



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. El nombre que debes dar a este fichero aparece al final del ejercicio correspondiente.
- 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, `validadirectorios.sh`). En cada ejercicio debes hacer exactamente lo que se pide (no se valorará la funcionalidad extra que no hubiese sido pedida explícitamente).

1. (1,0 punto) Implementa un guion shell llamado `validadirectorios.sh` que tiene que recibir uno o más nombres de ficheros como parámetros. El guion debe comprobar si los ficheros existen, son de tipo directorio y disponen de permisos `r` y `x` para el usuario que ejecuta el guion shell, en cuyo caso, debe devolver 0 como código de salida. En caso contrario, es decir, si hay al menos un parámetro que no cumple esas condiciones, debe devolver un código de salida 1 y mostrar por su salida de error estándar un mensaje de error y la lista de parámetros erróneos.

Si el número de parámetros es cero, el guion también mostrará un mensaje de error, pero devolverá 2 como código de salida.

A continuación se muestran varios ejemplos de ejecución de este guion:

```
[alumno@localhost ~]$ bash validadirectorios.sh
Uso: validadirectorios.sh directorio...
[alumno@localhost ~]$ echo $?
2
[alumno@localhost ~]$ bash validadirectorios.sh . /usr/ /etc/ /root noexiste
Los siguientes parámetros no son directorios o no tienen permisos r y x:
/root
noexiste
[alumno@localhost ~]$ echo $?
1
[alumno@localhost ~]$ bash validadirectorios.sh . /usr/ /etc/
[alumno@localhost ~]$ echo $?
0
```

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

2. (1,0 punto) Implementa un guion shell llamado `usodirectorio.sh` que debe recibir dos parámetros, un directorio y un número entero n mayor o igual que 0. El guion mostrará por pantalla los n ficheros regulares más grandes presentes en dicho directorio, ordenados de mayor a menor tamaño. Eso sí, si en el directorio hay menos de n ficheros regulares, se mostrarán los datos de todos los que tenga. Tras la lista de ficheros, el guion también debe mostrar el tamaño total (en bytes) de los ficheros listados.

El guion no debe comprobar los parámetros (supondremos que son siempre correctos) ni debe incluir los subdirectorios del directorio dado.

A continuación se muestran varios ejemplos de ejecución de este guion:

```
[alumno@localhost ~]$ bash usodirectorio.sh /etc/ssh/ 2
Fichero: moduli, tamaño: 525809 bytes
Fichero: sshd_config, tamaño: 3676 bytes
Tamaño total de los 2 ficheros regulares mostrados: 529485 bytes
[alumno@localhost ~]$ bash usodirectorio.sh /etc/ssh/ 10
```

```
Fichero: moduli, tamaño: 525809 bytes
Fichero: sshd_config, tamaño: 3676 bytes
Fichero: ssh_config, tamaño: 1874 bytes
Tamaño total de los 3 ficheros regulares mostrados: 531359 bytes
```

Observa que en el segundo caso se piden los 10 ficheros regulares más grandes, pero el directorio solo cuenta con 3.

Fichero de solución: **usodirectorio.sh**

3. (1,0 punto) Implementa un guion shell llamado `operaciones.sh` con la siguiente funcionalidad. El guion debe recibir un fichero de texto con diversas operaciones matemáticas, una por línea. Las operaciones permitidas son: suma («+»), resta («-»), multiplicación («*»), división entera («/») y resto o módulo («%»). El guion debe mostrar por pantalla el resultado de dichas operaciones según el ejemplo que aparece a continuación.

El guion no debe comprobar los parámetros (supondremos que son siempre correctos) ni el contenido del fichero dado (que también supondremos siempre correcto).

Es importante observar que el fichero con operaciones no contiene espacios en blanco en ningún sitio, mientras que la salida del guion mostrando el resultado de las operaciones sí (en concreto, hay un espacio antes y después del operador, y antes y después del símbolo «=»).

A continuación se muestra un ejemplo de ejecución de este guion:

```
[alumno@localhost ~]$ cat ejemplo.txt
23*2
23/2
23%2
23+2
23-2
1991/3
1991%3
[alumno@localhost ~]$ bash operaciones.sh ejemplo.txt
23 * 2 = 46
23 / 2 = 11
23 % 2 = 1
23 + 2 = 25
23 - 2 = 21
1991 / 3 = 663
1991 % 3 = 2
```

Fichero de solución: **operaciones.sh**

4. (1,0 punto) Implementa un guion shell llamado `cuentalettras.sh` que reciba como parámetro un único fichero de texto y como resultado muestre una lista de todas las letras del abecedario contenidas en el fichero (una por línea) ordenadas de mayor a menor frecuencia de aparición. La frecuencia, es decir, el número de veces que una letra aparece, debe mostrarse tras la letra y entre paréntesis (ver ejemplo).

El guion no debe comprobar los parámetros (supondremos que son siempre correctos) ni el contenido del fichero (que también supondremos siempre correcto, sin vocales acentuadas). Además, el guion no distinguirá entre mayúsculas y minúsculas a la hora de contar las apariciones de una letra.

