

DOCUMENTO EXPLICATIVO DE LA PRÁCTICA 3

Manuel Casas Castro
i72cascm@uco.es

#Ejercicio 1

```
#!/bin/bash
echo -e "\n"
```

CONTADOR DE ARGUMENTOS

#-----

```
if [ $# -ne 1 ];
then
    echo "Debe introducir $0 <directorio>"
    echo -e "\n"
    exit 1
fi
```

CONTADOR DE ARCHIVOS

#-----

```
contador_c=0;
contador_h=0;
```

```
# buscar archivos .c y contarlos
for x in $(find $1 -name "*.c")
do
    let contador_c=$contador_c+1;
done
```

```
#buscar archivos .h y contarlos
for x in $(find $1 -name "*.h")
do
    let contador_h=$contador_h+1;
done
```

#imprimir resultados

```
echo "En el directorio hay $contador_c archivos .c y $contador_h archivos .h."
echo -e "\n"
```

#imprimir parámetros de archivos

```
for x in $(find $1 -name "*.c" && find -name "*.h")
do
    echo "El fichero $x tiene $(cat $x | wc -l) líneas y $(cat $x | wc -m) caracteres."
done | sort -nr -k8
echo -e "\n"
```

#ejercicio 2

#!/bin/bash
echo -e "\n"

CONTADOR ARGUMENTOS

#-----

```
if [ $# -eq 0 ] || [ $# -ge 3 ];  
then  
    echo "Introduzca $0 <Directorio> y opcionalmetne, una cantidad de bytes"  
    echo -e "\n"  
    exit 1  
else  
    if [ $# -eq 1 ]; #Si se introduce un tamaño, lo denotaremos como la referencia, en caso contrario el  
tamaño será 0 por defecto  
    then  
        tam=0  
    else  
        tam=$2  
    fi
```

#MOSTRAR DATOS

#-----

```
flag=0  
for x in $(find $1 -size "$tam"c -or -size +"$tam"c)  
do  
    if [ "$flag" -eq 0 ]  
    then  
        echo -e "\n"  
        echo "Nombre; LongitudUsuario; FechaModificacion; FechaAcceso; Tamaño; Bloques;  
Permisos; Ejecutable"  
        flag=1 #Flag no permitirá que se muestre por pantalla mas de una vez el mensaje que explica el  
formato de las líneas de datos.  
    fi
```

```
if [ -x $x ];  
then  
    ejecutable=1  
else  
    ejecutable=0  
fi
```

#Datos que se nos piden

```
nombre=$(basename $x)  
let longitudUsuario=$(stat -c "%U" $x | wc -m)-1  
fechaModificacion=$(stat -c "%y" $x)  
fechaAcceso=$(stat -c "%x" $x)  
tamano=$(stat -c "%s" $x)  
bloques=$(stat -c "%b" $x)  
permisos=$(stat -c "%A" $x)
```

```
echo "$nombre; $longitudUsuario; $fechaModificacion; $fechaAcceso; $tamano; $bloques;
$permisos; $ejecutable"
done | sort -n -k5 -t ";"
fi
echo -e "\n"
```

#Ejercicio 3

```
#!/bin/bash
echo -e "\n"
```

CONTADOR DE ARGUMENTOS

#-----

```
if [ $# -eq 0 ] || [ $# -ge 4 ];
then
    echo "Debe introducir el directorio y opcionalmente un umbral inferior y otro superior."
    echo -e "\n"
    exit 1
else
    if [ $# -eq 1 ];
    then
        umbral1=10000
        umbral2=100000
    else
        if [ $# -eq 3 ];
        then
            umbral1=$2
            umbral2=$3
        fi
    fi
fi
```

#CREAR CARPETAS

#-----

```
if [ -e pequenos ] || [ -e medianos ] || [ -e grandes ]
then
    echo "Las carpetas de salida ya existen, se va a proceder a borrarlas..."
    rm -d -r pequenos
    rm -d -r medianos
    rm -d -r grandes
fi
```

```
echo "Creando las carpetas pequeños, medianos y grandes..."
mkdir pequenos
mkdir medianos
mkdir grandes
```

COPIANDO ARCHIVOS

```
#-----
```

```
echo "Copiando los archivos..."
```

```
for x in $(find $1)
do
    tam=$(stat -c "%s" $x)

    if [ $tam -lt $umbral1 ];
    then
        if [ -f $x ];
        then
            cp $x pequenos
        fi
    else
        if [ $tam -gt $umbral2 ];
        then
            if [ -f $x ];
            then
                cp $x grandes
            fi
        else
            if [ -f $x ];
            then
                cp $x medianos
            fi
        fi
    fi

done
fi

echo -e "\n"
```

