HowTo ASIX RAID

Redundant Array of Inexpensive Disks

Documentació

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

- 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.

Exercici Pràctic:

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

```
# 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.

```
# 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>
```

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
```

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.

```
# 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

```
# 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
```

Exercici Pràctic: (2)

Crear un raid de nivell 5 amb tres unitats (loop0, loop1 i loop2, més un disc de spare).

```
# 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 preparat

Aquest exemple és el mateix que l'anterior on d'un RAID5 de tres unitats més una de spare de n'han espatllat dues. Primer ha entrat en funcionament l'spare. Llavors s'ha degradat el RAID.

En aquest exemple s'han eliminat les dues unitats que no funcionaven i llavors s'han afegir dues unitats noves. Es pot observar com s'ha recuperat de la fallida i va tot 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
```

Utilitat 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.