



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)

- (0,90 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:
 - (0,18 puntos) **Sin crear grupos que no se piden**, crea el usuario `arturo` con nombre completo `Arturo Pérez Reverte` y grupo primario `alumno`. Haz lo propio con el usuario `jeronimo` con nombre completo `Jerónimo Salmerón Tristante` y grupo primario también `alumno`.
 - (0,12 puntos) Crea los grupos `escritor` y `bestseller`.
 - (0,36 puntos) Para el usuario `jeronimo`, fija el grupo `escritor` como grupo secundario. Para el usuario `arturo`, cambia su grupo primario a `bestseller` y establece el grupo `escritor` como su grupo secundario. Muestra, a través de la orden adecuada, los grupos a los que pertenecen tanto el usuario `arturo` como el usuario `jeronimo`.
 - (0,24 puntos) Crea en `/home` el directorio `novelas`. El usuario propietario del mismo será `root` y deberás hacer lo necesario para que tanto el usuario `arturo` como `jeronimo` 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/novelas`.
- (2,78 puntos) Realiza las siguientes tareas sobre gestión de discos y sistemas de ficheros:
 - (0,36 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 500 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 -l /dev/sd[b-d]`.
 - (0,36 puntos) A partir de la primera partición de cada uno de los discos anteriores, crea el grupo de volúmenes `repositorio`. ¿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.
 - (0,48 puntos) Usando el grupo de volúmenes anterior, queremos crear el volumen lógico `novelas` 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? Además de responder a la preguntas, muestra la salida de las órdenes que hayas usado para responder a las cuestiones.

- d) (0,24 puntos) Crea en el volumen lógico anterior un sistema de ficheros `ext4`. ¿Cuántos ficheros, como máximo, podría albergar el sistema de ficheros que acabas de crear (adjunta la salida de la orden que te lo indica)?
- e) (0,48 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/novelas/`, con los mismos permisos que hubo que definir para resolver el apartado 1.d.
- f) (0,36 puntos) Como `arturo`, crea un enlace llamado `/home/arturo/mis-novelas` al directorio `/home/novelas/`. ¿Qué tipo de enlace has tenido que crear? ¿Qué tamaño tiene el fichero `/home/arturo/mis-novelas`? Justifica las respuestas a ambas cuestiones.
- g) (0,5 puntos) Haz lo necesario para activar las cuotas para dicho sistema de ficheros, de forma que los usuarios `jeronimo` y `arturo` tengan como límite *soft* 300 MiB y *hard* 400 MiB, con un periodo de gracia de 5 días, y puedan crear un máximo de 10 000 ficheros sin que se pueda sobrepasar este límite.

3. (0,78 puntos) Configura un sistema de copias de seguridad mediante la orden `tar` de la siguiente manera:

- a) (0,42 puntos) Todos los domingos a las 4:00 a.m. se tiene que hacer una copia de seguridad total del directorio `/home` en un fichero llamado `/home-domingo.tar.xz`. Observa que la copia queda comprimida mediante `xz`. Busca en la página de manual de `tar` la opción adecuada para que se use dicho programa de compresión.
- b) (0,36 puntos) Los miércoles y los viernes, también a las 4:00 a.m., se tiene que hacer una copia de seguridad, pero solo de aquellos ficheros que se hayan *modificado* desde que finalizó la copia de seguridad del domingo. Estas copias se almacenarán en ficheros comprimidos llamados `/home-<nombredia>.tar.xz`, siendo `<nombredia>` el día de la semana correspondiente.

4. (0,54 puntos) Realiza los ejercicios que tienes a continuación relacionados con la gestión de la E/S:

- a) (0,24 puntos) Averigua qué módulo de Linux es el encargado de gestionar la tarjeta de red de la máquina virtual, cuyo modelo es «RTL-8100/8101L/8139 PCI Fast Ethernet Adapter».
- b) (0,3 puntos) Ejecuta en segundo plano la orden `sleep 1000` con la prioridad más alta posible de peticiones de E/S. Demuestra, con la orden adecuada, que efectivamente, el proceso se ejecuta de la forma solicitada. Nota: recuerda que la orden `sleep 1000` simplemente se queda bloqueada durante 1000 segundos y luego termina.

Guiones Shell (5 puntos)

1. (5 puntos) Esta parte se compone de 3 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, `copia_ddf.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.

- a) (2,0 puntos) Implementa un guion shell llamado `copia_ddf.sh` que reciba tres parámetros (un directorio origen, un directorio destino y un nombre de fichero) y copie el fichero dado como tercer parámetro del directorio origen (primer parámetro) al directorio destino (segundo parámetro). Observa que el tercer parámetro no es una ruta, sino un simple nombre de fichero que debe existir en el origen.

