Volúmenes lógicos y RAID software en GNU/Linux

CDA 2021/22

 $15~\mathrm{de}$ septiembre de 2021

Índice

1.	Des	cripción	1		
2.	Ente	orno de prácticas	2		
	2.1.	Software de virtualización VIRTUALBOX	2		
	2.2.	Imágenes a utilizar	2		
	2.3.	Discos disponibles en la máquina virtual	3		
3.	. LVM (Logical Volume Manager)				
	3.1.	PREVIO: Conceptos y funcionamiento de LVM	3		
		3.1.1. Elementos	3		
		3.1.2. Funcionalidades adicionales	4		
		3.1.3. Resumen de comandos relevantes	4		
	3.2.	EJEMPLO 1: Volumen lógico sobre /dev/sdc y /dev/sdd $\ldots \ldots \ldots$	5		
	3.3.	EJEMPLO 2: Ampliación de volumen lógico con $/dev/sde$	9		
4.	4. RAID por software		14		
	4.1.	PREVIO: Fundamentos RAID software en GNU/Linux	14		
		4.1.1. Funcionamiento	14		
		4.1.2. Resumen de comandos relevantes	14		
	4.2.	EJEMPLO 1: Niveles RAID (RAID0, RAID1, RAID5)	15		
	4.3.	EJEMPLO 2: Niveles RAID anidados (RAID10, RAID01)	19		
5.	Tare	Tareas entregables			
	5.1.	Gestión de un RAID5	23		
	5.2.	Documentación a entregar	24		

1. Descripción

Descripción y ejemplo de uso de volúmenes lógicos (LVM) en GNU/Linux.

• LVM2 ((Linux) Logical Volume Manager, versión 2) permite la gestión de volúmenes lógicos en entornos GNU/Linux

- LVM "agrupa" dispositivos de bloque "físicos" (PVs: *physical volumes*), presentes en el sistema, en grupos de volúmenes (VGs: Volume Groups)
- LVM "reparte" esos VGs para crear nuevos dispositivos de bloques "lógicos" (LVs: logical volumes)

Para el sistema los LVs creados se comportan como dispositivos de bloque convencionales (pueden particionarse, formatearse, montarse, comaprtirse, etc)

Descripción y ejemplo de uso de RAID por software en GNU/Linux.

- El kernel Linux cuenta con una implementación software de diversos niveles RAID (tanto "oficiales" como específicos de Linux-RAID)
- La herramienta mdadm ofrece las utilidades de línea de comandos para crear y gestionar los arrays de discos

para definir nuevos dispositivos de bloque con diversas "mejoras"

- mayor capacidad de almacenamiento (RAID0, RAID5)
- mayor velocidad de lectura (todos) y/o escritura (RAID0)
- mayor tolerancia a fallos (RAID1, RAID5)

Entorno de prácticas 2.

Software de virtualización VIRTUALBOX 2.1.

En estas prácticas se empleará el software de virtualización VIRTUALBOX para simular los equipos GNU/Linux sobre los que se realizarán las pruebas.

- Página principal: http://virtualbox.org
- Más información: http://es.wikipedia.org/wiki/Virtualbox

2.2.Imágenes a utilizar

- 1. Scripts de instalación
 - para GNU/Linux: ejercicio-lvm-raid.sh alumno@pc: \$ sh ejercicio-lvm-raid.sh
 - para MS windows: ejercicio-lvm-raid.ps1 Powershell.exe -executionpolicy bypass -file ejercicio-lvm-raid.ps1

Notas:

- Se pedirá un identificador (sin espacios) para poder reutilizar las versiones personalizadas de las imágenes creadas (usad por ejemplo el nombre del grupo de prácticas o el login LDAP)
- En ambos scripts la variable **\$DIR_BASE** especifica donde se descargarán las imágenes y se crearán las MVs. Por defecto en GNU/Linux será en \$HOME/CDA2122 y en Windows en C:/CDA2122. Puede modificarse antes de lanzar los scripts para hacer la instalación en otro directorio más conveniente (disco externo, etc)
- Es posible descargar las imágenes comprimidas manualmente (o intercambiarlas con USB), basta descargar los archivos con extensión .vdi.zip de http://ccia.esei.uvigo.es/docencia/CDA/2122/practicas/ y copiarlos en el directorio anterior (**\$DIR_BASE**) para que el script haga el resto.

RAID "combina" dispositivos de bloque, presentes en el sistema, mediante *striping* (RAID0, RAID5) *mirroring* (RAID1) bloques de paridad (RAID5)

- Si no lo hacen desde el script anterior, se pueden arrancar las instancias VIRTUALBOX desde el interfaz gráfico de VirtualBOX o desde la línea de comandos con VBoxManage startvm <nombre MV>_<id>
- 2. Imágenes descargadas
 - base_cda.vdi (0,82 GB comprimida, 2,9 GB descomprimida): Imagen genérica (común a todas las MVs) que contiene las herramientas a utilizar

Contiene un sistema Debian 9 con herramientas gráficas y un entorno gráfico ligero LXDE (*Lighweight X11 Desktop Environment*) [LXDE].

- swap1GB.vdi: Disco de 1 GB formateado como espacio de intercambio (SWAP)
- 3. Usuarios configurados e inicio en el sistema
 - Usuarios disponibles

login	password
root	purple
usuario	usuario

• Acceso al entorno gráfico una vez logueado (necesario para poder copiar y pegar desde/hacia el anfitrión)

root@datos:~# startx

 Habilitar copiar y pegar desde/hacia el anfitrión en el menú Dispositivos -> Portapapeles compartido -> bidir de la ventana de la máquina virtual.

2.3. Discos disponibles en la máquina virtual

- /dev/sda: disco principal, 10 GB (con la partición ext4 /dev/sda1 montada en /)
- /dev/sdb: disco para swap, 1 GB (swap en la partición /dev/sdb1)
- /dev/sdc: disco para ejercicios, 100 MB (no particionado)
- /dev/sdd: disco para ejercicios, 100 MB (no particionado)
- /dev/sde: disco para ejercicios, 200 MB (no particionado)
- /dev/sdf: disco para ejercicios, 100 MB (no particionado)

3. LVM (Logical Volume Manager)

3.1. PREVIO: Conceptos y funcionamiento de LVM

Soluciones similares bajo otros entornos: Wikipedia: Logical volume management (ver tabla en sección Implementations)

3.1.1. Elementos

Componentes gestionados "externamente" por el administrador

- Volúmenes físicos (*PV: Physical volume*): dispositivos de bloque ("físicos" o no) sobre los que se organizan los volúmenes gestionados por LVM.
- Grupos de volúmenes (VG: Volume group): agrupación de volúmenes físicos
- Volúmenes lógicos (*LV: Logical volume*): porciones de un VG puestas a disposición del sistema como dispositivos de bloques (constituidas a su vez por porciones de los PVs que componen el VG)

Componentes "internos", gestionados por LVM

- Extensiones físicas (*PE: Physical extents*): unidades mínimas (≈ bloques) en las que se organizan los volúmenes físicos (PVs) por parte del software LVM
- Extensiones lógicas (*LE: Logical extents*): unidades mínimas (≈ bloques) en las que se organizan los volúmenes lógicos (LVs) por parte del software LVM



