



Apellidos:	Nombre:
Grupo: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> PCEO	DNI:

Instrucciones (MUY IMPORTANTE):

- Tienes **2 horas** en total para hacer las dos partes del examen práctico (administración y guiones shell).
- La puntuación de cada parte del examen (administración y guiones shell) será de hasta 5 puntos.
- El examen se realizará utilizando la **máquina virtual de Fedora 32** proporcionada por el profesor. Solo podrás consultar la información disponible en la máquina virtual y los boletines de prácticas proporcionados por el profesor.
- Para descargarte el material necesario (este enunciado y los boletines de prácticas), ejecuta por primera vez la orden `uploader` como usuario **alumno**. Todo este material se descargará en el directorio `/home/alumno/RecursosISO`.
- Para la **parte de guiones shell**, tienes que entregar un fichero de texto por cada apartado del ejercicio propuesto con la solución del apartado. El nombre que debes dar a este fichero aparece al final del apartado correspondiente y todos los ficheros deberán estar en el directorio de la máquina virtual `/home/alumno/ExamenISO`.
- Para la **parte de administración**, debes entregar tus soluciones en un fichero de texto llamado **bitacora.txt**. Este fichero se ha descargado en `/home/alumno/RecursosISO` junto con el resto del material del examen y debes copiarlo también al directorio `/home/alumno/ExamenISO` de la máquina virtual. Este fichero debe contener **todas** las órdenes necesarias para resolver los diferentes ejercicios. Además, ten en cuenta que:
 - No puedes utilizar ninguna herramienta gráfica de administración.
 - Debes copiar la salida, si la hubiera, producida por las órdenes que has indicado. Si la salida fuera muy extensa, se puede abreviar mostrando solo la información relevante.
 - **Los datos que aparezcan mágicamente penalizarán.**
 - Se dará respuesta a cada apartado de los ejercicios proporcionando la solución a continuación de la línea identificativa correspondiente. Estas líneas de texto siguen un formato pautado y están ya incorporadas en el fichero **bitacora.txt** que se ha descargado. No cambies ni elimines el contenido de dichas líneas.
 - Si has tenido que modificar *manualmente* algún fichero de configuración, copia en la bitácora el contenido final del fichero o indica claramente qué cambios has hecho y dónde has realizado dichos cambios.
- **Una vez finalizado el examen**, verifica que los ficheros que deseas subir se encuentran en el directorio `/home/alumno/ExamenISO` de la máquina virtual. A continuación, como usuario **alumno**, 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.

Administración (5 puntos)

- (1,2 puntos) Realiza las siguientes tareas sobre gestión de usuarios sin modificar ningún fichero de configuración a mano (salvo que se diga lo contrario), mostrando tanto las órdenes utilizadas como la salida producida por cada una de ellas:
 - (0,12 puntos) Crea el grupo `estudiante`. ¿Qué GID se le ha asignado?
 - (0,18 puntos) Sin crear grupos que no se piden, crea el usuario `alumna` con nombre completo `Alumna de ISO` y grupo primario `estudiante`. Con la orden correspondiente, muestra el UID asignado al nuevo usuario, así como los grupos a los que pertenece.
 - (0,18 puntos) Para el usuario `alumna`, establece como contraseña la cadena `"c4mb14m3"` y, a continuación, fuerza a que tenga que cambiarla tras el primer login. Demuestra que lo que has hecho funciona.
 - (0,18 puntos) Para el usuario `alumno`, fija el grupo `estudiante` como su grupo secundario. Para el usuario `alumna`, crea el grupo `alumna` y establéclo como su grupo secundario. Tanto para `alumno` como para `alumna`, adjunta la salida de la orden que permite consultar los grupos de un usuario.
 - (0,18 puntos) Crea en `/home` el directorio `practicas`. El usuario propietario del mismo será `root` y deberás hacer lo necesario para que los usuarios `alumno` y `alumna` puedan leer y modificar el contenido del directorio y utilizarlo como directorio actual, y que nadie más pueda hacer nada de esto sobre el directorio. Adjunta la salida de `ls -ld /home/practicas`.
 - (0,12 puntos) Haz lo necesario para que a pesar de que tanto `alumno` como `alumna` tienen la posibilidad de modificar el contenido del directorio, uno no pueda borrar los ficheros del otro. Adjunta la salida de `ls -ld /home/practicas`.
 - (0,24 puntos) Haz lo necesario para que cualquier fichero creado en el directorio `/home/practicas` por los usuarios `alumno` y `alumna`, quede con el grupo propietario `estudiante`. A continuación, como usuario `alumno`, indica el grupo activo que tienes y crea después directamente el fichero vacío `bolentines_ISO.pdf` en `/home/practicas` (es decir, no puedes crear primero el fichero y luego cambiarle el grupo propietario). Adjunta la salida de `ls -l /home/practicas`.
- (2,48 puntos) Realiza las siguientes tareas sobre gestión de discos y sistemas de ficheros:

