

1) Realizar un script que, cada vez que se ejecute, muestre por pantalla (usando el comando echo):

- a) El PID del proceso que ejecuta el script.
- b) El número de argumentos con que se ejecuta el script.
- c) Los argumentos que tiene el script.
- d) Cuál es el primer argumento.
- e) El argumento 0.
- f) Que visualice cuál es el primer argumento, una vez que haya desplazado todos los argumentos una posición hacia la izquierda.
- g) El directorio de inicio del usuario actual.
- h) La lista de directorios de búsqueda.
- i) El directorio de trabajo actual.

```
#!/bin/bash

echo "a)PID: $$"
echo "b)Num argumentos: $#"
```

```
echo "c)Argumentos: $@"
echo "d)Primer argumento: $1"
echo "e)Argumento 0: $0"
#f)
shift 1
echo "f)Primer argumento, una vex que haya desplazado todos los argumentos
una posicion hacia la izquierda $1"
echo "g)Directorio de inicio del usuario actual $HOME"
echo "h)Lista de directorios de búsqueda: $PATH"
echo "i)Directorio de trabajo actual: $PWD"
```

2) Realizar un script que muestre el listado largo de un directorio que le pasaremos como parámetro e indique con un mensaje si ha realizado o no dicha tarea.

```
#!/bin/bash

if [ -d "$1" ] ; then
    echo "`ls -l $1`";
    echo "se ha realizado con exito";
else
    echo "no se ha realizado con exito";
fi

~
```

```
ivan@ivan-VirtualBox:~$ vim ej2
ivan@ivan-VirtualBox:~$ bash ej2 dir1
total 0
se ha realizado con exito
```

3) Escribir un script, el cual según el número de argumentos que se le pasan ha de hacer lo siguiente:

```
ivan@ivan-VirtualBox:~$ vim ej3
```

```
#!/bin/bash
if [ $# -lt 1 ]; then
    echo "Sintaxis: nombre_script arg1 arg2 ...";
elif [ $# -eq 1 ]; then
    echo "$1";
elif [ $# -eq 2 ]; then
    if [ "$1" -eq "$2" ]; then
        echo "$1 es igual que $2";
    elif [ "$1" -lt "$2" ]; then
        echo "$1 es menor que $2";
    else
        echo "$1 es mayor a $2";
    fi;
elif [ "$#" -eq 4 ] || [ "$#" -eq 3 ]; then
    echo "3 o 4 argumentos";
else
    echo "más de 4 argumentos";
fi
```

a) Si no se le pasa ningún argumento mostrará en pantalla el siguiente mensaje:

Sintaxis: nombre\_script arg1 arg2 ...

```
ivan@ivan-VirtualBox:~$ bash ej3
Sintaxis: nombre_script arg1 arg2 ...
```

b) Si se le pasa un argumento mostrará en pantalla el mismo argumento.

```
ivan@ivan-VirtualBox:~$ bash ej3 1
1
```

c) Si se le pasan 2 argumentos deberá mostrar en pantalla el resultado de su comparación mediante alguno de los siguientes mensajes:

arg1 es igual a arg2 o arg1 y arg2 son diferentes

```
ivan@ivan-VirtualBox:~$ bash ej3 1 2
1 es menor que 2
ivan@ivan-VirtualBox:~$ bash ej3 2 1
2 es mayor a 1
```

d) Si se le pasan 3 o 4 argumentos mostrará en pantalla el mensaje:

3 ó 4 argumentos

```
ivan@ivan-VirtualBox:~$ bash ej3 2 1 2
3 o4 argumentos
ivan@ivan-VirtualBox:~$ bash ej3 2 1 2 4
3 o4 argumentos
```

4) Sin usar la orden if, sino usando los operadores && y ||, escribir un script, que compruebe si existe un fichero (que no es un directorio) de nombre el primer argumento que se le pasa al script. Si el fichero existe debe chequear si se tiene permiso de lectura. Si se tiene dicho permiso se debe devolver el mensaje: tiene permiso de lectura, en caso contrario se devolverá el mensaje: no tiene permiso de lectura. Si el fichero no existe ha de devolver el mensaje no existe.

```
ivan@ivan-VirtualBox:~$ vim ejer4
```

```
[ -f $1 ] && (echo "es un fichero" && [ -r $1 ] && echo "tiene permiso de lectura" || echo "no tiene permiso de lectura" ) || echo "no existe"
```

```
ivan@ivan-VirtualBox:~$ bash ejer4 fichero1
es un fichero
tiene permiso de lectura
ivan@ivan-VirtualBox:~$ bash ejer4 dir1
no existe
```

