



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)**

- (1,20 puntos) Realiza las siguientes tareas de gestión de usuarios sin modificar ficheros de configuración a mano (salvo que se diga lo contrario):
  - (0,24 puntos) Haz lo necesario para que, a partir de este momento, cualquier nuevo usuario que se cree vea, al iniciar sesión, un resumen del estado actual y de la actividad del sistema desde que se arrancó el ordenador. Nota: usa la orden `vmstat` para mostrar la información indicada.
  - (0,36 puntos) Crea los usuarios `java` y `python`, cada uno con su propio grupo primario. Muestra, a través de la orden adecuada, los UID de los usuarios creados y los grupos a los que pertenece cada uno.
  - (0,12 puntos) Como usuario `java` demuestra que lo que has hecho en el primer apartado funciona.
  - (0,48 puntos) Sin crear nuevos grupos, haz lo necesario para que el usuario `python`, y solo este usuario, pueda acceder al contenido del directorio personal del usuario `java` sin restricciones (evidentemente, el usuario `java` también debe poder seguir accediendo a su propio directorio personal). Muestra, a través de las órdenes adecuadas, los UID y grupos a los que pertenecen los usuarios creados, y los atributos de los directorios personales de cada uno de ellos.
- (1,92 puntos) Realiza las siguientes tareas sobre gestión de discos:
  - (0,36 puntos) Usando particionado DOS, particiona los discos `sdb`, `sdc` y `sdd` con dos particiones primarias cada uno. Cada partición debe tener un tamaño de 400 MiB y ser de tipo «Linux LVM». No es necesario que indiques todos los pasos a la hora de crear las particiones, basta con que adjuntes la salida de la tabla de particiones resultante en cada disco.
  - (0,36 puntos) A continuación, crea un grupo de volúmenes llamado `iso-julio` formado por la primera partición de los discos `sdb` y `sdc` (date cuenta que no se usa `sdd`). ¿Cuál es el tamaño del grupo de volúmenes en extensiones físicas? ¿Cuántas extensiones físicas aporta cada volumen físico? Además de adjuntar la salida de la orden que proporciona la información que necesitas, debes responder con precisión a las preguntas que se te hacen.

c) (0,72 puntos) A partir del nuevo grupo de volúmenes, crea un volumen lógico de tamaño 300 MiB y nombre `lv-datos` que tenga el mejor rendimiento y fiabilidad posibles teniendo en cuenta las características del grupo de volúmenes creado. ¿Qué tipo de volumen lógico has creado? Justifica que ese, y no otro, es el tipo adecuado para cumplir con las restricciones de rendimiento y fiabilidad que se te indican. ¿Cuántas extensiones lógicas tiene el nuevo volumen lógico? ¿Cuántas extensiones físicas ocupa en total? Recuerda que además de adjuntar la salida de la orden que proporciona la información que necesitas, debes responder con precisión a las preguntas que se te hacen.

d) (0,48 puntos) Sin eliminar el volumen lógico creado, sustituye el volumen físico `sdb1` por la primera partición del disco `sdd` dentro del grupo de volúmenes `iso-julio`, de tal manera que `sdb1` deje de formar parte del grupo de volúmenes. Demuestra que, efectivamente, ahora el dispositivo `sdb1` no pertenece al grupo de volúmenes, pero sí `sdd1`. Muestra también el total de extensiones físicas libres en cada volumen físico de nuestro grupo de volúmenes.

3. (0,72 puntos) Resuelve los siguientes ejercicios sobre la gestión de los recursos del sistema:

a) (0,36 puntos) ¿Cuántos paquetes de nombre `kernel` hay instalados en el sistema? ¿Cuándo se instaló el de mayor versión? Observa que los paquetes a considerar deben llamarse `kernel`, no contener la palabra «kernel» en el nombre. Así, un paquete como `kernel-core` no se debe tener en cuenta. ¿Se ha modificado algún fichero del paquete `systemd` desde que se instaló? Si fuera así, indica, al menos, uno de los ficheros modificados.

b) (0,36 puntos) Planifica una tarea para que a las 20:30 de hoy guarde en el fichero `/root/memoria.txt` el estado de la memoria RAM y de la memoria de intercambio en ese instante. Tras programar la tarea, muestra que, efectivamente, está pendiente de ser ejecutada. Finalmente, borra la tarea para que no llegue a ejecutarse.

