

**1. Muestra por pantalla el contenido del fichero /etc/passwd y /etc/shadow**

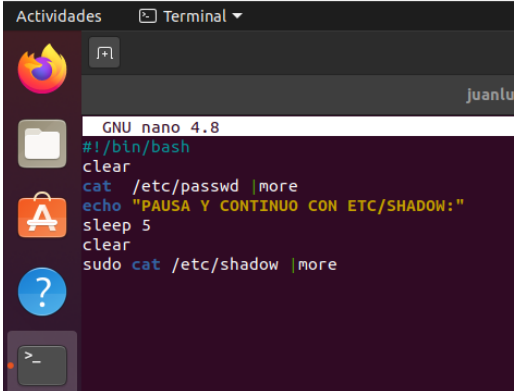
Sudo nano 1.sh

Chmod 777 1.sh

./1.sh

ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo] - Oracle VM

Archivo Máquina Ver Entrada



```
GNU nano 4.8
#!/bin/bash
clear
cat /etc/passwd | more
echo "PAUSA Y CONTINUO CON ETC/SHADOW:"
sleep 5
clear
sudo cat /etc/shadow | more
```

**2. Lee por parámetros 2 valores y muestra su suma**

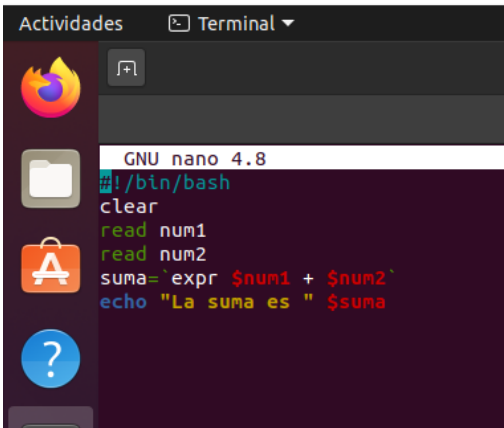
Sudo nano 2.sh

Chmod 777 2.sh

./2.sh

ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriend

Archivo Máquina Ver |

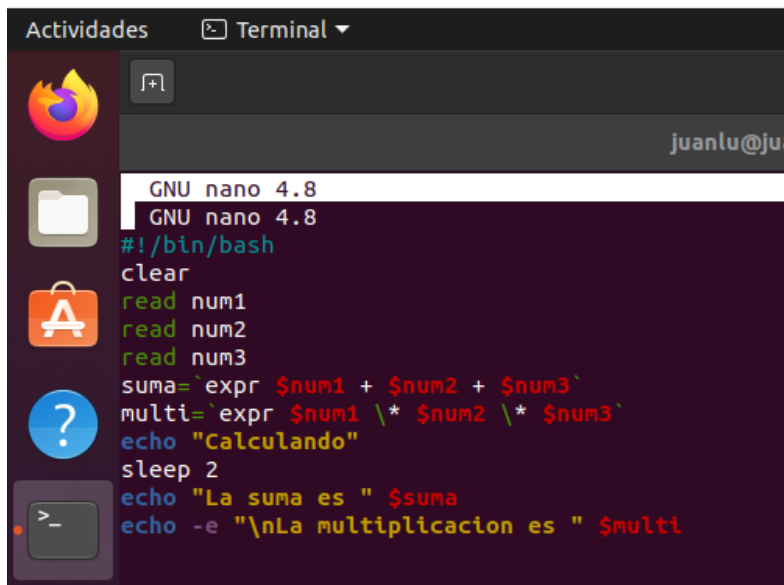


```
GNU nano 4.8
#!/bin/bash
clear
read num1
read num2
suma=`expr $num1 + $num2`
echo "La suma es " $suma
```

