Linux Day 2022

Dai file ai blocchi del disco: strutture e funzionalità

22 ottobre 2022
Giuseppe Sacco <giuseppe@sguazz.it>

Dai file ai blocchi del disco

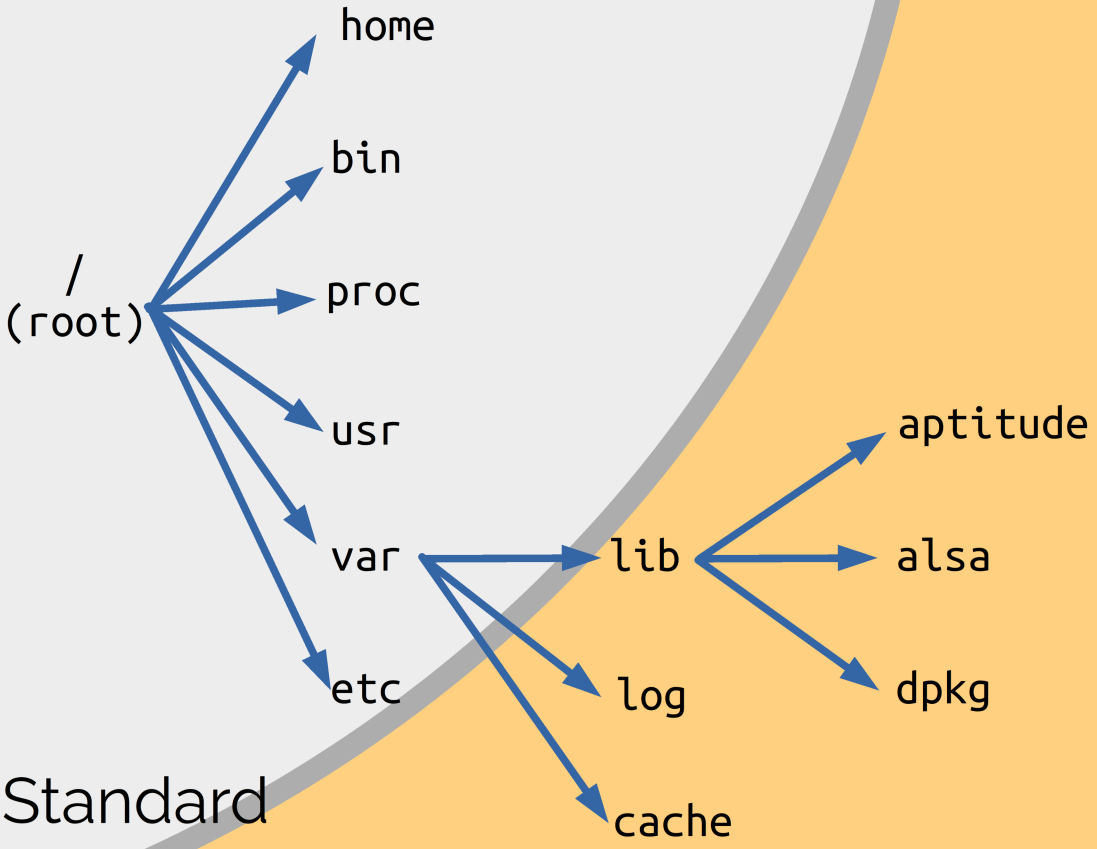
Cosa vedremo:
i file
etichette e partizioni
raid e multidevice
logical volume manager
snapshot
thin pool

