

- 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 vez que haya desplazado todos los argumentos una posición 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"
```

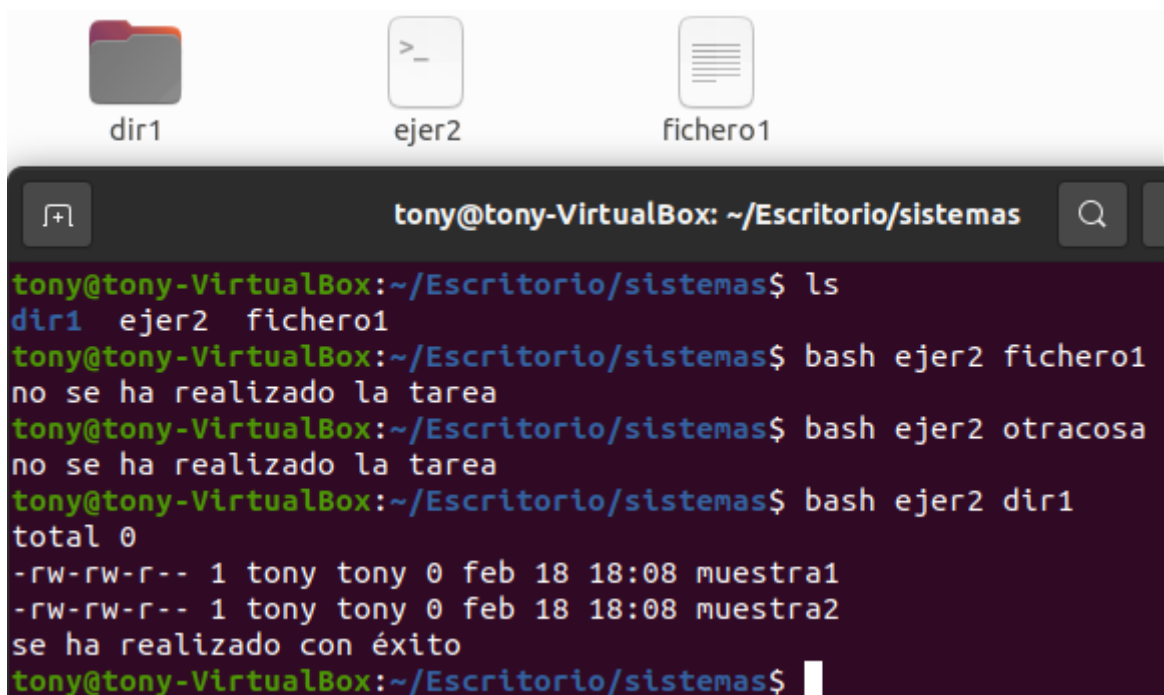
```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer1
a)PID: 3446
b)Num argumentos: 0
c)Argumentos:
d)Primer argumento:
e)Argumento 0: ejer1
f)Primer argumento, una vez que haya desplazado todos los argumentos una posición hacia la izquierda
g)Directorio de inicio del usuario actual /home/tony
h)Lista de directorios de búsqueda: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
i)Directorio de trabajo actual: /home/tony/Escritorio/sistemas
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

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.

Código:

```
#!/bin/bash

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



The screenshot shows a desktop environment with three icons: 'dir1' (a folder), 'ejer2' (a terminal icon), and 'fichero1' (a document icon). Below the desktop is a terminal window titled 'tony@tony-VirtualBox: ~/Escritorio/sistemas'. The terminal shows the following commands and output:

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ ls
dir1  ejer2  fichero1
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer2 fichero1
no se ha realizado la tarea
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer2 otracosa
no se ha realizado la tarea
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer2 dir1
total 0
-rw-rw-r-- 1 tony tony 0 feb 18 18:08 muestra1
-rw-rw-r-- 1 tony tony 0 feb 18 18:08 muestra2
se ha realizado con éxito
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

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

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

Sintaxis: nombre\_script arg1 arg2 ...

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

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

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

3 ó 4 argumentos

Código:

```
#!/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";
fi
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 1 2
1 es menor que 2
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 2 1
2 es mayor a 1
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 2 2
2 es igual que 2
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 1
1
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 dani_guapo
dani_guapo
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 1 2 3
3 o 4 argumentos
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 1 2 3 4
3 o 4 argumentos
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer3 1 2 3 4 5
más de 4
```

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.

Código:

```
#!/bin/bash
```

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

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ ls -l
total 20
drwxrwxr-x 2 tony tony 4096 feb 18 18:08 dir1
-rw-rw-r-- 1 tony tony 141 feb 18 18:22 ejer2
-rw-rw-r-- 1 tony tony 396 feb 18 19:15 ejer3
-rw-rw-r-- 1 tony tony 158 feb 22 18:14 ejer4
-rw-rw-r-- 1 tony tony 232 feb 18 20:01 ejer5
-rwxrwxrwx 1 tony tony 0 feb 18 18:26 fichero1
---x--x--x 1 tony tony 0 feb 22 18:16 fichero2
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer4 fichero1
es un fichero
tiene permiso de lectura
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer4 fichero2
es un fichero
no tiene permiso de lectura
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer4 dir1
no existe
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer4 otra_cosa
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:

Si el usuario pulsa una opción inexistente, el programa debe mostrar el mensaje opción errónea. El menú sólo se mostrará una vez. Si el usuario selecciona la opción 3 (Salir) se mostrará el mensaje Fin del programa.