A continuación se muestra un ejemplo de ejecución de este guion:

```
[alumno@localhost ~]$ cat nucleo_de_linux.txt
Nucleo de Linux (Wikipedia)

Linux es un nucleo mayormente libre semejante al nucleo de Unix, uno de los
principales ejemplos de software libre y de codigo abierto. Esta licenciado
bajo la GPL v2 salvo el hecho que tiene blobs binarios no-libres
y la mayor parte del software incluido en el paquete que se distribuye en su
sitio web es software libre. Esta desarrollado por colaboradores de todo el
mundo. El desarrollo del dia a dia tiene lugar en la Linux Kernel Mailing List
Archive.
[alumno@localhost ~]$ bash cuentalettras.sh nucleo_de_linux.txt
e (59)
l (37)
o (34)
i (33)
a (32)
r (24)
n (24)
s (23)
...
f (3)
k (2)
```

Fichero de solución: **cuentalettras.sh**

5. (1,0 punto) Implementa un guion shell llamado `accesoaficheros.sh` que reciba como parámetros un directorio seguido de una lista de usuarios y devuelva, para cada usuario de la lista, el número de ficheros de cualquier tipo que le pertenezcan a partir de ese directorio (incluyendo subdirectorios) y para los que, además, tenga algún permiso activo («r», «w» o «x») como propietario.

El guion no debe comprobar los parámetros dados (supondremos que son siempre correctos). Ahora bien, si durante la búsqueda de ficheros en el directorio se producen errores, el guion deberá mostrar también un mensaje reflejando este hecho (el mensaje debe aparecer por la salida de error estándar) y devolver 1 como código de salida. Si no se produce ningún error, no deberá mostrarse ningún mensaje al respecto y el código de salida será 0.

A continuación se muestran algunos ejemplos de ejecución de este guion:

```
[alumno@localhost ~]$ bash accesoaficheros.sh /home/alumno/ alumno root polkitd
Usuario 'alumno': 456 ficheros
Usuario 'root': 0 ficheros
Usuario 'polkitd': 0 ficheros
[alumno@localhost ExamenISO]$ echo $?
0
[alumno@localhost ~]$ bash accesoaficheros.sh /etc alumno root polkitd
Usuario 'alumno': 0 ficheros
Usuario 'root': 2152 ficheros
Usuario 'polkitd': 1 ficheros
Hay ficheros de '/etc' a los que no se ha podido acceder
[alumno@localhost ExamenISO]$ echo $?
1
```

Fichero de solución: **accesoaficheros.sh**

Soluciones

Se muestran a continuación las soluciones de los distintos ejercicios:

1. `#!/bin/bash`

```
test $# -eq 0 && echo "Uso: $0 directorio..." >&2 && exit 2
salida=0
for d in "$@"
do
    if test ! -d "$d" -o ! -r "$d" -o ! -x "$d"
    then
        if test $salida -eq 0
        then
            echo Los siguientes parámetros no son directorios o no tienen permisos r y
                x: >&2
            salida=1
        fi
        echo $d >&2
    fi
done

exit $salida
```

Script 1: validadirectorios.sh

2. `#!/bin/bash`

```
tamficheros=0
num=0
find "$1" -maxdepth 1 -type f -printf "%f:%s\n" | sort -k2,2nr -t: | head -n $2 | (
    while read linea
    do
        fichero=$(echo $linea | cut -f 1 -d :)
        tama=$(echo $linea | cut -f 2 -d :)
        echo "Fichero: $fichero, tamaño: $tama bytes"
        let tamficheros+=tama
        let num+=1
    done
    echo "Tamaño total de los $num ficheros regulares mostrados: $tamficheros bytes" )
```

Script 2: usodirectorio.sh

3. `#!/bin/bash`

```
while read linea
do
    for operacion in + - / "%" "*"
    do
        if echo $linea | grep -q "$operacion"
        then
            operando1=$(echo $linea | cut -f1 -d "$operacion")
            operando2=$(echo $linea | cut -f2 -d "$operacion")
            let resultado=$operando1$operacion$operando2
            echo "$operando1 $operacion $operando2 = $resultado"
        fi
    done
done < $1
```

Script 3: operaciones.sh

4. `#!/bin/bash`

```
grep -o "[[:alpha:]]" "$1" | sort | uniq -ci | sort -nr | while read linea
do
    cantidad=$(echo $linea | cut -f1 -d " ")
    letra=$(echo $linea | cut -f2 -d " ")
    echo "$letra ($cantidad)"
done
```

Script 4: cuentalettras.sh

5. `#!/bin/bash`

```
fichtemp=$(mktemp)
dir=$1
shift
for usuario in "$@"
do
    numficheros=$(find $dir -user $usuario -perm /700 2> $fichtemp | wc -l)
    echo "Usuario '$usuario': $numficheros ficheros"
done

if test -s $fichtemp
then
    echo "Hay ficheros de '$dir' a los que no se ha podido acceder" >&2
    exit 1
fi
exit 0
```

Script 5: accesoaficheros.sh