- a) (0,24 puntos) Usa particionado GPT para particionar los discos `sdb`, `sdc` y `sdd` de la máquina virtual, de forma que cada uno de ellos quede con 2 particiones, la primera de 400 MiB y la segunda del resto del espacio disponible. No es necesario indicar todos los pasos, basta con adjuntar la salida de la orden `fdisk` con la opción adecuada para mostrar las particiones de `/dev/sdb`, `/dev/sdc` y `/dev/sdd`.
- b) (0,18 puntos) A partir de la primera partición de cada uno de los discos anteriores, crea el grupo de volúmenes `storage`. ¿Cuántas extensiones físicas contiene en total el grupo de volúmenes? Además de responder a la pregunta, muestra la salida de la orden que da esta información.
- c) (0,48 puntos) Usando el grupo de volúmenes anterior, queremos crear el volumen lógico `lv-disk` del tamaño máximo permitido. Dicho volumen debe ser capaz de ofrecer más rendimiento que el de un único disco, y a la vez, la capacidad de seguir funcionando aún cuando uno de los discos caiga. ¿Qué tipo de volumen lógico has creado (explica por qué cumple las dos características anteriores)? ¿Qué tamaño total tiene en extensiones lógicas? Además de responder a las preguntas, muestra la salida de las órdenes que hayas usado para responder a las cuestiones.
- d) (0,2 puntos) Crea en el volumen lógico anterior un sistema de ficheros `ext4`. ¿Qué tamaño de bloque lógico usa el sistema de ficheros creado? ¿Cuántas copias del superbloque tiene en total el sistema de ficheros? Adjunta la salida de la orden/órdenes que hayas usado para obtener la información pedida.
- e) (0,36 puntos) Haz lo necesario para que cada vez que se arranque el ordenador, el sistema de ficheros del apartado anterior quede montado en el directorio `/home/practicas` y los usuarios `alumno` y `alumna` puedan usarlo de acuerdo con lo establecido en los apartados 1.e y 1.f. Deja el sistema de ficheros montado (demuestra que queda montado con la orden correspondiente)
- f) (0,18 puntos) Como usuario `alumna`, crea un enlace llamado `/home/alumna/mi-enlace` al directorio `/home/practicas`. ¿Qué tipo de enlace has tenido que crear? ¿Qué tamaño tiene el enlace creado? Justifica las respuestas a ambas cuestiones.
- g) (0,24 puntos) De nuevo como usuario `alumna`, copia de forma recursiva y preservando todos los atributos, el contenido del directorio `/var/log` en `/home/practicas`, evitando que se muestre por pantalla cualquier mensaje por falta de permisos que pudiese aparecer durante la copia. ¿Qué espacio total en disco (en MiB) ocupa ahora el directorio `/home/practicas`? ¿Qué porcentaje del espacio del sistema de ficheros está ocupado?
- h) (0,24 puntos) Como usuario `root`, en un fichero llamado `/root/copia-seguridad.cpio`, haz una copia de seguridad usando la orden `cpio`, que contenga todos los ficheros regulares del directorio `/home/practicas` que tengan un tamaño igual o superior a 1 KiB (1024 bytes). La copia de seguridad debe quedar con las rutas absolutas de los ficheros respaldados. Lista el contenido de la copia de seguridad creada. ¿Cuántos ficheros se han incluido en total?
- i) (0,36 puntos) Dado que se va a usar el disco `/dev/sdc` para otra tarea, haz lo necesario para eliminar la partición `/dev/sdc1` del grupo de volúmenes, sin que el volumen lógico deje de funcionar. Obviamente deberás usar para ello alguna de las otras particiones disponibles. ¿Sigue cumpliendo el volumen lógico las propiedades de rapidez y tolerancia a fallos con las que fue creado? Justifica tu respuesta.

