

CFGS – 1º DAW – 2020



Florida

Universitària

Sistemas informàtics

Título:

T6-AP6b_shell-script-2

Apellidos y nombre del autor/a:
Raúl Callado Montero

Fecha:
21/Marzo/2021

Detalle:

- Fecha límite de entrega: lunes, 22 de marzo de 2021, 23:59
- Se entregará un enlace a un repositorio que contenga los scripts de cada ejercicio.
- Dedicación estimada: 7 horas

Sumario

Punto 1.....2

Punto 2.....3

Punto 3.....3

Punto 4.....3

Punto 5.....3

Punto 6.....4

Punto 7.....4

Punto 8.....4

Notas previas a su corrección:

- He creado un directorio que contiene los recursos necesarios para esta actividad **./recursos/**, para que no se mezclan con el resto de ficheros.

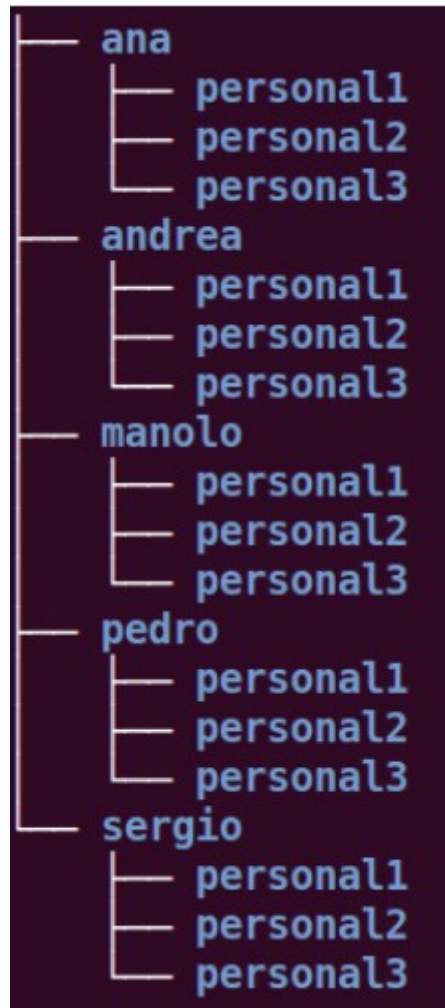
Enunciado general:

Resuelve cada uno de los siguientes problemas en un shell script. Échale un ojo al siguiente vídeo de aclaración.

Punto 1.

Enunciado:

Escribe un Shell script que cree una estructura de directorios similar a la siguiente para los nombres que se encuentren en el fichero nombres.txt, y teniendo en cuenta un número X de subdirectorios personal que se deben crear (para el caso de la imagen, X=3). El valor de la X será introducido por el usuario como parámetro a la llamada del script. Por ejemplo, si queremos crear 3 subdirectorios realizaríamos la siguiente llamada: ./script.sh 3



Propuesta:

Para empezar el script comprueba que el usuario a ingresado un argumento, si no es así finaliza. Tras ello crea una carpeta que contendrá la estructura de directorios solicitada en el ejercicio.

A través de dos bucles for genero la estructura accediendo al fichero **nombres.txt** guardado en otro directorio. También cuida de entrar y salir de los directorios en vez de crearlos con rutas relativas del tipo **\$nombre/directorio-\$numero**.

Por último una vez terminados los bucles muestra la estructura ejecutando el comando tree.

Script:

Punto 2.

Enunciado:

En el fichero precipitaciones.txt hay una serie de líneas que indican el día y la cantidad de litros que ha llovido. En este ejercicio deberás crear un script que muestre por pantalla la media de precipitaciones tal y como se ve en la siguiente ejecución:

```
juanmi@sobremesa:~/practica_scripts/examen$ ./script4.sh
La media de precipitaciones es 6
juanmi@sobremesa:~/practica_scripts/examen$
```

Propuesta:

El script recorre el fichero mencionado en el enunciado recogiendo los datos, pero en esta caso se usan un while. En la línea 24 he usado el comando bc con el parámetro `scale=2` para mostrar dos decimales.

Script:

Script-2

Punto 3.

Enunciado:

Escribe un script que muestre por pantalla el listado de días que no llovió, asumiendo que el primero es lunes.

