



Grupo y subgrupo:	Laboratorio:	Ordenador:
DNI:	Nombre:	<i>Fila:</i> <i>Col:</i> izq. der. <i>Puesto:</i> izq. centro der.
		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 física. 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 consta de 6 ejercicios independientes, cada uno de los cuales consistirá en la implementación de un pequeño guion shell concreto. El nombre que se dará al guion shell será el del ejercicio correspondiente, tal y como se establece en cada caso (ej. «ejercicio1.sh»). En cada ejercicio debes hacer exactamente lo que se pide (no se valorará la funcionalidad extra que no hubiese sido pedida explícitamente).

- (0,5 puntos) Implementa un guion shell que reciba como primer y único parámetro el nombre de un fichero regular que debe existir y se debe poder leer, en cuyo caso se devolverá el código de salida «0» sin hacer nada más. En caso contrario, se generará un mensaje de error y se devolverá el código de error correspondiente. A continuación se muestran varios ejemplos de ejecución:

```
[alumno@localhost ExamenISO]$ bash ejercicio1.sh ejercicio1.sh
[alumno@localhost ExamenISO]$ echo $?
0
[alumno@localhost ExamenISO]$ bash ejercicio1.sh
Uso: ejercicio1.sh fichero
[alumno@localhost ExamenISO]$ echo $?
1
[alumno@localhost ExamenISO]$ bash ejercicio1.sh fichero_no_existe
Error: Fichero 'fichero_no_existe' no existe o no se puede leer.
[alumno@localhost ExamenISO]$ echo $?
2
[alumno@localhost ExamenISO]$ bash ejercicio1.sh /etc/shadow
Error: Fichero '/etc/shadow' no existe o no se puede leer.
[alumno@localhost ExamenISO]$ echo $?
2
```

Fichero de solución: **ejercicio1.sh**

- (0,75 puntos) Implementa un guion shell que reciba como parámetros uno o más directorios distintos que deben existir y se deben poder leer, y que muestre un listado de los directorios recibidos, en el que aparecerá el nombre de cada directorio junto con «OK» si este existe y se puede leer, o «KO» si no se cumple lo anterior. Si todos los parámetros son directorios que existen y se pueden leer, el guion devolverá el código de salida «0». En caso contrario, se mostrará un mensaje de error y se devolverá el código de error «4». Supón que siempre se recibe, al menos, un parámetro y que no se dan directorios repetidos. A continuación se presenta la salida producida por el guion pedido:

```
[alumno@localhost ExamenISO]$ bash ejercicio2.sh /etc/xdg /etc/ssh /usr/lib/alsa/init /usr/lib64/pm-utils
/etc/xdg OK
/etc/ssh OK
/usr/lib/alsa/init OK
/usr/lib64/pm-utils OK

[alumno@localhost ExamenISO]$ echo $?
0

[alumno@localhost ExamenISO]$ bash ejercicio2.sh /etc/xdg /etc/ssh /dir/no/existe /usr/lib64/pm-utils /etc/grub.d
/etc/xdg OK
/etc/ssh OK
/dir/no/existe KO (no existe)
```

```
/usr/lib64/pm-utils OK
/etc/grub.d KO (no se puede leer)
```

```
Error: Hay directorios que no existen o no se pueden leer
```

```
[alumno@localhost ExamenISO]$ echo $?
4
```