**3. Que lea por teclado tres valores y muestra su suma y multiplicación**

ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo] - Oracle VM Virtu

Archivo Máquina Ver Entrada



```
GNU nano 4.8
GNU nano 4.8
#!/bin/bash
clear
read num1
read num2
read num3
suma=`expr $num1 + $num2 + $num3`
multi=`expr $num1 \* $num2 \* $num3`
echo "Calculando"
sleep 2
echo "La suma es " $suma
echo -e "\nLa multiplicacion es " $multi
```

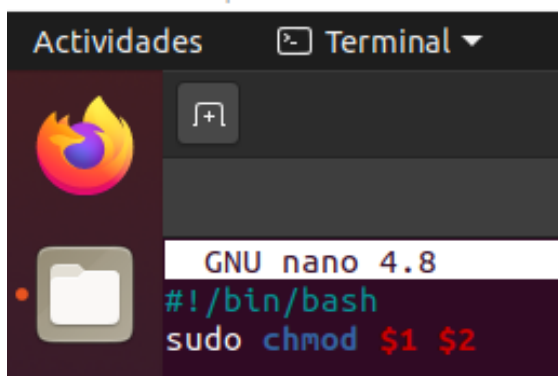
**4. Muestra los usuarios del sistema y accede al sistema con el usuario elegido**

```
GNU nano 4.8
#!/bin/bash
echo "Elige un usuario: "
grep home /etc/passwd |grep -v nologin | cut -d: -f1
read usuario
sudo login $usuario
```

**5. Crea un shell script que asigne ciertos derechos (especificados como primer argumento) a un fichero pasado como segundo argumento.**

ASO-Ubuntu 20.04-JUANLUIS.ACEB

Archivo Máquina Ver Entrada [



```
GNU nano 4.8
#!/bin/bash
sudo chmod $1 $2
```

**6. Hacer un guión de shell que cuente cuantos ficheros y cuantos directorios hay en el directorio pasado como argumento**

ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
Actividades Terminal
j
GNU nano 4.8
#!/bin/bash
arch=$(ls $1 -1F | grep -v / | grep -v @ | wc -l )
dire=$(ls $1 -1F | grep / | wc -l )
enla=$(ls $1 -1F | grep @ | wc -l )
echo "Archivos : "$arch " Directorios: " $( expr $dire + $enla )
```

**7. Elabora un shell script de forma que nos diga si un fichero pasado como argumento existe y si el usuario puede leer su contenido.**

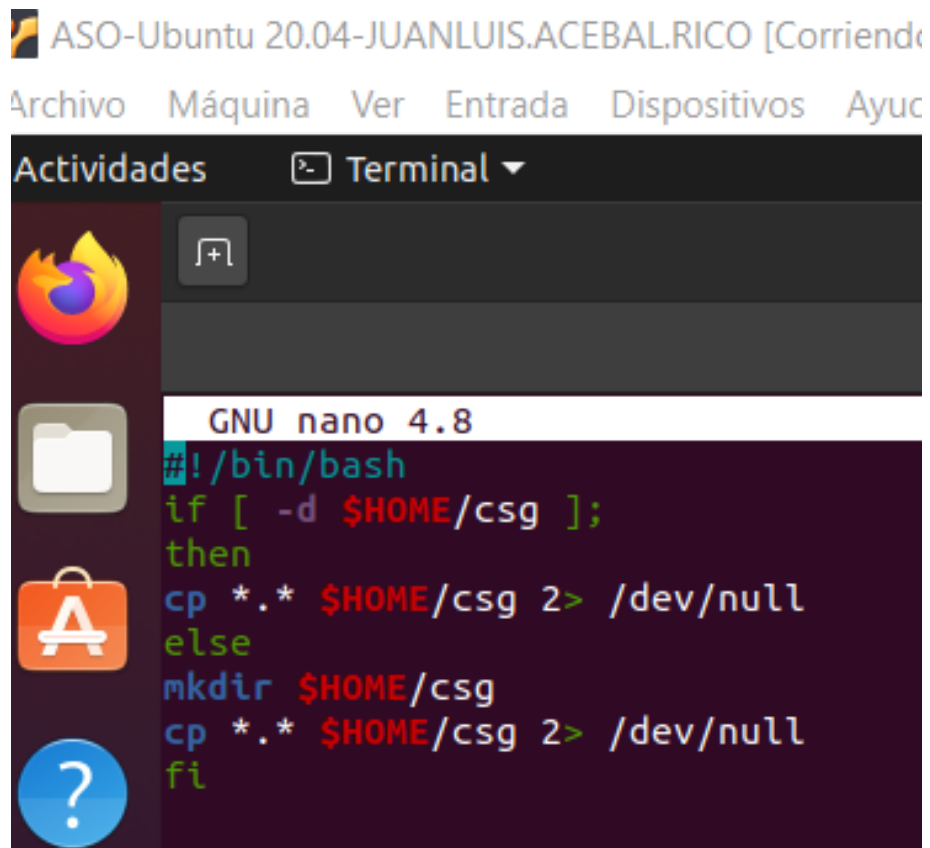
ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo] - (

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
Actividades Terminal
j
GNU nano 4.8
#!/bin/bash
echo "Dime que quieres comprobar: "
read x
if [ -d $x ];
then
echo "--Existe el directorio: "
cd $x
if [ $? -eq 0 ];
then
echo " y puedes leerlo"
else
echo " y no puedes leerlo"
fi
else
echo "--NO existe COMO directorio"
fi

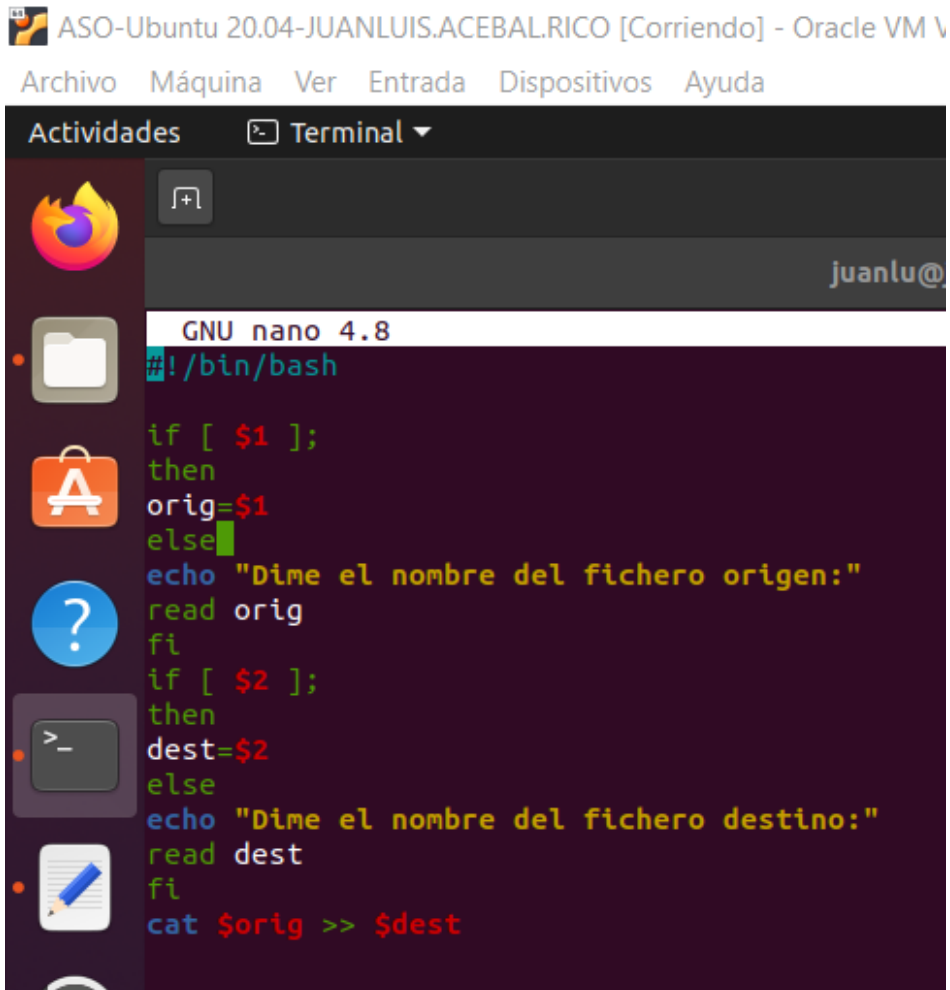
if [ -f $x ];
then
echo "--Existe el fichero: "
cat $x > /dev/null 2>&1
if [ $? -eq 0 ];
then
echo " y puedes leerlo"
else
echo " y no puedes leerlo"
fi
else
echo "--NO existe COMO fichero"
fi
```