El guion debe hacer diferentes comprobaciones y devolver los siguientes códigos de error:

- 1: el número de parámetros es incorrecto.
- 2: el primer parámetro no es directorio o no tiene permisos de lectura y ejecución.
- 3: el segundo parámetro no es directorio o no tiene permisos de escritura y ejecución.
- 4: el tercer parámetro no es un fichero regular en el directorio origen o no tiene permiso de lectura.
- 5: el tercer parámetro no debe existir como fichero (da igual que sea fichero regular, directorio, enlace simbólico, etc.) en el directorio destino.

Solo si todas las comprobaciones son exitosas, el fichero deberá copiarse.

A continuación se muestran algunas ejecuciones con error de este guion shell:

```
[alumno@localhost ~]$ bash copia_ddf.sh
USO: copia_ddf.sh <directorio_origen> <directorio_destino> <nombre_fichero>
[alumno@localhost ~]$ echo $?
1
[alumno@localhost ~]$ bash copia_ddf.sh /root /etc/ passwd
Error: el primer parametro debe ser un directorio con permiso de lectura y ejecución
[alumno@localhost ~]$ echo $?
2
[alumno@localhost ~]$ bash copia_ddf.sh dl /etc/ passwd
Error: el segundo parametro debe ser un directorio con permisos de escritura y ejecución
[alumno@localhost ~]$ echo $?
3
```

```
[alumno@localhost ~]$ bash copia_ddf.sh d1 d2 passwd
Error: el tercer parametro debe ser un fichero regular con permiso de lectura en el directorio origen
[alumno@localhost ~]$ echo $?
4
[alumno@localhost ~]$ bash copia_ddf.sh d1 d2 f13
Error: el tercer parametro no debe existir en el directorio destino
[alumno@localhost ~]$ echo $?
5
[alumno@localhost ~]$ bash copia_ddf.sh d1 d2 f11
Copiando fichero: d1/f11 --> d2/f11
[alumno@localhost ~]$ echo $?
0
```

Nota: los directorios de prueba d1 y d2 se han descargado junto con el examen.

Fichero de solución: **copia_ddf.sh**

- b) (1,5 puntos) Implementa un guion shell llamado `pon_extension.sh` que reciba como parámetro un fichero de texto, donde cada línea tendrá dos campos con la siguiente sintaxis:

```
<directorio> <espacios en blanco> <cadena alfanumérica>
```

Por cada línea, el guion shell cambiará el nombre de todo fichero regular que haya en el directorio que aparezca en el primer campo de la línea (sin entrar en sus subdirectorios). El nuevo nombre de cada fichero será el que tenía seguido de un punto «.» y la cadena alfanumérica que aparece como segundo campo de la línea.

Tras procesar el fichero, el guion mostrará en pantalla el número total de ficheros cuyo nombre haya sido cambiado.

El guion no debe comprobar los parámetros dados (supondremos que son siempre correctos) ni el contenido del fichero (que también supondremos siempre correcto, con directorios válidos y cadenas alfanuméricas correctas).

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

```
[alumno@localhost ~]$ cat fichero.cfg
d1    aaa
d2    bbb
[alumno@localhost ~]$ find d*
d1
d1/f13
d1/f12
d1/f11
d2
d2/f13
d2/f22
d2/f21
d2/f23
d3
d3/f33
d3/f32
d3/f31
[alumno@localhost ~]$ bash pon_extension.sh fichero.cfg
Número de ficheros cuyo nombre ha sido cambiado: 7
[alumno@localhost ~]$ find d*
d1
d1/f13.aaa
d1/f12.aaa
d1/f11.aaa
d2
d2/f13.bbb
d2/f22.bbb
d2/f21.bbb
d2/f23.bbb
d3
d3/f33
d3/f32
d3/f31
```

Fichero de solución: **pon_extension.sh**

- c) (1,5 puntos) Implementa un guion shell llamado `invierte_y_cuenta.sh` que reciba como único parámetro un fichero regular (suponemos que siempre va a existir y se va a poder leer), y que contendrá una serie de líneas de texto, algunas de las cuales pueden estar en blanco (es decir, sin palabras, aunque pueden contener espacios en blanco) y otras tendrán una o más palabras separadas por uno o más espacios en blanco. El guion debe imprimir únicamente las líneas que no estén en blanco, usando exactamente un espacio para separar cada una de sus palabras, y en orden inverso al que se encuentran en el fichero original (la última línea aparecerá en primer lugar, la penúltima en segundo lugar, etc.). Además, para cada línea se mostrará el número de palabras que contiene. A continuación se muestra un ejemplo de ejecución:

```
[alumno@localhost ~]$ cat ficherolineas.txt
tres      dos  uno
    siete seis cinco  cuatro

    diez  nueve  ocho
doce once

quince      catorce trece
[alumno@localhost ~]$ bash invierte_y_cuenta.sh ficherolineas.txt
quince catorce trece (-> 3)
doce once (-> 2)
diez nueve ocho (-> 3)
siete seis cinco cuatro (-> 4)
tres dos uno (-> 3)
```