NIVELES DE ABSTRACCIÓN EN LVM (Linux Volume Manager)

3.1.2. Funcionalidades adicionales

- Organización del espacio según niveles RAID: la asignación de LEs (*Logical Extents*) sobre los PEs (*Physical Extents*) disponibles puede configurarse de modo análogo a como se organizan los niveles RAID (con o sin replicación de PEs [*mirroring*]).
- Soporte de *snapshots*: posibilidad de realizar "instantáneas" donde se refleje la situación concreta de los volúmenes gestionados en un instante determinado

3.1.3. Resumen de comandos relevantes

Instalación (ya hecho en la MV del ejercicio)

root@datos:~# apt-get install lvm2

Gestión de volúmenes físicos

- pvcreate <dispositivo/particion> da de alta un nuevo dispositivo de bloques como un "volumen físico" disponible para formar parte de un grupo de volúmenes
- pvremove <dispositivo/particion> eliminar un "volumen físico" que fue dado de alta para formar parte de un grupo de volúmenes
- pvdisplay [-m] muestra información de los "volúmenes físicos"

Gestión de grupos de volúmenes

 vgcreate [opciones] <nombre grupo><dispositivo/particion> crea un "grupo de volúmenes" empleando los "volúmenes físicos" indicados

- vgremove elimina un "grupo de volúmenes" previamente creado
- vgextend altera el tamaño de uno de los "grupos de volúmenes" existentes
- vgdisplay muestra información de los "grupos de volúmenes"

Gestión de volúmenes lógicos

- lvcreate [especificación tamaño (-L -1)] --name <nombre LV><nombre grupo> crea un "volúmen lógico" dentro del "grupo de volúmenes indicado"
- lvremove elimina un "volumen lógico" previamente creado
- lvextend altera el tamaño de uno de los "volúmenes lógicos" existentes
- lvdisplay muestra información de los "volúmenes lógicos"

3.2. EJEMPLO 1: Volumen lógico sobre /dev/sdc y /dev/sdd

1. Crear una partición primaria de tipo *Linux LVM* (identificador 8e) en /dev/sdc y /dev/sdd, empleando fdisk, cfdisk, parted ó gparted, asignándole todo el espacio disponible [99MB] (previamente se debe crear la tabla de particiones).

Haciéndolo desde línea de comando con el modo interactivo de parted sería:

```
root@datos:~# parted /dev/sdcroot@datos:~# parted /dev/sdd(parted) mklabel msdos(parted) mklabel msdos(parted) mkpart primary 1M 100%(parted) mkpart primary 1M 100%(parted) set 1 lvm on(parted) set 1 lvm on(parted) quit(parted) quit
```

Se pueden comprobar los nuevos dispositivos disponibles con lsblk o parted -1

```
root@datos:~# lsblk
root@datos:~# parted -1
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 105MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start
                End
                       Size
                              Type
                                       File system Flags
1
       1049kB 105MB 104MB
                             primary
                                                   lvm
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdd: 105MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start
                End
                       Size
                              Туре
                                       File system Flags
1
       1049kB 105MB 104MB primary
                                                   lvm
. . .
```

2. Registrar ambas particiones como volúmenes físicos (PVs) de LVM.

```
root@datos:~# pvcreate /dev/sdc1 /dev/sdd1
Physical volume "/dev/sdc1" successfully created
Physical volume "/dev/sdd1" successfully created
```

Puede comprobarse la lista de PVs disponibles con pvdisplay (añadiendo la opción -m se detalla la información de los *physical extents (PE)* que los componen) o también con pvs.

```
root@datos:~# pvdisplay -m
  "/dev/sdc1" is a new physical volume of "99,00 MiB"
  --- NEW Physical volume ---
 PV Name
                       /dev/sdc1
 VG Name
 PV Size
                       99,00 MiB
 Allocatable
                       NO
 PE Size
                       0
 Total PE
                       0
 Free PE
                       0
 Allocated PE
                       0
 PV UUID
                       nda2kl-Rv1z-iCWo-zHGF-mOwG-Z1cO-3FNmqd
 "/dev/sdd1" is a new physical volume of "99,00 MiB"
 --- NEW Physical volume ---
 PV Name
                       /dev/sdd1
 VG Name
 PV Size
                       99,00 MiB
 Allocatable
                       NO
 PE Size
                       0
 Total PE
                       0
 Free PE
                       0
 Allocated PE
                       0
 PV UUID
                       u6c07y-wSjs-OBY0-ID0Z-7CNC-QLPQ-8Lo31B
root@datos:~# pvs
 ΡV
            VG
                   Fmt Attr PSize PFree
  /dev/sdc1
                   lvm2 a-- 99,00m 99,00m
                   lvm2 a-- 99,00m 99,00m
  /dev/sdd1
```

3. Crear un volume group (VG) de nombre vg_cda que gestione los PVs anteriores

```
root@datos:~# root@datos:~# vgcreate vg_cda /dev/sdc1 /dev/sdd1
Volume group "vg_cda" successfully created
```

Puede comprobarse la lista de VGs disponibles con vgdisplay o también con vgs.

```
root@datos:~# vgdisplay
  --- Volume group --
 VG Name
                        vg_cda
 System ID
 Format
                        lvm2
 Metadata Areas
                        2
 Metadata Sequence No 1
                       read/write
 VG Access
                       resizable
 VG Status
 MAX LV
                       0
 Cur LV
                       0
 Open LV
                       0
 Max PV
                       0
 Cur PV
                       2
 Act PV
                        2
                       192,00 MiB
 VG Size
 PE Size
                       4,00 MiB
 Total PE
                       48
 Alloc PE / Size
                       0/0
 Free PE / Size
                       48 / 192,00 MiB
 VG UUID
                       YSd6fk-OcmK-FbvH-Bm7S-DXLt-Dkei-ZTPdoQ
```

```
root@datos:~# vgs
VG #PV #LV #SN Attr VSize VFree
vg_cda 2 0 0 wz--n- 192,00m 192,00m
```

Se han creado 48 PEs (*Physical Extensions*) de 4 MB cada uno (24 en cada uno de los dos PVs), incialmente todos disponibles.