8. Hacer un guión de shell que copie todos los ficheros del directorio actual en un directorio llamado csg que está en tu directorio personal. Si el directorio no existe el guión lo debe de crear y copiar los ficheros. No deben aparecer por pantalla los posibles errores en la copia de archivos.



```
ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo]
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
GNU nano 4.8
#!/bin/bash
if [ -d $HOME/csg ];
then
cp *.* $HOME/csg 2> /dev/null
else
mkdir $HOME/csg
cp *.* $HOME/csg 2> /dev/null
fi
```

9. Script que copie un fichero regular en otro, ambos pasados como argumentos. Si no se le pasan los argumentos, lo comprobará, y solicitará al usuario que los introduzca interactivamente.



The screenshot shows a terminal window titled "ASO-Ubuntu 20.04-JUANLUIS.ACEBAL.RICO [Corriendo] - Oracle VM". The terminal is running the GNU nano 4.8 editor. The script being edited is a shell script that checks for two arguments. If the first argument is provided, it sets 'orig' to the first argument. If the second argument is provided, it sets 'dest' to the second argument. If either argument is missing, it prompts the user to enter the file name. Finally, it uses the 'cat' command to copy the file from 'orig' to 'dest'.

```
#!/bin/bash

if [ $1 ];
then
orig=$1
else
echo "Dime el nombre del fichero origen:"
read orig
fi
if [ $2 ];
then
dest=$2
else
echo "Dime el nombre del fichero destino:"
read dest
fi
cat $orig >> $dest
```

10. Shell script que copie el fichero pasado como primer argumento al segundo, el tercero al al cuarto y así sucesivamente. No se conoce el nº de argumentos.'

11. Muestra un menú con diferentes herramientas de monitorización y ejecuta la herramienta seleccionada. Estará distribuidas por categorías en submenús. Tendrá una opción Salir. Tres versiones para hacerlo, y lo harás en el orden siguiente:

a. El menú solo se ejecuta una vez (sólo con case)

```
1 #!/bin/bash
2 function op1(){
3 sel1=1
4 while [ 3 -ne $sel1 ]
5 do
6 clear
7 echo "HERRAMIENTAS DE PROCESOS"
8 echo "1. ps -u"
9 echo "2. top"
10 echo "3. Atras"
11 echo "Elige una opción:"
12 read sel1
13 done
14 }
15 function op2(){
16 echo "NADA 2"
17 read -n 1 nada
18 }
19 function op3(){
20 echo "NADA 3"
21 read -n 1 nada
22 }
23 function op4(){
24 echo "NADA 4"
25 read -n 1 nada
26 }
27 sel=0
28 #while [ 5 -ne $sel ]
29 #do
30 clear
31 echo "HERRAMIENTAS DE MONITORIZACION"
32 echo "1. De procesos"
33 echo "2. De almacenamiento"
34 echo "3. De memoria"
35 echo "4. De red"
36 echo "5. Salir"
37 echo "Elige una opción:"
38 read sel
39 case $sel in
40 1)op1;;
41 2)op2;;
42 3)op3;;
43 4)op4;;
44 esac
45 #done
```

