



EJERCICIOS DE KERNEL Y PROCESOS

Ejercicio 1) Práctica sobre procesos y prioridades

Crear un proceso que se pueda lanzar en background en el sistema. Ejecutarlo cuatro veces en background (simultáneamente). Realizar las siguientes operaciones mientras se encuentra en ejecución:

Para ejecutar un proceso en segundo plano se utiliza el “&” desde la terminal, para esta práctica voy a crear un script que duerma durante 100 segundos infinitamente.

```
nodo021@nodo021:~$ cat proceso.sh
#!/bin/bash

while true; do

sleep 100

done
```

El único inconveniente de este script es que su estado en vez de R es S (suspended).

```
nodo021@nodo021:~$ ./proceso.sh &
[3] 7057
nodo021@nodo021:~$ ./proceso.sh &
[4] 7059
nodo021@nodo021:~$ ./proceso.sh &
[5] 7061
nodo021@nodo021:~$ ./proceso.sh &
[6] 7063
```

- Localizar los PIDs de las instancias de los procesos

Para localizar los PIDs de las instancias de los procesos utilizamos el comando “ps -A” que muestra todos los procesos, podemos filtrar con grep por el nombre del comando ejecutado. También podemos utilizar el comando “pgrep -l <nombre_proceso>” para buscarlo.

```
nodo021@nodo021:~$ ps -A | grep proceso
7057 pts/0    00:00:00 proceso.sh
7059 pts/0    00:00:00 proceso.sh
7061 pts/0    00:00:00 proceso.sh
7063 pts/0    00:00:00 proceso.sh
```

El PID es el primer número a la izquierda.

- Identificar los procesos "padre" de cada una de ellas

Para identificar el padre de un proceso podemos utilizar el comando “ps -o ppid = <PID>”.

```
nodo021@nodo021:~$ ps -o ppid= -p 7057
3319
```

Podemos seguir ejecutando este comando hasta que llegemos al proceso padre de todos con PID 1.

- Realizar una representación de los procesos en ejecución en forma de árbol desde la shell

Para representar un árbol de procesos podemos utilizar el comando “pstree”.

```
nodo021@nodo021:~$ pstree
systemd--ModemManager--2*[{ModemManager}]
--NetworkManager--2*[{NetworkManager}]
--VBoxDRMClient--3*[{VBoxDRMClient}]
--VBoxService--8*[{VBoxService}]
--accounts-daemon--2*[{accounts-daemon}]
--acpid
--avahi-daemon--avahi-daemon
--colord--2*[{colord}]
--cron
--cups-browsed--2*[{cups-browsed}]
--cupsd--3*[{dbus}]
--dbus-daemon
--gdm3--gdm-session-wor--gdm-wayland-ses--gnome-session-b--2*[{gnome-sessi+
--2*[{gdm3}]--2*[{gdm-session-wor}]
--gnome-keyring-d--3*[{gnome-keyring-d}]
--irqbalance--{irqbalance}
--2*[kerneloops]
--networkd-dnspat
--packagekitd--2*[{packagekitd}]
--polkitd--2*[{polkitd}]
--power-profiles--2*[{power-profiles-}]
--rsyslogd--3*[{rsyslogd}]
--rtkit-daemon--2*[{rtkit-daemon}]
--snapd--9*[{snapd}]
--switcheroo-cont--2*[{switcheroo-cont}]
systemd--(sd-pam)
--atspi2-registr--2*[{atspi2-registr}]
--dbus-daemon
--dconf-service--2*[{dconf-service}]
--evolution-adre--5*[{evolution-adre}]
--evolution-calen--8*[{evolution-calen}]
--evolution-sourc--3*[{evolution-sourc}]
--2*[gjs--6*[{gjs}]
--gnome-session-b--atspi-bus-laun--dbus-daemon
--3*[{atspi-bus-laun}]
--evolution-alarm--5*[{evolution-alarm}]
--gsd-disk-utilit--2*[{gsd-disk-utilit}]
--update-notifier--3*[{update-notifier}]
--3*[{gnome-session-b}]
--gnome-session-c--{gnome-session-c}
--gnome-shell--Xwayland
--gjs--7*[{gjs}]
--19*[{gnome-shell}]
--gnome-shell-cal--5*[{gnome-shell-cal}]
--gnome-terminal--gnome-terminal--3*[{gnome-terminal-}]
--gnome-terminal--bash--man--2*[{man--man}]
--3*[{man}]
--pager
--preconv
--man--5*[{man}]
--pager
--preconv
--4*[{proceso.sh--sleep]
```

- Modificar la prioridad de alguno de los procesos para que se ejecute proporcionalmente el doble de tiempo que el resto

Con el comando “ps -fc” podemos ver otras columnas como la de prioridad del proceso.

