

U4. Administración de sistemas Linux

Ejercicios Scripts Repaso 2

1. Crea un script que escoja un fichero o directorio aleatoriamente del directorio en el que se ejecuta y pregunte si quieres que lo borre.

SOL:

```
#!/bin/bash
listaFicheros=( `ls` );
nF=$((RANDOM%#listaFicheros)); n=$((RANDOM%nF));
fichero=${listaFicheros[$n]};
echo "¿Quiéres que borre ${fichero}? Indíquelos con s(sí) o n(no):"
read -n1 respuesta #echo $respuesta
if [ ${respuesta} = "s" ]
then
    if [ -d $fichero ]
    then
        #Es un directorio
        #rmdir $fichero
        echo "Borramos el directorio $fichero"
    else
        #Es un fichero -f $fichero es true
        #rm $fichero
        echo "Borramos el fichero $fichero"
    fi
    echo "Ya está borrado!"
else
    echo "Lo
borramos..."; sleep 3;
    echo "Es broma!";
fi
```

2. Crea un script que devuelva los ficheros del directorio actual e indique un mensaje como el siguiente: `[L] [SEP]`

Fichero *nombre_fichero* y su extensión *extensión* `[L] [SEP]`

Fichero sin extensión: `[L] [SEP]`

SOL:

```
#!/bin/bash echo "Los ficheros del directorio
actual son:" for i in `ls`
do
    if [ -f $i
]
then
    ext=${i#*.*}
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

```
        if [ $ext != $i ]
        then
            echo "Fichero: ${i%.*} y su extensión ${ext}."
        else
            echo "Fichero sin extensión: $i"
        fi
    fi
done
```

3. Crea un script que devuelva el nombre de ficheros y directorios que contiene el directorio en el que lo ejecutamos, pero indicando el nombre, propietario y tamaño. Debe estar ordenado por tamaño de menor a mayor [L]
[SEP]

SOL:

```
#!/bin/bash
#Script que devuelve nombre, propietario y tamaño ordenado por tamaño
#No en orden: ls -l | tr -s " " " " | cut -d" " -f4,5,9 | sort -n -k2 nombre=( `ls -l
| tr -s " " " " | sort -nk5 | cut -d" " -f9` ) prop=( `ls -l | tr -s " " " " | sort -nk5
| cut -d" " -f3` ) tam=( `ls -l | tr -s " " " " | sort -nk5 | cut -d" " -f5` ); #da
error si usas tamaño for ((i=0;i<=${#nombre[*]}; i++)) do      echo
"${nombre[$i]} ${prop[$i]} ${tam[$i]}" done
```

4. Crea un script que devuelva una lista de los usuarios del sistema con Shell bash eliminando del listado al usuario que ha ejecutado el script. [L]
[SEP]

SOL:

```
#!/bin/bash
#Script que devuelve usuarios del sistema con shell bash excepto el que ejecuta
yo=`whoami` usuarios=`grep "/bin/bash" /etc/passwd | cut -d":" -f1` echo
${usuarios/${yo}/""}
```

5. Crea un script Shell que admita 1 parámetro y que muestre por pantalla el UID y el GID primario del usuario que se le pasa como parámetro. La salida debe ser la siguiente:

El UID del usuario es ... [L]
[SEP]

El GID del grupo primario del usuario es

SOL:

```
#!/bin/bash
if [ ! -z $1 ]
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

```
then uid=`cat /etc/passwd | grep -w $1 | cut -d ':' -f2`
    gid=`cat /etc/passwd | grep -w $1 | cut -d ':' -f3`
    if [ ! -z $uid ] then
        echo El UID del usuario $1 es $uid
        echo El GID del usuario $1 es $gid
    else
        echo "El usuario $1 no existe." fi
else
    echo "Debes introducir un parámetro"
fi
```

6. Crea un programa que reciba por parámetro el nombre de un usuario y MATE a todos sus procesos.

```
SOL
#!/bin/bash
if [ ! -z $1 ]
then
    for p in `ps -u $1 | tr -s ' ' | cut -d ' ' -f1`
    do
        kill -9 $p
    done else
        echo "Debes indicar el usuario" fi
```

7. Crea un script que reciba como parámetro la ruta absoluta y devuelva un mensaje indicando si es un fichero, un directorio o si no existe. En caso de que no exista le preguntará al usuario si quiere crear el directorio con ese nombre SOL: #!/bin/bash

```
if [ -a $1 ]
then
    if [ -d $1 ] then echo "Es un
    directorio"
    else echo "Es un fichero"
    fi
else echo "No existe."
    read -p "¿Quiere crear el directorio $1?" resp
    if [ resp = "s" ] then
        mkdir $1
        echo "El directorio ha sido creado" fi
fi
```

8. Crea un programa que reciba como parámetro el nombre de un usuario del sistema e indique si está conectado o no.

```
SOL: #!/bin/bash
if [ ! -z $1 ]
then c=`who | grep -w $1
    2>/dev/null`
    if [ -z $c ]
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

