HowTo ASIX RAID

Redundant Array of Inexpensive Disks

Curs 2018 -2019

Aprenentatges treballats

1. Tipus de RAID.
    a. RAID0, RAID1, RAID2, RAID3, RAID4, RAID5, RAID6 i RAID10.
    b. Raids a implementar: RAID1 i RAID5,
2. Creació i funcionament de raids.
    a. Creació.
    b. Examinar el funcionament: /proc/mdadm, examine, detail, scan.
    c. Unitats de spare.
    d. Creació de errades amb fail.
3. Creació / automatització.
    a. Creació del fitxer de configuració. Automàtic amb examine scan.
    b. Ensamblatge automatitzat amb scan.
    c. Metadades: Marques de partició. Examinar les marques amb hexdump. Eliminar les metadades amb --zero-superbloc.
4. Modificacions del format:
    a. Incrementar / decrementar el nuémro d'elements del array. Totals i spare.
    b. Incrementar / decrementar l'espai d'emmegatzament.
    c. Convertir el raid de un level a un altre
5. RAID + LVM
    a. Aplicar al raid un sistema de fitxers LVM

Documentació

Aquest document ha estat elaborant utilitzant com a eina de treball un sistema GNU/Linux Fedora 20.

- Apunts @edt ASIX-M11
- RAID
    - objectius-raid
    - HowTo-ASIX-RAID
    - Manual ordre mdadmin
    - deprecated: Software-raid-howto.pdf (autor mdadmin)

- Documentació de les pàgines man de les ordres.
- Fedora Documentation: Fedora 14, Storage Administration Guide, Chapter 12: RAID Redundant Array of Inexpensive Disks
- Fedora Documentation: Fedora 20, Installation Guide, Chapter 9.4.12: Create software RAID
- The software RAID-HowTO de Jakob Østergaard's.

```
# rpm -ql mdadm
/etc/cron.d/raid-check
/etc/libreport/events.d/mdadm_event.conf
/etc/sysconfig/raid-check
/usr/lib/systemd/system/mdmonitor.service
/usr/lib/udev/rules.d/64-md-raid.rules
/usr/lib/udev/rules.d/65-md-incremental.rules
/usr/sbin/mdadm
/usr/sbin/mdmon
/usr/sbin/raid-check
/usr/share/doc/mdadm-3.2.6
/usr/share/doc/mdadm-3.2.6/COPYING
/usr/share/doc/mdadm-3.2.6/ChangeLog
/usr/share/doc/mdadm-3.2.6/TODO
/usr/share/doc/mdadm-3.2.6/mdadm.conf-example
/usr/share/doc/mdadm-3.2.6/syslog-events
/usr/share/man/man4/md.4.gz
/usr/share/man/man5/mdadm.conf.5.gz
/usr/share/man/man8/mdadm.8.gz
/usr/share/man/man8/mdmon.8.gz
/usr/usr/lib/tmpfiles.d/mdadm.conf
/var/run/mdadm
```

Observant el man de ***mdadm*** es poden esbrinar ordres relacionades, els autors i enllaços a pàgines de documentació pròpies dels creadors dels MD.

```
SEE ALSO
       For further information on mdadm usage, MD and the various levels of RAID, see:

       http://raid.wiki.kernel.org/

       (based upon Jakob Østergaard's Software-RAID.HOWTO)

       The latest version of mdadm should always be available from

       http://www.kernel.org/pub/linux/utils/raid/mdadm/

       Related man pages:

       mdmon(8), mdadm.conf(5), md(4).

       raidtab(5), raid0run(8), raidstop(8), mkraid(8).
```

RAID Redundant Array of Inexpensive Disks

Descripció de LVM extreta de Fedora Documentation/14 Storage Guide:

> The basic idea behind RAID is to combine multiple small, inexpensive disk drives into an array to accomplish performance or redundancy goals not attainable with one large and expensive drive. This array of drives appears to the computer as a single logical storage unit or drive.
>
> RAID allows information to be spread across several disks. RAID uses techniques such as disk striping (RAID Level 0), disk mirroring (RAID Level 1), and disk striping with parity (RAID Level 5) to achieve redundancy, lower latency, increased bandwidth, and maximized ability to recover from hard disk crashes.
>
> RAID distributes data across each drive in the array by breaking it down into consistently-sized chunks (commonly 256K or 512k, although other values are acceptable). Each chunk is then written to a hard drive in the RAID array according to the RAID level employed. When the data is read, the process is reversed, giving the illusion that the multiple drives in the array are actually one large drive.

Firmware RAID:

Firmware RAID (also known as ATARAID) is a type of software RAID where the RAID sets can be configured using a firmware-based menu. The firmware used by this type of RAID also hooks into the BIOS, allowing you to boot from its RAID sets. Different vendors use different on-disk metadata formats to mark the RAID set members. The Intel Matrix RAID is a good example of a firmware RAID system.

Hardware RAID

The hardware-based array manages the RAID subsystem independently from the host. It presents a single disk per RAID array to the host.

A Hardware RAID device may be internal or external to the system, with internal devices commonly consisting of a specialized controller card that handles the RAID tasks tranparently to the operating system and with external devices commonly connecting to the system via SCSI, fiber channel, iSCSI, InfiniBand, or other high speed network interconnect and presenting logical volumes to the system.

RAID controller cards function like a SCSI controller to the operating system, and handle all the actual drive communications. The user plugs the drives into the RAID controller (just like a normal SCSI controller) and then adds them to the RAID controllers configuration, and the operating system won't know the difference.

Software RAID

Software RAID implements the various RAID levels in the kernel disk (block device) code. It offers the cheapest possible solution, as expensive disk controller cards or hot-swap chassis are not required.

Software RAID also works with cheaper IDE disks as well as SCSI disks. With today's faster CPUs, Software RAID also generally outperforms Hardware RAID.

The Linux kernel contains a multi-disk (MD) driver that allows the RAID solution to be completely hardware independent. The performance of a software-based array depends on the server CPU performance and load.

Tipus de RAID (en Fedora)

RAID supports various configurations, including levels 0, 1, 4, 5, 6, 10, and linear. These RAID types are defined as follows:

Level 0

RAID level 0, often called "striping," is a performance-oriented striped data mapping technique. This means the data being written to the array is broken down into strips and written across the member disks of the array, allowing high I/O performance at low inherent cost but provides no redundancy.

Many RAID level 0 implementations will only stripe the data across the member devices up to the size of the smallest device in the array. This means that if you have multiple devices with slightly different sizes, each device will get treated as though it is the same size as the smallest drive.

Therefore, the common storage capacity of a level 0 array is equal to the capacity of the smallest member disk in a Hardware RAID or the capacity of smallest member partition in a Software RAID multiplies by the number of disks or partitions in the array.

Level 1

RAID level 1, or "mirroring," has been used longer than any other form of RAID. Level 1 provides redundancy by writing identical data to each member disk of the array, leaving a "mirrored" copy on each disk. Mirroring remains popular due to its simplicity and high level of data availability.

Level 1 operates with two or more disks, and provides very good data reliability and improves performance for read-intensive applications but at a relatively high cost.

The storage capacity of the level 1 array is equal to the capacity of the smallest mirrored hard disk in a Hardware RAID or the smallest mirrored partition in a Software RAID. Level 1 redundancy is the highest possible among all RAID types, with the array being able to operate with only a single disk present.

Level 4

Level 4 uses parity concentrated on a single disk drive to protect data. Because the dedicated parity disk represents an inherent bottleneck on all write transactions to the RAID array, level is seldom used without accompanying technologies such as write-back caching, or in specific circumstances where the system administrator is intentionally designing the software RAID device with this bottleneck in mind (such as an array that will have little to no write transactions once the array is populated with data). RAID level 4 is so rarely used that it is not available as an option in Anaconda. However, it could be created manually by the user if truly needed.

The storage capacity of Hardware RAID level 4 is equal to the capacity of the smallest member partition multiplied by the number of partitions minus one. Performance of a RAID level 4 array will always be asymmetrical, meaning reads will outperform writes. This is because writes consume extra CPU and main memory bandwidth when generating parity, and then also consume extra bus bandwidth when writing the actual data to disks because you are writing not only the data, but also the parity. Reads need only read the data and not the parity unless the array is in a degraded state. As a result, reads generate less traffic to the drives and across the busses of the computer for the same amount of data transfer under normal operating conditions.

Level 5

This is the most common type of RAID. By distributing parity across all of an array's member disk drives, RAID level 5 eliminates the write bottleneck inherent in level 4. The only performance bottleneck is the parity calculation process itself. With modern CPUs and Software RAID, that is usually not a bottleneck at all since modern CPUs can generate parity very fast. However, if you have a sufficiently large number of member devices in a software RAID5 array such that the combined aggregate data transfer speed across all devices is high enough, then this bottleneck can start to come into play.

As with level 4, level 5 has asymmetrical performance, with reads substantially outperforming writes. The storage capacity of RAID level 5 is calculated the same way as with level 4.

Level 6

This is a common level of RAID when data redundancy and preservation, and not performance, are the paramount concerns, but where the space inefficiency of level 1 is not acceptable. Level 6 uses a complex parity scheme to be able to recover from the loss of any two drives in the array.

