

# Ejercicios VII – Tema 5

## Ejercicio 32 – Programación shell

Realiza scripts de shell con las tareas que se indican a continuación y nombre ejerX.sh.

1. Borrará la pantalla y mostrará cuál es nuestro usuario y nuestra ubicación.

```
#!/bin/bash
clear
echo $LOGNAME
echo $PWD
```

2. Borrará la pantalla, mostrará el contenido del directorio "/" en formato largo y, a continuación, nuestro directorio personal en formato largo con los archivos ocultos.

```
#!/bin/bash
clear
ls -l /
ls -la /home/ciclost
```

3. Se pedirá al usuario a través de la línea de comandos que escriba un mes y un año para, a continuación, mostrar el calendario de ese mes y ese año.

```
#!/bin/bash
echo -n "Di el número de un mes del año: "
read mes
echo -n "Escribe un año: "
read anyo
cal $mes $anyo
```

4. Se pedirá al usuario a través de la línea de comandos el nombre de un fichero. Si el fichero existe, se mostrará un mensaje diciendo que existe y cuáles son sus permisos, su tamaño y su propietario. Si no existe, se mostrará un mensaje indicando que el fichero no existe.

```
#!/bin/bash
echo -n "Escribe el nombre de un fichero: "
read fichero
if [ -f $fichero ]; then
    echo "el fichero $fichero existe."
    ls -lh $fichero | cut -d " " -f 1,3,5
else
    echo "el fichero $fichero no existe."
fi
```

5. Se le pasará un nombre de fichero como parámetro a nuestro script y, después, si existe y se puede escribir en él, se editará con nano.

```
#!/bin/bash
if [ $# -ne 1 ]; then
    echo "Error, has de introducir un solo parámetro."
else
    if [ -f $1 ];then
        if [ -w $1 ]; then
            nano $1
        else
            echo "Error, el fichero $1 no tiene permisos de escritura."
        fi
    else
        echo "Error, el fichero $1 no existe."
    fi
fi
```

6. Se pedirá al usuario que escriba el nombre de un fichero. Si el fichero tiene activo los permisos de lectura, mostrará su contenido de forma paginada por pantalla.

```
#!/bin/bash
echo -n "Escribe el nombre de un fichero: "
read fichero
if [ -r $fichero ]; then
    more $fichero
else
    echo "Error, el fichero no existe."
fi
```

7. Se pedirá al usuario que escriba el nombre de un fichero, se comprobará que ese fichero existe, y si existe, pedirá que introduzcas el nuevo nombre nuevo que se le asignará. Si ese nombre nuevo no existe como fichero, se cambia el nombre del fichero. Después, se mostrará por pantalla un mensaje de texto donde diga que al fichero \$fich1 se le ha puesto el nombre \$fich2.

```
#!/bin/bash
echo -n "Escribe el nombre de un fichero. "
read fich
fich2=$fich
if [ -f $fich ]; then
    echo -n "Introduce un nuevo nombre para el fichero: "
    read nomb
    if [ -f $nomb ]; then
        echo "El nombre elegido existe"
    else mv $fich $nomb
        echo "El archivo $fich2 ahora se llama $nomb"
    fi
else echo "Ese archivo no existe"
fi
```

**8.** Se pedirá al usuario el nombre de un usuario. A continuación, se comprobará si el usuario existe en el sistema, se mostrará información sobre su uid y los grupos a los que pertenece ese usuario.

```
#!/bin/bash
echo -n "Di tu nombre de usuario: "
read usu
grep "^$usu:" /etc/passwd > /dev/null
if [ $? -eq 0 ]; then
    id $usu
else
    echo "El usuario no existe"
```

**9.** Se pasarán como parámetros al programa una serie de directorios que se irán creando. Se comprobará, para cada uno de ellos, que no exista previamente y, si existe, mostrará un mensaje de error.

```
#!/bin/bash
if [ $# -eq 0 ]; then
    echo "Error, debes introducir al menos un nombre de directorio."
else
    while [ $# -ne 0 ]; do
        if [ -d $1 ]; then
            echo "Error, el directorio $1 ya existe"
        else
            mkdir $1
            echo "El directorio $1 ha sido creado"
        fi
        shift
    done
fi
```