```
        then echo $1 no está conectado
        else
            echo $1 está conectado
        fi
    else
        echo "Debes indicar el usuario" fi
```

9. Hacer un script que busque la presencia del fichero pasado como argumento en alguno de los directorios referenciados en la variable \$PATH SOL

```
#!/bin/bash
if [ -a $1 ]
then
    for dir in `echo $PATH | tr -s ':' ' '`
    do
        if [ -a ${dir}/${1} ]
        then
            echo el fichero $1 está en $dir
        fi
    done
else
    echo "El fichero no existe" fi
```

10. Escribir un script que para cada argumento que reciba realice una de las siguientes operaciones:

```
    a. si es un directorio ha de listar los ficheros que contiene,
    b. si es un fichero regular lo tiene que mostrar por pantalla,
    c. en otro caso, que indique que no es ni un fichero ni un
    directorio.    #!/bin/bash    if [ $# -gt 0 ]
then
    for fich in $*
do
    if [ -d $fich ]
then
        echo "usando ls"
        ls $fich
    elif [ -f $fich ]
then
        cat
        $fich
    else
        echo $fich no es ni un fichero ni un directorio
    fi
done
else
    echo "Debes introducir algún parámetro"
fi
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

11. Crea un guión shell que reciba como parámetro la ruta completa de un directorio y muestre por pantalla la suma del tamaño de todos los ficheros que contiene. [1] [SEP] SOL: #!/bin/bash

```
if [ -a $1 ]
then tot=0
    for k in `ls -l $1 | grep ^- | tr -s " " " " | cut -d " " -f5`
    do let tot=$tot + $k
    done
    echo El tamaño de los ficheros de $1 es $tot
else
    echo "Debes introducir un directorio existente"
fi
```

12. Crea una versión distinta del script anterior que además de mostrar la suma, muestre el tamaño promedio con 2 decimales. [1] [SEP]

```
SOL: #!/bin/bash
if [ -a $1 ] then contenidos=`ls -l $1 | grep ^- | tr -s " " " " | cut
-d " " -f5` sum=0
    numc=`ls -l | grep ^- | wc -l`; let
    numc--;
    #numc=`echo $contenidos | wc -w`
    for i in $contenidos
    do
        let sum=$sum+$i
    done
    echo $sum
    echo "scale=2; $sum/$numc" | bc
else
    echo "Debes introducir un directorio existente"
fi
```

13. Construye un programa que reciba como parámetros el nombre de un directorio y un tamaño en bytes. El programa debe mostrar un listado de todas las entradas de ese directorio, cuyo tamaño sea inferior al indicado por parámetro. SOL #!/bin/bash

```
if [ -a $1 ] then
    if [ `echo $2 | grep -x -q "[0-9]\{+\}` ]
    then
        for k in `ls -l $1 | grep -v total | tr -s ' '*` #ponemos * para
        separar fácilmente
        do nom=`echo $k | cut -d '*' -f8`
            tam=`echo $k | cut -d '*' -f5`
            if [ $tam -lt $2 ]
            then echo $nom
            fi
        done
    fi
fi
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

```
        else
            echo "Debes incluir un número como 2º argumento"
        fi
    else
        echo "Debes introducir un directorio existente"
    fi
```

14. Crea un programa que acepte como parámetros una serie de nombres de ficheros y como último parámetro, el nombre de un directorio. El programa debe mostrar un mensaje indicando cuantos de los ficheros que recibió por parámetro, se encuentran en el directorio indicado.

SOL:

```
#!/bin/bash
if [ $# -gt 1 ]
then
    param=$* #almaceno los parámetros      let
    nf=$(( $# - 1 )) #me quedo con la posición del directorio  shift $nf
    dir=$1
    ficheros=$(param%$dir) #elimino el directorio de los parámetros
    acum=0
    for k in $ficheros
    do if [ -a ${dir}/${k} ] then
        let acum++
    fi
    done
    echo "El número de archivos es $acum"
else
    echo "Debes introducir un mínimo de 2 parámetros" fi
```

15. Realizar un script que reciba como parámetros una extensión de archivo (sin punto) y el nombre completo de un directorio. El programa preguntará al usuario si desea copiar o mover los archivos con esa extensión, del directorio actual al especificado como parámetro. SOL:

```
#!/bin/bash
if [ $# -eq 2 ]
then
    if [ -a $2 ]
    then
        echo "1. Deseo copiar los ficheros"
        echo "2. Deseo mover los ficheros"
        read op
        for k in `ls | grep .$1`
        do if [ $op -eq 1
            ] then cp $k $2
            fi
            if [ $op -eq 2 ] then
                mv $k $2
            fi
        done
    fi
fi
```

Implantación de Sistemas Operativos U4.

Administración de sistemas Linux

```
                fi
            done
        else echo "El directorio debe existir"
        fi
    else
        echo "Debes introducir 2 parámetros" fi
```