

1 RAID en Linux

1.1 Sumario

- 1 Introducción
- 2 Software necesario
- 3 Instalación
- 4 Creación dun volume RAID 1 para gardar datos de xeito seguro
 - ◆ 4.1 Preparar dous discos duros
 - ◆ 4.2 Crear o array
 - ◆ 4.3 Crear e montar o sistema de arquivos
- 5 Reparación da avaría dun disco duro do RAID 1
- 6 Eliminar un dispositivo RAID
- 7 Manuais e enlaces interesantes

1.2 Introducción

Nesta práctica crearemos volumes RAID por software empregando un sistema operativo linux.

1.3 Software necesario

Para a creación e administración dun RAID por software necesitaremos o paquete `mdadm`.

A antiga colección de utilidades para RAID dos paquetes `raidtools` e `raidtools2` deixouse de empregar actualmente posto que dependía dun ficheiro de configuración (`/etc/raidtab`) difícil de manter, e as súas funcións eran limitadas. Dende agosto de 2001, existe a ferramenta `mdadm` (*multiple devices admin*), este paquete permítenos xestionar os RAIDs por software dun xeito moito mais simple e robusto. Actualmente é o estándar.

1.4 Instalación

Primeiramente, poñémonos como `root`:

```
$ sudo su
```

Logo, actualizamos o sistema operativo:

```
$ apt-get update
$ apt-get upgrade
```

Instalamos `mdadm`:

```
$ sudo apt-get install mdadm
```

1.5 Creación dun volume RAID 1 para gardar datos de xeito seguro

Crear dous discos duros virtuais que se empregarán para facer un RAID 1. Nas explicacións seguintes imos supoñer que os discos creados son `sdb` e `sdz`.

1.5.1 Preparar dous discos duros

Antes de crear o RAID precisamos preparar as particións a empregar polo `array`. Isto podería facerse cunha ferramenta gráfica tipo `Gparted` ou empregando unha aplicación de terminal. Para ver os discos/particións existentes neste intre:

```
$ sudo fdisk -l
# Isto terá unha saída parecida á seguinte para cada un dos discos existentes no equipo:
Disk /dev/sdb: 1500.3 GB, 1500301910016 bytes
255 heads, 63 sectors/track, 182401 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x000f1f05
```

Para cada un dos discos que empreguemos para o array debemos crear unha partición que se marcará como "partición RAID" (imos supoñer que os discos duros cós que faremos o RAID serán `sdb` e `sdcl`):

```
$ fdisk /dev/sdb
# Isto executará a ferramenta fdisk.
# Crear unha partición en cada disco que ocupe o espazo completo de cada un deles:
Command (m for help): n
Command action
e      extended
p      primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-182401, default 1): [blank]
Last cylinder or +size or +sizeM or +sizeK (1-182401, default 182401): [blank]
# Elixindo os valores por defecto elixirase o primeiro e o último cilindro,
# así a partición ocupará todo o espazo do disco.
# O seguinte será marcar a partición como parte dun array RAID:
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid auto)
# Agora hai que escribir todos estes cambios no disco:
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Recorda facer o mesmo no outro disco **sdcl**, isto podes facelo con `fdisk` ou empregando a aplicación `sfdisk` do seguinte xeito:

```
$ sfdisk -d /dev/sdb | sfdisk /dev/sdcl
```

Nota: Se empregas o sistema operativo Debian 9 ou superior non é necesario realizar as particións, vale con dispositivos completos. O importante é que teñan o mesmo tamaño. Incluso podemos "misturar" particións con dispositivos.

1.5.2 Crear o array

Para crear o array, empregárase `mdadm`. Débese especificar o nivel de RAID, así como cantos discos vai a empregar.

```
$ mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdcl
```

No modo *verbose* a aplicación dará información extra.

No comando anterior o comando creou o array RAID 1 en `/dev/md0`, usando para iso 2 particións, neste caso `/dev/sdb1` e `/dev/sdcl`.

Mentres o array está construíndose pódese ver o seu estado no arquivo `/proc/mdstat`.

```
$ watch cat /proc/mdstat
```

Este comando *watch* executa, por defecto cada 2 segundos, o comando seguinte que neste caso é `cat /proc/mdstat`, polo que imos vendo a secuencia da creación do array.

Agora que está feito o array, precisamos editar a configuración de `mdadm` para que se volva a montar cando se reinicie o sistema.