5) Realizar un script que muestre en pantalla el siguiente menú:

Menu

- 1) Visualizar un listado largo del directorio de trabajo
- 2) Visualizar el directorio de trabajo
- 3) Salir

Pulse una Opcion:

```
ivan@ivan-VirtualBox:~$ vim ej5
```

```
#!/bin/bash

clear
echo "Menu"
echo "1) Visualizar un listado largo del directorio de trabajo."
echo "2) Visualizar el directorio de trabajo."
echo "3) Salir."

read opcion

case $opcion in
1)
echo `ls -l`
;;Ayuda
2)
pwd
;;
3)
echo "Fin del programa"
exit 0;;
*) echo "Opción errónea"
esac
```

```
ivan@ivan-VirtualBox:~$ bash ej5
```

```
Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
1
total 56 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Descargas drwxrwxr-x 2 ivan i
van 4096 feb 24 21:02 dir1 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Documentos
-rw-rw-r-- 1 ivan ivan 136 feb 25 22:21 ej2 -rw-rw-r-- 1 ivan ivan 401 feb 25 2
2:33 ej3 -rw-rw-r-- 1 ivan ivan 289 feb 25 22:53 ej5 -rw-rw-r-- 1 ivan ivan 143
feb 25 22:42 ejer4 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Escritorio -rw-rw-
r-- 1 ivan ivan 0 feb 25 22:41 fichero -rw-rw-r-- 1 ivan ivan 17 feb 25 22:41 f
ichero1 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Imágenes drwxr-xr-x 2 ivan iva
n 4096 feb 24 14:55 Música drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Plantillas
drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Público drwxr-xr-x 2 ivan ivan 4096 fe
b 24 14:55 Vídeos
```

```
Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
2
/home/ivan
```

```
Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
3
Fin del programa
```

```
Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
4
Opción errónea
```

6) Realizar un script que pida por pantalla al usuario confirmación de su deseo de salir. Si el usuario pulsa "s", "S", "SI" o "si" el script debe borrar la pantalla y terminar. Si lo que pulsa es "n", "N", "no" o "NO" el script debe mostrar por pantalla lo que el usuario ha introducido y finalizar con la ejecución. En cualquier otro caso el script indicará que lo que ha pulsado no es correcto y también terminará con la ejecución del script

```
#!/bin/bash

echo "Desea salir?"
read opcion

case $opcion in
si | SI | s | S)
clear
;;
no | NO | N | n)
echo "$opcion"
;;
*) echo "Opción errónea"
esac
```

```
ivan@ivan-VirtualBox:~$ bash ej6
Desea salir?
n
n
ivan@ivan-VirtualBox:~$ bash ej6
Desea salir?
N
N
ivan@ivan-VirtualBox:~$ bash ej6
Desea salir?
NO
NO
ivan@ivan-VirtualBox:~$ bash ej6
Desea salir?
no
no
ivan@ivan-VirtualBox:~$ bash ej6
Desea salir?
hola
Opción errónea
```

7) Escribir un script que calcule la suma y el producto de los dos argumentos que se deben proporcionar al script.

```
ivan@ivan-VirtualBox:~$ vim ej7
```

```
#!/bin/bash

if [ $# -eq 2 ]; then
    suma=$(( $1 * $2 ))
    echo "suma:" $suma;
    mult=$(( $1 * $2 ))
    echo "multiplicación:" $mult;
elif [ $# -lt 2 ]; then
    echo "has escrito argumentos de más (solo 2)";
else
    echo "has escrito más de 2 argumentos (solo 2)";
fi
```

```
ivan@ivan-VirtualBox:~$ bash ej7 3 5
suma: 15
multiplicación: 15
ivan@ivan-VirtualBox:~$ bash ej7 3 5 5
has escrito más de 2 argumentos (solo 2)
ivan@ivan-VirtualBox:~$ bash ej7 3
has escrito argumentos de más (solo 2)
```