Ejercicio 4

```
#!/bin/bash
echo -e "\n"
```

CONTADOR DE ARGUMENTOS

#-----

```
if [ $# -ne 0 ];
then
    echo "NO INTRODUCZA ARGUMENTOS"
    echo -e "\n"
    exit 1
fi
```

#CONTADOR ARCHIVOS

#-----

```
for x in `ls -l`
do
    if [[ -f $x ]]
    then
        let nArchivos+=1
    fi
done

echo "Número de archivos: "$nArchivos""
echo
```

#USUARIOS LOGEADOS

#-----

```
echo "Lista de usuarios logeados en este instante: "
who | awk '{print $1}' | sort -k1 | uniq
echo
```

#CONTADOR DE CARACTERES

#-----

```
echo "Introduzca caracter a contar."
read -t5 caracter
```

```
if [[ -z $caracter ]]
then
    caracter=a
fi
```

```
for x in `ls -R`
do
    if [[ $x =~ $caracter ]]
    then
        let nCaracter+=1
    fi
```

```
done
```

```
echo "El caracter '$character' aparece $nCaracter veces en los ficheros o carpetas del directorio actual."
```

```
echo -e "\n"
```

```
# Ejercicio 5
```

```
#!/bin/bash  
echo -e "\n"
```

```
#CONTADOR ARGUMENTOS
```

```
#-----
```

```
if [ $# -eq 0 ];  
then  
    echo "Necesario introducir fichero o directorio para comprimir."  
    echo -e "\n"  
    exit 1  
fi
```

```
# BORRAR ARCHIVOS
```

```
#-----
```

```
if [ -d $HOME/Copia ]  
then  
    find $HOME/Copia/ -type f | while read x  
    do  
        seg=$(date +%s)  
        modif=$(stat -c "%Y" $x)  
  
        diferencia=$((seg - modif))  
  
        if [ $diferencia -gt 200 ];  
        then  
            rm $x  
            echo "Se ha borrado $x, llevaba $diferencia segundos sin modificarse"  
        fi  
    done  
fi
```

```
# ARCHIVOS VALIDOS
```

```
#-----
```

```
nExist=0

for x
do
    if [ -e $x ];
    then
        fecha=$(date +%s)

        Array[$nExist]="$x"
        let nExist=$nExist+1

    else
        echo "El argumento <$x> no existe."
    fi
done

if [ $nExist -eq 0 ]
then
    echo "No se ha introducido ningún argumento válido. Terminamos la ejecución."
    exit 1
fi
```

PROCESO

#-----

```
tar -czf copia-$USER-$fecha.tar.gz ${Array[*]}
echo "Comprimiendo los archivos..."

if [ ! -e $HOME/Copia ]
then
    echo "Creando la carpeta de copia..."
    mkdir $HOME/Copia
fi

mv copia-$USER-$fecha.tar.gz $HOME/Copia/

echo "La copia de seguridad se ha creado correctamente."
echo -e "\n"
```


Ejercicio 6

```
#!/bin/bash
echo -e "\n"
```

```
function recorrerDirectorio ()
{
    find "$1" -maxdepth 1 | while read f
    do
        if [ ! -d "$f" ]
        then
            echo -n "<ul>" >> "$FicheroHTML"
            echo -n "<li>$f</li>" >> "$FicheroHTML"
            echo "</ul>" >> "$FicheroHTML"

        else
            if [ "$1" != "$f" ]
            then
                echo "<ul>" >> "$FicheroHTML"
                echo "<li><strong>$f</strong></li>" >> "$FicheroHTML"
                recorrerDirectorio "$f"

                echo "</ul>" >> "$FicheroHTML"

            fi
        fi
    done
}
```

CONTADOR DE ARGUMENTOS

```
# -----
```

```
if [ $# -ne 1 ]
then
    echo "ERROR: Debe introducir $0 <nombre_directorio>"
    echo -e "\n"
    exit 1
fi

if [ ! -d "$1" ]
then
    echo "El argumento <$1> no es un directorio"
    echo -e "\n"
    exit 1
fi
```

PROCESAMIENTO

#-----

```
Fichero=$(readlink -f "$1")
```

```
FicheroHTML="$(basename "$Fichero").html"
```

```
echo "Generando el listado de la carpeta &(basename "$Fichero") sobre el fichero  
$FicheroHTML..."
```

```
cat > "$FicheroHTML" << EOF  
<html>  
<head>  
<title>Listado del directorio $(basename "$Fichero")</title>  
</head>  
<body>  
<h1>Listado del directorio $(basename "$Fichero")</h1>  
EOF
```

```
recorrerDirectorio "$Fichero"
```

```
cat >> "$FicheroHTML" << EOF  
</body>  
</html>  
EOF
```

```
echo "Terminado!"  
echo -e "\n"
```