4. (1,16 puntos) Realiza los ejercicios que tienes a continuación relacionados con la gestión de la E/S:

a) (0,40 puntos) ¿Qué licencia de software utiliza el módulo `amdgpu`? ¿De qué otros módulos depende? ¿Qué módulos dependen de `amdgpu`? Ten en cuenta que, en estas dos últimas preguntas, la respuesta puede ser «ninguno».

b) (0,36 puntos) ¿Qué planificador de disco está usando `sdb`? Sabiendo que `sdb` es un disco duro, cambia su planificador al que, *a priori*, ofrecería un peor rendimiento. Justifica tu decisión sobre el planificador elegido.

c) (0,40 puntos) Dada la siguiente orden `find`,

```
find /etc /home /usr /var -type f -exec grep -l alumno {} \; >/dev/null 2>&1 &
```

haz lo necesario para que la misma se ejecute en segundo plano de tal manera que sus peticiones de disco tengan la menor prioridad posible dentro de la clase «mejor esfuerzo (*best effort*)». Respecto al uso de la CPU, la orden se debe ejecutar con una prioridad base igual a la del padre más 10. Demuestra, con las órdenes adecuadas, que la orden `find` se ejecuta con las prioridades indicadas. Nota: la orden se tiene que ejecutar desde el principio con esas prioridades, es decir, no es posible cambiar las prioridades una vez la orden esté en ejecución.

### 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, `validahex.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) (1,5 puntos) Implementa un guion shell llamado `validahex.sh` que tiene que recibir uno o más números hexadecimales como parámetros. El guion debe comprobar si los parámetros dados son números hexadecimales válidos (es decir, números formados por los caracteres del 0 al 9 y del A al F precedidos por 0X, sin importar si las letras son mayúsculas o minúsculas), en cuyo caso debe devolver 0 como código de salida. En caso contrario, es decir, si hay al menos un parámetro que no es un número hexadecimal, 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 validahex.sh
Uso: validahex.sh hexadecimal...
[alumno@localhost ~]$ echo $?
2
[alumno@localhost ~]$ bash validahex.sh a 0X 9 0x10014400 0xabcdefg 0xabcdef
Los siguientes parámetros no son números hexadecimales válidos:
a
0X
9
0xabcdefg
[alumno@localhost ~]$ echo $?
1
```

```
[alumno@localhost ~]$ bash validahex.sh 0x10014400 0xAAFF0023 0Xabab
[alumno@localhost ~]$ echo $?
0
```

Fichero de solución: **validahex.sh**

- b) (1,5 puntos) Implementa un guion shell llamado `margenes.sh` que reciba como parámetro un único fichero de texto y como resultado muestre el contenido de ese mismo fichero, pero empezando y terminando cada línea con el carácter «|». Los caracteres «|» deben estar alineados verticalmente tanto en el margen izquierdo como en el derecho.

El guion no debe comprobar los parámetros (supondremos que son siempre correctos).

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 margenes.sh 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.|
```

Observa que los caracteres «|» del margen izquierdo siempre son el primer carácter de cada línea, mientras que los del margen derecho se encuentran alineados a continuación del último carácter de la línea más larga del fichero de entrada. El texto en sí quedará alineado a la izquierda, por lo que, para aquellas líneas que sean más cortas que la más larga, habrá que rellenar con espacios a la derecha para poder dibujar la barra vertical del margen derecho en su posición correcta.

Fichero de solución: **margenes.sh**

- c) (2,0 puntos) Implementa un guion shell llamado `nmasgrandes.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 (incluidos sus subdirectorios), ordenados de mayor a menor tamaño. Tras ellos debe mostrar un mensaje (ver ejemplo más abajo) donde aparezca el tamaño en bytes total de los ficheros mostrados y el porcentaje que ese tamaño representa sobre el tamaño total de todos los ficheros regulares que hay en el directorio (incluidos subdirectorios).

El guion no debe comprobar los parámetros (supondremos que son siempre correctos). Además, si la búsqueda en el directorio produce errores (por ejemplo, por falta de permisos), dichos errores no deben aparecer por pantalla.