8) Modificar el script del ejercicio 5 de forma que se muestre el menú en pantalla hasta que el usuario pulse la opción 3 (Salir). Además, el script debe permitir que el usuario también pueda introducir las distintas opciones del menú mediante literales (uno, dos, .., etc).

```
while :
do

clear
echo "Menu"
echo "1) Visualizar un listado largo del directorio de trabajo."
echo "2) Visualizar el directorio de trabajo."
echo "3) Salir."

read opcion

case $opcion in
1 | uno)
echo `ls -l`
read foo
;;
2 | dos)
pwd
read foo
;;
3 | tres)
echo "Fin del programa"
exit 0;;
*) echo "Opción errónea"
sleep 1
esac
done
```

```

Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
1
total 64 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Descargas drwxrwxr-x 2 ivan i
van 4096 feb 24 21:02 dir1 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Documentos
-rw-rw-r-- 1 ivan ivan 136 feb 25 22:21 ej2 -rw-rw-r-- 1 ivan ivan 401 feb 25 2
2:33 ej3 -rw-rw-r-- 1 ivan ivan 351 feb 25 23:16 ej5 -rw-rw-r-- 1 ivan ivan 156
feb 25 22:59 ej6 -rw-rw-r-- 1 ivan ivan 259 feb 25 23:07 ej7 -rw-rw-r-- 1 ivan
ivan 143 feb 25 22:42 ejer4 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Escritori
o -rw-rw-r-- 1 ivan ivan 0 feb 25 22:41 fichero -rw-rw-r-- 1 ivan ivan 17 feb 2
5 22:41 fichero1 drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Imágenes drwxr-xr-x 2
ivan ivan 4096 feb 24 14:55 Música drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Pl
antillas drwxr-xr-x 2 ivan ivan 4096 feb 24 14:55 Público drwxr-xr-x 2 ivan iva
n 4096 feb 24 14:55 Videos

```

```

Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
2
/home/ivan

```

```

Menu
1) Visualizar un listado largo del directorio de trabajo.
2) Visualizar el directorio de trabajo.
3) Salir.
3
Fin del programa

```

9) Escribir tres scripts: a) Un script que devuelva los argumentos del script pero de la siguiente manera: 1a línea - todos los argumentos 2a línea - todos los argumentos menos el 1 3a línea - todos los argumentos menos el 1 2 4a línea - todos los argumentos menos el 1 2 3 . última línea - el último argumento.

```

ivan@ivan-VirtualBox:~$ vim ej9

```

```

#!/bin/bash

echo $@
a=$#
for (( i=0; i<=$a; i++))
do
    shift 1
    echo $@
done

```

```
ivan@ivan-VirtualBox:~$ bash ej9 a b c d e f
a b c d e f
b c d e f
c d e f
d e f
e f
f
```

b) Un script que devuelva por pantalla todos los números que hay entre 0 y 99, ambos inclusive.

```
#!/bin/bash

for (( i=0; i<100; i++ ))
do
    echo "$i"
done
```

```
ivan@ivan-VirtualBox:~$ bash ej9bb
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
```

c) Y por último otro que calcule la suma y el producto de los argumentos proporcionados al script.  
El igual al 7.



10) Escribir dos scripts llamados intrountil e introwhile. Ambos van a realizar lo mismo, pedir una cadena, borrar la pantalla y mostrar el mensaje: La cadena introducida es cadena. De nuevo volverá a pedir otra cadena y harán lo mismo que antes. Esto se repetirá indefinidamente hasta que se introduzca la cadena "fin" o "end".

```
#!/bin/bash

echo "escribe una cadena:"
read cad
clear

until [ $cad = "end" ] || [ $cad = "fin" ]
do
echo "la cadena introducida es: $cad"

echo "escribe una cadena:"
read cad
clear
done
```

```
ivan@ivan-VirtualBox:~$ bash ej10
```

```
la cadena introducida es: hola
escribe una cadena:
█
```

Con while:

```
#!/bin/bash

echo "escribe una cadena:"
read cad
clear

while [ $cad = "end" ] && [ $cad = "fin" ]
do
echo "la cadena introducida es: $cad"

echo "escribe una cadena:"
read cad
clear
done
```

```
la cadena introducida es: hola
escribe una cadena:
█
```