Fichero de solución: **ejercicio2.sh**

3. (1 punto) Considera un guion shell que recibe como único parámetro (no hay que hacer ninguna comprobación del parámetro) el nombre de un fichero cuyo contenido es una o más rutas (tanto absolutas como relativas) de directorio. Cada ruta estará en una línea aparte y no habrá espacios en los nombres de los directorios que la configuran. El guion elaborará un listado numerado y ordenado en orden alfabético creciente con las rutas absolutas (es decir, aquellas que comienzan por el carácter «/») distintas (es decir, si una misma ruta absoluta aparece varias veces¹ se incluirá una sola vez en el listado) y mostrará al final el número total de rutas absolutas distintas que ha encontrado. A continuación se presenta la salida producida por el guion pedido:

```
[alumno@localhost ExamenISO]$ cat ../Recursos/dir_ej3.txt
/etc/xdg
/etc/ssh
./usr/bin
/usr/lib/alsa/init
/etc/ssh
../home/alumno
/usr/lib64/pm-utils
/etc/xdg
/etc/ssh
[alumno@localhost ExamenISO]$ bash ejercicio3.sh ../Recursos/dir_ej3.txt
Rutas absolutas encontradas:
1: /etc/ssh
2: /etc/xdg
3: /usr/lib64/pm-utils
4: /usr/lib/alsa/init
Total: 4 rutas
```

Fichero de solución: **ejercicio3.sh**

4. (0,75 puntos) Implementa un guion shell que reciba como único parámetro (no hay que hacer ninguna comprobación del parámetro) el nombre de un fichero que debería contener uno o más directorios expresados solamente con sus rutas absolutas (es decir, comenzarán necesariamente por el carácter «/»), y que los listará en orden alfabético creciente. En caso de que haya elementos que no sean rutas absolutas, el guion devolverá el código de error 3, e informará de la lista de estos elementos (en el orden en el que aparezcan en el fichero). Supón que las rutas están en una o varias líneas dentro del fichero, es decir es posible que en la misma línea aparezcan varias rutas (absolutas o no) separadas por uno o varios espacios, y que no hay rutas repetidas dentro del fichero. Supón además que no hay espacios en los nombres de los directorios de las rutas. A continuación se presenta la salida producida por el guion pedido:

```
[alumno@localhost ExamenISO]$ cat ../Recursos/dir_ej4_1.txt
/etc/xdg          /etc/ssh
/usr/lib/alsa/init  home/alumno  bin
/usr/lib64/pm-utils  ../etc/cups
usr/lib64/

[alumno@localhost ExamenISO]$ bash ejercicio4.sh ../Recursos/dir_ej4_1.txt
Error: directorios no válidos en '../Recursos/dir_ej4_1.txt':
home/alumno
bin
../etc/cups
usr/lib64/

[alumno@localhost ExamenISO]$ echo $?
3

[alumno@localhost ExamenISO]$ cat ../Recursos/dir_ej4_2.txt
/etc/xdg          /etc/ssh
/usr/lib/alsa/init  /home/alumno  /bin
/usr/lib64/pm-utils  /etc/cups

[alumno@localhost ExamenISO]$ bash ejercicio4.sh ../Recursos/dir_ej4_2.txt
Rutas absolutas encontradas:
/bin
/etc/cups
```

¹Por simplicidad, el guion podrá considerar como distintas dos rutas absolutas que únicamente se diferencian en que una acaba con el carácter «/» y la otra sin él.

```
/etc/ssh
/etc/xdg
/home/alumno
/usr/lib64/pm-utils
/usr/lib/alsa/init
```

```
[alumno@localhost ExamenISO]$ echo $?
0
```

Fichero de solución: **ejercicio4.sh**

5. (1 punto) Implementa un guion shell que reciba como único parámetro el nombre de un directorio que existe y se puede leer (no habrá que hacer comprobaciones del parámetro), y que calcule el tamaño aproximado del mismo, entendido este como la suma de los tamaños de todos los ficheros regulares que contiene (considerando también los de los subdirectorios que pudiese tener). En concreto el tamaño aproximado se dará en bytes si este es menor de 1024 bytes, o KiB en caso contrario. Cualquier error que pudiese producirse durante la exploración del directorio (por falta de permisos) habrá de ser suprimido. A continuación se presenta la salida producida por el guion pedido:

```
[alumno@localhost ExamenISO]$ bash ejercicio5.sh /etc/alsa
/etc/alsa: 458 bytes
```

```
[alumno@localhost ExamenISO]$ bash ejercicio5.sh /etc/ssh
/etc/ssh: 519 KiB
```

