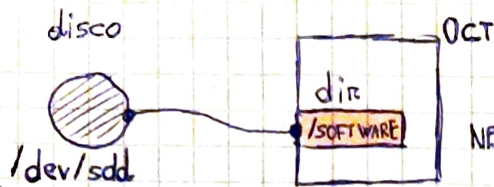
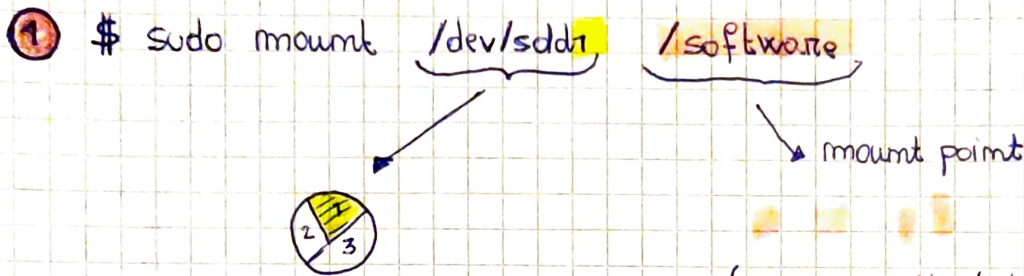


# DISCHI



MONTARE UN DISCO IN UNA DIRECTORY

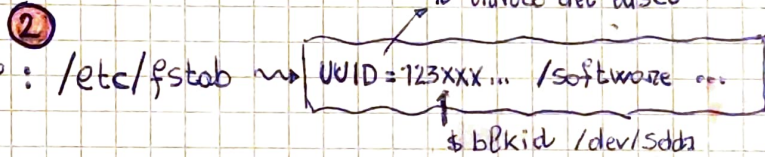
NB: /mnt è un generico punto di montaggio  
Per montaggi permanenti meglio crearne uno dedicato



- L'intero disco è identificato come /dev/sdd, ma per inizializzarlo è necessario partizionarlo in n partizioni (n=1,2,...)

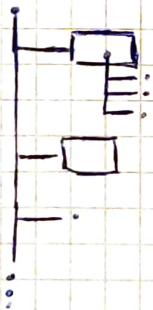
DISCO: /dev/sdd  
PARTIZIONE: /dev/sdd1  
PUNTO DI MONTAGGIO: /software

NB: Se voglio montaggio automatico: /etc/fstab



## FILE SYSTEM

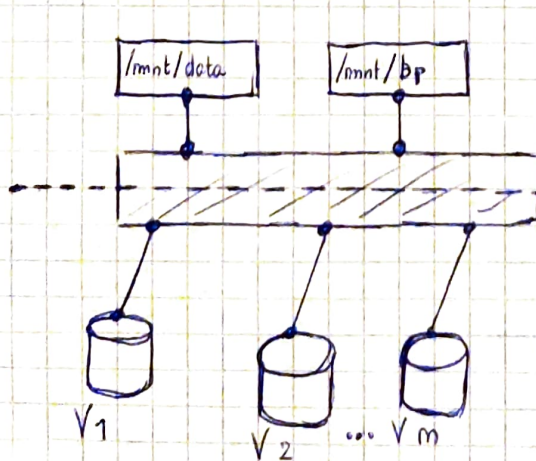
Organizzazione dei file su disco



Tipologie di FS:

- ZFS ...
- XFS per alte prestazioni: server, db, ...
- EXT4 predefinito: desktop, laptop, ...

## VOLUMES



Volumi Logici

$$\text{Volume} = \sum_{i=1}^n V_i$$

astrazione per disaccoppiare.

Volumi fisici  
da aggiungere o togliere a piacimento



## CHECK SPACE

\$ **df -h** (disk free)

- Mounted filesystems

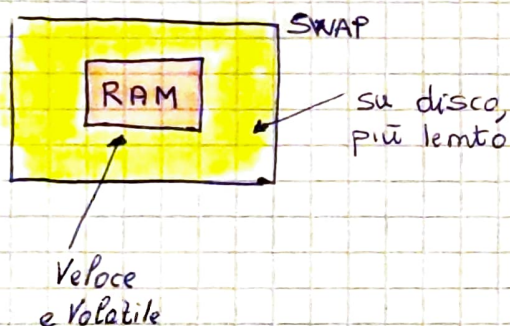
\$ **du -h /var/log** (disk usage)

- Spazio disco di /var/log

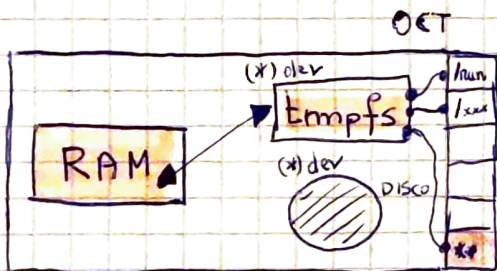
\$ **lsblk** (list block devices)