Fichero de solución: **invierte_y_cuenta.sh**

Soluciones

Administración

*** Ejercicio 1 ***

1.a.-

```
[root@localhost alumno]# useradd -c "Arturo Pérez Reverte" -g alumno arturo
[root@localhost alumno]# useradd -c "Jerónimo Salmerón Tristante" -g alumno jeronimo
```

1.b.-

```
[root@localhost alumno]# groupadd escritor
[root@localhost alumno]# groupadd bestseller
```

1.c.-

```
[root@localhost alumno]# usermod jeronimo -G escritor
[root@localhost alumno]# usermod arturo -g bestseller -G escritor
[root@localhost alumno]# groups jeronimo
jeronimo : alumno escritor
[root@localhost alumno]# groups arturo
arturo : bestseller escritor
```

1.d.-

```
[root@localhost alumno]# mkdir /home/novelas
[root@localhost alumno]# chown :escritor /home/novelas/
[root@localhost alumno]# ls -ld /home/novelas/
drwxr-xr-x. 2 root escritor 4096 jun  6 14:18 /home/novelas/
[root@localhost alumno]# chmod 770 /home/novelas/
[root@localhost alumno]# ls -ld /home/novelas/
drwxrwx---. 2 root escritor 4096 jun  6 14:18 /home/novelas/
```

*** Ejercicio 2 ***

2.a.-

```
[root@localhost ~]# fdisk /dev/sdb
```

Bienvenido a fdisk (util-linux 2.32.1).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0x9aac38d8.

Orden (m para obtener ayuda): g
Se ha creado una nueva etiqueta de disco GPT (GUID: CB29EFD1-1D5C-B541-B665-61D9A1D38D2B).

Orden (m para obtener ayuda): n
Número de partición (1-128, valor predeterminado 1):
Primer sector (2048-2097118, valor predeterminado 2048):
Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-2097118, valor predeterminado 2097118): +500M

Crea una nueva partición 1 de tipo 'Linux filesystem' y de tamaño 500 MiB.

Orden (m para obtener ayuda): n
Número de partición (2-128, valor predeterminado 2):
Primer sector (1026048-2097118, valor predeterminado 1026048):
Último sector, +sectores o +tamaño{K,M,G,T,P} (1026048-2097118, valor predeterminado 2097118):

Crea una nueva partición 2 de tipo 'Linux filesystem' y de tamaño 523 MiB.

Orden (m para obtener ayuda): w
Se ha modificado la tabla de particiones.
Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.
[root@localhost alumno]# fdisk -l /dev/sd[b-d]
Disco /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 sectores
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: CB29EFD1-1D5C-B541-B665-61D9A1D38D2B

Disposit.	Comienzo	Final	Sectores	Tamaño	Tipo
/dev/sdb1	2048	1026047	1024000	500M	Sistema de ficheros de Linux
/dev/sdb2	1026048	2097118	1071071	523M	Sistema de ficheros de Linux

Disco /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectores
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: B22B0668-1820-174C-94AB-48EA6999996E

Disposit.	Comienzo	Final	Sectores	Tamaño	Tipo
/dev/sdc1	2048	1026047	1024000	500M	Sistema de ficheros de Linux
/dev/sdc2	1026048	2097118	1071071	523M	Sistema de ficheros de Linux

Disco /dev/sdd: 1 GiB, 1073741824 bytes, 2097152 sectores
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: 34D59047-48BB-2A4B-B6FE-9F54AA6B972E

Disposit.	Comienzo	Final	Sectores	Tamaño	Tipo
/dev/sdd1	2048	1026047	1024000	500M	Sistema de ficheros de Linux
/dev/sdd2	1026048	2097118	1071071	523M	Sistema de ficheros de Linux

2.b.-

```
[root@localhost alumno]# vgcreate repositorio /dev/sd[b-d]1
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdc1" successfully created.
Physical volume "/dev/sdd1" successfully created.
Volume group "repositorio" successfully created
[root@localhost alumno]# vgdisplay repositorio
--- Volume group ---
VG Name                repositorio
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                 1,45 GiB
PE Size                 4,00 MiB
Total PE                372
Alloc PE / Size         0 / 0
Free PE / Size          372 / 1,45 GiB
VG UUID                2FrAtK-FWYg-t49X-YenL-3QkH-lPkp-GoelRw
```