Podemos observar que la prioridad de todos los procesos que hemos lanzado es de 19. Cuanto menor sea el número de prioridad, mayor será su prioridad.

```
nodo021@nodo021:~$ ps -fc
UID          PID    PPID  CLS   PRI  STIME  TTY          TIME CMD
nodo021      3701      3400  TS     19  19:27 pts/3        00:00:00 bash
nodo021     10838      3701  TS     19  20:03 pts/3        00:00:00 /bin/bash ./proceso.sh
nodo021     10839     10838  TS     19  20:03 pts/3        00:00:00 sleep 100
nodo021     10840      3701  TS     19  20:03 pts/3        00:00:00 /bin/bash ./proceso.sh
nodo021     10841     10840  TS     19  20:03 pts/3        00:00:00 sleep 100
nodo021     10842      3701  TS     19  20:03 pts/3        00:00:00 /bin/bash ./proceso.sh
nodo021     10843     10842  TS     19  20:03 pts/3        00:00:00 sleep 100
nodo021     10844      3701  TS     19  20:03 pts/3        00:00:00 /bin/bash ./proceso.sh
nodo021     10845     10844  TS     19  20:03 pts/3        00:00:00 sleep 100
nodo021     11076      3701  TS     19  20:05 pts/3        00:00:00 ps -fc
```

Con el comando “nice -n 10 <proceso>” podemos ejecutar el mismo pero con una prioridad diferente. Con el comando “renice -n <PRIORITY> -p <PID>” podemos cambiar la prioridad de un proceso que se está ejecutando.

```
nodo021@nodo021:~$ renice 10 -p 10838
10838 (process ID) prioridad anterior 0, nueva prioridad 10
nodo021@nodo021:~$ ps -fc | grep 10838
nodo021      10838      3701  TS      9  20:03 pts/3        00:00:00 /bin/bash ./proceso.sh
```



- Terminar la ejecución de uno de los procesos con el comando "kill"

Para terminar la ejecución de un proceso utilizamos el comando "kill -9 <PID>", la opción -9 es para matar todos los procesos que puedes para ese PID.

```
nodo021@nodo021:~$ kill -9 7322
nodo021@nodo021:~$ ps -afc
```

UID	PID	PPID	CLS	PRI	STIME	TTY	TIME	CMD
nodo021	2614	2602	TS	19	19:12	tty2	00:00:00	/usr/libexec/gnome-session-bin
nodo021	7316	7298	TS	19	19:27	pts/0	00:00:00	/bin/bash ./proceso.sh
nodo021	7318	7298	TS	19	19:28	pts/0	00:00:00	/bin/bash ./proceso.sh
nodo021	7320	7298	TS	19	19:28	pts/0	00:00:00	/bin/bash ./proceso.sh
nodo021	7374	7324	TS	19	19:29	pts/1	00:00:00	man kill
nodo021	7382	7374	TS	19	19:29	pts/1	00:00:00	pager
nodo021	7392	7316	TS	19	19:29	pts/0	00:00:00	sleep 100
nodo021	7393	7318	TS	19	19:29	pts/0	00:00:00	sleep 100
nodo021	7394	7320	TS	19	19:29	pts/0	00:00:00	sleep 100
nodo021	7395	2547	TS	19	19:29	pts/0	00:00:00	sleep 100
nodo021	7398	7298	TS	19	19:30	pts/0	00:00:00	ps -afc
[4]+	Terminado (killed)							./proceso.sh

- Cambiar la ejecución de otro de los procesos a "foreground"

Para traer devuelta el proceso utilizamos el comando "fg % <JOB_ID>"

Con el comando "jobs" podemos ver los procesos que están siendo ejecutados en el background.

```
nodo021@nodo021:~$ jobs
[1]  Ejecutando          ./proceso.sh &
[2]-  Ejecutando          ./proceso.sh &
[3]+  Ejecutando          ./proceso.sh &
nodo021@nodo021:~$ fg %1
./proceso.sh
```

En este caso como el proceso es un script que lo único que hace ese esperar no se muestra nada por pantalla, pero si que podemos observar que la terminal está bloqueada por el proceso.



- Poner el proceso en modo STOPPED. Verificar su estado con el comando "ps"

Para poner un proceso en modo STOPPED podemos utilizar "CTRL + Z" cuando está siendo utilizado desde la misma terminal o podemos mandar una señal con el comando kill.

He aprovechado la situación anterior para utilizar "CTRL + Z". Con el comando "jobs" podemos observar que el proceso está detenido.

```
nodo021@nodo021:~$ jobs
[1]+  Detenido                ./proceso.sh
[2]   Ejecutando              ./proceso.sh &
[3]-  Ejecutando              ./proceso.sh &
```

Para visualizar el proceso parado con el comando "ps" podemos incluir la opción "ax" para mostrar la columna de STAT e identificar si el proceso está parado mediante la letra "T".

```
nodo021@nodo021:~$ ps ax | grep proceso
 7316 pts/0    T           0:00 /bin/bash ./proceso.sh
 7318 pts/0    S           0:00 /bin/bash ./proceso.sh
 7320 pts/0    S           0:00 /bin/bash ./proceso.sh
 7505 pts/0    S+          0:00 grep --color=auto proceso
```

STAT CODE	DESCRIPTION
D	Uninterruptible sleep (usually IO)
R	Running or runnable (on run queue)
S	Interruptible sleep (waiting for an event to complete)
T	Stopped, either by a job control signal or because it is being traced
W	paging (not valid since the 2.6.xx kernel)
X	dead (should never be seen)
Z	Defunct ("zombie") process, terminated but not reaped by its paren

- Detener la ejecución de este proceso con el comando "kill" pero esta vez indicando el número de "job" en lugar del PID del proceso.