This complex parity scheme creates a significantly higher CPU burden on sofware RAID devices and also imposes an increased burden during write transactions. As such, not only is level 6 asymmetrical in performance like levels 4 and 5, but it is considerably more asymmetrical.

The total capacity of a RAID level 6 array is calculated similarly to RAID level 5 and 4, except that you must subtract 2 devices (instead of 1) from the device count for the extra parity storage space.

Level 10

This RAID level attempts to combine the performance advantages of level 0 with the redundancy of level 1. It also helps to alleviate some of the space wasted in level 1 arrays with more than 2 devices. With level 10, it is possible to create a 3-drive array configured to store only 2 copies of each piece of data, which then allows the overall array size to be 1.5 times the size of the smallest devices instead of only equal to the smallest device (like it would be with a 3-device, level 1 array).

The number of options available when creating level 10 arrays (as well as the complexity of selecting the right options for a specific use case) make it impractical to create during installation. It is possible to create one manually using the command line mdadm tool. For details on the options and their respective performance trade-offs, refer to man md.

Linear RAID

Linear RAID is a simple grouping of drives to create a larger virtual drive. In linear RAID, the chunks are allocated sequentially from one member drive, going to the next drive only when the first is completely filled. This grouping provides no performance benefit, as it is unlikely that any I/O operations will be split between member drives. Linear RAID also offers no redundancy and, in fact, decreases reliability — if any one member drive fails, the entire array cannot be used. The capacity is the total of all member disks.

La descripció que fa l'aplicació gràfica de fedora en el procés d'istal·lació , de cada tipus de raid permès és el següent:

RAID0 (Performance)
Distributes data across multiple storage devices. Level 0 RAIDs offer increased performance over standard partitions, and can be used to pool the storage of multiple devices into one large virtual device. Note that Level 0 RAIDS offer no redundancy and that the failure of one device in the array destroys the entire array. RAID 0 requires at least two RAID partitions.

RAID1 (Redundancy)
Mirrors the data on one storage device onto one or more other storage devices. Additional devices in the array provide increasing levels of redundancy. RAID 1 requires at least two RAID partitions.

RAID4 (Error Checking)
Distributes data across multiple storage devices, but uses one device in the array to store parity information that safeguards the array in case any device within the array fails. Because all parity information is stored on the one device, access to this device creates a bottleneck in the performance of the array. RAID 4 requires at least three RAID partitions.

RAID5 (Distributed Error Checking)
Distributes data and parity information across multiple storage devices. Level 5 RAIDs therefore offer the performance advantages of distributing data across multiple devices, but do not share the performance bottleneck of level 4 RAIDs because the parity

information is also distributed through the array. RAID 5 requires at least three RAID partitions.

RAID6 (Redundant Error Checking)
Level 6 RAIDs are similar to level 5 RAIDs, but instead of storing only one set of parity data, they store two sets. RAID 6 requires at least four RAID partitions.

RAID10 (Performance, Redundancy)
Level 10 RAIDs are *nested RAIDs* or *hybrid RAIDs*. Level 10 RAIDs are constructed by distributing data over mirrored sets of storage devices. For example, a level 10 RAID constructed from four RAID partitions consists of two pairs of partitions in which one partition mirrors the other. Data is then distributed across both pairs of storage devices, as in a level 0 RAID. RAID 10 requires at least four RAID partitions.

Pràctica (1): Treball bàsic amb raid

Crear el RAID

Crear tres unitats físiques ‘imaginaries’ usant la utilitat *dd* per generar espai de disc virtual. Assignar aquests fitxers a un dispositiu físic de *loopback*. És a dir, en lloc de crear tres particions de debò tipus /dev/sda2, /dev/sda3 i /dev/sda4 ens inventem les particions /dev/loop0, /dev/loop1 i /dev/loop2.

Exemple d’ordres RAID:

```
# mdadm --create /dev/md0 --chunk=4 --level=1 --raid-devices=3 /dev/loop0 /dev/loop1
/dev/loop2
```

```
# Crear les fitxers imatge
# dd if=/dev/zero of=disk01.img bs=1k count=100K
102400+0 registres llegits
102400+0 registres escrits
104857600 octets (105 MB) copiats, 0,676056 s, 155 MB/s
# dd if=/dev/zero of=disk02.img bs=1k count=100K
# dd if=/dev/zero of=disk03.img bs=1k count=100K

# Assignar-los al loopback
# losetup /dev/loop0 /opt/lvm/disk01.img
# losetup /dev/loop1 /opt/lvm/disk02.img
# losetup /dev/loop2 /opt/lvm/disk03.img

# losetup -a
/dev/loop0: [2053]:1217 (/opt/lvm/disk01.img)
/dev/loop1: [2053]:1218 (/opt/lvm/disk02.img)
/dev/loop2: [2053]:1220 (/opt/lvm/disk03.img)
```

Un cop disposem de les tres particions virtuals les integrem a un RAID format per totes tres.

```
# mdadm --create /dev/md0 --chunk=4 --level=1 --raid-devices=3 /dev/loop0
/dev/loop1 /dev/loop2
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
Continue creating array?
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

```
# tree /dev/disk
/dev/disk
|-- by-id
|   |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69 -> ../../sda
|   |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69-part1 -> ../../sda1
……
|   |-- ata-FUJITSU_MHV2100AT_PL_NSA3T6329W69-part7 -> ../../sda7
|   |-- ata-MATSHITADVD-RAM_UJ-841S -> ../../sr0
|   |-- md-name-portatil.localdomain:0 -> ../../md0
|   `-- md-uuid-b5fd01dc:53a820d3:190ae832:4f3144f8 -> ../../md0
```

Ara el sistema disposa d’un nou dispositiu anomenat **/dev/md0** que és un disc RAID format per les tres particions loop0, loop1 i loop2. Es tracta d’un raid de tipus 1 amb tres discs miralls. Però el sistema el veu com un sol disc de 100M.

Ara cal assignar-li un sistema de fitxers (formatar-lo) i muntar-lo al *filesystem* per poder-lo utilitzar. En l’exemple es munta a *mnt* i s’hi copien les dades del directori *boot*. Es pot observar amb el ***df*** l’espai total, lliure i ocupat del raid (sembla que massa ocupat i tot!).

```
# mkfs -t ext4 /dev/md0
mke2fs 1.42.3 (14-May-2012)
Discarding device blocks: fet
Etiqueta del sistema de fitxers=
OS type: Linux
Mida del bloc=1024 (log=0)
Mida del fragment=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
25584 inodes, 102272 blocks
5113 blocks (5.00%) reserved for the super user
Bloc de dades inicial=1
Màxim de blocs del sistema de fitxers=67371008
13 grups de blocs
8192 blocs per grup, 8192 fragments per grup
1968 nodes-i per grup
Còpies de seguretat del superbloc desades en els blocs:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: fet
Escriptura de les taules de nodes-i:fet
Creació del registre de transaccions (4096 blocs): fet
Escriptura de la informació dels súperblocs i de comptabilitat del sistema de fitxers:fet

# blkid
/dev/loop0: UUID="b5fd01dc-53a8-20d3-190a-e8324f3144f8"
UUID_SUB="b36d27f4-3024-029e-42df-5e2d0cd3517d" LABEL="portatil.localdomain:0"
TYPE="linux_raid_member"
/dev/loop1: UUID="b5fd01dc-53a8-20d3-190a-e8324f3144f8"
UUID_SUB="183ac428-ed70-50b0-e30f-b2f9de67716e" LABEL="portatil.localdomain:0"
```

```
TYPE="linux_raid_member"
/dev/loop2: UUID="b5fd01dc-53a8-20d3-190a-e8324f3144f8"
UUID_SUB="f8e403c8-70e1-845d-e5a7-a13885fd6119" LABEL="portatil.localdomain:0"
TYPE="linux_raid_member"
….
/dev/md0: UUID="005caef9-e1e0-429a-bc81-7fcb5ba290cb" TYPE="ext4"
# mount /dev/md0 /mnt/

# cp -r /boot/ /mnt/

# df -h
S. fitxers        Mida En ús Lliure  %Ús Muntat a
….
/dev/md0          93M   93M      0 100% /mnt
```

Examinar el RAID

L'ordre ***mdadm*** permet examinar i governar els diversos RAID del sistema. També ***/proc*** proporciona informació dels RAID.

Exemple d'ordres RAID:

```
# mdadm --detail --scan
# mdadm --detail /dev/md0
# mdadm --query /dev/loop0
# mdadm --examine /dev/loop0
# cat /proc/mdstat
```

```
# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 name=portatil.localdomain:0
UUID=b5fd01dc:53a820d3:190ae832:4f3144f8

# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Fri Feb  6 20:56:09 2015
        Raid Level : raid1
        Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 102272 (99.89 MiB 104.73 MB)
   Raid Devices : 3
  Total Devices : 3
        Persistence : Superblock is persistent

        Update Time : Fri Feb  6 21:24:20 2015
        State : clean
 Active Devices : 3
Working Devices : 3
```

```
 Failed Devices : 0
  Spare Devices : 0

           Name : portatil.localdomain:0  (local to host portatil.localdomain)
           UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
         Events : 17

    Number   Major   Minor   RaidDevice State
       0       7        0        0      active sync   /dev/loop0
       1       7        1        1      active sync   /dev/loop1
       2       7        2        2      active sync   /dev/loop2

# mdadm --query /dev/loop0
/dev/loop0: is not an md array
/dev/loop0: device 0 in 3 device active raid1 /dev/md0.  Use mdadm --examine for more detail.