Dai file ai blocchi del disco

i file

Struttura gerarchica

La classica gerarchia di directory che compone un *file system*

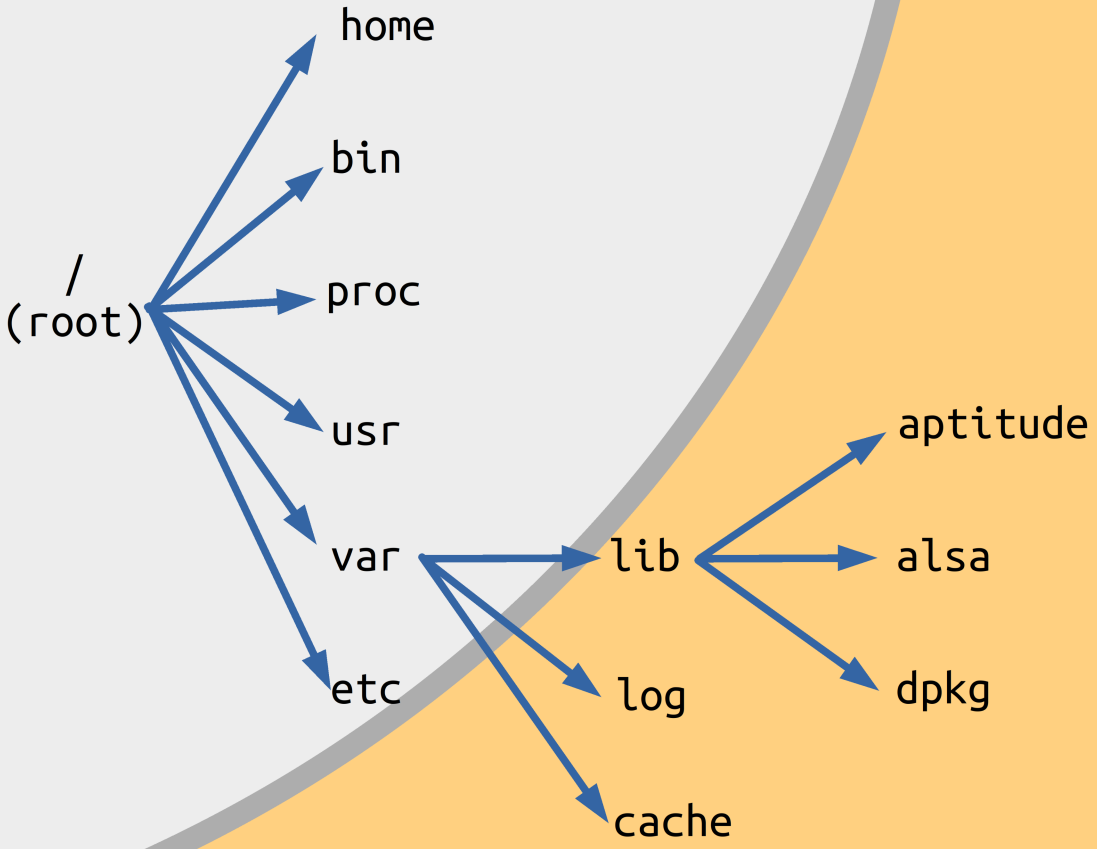


FHS: Filesystem Hierarchy Standard

Tipi di nodi

I nodi di questo albero sono:

- directory
- file normali
- *device* a blocchi
- *link* simbolici
- *named pipe*
- …



Esempi di nodi

Esempi di diversi nodi del *file system*, con attributi:

```
root@linuxday2022:~# ls -ld /run/systemd/private /etc/alternatives/editor /etc /dev/vda /dev/tty
/bin/dash
-rwxr-xr-x   1 root root 125560 10 dic  2020 /bin/dash
crw-rw-rw-   1 root tty    5, 0 21 ott 21.01 /dev/tty
brw-rw----   1 root disk 254, 0 21 ott 12.57 /dev/vda
drwxr-xr-x 131 root root  12288 21 ott 19.30 /etc
lrwxrwxrwx   1 root root      9 10 set 12.53 /etc/alternatives/editor -> /bin/nano
srwxrwxrwx   1 root root      0 21 ott 12.57 /run/systemd/private
```

Attributi estesi

Altri attributi (xattr su Ext4):

```
root@linuxday2022:~# lsattr /usr/ 2>/dev/null
-----------I--e------- /usr/sbin
--------------e------- /usr/local
--------------e------- /usr/games
--------------e------- /usr/src
--------------e------- /usr/include
--------------e------- /usr/libx32
--------------e------- /usr/libexec
--------------e------- /usr/lib64
--------------e------- /usr/lib
-----------I--e------- /usr/share
-----------I--e------- /usr/bin
--------------e------- /usr/lib32
```

Accesso Control List

Autorizzazioni complesse (ACL):

```
root@linuxday2022:~# setfacl -m u:giuseppe:rwx testacl
root@linuxday2022:~# ls -l testacl
-rw-rwxr--+ 2 root root 5 21 ott 21.08 testacl
root@linuxday2022:~# getfacl  testacl
# file: testacl
# owner: root
# group: root
user::rw-
user:giuseppe:rwx
group::r--
mask::rwx
other::r--
```

i-node

Gli attributi princiali vengono memorizzati negli i-node

```
root@linuxday2022:~# touch testacl
root@linuxday2022:~# ls -1i testacl
5242889 testacl
root@linuxday2022:~# ln testacl testacl2
root@linuxday2022:~# ls -1i testacl*
5242889 testacl
5242889 testacl2
```