Para este cometido podemos utilizar el "%" para representar el número de job.

```
nodo021@nodo021:~$ kill -9 %1

[1]+  Detenido                ./proceso.sh
nodo021@nodo021:~$ jobs
[1]+  Terminado (killed)     ./proceso.sh
[2]   Ejecutando             ./proceso.sh &
[3]-  Ejecutando             ./proceso.sh &
```

- Identificar qué procesos quedan en ejecución de los iniciales y verificar cómo se reparten el tiempo de uso de la CPU.

Quedan 2 procesos activos de los 4 creados.

```
nodo021@nodo021:~$ ps ax | grep proceso
7318 pts/0    S          0:00 /bin/bash ./proceso.sh
7320 pts/0    S          0:00 /bin/bash ./proceso.sh
7535 pts/0    S+         0:00 grep --color=auto proceso
```

Se puede visualizar el tiempo de CPU de los procesos con el comando "htop" que sirve para monitorizar los recursos del sistema.

```

5.3% Tasks: 137, 287 thr; 1 running
4.6% Load average: 0.12 0.08 0.07
3.3% Uptime: 01:00:41
3.4%
Mem[ 985M/5.78G]
Swap[ 0K/2.00G]

PID USER      PRI  NI  VIRT   RES   SHR  S CPU% MEM%   TIME+  Command
3326 nodo021  20   0  2792M 61444 44984 S   0.0  1.0  0:00.01 gjs /usr/share/gnome-shell/extensions/dingrastersoft.com/ding.js -E -P /usr/share/gnome-shell/extensions/dingrastersoft.com -M 0 -D 0:0
3327 nodo021  20   0  2792M 61444 44984 S   0.0  1.0  0:00.00 gjs /usr/share/gnome-shell/extensions/dingrastersoft.com/ding.js -E -P /usr/share/gnome-shell/extensions/dingrastersoft.com -M 0 -D 0:0
3328 nodo021  20   0  2676  1564  1280 S   0.0  0.0  0:00.00 VBoxClient --vmsvga-session
3329 nodo021  20   0  151M  2204  1920 S   0.0  0.0  0:00.51 VBoxClient --vmsvga-session
3330 root      20   0  277M  2176  2048 S   0.0  0.0  0:00.48 VBoxDMClient
3331 nodo021  20   0  151M  2204  1920 S   0.0  0.0  0:00.00 VBoxClient --vmsvga-session
3332 nodo021  20   0  151M  2204  1920 S   0.0  0.0  0:00.51 VBoxClient --vmsvga-session
3333 nodo021  20   0  2792M 61444 44984 S   0.0  1.0  0:00.00 gjs /usr/share/gnome-shell/extensions/dingrastersoft.com/ding.js -E -P /usr/share/gnome-shell/extensions/dingrastersoft.com -M 0 -D 0:0
3335 nodo021  20   0  612M 14592 12160 S   0.0  0.2  0:00.00 xdg-desktop-portal
3392 nodo021  20   0  38512 19200 11520 S   0.0  0.3  0:00.07 python3 /usr/bin/gnome-terminal --wait
3395 nodo021  20   0  377M 28036 20356 S   0.0  0.5  0:00.00 gnome-terminal.real --wait
3396 nodo021  20   0  377M 28036 20356 S   0.0  0.5  0:00.00 gnome-terminal.real --wait
3398 nodo021  20   0  377M 28036 20356 S   0.0  0.5  0:00.00 gnome-terminal.real --wait
3399 nodo021  20   0  377M 28036 20356 S   0.0  0.5  0:00.00 gnome-terminal.real --wait
3401 nodo021  20   0  553M 60340 45396 S   0.0  1.0  0:00.00 gnome-terminal-server
3403 nodo021  20   0  553M 60340 45396 S   0.0  1.0  0:00.00 gnome-terminal-server
3404 nodo021  20   0  553M 60340 45396 S   0.0  1.0  0:00.02 gnome-terminal-server
3418 nodo021  20   0  13752 5120  3712 S   0.0  0.1  0:00.06 bash
3446 nodo021  20   0  485M 29828 22020 S   0.0  0.5  0:00.25 update-notifier
3454 nodo021  20   0  485M 29828 22020 S   0.0  0.5  0:00.12 update-notifier
3456 nodo021  20   0  485M 29828 22020 S   0.0  0.5  0:00.00 update-notifier
3457 nodo021  20   0  485M 29828 22020 S   0.0  0.5  0:00.00 update-notifier
3554 nodo021  30  10  699M 200M  104M S   0.0  3.4  0:03.94 python3 /usr/bin/update-manager --no-update --no-focus-on-map
3555 nodo021  30  10  699M 200M  104M S   0.0  3.4  0:00.00 python3 /usr/bin/update-manager --no-update --no-focus-on-map
3557 nodo021  30  10  699M 200M  104M S   0.0  3.4  0:00.01 python3 /usr/bin/update-manager --no-update --no-focus-on-map
3558 nodo021  30  10  699M 200M  104M S   0.0  3.4  0:00.00 python3 /usr/bin/update-manager --no-update --no-focus-on-map
3647 nodo021  20   0  906M  363M  143M S   0.0  6.1  0:00.00 gnome-shell
3681 nodo021  20   0  13752 5120  3712 S   0.0  0.1  0:00.07 bash
3701 nodo021  20   0  13992 5376  3840 S   0.0  0.1  0:00.44 bash
10641 nodo021  20   0  13752 5120  3712 S   0.0  0.1  0:00.01 bash
10649 root      20   0  17212 6240  5240 S   0.0  0.1  0:00.00 sudo login operador
10650 root      20   0  17212 2552  1536 S   0.0  0.0  0:00.00 sudo login operador
10651 root      20   0  16412 5804  4864 S   0.0  0.1  0:00.02 login
10766 operador 20   0  13820 5376  3840 S   0.0  0.1  0:00.09 -bash
10831 root      20   0  1432M 11384 19456 S   0.0  0.5  0:00.00 snmpd
10838 nodo021  30  10  12720 34040 2456 S   0.0  0.1  0:00.01 bash ./proceso.sh
Next: S-F3prev EscCancel Search: proceso
```