3. (0,96 puntos) Con respecto a la gestión de los recursos del sistema:

- a) (0,36 puntos) Localiza el paquete que contiene el programa `/bin/time`, averigua el número de ficheros que tiene y cuándo fue instalado.
- b) (0,18 puntos) Determina la cantidad de memoria RAM del sistema. ¿Qué cantidad está siendo usada en este momento para cualquier finalidad?
- c) (0,18 puntos) Como usuario `alumno`, lanza el programa `evince` en segundo plano (`evince &`) estableciendo su número NICE de forma que la prioridad dinámica del proceso resultante sea 30 (no cierres el programa cuando se abra). Demuestra que el proceso se ha creado con el número NICE que estableciste.
- d) (0,24 puntos) Ejecuta dos veces más el programa `evince` en segundo plano y, a continuación, como usuario `root`, programa el cierre forzoso de todos los procesos creados a partir el programa `evince` (los procesos deben acabar incondicionalmente) para dentro de 5 minutos contados a partir del momento en el que hagas este apartado.

4. (0,36 puntos) En cuanto a la gestión de la E/S:

- a) (0,18 puntos) Localiza el modelo de tarjeta de red Ethernet que está instalada en el sistema.
- b) (0,18 puntos) ¿Qué planificador se está usando para el primer disco duro de la máquina virtual?. Si no es `none`, cámbialo para que se use este a partir de ahora en la sesión actual.

Guiones Shell (5 puntos)

Esta parte se compone de 2 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, `estan_ficheros.sh`). En cada ejercicio debes hacer exactamente lo que se pide (no se valorará la funcionalidad extra que no hubiese sido pedida explícitamente).

Junto con el PDF del examen se ha descargado el archivo `recursos-guiones.tgz` que contiene ficheros que pueden ayudarte a resolver los diferentes ejercicios.

1. (2 puntos) Implementa un guion shell llamado `estan_ficheros.sh` que reciba como parámetros un fichero de texto, en donde en cada línea aparecerá el nombre de un fichero regular y el nombre de un directorio. El guion elaborará un listado en el que se informará si cada uno de los ficheros regulares que aparecen en el fichero de texto pasado como primer argumento están en el directorio pasado como segundo, considerando, también, los subdirectorios que pudiese tener. Finalmente, el guion informará acerca del porcentaje de ficheros regulares que ha encontrado en el directorio.

El guion debe realizar las siguientes comprobaciones sobre los parámetros dados:

- El guion recibe exactamente 2 parámetros. En caso contrario, se debe informar de ello y devolver 1 como código de salida (ver ejemplos de ejecución).
- El primer parámetro debe ser un fichero regular que se pueda leer y tenga un tamaño mayor que cero. En caso contrario, se debe informar de ello y devolver 2 como código de salida (ver ejemplos de ejecución).
- El segundo parámetro debe ser un directorio que exista y se pueda acceder a sus entradas y a las de los subdirectorios que pudiese tener (supondremos que los subdirectorios siempre tendrán los permisos para poder ser accedidos). En caso contrario, se debe informar de ello y devolver 3 como código de salida (ver ejemplos de ejecución).

A continuación se muestran algunos ejemplos de ejecución de este guion. Ten en cuenta que la salida debe quedar formateada tal y como se muestra (presta atención al uso de los caracteres ' y " en los mensajes de salida):

```
[alumno@localhost ~]$ ls -R recursos-guiones/
recursos-guiones/:
dir listado.txt

recursos-guiones/dir:
a1.c a2.b a3.c a4.txt a5.1 a6 a7.txt a8.c a9 dir1 dir2 dir3

recursos-guiones/dir/dir1:
cinco cuatro.123 dos.pdf ocho.sinext seis.pdf siete.txt tres.nada uno.txt

recursos-guiones/dir/dir2:
img1.jpg img2.gif img3.jpg img4.gif img5.jpeg img6 img7.bmp img8.gif img9.png

recursos-guiones/dir/dir3:
sin1 sin2.sinextension

[alumno@localhost ~]$ cat recursos-guiones/listado.txt
img2.gif
imagen1.png
siete.txt
ocho.pdf
a8.c
cinco
seis

[alumno@localhost ~]$ bash estan_ficheros
USO: estan_ficheros.sh fichero directorio
[alumno@localhost ~]$ echo $?
1

[alumno@localhost ~]$ bash estan_ficheros.sh ficheronoe existe /root
'ficheronoe existe' no es un fichero regular, no se puede leer o está vacío
[alumno@localhost ~]$ echo $?
2

[alumno@localhost ~]$ bash estan_ficheros.sh recursos-guiones/listado.txt /root
'/root' no es un directorio o no se puede leer y/o atravesar
[meacacio@sandecker bash]$ echo $?
3

[alumno@localhost ~]$ bash estan_ficheros.sh recursos-guiones/listado.txt recursos-guiones/dir
Fichero "img2.gif": encontrado.
```

```
Fichero "imagen1.png": no encontrado.
Fichero "siete.txt": encontrado.
Fichero "ocho.pdf": no encontrado.
Fichero "a8.c": encontrado.
Fichero "cinco": encontrado.
Fichero "seis": no encontrado.
Se han encontrado el 57% de los ficheros regulares de los listados en "recursos-guiones/listado.txt"
en "recursos-guiones/dir/".
[meacacio@sandecker bash]$ echo $?
0
```

