



9 DE DICIEMBRE DE 2018

SCRIPTS
SISTEMAS INFORMÁTICOS

JORDAN PERUCHO

1 DAM

ÍNDICE

Ejercicios.....	3
Introducción al Bash.....	3
Operaciones Aritméticas.....	3
Condiciones.....	4
Bucles.....	5

Ejercicios

Introducción al Bash

1. Hacer un script y ponerle de nombre primero.sh que saque por pantalla un mensaje de bienvenida a la programación por scripts.

```
#!/bin/bash
echo "BIENVENIDO A BASH"
```

2. Hacer un nuevo comando lp.sh, para sacar el contenido de un directorio pero paginado.

```
#!/bin/bash
ls -l $(pwd) | less
```

3. Hacer un nuevo comando pcinf.sh que saque toda la información importante del PC, nombre del pc, nombre del usuario conectado, path, sistema operativo.

```
#!/bin/bash
echo "Nombre del PC: " $HOSTNAME
echo "Nombre del Usuario: " $USER
echo "PATH del Sistema: " $PATH
echo "S.O: " $(uname)
```

4. Hacer un nuevo comando pcred.sh que saque toda la información importante de la red del PC: IP, mascara, IP del router.

```
#!/bin/bash
echo "Dirección IP: " $(ifconfig enp0s3 | grep inet | tr " " ":" | tr -s ":" | cut -d: -f3 | head -1)
echo "Mascara de Red: " $(ifconfig enp0s3 | grep netmask | tr " " ":" | tr -s ":" | cut -d: -f5 | head -1)
echo "Dirección IP Router: " $(arp -a | tr [\(\)] " " | tr " " ":" | tr -s ":" | cut -d: -f2)
```

Operaciones Aritméticas

1. Crea el archivo suma.sh que sume números y mostrar el resultado por pantalla.
 - a. Que sume todos los números que pasemos como parámetro, pueden ser mas de dos.
Ej1: suma 2 6
Ej2: suma 2 6 4 1
Ej3: suma 2
 - b. Lo mismo que en c) pero además que nos diga cuantos parámetros hemos pasado.

```
#!/bin/bash
if [ $1 != $1 ]
then
    read -p "Introduce el primer valor: " num1
    read -p "Introduce el segundo valor: " num2
    echo $(( $num1 + $num2 ))
else
    for i in $*
    do
        resultado=$(( $resultado + $i ))
    done

    echo "El resultado de la suma es" $resultado
    echo "Se han pasado" $# "parametros"
fi
```

Condiciones

Bucles

1. Hacer un script para que muestre 10 líneas:
 - a. línea 1
línea 2
.....
línea 9
línea 10
 - b. Nos pregunta el número de líneas
 - c. Pasamos el número de líneas como parámetro
 - d. Si no pasamos un parámetro nos pregunta el número de líneas

```
#!/bin/bash
if [ $# -eq 1 ]
then
    for LINEA in `seq 1 $1`
    do
        echo "Linea " $LINEA
    done
else
    read -p "Cuántas líneas quieres? " num
    for LINEA in `seq 1 $num`
    do
        echo "Linea " $LINEA
    done
fi
```

2. Escriba un programa suma100 que calcule el resultado de sumar todos los números del 1 al 100 secuencialmente, es decir: 1 + 2 + 3 + 4 + 5...

```
#!/bin/bash
for LINEA in {1..100}
do
    resultado=$(( $resultado+$i ))
done
echo $resultado
```

3. Cree un script que pida números al usuario hasta que este teclee la palabra "promedio", después calcule la media aritmética entre todos ellos.

```
#!/bin/bash
while [[ "$num" != "promedio" ]]
do
    read -p "Introduce numeros o 'promedio': " num
    resultado=$((resultado+$num))
    if [ "$num" != "promedio" ]
    then
        ((cont++))
    fi
done
echo $((resultado/$cont))
```

4. Cree un shell script llamado tabla que a partir de un número que se le pasará como argumento obtenga la tabla de multiplicar de ese número.

Si se llama con: tabla 5 debe responder:

TABLA DE MULTIPLICAR DEL 5

=====

5 * 1 = 5

5 * 2 =10

...

