



# Introducción a los Sistemas Operativos

Prueba final de prácticas

Parte I: guiones shell (5 puntos)

3 de noviembre de 2023

Grupo y subgrupo:	Laboratorio:	Ordenador: <i>Fila (1, 2, 3, 4): Col (izq., der.): Puesto (izq., centro, der.):</i>
DNI:	Nombre:	Apellidos:

**Instrucciones (MUY IMPORTANTE):**

- Tienes **2 horas** para hacer el examen. Si decides no hacerlo o terminas antes de tiempo, no podrás abandonar el laboratorio hasta que se te indique.
- El enunciado del examen tendrás que entregarlo al profesor al finalizar el tiempo. **No olvides anotar en la cabecera tu nombre, DNI y grupo.**
- El examen se realizará utilizando la máquina virtual. Solo podrás consultar las páginas de manual y los boletines de prácticas proporcionados por el profesor (no habrá conexión a Internet durante el examen).
- Ejecuta la orden `uploader` para descargarte el material necesario (este enunciado, los boletines de prácticas y los ficheros/directorios de prueba). Todo este material se descargará en el directorio `/home/alumno/Recursos`.
- Tienes que entregar un fichero de texto por cada ejercicio con la solución del mismo.
- Una vez finalizado el examen, verifica que los ficheros que deseas subir se encuentran en el directorio `/home/alumno/ExamenISO`. A continuación, ejecuta la orden `uploader` y sigue las instrucciones indicadas para realizar la entrega. **Recuerda escribir en tu hoja de examen el código que te ha devuelto la orden.**
- Si deseas hacer alguna observación, añádela como comentario dentro de los ficheros que subas.

El examen se compone de 5 ejercicios independientes, cada uno de los cuales consistirá en la implementación de un pequeño guion shell concreto. El nombre que necesariamente se dará a cada guion shell se indica en el ejercicio correspondiente (por ejemplo, `calculatotalcompra.sh`). En cada ejercicio debes hacer exactamente lo que se pide (no se valorará la funcionalidad extra que no hubiese sido pedida explícitamente).

**IMPORTANTE:** Los ejercicios 1 y 2 son considerados de resolución sencilla. Para que el resto del examen sea evaluado y calificado, es necesario realizar correctamente, al menos, uno de esos dos guiones (su ejecución debe dar el mismo resultado que los ejemplos que se aportan y, obviamente, ser aplicable a otros posibles ejemplos).

1. (0,5 puntos) Implementa un guion shell llamado `calculatotalcompra.sh` que recibe como parámetro un fichero. Cada una de las líneas de texto que contiene este fichero de entrada se corresponde con la descripción de un producto que hemos comprado, indicando, con una única palabra, el nombre del producto y, a continuación, tras un espacio en blanco, su importe en euros. El guión debe mostrar un mensaje indicando el importe total de las compras especificadas en este fichero. No se debe comprobar el parámetro de entrada ni su contenido (supondremos que siempre es correcto). Un ejemplo de su funcionamiento sería:

```
$ cat ficherocompra.txt
ordenador 600 euros
impresora 250 euros
monitor 300 euros
mesa 100 euros
silla 150 euros
lampara 20 euros

$ bash calculatotalcompra.sh ficherocompra.txt
El coste total de la compra realizada es 1420 euros
```

Fichero de solución: **`calculatotalcompra.sh`**

2. (0,75 puntos) Implementa un guion shell llamado `totaltamafic.sh` que recibe como primer parámetro un directorio y como segundo parámetro el nombre de un usuario. El guión debe mostrar por pantalla la suma de los tamaños de todos los ficheros regulares de dicho directorio (incluidos sus subdirectorios) que pertenezcan a dicho usuario. Cualquier error que pudiese producirse durante la exploración del directorio (por falta de permisos) habrá de ser suprimido (no debe aparecer en pantalla). No se deben comprobar los parámetros de entrada (supondremos que siempre son correctos). Un ejemplo de su funcionamiento sería:

```
$ bash totaltamafic.sh /etc root
El tamaño total de los ficheros del directorio /etc que pertenecen al usuario root es:
26851062 bytes

$ bash totaltamafic.sh /run alumno
El tamaño total de los ficheros del directorio /run que pertenecen al usuario alumno es:
3063 bytes
```

Fichero de solución: **totaltamafic.sh**

3. (1 punto) Implementa un guion shell llamado `corrigecontas.sh` que recibe como parámetro un fichero. Cada línea de este fichero tiene el formato `expresion=resultado`, donde `expresion` sigue la misma sintaxis que las expresiones aritméticas utilizadas para la orden `let`, mientras que `resultado` es un número entero. Además, considera que no hay espacios en blanco en ningún sitio de estas líneas de texto.