- elenco gerarchico di dischi e partizioni

## SWAP SPACE (SWAP = scambio / baratto)



**tmpfs**



- filesystem temporaneo che usa la RAM per archiviare i dati. ~~ultra~~ ultra-rapido

• \$ mount | grep tmpfs

```
tmpfs on /run
tmpfs on /dev/shm
...
```

- Posso leggere e scrivere in /tmpfs
- Posso anche crearmi uno spozietto riservato

\$ **mkdir /mnt/mytmpfs** (\*\*)

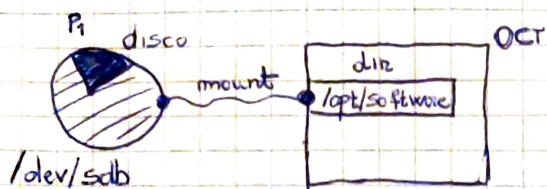
\$ **mount tmpfs -o size=100M tmpfs /mnt/mytmpfs**

\$ **leggo/scrivo** ~ /mnt/mytmpfs

(\*) Sono tutte risorse che necessitano di essere montate (mount)



## COME AGGIUNGERE UN DISCO



### # fdisk -l

Viene assegnato in automatico un nome al nuovo disco  
Con questo comando vedo tutti i dischi attaccati  
Chiamiamolo /dev/sdb

### # fdisk /dev/sdb

Inizializzazione del disco > m > w

### # fdisk -l

Dovrei vedere tutto correttamente, disco e partizione

### # mkfs.xfs /dev/sdb1

Adesso devo creare il filesystem  
Con questo comando formatto la partizione /dev/sdb1 usando XFS.

NB: formattare significa prepararlo per l'archiviazione dei dati, configurandolo con un filesystem specifico.

### # lsblk -f

Vedo i vari dischi con la tipologia di filesystem montato.  
Vedo anche il suo UUID (in alternativa # blkid /dev/sdb1)

### # mkdir /opt/software

Creo una directory che servirà da mount point.

### # mount /dev/sdb1 /opt/software

Montiamo la partizione del disco nella directory

### # mount | grep sdb1

facciamo un check.

- Dobbiamo rendere la mount permanente per fare in modo che sopravviva ai reboot. Quindi devo modificare il /etc/fstab.

### # nano /etc/fstab

/dev/disk/by-uuid/xxx... /opt/software xfs defaults 0 0 6

- Se voglio poi permettere agli utenti di accedere devo modificare i permessi di `/opt/software`.



# LOGICAL VOLUMES

1) Aggiungiamo un disco e poi vediamo che lo ha preso

# fdisk -l

• Supponiamo che si chiami /dev/sdb

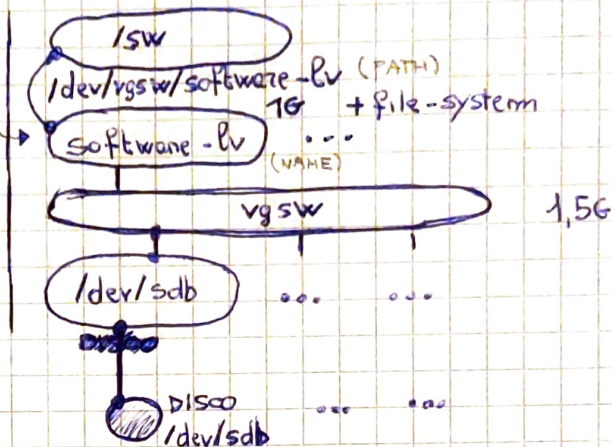
(E) • mount

(D) • filesystems

(C) • logical volumes

(B) • Volume group

(A) • Physical Volumes



2) # lsblk

Identifichiamo il disco che vogliamo convertire in volume logico (sdb)

3) # pvcreate /dev/sdb.

Creiamo il volume fisico (A)

# pvs (PV show)

Vediamo tutti i volumi fisici, tra cui /dev/sdb

# pvdisplay /dev/sdb

Vediamo un po' di dettagli del volume fisico.

4) Creiamo il Volume group (B)

# vgcreate vgsw /dev/sdb

Facciamo poi qualche check con: # vgs (VG show) e

# vgdisplay vgsw

5) Creiamo il logical volume (C)

# lvcreate -L 1G -m software-lv vgsw  
size name

Facciamo qualche check: # lvs , # lvdisplay /dev/vgsw/software-lv

(F)

6) Dobbiamo mettere un file-system sul nostro nuovo volume ①

```
# mkfs.xfs /dev/vgsw/software-lv
```

7) Creiamo un mount point per il nuovo fs che di fatto è come se fosse un disco+fs da montare.

```
# mkdir /sw
```

```
# mount /dev/vgsw/software-lv /sw
```

Tra le mount compaiono come: /dev/mapper/vgsw--software--lv

```
# df -h /sw
```

8) Mappiamo il tutto in /etc/fstab (senza usare uuid)

```
/dev/vgsw/software-lv /sw xfs ..
```