Puede verse el estado de los PEs con pvdisplay -m (ahora están vinculados al VG vg_cda con sus 24 PEs libres [del 0 al 23])

```
root@datos:~# pvdisplay -m
 --- Physical volume ---
 PV Name
                       /dev/sdc1
 VG Name
                       vg_cda
 PV Size
                       99,00 MiB / not usable 3,00 MiB
 Allocatable
                       yes
 PE Size
                       4,00 MiB
 Total PE
                       24
 Free PE
                       24
 Allocated PE
                       0
 PV UUID
                       nda2kl-Rv1z-iCWo-zHGF-mOwG-Z1cO-3FNmqd
 --- Physical Segments ---
 Physical extent 0 to 23:
   FREE
 --- Physical volume ---
 PV Name
                       /dev/sdd1
 VG Name
                       vg_cda
 PV Size
                       99,00 MiB / not usable 3,00 MiB
 Allocatable
                       yes
 PE Size
                       4,00 MiB
 Total PE
                       24
 Free PE
                       24
 Allocated PE
                       0
 PV UUID
                       u6c07y-wSjs-OBY0-ID0Z-7CNC-QLPQ-8Lo31B
 --- Physical Segments ---
 Physical extent 0 to 23:
   FREE
```

4. Crear un logical volume (LV) de nombre volumen1 "dentro" del VG vg_cda y de tamaño 150 MB.

```
root@datos:~# lvcreate --size 150M --name volumen1 --verbose vg_cda
   DEGRADED MODE. Incomplete RAID LVs will be processed.
   Setting logging type to disk
   Finding volume group "vg_cda"
 Rounding up size to full physical extent 152,00 MiB
   Archiving volume group "vg_cda" metadata (seqno 1).
   Creating logical volume volumen1
   Creating volume group backup "/etc/lvm/backup/vg_cda" (seqno 2).
   Activating logical volume "volumen1".
   activation/volume_list configuration setting not defined: Checking only host tags for vg_cda/volumen1
   Creating vg_cda-volumen1
   Loading vg_cda-volumen1 table (253:0)
   Resuming vg_cda-volumen1 (253:0)
   Wiping known signatures on logical volume "vg_cda/volumen1"
   Initializing 4,00 KiB of logical volume "vg_cda/volumen1" with value 0.
   Creating volume group backup "/etc/lvm/backup/vg_cda" (seqno 2).
 Logical volume "volumen1" created
```

Puede comprobarse la lista de LVs disponibles con lvdisplay (añadiendo la opción -m se detalla la información de los *physical extents (PE)* y *logical extends (LE)* que los componen)

```
root@datos:~# lvdisplay -m
  --- Logical volume ---
 LV Path
                         /dev/vg_cda/volumen1
 LV Name
                         volumen1
 VG Name
                         vg_cda
 LV UUID
                         QhdcI2-Cc2H-rplK-4IEz-EiRX-Vuu7-SwU8cc
                        read/write
 LV Write Access
 LV Creation host, time datos.cda.net, 2018-09-19 09:40:01 +0200
 LV Status
                         available
 # open
                         0
 LV Size
                         152,00 MiB
 Current LE
                         38
                         2
 Segments
                         inherit
 Allocation
 Read ahead sectors
                         auto
 - currently set to
                         256
 Block device
                         253:0
  --- Segments ---
 Logical extents 0 to 23:
    Type linear
    Physical volume /dev/sdc1
    Physical extents 0 to 23
 Logical extents 24 to 37:
    Type linear
    Physical volume /dev/sdd1
    Physical extents 0 to 13
root@datos:~# pvdisplay -m
  --- Physical volume ---
 PV Name
                        /dev/sdc1
 VG Name
                        vg_cda
 PV Size
                        99,00 MiB / not usable 3,00 MiB
 Allocatable
                        yes (but full)
 PE Size
                        4,00 MiB
 Total PE
                        24
 Free PE
                        0
 Allocated PE
                        24
 PV UUID
                        nda2kl-Rv1z-iCWo-zHGF-mOwG-Z1cO-3FNmqd
 --- Physical Segments ---
 Physical extent 0 to 23:
    Logical volume /dev/vg_cda/volumen1
    Logical extents 0 to 23
 --- Physical volume ---
 PV Name
                        /dev/sdd1
 VG Name
                        vg_cda
                        99,00 MiB / not usable 3,00 MiB
 PV Size
 Allocatable
                        yes
 PE Size
                        4,00 MiB
 Total PE
                        24
 Free PE
                        10
 Allocated PE
                        14
 PV UUID
                        u6c07y-wSjs-OBY0-ID0Z-7CNC-QLPQ-8Lo31B
  --- Physical Segments ---
 Physical extent 0 to 13:
    Logical volume /dev/vg_cda/volumen1
    Logical extents 24 to 37
 Physical extent 14 to 23:
    FREE
```

Se ha creado un dispositivo de 152 MB (38 LEs (*Logical Extends* de 4 MB [del 0 al 37])), usando una asignación "lineal" (opción por defecto) que ocupa los 24 PEs del PV /dev/sdc1 y los 14 primeros PEs del PV /dev/sdd1.

El nuevo volumen creado (volumen1) aparecerá como dispositivo de bloques en /dev/vg_cda/volumen1 (también con el alias /dev/mapper/vg_cda-volumen1)

```
root@datos:~# ls -l /dev/vg_cda/volumen1
lrwxrwxrwx 1 root root 7 Sep 17 09:43 /dev/vg_cda/volumen1 -> ../dm-0
root@datos:~# ls -l /dev/dm-0
brw-rw---- 1 root disk 253, 0 Sep 17 09:55 /dev/dm-0
root@datos:~# lsblk
root@datos:~# parted -l
```

Nota:

 Puede consultar la Base de Datos del servicio udev (gestiona el hotplug de dispositivos) con el comando udevadm info para extraer todos los "alias" del dispositivo en /dev/

```
root@datos:~# udevadm info -q symlink /dev/vg_cda/volumen1
ó
root@datos:~# udevadm info -q all /dev/vg_cda/volumen1
```

5. Formatear ese dispositivo de bloques como ext3 y montarlo en /mnt/uno

```
root@datos:~# mkfs.ext3 -F /dev/vg_cda/volumen1
root@datos:~# mkdir /mnt/uno
root@datos:~# mount -t ext3 /dev/vg_cda/volumen1 /mnt/uno
root@datos:~# df -hT
S.ficheros Tipo Tamaño Usados Disp Uso% Montado en
/dev/sda1 ext4 9,8G 1,9G 7,4G 21% /
...
/dev/mapper/vg_cda-volumen1 ext3 144M 1,6M 135M 2% /mnt/uno
```

Nota: La opción -F de mkfs.ext3 no es estrictamente necesaria.

- Fuerza la creación de las estructuras de datos del sistema de ficheros *ext3* en el dispositivo de bloques formateado, sobreescribiendo los posibles datos preexistentes.
- Se usa a lo largo de este ejemplo ya que es posible que pruebas previas hubieran dejado datos en el dispositivo de bloques. En un escenario normal no sería necesario emplearla.

3.3. EJEMPLO 2: Ampliación de volumen lógico con /dev/sde

1. Crear una partición primaria de tipo *Linux LVM* (identificador 8e) en /dev/sde (usando fdisk, cfdisk, parted ó gparted), asignándole todo el espacio disponible (previamente se debe crear la tabla de particiones).