El script debe mostrar el contenido del fichero, añadiendo, a continuación de cada línea, un mensaje para indicarle al usuario si el resultado dado en la línea es correcto o no y, en este último caso, además informarle de cuál habría sido el resultado correcto. No se debe comprobar el parámetro de entrada ni su contenido (supondremos que sigue la sintaxis correcta). El fichero puede tener líneas completamente en blanco. Por ejemplo:

```
$ cat ficherocontas.txt
2*3+4=10

2*(3+4)=10

2*(3+4)=14
8-10=-4
5%3=2

$ bash corrigecontas.sh ficherocontas.txt
2*3+4=10 -> El resultado dado es correcto
2*(3+4)=10 -> El resultado dado no es correcto. El resultado correcto sería: 14
2*(3+4)=14 -> El resultado dado es correcto
8-10=-4 -> El resultado dado no es correcto. El resultado correcto sería: -2
5%3=2 -> El resultado dado es correcto
```

Fichero de solución: **corrigecontas.sh**

4. (1,25 puntos) Implementa un guion shell llamado `catenesimo.sh` que recibe al menos dos parámetros. El primer parámetro debe ser un número natural positivo, es decir, un número entero sin signo mayor que cero. El resto de parámetros deben ser ficheros regulares. Si el primer parámetro es igual a  $N$ , entonces el guión mostrará en pantalla el contenido del  $N$ -ésimo fichero de entre los pasados como parámetros (empezando a contar estos ficheros por 1). Así, por ejemplo, la siguiente ejecución mostraría en pantalla el contenido del fichero `agenda/martes.txt`:

```
$ bash catenesimo.sh 2 /home/alumno/t1.txt agenda/martes.txt configuracion.txt
```

Se debe comprobar que todos los parámetros son correctos, es decir, que al menos hay 2 parámetros, que el primer parámetro es un número natural positivo menor que el número total de parámetros y que el resto de parámetros corresponden todos ellos a ficheros regulares con permiso de lectura. En caso contrario, este es el listado de códigos de salida y de mensajes de error que se deben producir:

- 1: El número mínimo de parámetros es 2.
- 2: El primer parametro deber ser un número natural positivo.
- 3: El primer parametro debe ser menor que el número total de parámetros.
- 4: El parámetro <fichero> debe ser el nombre de un fichero regular.
- 5: El parámetro <fichero> debe tener permiso de lectura.

A continuación se muestra un ejemplo de ejecuciones que producirían algún tipo de error y una ejecución correcta:

```
$ bash catenesimo.sh tres
El número mínimo de parámetros es 2
USO: catenesimo <numero> <fichero>...
$ echo $?
1
$ bash catenesimo.sh dos saludos.txt fichero compra.txt cancion.txt
El primer parametro debe ser un numero natural positivo (entero mayor que 0, sin signo)
USO: catenesimo <numero> <fichero>...
$ echo $?
2
$ bash catenesimo.sh 0 saludos.txt fichero compra.txt cancion.txt
El primer parametro debe ser un numero natural positivo (entero mayor que 0, sin signo)
USO: catenesimo <numero> <fichero>...
$ echo $?
2
$ bash catenesimo.sh 5 saludos.txt fichero compra.txt cancion.txt
El primer parametro debe ser un número menor que el número total de parámetros
USO: catenesimo <numero> <fichero>...
$ echo $?
3
$ bash catenesimo.sh 3 saludos.txt /home/alumno fichero compra.txt cancion.txt
El parametro '/home/alumno' debe ser el nombre de un fichero regular
USO: catenesimo <numero> <fichero>...
$ echo $?
4
$ bash catenesimo.sh 3 saludos.txt /etc/shadow fichero compra.txt cancion.txt
El parametro '/etc/shadow' debe tener permiso de lectura
USO: catenesimo <numero> <fichero>...
$ echo $?
5
$ bash catenesimo.sh 2 saludos.txt fichero compra.txt cancion.txt
ordenador 600 euros
impresora 250 euros
monitor 300 euros
mesa 100 euros
silla 150 euros
lampara 20 euros
```

Fichero de solución: **catenesimo.sh**

5. (1,5 puntos) Implementa un guion shell llamado `cuentapalabraficheros.sh` que recibe al menos dos parámetros. El primer parámetro es una palabra y el resto de parámetros son ficheros regulares. El script debe mostrar el número de apariciones de la palabra dada (sin distinguir entre mayúsculas y minúsculas) en cada uno de estos ficheros, mostrando esta información por orden creciente de dicho número de apariciones. Al final, el script debe mostrar también el número total de apariciones de la palabra en el conjunto de ficheros.