# mdadm --examine /dev/loop0
/dev/loop0:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
           Name : portatil.localdomain:0  (local to host portatil.localdomain)
  Creation Time : Fri Feb  6 20:56:09 2015
     Raid Level : raid1
   Raid Devices : 3

 Avail Dev Size : 204672 (99.95 MiB 104.79 MB)
     Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 204544 (99.89 MiB 104.73 MB)
    Data Offset : 128 sectors
   Super Offset : 8 sectors
          State : clean
    Device UUID : b36d27f4:3024029e:42df5e2d:0cd3517d

    Update Time : Fri Feb  6 21:26:14 2015
       Checksum : 8bdf41ce - correct
         Events : 17

   Device Role : Active device 0
   Array State : AAA ('A' == active, '.' == missing)
```

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop2[2] loop1[1] loop0[0]
      102272 blocks super 1.2 [3/3] [UUU]
```

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

Generar errada i recuperació

El software de *mdadm* permet simular que s'ha produït una errada de software en un dels discs RAID. Quan un disc dels que formen el RAID es malmet es pot intentar un procés de recuperació (segons el tipus de RAID usat) o simplement eliminar un dels discs i substituir-lo per un de nou.

Exemple d'ordres RAID:

```
# mdadm /dev/md0 --fail /dev/loop1
# mdadm /dev/md0 --remove /dev/loop1
# mdadm --manage /dev/md0 --add /dev/loop3
```

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop2[2] loop1[1] loop0[0]
        102272 blocks super 1.2 [3/3] [UUU]

# mdadm /dev/md0 --fail /dev/loop1
mdadm: set /dev/loop1 faulty in /dev/md0

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop2[2] loop1[1](F) loop0[0]
        102272 blocks super 1.2 [3/2] [U_U]

# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Fri Feb  6 20:56:09 2015
        Raid Level : raid1
        Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 102272 (99.89 MiB 104.73 MB)
   Raid Devices : 3
  Total Devices : 3
        Persistence : Superblock is persistent

        Update Time : Fri Feb  6 21:44:57 2015
        State : clean, degraded
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 0

        Name : portatil.localdomain:0  (local to host portatil.localdomain)
```

```
            UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
            Events : 19

            Number   Major   Minor   RaidDevice State
            0        7        0        0        active sync   /dev/loop0
            1        0        0        1        removed
            2        7        2        2        active sync   /dev/loop2

            1        7        1        -        faulty   /dev/loop1
```

Un cop ha fallat el dispositiu /dev/loop1 s’elimina del raid:

```
# mdadm /dev/md0 --remove /dev/loop1
mdadm: hot removed /dev/loop1 from /dev/md0

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop2[2] loop0[0]
            102272 blocks super 1.2 [3/2] [U_U]

# dd if=/dev/zero of=disc04.img bs=1k count=100k
# losetup /dev/loop3 /opt/raid/disc04.img
# mdadm --manage /dev/md0 --add /dev/loop3
mdadm: added /dev/loop3

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop3[3] loop2[2] loop0[0]
            102272 blocks super 1.2 [3/3] [UUU]

# mdadm --detail /dev/md0
/dev/md0:
            Version : 1.2
  Creation Time : Fri Feb  6 20:56:09 2015
            Raid Level : raid1
            Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 102272 (99.89 MiB 104.73 MB)
   Raid Devices : 3
/  Total Devices : 3
            Persistence : Superblock is persistent

            Update Time : Fri Feb  6 22:01:15 2015
            State : clean
 Active Devices : 3
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 0

            Name : portatil.localdomain:0  (local to host portatil.localdomain)
```

```
        UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
        Events : 41

        Number   Major   Minor   RaidDevice State
        0         7        0        0        active sync   /dev/loop0
        3         7        3        1        active sync   /dev/loop3
        2         7        2        2        active sync   /dev/loop2
```

Aturar / Engegar el RAID

Per engegar un raid hi ha tres mecanísmes:

- Demanar-li a mdadm que examini totes les particions del sistema (*scan*) i 'assembli' aquelles que creu que formen un raid (*assemble*). Cal recordar que les particions tenen un superblock que indica si són part d'un Array i de quin. Aò no vol dir que sempre ho faci bé. Podem tenir vàries particions etiquetades (algunes antigues) i que mdadmin no 'assembli' allò que volem.
- Manar a mdadm que ajunti les particions concretes que li indiquem com a arguments. En aquest cas forcem que usi les particions indicades.
- Haver generat un fitxer de configuració /etc/mdadm.conf on hi ha la informació necessària per engegar automatitzadament el raid. Aquesta és la opció per poder engegar els raid a l'arrancada del sistema.

Exemple d'ordres RAID:

```
# mdadm --stop /dev/md0
# mdadm --assemble --scan
# mdadmin --assemble /dev/md0 --run /dev/loop0 /dev/loop1/dev/loop2
# mdadm --detail --scan > /etc/mdadm.conf
```

```
# mdadm --stop /dev/md0
mdadm: Cannot get exclusive access to /dev/md0:Perhaps a running process, mounted
filesystem or active volume group?
# umount /mnt

# mdadm --stop /dev/md0
mdadm: stopped /dev/md0

# mdadm --assemble --scan
mdadm: failed to add /dev/loop3 to /dev/md0: Invalid argument
mdadm: /dev/md0 has been started with 2 drives (out of 3).

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop0[0] loop2[2]
        102272 blocks super 1.2 [3/2] [U_U]
```

```
# mdadm --detail /dev/md0
/dev/md0:
            Version : 1.2
  Creation Time : Fri Feb  6 20:56:09 2015
            Raid Level : raid1
            Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 102272 (99.89 MiB 104.73 MB)
   Raid Devices : 3
  Total Devices : 2
            Persistence : Superblock is persistent

            Update Time : Fri Feb  6 22:05:55 2015
            State : clean, degraded
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

            Name : portatil.localdomain:0  (local to host portatil.localdomain)
            UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
            Events : 41

            Number   Major   Minor   RaidDevice State
            0        7       0       0       active sync   /dev/loop0
            1        0       0       1       removed
            2        7       2       2       active sync   /dev/loop2

# mdadm -v /dev/md0 --add /dev/loop3
mdadm: added /dev/loop3
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop3[3] loop0[0] loop2[2]
            102272 blocks super 1.2 [3/2] [U_U]
            [========>...........]  recovery = 43.5% (44800/102272) finish=0.1min
speed=7466K/sec
unused devices: <none>

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop3[3] loop0[0] loop2[2]
            102272 blocks super 1.2 [3/2] [U_U]
            [================>....]  recovery = 81.0% (83200/102272) finish=0.0min
speed=7563K/sec
unused devices: <none>

# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 loop3[3] loop0[0] loop2[2]
            102272 blocks super 1.2 [3/3] [UUU]
```

```
# mdadm --detail /dev/md0
/dev/md0:
            Version : 1.2
  Creation Time : Fri Feb  6 20:56:09 2015
            Raid Level : raid1
            Array Size : 102272 (99.89 MiB 104.73 MB)
  Used Dev Size : 102272 (99.89 MiB 104.73 MB)
   Raid Devices : 3
  Total Devices : 3
            Persistence : Superblock is persistent

            Update Time : Sat Feb  7 14:00:03 2015
            State : clean
 Active Devices : 3
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 0

            Name : portatil.localdomain:0  (local to host portatil.localdomain)
            UUID : b5fd01dc:53a820d3:190ae832:4f3144f8
            Events : 62

            Number   Major   Minor   RaidDevice State
            0        7        0        0        active sync   /dev/loop0
            3        7        3        1        active sync   /dev/loop3
            2        7        2        2        active sync   /dev/loop2
```

Automatitzar l'arrancada del RAID

Per automatitzar l'arrancada es genera un fitxer de configuració ***mdadm.conf***. També cal desar al ***/etc/fstab*** l'entrada per a que munti el RAID automàticament si es vol que sigui així.