Haciéndolo desde línea de comando con el modo interactivo de parted sería:

```
root@datos:~# parted /dev/sde
(parted) mklabel msdos
(parted) mkpart primary 1M 100%
(parted) set 1 lvm on
(parted) quit
```

Se pueden comprobar los nuevos dispositivos disponibles con lsblk o parted -1

```
root@datos:~# lsblk
root@datos:~# parted -l
....
Model: ATA VBOX HARDDISK (scsi)
```

```
Disk /dev/sde: 210MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start End Size Type File system Flags
1 1049kB 210MB 209MB primary lvm
...
```

2. Registrar la nueva partición como volumen físico (PV) de LVM.

```
root@datos:~# pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created
```

3. Añadir el nuevo PV al volume group (VG) vg_cda con el comando vgextend

```
root@datos:~# vgextend vg_cda /dev/sde1
Volume group "vg_cda" successfully extended
```

Puede comprobarse la nueva información de los VGs disponibles con vgdisplay y el estado de los PVs con pvdisplay -m

```
root@datos:~# vgdisplay
  --- Volume group ---
 VG Name
                        vg_cda
 System ID
                       lvm2
 Format
 Metadata Areas
                        3
 Metadata Sequence No 3
 VG Access
                       read/write
 VG Status
                       resizable
 MAX LV
                       0
 Cur LV
                       1
 Open LV
                       1
 Max PV
                       0
 Cur PV
                       3
 Act PV
                       3
                       388,00 MiB
 VG Size
 PE Size
                       4,00 MiB
 Total PE
                       97
 Alloc PE / Size
                       38 / 152,00 MiB
 Free PE / Size
                       59 / 236,00 MiB
 VG UUID
                       YSd6fk-OcmK-FbvH-Bm7S-DXLt-Dkei-ZTPdoQ
root@datos:~# pvdisplay -m
  --- Physical volume ---
 PV Name
                       /dev/sdc1
 VG Name
                       vg_cda
 PV Size
                       99,00 MiB / not usable 3,00 MiB
 Allocatable
                       yes (but full)
 PE Size
                       4,00 MiB
 Total PE
                        24
 Free PE
                        0
 Allocated PE
                       24
 PV UUID
                       nda2kl-Rv1z-iCWo-zHGF-mOwG-Z1cO-3FNmqd
  --- Physical Segments ---
 Physical extent 0 to 23:
   Logical volume /dev/vg_cda/volumen1
   Logical extents 0 to 23
 --- Physical volume ---
```

```
PV Name
                      /dev/sdd1
VG Name
                      vg_cda
PV Size
                      99,00 MiB / not usable 3,00 MiB
Allocatable
                      yes
                      4,00 MiB
PE Size
Total PE
                      24
Free PE
                      10
Allocated PE
                      14
PV UUID
                      u6c07y-wSjs-OBY0-ID0Z-7CNC-QLPQ-8Lo31B
--- Physical Segments ---
Physical extent 0 to 13:
  Logical volume /dev/vg_cda/volumen1
  Logical extents 24 to 37
Physical extent 14 to 23:
  FREE
--- Physical volume ---
PV Name
                      /dev/sde1
VG Name
                      vg_cda
PV Size
                      199,00 MiB / not usable 3,00 MiB
Allocatable
                     yes
PE Size
                      4,00 MiB
Total PE
                      49
Free PE
                      49
Allocated PE
                      0
PV UUID
                      qusPN8-AaOz-hK8q-9j16-lXEe-RDiA-D3kFiE
--- Physical Segments ---
Physical extent 0 to 48:
  FREE
```

4. Crear un *logical volume (LV)* de nombre volumen2 y tamaño 20 MB ubicado "dentro" del VG vg_cda (al ampliar volumen1 permitirá evidenciar la flexibilidad de LVM para gestionar el espacio disponible)

root@datos:~# lvcreate --size 20M --name volumen2 --verbose vg_cda

Puede comprobarse la lista de LVs disponibles con lvdisplay (añadiendo la opción -m se detalla la información de los *physical extents (PE)* y *logical extends (LE)* que los componen)

```
root@datos:~# lvdisplay -m
  --- Logical volume ---
 LV Path
                         /dev/vg_cda/volumen1
 LV Name
                         volumen1
 VG Name
                         vg_cda
 LV UUID
                         QhdcI2-Cc2H-rplK-4IEz-EiRX-Vuu7-SwU8cc
 LV Write Access
                       read/write
 LV Creation host, time datos.cda.net, 2018-09-19 09:40:01 +0200
 LV Status
                        available
 # open
                         1
 LV Size
                         152,00 MiB
 Current LE
                         38
 Segments
                         2
 Allocation
                         inherit
 Read ahead sectors
                         auto
  - currently set to
                         256
 Block device
                         253:0
  --- Segments ---
 Logical extents 0 to 23:
    Type linear
    Physical volume /dev/sdc1
    Physical extents 0 to 23
```

```
Logical extents 24 to 37:
  Type linear
  Physical volume /dev/sdd1
  Physical extents 0 to 13
--- Logical volume ---
                      /dev/vg_cda/volumen2
LV Path
LV Name
                      volumen2
VG Name
                      vg_cda
LV UUID
                      cthIWU-s4QN-InlF-SAf9-trYn-QqW0-2UAD1B
LV Write Access read/write
LV Creation host, time datos.cda.net, 2018-09-19 10:03:07 +0200
LV Status
                      available
# open
                      0
LV Size
                      20,00 MiB
Current LE
                      5
Segments
                      1
Allocation
                      inherit
Read ahead sectors
                      auto
- currently set to
                      256
Block device
                      253:1
--- Segments ---
Logical extents 0 to 4:
  Type linear
  Physical volume /dev/sdd1
  Physical extents 14 to 18
```

El nuevo volumen ocupa 5 LE de 4MB, almacenados en los PEs 14 a 18 del PV /dev/sdd1. El volumen creado (volumen2) aparecerá como dispositivo de bloques en /dev/vg_cda/volumen2

```
root@datos:~# ls -l /dev/vg_cda/volumen2
lrwxrwxrwx 1 root root 7 Sep 17 10:03 /dev/vg_cda/volumen2 -> ../dm-1
root@datos:~# ls -l /dev/dm-1
brw-rw---- 1 root disk 253, 1 Sep 17 10:03 /dev/dm-1
```

- 5. Expandir el LV volumen1, aumentando su tamaño con todo el espacio disponible en el grupo de volúmenes vg_cda.
 - a) Desmontar el sistema de ficheros

root@datos:~# umount /mnt/uno/

b) Aumentar el tamaño del LV volumen1 usando el comando lvextend (o también lvresize)

```
root@datos:~# lvextend --extents +100%FREE /dev/vg_cda/volumen1
Size of logical volume vg_cda/volumen1 changed from 152,00 MiB (38 extents) to 368,00 MiB (92 extents).
Logical volume volumen1 successfully resized
```

Ver la situación actual de los PEs y LEs con lvdisplay -m y/o pvdisplay -m

root@datos:~# lvdisplay -m						
Logical volume						
LV Path	/dev/vg_cda/volumen1					
LV Name	volumen1					
VG Name	vg_cda					
LV UUID	QhdcI2-Cc2H-rplK-4IEz-EiRX-Vuu7-SwU8cc					
LV Write Access	read/write					
LV Creation host, time	datos.cda.net, 2018-09-19 09:40:01 +0200					
LV Status	available					
# open	0					
LV Size	368,00 MiB					