Los ficheros contienen solamente palabras formadas únicamente por caracteres alfabéticos. Estas palabras están separadas por uno o varios espacios en blanco y sin signos de puntuación de ningún tipo. No se deben comprobar los parámetros de entrada (supondremos que siempre son correctos). Por ejemplo:

```
$ cat saludos.txt
hola
adios
HOLA hola
hasta luego
hola adios
holastico
adios adios
HOLA
adios hola hola adios
```

```
$ cat cancion.txt
hola don pepito
hola don jose
paso usted por mi casa
por su casa yo pase
adios don pepito
adios don jose
```

```
$ bash cuentapalabraficheros.sh hola saludos.txt cancion.txt
La palabra "hola" aparece 2 veces en el fichero "cancion.txt"
La palabra "hola" aparece 7 veces en el fichero "saludos.txt"
Numero total de veces que aparece la palabra "hola" en el conjunto de ficheros es: 9
```

Fichero de solución: **`cuentapalabraficheros.sh`**

# Soluciones

Se muestran a continuación las soluciones de los distintos ejercicios:

1. `#!/bin/bash`  
`# calculatotalcompra.sh`

```
total=0
while read linea
do
    precio=$(echo $linea | cut -f2 -d" ")
    let total+=precio
done < $1
echo "El coste total de la compra realizada es $total euros"
```

Script 1: calculatotalcompra.sh

2. `#!/bin/bash`  
`# totaltamafic.sh`

```
total=0
find $1 -type f -user $2 -printf "%s\n" 2>/dev/null |
(
    while read linea
    do
        let total+=$linea
    done;
    echo "El tamaño total de los ficheros del directorio $1 que pertenecen al
        usuario $2 es:"
    echo "$total bytes"
)
```

Script 2: totaltamafic.sh

3. `#!/bin/bash`  
`# corrigecuentas.sh`

```
while read linea
do
    if ! test -z $linea
    then
        echo -n "$linea -> "
        expresion=$(echo $linea | cut -d"=" -f1)
        resultado=$(echo $linea | cut -d"=" -f2)
        let resultadocorrecto=$expresion
```

```

        if test $resultado -eq $resultadocorrecto
        then
            echo "El resultado dado es correcto"
        else
            echo "El resultado dado no es correcto. El resultado correcto sería: $resultadocorrecto"
        fi
    fi
done < $1

```

### Script 3: corrigecuentas.sh

4. `#!/bin/bash`  
`# catenesimo.sh`

```

if test $# -lt 2
then
    echo "El numero minimo de parametros es 2" >&2
    echo "USO: $0 <numero> <fichero>..." >&2
    exit 1
fi

```

```

if ! echo $1 | grep -qx "[0-9]\+"
then
    echo "El primer parametro debe ser un numero natural positivo (entero mayor que 0, sin signo)" >&2
    echo "USO: $0 <numero> <fichero>..." >&2
    exit 2
fi

```

```

if ! test $1 -gt 0
then
    echo "El primer parametro debe ser un numero natural positivo (entero mayor que 0, sin signo)" >&2
    echo "USO: $0 <numero> <fichero>..." >&2
    exit 2
fi

```

```

if test $1 -ge $#
then
    echo "El primer parametro debe ser un numero menor que el numero total de parametros" >&2
    echo "USO: $0 <numero> <fichero>..." >&2
    exit 3
fi

```

```

numero=$1
shift

```

```

for f in $@
do

```

```

if test ! -f $f
then
    echo "El parametro '$f' debe ser el nombre de un fichero regular" >&2
echo "USO: $0 <numero> <fichero>..." >&2
    exit 4
fi
if test ! -r $f
then
    echo "El fichero '$f' debe tener permiso de lectura" >&2
echo "USO: $0 <numero> <fichero>..." >&2
    exit 5
fi
done

fichero=${!numero}
echo "El contenido del fichero '$fichero' es:"
cat $fichero

```

Script 4: catenesimo.sh

5. `#!/bin/bash`  
`# cuentapalabraficheros.sh`

```

palabra=$1
shift
ficheros="$@"
total=0

for fichero in $ficheros
do
    apariciones=$(grep -woi $palabra $fichero | wc -l)
    echo "$fichero:$apariciones"
done |
sort -t":" -k2n |
(
    while read linea
    do
        fichero=$(echo $linea | cut -d":" -f1)
        apariciones=$(echo $linea | cut -d":" -f2)
        let total+=$apariciones
        echo "La palabra \"$palabra\" aparece $apariciones veces en el
            fichero \"$fichero\""
    done;
    echo -e "Numero total de veces que aparece la palabra \"$palabra\" en el
        conjunto de ficheros es: $total\n\n"
)

```

Script 5: cuentapalabraficheros.sh