```
# mdadm --detail --scan > /etc/mdadm.conf
# cat /etc/mdadm.conf
ARRAY /dev/md0 metadata=1.2 name=portatil.localdomain:0
UUID=b5fd01dc:53a820d3:190ae832:4f3144f8

# cat /etc/fstab
/dev/md0 /mnt ext4 default 0 0
```

Pràctica proposada: level 1 raid + spare

Crear un raid de Level1 amb dues particions (loop0 i loop1) i dos discs d'spare. I practicar:

- la creació del raid.
- observar-ne les dades.

- fail de un disc (spare entra en acció)
- fail de un altre disc (spare entra en acció)
- eliminar els dos dic fail.
- afegir de nou els dos disc (ara fan el rol de spare)

Pràctica (2): Raid Level 5

Exemple Raid Level 5: degradar i failed

En aquest exercici pràctic aprendrem a:

- crear un rad level 5 amb tres discs i un de spare.
- formatar, muntar i omplir amb dades el device /dev/md0. La mida d'emmagatzemament útil és ⅔ de l'espai dels tres discs.
- generar un fail a un dels dics del raid i observar com el disc de spare entra en acció. Estat: 3 Raid,1 Fail ,0 Spare.
- generar un nou fail a un dels discs i ara el raid queda degradat. Manté les dades però ja no hi ha redundància. Estat: 2 Raid, 2 Fail, 0 Spare
- amb un nou fail el raid quedarà aturat i es perdran del tot les dades, que ja no seran recuperàbles. Estat 1 raid, 3 Fail, 0 Spare.
- encara que es treguin els dos dics fails i se n'afegeixin dos de nous (els mateixos) el raid ja no pot funcionar degut a la pèrdua de dades.
- tampoc encara que s'aturi i s'engegi de nou.

Crear un raid de nivell 5 amb tres unitats (loop0, loop1 i loop2, més un disc de spare). Amb l'opció verbose podem observar que ens diu que el layout és left-symmetric, i que un dels dics és significativament més gran que els altres (més de un 1%).

```
# mdadm -v --create /dev/md0 --level 5 --raid-devices 3 /dev/loop0 /dev/loop1
/dev/loop2 --spare-devices 1 /dev/loop3
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Fri Feb  6 20:56:09 2015
mdadm: /dev/loop1 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Fri Feb  6 20:56:09 2015
mdadm: /dev/loop2 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Fri Feb  6 20:56:09 2015
mdadm: /dev/loop3 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Fri Feb  6 20:56:09 2015
mdadm: size set to 101888K
Continue creating array?
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[4] loop3[3](S) loop1[1] loop0[0]
        203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
```

```
# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
  Creation Time : Sat Feb  7 14:23:17 2015
           Raid Level : raid5
           Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 4
           Persistence : Superblock is persistent

           Update Time : Sat Feb  7 14:23:39 2015
           State : clean
 Active Devices : 3
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1

           Layout : left-symmetric
           Chunk Size : 512K

           Name : portatil.localdomain:0  (local to host portatil.localdomain)
           UUID : efa4df5b:9cc6b5b7:68f0a73f:a4cf4931
           Events : 18

           Number   Major   Minor   RaidDevice State
           0         7         0         0         active sync   /dev/loop0
           1         7         1         1         active sync   /dev/loop1
           4         7         2         2         active sync   /dev/loop2

           3         7         3         -         spare   /dev/loop3
```

```
# mkfs -t ext4 /dev/md0
mke2fs 1.42.3 (14-May-2012)
Discarding device blocks: fet
Etiqueta del sistema de fitxers=
OS type: Linux
Mida del bloc=1024 (log=0)
Mida del fragment=1024 (log=0)
Stride=512 blocks, Stripe width=1024 blocks
51000 inodes, 203776 blocks
10188 blocks (5.00%) reserved for the super user
Bloc de dades inicial=1
Màxim de blocs del sistema de fitxers=67371008
25 grups de blocs
8192 blocs per grup, 8192 fragments per grup
2040 nodes-i per grup
Còpies de seguretat del superbloc desades en els blocs:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: fet
Escriptura de les taules de nodes-i:fet
Creació del registre de transaccions (4096 blocs): fet
Escriptura de la informació dels súperblocs i de comptabilitat del sistema de fitxers:fet

# mount /dev/md0 /mnt/

# df -h
S. fitxers         Mida En ús Lliure  %Ús Muntat a
...
/dev/md0           189M  1,6M   178M   1% /mnt
```

Observar que en tractar-se d'un RAID5 format per tres unitats de 100M més una de spare de 100M, l'espai utilitzable d'emmagatzemament és proper als 200M. Dels tres discos de RAID dos emmagatzemem dades i un tercer paritat, però no tal qual (seria raid 4) sinó que entre els tres discos es barregen dades i paritat.

Així doncs, si falla un dels tres discos el sistema deixa de funcionar. Si hi ha un disc de spare, aquest s'hauria d'activar automàticament per solventar el problema. Si un altre disc falla, llavors el RAID deixa de funcionar.

```
# mdadm /dev/md0 --fail /dev/loop1
mdadm: set /dev/loop1 faulty in /dev/md0

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[4] loop3[3] loop1[1](F) loop0[0]
         203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [U_U]
         [=======>............]  recovery = 37.0% (38232/101888) finish=0.0min
speed=12744K/sec

# mdadm --detail /dev/md0
/dev/md0:
         Version : 1.2
  Creation Time : Sat Feb  7 14:23:17 2015
         Raid Level : raid5
         Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 4
         Persistence : Superblock is persistent
         Update Time : Sat Feb  7 14:45:09 2015
         State : clean, degraded, recovering
 Active Devices : 2
Working Devices : 3
 Failed Devices : 1
  Spare Devices : 1
         Layout : left-symmetric
         Chunk Size : 512K
 Rebuild Status : 81% complete
         Name : portatil.localdomain:0  (local to host portatil.localdomain)
         UUID : efa4df5b:9cc6b5b7:68f0a73f:a4cf4931
         Events : 33
         Number   Major   Minor   RaidDevice State
         0        7       0       0        active sync   /dev/loop0
         3        7       3       1        spare rebuilding   /dev/loop3
         4        7       2       2        active sync   /dev/loop2

         1        7       1       -        faulty   /dev/loop1

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[4] loop3[3] loop1[1](F) loop0[0]
         203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

# mdadm /dev/md0 --remove /dev/loop1
mdadm: hot removed /dev/loop1 from /dev/md0
```

Ara el RAID5 disposa de tres unitats loop0, loop2 i loop3, si una d’elles falla deixarà de funcionar.

```
# mdadm /dev/md0 --fail /dev/loop2
mdadm: set /dev/loop2 faulty in /dev/md0

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[4](F) loop3[3] loop0[0]
        203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]

# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Sat Feb  7 14:23:17 2015
        Raid Level : raid5
        Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 3
        Persistence : Superblock is persistent

        Update Time : Sat Feb  7 14:49:59 2015
        State : clean, degraded
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 0

        Layout : left-symmetric
        Chunk Size : 512K

        Name : portatil.localdomain:0  (local to host portatil.localdomain)
        UUID : efa4df5b:9cc6b5b7:68f0a73f:a4cf4931
        Events : 42

        Number   Major   Minor   RaidDevice State
        0        7       0       0       active sync   /dev/loop0
        3        7       3       1       active sync   /dev/loop3
        2        0       0       2       removed

        4        7       2       -       faulty   /dev/loop2
```

El RAID encara funciona. Anem a provar a espatllar una nova unitat,per exemple loop3.

```
# mdadm /dev/md0 --fail /dev/loop3
mdadm: set /dev/loop3 faulty in /dev/md0

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[4](F) loop3[3](F) loop0[0]
        203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/1] [U__]
```

```
# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
  Creation Time : Sat Feb  7 14:23:17 2015
           Raid Level : raid5
           Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 3
           Persistence : Superblock is persistent

           Update Time : Sat Feb  7 14:52:34 2015
           State : clean, FAILED
 Active Devices : 1
Working Devices : 1
 Failed Devices : 2
  Spare Devices : 0

           Layout : left-symmetric
           Chunk Size : 512K

           Name : portatil.localdomain:0  (local to host portatil.localdomain)
           UUID : efa4df5b:9cc6b5b7:68f0a73f:a4cf4931
           Events : 46

           Number   Major   Minor   RaidDevice State
           0        7        0        0        active sync   /dev/loop0
           1        0        0        1        removed
           2        0        0        2        removed

           3        7        3        -        faulty   /dev/loop3
           4        7        2        -        faulty   /dev/loop2
```

Finalment anem a afegir dues noves unitats al raid, primer extreurem les que han fallat (loop2 i loop3) i les reemplaçarem per dues de noves, que casualment tornen a ser loop2 i loop3.

```
# mdadm /dev/md0 --remove /dev/loop2 /dev/loop3
mdadm: hot removed /dev/loop2 from /dev/md0
mdadm: hot removed /dev/loop3 from /dev/md0

# mdadm /dev/md0 --add /dev/loop2 /dev/loop3
mdadm: /dev/md0 has failed so using --add cannot work and might destroy
mdadm: data on /dev/loop2.  You should stop the array and re-assemble it.

# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

```
# mdadm --assemble --scan
mdadm: /dev/md/0 assembled from 1 drive - not enough to start the array.
mdadm: No arrays found in config file or automatically
```

No funciona!. S'han perdut les dades d'un dels dics, ja no es possible reanudar-lo. S'han apagat massa discs i s'han perdut les dades

Exemple Raid Level 5: degradar i recuperar

Aquest exemple és el mateix que l'anterior on d'un RAID5 de tres unitats més una de spare i se n'han espatllat dues. Primer ha entrat en funcionament l'spare però en el segon fail el raid ha quedaty degradat (sense redundància). A continuació s'han eliminat els dos discs fail i se n'han afegit dos de nous (els mateixos). El raid ha fet la sincronització en un dels discs i l'altre ha passat a ser spare.