- Detener la ejecución de uno de los procesos con el comando "killall"

Con el comando "killall" nos permite eliminar todos los procesos mediante el nombre, grupo, etc., pero en este caso como el nombre del proceso es el mismo no podemos utilizar esta opción. También existe la opción "-f" para hacer match del nombre exacto del proceso con un límite de 15 caracteres.

La ejecución del comando sería "killall -9 proceso"

- Detener la ejecución de otro con "pkill"

A diferencia de los comando "kill" y "killall", el comando "pkill" nos permite utilizar expresiones regular y otras opciones útiles.

El comando a ejecutar sería el mismo que con "kill" y "killall", "pkill <NOMBRE_PROCESO>".

```
nodo021@nodo021:~$ pkill proceso
[2]- Terminado      ./proceso.sh
[3]+ Terminado      ./proceso.sh
```

No he encontrado información para utilizar "pkill" con un PID. Como era de esperar se eliminan los dos procesos restantes.

- Ejecutar el proceso de nuevo en background, pero esta vez con el comando NOHUP. A continuación, cerrar la terminal y abrir otra. Tratar de localizar el proceso. ¿Existe o ha desaparecido? ¿Por qué?

```
nodo021@nodo021:~$ nohup ./proceso.sh &
[1] 7764
nodo021@nodo021:~$ nohup: se descarta la entrada y se añade la salida a 'nohup.out'
```

El comando nohup permite que el proceso no se muera cuando el proceso padre que lo ha lanzado termine, en este caso la terminal.

```
nodo021@nodo021:~$ ps aux | grep proceso
nodo021      7764  0.0  0.0  12588  3456 ?        S    20:12   0:00 /bin/bash ./proceso.sh
nodo021      7819  0.0  0.0  11716  2560 pts/0    S+   20:13   0:00 grep --color=auto proceso
```

Se puede observar que sigue existiendo el proceso.

- Detener la ejecución de este proceso con algún comando que envíe una señal, indicando la denominación de la señal en TEXTO (SIG.... etc.)

El comando “kill” tiene la opción “-s” para pasar el código de la señal.

```
nodo021@nodo021:~$ kill -s KILL 7764
nodo021@nodo021:~$ ps aux | grep proceso
nodo021      7918  0.0  0.0  11716  2560 pts/0    S+   20:20   0:00 grep --color=auto proceso
```