Fichero de solución: **estan_ficheros.sh**

2. (3 puntos) Implementa un guion shell llamado `ls_ext.sh` que recibe como argumentos una lista de directorios (si no recibe ninguno, se asumirá el directorio actual) que existen, pueden ser leídos y no tienen espacios en blanco en sus nombres (no es necesario que realices estas comprobaciones, se asume que se cumplirán siempre). El guion elaborará un listado de los ficheros regulares con extensión que aparecen en cada directorio, sin considerar los que pudiese haber en los posibles subdirectorios que tuviesen, agrupándolos por extensión y precediéndolos del nombre del directorio. Se supone que un fichero regular tiene extensión cuando su nombre acaba en un punto seguido por hasta 3 caracteres alfabéticos («fichero.txt» o «fichero.c» son ejemplos de nombres de fichero con extensión; «fichero.fichero», «fichero» o «fichero.1», no se consideran nombres de fichero con extensión). Los nombres de los ficheros regulares sin extensión no aparecerán en el listado. Por simplicidad, supón que, como mucho, hay un único carácter punto en el nombre de cada fichero.

Si algún directorio no tuviese ficheros regulares con extensión, este no aparecerá en el listado (se saltará) y el guion, cuando termine, devolverá como código de salida el valor 1. En otro caso (cada uno de los directorios pasados como argumento tiene, al menos, un fichero regular con extensión), el código de salida devuelto por el guion será 0.

Para resolver el guion, sigue los dos pasos que se indican:

- a) (1,5 puntos) Crea, en primer lugar, un guion que elabore un listado en el que se muestren las extensiones únicas encontradas en cada uno de los directorios pasados como argumento (sin entrar en subdirectorios) indicando el nombre del directorio. Si no se pasan argumentos, se asumirá el directorio actual. En el listado solamente aparecerán los directorios que contienen ficheros regulares con extensión. El código de salida será establecido de acuerdo a lo especificado en el enunciado general. A continuación se muestran varios ejemplos de ejecución de este guion:

```
[alumno@localhost ~]$ bash ls_ext.a.sh recursos-guiones/dir/dir1 recursos-guiones/dir/dir2
recursos-guiones/dir/dir1: pdf txt
recursos-guiones/dir/dir2: bmp gif jpg
[alumno@localhost ~]$ echo $?
0
```

```
[alumno@localhost ~]$ bash ls_ext.a.sh recursos-guiones/dir recursos-guiones/dir/dir3 \
> recursos-guiones/dir/dir2
recursos-guiones/dir: b c txt
recursos-guiones/dir/dir2: bmp gif jpg
[alumno@localhost ~]$ echo $?
1
```

```
[alumno@localhost ~]$ cd recursos-guiones/dir
[alumno@localhost dir]$ bash ../../ls_ext.a.sh
.: b c txt
[alumno@localhost ~]$ echo $?
0
```

Fichero de solución: **ls_ex.a.sh**

- b) (1,5 puntos) Completa el guion para que genere el listado pedido en el enunciado. A continuación se muestran un ejemplo de cómo debería quedar el listado:

```
[alumno@localhost ~]$ bash ls_ext.b.sh recursos-guiones/dir recursos-guiones/dir/dir3 \
> recursos-guiones/dir/dir2
recursos-guiones/dir:
{ a2 }.b
{ a1 a3 a8 }.c
{ a4 a7 }.txt
recursos-guiones/dir/dir2:
{ img7 }.bmp
{ img2 img4 img8 }.gif
{ img1 img3 }.jpg
[alumno@localhost ~]$ echo $?
1
```