Spazio occupato /1

Lo spazio allocato è a (gruppi di) blocchi:

```
root@linuxday2022:~# touch test
root@linuxday2022:~# ls -l test
-rw-r--r-- 1 root root 0 21 ott 21.02 test
root@linuxday2022:~# stat test
  File: test
  Dim.: 0           Blocchi: 0          Blocco di IO: 4096   file regolare vuoto
[...]
root@linuxday2022:~# du test
0    test
root@linuxday2022:~# echo -n testo > test
root@linuxday2022:~# ls -l test
-rw-r--r-- 1 root root 5 21 ott 21.03 test
root@linuxday2022:~# stat test
  File: test
  Dim.: 5           Blocchi: 8          Blocco di IO: 4096   file regolare
[...]
root@linuxday2022:~# du -k test
4    test
root@linuxday2022:~# du -b test
5    test
```

Spazio occupato /2

Lo spazio a volte NON è allocato ma solo promesso:

```
root@linuxday2022:/mnt# truncate --size 1G test
root@linuxday2022:/mnt# ls -l test
-rw-r--r-- 1 root root 1073741824 21 ott 22.05 test
root@linuxday2022:/mnt# du -k test
0    test
root@linuxday2022:/mnt# stat test
  File: test
  Dim.: 1073741824    Blocchi: 0          Blocco di IO: 4096   file regolare
[...]
root@linuxday2022:/mnt# df -k .
File system    1K-blocchi  Usati Disponib. Uso% Montato su
/dev/vdb1       104805380 763752 104041628   1% /mnt
```

Spazio occupato /3

Lo spazio a volte NON è allocato se non necessario:

```
root@linuxday2022:/mnt# for i in $(seq 1 999999999); do echo “Breve prologo”; done > test3
root@linuxday2022:/mnt# df -k .
File system    1K-blocchi    Usati Disponib. Uso% Montato su
/dev/vdb1       104805380 18336732  86468648  18% /mnt
root@linuxday2022:/mnt# ls -l test3
-rw-r--r-- 1 root root 17996400180 22 ott 01.23 test3
root@linuxday2022:/mnt# cp --reflink test3 test3-copia
root@linuxday2022:/mnt# ls -l
totale 35149324
-rw-r--r-- 1 root root 17996400180 22 ott 01.23 test3
-rw-r--r-- 1 root root 17996400180 22 ott 01.25 test3-copia
root@linuxday2022:/mnt# du -sk *
17574708    test3
17574616    test3-copia
root@linuxday2022:/mnt# df -k .
File system    1K-blocchi    Usati Disponib. Uso% Montato su
/dev/vdb1       104805380 18336736  86468644  18% /mnt
root@linuxday2022:/mnt# dd if=/dev/zero of=test3 conv=notrunc bs=512 count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5,1 MB, 4,9 MiB) copied, 0,121072 s, 42,3 MB/s
root@linuxday2022:/mnt# df -k .
File system    1K-blocchi    Usati Disponib. Uso% Montato su
/dev/vdb1       104805380 18341800  86463580  18% /mnt
```

Copy on write!

Dai file ai blocchi del disco

Cosa abbiamo visto:
Il file system è un insieme gerarchi di nomi
I nodi possono essere file con dati, device, altro
I file usano spazio disco in maniera variegata

Dai file ai blocchi del disco

etichette e partizioni

Tabelle delle partizioni

L'etichetta del disco è il contenitore della tabella delle partizioni. Sono molto diffuse MBR e GPT.