b. El menú se repite hasta que se elija la opción 5 (con while y case)

```
1 #!/bin/bash
2 function op1(){
3 sel1=1
4 while [ 3 -ne $sel1 ]
5 do
6 clear
7 echo "HERRAMIENTAS DE PROCESOS"
8 echo "1. ps -u"
9 echo "2. top"
10 echo "3. Atras"
11 echo "Elige una opción:"
12 read sel1
13 done
14 }
15 function op2(){
16 echo "NADA 2"
17 read -n 1 nada
18 }
19 function op3(){
20 echo "NADA 3"
21 read -n 1 nada
22 }
23 function op4(){
24 echo "NADA 4"
25 read -n 1 nada
26 }
27 sel=0
28 while [ 5 -ne $sel ]
29 do
30 clear
31 echo "HERRAMIENTAS DE MONITORIZACION"
32 echo "1. De procesos"
33 echo "2. De almacenamiento"
34 echo "3. De memoria"
35 echo "4. De red"
36 echo "5. Salir"
37 echo "Elige una opción:"
38 read sel
39 case $sel in
40 1)op1;;
41 2)op2;;
42 3)op3;;
43 4)op4;;
44 esac
45 done
```

c. El submenú se repetirá hasta que se introduzca la opción 3 que volverá al menú principal (con varios while anidados y case para cada uno de ellos)

```
1 #!/bin/bash
2 function op1(){
3 sel1=1
4 while [ 3 -ne $sel1 ]
5 do
6 clear
7 echo "HERRAMIENTAS DE PROCESOS"
8 echo "1. ps -u"
9 echo "2. top"
10 echo "3. Atras"
11 echo "Elige una opción:"
12 read sel1
13 case $sel1 in
14 1)ps -u
15 sleep 10;;
16 2)top ;;
17 esac; done; }
18 function op2(){
19 echo "NADA 2"
20 read -n 1 nada; }
21 function op3(){
22 echo "NADA 3"
23 read -n 1 nada; }
24 function op4(){
25 echo "NADA 4"
26 read -n 1 nada; }
27 sel=0
28 while [ 5 -ne $sel ]
29 do
30 clear; echo "HERRAMIENTAS DE MONITORIZACION"
31 echo "1. De procesos"
32 echo "2. De almacenamiento"
33 echo "3. De memoria"
34 echo "4. De red"
35 echo "5. Salir"
36 echo "Elige una opción:"
37 read sel
38 case $sel in
39 1)op1;;
40 2)op2;;
41 3)op3;;
42 4)op4;;
43 esac; done
```



12. Escribe un script llamado `apagar.sh` que pasando el argumento `N`, permita:

a. Reiniciar el equipo ahora

b. Reiniciar el equipo dentro de `N` minutos (pasado como argumento). Si el

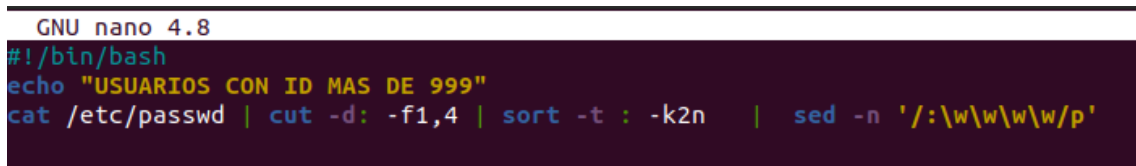
usuario no escribe ningún argumento, entonces se reiniciará el equipo en

3 minutos



```
1 #!/bin/bash
2 if [ $1 ];
3     then
4         if [[ $1 -gt 0 ]] 2> /dev/null
5             then
6                 sudo shutdown -r +$1
7             else
8                 sudo shutdown -r +3
9             fi
10    else
11        sudo shutdown -r now
12 fi
```