Fichero de solución: **ls_ex.b.sh**

Soluciones

Administración

*** Ejercicio 1 ***

1.a.-

```
[alumno@localhost ~]$ su -l
Contraseña:
[root@localhost ~]# groupadd estudiante
[root@localhost ~]# grep -w estudiante /etc/group
estudiante:x:1001:
```

Se ha creado con el GID 1001

1.b.-

```
[root@localhost ~]# useradd -g estudiante -c "Alumna de ISO" alumna
[root@localhost ~]# id alumna
uid=1001(alumna) gid=1001(estudiante) grupos=1001(estudiante)

# Vemos que se le ha asignado el UID 1001 y que pertenece únicamente al grupo estudiante.
```

1.c.-

```
[root@localhost alumno]# echo '"c4mb14m3"' | passwd --stdin alumna
Cambiando la contraseña del usuario alumna.
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[root@localhost alumno]# passwd -e alumna
Expirando contraseña para el usuario alumna.
passwd: Éxito
[root@localhost alumno]# exit
exit
[alumno@localhost ~]$ su -l alumna
Contraseña:
Debe cambiar la contraseña inmediatamente (obligado por el administrador)
Current password:
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
[alumna@localhost ~]$ exit
cerrar sesión
[alumno@localhost ~]$ su -l
Contraseña:
[root@localhost ~]#
```

1.d.-

```
[root@localhost ~]# usermod -aG estudiante alumno
[root@localhost ~]# groupadd alumna
[root@localhost ~]# usermod -aG alumna alumna
[root@localhost ~]# id alumno alumna
uid=1000(alumno) gid=1000(alumno) grupos=1000(alumno),10(wheel),1001(estudiante)
uid=1001(alumna) gid=1001(estudiante) grupos=1001(estudiante),1002(alumna)
```

1.e.-

```
[root@localhost /]# cd /home
[root@localhost home]# ls
alumna alumno
[root@localhost home]# mkdir practicas
[root@localhost home]# chgrp estudiante practicas
[root@localhost home]# chmod 770 practicas
[root@localhost home]# ls -ld practicas/
drwxrwx---. 2 root estudiante 4096 may 27 13:26 practicas/
```

1.f.-

```
[root@localhost home]# chmod +t practicas/
[root@localhost home]# ls -ld practicas/
drwxrwx--T. 2 root estudiante 4096 may 27 13:26 practicas/
```

1.g.-

```
[root@localhost home]# chmod g+s practicas/
[root@localhost home]# ls -ld practicas/
drwxrws--T. 2 root estudiante 4096 may 27 13:26 practicas/
[root@localhost home]# su -l alumno
[alumno@localhost ~]$ id
```

```

uid=1000(alumno) gid=1000(alumno) grupos=1000(alumno),10(wheel),1001(estudiante) contexto=unconfined_u:
unconfined_r:unconfined_t:s0-s0:c0.c1023
[alumno@localhost ~]$ touch ../practicas/boletines_ISO.pdf
[alumno@localhost ~]$ ls -l ../practicas/boletines_ISO.pdf
-rw-rw-r--. 1 alumno estudiante 0 may 27 13:30 ../practicas/boletines_ISO.pdf

# Ejecutando «id» sin argumentos, el grupo activo aparece en gid= (como vemos, «es alumno»)

*** Ejercicio 2 ***

2.a.-

[root@localhost ~]# fdisk -l /dev/sd[b-d]
Disco /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectores
Modelo de disco: QEMU HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: gpt
Identificador del disco: 0CC4EED5-D17D-E441-9C29-DA37D29E6A5E

Disposit.  Comienzo    Final Sectores Tamaño Tipo
/dev/sdb1      2048    821247    819200    400M Sistema de ficheros de Linux
/dev/sdb2      821248  2097118  1275871    623M Sistema de ficheros de Linux

Disco /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectores
Modelo de disco: QEMU HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: gpt
Identificador del disco: DA107135-C1E4-DE4C-8183-A0B60122F670

Disposit.  Comienzo    Final Sectores Tamaño Tipo
/dev/sdc1      2048    821247    819200    400M Sistema de ficheros de Linux
/dev/sdc2      821248  2097118  1275871    623M Sistema de ficheros de Linux

Disco /dev/sdd: 1 GiB, 1073741824 bytes, 2097152 sectores
Modelo de disco: QEMU HARDDISK
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: gpt
Identificador del disco: FCC4D362-A333-094D-9BFC-FC1A1B747F34

Disposit.  Comienzo    Final Sectores Tamaño Tipo
/dev/sdd1      2048    821247    819200    400M Sistema de ficheros de Linux
/dev/sdd2      821248  2097118  1275871    623M Sistema de ficheros de Linux

2.b.-

[root@localhost ~]# vgcreate storage /dev/sd[bcd]1
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdc1" successfully created.
Physical volume "/dev/sdd1" successfully created.
Volume group "storage" successfully created
[root@localhost ~]# vgdisplay storage
--- Volume group ---
VG Name                storage
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                 0
Cur PV                 3
Act PV                 3
VG Size                 1,16 GiB
PE Size                 4,00 MiB
Total PE                297
Alloc PE / Size         0 / 0
Free PE / Size           297 / 1,16 GiB
VG UUID                 J9IYiJ-uITR-yDbZ-POU8-UDuq-Ogx2-8JOFzy

# El número total de extensiones físicas del grupo de volúmenes «storage» es 297,