Master boot record

```
root@linuxday2022:~# fdisk -l /dev/vda
Disk /dev/vda: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x374c1c9e

Dispositivo Avvio   Start      Fine   Settori  Size Id Tipo
/dev/vda1            2048   2099199   2097152    1G 83 Linux
/dev/vda2         2099200   3147775   1048576  512M 82 Linux swap / Solaris
/dev/vda3         3147776 209712509 206564734 98,5G 83 Linux
```

GUID Partition Table

```
giuseppe@uefi:~# fdisk -l /dev/sda
Disk /dev/sda: 238,47 GiB, 256060514304 bytes, 500118192 sectors
Disk model: LITEONIT LCM-256
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: C9635C3C-CEDB-4B28-8A0C-3EDA9F14277B

Device       Start       End   Sectors   Size Type
/dev/sda1     2048    249855    247808   121M EFI System
/dev/sda2   249856    737279    487424   238M Linux filesystem
/dev/sda3   737280 500117503 499380224 238,1G Linux LVM
```

MBR

- permette solo 4 partizioni primarie
- viene memorizzato nel primo settore del disco
- è conosciuto dal BIOS di quasi ogni computer
- contiene un eseguibile per l'avvio del computer
- non gestisce lo spazio oltre i 2TB

```
                                        Disk: /dev/vdb
                        Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                                 Label: dos, identifier: 0x03c0db26

    Dispositivo        Avvio                Start           Fine         Settori       Size       Id Tipo
    /dev/vdb1                                2048        20973567        20971520        10G       83 Linux
>>  /dev/vdb2                            20973568        41945087        20971520        10G       83 Linux
    /dev/vdb3                            41945088        62916607        20971520        10G       83 Linux
    /dev/vdb4                            62916608       209715199       146798592        70G        5 Esteso
    ├─/dev/vdb5                          62918656        83890175        20971520        10G       83 Linux
    ├─/dev/vdb6                          83892224       104863743        20971520        10G       83 Linux
    ├─/dev/vdb7                         104865792       125837311        20971520        10G       83 Linux
    ├─/dev/vdb8                         125839360       146810879        20971520        10G       83 Linux
    ├─/dev/vdb9                         146812928       167784447        20971520        10G       83 Linux
    ├─/dev/vdb10                        167786496       188758015        20971520        10G       83 Linux
    └─/dev/vdb11                        188760064       209715199        20955136        10G       83 Linux

 Partition type: Linux (83)

        [Avviabile]  [ Elimina ]  [  Resize ]  [   Esci  ]  [   Tipo  ]  [  Guida  ]  [  Scrivi ]  [   Dump  ]
```

GPT

- viene memorizzato all'inizio e alla fine del disco
- è conosciuto dal BIOS dei computer moderni
- con EFI contiene tanti eseguibili per l'avvio, anche di diversa architettura
- gestisce lo spazio oltre i 2TB (e utilizza LBA anziché CHS)
- ha un *checksum*

```
                                        Disk: /dev/vdb
                                Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                            Label: gpt, identifier: 2C70F1BA-CCBA-1249-8CD9-3DAD52E15472

    Dispositivo                     Start              Fine          Settori          Size Tipo
    /dev/vdb1                        2048           20973567         20971520           10G Linux filesystem
    /dev/vdb2                    20973568           41945087         20971520           10G Linux filesystem
    /dev/vdb3                    41945088           62916607         20971520           10G Linux filesystem
    /dev/vdb4                    62916608           83888127         20971520           10G Linux filesystem
    /dev/vdb5                    83888128          104859647         20971520           10G Linux filesystem
    /dev/vdb6                   104859648          125831167         20971520           10G Linux filesystem
    /dev/vdb7                   125831168          146802687         20971520           10G Linux filesystem
    /dev/vdb8                   146802688          167774207         20971520           10G Linux filesystem
    /dev/vdb9                   167774208          188745727         20971520           10G Linux filesystem
>>  /dev/vdb10                  188745728          209715166         20969439           10G Linux filesystem

  Partition UUID: A1C5D80A-F0FE-5544-9143-B5B17F7050EC
  Partition type: Linux filesystem (0FC63DAF-8483-4772-8E79-3D69D8477DE4)

              [ Elimina ]  [  Resize ]  [   Esci  ]  [   Tipo  ]  [  Guida  ]  [  Scrivi ]  [   Dump  ]
```

Partizioni per un PC

- viene usato subito l'intero disco
- lo *swap* è utile per spostarci piccole parti di dati non usati
- lo *swap* può essere usato per l'ibernazione
- le partizioni potrebbero essere cifrate
- non sono necessari ridondanza e *snapshot*
- partizionamento GPT con avvio tramite EFI

```
                                        Disk: /dev/vdc
                        Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                     Label: gpt, identifier: 8B262FDE-25B9-EE4B-91D7-D6FAFCDBDD57

    Dispositivo                     Start              Fine           Settori          Size Tipo
    /dev/vdc1                        2048            264191            262144          128M EFI System
    /dev/vdc2                      264192         207882239         207618048           99G Linux filesystem
>>  /dev/vdc3                   207882240         209713151           1830912          894M Linux swap
```