Current LE 92 Segments 4 Allocation inherit Read ahead sectors auto - currently set to 256 Block device 253:0 --- Segments ---Logical extents 0 to 23: Type linear Physical volume /dev/sdc1 Physical extents 0 to 23 Logical extents 24 to 37: Type linear Physical volume /dev/sdd1 Physical extents 0 to 13 Logical extents 38 to 86: Type linear Physical volume /dev/sde1 Physical extents 0 to 48 Logical extents 87 to 91: Type linear Physical volume /dev/sdd1 Physical extents 19 to 23 --- Logical volume ---/dev/vg_cda/volumen2 LV Path LV Name volumen2 VG Name vg_cda LV UUID cthIWU-s4QN-InlF-SAf9-trYn-QqW0-2UAD1B LV Write Access read/write LV Creation host, time datos.cda.net, 2018-09-19 10:03:07 +0200 LV Status available # open 0 LV Size 20,00 MiB Current LE 5 Segments 1 Allocation inherit Read ahead sectors auto - currently set to 256 Block device 253:1 --- Segments ---Logical extents 0 to 4: Type linear Physical volume /dev/sdd1 Physical extents 14 to 18

En la situación actual tenemos la siguiente distribución

- El LV volumen1 con 368 MB (92 LEs), ocupando los PEs 0-23 de /dev/sdc1, los PEs 0-13 y 19-23 de /dev/sdd1 y los PEs 0-48 de /dev/sde1
- El LV volumen2 con 20 MB (5 LEs), ocupando los PEs 14-18 de /dev/sdd1
- c) Expandir el sistema de ficheros con resize2fs (esta herramienta requiere comprobar previamente el sistema de ficheros con e2fsck -f)

```
root@datos:~# e2fsck -f /dev/vg_cda/volumen1
e2fsck 1.42.12 (29-Aug-2014)
Paso 1: Verificando nodos-i, bloques y tamaños
Paso 2: Verificando la estructura del directorio
```

```
Paso 3: Revisando la conectividad de directorios
Paso 4: Revisando las cuentas de referencia
Paso 5: Revisando el resumen de información de grupos
/dev/vg_cda/volumen1: 11/38912 ficheros (0.0% no contiguos), 10578/155648 bloques
root@datos:~# resize2fs /dev/vg_cda/volumen1
resize2fs 1.42.12 (29-Aug-2014)
Cambiando el tamaño del sistema de ficheros en /dev/vg_cda/volumen1 a 376832 (1k) bloques.
The filesystem on /dev/vg_cda/volumen1 is now 376832 (1k) blocks long.
```

6. Montar de nuevo el sistema de ficheros en /mnt/uno y comprobar el tamaño

```
root@datos:~# mount -t ext3 /dev/... /mnt/uno
root@datos:~# df -hT
S.ficheros Tipo Tamaño Usados Disp Uso% Montado en
/dev/sda1 ext4 9,8G 1,9G 7,4G 21% /
...
/dev/mapper/vg_cda-volumen1 ext3 353M 2,1M 332M 1% /mnt/uno
```

4. RAID por software

4.1. PREVIO: Fundamentos RAID software en GNU/Linux

4.1.1. Funcionamiento

Más detalles: Linux kernel RAID Wiki

4.1.2. Resumen de comandos relevantes

Instalación (ya hecho en la MV del ejercicio)

root@datos:~# apt-get install mdadm

Definición/gestión de los arrays de discos (detalles en Guía mdadm)

- mdadm --create <nombre dispositivo RAID>--level=<nivel RAID>--raid-devices=<num. dispositivos><dispos crea un array del nivel indicado utilizando los dispositivos que se enumeran
- mdadm --asemble <nombre dispositivo RAID><opciones>: recupera y activa un array previamente definido
- mdadm --manage <nombre dispositivo RAID><opciones>: operaciones de gestión de un array determinado
- mdadm --remove <nombre dispositivo RAID><dispositivo/particion> (alias a mdadm --manage --remove): elimina un dispositivo del array (por fallo, etc)
- mdadm --stop <nombre dispositivo RAID> (alias a mdadm --misc --stop): detiene un array no montado
- mdadm --run <nombre dispositivo RAID> (alias a mdadm --misc --start): inicia un array previamente detenido
- mdadm --grow <nombre dispositivo RAID><opciones> : modifica el tamaño de un array
- mdadm --zero-superblock <dispositivo/particion> : "limpia" el superbloque creado en un dispositivo que fue parte de un array

Información del estado de los arrays de discos

mdadm --detail --scan

cat /proc/mdstat

Configuración/definición permanente de los arrays a hacer disponibles en el arranque del sistema

- /etc/mdadm.conf (mismo formato que mdadm --detail --scan)
- Ver RAID setup (sección Saving your RAID configuration)

Resumen de comandos usuales: Mdadm Cheat Sheet

4.2. EJEMPLO 1: Niveles RAID (RAID0, RAID1, RAID5)

1. PREVIO: Desmontar los sistemas de ficheros y deshabilitar los volúmenes (físicos y lógicos) y el grupo de volúmenes de los ejemplos anteriores.

root@datos:~# umount /mnt/uno
root@datos:~# lvremove /dev/vg_cda/volumen1
root@datos:~# lvremove /dev/vg_cda/volumen2
root@datos:~# vgremove vg_cda
root@datos:~# pvremove /dev/sdc1 /dev/sdd1 /dev/sde1

2. Crear una partición primaria de tipo *Linux raid autodetect* (identificador fd) en /dev/sdc, /dev/sdd y /dev/sde (empleando fdisk, cfdisk, parted ó gparted), asignándole todo el espacio disponible (previamente se debe crear la tabla de particiones).

Haciéndolo desde línea de comandos con el modo interactivo de parted sería:

```
root@datos:~# parted /dev/sdc
                                      root@datos:~# parted /dev/sdd
                                                                           root@datos:~# parted /dev/sde
 (parted) mklabel msdos
                                      (parted) mklabel msdos
                                                                           (parted) mklabel msdos
 (parted) mkpart primary 1M 100%
                                      (parted) mkpart primary 1M 100%
                                                                           (parted) mkpart primary 1M 100%
 (parted) set 1 raid on
                                      (parted) set 1 raid on
                                                                           (parted) set 1 raid on
                                      (parted) quit
 (parted) quit
                                                                           (parted) quit
Nota: Pedirá confirmación para borrar la tabla de particiones anterior (Si se han hecho los ejercicios anteriores,
```

realmente bastaría con el comando set 1 raid on).

Se pueden comprobar los nuevos dispositivos disponibles con lsblk ó parted -1

```
root@datos:~# lsblk
root@datos:~# parted -1
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 105MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start
                End
                       Size
                              Type
                                       File system Flags
       1049kB
               105MB 104MB primary
                                                   raid
1
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdd: 105MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
Number Start
                End
                       Size
                              Type
                                       File system Flags
       1049kB 105MB 104MB primary
                                                   raid
1
```

Model: ATA VBOX HARDDISK (scsi)

Disk /dev/sde: 210MB Sector size (logical/physical): 512B/512B Partition Table: msdos Disk Flags: Number Start End Size Type File system Flags 1 1049kB 210MB 209MB primary raid

3. Crear un RAID0 (modo stripe), con nombre /dev/md_RAID0, haciendo uso de las tres particiones

root@datos:~# root@datos:~# mdadm --create --verbose /dev/md/md_RAIDO --level=stripe --raid-devices=3 /dev/sdc1 mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_RAIDO started.