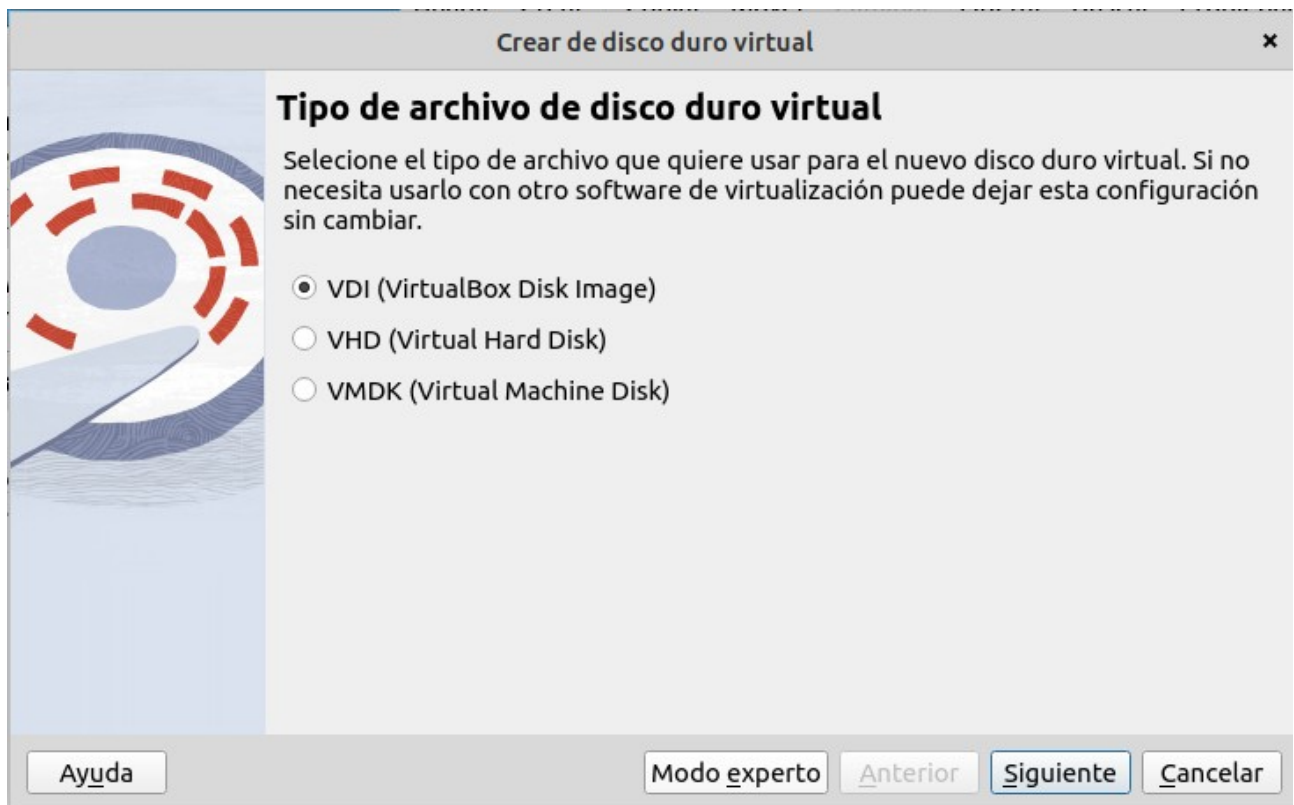
Ejercicio 2) Trabajo con discos, particiones y sistemas de ficheros

- Utilizando el software de virtualización Virtualbox, asignar nuevos discos de 8 GB cada uno a los sistemas Linux configurados en nuestros equipos.

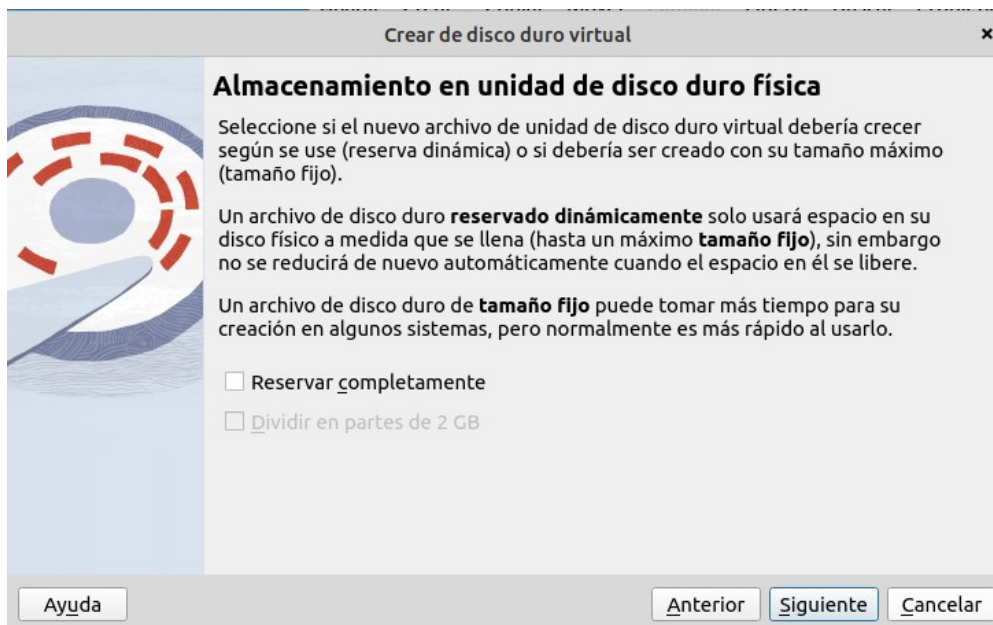
Para crear discos nuevos en virtualbox debemos acceder a la herramienta de medios, desde ahí podemos crear y añadir los discos. En este caso utilizamos la opción “Crear”.