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

```
[alumno@localhost ~]$ bash nmasgrandes.sh /etc/ 0

Los ficheros mostrados ocupan 0 bytes (0%) del total
de todos los ficheros regulares del directorio '/etc/'
[alumno@localhost ~]$ bash nmasgrandes.sh /etc/ 1
Fichero: /etc/udev/hwdb.bin, tamaño: 12029919 bytes

Los ficheros mostrados ocupan 12029919 bytes (43%) del total
de todos los ficheros regulares del directorio '/etc/'
[alumno@localhost ~]$ bash nmasgrandes.sh /etc/ 5
Fichero: /etc/udev/hwdb.bin, tamaño: 12029919 bytes
Fichero: /etc/selinux/targeted/policy/policy.33, tamaño: 3642510 bytes
Fichero: /etc/services, tamaño: 701745 bytes
Fichero: /etc/brltty/Contraction/zh-tw.cti, tamaño: 640012 bytes
Fichero: /etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt, tamaño: 624010 bytes

Los ficheros mostrados ocupan 17638196 bytes (63%) del total
de todos los ficheros regulares del directorio '/etc/'
```

Fichero de solución: **nmasgrandes.sh**

# Soluciones

## Administración

\*\*\* Ejercicio 1 \*\*\*

1.a.-

```
[alumno@localhost ExamenISO]$ su
Contraseña:
[root@localhost ExamenISO]# vim /etc/skel/.bash_profile
[root@localhost ExamenISO]# cat /etc/skel/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs
vmstat
```

1.b.-

```
[root@localhost ExamenISO]# useradd java
[root@localhost ExamenISO]# useradd python
[root@localhost ExamenISO]# id java python
uid=1001(java) gid=1001(java) grupos=1001(java)
uid=1002(python) gid=1002(python) grupos=1002(python)
```

1.c.-

```
[root@localhost ExamenISO]# su -l java
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
 r  b   swpd   free   buff   cache   si   so    bi    bo    in   cs us sy id wa st
 0  0         0 468748 65296 562448    0    0   763   201  444 667 10  5 81  3  0
[java@localhost ~]$ exit
cerrar sesión
```

1.d.-

```
[root@localhost ExamenISO]# usermod -a -G java python
[root@localhost ExamenISO]# chmod 770 /home/java/
[root@localhost ExamenISO]# id java python
uid=1001(java) gid=1001(java) grupos=1001(java)
uid=1002(python) gid=1002(python) grupos=1002(python),1001(java)
[root@localhost ExamenISO]# ls -ld /home/java
drwxrwx---. 4 java java 4096 may 30 18:59 /home/java
[root@localhost ExamenISO]# ls -ld /home/python
drwx-----. 3 python python 4096 may 30 18:59 /home/python
[root@localhost ExamenISO]#
```

\*\*\* Ejercicio 2 \*\*\*

2.a.-

```
[root@localhost ExamenISO]# fdisk -l /dev/sd[bcd]
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: dos
Identificador del disco: 0x73e6d182

Disposit.  Inicio Comienzo   Final Sectores Tamaño Id Tipo
/dev/sdb1          2048  821247    819200   400M 8e Linux LVM
/dev/sdb2          821248 1640447    819200   400M 8e Linux LVM
```

```
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: dos
Identificador del disco: 0xa71d790a
```

```
Disposit.  Inicio Comienzo   Final Sectores Tamaño Id Tipo
/dev/sdc1          2048  821247    819200   400M 8e Linux LVM
```

```
/dev/sdc2          821248 1640447   819200   400M 8e Linux LVM
```

```
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: dos
Identificador del disco: 0x01bdf142
```

```
Disposit.  Inicio Comienzo   Final Sectores Tamaño Id Tipo
/dev/sdd1          2048   821247   819200   400M 8e Linux LVM
/dev/sdd2          821248 1640447   819200   400M 8e Linux LVM
```

2.b.-

```
[root@localhost ExamenISO]# vgcreate iso-julio /dev/sd[bc]1
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdc1" successfully created.
Volume group "iso-julio" successfully created
[root@localhost ExamenISO]# vgsdisplay iso-julio
--- Volume group ---
VG Name                iso-julio
System ID
Format                 lvm2
Metadata Areas         2
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                 2
Act PV                  2
VG Size                 792,00 MiB
PE Size                 4,00 MiB
Total PE                198
Alloc PE / Size         0 / 0
Free PE / Size          198 / 792,00 MiB
VG UUID                 767r5F-pmT2-JbUM-CIIe-C42w-SJYr-trTl37
```