Partizioni per un server /1

- viene usato subito solo una parte del disco
- lo *swap* è grande se ci sono molti utenti
- si usano partizioni diverse per dati diversi
- sono necessari ridondanza e *snapshot*
- partizionamento GPT per gestire oltre 2TB

```
                                    Disk: /dev/vdc
                     Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                  Label: gpt, identifier: 8B262FDE-25B9-EE4B-91D7-D6FAFCDBDD57

    Dispositivo                Start               Fine            Settori          Size Tipo
    /dev/vdc1                   2048             264191             262144          128M EFI System
    /dev/vdc2                 264192           21235711           20971520           10G Linux swap
>>  /dev/vdc3                21235712          209713151          188477440         89,9G Linux RAID
```

Dai file ai blocchi del disco

Cosa abbiamo visto:
i dischi devono avere una tabella delle partizioni
le partizioni sono di tipo diverso
l'etichetta entra in gioco all'avio del computer
diversi schemi di partizionamento

Dai file ai blocchi del disco

raid e multidevice

multidevice

- fa parte del kernel di Linux
- gestisce tutti i tipi di raid (1, 4, 5, 6 e 10)
- viene detto **md** e non va confuso con **dm** (*device mapper*)
- include un sistema di controllo che manda un email in caso di guasto
- permette dinamicamente di sostituire dischi guasti
- permette di aggiungere dischi ad *array* già configurati

Esempio multidevice /1

1. definire su ogni disco una partizione di tipo RAID

```
                                                Disk: /dev/vdc
                                Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                                         Label: dos, identifier: 0x3958fed5

    Dispositivo      Avvio          Start          Fine       Settori      Size     Id Tipo
>>  /dev/vdc1                        2048     209715199     209713152      100G     fd
Autorilevamento raid di Linux
```

Esempio multidevice /2

2. creare il raid (1, mirror)

```
root@linuxday2022:~# mdadm --create /dev/md/mirror --level 1 --raid-devices 2 /dev/vdb1
/dev/vdc1
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? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/mirror started.
```

Esempio multidevice /3

3. controllare lo stato del multidisk

```
root@linuxday2022:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : active raid1 vdc1[1] vdb1[0]
      104790016 blocks super 1.2 [2/2] [UU]
      [>....................]  resync =  1.4% (1556480/104790016) finish=71.9min
speed=23900K/sec

unused devices: <none>
```

Esempio multidevice /4

4. creare il file system e montarlo

```
root@linuxday2022:~# mkfs -t xfs /dev/md/mirror
meta-data=/dev/md/mirror         isize=512    agcount=4, agsize=6549376 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=0
data     =                       bsize=4096   blocks=26197504, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=12791, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
Discarding blocks...Done.
root@linuxday2022:~# mount /dev/md/mirror /mnt
```

Esempio multidevice /5

5. sostituire un disco

```
root@linuxday2022:~# mdadm /dev/md/mirror -f /dev/vdc1 -r /dev/vdc1
mdadm: set /dev/vdc1 faulty in /dev/md/mirror
mdadm: hot removed /dev/vdc1 from /dev/md/mirror
root@linuxday2022:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : active raid1 vdb1[0]
      104790016 blocks super 1.2 [2/1] [U_]
root@linuxday2022:~# mdadm /dev/md/mirror -a /dev/vdc1
mdadm: added /dev/vdc1
root@linuxday2022:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md127 : active raid1 vdc1[2] vdb1[0]
      104790016 blocks super 1.2 [2/1] [U_]
      [>....................]  recovery =  0.1% (113216/104790016) finish=15.4min speed=113216K/sec
```

Esempio multidevice /6

6. distruggere il raid

```
root@linuxday2022:~# umount /mnt
root@linuxday2022:~# mdadm --stop /dev/md/mirror
mdadm: stopped /dev/md/mirror
root@linuxday2022:~# mdadm --zero-superblock /dev/vdb1 /dev/vdc1
```

Dai file ai blocchi del disco

Cosa abbiamo visto:
i dischi si possono gestire a gruppi
i file system usano i raid come fossero un solo disco