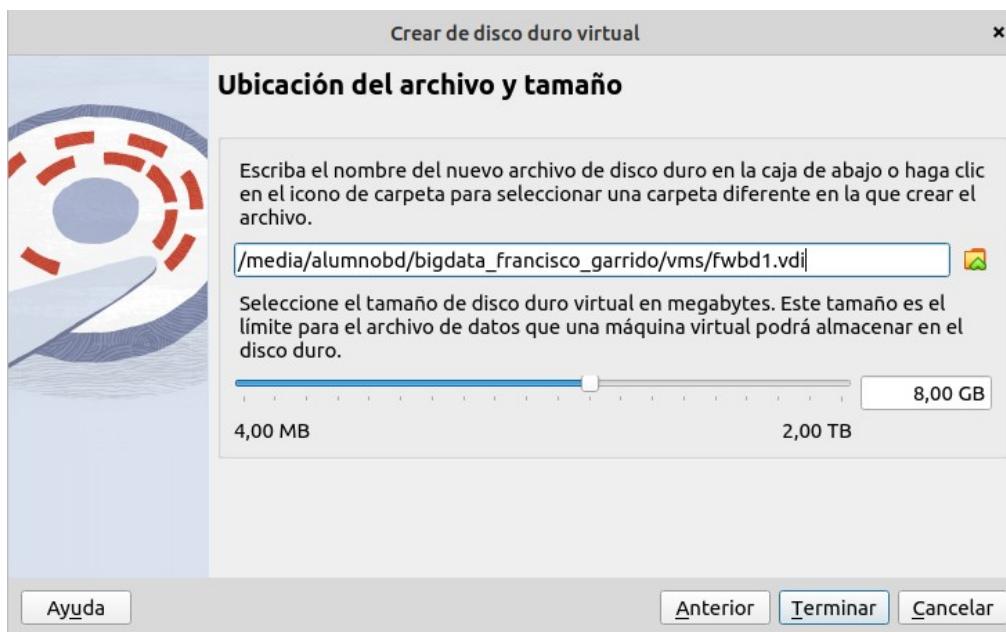
Creamos un disco VDI (VirtualBox Disk Image).



En el siguiente paso no marcamos ninguna opción y el disco se creará dinámicamente.



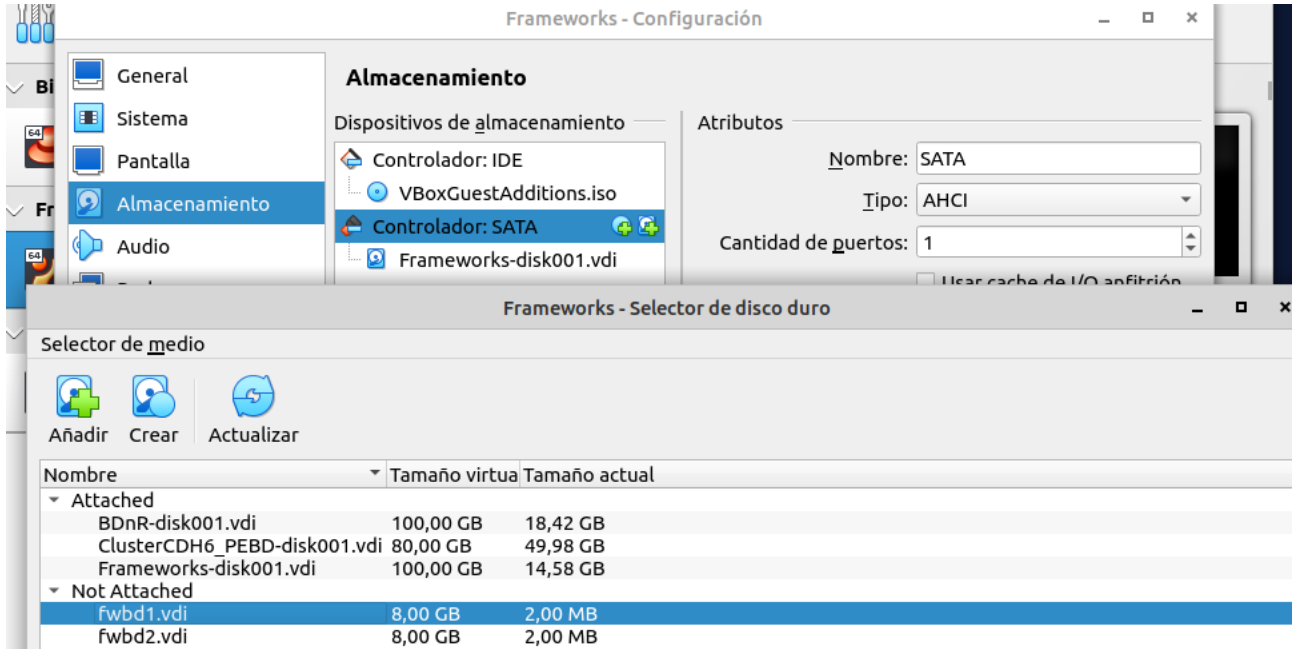
Seleccionamos la ubicación y indicamos el nombre y el tamaño de nuestro disco virtual.