```
$ nano /etc/mdadm/mdadm.conf
# Este sería un 'mdadm.conf' calquera, onde aparece a liña que temos que engadir.

# by default, scan all partitions (/proc/partitions) for MD superblocks.
# alternatively, specify devices to scan, using wildcards if desired.
DEVICE partitions

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST

# instruct the monitoring daemon where to send mail alerts
MAILADDR root

# definitions of existing MD arrays
ARRAY /dev/md0 level=1 num-devices=2 UUID=31283299:0c118170:8b4eb9a8:4e935bf2
```

Pero ollo, para coñecer o contido exacto desta liña deberemos executar o comando que aparece a continuación.

```
$ mdadm --detail --scan
```

Para engadir automaticamente a liña que define o array ao ficheiro `/etc/mdadm/mdadm.conf`, hai que executar o seguinte comando:

```
$ mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Olo!! - En debian 9 hai que executar o comando `update-initramfs` para actualizar o *initial RAM disk*:

```
$ update-initramfs -u
```

- [Que é initramfs?](#)
- [Modificación do initramfs](#)

◊ [update-initramfs](#)

1.5.3 Crear e montar o sistema de arquivos

Agora que o array está feito é necesario formatealo. Empregaremos o sistema de arquivos **ext3**.

```
# mkfs.ext3 /dev/md0
```

Isto levaralle un tempo.

O seguinte a facer é engadir o array ao ***fstab***, que fará que se monte automaticamente cando o sistema arranxa. Isto pode facerse editando o arquivo `/etc/fstab`.

```
$ mkdir /media/raid
$ nano /etc/fstab
# O fstab debería conter varias entradas e, ao final debemos engadir esta liña:
/dev/md0          /media/raid      ext3              @@defaults
```

Agora, móntase o array:

```
$ mount -a
```

1.6 Reparación da avaría dun disco duro do RAID 1

Apagamos a máquina virtual e simulamos a avaría dun dos discos pertencentes ao RAID 1 desconectándoo da máquina.

Cando iniciamos o equipo hai que activar o volume `/dev/md0` que non se activa cando se lle avaría un disco. Empregarase o comando:

```
$ mdadm --run /dev/md0
$ mount -a
```

Para ver cómo están os volumes empregamos o comando:

```
$ mdadm --detail --scan
$ mdadm --detail /dev/md0
```

Particionamos o disco duro novo que aparece como **sdc** exactamente igual que o que quedou do RAID 1 que é o **sdb** (en Debian 9 xa non é necesario este paso, só precisamos un HD do mesmo tamaño):

```
$ sfdisk -d /dev/el que está bien | sfdisk /dev/sdc(repuesto)
```

Podemos ver que quedou ben particionado empregando o comando:

```
$ fdisk -l
```

Logo engádese a partición do novo disco ao array:

```
$ mdadm --manage /dev/md0 --add /dev/sdc1(REPUESTO)
```

Unha vez engadida a reconstrución do volume comeza automaticamente, podemos ver como vai a reconstrución:

```
$ watch cat /proc/mdstat
```

Por último, hai que montar as particións do volume para poder traballar con el e ler outra vez os arquivos existentes nel:

```
# mount -a
```

1.7 Eliminar un dispositivo RAID

Para eliminar un dispositivo RAID, primeiro hai que desactivalo executando o seguinte comando como root.

```
$ mdadm --stop /dev/md0
```

Unha vez desactivado, hai que eliminalo:

```
$ mdadm --remove /dev/md0
```

Recordade eliminar:

- A liña engadida en */etc/mdadm/mdadm.conf*
- A liña engadida en */etc/fstab*
- A carpeta */mnt/raid*

Finalmente, sobreescribir con zeros o superbloque md en todos os dispositivos asociados ao array:

```
$ mdadm --zero-superblock --force /dev/sdb1 /dev/sdc1
```

1.8 Manuais e enlaces interesantes

- [Páxinas man de mdadm](#)
- [Reparar un volume RAID 1](#)
- [Manual de mke2fs](#)
- [Arquivo fstab](#)
- [Manual de mount](#)
- [Linux Raid](#)
- [Calcular velocidade lectura/escritura dun disco:](#)

```
$ hdparm -tT /dev/sda
```