Dai file ai blocchi del disco

logical volume manager

LVM /1

1. definire su ogni disco una partizione di tipo LVM

```
                                                Disk: /dev/vdb
                                   Size: 100 GiB, 107374182400 bytes, 209715200 sectors
                                            Label: dos, identifier: 0xe480784d

    Dispositivo        Avvio               Start             Fine         Settori        Size       Id Tipo
>>  /dev/vdb1                               2048        209715199       209713152        100G       8e Linux LVM
```

LVM /2

2. definire volumi fisici e gruppo di volumi

```
root@linuxday2022:~# pvcreate /dev/vdb1 /dev/vdc1
  Physical volume "/dev/vdb1" successfully created.
  Physical volume "/dev/vdc1" successfully created.
root@linuxday2022:~# vgcreate vg /dev/vdb1 /dev/vdc1
  Volume group "vg" successfully created
root@linuxday2022:~# vgdisplay vg
  --- Volume group ---
  VG Name               vg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               199,99 GiB
  PE Size               4,00 MiB
  Total PE              51198
  Alloc PE / Size       0 / 0
  Free  PE / Size       51198 / 199,99 GiB
  VG UUID               jCBYtz-yO6i-NFLL-loej-5Jsi-hFn8-A7UhV6
```

LVM /3

3. definire un volume logico in mirror

```
root@linuxday2022:~# lvcreate --type mirror --mirrors 1 --size 5G --name lv1 vg
  Logical volume "lv1" created.
root@linuxday2022:~# lvdisplay /dev/vg/lv1
  --- Logical volume ---
  LV Path                /dev/vg/lv1
  LV Name                lv1
  VG Name                vg
  LV UUID                tocbgq-1ac1-2uhu-0UQS-KaCz-QjzI-1jMpb9
  LV Write Access        read/write
  LV Creation host, time linuxday2022, 2022-10-22 08:38:14 +0200
  LV Status              available
  # open                 0
  LV Size                5,00 GiB
  Current LE             1280
  Mirrored volumes       2
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:3
```

LVM /4

4. definire un volume logico semplice

```
root@linuxday2022:~# lvcreate --size 5G --name lv2 vg
  Logical volume "lv2" created.
root@linuxday2022:~# lvdisplay /dev/vg/lv2
  --- Logical volume ---
  LV Path                /dev/vg/lv2
  LV Name                lv2
  VG Name                vg
  LV UUID                bfBSo3-Utyk-DUI2-953h-8kOH-fDCe-sQ5xLg
  LV Write Access        read/write
  LV Creation host, time linuxday2022, 2022-10-22 08:49:53 +0200
  LV Status              available
  # open                 0
  LV Size                5,00 GiB
  Current LE             1280
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           252:4
```

LVM /5

5. creare i file system e usarli

```
root@linuxday2022:~# ls -l /dev/mapper/
totale 0
crw------- 1 root root 10, 236 22 ott 08.37 control
lrwxrwxrwx 1 root root       7 22 ott 08.48 vg-lv1 -> ../dm-3
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mimage_0 -> ../dm-1
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mimage_1 -> ../dm-2
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mlog -> ../dm-0
lrwxrwxrwx 1 root root       7 22 ott 08.49 vg-lv2 -> ../dm-4
root@linuxday2022:~# mkfs -t xfs /dev/mapper/vg-lv1
root@linuxday2022:~# mkfs -t ext4 /dev/mapper/vg-lv2
root@linuxday2022:~# mkdir /mnt2
root@linuxday2022:~# mount /dev/mapper/vg-lv1 /mnt
root@linuxday2022:~# mount /dev/mapper/vg-lv2 /mnt2
```

LVM e device mapper /1

LVM è implementato tramite il *device mapper* (**dm**) del kernel che mappa i vari pezzi dei volumi logici sulle porzioni di disco

```
root@linuxday2022:~# ls -l /dev/mapper/
totale 0
crw------- 1 root root 10, 236 22 ott 08.37 control
lrwxrwxrwx 1 root root       7 22 ott 08.48 vg-lv1 -> ../dm-3
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mimage_0 -> ../dm-1
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mimage_1 -> ../dm-2
lrwxrwxrwx 1 root root       7 22 ott 08.43 vg-lv1_mlog -> ../dm-0
lrwxrwxrwx 1 root root       7 22 ott 08.49 vg-lv2 -> ../dm-4
root@linuxday2022:~# dmsetup status
vg-lv1: 0 10485760 mirror 2 252:1 252:2 2560/2560 1 AA 3 disk 252:0 A
vg-lv1_mimage_0: 0 10485760 linear
vg-lv1_mimage_1: 0 10485760 linear
vg-lv1_mlog: 0 8192 linear
vg-lv2: 0 10485760 linear
```