**10.** Es un ejercicio similar al anterior, pero se preguntará al usuario para que introduzca por teclado un nombre de directorio y, una vez creado, o después del mensaje de error, te pregunte si quieres crear otro. Se le responderá "s" o "n". Si es "s" se volverá a preguntar por otro directorio. Si es "n" terminará el script.

```
#!/bin/bash
resp1="s"
until [ $resp1 = "n" ]; do
    echo -n "Introduce un nombre para un directorio: "
    read nomb1
    if [ -d $nomb1 ]; then
        echo "Error, el directorio $nomb1 ya existe"
    else
        mkdir $nomb1
        echo "El directorio $nomb1 ha sido creado"
    fi
    echo -n "Quieres crear otro directorio? (s/n): "
    read resp1
done
```

11. Se mostrarán de forma paginada las variables del sistema. Después, mediante variables se mostrará nuestro subdirectorio personal, el shell que usamos y nuestro nombre de usuario.

```
#!/bin/bash
env | more
echo
echo "Directorio personal: $HOME"
echo "Shell empleado: $SHELL"
echo "Nombre de usuario: $USER"
```

## Ejercicio 33 – Programación shell II

Realiza scripts de shell con las tareas que se indican a continuación y nombre ejerX.sh.

1. Crea un script que muestre por pantalla el resultado de de las siguientes operaciones. Debes tener en cuenta que a, b y c son variables enteras que tienen asignados los valores: a = 5, b = 7, c = - 3.

1.  $a+b+c$
2.  $2 * b + 3 * (a-c)$
3.  $a/b$
4.  $a\%b$
5.  $a/c$
6.  $a\%c$
7.  $a*b/c$
8.  $a * (b / c)$
9.  $(a * c) \% b$
10.  $a * (c \% b)$

```
#!/bin/bash
varA=5 ; varB=7 ; varC=-3
echo "A = $varA B = $varB C = $varC"
let oper1=varA+varB+varC
echo "A+B+C = $oper1"
let oper2=2*varB+3*varA-varC
echo "2*B + 3*(A-C) = $oper2"
let oper3=varA/varB
echo "A/B = $oper3"
let oper4=varA%varB
echo $oper4
let oper5=varA/varC
echo $oper5
let oper6=varA%varC
echo $oper6
let oper7=varA*varB/varC
echo $oper7
let oper8=varA*varB%varC
echo $oper8
let oper9=varA*varC%varB
echo $oper9
let oper10=varA*varC%varB
echo $oper10
```

2. Modifica el script anterior para que pida al usuario el valor de las variables a, b y c y después calcule las operaciones y el valor de las variables introducidas.

```
#!/bin/bash
echo -n "Escribe un numero entero: "
read varA
echo -n "Escribe un numero entero: "
read varB
echo -n "Escribe un numero entero: "
read varC
echo
let oper1=($varA+$varB)+$varC
echo $oper1
echo
#let oper2=(2*$varB)+(3*($varA-$varC))
#echo $oper2
echo
let oper3=($varA/$varB)
echo $oper3
echo
let oper4=(varA%varB)
echo $oper4
echo
let oper5=($varA/$varC)
echo $oper5
echo
let oper6=($varA%$varB)
echo $oper6
echo
#let oper7=$varA*($varB/$varC)
#echo $oper7
echo
let oper8=($varA*varC)%varB
echo $oper8
echo
#let oper9=$varA*(varC%varB)
#echo $oper9
```

3. Crea un script que al ejecutarlo muestre por pantalla uno de estos mensajes: "Buenos días", "Buenas tardes" o "Buenas noches", en función de la hora que sea en el sistema (de 8:00 de la mañana a 15:00 será mañana, de 15:00 a 20:00 será tarde y el resto será noche).

```
#!/bin/bash
f1=$(date | cut -d " " -f5 | cut -d ":" -f1)
f2=$(date | cut -d " " -f5 | cut -d ":" -f2)
f3=$(date | cut -d " " -f5 | cut -d ":" -f3)
let hora=$f1*60
let minu=($hora+$f2)*60
let segtot=$hora+$minu+$f3
if [ $segtot -gt 28799 ] && [ $segtot -le 53999 ]; then
    echo "Buenos días"
elif [ $segtot -gt 53999 ] && [ $segtot -le 71999 ]; then
    echo "Buenas tardes"
else
    echo "Buenas noches"
fi
```