13. Shell script que muestre los usuarios que tengan ID superior a 999



```
GNU nano 4.8
#!/bin/bash
echo "USUARIOS CON ID MAS DE 999"
cat /etc/passwd | cut -d: -f1,4 | sort -t: -k2n | sed -n '/:\w\w\w\w/p'
```

14. A todos los archivos del directorio que pases como parámetro, y que tengan

permisos de escritura para otros, se lo quitarás. Si no pasas ningún argumento,

dará mensaje correspondiente.

```
#!/bin/bash
if [ $1 ]
then
if [ -d $1 ]
then
#chmod -v o-w $1 *.*
IFS='
'

for i in $(ls $1)
do
if [ -f $i ]
then
echo "HOLA:$1/$i"
if [ "${1: -1}" = "/" ]
then
ruta=$1$i
else
ruta=$1/$i
fi
chmod -v o-w $ruta
fi
done

else
echo "No es un directorio"
fi
else
echo "No pasaste ningún argumento"
fi
```

15. Guión shell que genere un fichero llamado listaetc que contenga los ficheros con permiso de lectura que haya en el directorio /etc

```

GNU nano 4.8
#!/bin/bash
direct="/etc"
echo "" > listaetc
echo "" > listaetc-r
echo "Originalmente hay: "
ls -1F $direct | grep -v / | wc -l
for a in $(ls -1F $direct | grep -v / )
do
ruta="$direct/$a"
#echo "$ruta"
if [ -r $ruta ]
then
echo "$ruta" >> listaetc
else
echo "$ruta" >> listaetc-r
fi
done
echo "Finalmente hay:"
cat listaetc | grep -v "^[ ]*$" | wc -l
echo "Sin lectura:"
cat listaetc-r | grep -v "^[ ]*$" | wc -l

```

16. Elabora un script que mate todos los procesos man que estén ejecutándose. No utilizar killall

```

GNU nano 4.8
#!/bin/bash
man man &
#man nano &
#man echo &

echo "A MATAR:"
ps aux | grep "\ man\ "
for a in $(ps aux | grep "\ man\ " | cut -d " " -f6)
do
kill -n 9 $a
done

echo "Resultado:"

ps aux | grep "\ man\ "

```

a. Modifica el ejercicio dándole el nombre del proceso mediante parámetro

```
GNU nano 4.8
#!/bin/bash
echo "Dime el nombre del proceso a matar:"
read nombre
echo "A MATAR:"
ps aux | grep "$nombre"
for a in $(ps aux | grep "$nombre" | cut -d " " -f6)
do
echo "Matar a $a? S/N"
read cont
if [ "$cont" = "S" ] || [ "$cont" = "s" ]; then
echo "-----"
echo "Matando: $a"
kill -n 9 $a
fi
done
echo "Resultado:"
ps aux | grep "$nombre"
```

17. Modificar el ejercicio anterior para que antes del matar el proceso, en este caso será el proceso sleep, mande un mensaje instantáneo mediante la orden write al usuario en cuestión. En el mensaje se advierte que deje de usar el comando sleep.

```
GNU nano 4.8 16.sh
#!/bin/bash
echo "Dime el nombre del proceso a matar:"
read nombre
echo "A MATAR:"
ps aux | grep "$nombre"
for a in $(ps aux | grep "$nombre" | cut -d " " -f6)
do
echo "Matar a $a? S/N"
read cont
if [ "$cont" = "S" ] || [ "$cont" = "s" ]; then
if [ $(ps aux | grep "sleep" | grep $a | cut -d " " -f6) ]
then
echo "Tiene sleep, mandamos mensaje al dueño del proceso:"
ps aux | grep "sleep" | grep $a | cut -d " " -f1
for (( i=1; i<7; i++ )) do
echo "Te he matado el proceso sleep con PID $a" | write $(ps aux | grep "sleep" | grep $a | cut -d " " -f1) tty$1 2> /dev/null
done
fi
echo "-----"
echo "Matando: $a"
kill -n 9 $a
fi
done
echo "Resultado:"
ps aux | grep "$nombre"
```