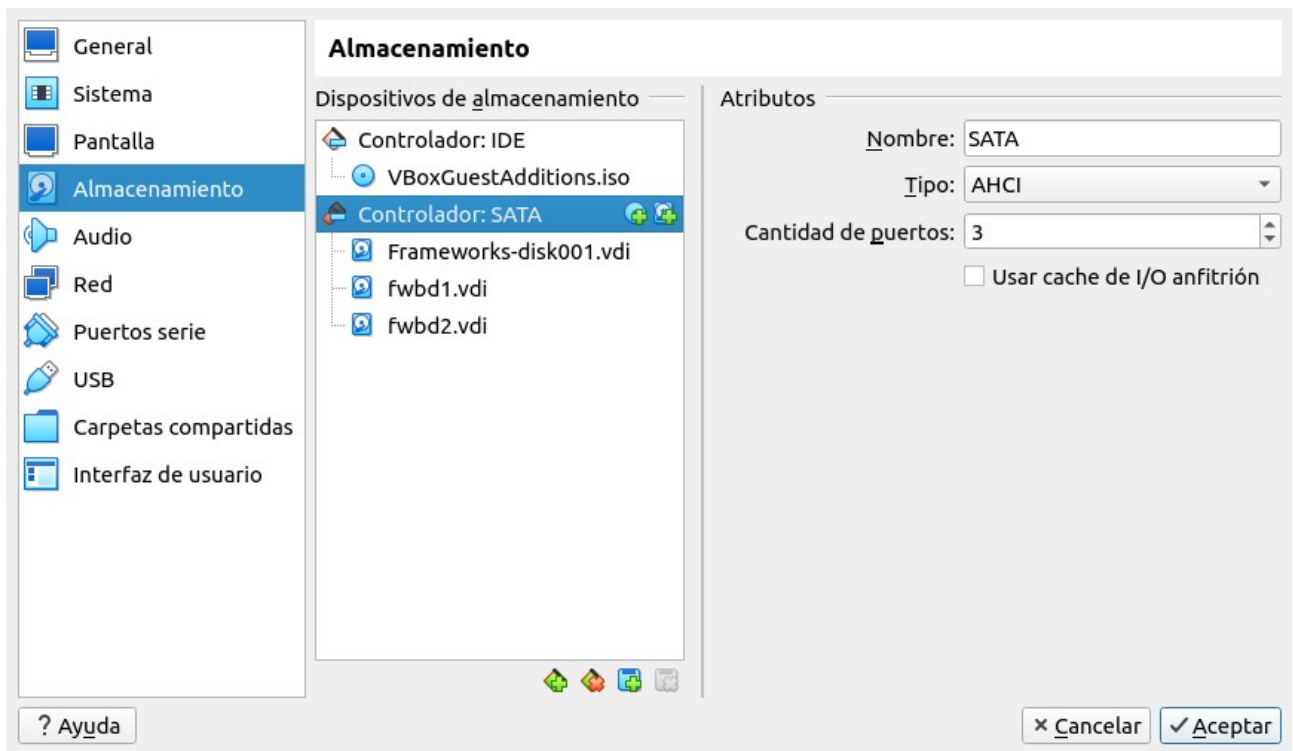
Repetimos el proceso para el siguiente disco.

Nombre	Tamaño virtual	Tamaño actual
BDnR-disk001.vdi	100,00 GB	18,42 GB
ClusterCDH6_PEBD-disk001.vdi	80,00 GB	49,98 GB
Frameworks-disk001.vdi	100,00 GB	14,58 GB
fwbd1.vdi	8,00 GB	2,00 MB
fwbd2.vdi	8,00 GB	2,00 MB

Para añadir los discos a la maquina tenemos que acceder a la configuración de la misma y el apartado de almacenamiento. Si no es en caliente, se debe apagar la maquina antes de añadirlos.



Una vez añadidos, se pueden visualizar en el apartado de almacenamiento.



- Particionar cada uno de los discos con dos particiones primarias de 4 GB cada una.

Lo primero que debemos hacer es identificar los discos en la maquina podemos utilizar el comando “lsblk”. En este caso vemos que se identifican como sdb y sdc.

Se lo he pasado al comando “tail” para no mostrar todos los discos con sus particiones.

```
nodo021@nodo021:~$ lsblk | tail -n 3
sdb      8:16    0      8G    0 disk
sdc      8:32    0      8G    0 disk
sr0     11:0     1     51M    0 rom  /media/nodo021/VBox_GAs_7.0.14
```

Con el comando “fdisk /dev/sdx” podemos crear particiones con una guía de partición seleccionando la opción “n”.

```
nodo021@nodo021:~$ sudo fdisk /dev/sdc
Bienvenido a fdisk (util-linux 2.37.2).
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 0xcf979ba3.

Orden (m para obtener ayuda): m
```

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primary, 0 extended, 4 free)
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1
```

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (1 primary, 0 extended, 3 free)
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (2-4, valor predeterminado 2): 2
Primer sector (8390656-16777215, valor predeterminado 8390656):
```

```
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8390656-16777215, valor predeterminado 16777215):
Crea una nueva partición 2 de tipo 'Linux' y de tamaño 4 GiB.
```

Repetimos el proceso para el otro disco.

- Conseguir que las particiones sean reconocidas por el sistema.

Con el comando `fdisk` las particiones las reconoce el sistema automáticamente, con el comando `lsblk` lo podemos comprobar.

```
sdb      8:16  0      8G  0 disk
├─sdb1    8:17  0      4G  0 part
├─sdb2    8:18  0      4G  0 part
└─sdc      8:32  0      8G  0 disk
   ├─sdc1    8:33  0      4G  0 part
   └─sdc2    8:34  0      4G  0 part
```

- En cada uno de los discos, generar un sistema de ficheros de tipo `ext4` en la primera partición y otro de tipo `xfs` en la segunda.