4. Crea un script que pida dos números al usuario y muestre el mayor de ellos por pantalla o, si son iguales, un mensaje que lo indique.

```
#!/bin/bash
echo -n "Introduce un número: "
read num1
echo "Introduce otro número: "
read num2
if [ $num1 -gt $num2 ]; then
    echo "$num1 es el mayor."
elif [ $num2 -gt $num1 ]; then
    echo "$num2 es mayor."
else
    echo "Ambos número son iguales."
fi
```

5. Crea un script que pida un número e indique si se trata de un número par o impar.

```
#!/bin/bash
echo -n "Introduce un número: "
read num1
let res=$num1%2
if [ $num1 -le 0 ]; then
    echo "Error, has de introducir un valor mayor de 0."
elif [ $res -eq 0 ]; then
    echo "$num1 es par"
else
    echo "$num1 es impar"
fi
```

**6.** Crea un script que pida al usuario tres valores enteros y muestre por pantalla el valor máximo y el valor mínimo de ellos o si existen valores iguales entre ellos.

```
#!/bin/bash
echo -n "Introduce el primer número: "
read numA
echo -n "Introduce el segundo número: "
read numB
echo -n "introduce el tercer número: "
read numC

if [ $numA -eq $numB ] && [ $numB -eq $numC ];then
    echo "Los tres números son iguales."
else
    if [ $numA -gt $numB ] && [ $numB -gt $numC ]; then
        echo "El número $numA es el valor máximo."
    elif [ $numB -gt $numA ] && [ $numB -gt $numC ]; then
        echo "El número $numB es el valor máximo."
    else echo "El número $numC es el valor máximo."
    fi
    if [ $numA -lt $numB ] && [ $numB -lt $numC ]; then
        echo "El número $numA es el valor mínimo."
    elif [ $numB -lt $numA ] && [ $numB -lt $numC ]; then
        echo "El número $numB es el valor mínimo."
    else echo "El número $numC es el valor mínimo."
    fi
fi
```

7. Crea un script que pida un número entero y determine si es múltiplo de 2 y de 5.

```
#!/bin/bash
echo -n "introduce un número entero: "
read num1
let divi2=num1%2
let divi5=num1%5
if [ $num1 -le 0 ]; then
    echo "Error, has de introducir un valor mayor de 0."
elif [ $divi2 -eq 0 ] && [ $divi5 -eq 0 ]; then
    echo "$num1 es múltiplo de 2 y 5."
else
    echo "$num1 no es múltiplo de 2 y 5."
fi
```

8. Crea un script que pida una nota e imprima por pantalla la calificación en formato "Apto" o "No Apto" según si la nota es mayor, menor o igual que 5.

```
echo -n "Indica tu nota: "
read nota1
if [ $nota1 -lt 0 ] || [ $nota1 -gt 10 ]; then
    echo "Error, la nota debe de ser un número entre 0 y 10."
else
    if [ $nota1 -lt 5 ]; then
        echo "No apto"
    else echo "Apto"
    fi
fi
```

9. Modifica el script anterior para que pida la nota de un examen (número entero entre 0 y 10) e imprima por pantalla la calificación en formato "Muy deficiente", si la nota es inferior a 3, "Suspenso", si la nota es menor que 5 y mayor que 3, "Aprobado" si está entre 5 inclusive y 7 sin incluir, "Notable" si está entre 7 inclusive y 9 sin incluir, "Sobresaliente" si está entre 9 inclusive y 10 sin incluir y "Matrícula de honor" si la nota es igual a 10.

```
echo -n "Indica tu nota: "
read nota1
if [ $nota1 -lt 0 ] || [ $nota1 -gt 10 ]; then
    echo "Error, la nota debe de ser un número entre 0 y 10."
else
    if [ $nota1 -lt 3 ]; then
        echo "Muy deficiente."
    elif [ $nota1 -ge 3 ] && [ $nota1 -lt 5 ]; then
        echo "Suspenso"
    elif [ $nota1 -ge 5 ] && [ $nota1 -lt 7 ]; then
        echo "Aprobado."
    elif [ $nota1 -ge 7 ] && [ $nota1 -lt 9 ]; then
        echo "Notable."
    elif [ $nota1 -ge 9 ] && [ $nota1 -lt 10 ]; then
        echo "Sobresaliente."
    else echo "Matrícula de honor."
    fi
fi
```