Verificar el estado de /proc/mdstat (opcionalmente con mdadm --detail o parted -1)

unused devices: <none>

root@datos:~# ls -l /dev/md/md_RAID0
lrwxrwxrwx 1 root root 8 Sep 17 13:32 /dev/md/md_RAID0 -> ../md127

root@datos:~# ls -l /dev/md127 brw-rw---- 1 root disk 9, 127 Sep 17 13:32 /dev/md127

root@datos:~# parted -1
...

Model: Linux Software RAID Array (md) Disk /dev/md127: 413MB Sector size (logical/physical): 512B/512B Partition Table: loop Disk Flags:

Number Start End Size File system Flags 1 0,00B 413MB 413MB ext3

root@datos:~# mdadm --detail /dev/md/md_RAID0

Nota:

ó

 Puede consultar la Base de Datos del servicio udev (gestiona el hotplug de dispositivos) con el comando udevadm info para extraer todos los "alias" del dispositivo en /dev/

root@datos:~# udevadm info -q symlink /dev/md/md_RAID0

root@datos:~# udevadm info -q all /dev/md/md_RAID0

Formatearlo como ext3, montarlo y comprobar el espacio disponible.

root@datos:~# mkfs.ext3 -F /dev/md/md_RAID0
root@datos:~# mount -t ext3 /dev/md/md_RAID0 /mnt/uno
root@datos:~# df -Th /mnt/uno/
S.ficheros Tipo Tamaño Usados Disp Uso% Montado en
...
/dev/md127 ext3 374M 2,3M 352M 1% /mnt/uno

Desmontarlo y eliminar el array para las siguientes pruebas.

```
root@datos:~# umount /mnt/uno/
root@datos:~# mdadm --stop /dev/md/md_RAIDO
mdadm: stopped /dev/md/md_RAIDO
```

Limpiar los "superbloques RAID" en los dispositivos individuales

root@datos:~# mdadm --zero-superblock /dev/sdc1 root@datos:~# mdadm --zero-superblock /dev/sdd1 root@datos:~# mdadm --zero-superblock /dev/sde1

4. Crear un RAID1 (modo mirror), con nombre md_RAID1, haciendo uso de las tres particiones

```
root@datos:~# mdadm --create --verbose /dev/md/md_RAID1 --level=mirror --raid-devices=3 /dev/sdc1 /dev/sdd1 /dev
mdadm: /dev/sdc1 appears to be part of a raid array:
      level=linear devices=3 ctime= Sat Sep 17 13:41:55 2018
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device. If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
mdadm: /dev/sdd1 appears to be part of a raid array:
     level=linear devices=3 ctime= Sat Sep 17 13:41:55 2018
mdadm: /dev/sde1 appears to be part of a raid array:
     level=linear devices=3 ctime= Sat Sep 17 13:41:55 2018
mdadm: size set to 101248K
mdadm: largest drive (/dev/sde1) exceeds size (101248K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/md_RAID1 started.
Verificar el estado de /proc/mdstat (opcionalmente con mdadm --detail o parted -1)
root@datos:~# cat /proc/mdstat
Personalities : [linear] [raid0] [raid1]
md127 : active raid1 sde1[2] sdd1[1] sdc1[0]
      101248 blocks super 1.2 [3/3] [UUU]
unused devices: <none>
root@datos:~# ls -l /dev/md/md_RAID1
lrwxrwxrwx 1 root root 8 Sep 17 13:44 /dev/md/md_RAID1 -> ../md127
root@datos:~# ls -l /dev/md127
brw-rw---- 1 root disk 9, 127 Sep 17 13:44 /dev/md127
root@datos:~# parted -1
. . .
Model: Linux Software RAID Array (md)
Disk /dev/md127: 104MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
```

Number Start End Size File system Flags 1 0,00B 104MB 104MB ext3

root@datos:~# mdadm --detail /dev/md/md_RAID1

Formatearlo como ext3, montarlo y comprobar el espacio disponible.

```
root@datos:~# mkfs.ext3 -F /dev/md/md_RAID1
mke2fs 1.42.12 (29-Aug-2014)
/dev/md/md_RAID1 contiene un sistema de ficheros ext3
última fecha de montaje Sat Sep 17 13:25:50 2018
¿Continuar de todas formas? (s,n) s
root@datos:~# mount -t ext3 /dev/md/md_RAID1 /mnt/uno
root@datos:~# df -Th
                        Tamaño Usados Disp Uso% Montado en
S.ficheros
               Tipo
. . .
/dev/md127
               ext3
                           92M
                                 1,6M
                                        86M
                                               2% /mnt/uno
Desmontarlo y eliminar el array para las siguientes pruebas.
```

```
root@datos:~# root@datos:~# umount /mnt/uno/
root@datos:~# mdadm --stop /dev/md/md_RAID1
mdadm: stopped /dev/md/md_RAID1
```

Limpiar los "superbloques RAID" en los dispositivos individuales

root@datos:~# mdadm --zero-superblock /dev/sdc1 root@datos:~# mdadm --zero-superblock /dev/sdd1 root@datos:~# mdadm --zero-superblock /dev/sde1

5. Crear un RAID5, con nombre md_RAID5, haciendo uso de las tres particiones (verificar el estado de /proc/mdstat)

```
root@datos:~# mdadm --create --verbose /dev/md/md_RAID5 --level=raid5 --raid-devices=3 /dev/sdc1 /dev/sdd1 /dev/
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: /dev/sdc1 appears to be part of a raid array:
      level=raid0 devices=3 ctime= Sat Sep 17 13:52:43 2018
mdadm: /dev/sdd1 appears to be part of a raid array:
      level=raid0 devices=3 ctime= Sat Sep 17 13:52:43 2018
mdadm: /dev/sde1 appears to be part of a raid array:
      level=raid0 devices=3 ctime= Sat Sep 17 13:52:43 2018
mdadm: size set to 100352K
mdadm: largest drive (/dev/sde1) exceeds size (100352K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/md_RAID5 started.
Verificar el estado de /proc/mdstat (opcionalmente con mdadm --detail o parted -1)
root@datos:~# cat /proc/mdstat
Personalities : [linear] [raid0] [raid1] [raid6] [raid5] [raid4]
md127 : active raid5 sde1[3] sdd1[1] sdc1[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
root@datos:~# ls -l /dev/md/md_RAID5
lrwxrwxrwx 1 root root 8 Sep 17 13:54 /dev/md/md_RAID5 -> ../md127
root@datos:~# ls -l /dev/md127
brw-rw---- 1 root disk 9, 127 Sep 17 13:54 /dev/md127
root@datos:~# parted -1
Model: Linux Software RAID Array (md)
```

```
Disk /dev/md127: 206MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
Number Start End
                     Size
                            File system Flags
       0,00B 206MB 206MB ext3
1
root@datos:~# mdadm --detail /dev/md/md_RAID5
Formatearlo como ext3, montarlo y comprobar el espacio disponible.
root@datos:~# mkfs.ext3 -F /dev/md/md_RAID5
mke2fs 1.42.12 (29-Aug-2014)
/dev/md/md_RAID5 contiene un sistema de ficheros ext3
última fecha de montaje Sat Sep 17 13:38:30 2018
¿Continuar de todas formas? (s,n) s
root@datos:~# mount /dev/md/md_RAID5 /mnt/uno/
root@datos:~# df -Th
                        Tamaño Usados Disp Uso% Montado en
S.ficheros
             Tipo
. . .
/dev/md127
                          186M
                                1,6M 175M
                                             1% /mnt/uno
               ext3
Desmontarlo y eliminar el array para las siguientes pruebas.
root@datos:~# root@datos:~# umount /mnt/uno/
root@datos:~# mdadm --stop /dev/md/md_RAID5
mdadm: stopped /dev/md/md_RAID5
```