Para formatear las particiones se puede utilizar el comando `mkfs -t <formato> /dev/sdx`.

```
nodo021@nodo021:~$ sudo mkfs -t ext4 /dev/sdb1
mke2fs 1.46.5 (30-Dec-2021)
Se está creando un sistema de ficheros con 1048576 bloques de 4k y 262144 nodos-i
UUID del sistema de ficheros: a98aef31-ea20-4b01-97f7-5212c91124bc
Respalos del superbloque guardados en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de archivos: hecho

nodo021@nodo021:~$ sudo mkfs -t ext4 /dev/sdc1
mke2fs 1.46.5 (30-Dec-2021)
Se está creando un sistema de ficheros con 1048576 bloques de 4k y 262144 nodos-i
UUID del sistema de ficheros: 9ee5cfbc-921a-413f-856a-2093385515ca
Respalos del superbloque guardados en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de archivos: hecho
```

XFS es un sistema de archivos de 64 bits con registro de bitácora de alto rendimiento creado por SGI para su implementación de UNIX (<https://es.wikipedia.org/wiki/XFS>)

Con el comando `mkfs.xfs /dev/sdx` podemos formatear la partición (sin indicar la geometría de la franja y el tamaño de los trozos de RAID), con `lsblk -f` comprobamos el formato de particiones.

NOTA: Es necesario instalar `xfsprogs` para utilizar el comando `mkfs.xfs`.

```
sdb
├─sdb1    ext4    1.0      a98aef31-ea20-4b01-97f7-5212c91124bc
├─sdb2    xfs      f6be6948-96dd-49c7-a755-2e3b7faf8a91
└─sdc
   ├─sdc1    ext4    1.0      9ee5cfbc-921a-413f-856a-2093385515ca
   └─sdc2    xfs      dee30f6e-1b8b-444b-a275-dc2b4dc0d9d6
```

- Crear un nuevo usuario "operador" con los parámetros por defecto

He utilizado el comando “sudo adduser operador” para que sea más sencillo.

```
nodo021@nodo021:~$ cat /etc/passwd | grep operador
operador:x:1002:1002:operador,,,:/home/operador:/bin/bash
```

- Montar los sistemas de ficheros recién creados en los puntos de montaje ~/extra/aux1 y ~/extra/aux2 del usuario "operador" con los permisos oportunos para que este usuario pueda acceder a ellos.

Lo primero de todo es crear las carpetas con el comando “mkdir”.

Lo siguiente es montar las particiones en las carpetas creadas con el comando “mount -t <formato> /dev/sdx”

```
nodo021@nodo021:~$ sudo mkdir -p /home/operador/extra/aux1
nodo021@nodo021:~$ sudo mkdir -p /home/operador/extra/aux2
nodo021@nodo021:~$ lsblk -f | tail -n 6
└─sdb1 ext4      1.0                a98aef31-ea20-4b01-97f7-5212c91124bc
└─sdb2 xfs              f6be6948-96dd-49c7-a755-2e3b7faf8a91
sdc
└─sdc1 ext4      1.0                9ee5cfbc-921a-413f-856a-2093385515ca
└─sdc2 xfs              dee30f6e-1b8b-444b-a275-dc2b4dc0d9d6
sr0    iso9660  Joliet Extension VBOX_GAS_7.0.14 2024-01-15-14-48-13-93      0    1
00% /media/nodo021/VBox_GAS_7.0.14
nodo021@nodo021:~$ sudo mount -t ext4 /dev/sdb1 /home/operador/extra/aux1
nodo021@nodo021:~$ sudo mount -t ext4 /dev/sdc1 /home/operador/extra/aux2
```

Por último modificamos los permisos de las carpetas para que los usuarios puedan leer y ejecutar solamente.

Por lo que tenemos los permisos de esta manera “rwx-xr-x” para las carpetas. También podemos utilizar el comando “chown” para que la carpeta pertenezca al usuario operador u otros comandos que haría una función similar.

```
nodo021@nodo021:~$ sudo chmod -R 755 /home/operador/
```

Se que la opción -R no es la más indicada pero son permisos poco permisivos, esta opción es para hacer el chmod recursivamente.

- Entrar en el sistema con el usuario "operador" y verificar que tiene acceso a los sistemas recién montados.

```
operador@nodo021:~$ ls -la extra/
total 16
drwxr-xr-x 4 root    root    4096 feb 27 19:48 .
drwxr-xr-x 4 operador operador 4096 feb 27 19:55 ..
drwxr-xr-x 3 root    root    4096 feb 27 19:34 aux1
drwxr-xr-x 3 root    root    4096 feb 27 19:34 aux2
```

BIBLIOGRAFÍA

- <https://www.baeldung.com/linux/find-parent-pid>
- <https://devhints.io/bash>
- <https://www.atlantic.net/vps-hosting/how-to-set-linux-process-priority-using-nice-and-renice-commands/>
- <https://www.baeldung.com/linux/foreground-background-process>
- https://ioflood.com/blog/nohup-linux-command/#:~:text=To%20run%20a%20process%20in,is%20simple%3A%20nohup%20your_command%20%26%20.&text=In%20this%20example%2C%20we're,background%20using%20the%20nohup%20command.
- <https://www.digitalocean.com/community/tutorials/create-a-partition-in-linux>
- https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/managing_file_systems/assembly_creating-an-xfs-file-system-getting-started-with-xfs
- https://medium.com/@The_CodeConductor/how-to-change-linux-process-priority-af55d074bf3d