- crear el raid level 5 de 3 raid i un spare.
- simular dos fails.
- *Atenció*: si heu simulat els dos fails ràpidament no haureu deixat temps al raid de sincronitzar de nou el disc de spare i haurà passat a *fail*. Caldrà tornar a començar (i fer-ho més a poc a poc).
- observar estat de degraded: 2 raid, 2 fails, 0 spare.
- treure els dos discs fail
- afegir dos dics nous al raid (els dos que hem tret)
- observar que es fa la sincronització i passa a estat ok.

```
# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop3[3] loop0[0]
      203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [U_U]

# mdadm --detail /dev/md0
/dev/md0:
      Version : 1.2
  Creation Time : Sat Feb  7 15:10:35 2015
      Raid Level : raid5
      Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 2
      Persistence : Superblock is persistent
      Update Time : Sat Feb  7 15:14:21 2015
      State : clean, degraded
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
```

```
  Spare Devices : 0
           Layout : left-symmetric
           Chunk Size : 512K
           Name : portatil.localdomain:0  (local to host portatil.localdomain)
           UUID : 8a1b6778:6955670b:931d8ae7:c53a0de4
           Events : 45
           Number   Major   Minor   RaidDevice State
           0        7        0        0        active sync   /dev/loop0
           1        0        0        1        removed
           3        7        3        2        active sync   /dev/loop3

# mdadm /dev/md0 --add /dev/loop1
mdadm: added /dev/loop1

# mdadm /dev/md0 --add /dev/loop2
mdadm: added /dev/loop2

# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4]
md0 : active raid5 loop2[5](S) loop1[4] loop3[3] loop0[0]
           203776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
  Creation Time : Sat Feb  7 15:10:35 2015
           Raid Level : raid5
           Array Size : 203776 (199.03 MiB 208.67 MB)
  Used Dev Size : 101888 (99.52 MiB 104.33 MB)
   Raid Devices : 3
  Total Devices : 4
           Persistence : Superblock is persistent
           Update Time : Sat Feb  7 15:16:20 2015
           State : clean
 Active Devices : 3
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1
           Layout : left-symmetric
           Chunk Size : 512K
           Name : portatil.localdomain:0  (local to host portatil.localdomain)
           UUID : 8a1b6778:6955670b:931d8ae7:c53a0de4
           Events : 67
           Number   Major   Minor   RaidDevice State
           0        7        0        0        active sync   /dev/loop0
           4        7        1        1        active sync   /dev/loop1
           3        7        3        2        active sync   /dev/loop3
           5        7        2        -        spare   /dev/loop2
```

```
# df -h
```

```
S. fitxers        Mida En ús Lliure  %Ús Muntat a
….
/dev/md0          189M   93M    86M  52% /mnt
```

Pràctica (3): start/stop/assemble/md127

Ordre mdadm

L'utilitat GNU/Linux d'administració de discs RAID és *mdadmin*, aquest apartat mostra part del contingut del seu man.

```
MDADM(8)                                                        MDADM(8)
NAME
        mdadm - manage MD devices aka Linux Software RAID
SYNOPSIS
        mdadm [mode] <raiddevice> [options] <component-devices>

DESCRIPTION
RAID devices are virtual devices created from two or more real block devices. This
allows multiple devices (typically disk drives or partitions thereof) to be combined into a
single device to hold (for example) a single filesystem. Some RAID levels include
redundancy and so can survive some degree of device failure.
--
Linux Software RAID devices are implemented through the md (Multiple Devices) device
driver.

Currently, Linux supports LINEAR md devices, RAID0 (striping), RAID1 (mirroring),
RAID4, RAID5, RAID6, RAID10, MULTIPATH, FAULTY, and CONTAINER.

MULTIPATH is not a Software RAID mechanism, but does involve multiple devices: each
device is a path to one common physical storage device. New installations should not use
md/multipath as it is not well supported and has no ongoing development. Use the Device
Mapper based multipath-tools instead.

FAULTY is also not true RAID, and it only involves one device. It provides a layer over a
true device that can be used to inject faults.

CONTAINER is different again. A CONTAINER is a collection of devices that are
managed as a set. This is similar to the set of devices connected to a hardware RAID
controller. The set of devices may contain a number of different RAID arrays each utilising
some (or all) of the blocks from a number of the devices in the set. For example, two
devices in a 5-device set might form a RAID1 using the whole devices. The remaining
three might have a RAID5 over the first half of each device, and a RAID0 over the second
half.
```

MODES
mdadm has several major modes of operation:

Assemble
Assemble the components of a previously created array into an active array. Components can be explicitly given or can be searched for. mdadm checks that the components do form a bona fide array, and can, on request, fiddle superblock information so as to assemble a faulty array.

Build
Build an array that doesn't have per-device metadata (superblocks). For these sorts of arrays, mdadm cannot differentiate between initial creation and subsequent assembly of an array. It also cannot perform any checks that appropriate components have been requested. Because of this, the Build mode should only be used together with a complete understanding of what you are doing.

Create
Create a new array with per-device metadata (superblocks). Appropriate metadata is written to each device, and then the array comprising those devices is activated. A 'resync' process is started to make sure that the array is consistent (e.g. both sides of a mirror contain the same data) but the content of the device is left otherwise untouched.
The array can be used as soon as it has been created. There is no need to wait for the initial resync to finish.

Follow or Monitor
Monitor one or more md devices and act on any state changes. This is only meaningful for RAID1, 4, 5, 6, 10 or multipath arrays, as only these have interesting state. RAID0 or Linear never have missing, spare, or failed drives, so there is nothing to monitor.

Grow
Grow (or shrink) an array, or otherwise reshape it in some way. Currently supported growth options including changing the active size of component devices and changing the number of active devices in Linear and RAID levels 0/1/4/5/6, changing the RAID level between 0, 1, 5, and 6, and between 0 and 10, changing the chunk size and layout for RAID 0,4,5,6, as well as adding or removing a write-intent bitmap.

Incremental Assembly
Add a single device to an appropriate array. If the addition of the device makes the array runnable, the array will be started. This provides a convenient interface to a hot-plug system. As each device is detected, mdadm has a chance to include it in some array as appropriate. Optionally, when the --fail flag is passed in we will remove the device from any active array instead of adding it.
If a CONTAINER is passed to mdadm in this mode, then any arrays within that container will be assembled and started.

Manage
This is for doing things to specific components of an array such as adding new spares and removing faulty devices.

Misc
This is an 'everything else' mode that supports operations on active arrays, operations on component devices such as erasing old superblocks, and information gathering operations.

Auto-detect
This mode does not act on a specific device or array, but rather it requests the Linux Kernel to activate any auto-detected arrays.

FILES

/proc/mdstat
If you're using the /proc filesystem, /proc/mdstat lists all active md devices with information about them. mdadm uses this to find arrays when --scan is given in Misc mode, and to monitor array reconstruction on Monitor mode.

/etc/mdadm.conf
The config file lists which devices may be scanned to see if they contain MD super block, and gives identifying information (e.g. UUID) about known MD arrays. See mdadm.conf(5) for more details.

/dev/md/md-device-map
When --incremental mode is used, this file gets a list of arrays currently being created.

mdadm --query /dev/name-of-device
This will find out if a given device is a RAID array, or is part of one, and will provide brief information about the device.

mdadm --assemble --scan
This will assemble and start all arrays listed in the standard config file. This command will typically go in a system startup file.

mdadm --stop --scan
This will shut down all arrays that can be shut down (i.e. are not currently in use). This will typically go in a system shutdown script.

mdadm --follow --scan --delay=120
If (and only if) there is an Email address or program given in the standard config file, then monitor the status of all arrays listed in that file by polling them ever 2 minutes.

mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hd[ac]1
Create /dev/md0 as a RAID1 array consisting of /dev/hda1 and /dev/hdc1.

echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf
mdadm --detail --scan >> mdadm.conf
This will create a prototype config file that describes currently active arrays that are known to be made from partitions of IDE or SCSI drives. This file should be reviewed before being used as it may contain unwanted detail.

mdadm --create --help

Provide help about the Create mode.

mdadm --config --help
Provide help about the format of the config file.

mdadm --help
Provide general help.

Exemple-1: regeneració automàtica / regeneració per nom de raid

Usant el fitxer /etc/mdadm.conf es pot desar la configuració del array(s) a /etc per tal d'automatitzar l'arrancada del array. En engegar el sistema automàticament posarà en marxa els arrays que s'hi indiquen. El fitxer també permet fer *l'assemble* del raid per el nom de raid.

