

Irene Jimenez Moron
1º DAM

1.- Escriba un script que elimine un archivo o directorio pasando como parámetro, y le pregunte si está seguro de llevar a cabo la acción.

```
#!/bin/bash

clear

if [[ -e $1 ]]; then
    if [[ -d $1 ]]; then
        read -p "¿Estás seguro de que quieres borrar el directorio $1? (S/N): "
opcion
        if [[ $opcion == "s" || $opcion == "S" ]]; then
            rm -r $1
        fi
    else
        read -p "¿Estás seguro de que quieres borrar el fichero $1? (S/N): "
opcion
        if [[ $opcion == "s" || $opcion == "S" ]]; then
            rm $1
        fi
    fi
else
    echo "No existe el directorio o fichero introducido."
fi
```

2.- Escribir un script que pueda mostrar información de un comando al ejecutar dicho script y pasar como parámetro el comando.

```
#!/bin/bash

clear
$1 --help
```

3.- Realiza un script que compruebe si el usuario actual del sistema es blas, si es así visualiza su nombre 5 veces, sino te despides de él amigablemente.

```
#!/bin/bash

USUARIO=$(whoami)

if [[ $USUARIO == blas ]]; then
    echo $USUARIO
    echo $USUARIO
    echo $USUARIO
    echo $USUARIO
    echo $USUARIO
else
    echo "Adios, $USUARIO."
fi
```

4.- En un fichero tengo una palabra clave. Haz un script que muestre si dicha palabra es el parámetro pasado o no.

```
#!/bin/bash

clear

if [[ ! -e ficheroclave.txt ]]; then
    $(echo "fichero" > ficheroclave.txt)
fi

clave=$(cat ficheroclave.txt)

if [[ $clave == $1 ]]; then
    echo "La palabra $1 es correcta"
else
    echo "Esa no es la palabra, inténtalo de nuevo"
fi
```

5.- Tenemos un menu principal:

1. Suma
 - Lee dos números y los suma.
2. Resta
 - Lee dos números y los resta.
3. Multiplicación
 - Lee dos número y los multiplica.
4. Salir
 - Termina el script.

```
#!/bin/bash

clear

read -p "Introduce:
1/Suma | 2/Resta | 3/Multiplicación | 4/Salir: " opcion

case $opcion in
    1) read -p "Primer número: " n1
        read -p "Segundo número: " n2
        suma=$((n1 + n2))
        echo "$n1 + $n2 = $suma"
        ;;
    2) read -p "Primer número: " n1
        read -p "Segundo número: " n2
        resta=$((n1 - n2))
        echo "$n1 - $n2 = $resta"
        ;;
    3) read -p "Primer número: " n1
        read -p "Segundo número: " n2
        multi=$((n1 * n2))
        echo "$n1 * $n2 = $multi"
        ;;
    *)
```

```
4) exit
esac
```

6.- Nos pide la edad y nos dice si es mayor de edad o menor.

```
#!/bin/bash

clear

read -p "Escribe tu edad: " edad

if [[ $edad -g 18 ]]; then
    echo "Eres mayor de edad"
else
    echo "Eres menor de edad"
fi
```

7.- Script que reciba un nombre de fichero, verifique que existe y que es un fichero de lectura-escritura, lo convierta en ejecutable para el usuario y el grupo y muestre el estado final de los permisos.

```
#!/bin/bash

clear

if [[ ! -f $1 ]]; then
    read -p "El fichero no existe o es un directorio."

    ¿Desea crear el fichero $1? S/N: " opcion
    if [[ $opcion == "S" || $opcion == "s" ]]; then
        echo > $1
    fi
elif [[ -r $1 && -w $1 ]]; then
    echo "Tiene permisos de lectura-escritura."
    chmod ug+x $1
    ls -l $1
else
    echo "El fichero no tiene permisos de lectura-escritura."
fi
```

8.- Script que nos diga al pulsar una tecla, si es letra, numero o carácter especial.

```
#!/bin/bash

clear

while true; do
    read -rsn1 entrada
    if [[ "$entrada" = [0-9] ]]; then
        echo "Es un número"
    elif [[ "$entrada" = [a-zA-Z] ]]; then
        echo "Es una letra"
    else
        echo "Es un caracter especial"
    fi
done
```

```
        fi
done
```

9.- Realizar un script que reciba varios parámetros y nos diga cuántos de esos parámetros son de directorios y cuantos son archivos. \$# contador que indica cuantos parámetros se pasan.

```
#!/bin/bash

clear

cont1=0
cont2=0
for i in $@; do
    if [[ -d $i ]]; then
        cont1=$((cont1+1))

        elif [[ -f $i ]]; then
            cont2=$((cont2+1))
    fi
done

echo "El número de directorios es $cont1"
echo "El Número de ficheros es $cont2"
```

10.- Mostramos menú, con productos para vender, luego nos pide que introduzcamos la opción. Luego mensaje que indica qué introduzca moneda. Si ponemos precio exacto nos da mensaje, "Gracias buen provecho", si ponemos menos, nos diga falta. Si poner mas valor, nos indique el cambio con mensaje.

```
#!/bin/bash

clear

read -p "Productos a la venta:
1) Producto 1: 1€
2) Producto 2: 2€
3)Producto 3: 3€
4) Salir
¿Qué producto desea comprar?:" opcion

case $opcion in
    1) read -p "Tiene que introducir 1€: " dinero
        if [[ $dinero = 1 ]]; then
            echo "Gracias buen provecho"
        elif [[ $dinero -gt 1 ]]; then
            echo "El cambio: "$((dinero - 1))
        elif [[ $dinero -lt 1 ]]; then
            echo "Falta dinero"
        fi
        ;;

    2) read -p "Tiene que introducir 2€: " dinero
        if [[ $dinero = 2 ]]; then
```

```

        echo "Gracias buen provecho"
    elif [[ $dinero -gt 2 ]]; then
        echo "El cambio: "$((dinero - 2))
    elif [[ $dinero -lt 2 ]]; then
        echo "Falta dinero"
    fi
;;

3) read -p "Tiene que introducir 3€: " dinero
    if [[ $dinero = 3 ]]; then
        echo "Gracias buen provecho"
    elif [[ $dinero -gt 3 ]]; then
        echo "El cambio: "$((dinero - 3))
    elif [[ $dinero -lt 3 ]]; then
        echo "Falta dinero"
    fi
;;

4) exit
;;

```

esac

11.- Realizar un script que pida introducir la ruta de un directorio por teclado (Hay que validar que la variable introducida sea un directorio) nos diga cuántos archivos y cuántos directorios hay dentro de ese directorio.

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

```
clear
```

```
read -p "Introduzca la ruta del directorio: " directorio
```

```
cont1=$(find $directorio/* -type f | wc -l)
```

```
cont2=$(find $directorio/* -type d | wc -l)
```

```
echo "Dentro del directorio hay $cont1 directorios y $cont2 ficheros."
```

12.- Realiza un script que introduzca número por parámetro y muestre tabla de multiplicar.

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

```
clear
```

```
echo "La tabla de multiplicar del $1 es: "
```

```
for (( i = 1; i <= 10; i++ )); do
```

```
    sol=$(( $1 * $i ))
```

```
    echo "$1 * $i = $sol"
```

```
done
```

13.- Script que limpie todas las reglas, y de permiso a todas las conexiones.

14.- Script que limpie todas las reglas, y prohíba cualquier conexión.

15.- Tendrá 3 parámetros: red(ip), entrada-salida, aceptar-denegar. Dará estos permisos a iptables.