

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

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

```
If [[ -e $1 ]]; then
```

```
    If [[ -d $1 ]]; then
```

```
        Read -p "¿Quieres borrar el directorio $1? (S/N): " opción
```

```
        If [[ $opcion == "s" || $opcion == "S" ]]; then
```

```
            Rm -r $1
```

```
        Fi
```

```
    Else
```

```
        Read -p "¿Quieres borrar el directorio $1? (S/N): " opción
```

```
        If [[ $opcion == "s" || $opcion == "S" ]]; then
```

```
            Rm $1
```

```
        Fi
```

```
    Fi
```

```
Else
```

```
    Echo "No existe el directorio o fichero introducido"
```

```
fi
```

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

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

```
$1 --help
```

- 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 "Que tengas buen día, $USUARIO"
```

```
fi
```

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

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

```
if [[ ! -e fichero.txt ]]; then
```

```
    $(echo "pepe" > fichero.txt)
```

```
fi
```

```
clave=$(cat fichero.txt)
```

```
if [[ $clave == $1 ]]; then
```

```
    echo "La palabra era $1 Enhorabuena"
```

```
else
```

```
    echo "No has acertado, prueba otra vez..."
```

```
fi
```

- Tenemos un menu principal:
  - (1) Suma
    - Lee dos números y los suma.
  - (2) Resta
    - Lee dos números y los resta.
  - (3) Multiplicación
    - Lee dos números y los multiplica.
  - (4) Salir
    - Termina el script.

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

```
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"

;;

*) exit 0

esac

```

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

```

#!/bin/bash

read -p "Introduce tu edad: " edad

if [[ $edad -ge 18 ]]; then

    echo "Eres mayor de edad."

else

    echo "Eres menor de edad."

fi

```

- 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

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
fi

```

```

        fi

elif [[ -r $1 && -w $1 ]]; then

    echo "Tiene permisos de lectura y escritura."

    chmod ug+x $1

    ls -l $1

else

    echo "El fichero no tiene permisos de lectura-escritura."

fi

```

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

```

#!/bin/bash

while true; do

    read -rsn1 entrada

    if [[ "$entrada" = [0-9] ]]; then

        echo "Ha introducido el número $entrada"

    elif [[ "$entrada" = [a-zA-Z] ]]; then

        echo "Ha introducido la letra $entrada"

    else

        echo "Ha introducido el caracter especial $entrada"

    fi

done

```

- realizar un script que reciba varios parametros y nos diga cuantos de esos parametros son de directorios y cuantos son archivos. \$# contador que indica cuantos parametros se pasan.

```

#!/bin/bash

```

#Script 9: Realizar un script que reciba varios parametros y nos diga cuantos de esos

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

```
contadorDirectorios=0
```

```
contadorFicheros=0
```

```
for i in $@; do
```

```
    if [[ -d $i ]]; then
```

```
        contadorDirectorios=$((contadorDirectorios+1))
```

```
    elif [[ -f $i ]]; then
```

```
        contadorFicheros=$((contadorFicheros+1))
```

```
    fi
```

```
done
```

```
echo "Número de directorios: $contadorDirectorios"
```

```
echo "Número de ficheros: $contadorFicheros"
```

- Mostramos menu, con productos para vender, luego nos pide que introduzcamos la opcion. luego mensaje que indica que 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
```

```
read -p "Productos a la venta:
```

```
1) Pizza barbacoa: 3€
```

```
2) Sandwich atún y pollo: 2€
```

```
3) Kinder Bueno: 1€
```

```
4) Salir
```

Introduzca el producto que desea comprar: " opcion

case \$opcion in

1) read -p "Introduce los 3 euros: " 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

::

2) read -p "Introduce los 2 euros: " 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 "Introduce los 1 euro: " 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

;;

*) exit 0

;;

esac

```

- 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 cuantos archivos y cuantos directorios hay dentro de ese directorio.

```

#!/bin/bash

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

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

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

echo "Dentro del directorio hay $contadorDirectorios directorios y
$contadorFicheros ficheros."

```



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

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

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

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

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

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

```
done
```

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

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

```
iptables -t nat -F
```

```
iptables -t mangle -F
```

```
iptables -F
```

```
iptables -X
```

```
iptables -Z
```

```
iptables -P INPUT ACCEPT
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -P FORWARD ACCEPT
```

```
iptables -I INPUT -j ACCEPT
```

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

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

```
iptables -t nat -F
```

```
iptables -t mangle -F
```

```
iptables -F
```

iptables -X

iptables -Z

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

iptables -I INPUT -j DROP

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