Recordeu que per engegar el raid (assemble) hi ha les opcions:

- en crear-lo.
- "mdadm --assemble --scan" que examina les particions i intenta posar en marxa el que creu coherent.
- "mdadm --assemble nom-raid" cal que aquest raid estigui definit en el /etc/mdadm.conf.
- automàticament en arrancar el sistema si hi ha el fitxet /etc/mdadm.conf.
- manualment amb l'ordre "mdadm --assemble nom-raid device1 device2 …".

```
[root@d01 m11]# mdadm --create /dev/md/backup --level=1 --raid-devices=2
/dev/loop0 /dev/loop1 --spare-devices=1 /dev/loop2

[root@d01 m11]# tree /dev/disk
/dev/disk
├── by-id
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ -> ../../sda
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part1 -> ../../sda1
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part5 -> ../../sda5
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part6 -> ../../sda6
│   ├── md-name-d01:backup -> ../../md127
│   ├── md-uuid-0606855a:bc05cd6d:e507c9da:466e46cb -> ../../md127
│   ├── wwn-0x5000c50064eb697f -> ../../sda
│   ├── wwn-0x5000c50064eb697f-part1 -> ../../sda1
│   ├── wwn-0x5000c50064eb697f-part5 -> ../../sda5
│   └── wwn-0x5000c50064eb697f-part6 -> ../../sda6
├── by-label
│   └── FEDORA24 -> ../../sda5
├── by-path
│   ├── pci-0000:00:1f.2-ata-1 -> ../../sda
│   ├── pci-0000:00:1f.2-ata-1-part1 -> ../../sda1
│   ├── pci-0000:00:1f.2-ata-1-part5 -> ../../sda5
│   └── pci-0000:00:1f.2-ata-1-part6 -> ../../sda6
```

```
└── by-uuid
        ├── 83339907-96a2-4d63-88ef-41bd4b1d13b1 -> ../../md127
        ├── b09b643e-5709-4a97-b79b-eba736188534 -> ../../sda6
        └── dd3d92dd-8b09-443b-b320-002a8aaa5175 -> ../../sda5
```

```
[root@d01 m11]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 loop2[2] loop0[3](S) loop1[1]
        204608 blocks super 1.2 [2/2] [UU]
unused devices: <none>

[root@d01 m11]# mdadm --examine --scan > /etc/mdadm.conf
[root@d01 m11]# cat /etc/mdadm.conf
ARRAY /dev/md/backup  metadata=1.2 UUID=0606855a:bc05cd6d:e507c9da:466e46cb
name=d01:backup
   spares=1

[root@d01 m11]# mdadm --stop /dev/md/backup
mdadm: stopped /dev/md/backup

[root@d01 m11]# mdadm --assemble /dev/md/backup
mdadm: /dev/md/backup has been started with 2 drives and 1 spare.

[root@d01 m11]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 loop2[2] loop0[3](S) loop1[1]
        204608 blocks super 1.2 [2/2] [UU]
unused devices: <none>
```

Exemple-2: Assemble indicant les parts

En aquest apartat veurem com generar un raid indicant les particions que n'han de formar part. Engega un raid de level 1 amb tres discs raids i cap spare.

```
# mdadm --assemble /dev/md/myraid1 --run /dev/loop0 /dev/loop1 /dev/loop2
mdadm: /dev/md/myraid1 has been started with 3 drives.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md127 : active raid5 loop0[6] loop2[4] loop1[5]
        202752 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

Exemple-3: Múltiples Raid.

En aquest exemple veurem com combinar múltiples raid. Un de anomenat /dev/md/backup i un anomenat /dev/md/liveone.

```
[root@d01 m11]# mdadm --create /dev/md/liveone --level=1 --raid-devices=2
/dev/sda7 /dev/sda8 --spare-devices=1 /dev/sda9

[root@d01 ~]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sda7[0] sda8[1] sda9[2](S)
        1047552 blocks super 1.2 [2/2] [UU]
unused devices: <none>

[root@d01 ~]# mdadm --detail /dev/md/liveone
/dev/md/liveone:
        Version : 1.2
  Creation Time : Tue Feb 14 13:48:16 2017
        Raid Level : raid1
        Array Size : 1047552 (1023.00 MiB 1072.69 MB)
  Used Dev Size : 1047552 (1023.00 MiB 1072.69 MB)
   Raid Devices : 2
  Total Devices : 3
        Persistence : Superblock is persistent
        Update Time : Tue Feb 14 13:48:43 2017
        State : clean
 Active Devices : 2
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1
        Name : d01:liveone  (local to host d01)
        UUID : d4e0178e:0a3e86d5:f1862966:1df82ae5
        Events : 17
        Number   Major   Minor   RaidDevice State
        0        8       7       0       active sync   /dev/sda7
        1        8       8       1       active sync   /dev/sda8
        2        8       9       -       spare   /dev/sda9
```

```
[root@d01 ~]# tree /dev/disk/
/dev/disk/
├── by-id
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ -> ../../sda
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part1 -> ../../sda1
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part5 -> ../../sda5
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part6 -> ../../sda6
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part7 -> ../../sda7
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part8 -> ../../sda8
│   ├── ata-ST500DM002-1BD142_Z3TRHNRQ-part9 -> ../../sda9
│   ├── md-name-d01:backup -> ../../md126
│   ├── md-name-d01:liveone -> ../../md127
│   ├── md-uuid-0606855a:bc05cd6d:e507c9da:466e46cb -> ../../md126
│   ├── md-uuid-d4e0178e:0a3e86d5:f1862966:1df82ae5 -> ../../md127
...
```

```
[root@d01 ~]# mdadm --examine --scan
ARRAY /dev/md/liveone  metadata=1.2 UUID=d4e0178e:0a3e86d5:f1862966:1df82ae5
name=d01:liveone
   spares=1
ARRAY /dev/md/backup  metadata=1.2 UUID=0606855a:bc05cd6d:e507c9da:466e46cb
name=d01:backup
   spares=1

[root@d01 ~]# mdadm --examine --scan > /etc/mdadm.conf
```

```
root@d01 ~]# mdadm --stop /dev/md/backup
mdadm: stopped /dev/md/backup

[root@d01 ~]# mdadm --stop /dev/md/liveone
mdadm: stopped /dev/md/liveone

[root@d01 ~]# mdadm --assemble --scan
mdadm: /dev/md/liveone has been started with 2 drives and 1 spare.
mdadm: /dev/md/backup has been started with 2 drives and 1 spare.
```

```
[root@d01 ~]# mdadm --stop --scan

[root@d01 ~]# mdadm --assemble /dev/md/backup
mdadm: /dev/md/backup has been started with 2 drives and 1 spare.

[root@d01 ~]# mdadm --assemble /dev/md/liveone
mdadm: /dev/md/liveone has been started with 2 drives and 1 spare.
```

Reshape: raid-devices / Level / size

Reshape raid-devices

GROW MODE
The GROW mode is used for changing the size or shape of an active array. For this to work, the kernel must support the necessary change. Various types of growth are being added during 2.6 development.

Currently the supported changes include

- change the "**size**" attribute for RAID1, RAID4, RAID5 and RAID6.
- increase or decrease the "raid-devices" attribute of RAID0, RAID1, RAID4, RAID5, and RAID6.
- change the **chunk-size** and layout of RAID0, RAID4, RAID5, RAID6 and RAID10.
- **convert** between RAID1 and RAID5, between RAID5 and RAID6, between RAID0, RAID4, and RAID5, and between RAID0 and RAID10 (in the near-2 mode).
- add a write-intent bitmap to any array which supports these bitmaps, or remove a write-intent bitmap from such an array.

En aquest apartat veurem exemples de modificar el shape del raid, per exemple:

- en un raid level 1 de 2 discs passar a un format de 3 discs + un de spare (es pot?).
- en un raid level 5 de 3 discs i un spare passar a 5 discs i un de spare.

Per fer aquest exercicis hem generat:

- sis discs de 100M del loop0 al loop5.
- 3 discs de 500M del loop7 al loop9.

RAID-DEVICES CHANGES

A RAID1 array can work with any number of devices from 1 upwards (though 1 is not very useful). There may be times which you want to increase or decrease the number of active devices. Note that this is different to hot-add or hot-remove which changes the number of inactive devices.

When reducing the number of devices in a RAID1 array, the slots which are to be removed from the array must already be vacant. That is, the devices which were in those slots must be failed and removed.

When the number of devices is increased, any hot spares that are present will be activated immediately.

Changing the number of active devices in a RAID5 or RAID6 is much more effort. Every block in the array will need to be read and written back to a new location. From 2.6.17, the Linux Kernel is able to increase the number of devices in a RAID5 safely, including restarting an interrupted "reshape". From 2.6.31, the Linux Kernel is able to increase or decrease the number of devices in a RAID5 or RAID6.

From 2.6.35, the Linux Kernel is able to convert a RAID0 in to a RAID4 or RAID5. mdadm uses this functionality and the ability to add devices to a RAID4 to allow devices to be added to a RAID0. When requested to do this, mdadm will convert the RAID0 to a RAID4, add the necessary disks and make the reshape happen, and then convert the RAID4 back to RAID0.

When decreasing the number of devices, the size of the array will also decrease. If there was data in the array, it could get destroyed and this is not reversible, so you should firstly shrink the filesystem on the array to fit within the new size. To

```
help  prevent  accidents, mdadm requires that the size of the array be decreased first with mdadm --grow --array-size.  This
is a reversible change which simply makes the end of the array inaccessible.  The integrity of any data can then be checked
before the non-reversible reduction in the number of devices is request.