18. A un shell script se le pasa como argumento el nombre de un fichero donde guardar la información, y que pida por teclado un nombre y un teléfono, lo guarde en el fichero que le hemos introducido como parámetro, hasta que le introduzcas la palabra "fin".

```
GNU nano 4.8
#!/bin/bash
echo "Dime el nombre del fichero"
read fichero
while [ "$dato" != "fin" ]
do
echo "-----"
echo "Dime nombre:"
read nombre
if [ "$nombre" == "fin" ]
then
echo "www" > /dev/null
else
echo "Dime telefono:"
read telefono
fi
if [ "$nombre" == "fin" ] || [ "$telefono" == "fin" ]
then
dato="fin"
else
echo "NOMBRE: $nombre TELEFONO: $telefono" >> $fichero
fi
#read dato
#echo "$dato"
done
echo "-----"
echo "-----"
echo "LEYENDO FICHERO: "
echo "-----"
echo "-----"
cat $fichero
```

19. Crear un shell script que te permita consultar la agenda anteriormente creada por nombre o por teléfono, se le da al script el nombre del fichero donde están los datos, como parámetro.

```
GNU nano 4.8
#!/bin/bash
echo "Dime el nombre del fichero:"
read fichero
echo "Dime el nombre de la persona o telefono de la persona:"
read busqueda
echo "-----"
echo "El resultado es: "
echo "-----"
cat $fichero | grep $busqueda | cut -d " " -f2,4 | tr -d $busqueda | tr -d " "
```

20. Busca todos los archivos que tengan dentro la palabra Hack y guárdalos en la carpeta /root/cuarentena

```
GNU nano 4.8
#!/bin/bash
if [ -d /root/cuarentena ]
then
echo "" 2> /dev/null
else
sudo mkdir /root/cuarentena
fi

#https://stackoverflow.com/es/q/350191
ls -R1 / 2> /dev/null |
while read l; do case $l in
*: ) d=${l%:};;
"") d=;;
*) ruta=$(echo "$d/$l");;
#-----
esac
if [ $(echo "$ruta" | grep hack ) ] 2> /dev/null
then
echo "COPIANDO: $ruta"
rutasin=`echo "$ruta" | tr "/" "-"`
echo "$rutasin"
sudo cp $ruta /root/cuarentena/$rutasin && echo "-----COPIADO"
fi
done
```

21. Crea el directorio /home/tu\_usuario/temporal

a. Copia todos los datos que hay en /sbin al directorio temporal

b. Crea el documento de texto comandos\_basicos, en el que incluirás algunos de los comandos que hay en temporal

```
#!/bin/bash
if [ -d /home/juanlu/temporal ]
then
echo "" > /dev/null
else
mkdir /home/juanlu/temporal
fi
#PARTE A
sudo cp /sbin/* /home/juanlu/temporal
#PARTE B
z=0
y=0
echo "" > comandos_basicos
for x in $(ls -lFa /home/juanlu/temporal | tr -d "*" | grep -v "\." )
do
z=`expr $z + 1`
y=`expr $y + 1`
if [ $(( $z % 2 )) == 0 ]
then
if [ $(( $y % 19 )) == 0 ]
then
echo "$z"
echo "$x" >> comandos_basicos
fi
fi
done
echo "PARTE B LISTADO-----"
cat comandos_basicos
```

c. Crea un Shell script que borre del directorio temporal todos los comandos que no estén en el documento comandos\_basicos

```
GNU nano 4.8
#!/bin/bash
for i in $(ls -1 /home/juanlu/temporal)
do
    bool=1
    for z in $(cat comandos_basicos)
    do
        if [ "$z" = "$i" ]
        then
            bool=0
            echo "$i"
        fi
    done
    if [ "$bool" == 1 ]
    then
        rm -f /home/juanlu/temporal/$i
    fi
done
```