# El nuevo grupo de volúmenes tiene un total de 198 extensiones físicas (línea Total PE)

```
[root@localhost ExamenISO]# pvdisplay /dev/sd[bc]1
--- Physical volume ---
PV Name                /dev/sdb1
VG Name                iso-julio
PV Size                 400,00 MiB / not usable 4,00 MiB
Allocatable             yes
PE Size                 4,00 MiB
Total PE                99
Free PE                 99
Allocated PE            0
PV UUID                 Wc4k5m-jDSv-Vsrl-yBQj-wAF5-rIlf-qNPNkk

--- Physical volume ---
PV Name                /dev/sdc1
VG Name                iso-julio
PV Size                 400,00 MiB / not usable 4,00 MiB
Allocatable             yes
PE Size                 4,00 MiB
Total PE                99
Free PE                 99
Allocated PE            0
PV UUID                 veln4J-1krc-UwzT-hlJZ-eSNq-zm68-P48Dbt
```

# Cada volumen físico aporta 99 extensiones físicas.

2.C.-

```
[root@localhost ExamenISO]# lvcreate -m 1 -L 300M -n lv-datos iso-julio
Logical volume "lv-datos" created.
```

# Al tener solo dos volúmenes físicos, se ha creado un volumen lógico reflejado  
# para mejorar la fiabilidad (si falla un disco, el volumen lógico sigue  
# funcionando con el disco que queda) y el rendimiento en las operaciones de  
# lectura. El rendimiento en las escrituras será similar al de un único disco.  
# Con dos volúmenes físicos se podrían haber creado también un volumen lógico  
# lineal o uno repartido, pero ninguno de ellos mejora la fiabilidad.

```
[root@localhost ExamenISO]# lvdisplay /dev/iso-julio/lv-datos
--- Logical volume ---
LV Path                /dev/iso-julio/lv-datos
LV Name                 lv-datos
VG Name                 iso-julio
LV UUID                 pAv03b-u2fu-sdTS-To01-tzOq-MLmf-cZxsCs
LV Write Access         read/write
LV Creation host, time  localhost.localdomain, 2023-05-30 19:07:14 +0200
LV Status                available
# open                  0
LV Size                 300,00 MiB
Current LE              75
Mirrored volumes        2
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:6
```

# El nuevo volumen lógico tiene 75 extensiones lógicas (línea Current LE).

```
[root@localhost ExamenISO]# vgdisplay iso-julio
--- Volume group ---
VG Name                 iso-julio
System ID
Format                  lvm2
Metadata Areas          2
Metadata Sequence No    3
VG Access                read/write
VG Status                resizable
MAX LV                  0
Cur LV                 1
Open LV                 0
Max PV                  0
Cur PV                 2
Act PV                  2
VG Size                 792,00 MiB
PE Size                  4,00 MiB
Total PE                 198
Alloc PE / Size          152 / 608,00 MiB
Free PE / Size           46 / 184,00 MiB
VG UUID                 767r5F-pmT2-JbUM-CIIe-C42w-SJYr-trTl37
```

# Como vemos, en el grupo de volúmenes hay ahora 152 extensiones físicas  
# ocupadas (línea Alloc PE / Size), que son las que ocupa el volumen lógico  
# creado pues es el único que existe en el grupo de volúmenes.