```
#!/bin/bash
```

Código:

```
clear
echo "MENU"
echo "1)Visualizar un listado largo del directorio de trabajo"
echo "2)Visulizar el directorio de trabajo"
echo "3)Salir"

read opcion

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

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
1
total 20 drwxrwxr-x 2 tony tony 4096 feb 18 18:08 dir1
tony 158 feb 22 18:14 ejer4 -rw-rw-r-- 1 tony tony 282 t
o2
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
2
/home/tony/Escritorio/sistemas
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
otro
Opción errónea
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
3
fin del programa
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

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
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer6
Desea salir?
no
no
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer6
Desea salir?
n
n
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer6
Desea salir?
N
N
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer6
Desea salir?
NO
NO
```

Con el si/SI/s/S no puedo poner capturas  
Porque lo borra xD

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer6
Desea salir?
otracosa
Opción errónea
```

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

```
#!/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
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer7 2 4
suma: 6
multiplicación: 8
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer7 2 4 3
has escrito más de 2 argumentos (solo 2)
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer7 2
has escrito argumentos de más (solo 2)
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

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).

```
#!/bin/bash
```

```
while :  
do
```

```
clear
```

```
echo "MENU"
```

```
echo "1)Visualizar un listado largo del directorio de trabajo"
```

```
echo "2)Visulizar 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)Visulizar el directorio de trabajo
3)Salir
1
total 32 drwxrwxr-x 2 tony tony 4096 feb 18 18:08 dir1 -
1 feb 18 18:22 ejer2 -rw-rw-r-- 1 tony tony 396 feb 18 1
tony tony 158 feb 22 18:14 ejer4 -rw-rw-r-- 1 tony tony
rw-rw-r-- 1 tony tony 236 feb 22 19:07 ejer6 -rw-rw-r--
9:29 ejer7 -rw-rw-r-- 1 tony tony 417 feb 22 20:20 ejer8
0 feb 18 18:26 fichero1 ---x--x--x 1 tony tony 0 feb 22
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
2
/home/tony/Escritorio/sistemas
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
3
fin del programa
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

```
MENU
1)Visualizar un listado largo del directorio de trabajo
2)Visulizar el directorio de trabajo
3)Salir
uno
total 32 drwxrwxr-x 2 tony tony 4096 feb 18 18:08 dir1 -rw-rw-r-- 1 tony tony 14
1 feb 18 18:22 ejer2 -rw-rw-r-- 1 tony tony 396 feb 18 19:15 ejer3 -rw-rw-r-- 1
tony tony 158 feb 22 18:14 ejer4 -rw-rw-r-- 1 tony tony 282 feb 22 18:48 ejer5 -
rw-rw-r-- 1 tony tony 236 feb 22 19:07 ejer6 -rw-rw-r-- 1 tony tony 262 feb 22 1
9:29 ejer7 -rw-rw-r-- 1 tony tony 417 feb 22 20:20 ejer8 -rwxrwxrwx 1 tony tony
0 feb 18 18:26 fichero1 ---x--x--x 1 tony tony 0 feb 22 18:16 fichero2
```

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.

```
#!/bin/bash

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

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer9a a b c d e f g
a b c d e f g
b c d e f g
c d e f g
d e f g
e f g
f g
g
```

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
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$ bash ejer9b
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
27
28
29
30
31
```

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

El ejercicio 9c es 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".

Con until:

```
#!/bin/bash
#con until

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
```

```
tony@tony-VirtualBox:~$
escribe una cadena:
algo
```

```
la cadena introducida es: algo
escribe una cadena:
alomas
```

```
la cadena introducida es: alomas
escribe una cadena:
end
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

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
```

```
tony@tony-VirtualBox:~$
escribe una cadena:
algo
```

```
la cadena introducida es: algo
escribe una cadena:
alomas
```

```
la cadena introducida es: alomas
escribe una cadena:
fin
```

```
tony@tony-VirtualBox:~/Escritorio/sistemas$
```

### Ej1:

```
#!/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 vez que haya desplazado todos los argumentos una posición 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"

### Ej2:

```
#!/bin/bash

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

### 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";
fi
```

#### Ej4:

```
#!/bin/bash
```

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

#### Ej5:

```
#!/bin/bash
```

```
clear  
echo "MENU"  
echo "1)Visualizar un listado largo del directorio de trabajo"  
echo "2)Visulizar el directorio de trabajo"  
echo "3)Salir"
```

```
read opcion
```

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

#### Ej6:

```
#!/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
```

### 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
```

### Ej8:

```
#!/bin/bash

while :
do

clear
echo "MENU"
echo "1)Visualizar un listado largo del directorio de trabajo"
echo "2)Visulizar 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
```

### Ej9a:

```
#!/bin/bash

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

### Ej9b:

```
#!/bin/bash

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

### Ej9c=Ej7

### Ej10a:

```
#!/bin/bash
#con until

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
```

### Ej10b:

```
#!/bin/bash  
# con while
```

```
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
```