```

```
# tal como se aprecia en la línea "Total PE" de la orden vgdisplay.
```

2.C.-

```
# Se ha de crear un volumen lógico de tipo raid5 (ver el sexto boletín de prácticas
# para una explicación detalla acerca de las propiedades de este tipo de
# volumen lógico. Para crearlo del máximo tamaño disponible, usaremos la orden
# «lvcreate» con la opción «-l 100%FREE».
```

```
[root@localhost ~]# lvcreate -i 2 --type raid5 -l 100%FREE -n lv-disk storage
Using default stripesize 64,00 KiB.
Rounding size (297 extents) down to stripe boundary size (296 extents)
Logical volume "lv-disk" created.
```

```
# Como se muestra a continuación, el nuevo volumen lógico tiene un tamaño de
# 784 MiB repartidos en 196 extensiones lógicas.
```

```
[root@localhost ~]# lvdisplay /dev/storage/lv-disk
--- Logical volume ---
LV Path                /dev/storage/lv-disk
LV Name                lv-disk
VG Name                storage
LV UUID                sB30UF-RtK9-lZcU-lQxb-6UMS-JmJG-QuKUPi
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2024-05-27 17:29:40 +0200
LV Status              available
# open                 0
LV Size                784,00 MiB
Current LE             196
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to    768
Block device           253:8
```

2.d.-

```
[root@localhost ~]# mkfs.ext4 /dev/storage/lv-disk
mke2fs 1.45.5 (07-Jan-2020)
Se está creando un sistema de ficheros con 200704 bloques de 4k y 50176 nodos-i
UUID del sistema de ficheros: 351d6ca4-df13-4e50-9ce0-e44afbc0103a
Respaldos del superbloque guardados en los bloques:
    32768, 98304, 163840

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (4096 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
```

```
# Vemos en la salida de la orden mkfs.ext4 que el nuevo sistema de ficheros
# tiene tamaño de bloque lógico de 4 KiB. Vemos, también, que hay, en total, 4
# copias del superbloque (el original y los 3 respaldos que nos indican)
```

2.e.-

```
[root@localhost ~]# vim /etc/fstab
[root@localhost ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sun Jul  5 09:58:53 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/fedora-root / ext4 defaults 1 1
UUID=9e456c9e-e0c4-47f0-a694-0a6f8abebd6f /boot ext4 defaults 1 2
/dev/mapper/fedora-swap none swap defaults 0 0
hostExamenISO /home/alumno/ExamenISO 9p trans=virtio,version=9p2000.L,noauto,x-systemd.automount,x-
systemd.device-timeout=30,user 0 0
UUID=351d6ca4-df13-4e50-9ce0-e44afbc0103a /home/practicas ext4 defaults 0 0

[root@localhost ~]# mount /home/practicas
[root@localhost ~]# ls -ld /home/practicas/
drwxr-xr-x. 3 root root 4096 may 27 17:35 /home/practicas/
[root@localhost ~]# chgrp estudiante /home/practicas/
[root@localhost ~]# chmod 770 /home/practicas/
[root@localhost ~]# chmod +t /home/practicas/
[root@localhost ~]# chmod g+s /home/practicas/
```

```
[root@localhost ~]# ls -ld /home/practicas/
drwxrws--T. 3 root estudiante 4096 may 27 17:35 /home/practicas/
[root@localhost ~]# mount | grep practicas
/dev/mapper/storage-lv--disk on /home/practicas type ext4 (rw,relatime,seclabel,stripe=32)
```