LVM e device mapper /2

Se si estende un volume logico e viene allocata una parte del disco non contigua a quella già allocata allora **dm** aggiunge una mappatura

```
root@linuxday2022:~# lvextend --size +1G /dev/vg/lv1
  Extending 2 mirror images.
  Size of logical volume vg/lv1 changed from 5,00 GiB (1280 extents) to 6,00 GiB (1536 extents).
  Logical volume vg/lv1 successfully resized.
root@linuxday2022:~# dmsetup status
vg-lv1: 0 12582912 mirror 2 252:1 252:2 2575/3072 1 AA 3 disk 252:0 A
vg-lv1_mimage_0: 0 10485760 linear
vg-lv1_mimage_0: 10485760 2097152 linear
vg-lv1_mimage_1: 0 10485760 linear
vg-lv1_mimage_1: 10485760 2097152 linear
vg-lv1_mlog: 0 8192 linear
vg-lv2: 0 10485760 linear
```

LVM e device mapper /3

Se si estende un volume logico e viene allocata una parte del disco contigua a quella già allocata allora **dm** non aggiunge una mappatura

```
root@linuxday2022:~# lvextend --size +1G /dev/vg/lv1
  Extending 2 mirror images.
  Size of logical volume vg/lv1 changed from 6,00 GiB (1536 extents) to 7,00 GiB (1792 extents).
  Logical volume vg/lv1 successfully resized.
root@linuxday2022:~# dmsetup status
vg-lv1: 0 14680064 mirror 2 252:1 252:2 3089/3584 1 AA 3 disk 252:0 A
vg-lv1_mimage_0: 0 10485760 linear
vg-lv1_mimage_0: 10485760 4194304 linear
vg-lv1_mimage_1: 0 10485760 linear
vg-lv1_mimage_1: 10485760 4194304 linear
vg-lv1_mlog: 0 8192 linear
vg-lv2: 0 10485760 linear
```

LVM e device mapper /4

Si può cambiare tipo di raid ad un singolo volume logico

```
root@linuxday2022:~# pvcreate /dev/vdd1 /dev/vde1
  Physical volume "/dev/vdd1" successfully created.
  Physical volume "/dev/vde1" successfully created.
root@linuxday2022:~# vgextend vg /dev/vdd1 /dev/vde1
  Volume group "vg" successfully extended
oot@linuxday2022:~# lvconvert --type raid5 /dev/vg/lv2
  Using default stripesize 64,00 KiB.
  Replaced LV type raid5 (same as raid5_ls) with possible type raid1.
  Repeat this command to convert to raid5 after an interim conversion has finished.
Are you sure you want to convert linear LV vg/lv2 to raid1 type? [y/n]: y
  Logical volume vg/lv2 successfully converted.
root@linuxday2022:~# lvs
  LV   VG Attr       LSize Pool Origin Data%  Meta%  Move Log        Cpy%Sync Convert
  lv1  vg mwi-aom--- 7,00g                                [lv1_mlog] 100,00
  lv2  vg rwi-a-r--- 5,00g                                           54,52
root@linuxday2022:~# lvconvert --type raid5 /dev/vg/lv2
  Using default stripesize 64,00 KiB.
  --stripes not allowed for LV vg/lv2 when converting from raid1 to raid5.
Are you sure you want to convert raid1 LV vg/lv2 to raid5 type? [y/n]: y
  Logical volume vg/lv2 successfully converted.
```

LVM e device mapper /5

E il device mapper cambierà di conseguenza

```
root@linuxday2022:~# dmsetup status
vg-lv1: 0 14680064 mirror 2 252:1 252:2 3584/3584 1 AA 3 disk 252:0 A
vg-lv1_mimage_0: 0 10485760 linear
vg-lv1_mimage_0: 10485760 4194304 linear
vg-lv1_mimage_1: 0 10485760 linear
vg-lv1_mimage_1: 10485760 4194304 linear
vg-lv1_mlog: 0 8192 linear
vg-lv2: 0 10485760 raid raid5_ls 2 AA 10485760/10485760 idle 0 0 -
vg-lv2_rimage_0: 0 10485760 linear
vg-lv2_rimage_1: 0 10485760 linear
vg-lv2_rmeta_0: 0 8192 linear
vg-lv2_rmeta_1: 0 8192 linear
```