When  relocating  the  first  few stripes on a RAID5 or RAID6, it is not possible to keep the data on disk completely
consistent and crashproof.  To provide the required safety, mdadm disables writes to the array while this "critical section" is
reshaped, and takes a backup ofthe  data  that  is  in that section.  For grows, this backup may be stored in any spare
devices that the array has, however it can also bestored in a separate file specified with the --backup-file option, and is
required to be specified for shrinks, RAID level changes and layout  changes.   If this option is used, and the system does
crash during the critical period, the same file must be passed to --assemble to restore the backup and reassemble the array.
When shrinking rather than growing the array, the reshape is done from the  end  towards  the beginning, so the "critical
section" is at the end of the reshape.
```

Level1 2r - Level1 3r/1s

En aquest apartat passarem de un raid level 1 de 2 discs a tenir 3 discs raid i un de spare (o potser l'apare no…)

```
# mdadm --create /dev/md/myraid1 --level=1 --raid-devices=2 /dev/loop0 /dev/loop1
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid1 devices=2 ctime=Thu Feb  7 18:18:13 2019
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/loop1 appears to be part of a raid array:
        level=raid1 devices=2 ctime=Thu Feb  7 18:18:13 2019
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid1 loop1[1] loop0[0]
        102272 blocks super 1.2 [2/2] [UU]

# mkfs -t ext4 /dev/md/myraid1
# mount /dev/md/myraid1 /mnt
```

```
# mdadm --grow /dev/md/myraid1 --raid-devices=3 --add /dev/loop2
--spare-devices=/dev/loop3
mdadm: :option --spare-devices not valid in grow mode

# mdadm --grow /dev/md/myraid1 --raid-devices=3 --add /dev/loop2
mdadm: added /dev/loop2
raid_disks for /dev/md/myraid1 set to 3

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
```

```
md127 : active raid1 loop2[2] loop1[1] loop0[0]
        102272 blocks super 1.2 [3/3] [UUU]

# umount /mnt
# mdadm --stop /dev/md/myraid1
mdadm: stopped /dev/md/myraid1
```

Level5 3r,1s / Level5 5r,1s

En aquest apartat passarem de tenir un raid level 5 amb tres discs de raid i un de spare a un format amb cinc discs de raid i un de spare

```
# mdadm --create /dev/md/myraid1 --level=5 --raid-devices=3 /dev/loop0 /dev/loop1
/dev/loop2 --spare-devices=1 /dev/loop3
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 18:36:09 2019
mdadm: /dev/loop1 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 18:36:09 2019
mdadm: /dev/loop2 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 18:36:09 2019
mdadm: /dev/loop3 appears to be part of a raid array:
        level=raid5 devices=3 ctime=Thu Feb  7 18:20:56 2019
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid5 loop2[4] loop3[3](S) loop1[1] loop0[0]
        202752 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

# mkfs -t ext4 /dev/md/myraid1
# mount /dev/md/myraid1 /mnt
```

```
# mdadm --grow /dev/md/myraid1 --raid-devices=5 --add /dev/loop4 /dev/loop5
mdadm: added /dev/loop4
mdadm: added /dev/loop5
mdadm: Need to backup 2048K of critical section..

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid5 loop5[6] loop4[5] loop2[4] loop3[3](S) loop1[1] loop0[0]
        202752 blocks super 1.2 level 5, 512k chunk, algorithm 2 [5/5] [UUUUU]
        [===================>.]  reshape = 99.4% (101376/101376) finish=0.0min
speed=14482K/sec

# umount /mnt
# mdadm --stop /dev/md/myraid1
```

Reshape: Level

```
LEVEL CHANGES

Changing  the  RAID  level of any array happens instantaneously.  However in the RAID5 to RAID6 case this requires a
non-standard layout of the RAID6 data, and in the RAID6 to RAID5 case that non-standard layout is required before the
change can be accomplished.   So  while  the level  change  is  instant,  the  accompanying layout change can take quite a
long time.  A --backup-file is required.  If the array is not simultaneously being grown or shrunk, so that the array size will
remain the same - for example, reshaping a 3-drive RAID5 into  a  4-drive RAID6  - the backup file will be used not just for a
"cricital section" but throughout the reshape operation, as described below under LAYOUT CHANGES.
```

Convertir un raid level 1 a un raid level 5

Partint de un raid level 1 amb dos discs i un de spare es generarà el raid level 5 amb tres discs. Cal observar que el disc de spare és necessari.

```
# mdadm --create /dev/md/myraid1 --level=1 --raid-devices=2 /dev/loop0 /dev/loop1
--spare-devices=1 /dev/loop3
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:08:47 2019
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/loop1 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:08:47 2019
mdadm: /dev/loop3 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:08:47 2019
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.

# mkfs -t ext4 /dev/md/myraid1
# mount /dev/md/myraid1 /mnt
```

```
# mdadm --grow /dev/md/myraid1 --level=5
mdadm: level of /dev/md/myraid1 changed to raid5

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid5 loop3[2](S) loop1[1] loop0[0]
        102272 blocks super 1.2 level 5, 64k chunk, algorithm 2 [2/2] [UU]

# umount /mnt
# mdadm --stop /dev/md/myraid1
```

Impossibily level change

En aquest exemple veiem que un raid level 1 amb més de dos dics no permet passar-lo a level 5.

```
# mdadm --create /dev/md/myraid1 --level=1 --raid-devices=3 /dev/loop0 /dev/loop1
/dev/loop2 --spare-devices=1 /dev/loop3
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid5 devices=2 ctime=Thu Feb  7 19:12:22 2019
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/loop1 appears to be part of a raid array:
        level=raid5 devices=2 ctime=Thu Feb  7 19:12:22 2019
mdadm: /dev/loop2 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:08:47 2019
mdadm: /dev/loop3 appears to be part of a raid array:
        level=raid5 devices=2 ctime=Thu Feb  7 19:12:22 2019
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.

# mdadm --grow /dev/md/myraid1 --level=5
mdadm: /dev/md/myraid1 is performing resync/recovery and cannot be reshaped

# mdadm --grow /dev/md/myraid1 --level=5
mdadm: Impossibly level change request for RAID1
```

Reshape: Size

> SIZE CHANGES
>
> Normally when an array is built the "size" is taken from the smallest of the drives. If all the small drives in an arrays are, one at a time, removed and replaced with larger drives, then you could have an array of large drives with only a small amount used. In this situation, changing the "size" with "GROW" mode will allow the extra space to start being used. If the size is increased in this way, a "resync" process will start to make sure the new parts of the array are synchronised.
>
> Note that when an array changes size, any filesystem that may be stored in the array will not automatically grow or shrink to use or vacate the space. The filesystem will need to be explicitly told to use the extra space after growing, or to reduce its size prior to shrinking the array.
>
> Also the size of an array cannot be changed while it has an active bitmap. If an array has a bitmap, it must be removed before the size can be changed. Once the change is complete a new bitmap can be created.

A partir de un raid level 1 de dos dics de 100M cada un i dos de spare de 500M cada un farem:

- crear el raid
- fallada dels dos discs (un a un) i entrada en funcionament dels spare.
- engrandir al màxim la mida dels dics de raid, ara s'aprofiten 100M dels 500M possibles del raid.

```
# mdadm --create /dev/md/myraid1 --level=1 --raid-devices=2 /dev/loop0 /dev/loop1
--spare-devices=2 /dev/loop7 /dev/loop9
mdadm: /dev/loop0 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:17:32 2019
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/loop1 appears to be part of a raid array:
        level=raid1 devices=3 ctime=Thu Feb  7 19:17:32 2019
mdadm: largest drive (/dev/loop7) exceeds size (102272K) by more than 1%
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid1 loop9[3](S) loop7[2](S) loop1[1] loop0[0]
        102272 blocks super 1.2 [2/2] [UU]

# mkfs -t ext4 /dev/md/myraid1
# mount /dev/md/myraid1 /mnt
```

```
# mdadm /dev/md/myraid1 --fail /dev/loop0
mdadm: set /dev/loop0 faulty in /dev/md/myraid1

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid1 loop9[3] loop7[2](S) loop1[1] loop0[0](F)
        102272 blocks super 1.2 [2/2] [UU]

# mdadm /dev/md/myraid1 --fail /dev/loop1
mdadm: set /dev/loop1 faulty in /dev/md/myraid1

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid1 loop9[3] loop7[2] loop1[1](F) loop0[0](F)
        102272 blocks super 1.2 [2/1] [U_]
        [==========>..........]  recovery = 50.0% (51200/102272) finish=0.0min
speed=51200K/sec

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
```

```
md127 : active raid1 loop9[3] loop7[2] loop1[1](F) loop0[0](F)
          102272 blocks super 1.2 [2/2] [UU]
```

```
# mdadm --detail /dev/md/myraid1
/dev/md/myraid1:
          Version : 1.2
  Creation Time : Thu Feb  7 19:25:13 2019
          Raid Level : raid1
          Array Size : 102272 (99.88 MiB 104.73 MB)
  Used Dev Size : 102272 (99.88 MiB 104.73 MB)
   Raid Devices : 2
  Total Devices : 4
          Persistence : Superblock is persistent

          Update Time : Thu Feb  7 19:29:25 2019
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 2
  Spare Devices : 0

          Name : hostedt:myraid1  (local to host hostedt)
          UUID : 411278dd:169d05d3:afc86d0c:c62f18ee
          Events : 55

          Number   Major   Minor   RaidDevice State
          3        7       9       0        active sync   /dev/loop9
          2        7       7       1        active sync   /dev/loop7

          0        7       0       -        faulty   /dev/loop0
          1        7       1       -        faulty   /dev/loop1
```

Incrementar al màxim el size, però no ho fa! Potser perqupe els discos de fail encara formen part del raid?