2.f.-

```
# Al tratarse de un enlace a un directorio, tenemos que crear un enlace simbólico.
# Vemos que el fichero queda con 15 bytes, que es justamente el número de caracteres
# que configuran la ruta que almacena (/home/practicas)
```

```
[root@localhost ~]# su -l alumna
[alumna@localhost ~]$ pwd
/home/alumna
[alumna@localhost ~]$ ln -s /home/practicas mi-enlace
[alumna@localhost ~]$ ls -l
total 0
lrwxrwxrwx. 1 alumna estudiante 15 may 27 17:46 mi-enlace -> /home/practicas
```

2.g.-

```
# Sin considerar lo que pudiese haber en el subdirectorio lost-found, para el que no
# tenemos permiso de acceso, «/home/practicas» ocupa 2,2M de disco.
# El sistema de ficheros, tras la copia, presenta un porcentaje de ocupación del 1%
```

```
[alumna@localhost ~]$ cd /home/practicas
[alumna@localhost practicas]$ cp -a /var/log/* . 2> /dev/null
[alumna@localhost mi-enlace]$ du -hs /home/practicas
du: no se puede leer el directorio '/home/practicas/lost+found': Permiso denegado
2,2M    /home/practicas
[alumna@localhost mi-enlace]$ df -h /dev/storage/lv-disk
S.ficheros          Tamaño Usados  Disp Uso% Montado en
/dev/mapper/storage-lv--disk  756M   3,7M  698M   1% /home/practicas
```

2.h.-

```
[root@localhost ~]# find /home/practicas -type f -size +1023c | cpio -o -F /root/copia-seguridad.cpio
4201 bloques
[root@localhost ~]# cpio -t -F /root/copia-seguridad.cpio
/home/practicas/lastlog
/home/practicas/Xorg.0.log.old
/home/practicas/hawkey.log-20200916
/home/practicas/aquota.user
/home/practicas/dnf.rpm.log
/home/practicas/Xorg.0.log
/home/practicas/Xorg.1.log.old
/home/practicas/dnf.librepo.log-20200916
/home/practicas/Xorg.1.log
/home/practicas/dnf.log
/home/practicas/README
/home/practicas/dnf.log.1
/home/practicas/wtmp
4201 bloques
```

```
# Vemos que se han respaldado un total de 13 ficheros regulares.
```

2.i.-

```
# Hay que extender, primero, el grupo de volúmenes para incluir una de las particiones que no estábamos
# usando (/dev/sdb2 en este caso) y que emplearemos para mover a ella las extensiones lógicas que
# ahora tenemos en /dev/sdc1. Después, procederemos a mover las extensiones, a reducir el grupo
# de volúmenes eliminado /dev/sdc1, y por último, a quitar la «marca» de volumen físico de
# /dev/sdc1:
```

```
[root@localhost alumno]# vgextend storage /dev/sdb2
Physical volume "/dev/sdb2" successfully created.
Volume group "storage" successfully extended
[root@localhost alumno]# pvmove /dev/sdc1 /dev/sdb2
/dev/sdc1: Moved: 2,42%
/dev/sdc1: Moved: 99,19%
/dev/sdc1: Moved: 100,00%
[root@localhost alumno]# vgreduce storage /dev/sdc1
Removed "/dev/sdc1" from volume group "storage"
[root@localhost alumno]# pvremove /dev/sdc1
Labels on physical volume "/dev/sdc1" successfully wiped.
\end{lstlisting}
```

```
# Ahora no se puede garantizar que sigan cumpliéndose las propiedades buscadas en el apartado 2.c.
# Por ejemplo, al tener todo el contenido del volumen lógico en dos discos (/dev/sdb y /dev/sdd) hemos
# perdido la capacidad para tolerar fallos (la caída de /dev/sdb dejaría el volumen
# lógico inutilizable).
```


*** Ejercicio 3 ***

3.a.-

```
[root@localhost ~]# rpm -qf /bin/time
time-1.9-11.fc32.x86_64
[root@localhost ~]# rpm -ql time | wc -l
12
[root@localhost ~]# rpm -qi time | grep -i install
Install Date: mié 14 jul 2021 12:09:45

# El fichero pertenece al paquete time-1.9-11.fc32.x86_64, que tiene 12 ficheros en total.
# El paquete fue instalado el 14 de julio de 2021
```