Fichero de solución: **ejercicio5.sh**

6. (1 punto) Implementa un guion shell que reciba como parámetros uno o más directorios distintos que existen y se pueden leer (no hay que hacer ninguna comprobación de los parámetros), y que liste los ficheros regulares que hay en cada directorio (sin considerar los que pudiese haber en los posibles subdirectorios que tuviese). En concreto, en el listado se numerarán los directorios y por cada uno de ellos se mostrarán los nombres de los ficheros regulares (sin la ruta), seguidos de su tamaño en bytes. En el listado, los directorios aparecerán ordenados por orden alfabético ascendente, mientras que los ficheros de cada directorio estarán ordenados en orden descendente de tamaño. Si algún directorio no contiene ficheros regulares se mostrará igualmente su nombre. A continuación se presenta la salida producida por el guion pedido:

```
[alumno@localhost ExamenISO]$ bash ejercicio6.sh /etc/ssh /home/alumno/Descargas /usr/lib/alsa/init /etc/xdg
1) /etc/ssh:
    moduli 525809 bytes
    sshd_config 3676 bytes
    ssh_config 1874 bytes
2) /etc/xdg:
    user-dirs.defaults 418 bytes
    user-dirs.conf 414 bytes
3) /home/alumno/Descargas:
4) /usr/lib/alsa/init:
    default 10866 bytes
    test 10699 bytes
    00main 1808 bytes
    hda 1494 bytes
    ca0106 1341 bytes
    info 932 bytes
    help 391 bytes
```

Fichero de solución: **ejercicio6.sh**

Soluciones

Se muestra a continuación las soluciones de los distintos ejercicios:

1. `#!/bin/bash`

```
test $# -ne 1 && echo "Uso: $0 fichero" >&2 && exit 1
test ! -f $1 -o ! -r $1 && echo "Error: Fichero '$1' no existe o no se puede leer" >&2
&& exit 2

exit 0
```

Script 1: ejercicio1.sh

2. `#!/bin/bash`

```
for directorio in $@
do
    if test -d $directorio -a -r $directorio
    then
        echo "$directorio OK"
    elif test ! -d $directorio
    then
        echo "$directorio KO (no existe)"
        error=1
    else
        echo "$directorio KO (no se puede leer)"
        error=1
    fi
done
if test -n "$error"
then
    echo
    echo "Error: Hay directorios que no existen o no se pueden leer" 1>&2
    exit 4
fi
```

Script 2: ejercicio2.sh

3. `#!/bin/bash`

```
echo "Rutas absolutas encontradas:"
cat $1 | tr ' ' '\n' | grep '^/' | sort | uniq | (while read ruta
do
    let numero+=1
    echo "$numero: $ruta"
done; echo "Total: $numero rutas")
```

Script 3: ejercicio3.sh

4. `#!/bin/bash`

```
rutasabsolutas=$(cat $1 | tr ' ' '\n' | grep '^/' | sort)
rutasnoabsolutas=$(cat $1 | tr ' ' '\n' | grep -v '^/' | grep -v '^$')

if test -n "$rutasnoabsolutas"
then
    echo "Error: directorios no válidos en '$1':" >&2
    echo -e "$rutasnoabsolutas" >&2
    exit 3
else
    echo "Rutas absolutas encontradas:"
    echo "$rutasabsolutas"
fi
```

Script 4: ejercicio4.sh

5. `#!/bin/bash`

```
unidades=bytes
total=0
```

```

for tamano in $(find $1 -type f -printf "%s\n" 2> /dev/null)
do
    let total+=tamano
done

if test $total -ge 1024
then
    let total/=1024
    unidades="KiB"
fi
echo "$1: $total $unidades"

```

Script 5: ejercicio5.sh

6. *#!/bin/bash*

```

lista_directorios=$(echo $@ | tr ' ' '\n' | sort)
for directorio in $lista_directorios
do
    let ndir+=1
    echo "$ndir) $directorio:"
    find $directorio -maxdepth 1 -type f -printf "\t%f %s bytes\n" | sort -k2,2nr -t' '
done

```

Script 6: ejercicio6.sh