UD6.PRACTICA3

EJERCICIO 1. Crea un script para bash que reciba N parámetros. Debe mostrarlos con el siguiente formato:

PARAMETRO 1. VALOR: XXX (donde XXX es el valor del parámetro 1)
PARAMETRO 2. VALOR: XXX (donde XXX es el valor del parámetro 2)
PARAMETRO 3. VALOR: XXX (donde XXX es el valor del parámetro 3)
...
PARAMETRO N. VALOR: XXX (donde XXX es el valor del parámetro N)

EJERCICIO 2. Crea un script para bash que reciba una cadena de texto como parámetro. El programa debe validar que el parámetro recibido no es vació (en caso de que sea vacío mensaje de error y fin del script). A continuación nos pedirá cadenas de texto por pantalla hasta que acertemos la que pasamos como parámetro. En el momento que acertemos, el programa nos indicará que hemos acertado, cuantos intentos hemos necesitado y finalizará. Si en algún momento introducimos una cadena vacía, deberá indicárnoslo y pedirnos otra cadena de nuevo para continuar con el juego (una cadena vacía no se considerara como un intento de acierto).

Abrir

ejercicio2.sh
~/scripts/EJERCICIO2

```
1
2 #!/bin/bash
3
4 contador=0
5 if  [ -n "$1" ]
6 then
7         while [ "$1" != "$cadena" ]
8         do
9         echo Introduce palabras a adivinar:
10        read cadena
11                if [ -n "$cadena" ]
12                then
13                        let contador++
14                else
15                        echo Has introducido una cadena vacia
16                fi
17        done
18 else
19        echo La cadena esta vacia
20        exit
21 fi
22
23 echo Has acertado, has necesitado $contador intentos
24
```

EJERCICIO 3. Crear un script para bash que esté preparado para recibir 3 parámetros. En caso de que se invoque con más o con menos parámetros, se mostrará un mensaje de error al respecto y el programa finalizará. En caso de que el número de parámetros sea correcto se deberá validar que se cumplen las siguientes condiciones:

- El valor del primer parámetro es /a
- El valor del segundo parámetro es /b
- El valor del tercer parámetro es ABC
-

En caso de que no se cumpla alguna de las condiciones anteriores, se mostrará un mensaje de error y el programa finalizará.

Abrir

ejercicio3.sh
~/scripts/EJERCICIO3

```
1 #!/bin/bash
2
3 if [ $# -ne 3 ]
4 then
5         echo El numero de parametros debe ser 3
6         exit
7 fi
8
9 if [ $1 != /a ]
10 then
11         echo El primer parametro debe ser /a
12         exit
13 elif [ $2 != /b ]
14 then
15         echo El segundo parametro debe ser /b
16         exit
17 elif [ $3 != ABC ]
18 then
19         echo El tercer parametro debe ser ABC
20         exit
21 fi
22
23 echo Fin del programa
24
```

EJERCICIO 4. Crear un script para bash que este preparado para recibir 3 parámetros. En caso de que se invoque con más o con menos parámetros, se mostrará un mensaje de error al respecto y el programa finalizará. En caso de que el número de parámetros sea correcto se deberán realizar las siguientes acciones en función al valor del tercer parámetro:

- Si el tercer parámetro vale COPIAR, se copiará el fichero indicado como primer parámetro al directorio indicado como segundo parámetro. Antes de realizar la copia se deberá comprobar que el fichero y el directorio existen. Si no existieran mensaje de error y fin de programa.
- Si el tercer parámetro vale MOVER, se moverá el fichero indicado como primer parámetro al directorio indicado como segundo parámetro. Antes de mover el archivo se deberá comprobar que el fichero y el directorio

UD6.PRACTICA3

existen. Si no existieran mensaje de error y fin de programa.
- Si el tercer parámetro tiene un valor diferente a COPIAR o a MOVER se mostrará un mensaje de error indicando que la sintaxis no es correcta.