Propuesta:

Este punto es muy parecido al anterior, pero en este caso al iterar por las líneas se comprueba que la cantidad es igual a 0 y se recoge el día. Este se utiliza más adelante para obtener el index del array dias, el cual contiene los días de la semana.

He usado algunos comando complejos para trabajar con el array, pero he intentado comentar el script lo mejor posible.

Script:

Script-3

Punto 4.

Enunciado:

Escribe un script que muestre un listado de los números pares que hay en el fichero numeros.txt así como un listado de los números impares. Además, deberá mostrar el total de cada tipo.

Propuesta:

Aquí he usado un for para recorrer el fichero `numeros.txt`. Por cada iteración se comprueba al valor de `$linea` para comprobar si el resto de su división entre 2 es 0. Dependiendo de ello se asignará el número a una lista y se incrementa el contador corresponde.

Después se muestra la información en pantalla con algo de formato para que se visualice mejor.

Script:

Script-4

Punto 5.

Enunciado:

Escribe un script que procese el fichero listado.txt. Cada línea de este fichero tiene un usuario, un sistema operativo y un valor numérico que indica el número de procesos que tiene en ejecución ese usuario en este sistema. Lo que se pretende que haga el script es que, recorra las líneas del fichero y calcule cuántos usuarios tienen Linux, cuántos Windows y el total de procesos de cada sistema. El resultado del script debería ser algo como:

```
Linux-> 10 158
Windows-> 9 243
juanmi@sobremesa:~/practica_scripts$
```

Propuesta:

En la declaración de variables, he creado dos de ellas temporales para que quede sea más legible el flujo de ejecución.

Script:

Script-5

Punto 6.

Enunciado:

Escribe un script que simule un comando de borrado de ficheros .txt. Para ello, el script tiene que recibir por parámetro un directorio y tiene que borrar todos los ficheros con extensión .txt que tenga ese directorio, mostrando al final, un mensaje con la cantidad de ficheros que se han borrado:

```
Se han borrado 10 ficheros
juanmi@sobremesa:~/practica_scripts$
```

Propuesta:

Para este ejercicio me he complicado un poco creando un sistema capaz de generar una estructura de ficheros de forma virtual, para no tener que crearlos cada vez que hago alguna prueba del script.

Por el resto he usado la variable del sistema `RANDOM` que genera un número cada vez que es llamada. También para comprobar la extensión de los ficheros uso `${i##*.}` que muestra todo aquello más allá del último punto.

Script:

Script-6

Punto 7.

Enunciado:

Partiendo del ejercicio anterior, antes de realizar el borrado, se deberá comprobar si existe el directorio antes de borrar, mostrando el mensaje "El directorio no existe" en caso que el directorio no exista. Hay diversas formas de saber si existe un directorio, una de ellas puede ser mediante el uso del comando pwd, que nos puede mostrar la ruta antes y después de intentar posicionarnos en el directorio.

Propuesta:

Sin darme cuenta en el ejercicio anterior he integrado esta funcionalidad, por lo que simplemente es una copia de este.

Script:

Script-7

Punto 8.

Enunciado:

Escribe un script que cree un listado de todas las direcciones IP que están conectadas en nuestra red local. Para ello, debemos utilizar el comando ping que nos devuelve cuántos paquetes han sido enviados correctamente. Podemos enviar un número concreto de paquetes mediante alguna opción de este comando, tal y como ya vimos en actividades de redes (utilizar el man si no lo recuerdas). El resultado de un ping a una máquina que está conectada sería el que se ve en la siguiente imagen. Como puedes ver, en este caso se ha enviado un paquete y se ha recibido correctamente (1 received), mientras que si no se recibe nada, el mensaje sería diferente:

```
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.  
64 bytes from 10.0.2.15: icmp_req=1 ttl=64 time=1.32 ms  
  
--- 10.0.2.15 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 1.322/1.322/1.322/0.000 ms  
juanmi@sobremesa:~/practica_scripts/script1$
```

Propuesta:

He utilizado la siguiente línea de ejecución que a través de tuberías obtiene los primeros dígitos de red:

```
•ip a show wlo1 | grep 'inet ' | awk '{print $2}' | sed 's/\([/*^]\)/\&/' | awk  
'{split($1,p,"."); $1=p[1]".p[2]".p[3]"; print $1}'
```

El resto de la ejecución es sencilla.

Script:

Script-8
