto volvemos a utilizar el mismo método de la opción uno, guardamos en una variable las líneas del archivo a leer y leemos todos los ID's, y nos quedamos tan solo los que coinciden con los que tengan los valores que hemos leído anteriormente.

```
case "$tecla" in
1)
num_l=$(cat rating_final.csv | wc -l)
x=1
z=1
declare -A rat
declare -A IDs
declare -A puntuacion
for (( i=1; i<=$num_l; i++))
do
j=$(cut -d "," -f 2,3 rating_final.csv | sort | cut -d "," -f 1 | head -$i | tail -1)
k=$(cut -d "," -f 2,3 rating_final.csv | sort | cut -d "," -f 2 | head -$i | tail -1)

if [ "$i" = '1' ]; then
IDs[1]=$(cut -d "," -f 2,3 rating_final.csv | sort | cut -d "," -f 1 | head -1 | tail -1)
puntuacion[1]=$(cut -d "," -f 2,3 rating_final.csv | sort | cut -d "," -f 2 | head -1 | tail -1)
else
if [ "$j" = "${IDs[$x]}" ]; then
puntuacion[$x]=$((puntuacion[$x] + $k))
((z++))
else
rat[$x]=$(echo "scale=2; ${puntuacion[$x]}/$z" | bc)
((x++))
z=1
IDs[$x]=$j
puntuacion[$x]=$k
fi
fi
done
```

```
num_l=$(cat geoplaces2.csv | wc -l)
x=0
declare -A IDs
for(( i=1; i<=$num_l; i++))
do
p=$(cut -d "," -f 1,16 geoplaces2.csv | sort | cut -d "," -f 2 | head -$i | tail -1)
d=$(cut -d "," -f 1,14 geoplaces2.csv | sort | cut -d "," -f 2 | head -$i | tail -1)
s=$(cut -d "," -f 1,13 geoplaces2.csv | sort | cut -d "," -f 2 | head -$i | tail -1)
a=$(cut -d "," -f 1,15 geoplaces2.csv | sort | cut -d "," -f 2 | head -$i | tail -1)
w=$(cut -d "," -f 1,12 geoplaces2.csv | sort | cut -d "," -f 2 | head -$i | tail -1)

if [ "$price1" = "$p" ] || [ "$price2" = "$p" ];then
if [ "$dresscode1" = "$d" ] || [ "$dresscode2" = "$d" ];then
if [ "$smokingarea1" = "$s" ] || [ "$smokingarea2" = "$s" ] || [ "$smokingarea3" = "$s" ];then
if [ "$accessibility1" = "$a" ] || [ "$accessibility2" = "$a" ];then
if [ "$alcohol1" = "$w" ] || [ "$alcohol2" = "$w" ];then
((x++))
IDs[$x]=$(cut -d "," -f 1 geoplaces2.csv | sort | head -$i | tail -1)
fi
fi
fi
fi
fi
fi
done
```

Una vez tenemos estos ID's volvemos a revisar cual tiene el rating más alto, en este caso en vez de utilizar el método de la opción 1, que es revisar todos los ID's, tan solo revisaremos los ID's que queremos, con el comando `grep`.

```
declare -A puntuacio
declare -A rat
z=0

for(( m=1; m<=$x; m++))
do
    gr=$(grep -n "${IDs[$m]}" rating_final.csv | cut -d ":" -f 1)
    num_lin=$(echo $gr | wc -w)

    for ((conta=1; conta<=$num_lin; conta++))
    do
        lin=$(echo $gr | cut -d " " -f $conta)
        punt=$(cut -d "," -f 2,3 rating_final.csv | sort | cut -d "," -f 2 | head -$lin | tail -1)
        puntuacio[$m]=$((puntuacio[$m] + $punt))
        ((z++))
    done

    rat[$m]=$((echo "scale=2; ${puntuacio[$m]}/$z" | bc))
done
z=0
```

Finalmente se muestra el restaurante recomendado por pantalla.

Opcion 3

Esta opción simplemente muestra por pantalla el valor numérico que tienen los valores "Price", "alcohol", "dresscode", "smokingarea" y "accessibility".

```
echo "Peso precio: $preu"
echo "Peso smoking area: $smokingarea"
echo "Peso alcohol: $alcohol"
echo "Peso dress code: $dresscode"
echo "Peso accesibility: $accesibility"
echo " "
echo "-----"
echo " "
echo "Precios: $valors_preu"
echo "Smoking areas: $valors_smokingarea"
echo "Alcohol services: $valors_alcohol"
echo "Dress codes: $valors_dresscode"
echo "Accessibilities: $valors_accesibility"
echo " "
```