3.b.-

```
[root@localhost ~]# free
              total        used        free      shared  buff/cache   available
Mem:          2024128      909272      222840        37996       892016       904604
Swap:          1572860           0       1572860
```

```
# Hay instalado un total de 2 GiB de memoria RAM. De esta cantidad de memoria,
# está en uso aproximadamente 1,7 GiB en total (teniendo en cuenta la memoria
# que se está usando para buffers y caché).
```

3.c.-

```
[root@localhost ~]# nice -n 10 evince &
[1] 2844
[root@localhost ~]# ps axo command,nice | grep evince
evince                  10
```

3.d.-

```
[root@localhost ~]# evince &
[root@localhost ~]# evince &
[root@localhost ~]# at NOW+5min
warning: commands will be executed using /bin/sh
at> killall -9 evince
at> <EOT>
job 1 at Mon May 27 18:39:00 2024
```

*** Ejercicio 4 ***

4.a.-

```
# Se trata de una Realtek Semiconductor Co., Ltd. RTL-8100/8101L/8139 PCI Fast Ethernet Adapter

[root@localhost alumno]# lspci | grep -i ethernet
00:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8100/8101L/8139 PCI Fast Ethernet
Adapter (rev 20)
```

4.b.-

```
# Consultamos primero el planificador que está en uso en este instante:
```

```
[root@localhost ~]# cat /sys/block/sda/queue/scheduler
mq-deadline kyber [bfq] none
```

```
# Dado que es «bfq», lo cambiamos por «none» como nos piden:
```

```
[root@localhost ~]# echo none > /sys/block/sda/queue/scheduler
[root@localhost ~]# cat /sys/block/sda/queue/scheduler
[none] mq-deadline kyber bfq
```

Guiones Shell

Se muestra a continuación las soluciones de los distintos ejercicios de la parte de bash:

1. #!/bin/bash

```
test $# -ne 2 && echo "USO: $0 fichero directorio" >&2 && exit 1
test ! -f $1 -o ! -r $1 -o ! -s $1 && echo "'$1' no es un fichero regular, no se puede leer o está
vacío" >&2 && exit 2
test ! -d $2 -o ! -r $2 -o ! -x $2 && echo "'$2' no es un directorio o no se puede leer y/o
atravesar" >&2 && exit 3

estan=0
while read nombre
do
    echo -n "Fichero \"$nombre\": "
```

```

if test "$(find $2 -type f -name "$nombre")" != ""
then
    echo "encontrado."
    let estan+=1
else
    echo "no encontrado."
fi
done < $1

echo "Se han encontrado el $((($estan*100/$(cat $1 | wc -l)))% de los ficheros regulares listados
en \"$1\" a partir del directorio \"$2\"."

```

Script 1: estan_ficheros.sh

2. a) #!/bin/bash

```

if test $# -eq 0
then
    directorios="."
else
    directorios="$@"
fi

salida=0

for dir in $directorios
do
    ficheros=$(find $dir -maxdepth 1 -type f -printf "%f\n")
    extensiones=$(echo "$ficheros" | cut -f2 -d'.' -s | grep -ix "[a-z]\{1,3\}" | sort | uniq)
    if test "$extensiones" != ""
    then
        echo "$dir: $(echo $extensiones | tr '\n' ' ')"
    else
        salida=1
    fi
done

exit $salida

```

Script 2: ls_ext.a.sh

b) #!/bin/bash

```

if test $# -eq 0
then
    directorios="."
else
    directorios="$@"
fi

salida=0

for dir in $directorios
do
    ficheros=$(find $dir -maxdepth 1 -type f -printf "%f\n")
    extensiones=$(echo "$ficheros" | cut -f2 -d'.' -s | grep -ix "[a-z]\{1,3\}" | sort | uniq)
    if test "$extensiones" != ""
    then
        echo "$dir:"
        for ext in $extensiones
        do
            nombres=$(echo "$ficheros" | grep "\.$ext$" | cut -f1 -d'.' | tr "\n" " ")
            echo -e "\t{ $nombres}.$ext"
        done
    else
        salida=1
    fi
done

exit $salida

```

Script 3: ls_ext.b.sh