2.c.-

Para disponer de un volumen lógico capaz de ofrecer mayor rendimiento que el
de un disco y capacidad de tolerar la caída de uno de los discos, debemos
decantarnos por la creación de un volumen lógico raid5. Para crearlo del
máximo tamaño disponible, usaremos la orden lvcreate con la opción -l 100%.

```
[root@localhost alumno]# lvcreate --type raid5 -i 2 -l 100%FREE -n novelas repositorio
Using default stripesize 64,00 KiB.
Logical volume "novelas" created.
```

Como muestra la siguiente orden, el nuevo volumen lógico tiene un tamaño de
984 MiB repartidos en 246 extensiones lógicas.

```
[root@localhost alumno]# lvdisplay /dev/repositorio/novelas
--- Logical volume ---
LV Path                /dev/repositorio/novelas
LV Name                 novelas
VG Name                 repositorio
LV UUID                 Gz0Ead-PwbN-s0e3-PyTO-6yze-dbPU-Fd12Ns
LV Write Access         read/write
LV Creation host, time  localhost.localdomain, 2019-06-10 12:29:26 +0200
LV Status                available
# open                   0
LV Size                  984,00 MiB
Current LE               246
Segments                 1
Allocation               inherit
```

```
Read ahead sectors      auto
- currently set to      768
Block device            253:8
```

2.d.-

```
[root@localhost alumno]# mkfs.ext4 /dev/repositorio/novelas
mke2fs 1.44.2 (14-May-2018)
Se está creando un sistema de ficheros con 251904 bloques de 4k y 62976 nodos-i
UUID del sistema de ficheros: d92c741f-8195-4627-a3fd-c9ecfbf50df9
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376

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 62976 nodos-i, siendo este el número máximo de ficheros que se pueden
# crear.
```

2.e.-

```
# Montamos en primer lugar el nuevo sistema de ficheros en /home/novelas y vemos
# que los permisos del sistema de ficheros no son los adecuados:
```

```
[root@localhost alumno]# mount /dev/repositorio/novelas /home/novelas/
[root@localhost alumno]# ls -ld /home/novelas/
drwxr-xr-x. 3 root root 4096 jun 10 12:36 /home/novelas/
```

```
# Los cambiamos de forma que respondan a lo especificado en los apartados
# correspondientes del ejercicio anterior:
```

```
[root@localhost alumno]# chown :escritor /home/novelas/
[root@localhost alumno]# chmod 770 /home/novelas/
[root@localhost alumno]# ls -ld /home/novelas/
drwxrwx---. 3 root escritor 4096 jun 10 12:36 /home/novelas/
```

```
# Una vez que el sistema de ficheros tiene los permisos adecuados, lo añadimos
# al fichero /etc/fstab para que se monte tras cada arranque:
```

```
[root@localhost alumno]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Jun 21 08:54:43 2018
#
# 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
#
/dev/mapper/fedora-root / ext4 defaults 1 1
UUID=58a955b2-9209-4d41-bb53-d8fe87178cf0 /boot ext4 defaults 1 2
/dev/mapper/fedora-swap swap swap defaults 0 0
hostExamenISO /home/alumno/ExamenISO 9p trans=virtio,version=9p2000.L,noauto,x-systemd.device-timeout=30,user 0 0
/dev/repositorio/novelas /home/novelas ext4 defaults 0 0
\end{lstlisting}
```

2.f.-

```
# Tenemos que crear un enlace simbólico, ya que, como se ha visto, los enlaces
# físicos a directorios no están permitidos. Para ello, tras pasar a usuario
# arturo, usamos la orden ln con la opción -s:
```

```
[root@localhost alumno]# su -l arturo
[arturo@localhost ~]$ ln -s /home/novelas/ ./mis-novelas
[arturo@localhost ~]$ ls -l
total 0
lrwxrwxrwx. 1 arturo bestseller 14 jun 10 16:48 mis-novelas -> /home/novelas/
```

```
# Obsevamos que el enlace tiene un tamaño de 14 bytes, que son los caracteres
# que configuran la ruta /home/novelas/ (el contenido del fichero).
```