5 * 9 = 45

5 * 10 =50

Mejore el shell script para que se verifique que sólo se le ha pasado un argumento y que éste es un número válido entre 1 y 10.

```
#!/bin/bash
if [ $1 != $1 ];
then
    echo "Se espera un parámetro entre el 0 y el 10."
else
    if [ $1 -le 10 ];
    then
        echo "TABLA DE MULTIPLICAR DEL" $1
        echo "===== "
        for i in {1..10}
        do
            echo $1 " * " $i " = " $((i*$1))
        done
    else
        echo "Se espera un parámetro entre el 0 y el 10."
    fi
fi
```

5. Implementar un guion de bash (monton) que imprima en su salida estándar (por defecto, la pantalla) un triángulo isósceles formado por asteriscos de texto, simétrico verticalmente, y de altura igual al número que se le pasa como único y obligatorio parámetro. Por ejemplo, si se teclea \$./monton 5 el programa deberá mostrar en su salida lo siguiente:

```
*
***
*****
*****
*****
```

El script deberá comprobar que fue llamado con corrección (es decir, que tiene un parámetro y sólo uno, y que éste es un entero mayor que cero). Obviamente, el programa deberá funcionar para cualquier entero positivo (ya que, aunque la salida no cupiese en pantalla, podría redireccionarse sin problemas a un fichero dado).

```
#!/bin/bash
cadena="*"
for i in `seq 1 $1`
do
    echo "$cadena"
    cadena=$cadena"*"
done
```

6. Cree un fichero que contenga una lista de cinco nombres de pila y, separadas por una coma, una edad expresada en años. Por ejemplo una línea será "Juan,24". Guarde ese fichero como "nombre-edad.txt" y escriba un guion que lo recorra y diga por pantalla "Tal nombre tiene tantos años." para cada una de las líneas, por ejemplo para la anterior dira "Juan tiene 24 años".

```
Juan,24
Pepe,15
Bata,34
Rafa,26
Luis,20
```

```
#!/bin/bash
for i in `cat nombre-edad.txt`
do
    nombre=$(echo $i | cut -d, -f1)
    edad=$(echo $i | cut -d, -f2)
    echo $nombre " tiene " $edad " años"
done
```


7. Utilizando el fichero del ejercicio anterior haga un guion que diga quien de la lista es el mayor y el menor, y cual es la media de edad.

```
#!/bin/bash
menor=$(cat nombre-edad.txt | sort -n -t, -k2 | head -1)
mayor=$(cat nombre-edad.txt | sort -n -t, -k2 | tail -1)
echo $(echo $menor | cut -d, -f1) "es el menor con" $(echo $menor | cut -d, -f2) "años"
echo $(echo $mayor | cut -d, -f1) "es el mayor con" $(echo $mayor | cut -d, -f2) "años"
cont=0
for i in `cat nombre-edad.txt | cut -d, -f2`
do
    resultado=$((resultado+$i))
    ((cont++))
done
echo "La media de edad es" $((resultado/$cont))
```

8. Con la misma lista, haga un gui3n que cree un usuario para cada uno de ellos en su m3quina, el grupo debe ser "ejercicios", y que haga despu3s un tar.gz (uno solo) de todos los directorios de usuario de la m3quina (excluyendo a root) con el nombre homes-backup.tar.gz y que se guarde en /root/backups

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

9. Cree un script que borre todos los usuarios pertenecientes al grupo "ejercicios" pero antes de borrar a cada uno debe mostrar su nombre por pantalla y pedir con confirmaci3n.
10. Haga un gui3n que mire todos los directorios que haya dentro de /home y que por cada uno de ellos compruebe si existe dicho usuario en el /etc/passwd, si el usuario no existe debe dar la opci3n de comprimir en un backup y borrar el directorio de ese usuario fantasma. Antes de probarlo cree varios directorios dentro de /home

11. Realice un shell-script 'piramide.sh' que admita como argumento un número 'N' y que optenga a la salida una serie de 'N' filas de forma triangular.

Para ./piramide.sh 12 la salida sería.

```
01
02 02
03 03 03
04 04 04 04
05 05 05 05 05
06 06 06 06 06 06
07 07 07 07 07 07 07
08 08 08 08 08 08 08 08
09 09 09 09 09 09 09 09 09
10 10 10 10 10 10 10 10 10 10
11 11 11 11 11 11 11 11 11 11 11
12 12 12 12 12 12 12 12 12 12 12 12
```

```
#!/bin/bash
for i in `seq 1 $1`
do
    linea=""
    for j in `seq 1 $i`
    do
        if [ $i -lt 10 ]
        then
            valor="0"$i
        else
            valor=$i
        fi
        linea=$linea" "$valor
    done
    echo $linea
done
```

12. Hacer un juego adivina.sh para que el ordenador piense un número aleatorio entre 0 y 100, tenemos 5 intentos para adivinarlo
13. Hacer un script separar al que le pasaremos como parámetro un fichero nums.txt que contendrá números, tendremos que comprobar que hemos pasado un parámetro y que ese fichero existe. La función de separar es enviar los números pares que contiene nums.txt al fichero par.txt y los números impares a impar.txt, mostrar el contenido de los dos ficheros.