2.d.-

```
[root@localhost ExamenISO]# vgextend iso-julio /dev/sdd1
Physical volume "/dev/sdd1" successfully created.
Volume group "iso-julio" successfully extended
[root@localhost ExamenISO]# pvmove /dev/sdb1 /dev/sdd1
/dev/sdb1: Moved: 23,68%
/dev/sdb1: Moved: 98,68%
/dev/sdb1: Moved: 100,00%
[root@localhost ExamenISO]# vgreduce iso-julio /dev/sdb1
Removed "/dev/sdb1" from volume group "iso-julio"
[root@localhost ExamenISO]# vgdisplay -v iso-julio
--- Volume group ---
VG Name                 iso-julio
System ID
Format                  lvm2
Metadata Areas          2
Metadata Sequence No    9
VG Access                read/write
VG Status                resizable
MAX LV                  0
Cur LV                 1
Open LV                 0
Max PV                  0
Cur PV                 2
Act PV                  2
VG Size                 792,00 MiB
PE Size                  4,00 MiB
Total PE                 198
Alloc PE / Size          152 / 608,00 MiB
Free PE / Size           46 / 184,00 MiB
VG UUID                 767r5F-pmT2-JbUM-CIIe-C42w-SJYr-trTl37

--- Logical volume ---
```

```

LV Path                /dev/iso-julio/lv-datos
LV Name                lv-datos
VG Name                iso-julio
LV UUID                pAvO3b-u2fu-sdTS-To01-tzOq-MLmf-cZxsCs
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2023-05-30 19:07:14 +0200
LV Status              available
# open                 0
LV Size                300,00 MiB
Current LE             75
Mirrored volumes       2
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    256
Block device           253:6

--- Physical volumes ---
PV Name                /dev/sdc1
PV UUID                ve1n4J-1krc-UwzT-h1JZ-eSNq-zm68-P48Dbt
PV Status              allocatable
Total PE / Free PE    99 / 23

PV Name                /dev/sdd1
PV UUID                t0CXWC-gelS-NyNM-0tRu-Ua4j-n2uI-APkUsI
PV Status              allocatable
Total PE / Free PE    99 / 23

# Vemos que el grupo de volúmenes ahora contiene sdc1 y sdd1 como volúmenes
# físicos. Ha desaparecido sdb1.

```

```

[root@localhost ExamenISO]# pvdisplay /dev/sdc1
--- Physical volume ---
PV Name                /dev/sdc1
VG Name                iso-julio
PV Size                400,00 MiB / not usable 4,00 MiB
Allocatable            yes
PE Size                4,00 MiB
Total PE               99
Free PE                23
Allocated PE           76
PV UUID                ve1n4J-1krc-UwzT-h1JZ-eSNq-zm68-P48Dbt

```

```

[root@localhost ExamenISO]# pvdisplay /dev/sdd1
--- Physical volume ---
PV Name                /dev/sdd1
VG Name                iso-julio
PV Size                400,00 MiB / not usable 4,00 MiB
Allocatable            yes
PE Size                4,00 MiB
Total PE               99
Free PE                23
Allocated PE           76
PV UUID                t0CXWC-gelS-NyNM-0tRu-Ua4j-n2uI-APkUsI

```

# En cada volumen físico hay 23 extensiones físicas libres

\*\*\* Ejercicio 3 \*\*\*

3.a.-

```

[root@localhost ExamenISO]# rpm -q kernel
kernel-5.7.7-200.fc32.x86_64
kernel-5.8.8-200.fc32.x86_64
kernel-5.11.22-100.fc32.x86_64

```

# Hay tres paquetes instalados con el mismo nombre (kernel), aunque diferentes versiones.

```

[root@localhost ExamenISO]# rpm -qi kernel-5.11.22-100.fc32.x86_64 | grep "Install Date"
Install Date: mié 14 jul 2021 12:06:34

```

# Se instaló en la fecha mostrada.

```

[root@localhost ExamenISO]# rpm -V systemd
.M..... c /etc/locale.conf
.M..... c /etc/machine-id
.M..... g /var/log/btmp
.M....G.. g /var/log/journal

```

# Han sufrido cambios los 4 ficheros que aparecen en el listado anterior. En concreto, en todos ellos han cambiado los permisos (M) y en el último ha

```
# cambiado, además, su grupo (G).
```

3.b.-

```
[root@localhost ExamenISO]# at 20:30
warning: commands will be executed using /bin/sh
at> free > /root/memoria.txt
at> <EOT>
job 1 at Tue May 30 20:30:00 2023
[root@localhost ExamenISO]# atq
1 Tue May 30 20:30:00 2023 a root
[root@localhost ExamenISO]# atrm 1
[root@localhost ExamenISO]# atq
[root@localhost ExamenISO]#
```

\*\*\* Ejercicio 4 \*\*\*

4.a.-

```
[root@localhost ExamenISO]# modinfo amdgpu | grep -i license
license:      GPL and additional rights
```

# Utiliza una licencia GPL con modificaciones.

```
[root@localhost ExamenISO]# modprobe --show-depends amdgpu
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/drm.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/media/cec/core/cec.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/drm_kms_helper.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/ttm/ttm.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/drm_ttm_helper.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/i2c/algos/i2c-algo-bit.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/scheduler/gpu-sched.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/iommu/amd/iommu_v2.ko.xz
insmod /lib/modules/5.11.22-100.fc32.x86_64/kernel/drivers/gpu/drm/amd/amdgpu/amdgpu.ko.xz
```