Dai file ai blocchi del disco

Cosa abbiamo visto:
si possono gestire volumi logici
i volumi possono essere di tipo diverso
il kernel implementa il device mapper

Dai file ai blocchi del disco

snapshot

Snapshot /1

Lo snapshot è la foto istantanea di un file system. Viene “scattata”, ad esempio, per fare un backup che richiede del tempo senza avere modifiche ai dati da copiare

Snapshot /2

Lo *snapshot* alloca una seconda porzione del disco nel quale vengono scritte le modifiche apportate al *file system* originale durante la copia. In alcuni sistemi, come LVM, funziona al contrario: nella nuova porzione di disco viene messo il dato vecchio mentre il nuovo va sul disco principale

Snapshot /2

Facciamo lo snapshot di un file system XFS su LVM

```
root@linuxday2022:~# touch /mnt/file-pre-snapshot
root@linuxday2022:~# ls -l /mnt
root@linuxday2022:~# lvcreate --snapshot --name lv1backup --size 1G /dev/vg/lv1
root@linuxday2022:~# mkdir /mnt3
root@linuxday2022:~# mount -o nouuid,ro /dev/vg/lv1backup /mnt3
root@linuxday2022:~# ls -l /mnt/* /mnt3/*
root@linuxday2022:~# lvs
root@linuxday2022:~# cp /bin/bash /mnt/
root@linuxday2022:~# ls -l /mnt/* /mnt3/*
root@linuxday2022:~# df
root@linuxday2022:~# umount /mnt3
root@linuxday2022:~# lvremove /dev/vg/lv1backup
```

Provare per credere!

Dai file ai blocchi del disco

Cosa abbiamo visto:
gli *snapshot* sono un metodo per duplicare i dati senza duplicare tutto lo spazio disco

Dai file ai blocchi del disco

pool

Pool /1

Un pool è uno spazio disco dal quale vengono allocati solo i blocchi veramente usati. Vi si possono definire volumi molto grandi anche se il disco è inizialmente piccolo

Pool /2

Facciamo un pool su LVM e un volume da 1TB

```
root@linuxday2022:~# lvcreate -L 25G --thinpool tp_pool vg
  Thin pool volume with chunk size 64,00 KiB can address at most 15,81 TiB of data.
  Logical volume "tp_pool" created.
root@linuxday2022:~# lvs
  LV      VG Attr       LSize  Pool Origin Data%  Meta%  Move Log        Cpy%Sync Convert
  lv1     vg mwi-a-m---  7,00g                                [lv1_mlog] 100,00
  lv2     vg rwi-a-r---  5,00g                                           100,00
  tp_pool vg twi-a-tz-- 25,00g             0,00   10,46
root@linuxday2022:~# lvcreate --name lv3 --virtualsize 1T --thinpool tp_pool vg
  Logical volume "lv3" created.
root@linuxday2022:~# lvs
  LV      VG Attr       LSize  Pool    Origin Data%  Meta%  Move Log        Cpy%Sync Convert
  lv1     vg mwi-a-m---  7,00g                                    [lv1_mlog] 100,00
  lv2     vg rwi-a-r---  5,00g                                               100,00
  lv3     vg Vwi-a-tz--  1,00t tp_pool        0,00
  tp_pool vg twi-aotz-- 25,00g                0,00   10,48
root@linuxday2022:~# mkfs -t xfs /dev/vg/lv3
root@linuxday2022:~# mount /dev/vg/lv3 /mnt
root@linuxday2022:~# df -k /mnt
File system        1K-blocchi   Usati  Disponib. Uso% Montato su
/dev/mapper/vg-lv3 1073217536 7517520 1065700016   1% /mnt
```

Provare per credere!

Pool /3

Il pool è stato dimensionato da 25Gb

- può estendersi automaticamente alla bisogna
- può essere configurato per avvisare se non c'è più spazio

Dai file ai blocchi del disco

Cosa abbiamo visto:
i pool detti *thin* sono quelli che allocano lo spazio solo al momento dell'uso
vi si possono definire volumi di dimensione arbitraria

Grazie a tutti

Giuseppe Sacco <giuseppe@sguazz.it>

Trovate questa presentazione su https://www.sguazz.it/~giuseppe/LinuxDay2022.pdf