

Scripts en bash

1. Uso de variables en shell-script

```
#Asignación:
nombre_variable=3
#Acceso a su contenido:
$nombre_variable
#Usamos el comando read para leer variables de teclado y el comando echo
para mostrar por pantalla
read nombre apellido1 apellido2 #esto lo leerà separado por espacios
echo Los valores leídos por teclado son: $nombre $apellido1 $apellido2
```

2. Variables especiales

```
$0 #nombre del script que se está ejecutando
$n #el valor del argumento pasado como posición n
 $# #número de argumentos
 $* #lista de todos los argumentos
 $$ #PID del proceso que se está ejecutando
 $! #PID del último proceso ejecutado
 $? #salida del último proceso ejecutado
```

3. Estructuras de ejecución

3.1. Estructura IF

```
if condicion1
then
    comandos si condicion1 se cumple
elif condicion2
    comandos si condicion2 se cumple
else
    comandos si condicion1 y condicion2 no se cumplen
fi
```

Ejemplo:

```
if [ $# = 1 ] #muy importante respetar los espacios entre los corchetes
```

```
then
    echo Cadenas iguales
fi
```

4.3. Enteros

```
-eq # iguales
-ne # distintos
-gt # dato 1 mayor que dato 2
-ge # dato 1 mayor o igual que dato 2
-lt # dato 1 menor que dato 2
-le # dato 1 menor o igual que dato2
```

Ejemplo:

```
#-p para insertar un texto en la lectura del input
read -p 'Primer número:' a
echo Primer número: $a
read -p 'Segundo número:' b
echo Segundo número: $b

if [ $a -lt 20 ]
then
    echo Primer número inferior a 20
fi

if [ $b -ge 30 ]
then
    echo Segundo número mayor o igual a 30
fi

if [ $a = $b ]
then
    echo Números iguales
fi
```

5. Otros comandos útiles

```
exit #permite salir del script de forma inmediata, se puede añadir una salida
break #permite salir de un bucle, break [n], donde n es el número de bucles de los que saldríamos si hay varios anidados
continue #salta iteraciones de un bucle, continue n, hace saltar n
```

7.7 EJ7

Crear un shell script en bash que permita crear de forma automática usuarios en Linux. Para ello, el script:

- Recibe como parámetro el nombre de un archivo. Si el número de parámetros es incorrecto, o si el parámetro no corresponde al nombre de un archivo, se mostrará en mensaje de error.
- El archivo tendrá una línea para cada grupo, con el nombre del grupo y el número de alumnos separados por el carácter dos puntos (:).
grupo1:numero
grupo2:numero
- El script va recorriendo las líneas del archivo y:
 - Crea el grupo. Al crear el grupo mostrará "Se ha creado el grupo..." y si ya existe el grupo, no mostrará nada.
 - Crea los usuarios con el nombre grupo1-1, grupo1-2, etc. hasta el número de usuarios del grupo y como grupo principal el grupo correspondiente. Para cada usuario creado correctamente, mostrará el mensaje "Se ha creado el usuario ...". Por si estos usuarios existiesen los eliminará previamente al crearlos junto con su directorio personal.

```
#!/bin/bash
# Comprobar parámetros
if [ "$#" -ne 1 ]
then
    echo "Numero de parametros incorrecto"
    exit
fi
# Si el parámetro no se corresponde con un fichero
if [ ! -f $1 ]
then
    echo "El parametro no se corresponde con un fichero"
    exit
fi
for linea in `cat $1`
do
    grupo=`echo $linea | cut -d':' -f1`
    numero=`echo $linea | cut -d':' -f2`
    # Comprobar si el comando es exitoso y descartar el error estándar
    if groupadd $grupo 2> /dev/null
    then
```