```
# mdadm --grow /dev/md/myraid1 --size=max
mdadm: component size of /dev/md/myraid1 has been set to 102320K

# mdadm --detail /dev/md/myraid1
/dev/md/myraid1:
          Version : 1.2
  Creation Time : Thu Feb  7 19:25:13 2019
          Raid Level : raid1
          Array Size : 102320 (99.92 MiB 104.78 MB)
  Used Dev Size : 102320 (99.92 MiB 104.78 MB)
```

```
# mdadm /dev/md/myraid1 --remove /dev/loop0
mdadm: hot removed /dev/loop0 from /dev/md/myraid1
```

```
# mdadm /dev/md/myraid1 --remove /dev/loop1
mdadm: hot removed /dev/loop1 from /dev/md/myraid1

# mdadm --grow /dev/md/myraid1 --size=max
mdadm: component size of /dev/md/myraid1 has been set to 511920K

# mdadm --detail /dev/md/myraid1
/dev/md/myraid1:
        Version : 1.2
  Creation Time : Thu Feb  7 19:25:13 2019
     Raid Level : raid1
     Array Size : 511920 (499.92 MiB 524.21 MB)
  Used Dev Size : 511920 (499.92 MiB 524.21 MB)

# umount /mnt

# mdadm --stop /dev/md/myraid1
mdadm: stopped /dev/md/myraid1
```

```
# mdadm --assemble --scan
mdadm: /dev/md/myraid1 has been started with 2 drives.
mdadm: /dev/md/myraid1_1 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_1 has been started with 1 drive (out of 3) and 2 spares.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.
mdadm: /dev/md/myraid1_2 assembled from 1 drive - not enough to start the array.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md125 : active raid1 loop2[2] loop3[3] loop4[4]
        102272 blocks super 1.2 [3/3] [UUU]

md127 : active raid1 loop9[3] loop7[2]
        511920 blocks super 1.2 [2/2] [UU]

md126 : inactive loop8[3](S)
        510976 blocks super 1.2

# mdadm --stop /dev/md126
mdadm: stopped /dev/md126

# mdadm --stop /dev/md125
mdadm: stopped /dev/md125

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
```

```
md127 : active raid1 loop9[3] loop7[2]
          511920 blocks super 1.2 [2/2] [UU]

# mount /dev/md127 /mnt

# df -h -t ext4
S. fitxers       Mida En ús Lliure  %Ús Muntat a
/dev/sda5        50G   36G       12G  76% /
/dev/sda7        173G   13G   151G   8% /opt
/dev/md127       93M  1,6M       85M   2% /mnt

# resize2fs  /dev/md127
resize2fs 1.42.13 (17-May-2015)
El sistema de fitxers a /dev/md127 està muntat a /mnt; cal un canvi de mida en línia
old_desc_blocks = 1, new_desc_blocks = 2
El sistema de fitxers a /dev/md127 té ara una llargària de 511920 (1k) blocs.

# df -h -t ext4
S. fitxers       Mida En ús Lliure  %Ús Muntat a
/dev/sda5        50G   36G       12G  76% /
/dev/sda7        173G   13G   151G   8% /opt
/dev/md127       481M  2,3M   456M   1% /mnt

# umount /mnt
# mdadm --stop /dev/md127
```

Underconstruction

Superblock: eliminar la marca de raid

```
# mdadm --assemble --scan
mdadm: /dev/md/myraid1 has been started with 2 drives.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_0 has been started with 3 drives.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.
mdadm: /dev/md/myraid1_1 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_1 assembled from 1 drive - not enough to start the array.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.

# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md126 : active raid1 loop3[3] loop2[2] loop4[4]
        102272 blocks super 1.2 [3/3] [UUU]

md127 : active raid1 loop9[3] loop7[2]
        511920 blocks super 1.2 [2/2] [UU]

# mdadm --stop /dev/md126 /dev/md127
mdadm: stopped /dev/md126
mdadm: stopped /dev/md127
```

```
# mdadm -v --zero-superblock /dev/loop2
# mdadm -v --zero-superblock /dev/loop3 /dev/loop4
```

```
# mdadm --assemble --scan
mdadm: /dev/md/myraid1 has been started with 2 drives.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: /dev/md/myraid1_0 assembled from 1 drive - not enough to start the array.
mdadm: Found some drive for an array that is already active: /dev/md/myraid1
mdadm: giving up.

# cat /proc/mdstat
```

```
Personalities : [raid6] [raid5] [raid4] [raid1]
md127 : active raid1 loop9[3] loop7[2]
        511920 blocks super 1.2 [2/2] [UU]
```

```
# mdadm --query /dev/loop0
/dev/loop0: is not an md array
/dev/loop0: device 0 in 2 device mismatch raid1 /dev/md/myraid1.  Use mdadm --examine
for more detail.

# mdadm -v --zero-superblock /dev/loop0

# mdadm --query /dev/loop0
/dev/loop0: is not an md array
```

arrancada de md

```
[root@localhost ~]# dmesg | grep RAID
[ 1077.570311] RAID1 conf printout:
[ 1077.570415] md: resync of RAID array md127
[ 1129.760153] RAID1 conf printout:

[root@localhost ~]# dmesg | less # mes buscar md
```

Resum d’ordres

Exemple d’ordres RAID:

```
# mdadm --create /dev/md0 --chunk=4 --level=1 --raid-devices=3 /dev/loop0 /dev/loop1 /dev/loop2

# mdadm -v --create /dev/md0 --level 5 --raid-devices 3 /dev/loop0 /dev/loop1 /dev/loop2 --spare-devices 1 /dev/loop3
```

Exemple d’ordres RAID:

```
# mdadm --detail --scan
# mdadm --detail /dev/md0
# mdadm --query /dev/loop0
# mdadm --examine /dev/loop0
# cat /proc/mdstat
```

Exemple d’ordres RAID:

```
# mdadm /dev/md0 --fail /dev/loop1
# mdadm /dev/md0 --remove /dev/loop1
# mdadm --manage /dev/md0 --add /dev/loop3
```

Exemple d’ordres RAID:

```
# mdadm --stop /dev/md0
# mdadm --assemble --scan
# mdadmin --assemble /dev/md0 --run /dev/loop0 /dev/loop1/dev/loop2
# mdadm --detail --scan > /etc/mdadm.conf
```

Exercicis classe

Practica 1 LVM / RAID:

- ❏ Crear dd de 400M, pvcreate, vgcreate (volumedisk) i 4 lv create (part1,part2,par3).
- ❏ Crear un raid de nivell 1 amb dos discs i un de spare (part1, part2, part3)
- ❏ Observar mdadmin detail del raid i examine de les parts i /proc/mdstat
- ❏ Espatllar par2, eliminar part2
- ❏ afegir part4 -> no es pot. Afegir part 2 si es pot.
- ❏ fallar part3 i part 3 → raid encara en funcionament però degraded
- ❏ esborar el md0. fer examine de les parts, fer examine de scan.
- ❏ amb mdadm assemble scan recrea de nou el raid amb nom md127 a partir de identificar les particions que tenen superbloc raid.

Practica 2 RAID + LVM

- ❏ crear 3 discs de 400M i posar-los al loop
- ❏ crear un raid /dev/md1 level 5 de les tres particions de 400M. Un raid de 800M.
- ❏ afegir un nou disc de spare de 400M (un nou dd)
- ❏ crear 4 volums lògics del 25% cada un del volumegrup “volumeraid” fet del /dev/md1
- ❏ formatar i muntar a /mnt un dels volums lògics
- ❏ fer un fail de un dels discs del raid i observar que es fa el rsyns. Des del punt de vista del mount de /mnt tot es transparent i no ha passat res.

Practica 3: eliminar-ho tot

- ❏ umount, eliminar els lv, vg, pv de /dev/md1 i stop de /dev/md1. Treure els loops
- ❏ stop de /dev/md0, eliminar els lv, vg, i pv

Pràctica conjunta RAID:

- ❏ funcionament examine, scann, mdadm.conf.
- ❏ examinar magicnumber, METADATA, hexdump.
- ❏ Grow: modificar raid-devices
- ❏ Grow: modificar level
- ❏ Grow: modificar size
- ❏ examinar: /dev/mapper, /sys/block/md, noms de raid, etc.

Paranoia:

- ❏ dd per fer un disc de 1G i amb fdisk fer-ne particions
- ❏ posar el disc al loop0
- ❏ kpatx per carregar-les al /dev/mapper/loop0p1, loop0p2, etc
- ❏ formatar les particion