Opcion 4

Esta última opción modifica el valor numérico de los valores que nosotros queramos. Introducimos que valores queremos modificar (Price, alcohol, dresscode, smokingarea o accesibility) y una vez finalizamos la lectura de los valores a modificar nos vuelve a preguntar qué valor numérico queremos darle a estos.

```
ti=0
while [ $ti -eq 0 ]
do
    echo "Indique nuevo peso para price: "
    read -r price
    valor_preu=$price
    ti=1
done

ti=0
while [ $ti -eq 0 ]
do
    echo "Indique nuevo peso para dress code: "
    read -r dresscode
    valor_dresscode=$dresscode
    ti=1
done
```

Opcion 5

Esta opción simplemente vuelve a dejar los valores numéricos de "Price", "alcohol", "dresscode", "accessibility" y "smokingarea" como estaban y sale del programa.

```
5)
preu1="medium"
preu2="medium"
preu3="medium"
valor_preu=0.4
dresscode1="informal"
dresscode2="casual"
dresscode3="casual"
valor_dresscode=0.1
alcohol1="Wine-Beer"
alcohol2="Wine-Beer"
alcohol3="Wine-Beer"
valor_alcohol=0.2
smokingarea1="none"
smokingarea2="not-permitted"
smokingarea3="only at bar"
smokingarea4="section"
smokingarea5="permitted"
valor_smokingarea=0.1
accessibility1="partially"
accessibility2="completely"
accessibility3="no_accessibility"
valor_accessibility=0.2

clear
exit
```

Descripción de los procedimientos utilizados a nivel funcional

Los comandos utilizados en esta práctica son:

Cat "nombre archivo": utilizado junto al comando "wc -l". Sirve para mostrar por pantalla todo lo que hay en un archivo.

Wc "nombre archivo": utilizado con el "-l" final. Sirve para contar las palabras de un archivo, utilizado con el "-l" final sirve para contar el número de líneas en vez de palabras.

Cat "nombre archivo" "| wc": denominado pipe se utiliza para combinar comandos. En este caso lo que hace es, primero muestra todo y finalmente por pantalla solo se imprime el número de líneas del archivo leído.

Cut -d "." -f 2: utilizado para cortar según lo que tengamos en los "", y en el número designado por -f. Ejemplo: Cut -d "," -f 2, nos mostrara solo lo que haya después de la segunda coma. Utilizado para obtener los ID's y puntuaciones, tanto como el nombre de los restaurantes y demás valores.

Echo "...": el comando más básico que simplemente muestra por pantalla lo que hay entre comillas. Utilizado para mostrar el resultado de lo que ha hecho el código.

Head - "...": muestra por pantalla las "." primeras líneas

Tail - "...": muestra por pantalla las "." últimas líneas. Utilizado junto el head para mostrar una sola línea en concreto.

Grep "... "nombre archivo": busca lo introducido en "." dentro del archivo. Utilizado con la terminación -n y cuts para saber el número de la línea donde se encontraba lo introducido entre comillas.

Descripción de los problemas surgidos durante la realización de la práctica y las soluciones halladas

Al realizar la práctica sí que es verdad que han surgido algunos problemas y más aun siendo primerizo en este entorno de Linux, pero poco a poco con información y dedicación se ha ido saliendo.

Uno de los principales problemas que tenía siempre era la manera en la que estructurar mis ideas planteadas principalmente en C, que es lo que había manejado hasta entonces.

También he tenido problemas al intentar buscar algo entre todos los restaurantes, cuya solución fue utilizar el comando "grep", tanto para que funcionara, como para acortar código y acortar el tiempo de espera del Shell Script a la hora de ejecutarse.

La mayoría de problemas son de este tipo, no me funciona algo porque está mal escrita la estructura y para solucionarlo busco información y corrijo esta.

Conclusiones

La realización de este programa puede resultar algo complicada y larga, pero al final es como cualquier otra actividad, es cuestión de ponerse e informarse, ya que este programa lo podría haber hecho en menos tiempo, pero debido a mi falta de conocimiento he tardado más de lo deseado tal vez. Aun así me llevo la satisfacción de haberlo completado y el hecho de que cada opción haga su función de manera correcta me hace pensar que algo he aprendido de esta práctica.

Como conclusión podría decir que este lenguaje no acaba de ser tan complicado pero como cada lenguaje tiene sus maneras de estructurarse y manera de utilizarse, en concreto este programa te enseña a utilizar prácticamente la mayoría de los comandos básicos de este lenguaje, cosa que está muy bien porque esto es lo que hemos aprendido al final. Diría que aún no he visto todo lo que es Shell Script, porque supongo que habrán mas comandos y maneras de aplicarse, etc..., pero me quedo con lo básico que está bastante bien, en un futuro ya sabré como empezar un Script y no iré tan perdido gracias a estos conocimientos adquiridos.