Limpiar los "superbloques RAID" en los dispositivos individuales

root@datos:~# mdadm --zero-superblock /dev/sdc1 root@datos:~# mdadm --zero-superblock /dev/sdd1 root@datos:~# mdadm --zero-superblock /dev/sde1

4.3. EJEMPLO 2: Niveles RAID anidados (RAID10, RAID01)

- 1. IMPORTANTE: Asegurar que están desmontados los sistemas de ficheros y eliminados los arrays de discos del ejemplo anterior.
- Crear una partición primaria de tipo Linux raid autodetect (identificador fd) en /dev/sdc y /dev/sdd asignándole todo el espacio disponible (previamente se debe crear la tabla de particiones). [ya está hecho en puntos anteriores]
- 3. Crear dos particiones primarias de tipo *Linux raid autodetect* (identificador fd) y 100 MB de tamaño cada una en /dev/sde (empleando fdisk, cfdisk, parted ó gparted). Previamente se debe crear una tabla de particiones vacía. [es necesario hacerlo (ahora son dos particiones)]

Haciéndolo desde línea de comando con el modo interactivo de parted sería:

root@datos:~# parted /dev/sde
(parted) mklabel msdos
(parted) mkpart primary 1M 50%
(parted) set 1 raid on
(parted) mkpart primary 50% 100%
(parted) set 2 raid on
(parted) quit

Se pueden comprobar los dispositivos disponibles con lsblk o parted -1

4. Crear un RAID10, con nombre md_RAID10, haciendo uso de las cuatro particiones.

Nota: El comando mdadm, con la opción --level=raid10, permite crear y parametrizar un esquema anidado RAID10. En este ejemplo se emulará ese RAID10 "manualmente" como ejemplo de la versatilidad de la herramienta.

 a) Crear los dos RAID1 de primer nivel ("internos"), uno con /dev/sdc1 y /dev/sde1 y otro con /dev/sdd1 y /dev/sde2

root@datos:~# mdadm --create --verbose /dev/md/md_interno1 --level=1 --raid-devices=2 /dev/sdc1 /dev/sde1 mdadm: /dev/sdc1 appears to be part of a raid array: level=raid5 devices=3 ctime= Sat Sep 17 14:20:30 2018 mdadm: Note: this array has metadata at the start and may not be suitable as a boot device. If you plan to store '/boot' on this device please ensure that your boot-loader understands md/v1.x metadata, or use --metadata=0.90 mdadm: /dev/sde1 appears to be part of a raid array: level=raid5 devices=3 ctime= Sat Sep 17 14:20:30 2018 mdadm: size set to 101248K Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_interno1 started. root@datos:~# mdadm --create --verbose /dev/md/md_interno2 --level=1 --raid-devices=2 /dev/sdd1 /dev/sde2 mdadm: /dev/sdd1 appears to be part of a raid array: level=raid5 devices=3 ctime= Sat Sep 17 14:20:30 2018 mdadm: Note: this array has metadata at the start and may not be suitable as a boot device. If you plan to store '/boot' on this device please ensure that your boot-loader understands md/v1.x metadata, or use --metadata=0.90 mdadm: size set to 101248K mdadm: largest drive (/dev/sde2) exceeds size (101248K) by more than 1% Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_interno2 started. b) Crear el RAIDO de segundo nivel ("externo"), usando los dos arrays anteriores root@datos:~# mdadm --create --verbose /dev/md/md_RAID10 --level=0 --raid-devices=2 /dev/md/md_interno1 /dev mdadm: chunk size defaults to 512K mdadm: /dev/md/md_interno1 appears to contain an ext2fs file system size=101248K mtime= Sat Sep 17 13:48:00 2018 mdadm: /dev/md/md_interno2 appears to contain an ext2fs file system size=101248K mtime= Sat Sep 17 13:48:00 2018 Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_raid10 started. Verificar el estado de /proc/mdstat root@datos:~# cat /proc/mdstat Personalities : [linear] [raid0] [raid1] [raid6] [raid5] [raid4] md125 : active raid0 md126[1] md127[0] 199680 blocks super 1.2 512k chunks md126 : active raid1 sde2[1] sdd1[0] 101248 blocks super 1.2 [2/2] [UU] md127 : active raid1 sde1[1] sdc1[0] 101248 blocks super 1.2 [2/2] [UU]

```
unused devices: <none>
```

```
root@datos:~# parted -1
  Model: Linux Software RAID Array (md)
  Disk /dev/md125: 204MB
  Sector size (logical/physical): 512B/512B
  Partition Table: unknown
  Disk Flags:
  Model: Linux Software RAID Array (md)
  Disk /dev/md126: 104MB
  Sector size (logical/physical): 512B/512B
  Partition Table: loop
  Disk Flags:
  Number Start End
                               File system Flags
                        Size
  1
         0,00B 104MB 104MB ext3
  Model: Linux Software RAID Array (md)
  Disk /dev/md127: 104MB
  Sector size (logical/physical): 512B/512B
  Partition Table: loop
  Disk Flags:
  Number Start End
                        Size
                               File system Flags
         0,00B 104MB 104MB
   1
                               ext3
  root@datos:~# mdadm --detail /dev/md/md_interno1
   root@datos:~# mdadm --detail /dev/md/md_interno2
   root@datos:~# mdadm --detail /dev/md/md_RAID10
c) Formatearlo como ext3, montarlo y comprobar el espacio disponible.
  root@datos:~# mkfs.ext3 -F /dev/md/md_RAID10
  root@datos:~# mount /dev/md/md_RAID10 /mnt/uno/
  root@datos:~# df -Th
                           Tamaño Usados Disp Uso% Montado en
  S.ficheros
                 Tipo
   /dev/md125
                  ext3
                            185M
                                   1,6M 174M 1% /mnt/uno
d) Desmontarlo y eliminar los arrays para las siguientes pruebas.
  root@datos:~# umount /mnt/uno
  root@datos:~# mdadm --stop /dev/md/md_RAID10
  root@datos:~# mdadm --stop /dev/md/md_interno1
  root@datos:~# mdadm --stop /dev/md/md_interno2
  Limpiar los "superbloques RAID" en los dispositivos individuales
```

root@datos:~# mdadm --zero-superblock /dev/sdc1 root@datos:~# mdadm --zero-superblock /dev/sdd1 root@datos:~# mdadm --zero-superblock /dev/sde1 root@datos:~# mdadm --zero-superblock /dev/sde2

Cuestión 1: ¿Por qué en este caso es conveniente que las dos particiones de /dev/sde esten asignadas a "subarrays" RAID1 distintos?