# El módulo amdgpu depende de todos los módulos que aparecen en las líneas  
# anteriores (exceptuando la última, que corresponde al propio módulo).

```
[root@localhost ExamenISO]# grep amdgpu /lib/modules/5.11.22-100.fc32.x86_64/modules.dep
kernel/drivers/gpu/drm/amd/amdgpu/amdgpu.ko.xz: kernel/drivers/iommu/amd/iommu_v2.ko.xz kernel/drivers/
gpu/drm/scheduler/gpu-sched.ko.xz kernel/drivers/i2c/algos/i2c-algo-bit.ko.xz kernel/drivers/gpu/
drm/drm_ttm_helper.ko.xz kernel/drivers/gpu/drm/ttm/ttm.ko.xz kernel/drivers/gpu/drm/drm_kms_helper
.ko.xz kernel/drivers/media/cec/core/cec.ko.xz kernel/drivers/gpu/drm/drm.ko.xz
```

# No hay ningún módulo que dependa de amdgpu, ya que este módulo no aparece a  
# la derecha de los dos puntos (:) en ninguna línea del fichero modules.dep

4.b.-

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

# El disco sdb está usando el planificador BFQ. Al ser un disco duro, el  
# planificar que, a priori, ofrecería un peor rendimiento sería «none». Este  
# planificador no ordena las peticiones para reducir los movimientos de las  
# cabezas de disco, ya que atiende las peticiones en orden de llegada. En este  
# sentido, es equivalente a un planificador FCFS que, como hemos visto en teoría,  
# ofrece los peores resultados en promedio en cuanto a rendimiento.

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

4.c.-

```
[root@localhost ExamenISO]# ionice -c 2 -n 7 nice -10 find /etc /home /usr /var -type f -exec grep -l
alumno {} \; >/dev/null 2>&1 &
[1] 5583
[root@localhost ExamenISO]# ps alx | head -1
F  UID  PID  PPID PRI  NI   VSZ  RSS WCHAN  STAT TTY          TIME COMMAND
[root@localhost ExamenISO]# ps alx | grep find
4      0  5583   2126  30   10 218252  3184 -        RN   pts/1        0:00 find /etc /home /usr /var -type
f -exec grep -l alumno {} ;
0      0  8645   2126  20    0 216096   656 -        S+   pts/1        0:00 grep --color=auto find
[root@localhost ExamenISO]# ionice -p 5583
best-effort: prioridad 7
```

Listing 1: bitacora.txt



## Guiones Shell

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

### 1. #!/bin/bash

```
test $# -eq 0 && echo "Uso: $0 hexadecimal..." >&2 && exit 2
salida=0
for e in "$@"
do
    if ! echo $e | grep -qxi "0x[0-9A-F]\+"
    then
        if test $salida -eq 0
        then
            echo Los siguientes parámetros no son números hexadecimales válidos: >&2
            salida=1
        fi
        echo $e >&2
    fi
done

exit $salida
```

### Script 2: validahex.sh

### 2. #!/bin/bash

```
longmax=$(wc -L $1 | cut -f1 -d " ")

while read linea
do
    echo -n "|"
    echo -n "$linea"
    longlinea=$(echo "$linea" | wc -c)
    let espacios=longmax-longlinea+1
    for i in $(seq 1 $espacios)
    do
        echo -n " "
    done
    echo "|"
done < $1
```

### Script 3: margenes.sh

### 3. #!/bin/bash

```
tamttotal=0
for t in $(find "$1" -type f -printf "%s\n" 2> /dev/null)
do
    let tamttotal+=t
done

tamficheros=0
for t in $(find "$1" -type f -printf "%s\n" 2> /dev/null | sort -nr | head -n $2)
do
    let tamficheros+=t
done

find "$1" -type f -printf "Fichero: %p, tamaño: %s bytes\n" 2> /dev/null | sort -k4,4nr | head -n $2

let porcentaje=(tamficheros*100)/tamttotal

echo
echo "Los ficheros mostrados ocupan $tamficheros bytes ($porcentaje%) del total"
echo "de todos los ficheros regulares del directorio '$1'"
```

### Script 4: nmasgrandes.sh