2.g.-

```
[root@localhost alumno]# vim /etc/fstab      # Cambiamos las opciones de montaje a default,usrquota
[root@localhost alumno]# mount -o remount /home/novelas
[root@localhost alumno]# mount | grep novelas
/dev/mapper/repositorio-novelas on /home/novelas type ext4 (rw,relatime,seclabel,quota,usrquota,stripe
=32)
[root@localhost alumno]# quotacheck -nm /home/novelas
[root@localhost alumno]# quotaon /home/novelas
```

```
[root@localhost alumno]# setquota -u arturo 300M 400M 0 10k /home/novelas
[root@localhost alumno]# setquota -u jeronimo 300M 400M 0 10k /home/novelas
[root@localhost alumno]# setquota -t 432000 0 /home/novelas/
\end{lstlisting}
```

*** Ejercicio 3 ***

3.a.-

```
[root@localhost ~]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@localhost ~]# crontab -l
00 04 * * sun tar cJf /home-domingo.tar.xz /home
```

3.b.-

```
[root@localhost ~]# crontab -e
crontab: installing new crontab
[root@localhost ~]# crontab -l
00 04 * * sun tar cJf /home-domingo.tar.xz /home
00 04 * * wed tar cJf /home-miercoles.tar.xz --newer-mtime="/home-domingo.tar.xz" /home
00 04 * * fri tar cJf /home-viernes.tar.xz --newer-mtime="/home-domingo.tar.xz" /home
```

*** Ejercicio 4 ***

4.a.-

```
[root@localhost ~]# lspci -v
...
00:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8100/8101L/8139 PCI Fast Ethernet
Adapter (rev 20)
Subsystem: Red Hat, Inc. QEMU Virtual Machine
Physical Slot: 3
Flags: bus master, fast devsel, latency 0, IRQ 11
I/O ports at c000 [size=256]
Memory at febd9000 (32-bit, non-prefetchable) [size=256]
Expansion ROM at feb80000 [disabled] [size=256K]
Kernel driver in use: 8139cp
Kernel modules: 8139cp, 8139too
```

Como podemos ver, se utiliza el módulo 8139cp.

4.b.-

```
[root@localhost ~]# ionice -c 1 -n 0 sleep 1000 &
[1] 1964
[root@localhost ~]# ionice -p 1964
realtime: prioridad 0
```

Para mostrar que el proceso se ejecuta con la prioridad de E/S solicitada, en
lugar de ionice también se puede usar la orden iotop si se ordena por el campo
PRIO en orden ascendente (es decir, cuando aparece PRIO<).

Listing 1: bitacora.txt

Guiones Shell

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

1. #!/bin/bash

```
if test $# -ne 3
then
    echo "USO: $0 <directorio_origen> <directorio_destino> <nombre_fichero>" >&2
    exit 1
fi

if test \( ! -d $1 \) -o \( ! -r $1 \) -o \( ! -x $1 \)
then
    echo "Error: el primer parametro debe ser un directorio con permiso de lectura y ejecución"
    >&2
    exit 2
fi

if test \( ! -d $2 \) -o \( ! -w $2 \) -o \( ! -x $2 \)
then
    echo "Error: el segundo parametro debe ser un directorio con permisos de escritura y ejecución"
    " >&2
    exit 3
```



```

fi

ruta_fic_ori="$1/$3"
ruta_fic_des="$2/$3"

if test ! -f $ruta_fic_ori -o ! -r $ruta_fic_ori
then
    echo "Error: el tercer parametro debe ser un fichero regular con permiso de lectura en el
    directorio origen" >&2
    exit 4
fi
if test -e $ruta_fic_des
then
    echo "Error: el tercer parametro no debe existir en el directorio destino" >&2
    exit 5
fi

echo "Copiando fichero: $ruta_fic_ori --> $ruta_fic_des"
cp $ruta_fic_ori $ruta_fic_des

```

Script 2: copia_ddf.sh

2. #!/bin/bash

```

num_fic=0
while read linea
do
    directorio=$(echo $linea | cut -f1 -d" ")
    extension=$(echo $linea | tr -s " " | cut -f2 -d" ")
    let num_fic+=$(find $directorio -type f -maxdepth 1 -printf "cambiando nombre de fichero: %f\n
    " -exec mv {} {}. $extension \; | wc -l)
done < $1
echo "Número de ficheros cuyo nombre ha sido cambiado: $num_fic"

```

Script 3: pon_extension.sh

3. #!/bin/bash

```

n_linea=0
while read linea
do
    if test ! -z "$linea"
    then
        n_palabras=$(echo $linea | wc -w)
        echo $n_linea:$linea "(-> $n_palabras)"
        let n_linea+=1
        let total_palabras+=n_palabras
    fi
done < "$1" | sort -nr | cut -f2 -d':'

```

Script 4: invierte_y_cuenta.sh