5. Crear un RAID01, con nombre md_RAID01, haciendo uso de las cuatro particiones.

a) Crear los dos RAID0 de primer nivel ("internos"), uno con /dev/sdc1 y /dev/sdd1 el otro con /dev/sdc1 y /dev/sde2 root@datos:~# mdadm --create --verbose /dev/md/md_interno1 --level=0 --raid-devices=2 /dev/sdc1 /dev/sdc1 mdadm: chunk size defaults to 512K mdadm: /dev/sdc1 appears to be part of a raid array: level=raid1 devices=2 ctime= Sat Sep 17 14:27:25 2018 mdadm: /dev/sdd1 appears to be part of a raid array: level=raid1 devices=2 ctime= Sat Sep 17 14:27:36 2018 Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_interno1 started. root@datos:~# mdadm --create --verbose /dev/md/md_interno2 --level=0 --raid-devices=2 /dev/sde1 /dev/sde2 mdadm: chunk size defaults to 512K mdadm: /dev/sde1 appears to be part of a raid array: level=raid1 devices=2 ctime= Sat Sep 17 14:27:25 2018 mdadm: /dev/sde2 appears to be part of a raid array: level=raid1 devices=2 ctime= Sat Sep 17 14:27:36 2018 Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_interno2 started. b) Crear el RAID1 de segundo nivel ("externo"), usando los dos arrays anteriores (verificar el estado de /proc/mdstat) root@datos:~# mdadm --create --verbose /dev/md/md_RAID01 --level=1 --raid-devices=2 /dev/md/md_interno1 /dev mdadm: /dev/md/md_interno1 appears to contain an ext2fs file system size=200704K mtime= Sat Sep 17 13:57:49 2018 mdadm: Note: this array has metadata at the start and may not be suitable as a boot device. If you plan to store '/boot' on this device please ensure that your boot-loader understands md/v1.x metadata, or use --metadata=0.90 mdadm: /dev/md/md_interno2 appears to contain an ext2fs file system size=200704K mtime= Sat Sep 17 13:57:49 2018 mdadm: size set to 200512K Continue creating array? y mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md/md_RAID01 started. Verificar el estado de /proc/mdstat root@datos:~# cat /proc/mdstat Personalities : [linear] [raid0] [raid1] [raid6] [raid5] [raid4] md125 : active raid1 md126[1] md127[0] 200512 blocks super 1.2 [2/2] [UU] md126 : active raid0 sde2[1] sde1[0] 201728 blocks super 1.2 512k chunks md127 : active raid0 sdd1[1] sdc1[0] 200704 blocks super 1.2 512k chunks unused devices: <none> root@datos:~# parted -1 . . . Model: Linux Software RAID Array (md) Disk /dev/md125: 205MB Sector size (logical/physical): 512B/512B Partition Table: unknown Disk Flags:

```
Model: Linux Software RAID Array (md)
   Disk /dev/md126: 207MB
   Sector size (logical/physical): 512B/512B
   Partition Table: loop
   Disk Flags:
   Number Start End
                        Size
                              File system Flags
   1
         0,00B 207MB 207MB ext3
   Model: Linux Software RAID Array (md)
   Disk /dev/md127: 206MB
   Sector size (logical/physical): 512B/512B
   Partition Table: loop
   Disk Flags:
   Number Start End
                               File system Flags
                         Size
   1
         0,00B 206MB 206MB ext3
   root@datos:~# mdadm --detail /dev/md/md_interno1
   root@datos:~# mdadm --detail /dev/md/md_interno2
   root@datos:~# mdadm --detail /dev/md/md_RAID01
c) Formatearlo como ext3, montarlo y comprobar el espacio disponible.
   root@datos:~# mkfs.ext3 -F /dev/md/md_RAID01
   root@datos:~# mount -t ext3 /dev/md/md_RAID01 /mnt/uno/
   root@datos:~# df -Th
   S.ficheros
                 Tipo
                          Tamaño Usados Disp Uso% Montado en
   . . .
   /dev/md125
                  ext3
                             186M
                                   1,6M 175M
                                                1% /mnt/uno
d) Demontarlo y eliminar el array para las siguientes pruebas.
   root@datos:~# umount /mnt/uno
   root@datos:~# mdadm --stop /dev/md/md_RAID01
   root@datos:~# mdadm --stop /dev/md/md_interno1
   root@datos:~# mdadm --stop /dev/md/md_interno2
   Limpiar los "superbloques RAID" en los dispositivos individuales
   root@datos:~# mdadm --zero-superblock /dev/sdc1
   root@datos:~# mdadm --zero-superblock /dev/sdd1
   root@datos:~# mdadm --zero-superblock /dev/sde1
   root@datos:~# mdadm --zero-superblock /dev/sde2
```

Cuestión 2: ¿Por qué en este caso sí es conveniente que las dos particiones de /dev/sde estén asignadas al mismo "subarray" RAID0?

5. Tareas entregables

5.1. Gestión de un RAID5

- 1. Preparación
 - a) Limpiar los discos y particiones del último ejemplo

- b) Crear un RAID5 utilizando los 3 dispositivos /dev/sdc1, /dev/sdd1 y /dev/sde1 tal y como se hizo en la última parte de la sección 4.2
- c) Formatear ese array con un sistema de ficheros *ext3* (usando la opción -F en caso de que pruebas previas hayan dejado datos no deseados), montarlo, crear con dd un archivo de 1MB relleno con ceros y comprobar con df el tamaño del sistema de ficheros y el espacio libre.
- 2. Tarea a realizar
 - a) Desmontar el sistema de ficheros
 - b) Particionar el dispositivo /dev/sdf creando una partición /dev/sdf1 de tipo RAID
 - c) Añadir este nuevo dispositivo al array RAID5 y ajustar el tamaño del array
 - d) Aumentar el tamaño del sistema de ficheros ext3 para que ocupe la totalidad del nuevo array RAID5 (comando resize2fs)
 - e) Montar el sistema de ficheros, comprobar la presencia del fichero de 1MB creado en el paso anterior y verificar con **df** el tamaño del nuevo sistema de ficheros y el espacio libre.

5.2. Documentación a entregar

- Práctica individual
- Entrega en MOOVI hasta el domingo 17/10/2021
- Documentación del entregable exclusivamente en formato PDF.
 - Responder a las cuestiones 1 y 2 incluidas en los ejemplos de la sección 4.3 sobre modos RAID anidados
 - Detallar los pasos seguidos en el punto 2 de la tarea entregable propuesta en la sección 5.1, indicando los comandos utilizados (y detallando su función), las opciones empleadas y las salidas obtenidas.
 - Incluir las salidas del comando mdadm --detail <array>
 - (1) antes de la adición del nuevo disco,
 - (2) después de añadirlo al array
 - (3) después de ajustar su tamaño y el del sistema de ficheros ext
3.
 - Justificar el tamaño del array RAID5 inicial y su tamaño después de añadir el dispositivo /dev/sdf1
 - Indicar las fuentes de información consultadas (webs, foros, páginas man, etc) para realizar la tarea descrita en la sección 5.1.