**10.** Crea un script para calcular el importe de una lista de productos. El usuario indicará al script el precio por unidad del artículo y el número de unidades (preguntará una vez por cada variable) y el programa sacará por pantalla el precio total. También preguntará si se desea salir después de cada artículo.

```
echo "Lista de la compra"
salir="s"
arti1=0
while [ $salir = "s" ]; do
    let arti1=arti1+1
    echo -n "Introduce el precio del artículo número $arti1: "
    read prec1
    echo -n "Introduce la cantidad del artículo $arti1: "
    read cant1
    let tota1=prec1*cant1
    let tota2=tota2+tota1
    echo -n "¿Quieres introducir más artículos? (s/n): "
    read salir
done
echo "En total tienes $arti1 articulo/os y cuestan $tota2 €"
```

**11.** Crea un script que calcule la nómina de un trabajador de tu empresa. El trabajador cobra un precio fijo por hora (que será preguntado al usuario) y se le retiene un 11% en concepto de IRPF. El programa debe pedir el nombre del trabajador, las horas trabajadas y el precio por hora. Como salida debe imprimir el sueldo bruto (sin descontar impuestos), la retención (lo que paga a Hacienda) y el sueldo neto (sueldo bruto menos los impuestos).

```
#!/bin/bash
echo "Calcular nómina"
echo -n "Nombre del trabajador: "
read nomb1
echo -n "Horas mensuales trabajadas: "
read hora1
echo -n "Precio en € por hora: "
read prec1
let brut1=hora1*prec1
let irpf1=(brut1*11)/100
let snet1=brut1-irpf1
echo "El sueldo bruto de $nomb1 es de $brut1 €"
echo "La retención del IRPF de $nomb1 es de $irpf1 €"
echo "$nomb1 cobra $snet1 € mensuales"
```

**12.** Crea un script que pida un número al usuario y saque por pantalla su tabla de multiplicar.

```
#!/bin/bash
echo -n "Introduce un número del 1 al 10: "
read tabl1
if [ $tabl1 -lt 1 ] || [ $tabl1 -gt 10 ]; then
    echo "Error, debes de introducir un número entero entre 1 y 10."
else
    echo "Tabla de multiplicar del $tabl1"
    for i in $(seq 1 10); do
        let resu1=tabl1*i
        echo "$tabl1 x $i = $resu1"
    done
fi
```



13. Crea un script para jugar a adivinar un número entre 1 y 10 (este número será generado al azar, para ello debes utilizar la función \$RANDOM). El programa irá preguntando números al usuario hasta que acierte.

```
#!/bin/bash
alea1=echo $((RANDOM%10))
echo -n "Di un número del 1 al 10"
read nume1
while [ $alea1 -ne nume1 ]; do
    echo "Error número equivocado."
    echo -n "Elige otro número: "
    read nume1
done
echo "¡¡ Has acertado !!"
```

14. Crea un script que pida una contraseña y permita tres intentos. Si el usuario da la contraseña correcta muestra un mensaje de bienvenida. Si no es así, mostrará un mensaje de error. La contraseña se encuentra especificada en una variable dentro del script.

```
#!/bin/bash
contra1="trocola"
inten1=1
echo -n "Introduce tu contraseña: "
read contra2
until [ $inten1 -eq 3 ] || [ $contra1 = $contra2 ]; do
    echo "Error, contraseña incorrecta."
    echo
    echo -n "Introduce tu contraseña: "
    read contra2
    let inten1=inten1+1
done
if [ $contra1 = $contra2 ]; then
    echo "Contraseña correcta, bienvenido!"
else
    echo "Error, contraseña incorrecta, terminal bloqueado."
fi
```

15. Crea un script que se le pase un nombre de archivo como parámetro y muestre por pantalla qué tipo de archivo es. Además, si es un directorio mostrará su contenido y si es un fichero lo mostrará por pantalla.

```
#!/bin/bash
if [ -f $1 ]; then
    echo "$1 es un fichero y su contenido es:"
    more $1
elif [ -d $1 ]; then
    echo "$1 es un directorio y su contenido es:"
    ls -l $1
else
    echo "El archivo no existe"
fi
```