Abrir — ejercicio4.sh — ~/scripts/EJERCICIO4

```
1 #!/bin/bash
2
3 if [ $# -ne 3 ]
4 then
5         echo El numero de parametros debe ser 3
6         exit
7 fi
8
9 if [ "$3" = "COPIAR" ]
10 then
11         if [ -e "$1" -a -d "$2" ]
12         then
13                 cp $1 $2
14                 echo Fichero copiado
15         else
16                 echo El fichero o el directorio no existen
17                 exit
18         fi
19 elif [ "$3" = "MOVER" ]
20 then
21         if [ -e "$1" -a -d "$2" ]
22         then
23                 mv $1 $2
24                 echo Fichero movido
25         else
26                 echo El fichero o el directorio no existen
27                 exit
28         fi
29 else
30         echo La sintaxis no es correcta
31         exit
32
33 fi
34
35 echo Fin del programa
```

EJERCICIO 5. Desarrolla un script para bash que nos permita crear usuarios y grupos en el sistema. Para ello mostrará indefinidamente el siguiente menú:

1.- Crear usuario
2.- Crear grupo
3.- Finalizar.

- La opción 1 nos permitirá crear un usuario. Para ello pedirá el nombre del usuario a crear y su contraseña. Con la información suministrada creará un usuario en el sistema.
- La opción 2 nos permitirá crear un grupo. Para ello pedirá el nombre del grupo a crear. Con la información suministrada creará un grupo en el sistema.
- La opción 3 finalizará la ejecución del script.

Abrir

ejercicio5.sh
~/scripts/EJERCICIO5

```
1 #!/bin/bash
2
3 opcion=0
4 while [ $opcion != 3 ]
5 do
6         echo Escoger una opcion:
7         echo 1.- Crear usuario
8         echo 2.- Crear grupo
9         echo 3.- Finalizar
10
11        read opcion
12
13 if [ $opcion = 1 ]
14 then
15        echo Nombre de usuario:
16        read usuario
17        echo Contraseña:
18        read passwd
19
20        sudo useradd -p `openssl passwd -6 $passwd` $usuario
21
22        echo Usuario creado con éxito
23
24 elif [ $opcion = 2 ]
25 then
26        echo Nombre de grupo:
27        read grupo
28
29        sudo groupadd $grupo
30
31        echo Grupo creado con éxito
32 fi
33
34 done
35
```

EJERCICIO 6. Desarrolla un script para bash que nos pida una fecha (número de día, número de mes y número de año). Deberá indicarnos si la fecha indicada es o no es válida. (La validación debe contemplar los años bisiestos)

Abrir | *ejercicio6.sh ~/scripts/EJERCICIO6 | Guardar

```
1
2 echo Introduzca dia:
3 read dia
4
5 if [ -z $dia ]
6 then
7         echo Debe introducir dia
8         exit
9 fi
10 echo Introduzca mes:
11 read mes
12
13 if [ -z $mes ]
14 then
15         echo Debe introducir mes
16         exit
17 fi
18 echo Introduzca año:
19 read anho
20
21 if [ -z $anho ]
22 then
23         echo Debe introducir año
24         exit
25 fi
26
27
28 if [ $dia -lt 1 -o $dia -gt 31 ]
29 then
30         echo Fecha invalida
31         exit
32 elif [ $mes -lt 1 -o $mes -gt 12 ]
33 then
34         echo Fecha invalida
35         exit
36 #Doy a escoger como año valido entre el 1 y el 9999
37 elif [ $anho -lt 1 -o $anho -gt 9999 ]
38 then
39         echo Fecha invalida
40         exit
41 #Los meses que tengan 30 dias solo pueden ser Abril, Junio, Septiembre o Noviembre
42 elif [ $dia -eq 30 ]
43 then
44         if [ $mes -eq 4 -o $mes -eq 6 -o $mes -eq 9 -o $mes -eq 11 ]
45         then
46                 echo Fecha valida
47                 exit
48         else
```

sh | Anchura del tabulador: 8 | Ln 67, Col 1

EJERCICIO 7. Desarrolla un script para bash que reciba 2 enteros. El primero debe de ser menor que el segundo. En caso de no cumplirse al condición anterior el script mostrará un mensaje de error y finalizará. En caso de que los parámetros sean correctos se deben mostrar por pantalla todos los múltiplos de 3 que hay entre el primer parámetro y el segundo.

Abrir

ejercicio7.sh
~/scripts/EJERCICIO7

```
1
2
3 echo Introduce un numero:
4 read numero1
5 echo Introduce otro numero:
6 read numero2
7
8 if [ $numero1 -ge $numero2 ]
9 then
10         echo El primer numero debe ser menor que el segundo
11         exit
12 fi
13
14 for (( i=numero1; i<=numero2;i++ ))
15 do
16         let resto=i%3
17
18          if [ $resto = 0 ]
19          then
20                  echo $i
21
22         fi
23
24 done
```