



Sistemi  
Operativi

Bruschi  
Monga Re

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

# Sistemi Operativi<sup>1</sup>

Mattia Monga

Dip. di Informatica  
Università degli Studi di Milano, Italia  
[mattia.monga@unimi.it](mailto:mattia.monga@unimi.it)

a.a. 2018/19

<sup>1</sup> © 2008–18 M. Monga. Creative Commons Attribuzione — Condividi allo stesso modo 4.0 Internazionale. <http://creativecommons.org/licenses/by-sa/4.0/deed.it>. Immagini tratte da [2] e da Wikipedia.



Sistemi  
Operativi

Bruschi  
Monga Re

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

# Lezione XVI: Memoria di massa



Sistemi  
Operativi

Bruschi  
Monga Re

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

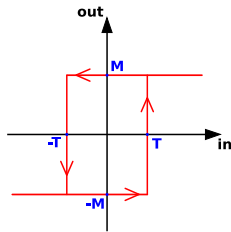
Esercizio

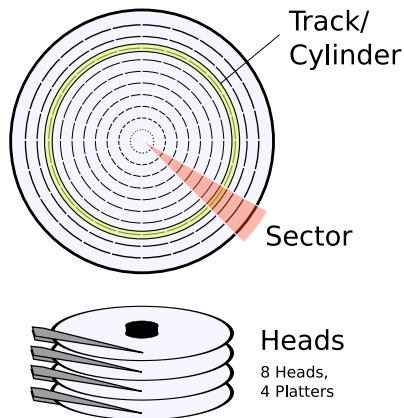
- Trovare la somma delle righe dei file .h con un numero di righe dispari.

Il disco fisso (*hard disk*) è generalmente una memoria magnetica.

Viene sfruttato il fenomeno del *ciclo di isteresi* di elementi magnetici (L'isteresi è la caratteristica di un sistema di reagire in ritardo alle sollecitazioni applicate e in dipendenza dello stato precedente).

Un ciclo di isteresi può essere ottenuto anche elettronicamente (*Schmitt trigger*).  
Le memorie USB, invece, sono basate su transistor NAND.





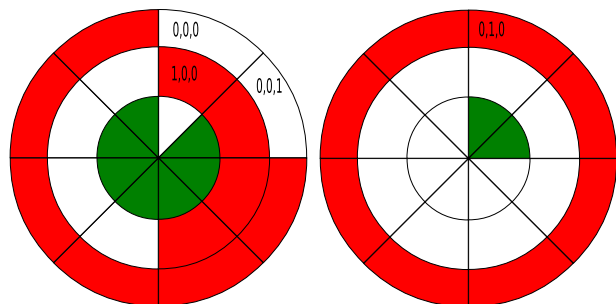
- Gli elementi contenenti dati (blocchi fisici) sono definiti da tre coordinate

- 1 **Cylinder** Il cilindro definito dall'insieme delle tracce corrispondenti dei vari piatti
- 2 **Head** La testina (per esempio, sopra e sotto)
- 3 **Sector** Lo spicchio


$$\begin{aligned} \text{blocksPerPlatterSide} &= \\ &(\text{cylindersPerPlatter}) * (\text{SectorsPerPlatter}) \\ \text{blocksPerPlatter} &= (\text{blocksPerPlatterSide}) * (\text{HeadsPerPlatter}) \\ \text{blocksPerPlatter} &= \\ &(\text{cylindersPerPlatter}) * (\text{SectorsPerPlatter}) * (\text{HeadsPerPlatter}) \\ \text{blocks} &= (\text{Cylinders}) * (\text{Heads}) * (\text{Sectors}) \end{aligned}$$

## Example

Un floppy disk con 80 cilindri, 2 testine, 18 settori  $\rightsquigarrow$  2880



- $C = 3$      $H = 2$      $S = 8$  totale blocchi 48
- zona (partizione) rossa  $0,0,2 \rightsquigarrow 1,0,3$

$$(1*(2*8)+0*8+3*1)-(0*(2*8)+0*8+2*1) = 19-2 = 17$$

In realtà 18 perché contiamo da zero



Sistemi  
Operativi

Bruschi  
Monga Re

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

$$T = \text{TempoDiRotazione} + \text{TempoDiRicerca} + \text{TempoDiAccesso}$$

Il tempo di rotazione è detto anche *latenza*

Il tempo di ricerca (*seek time*) può essere ottimizzato con algoritmi opportuni





## Example

76 124 17 269 201 29 137 12

- First Come First Served
- Shortest Seek First
- Scan/Look (Elevator)

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio



L'astrazione fornita dal s.o. per il disco è quella del **device a blocchi**. Il blocco è un *blocco logico*, potenzialmente diverso dal blocco fisico.

I device a blocchi sono *file speciali*, identificati da

- **Major number**: identifica la categoria del device (disco IDE, floppy)
- **Minor number**: numero d'ordine del device all'interno di una categoria



I file speciali si creano con `/usr/bin/mknod` generalmente in `/dev`

- *Device a blocchi* `b`
- *Device a caratteri* `c`
- *Named pipe* `p` (non ha major e minor)

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio



Lo spazio di memoria di uno hard-disk è ripartito in porzioni indipendenti (**partizioni**): in linea di principio possono contenere anche sistemi differenti. Generalmente contengono sotto-file-system il cui backup e/o aggiornamento è indipendente.

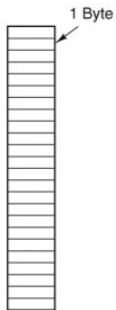
**Partition table sector** Contiene la descrizione di 4 partizioni (primarie) agli offset 446, 462, 478, 494

**Partizione** Una zona *contigua* del disco (CHS)

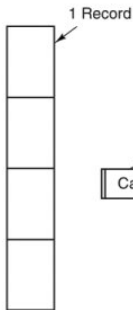
**Partizione estesa** Una partizione che permette una nuova suddivisione (**partizioni logiche**) grazie ad un nuovo PTS



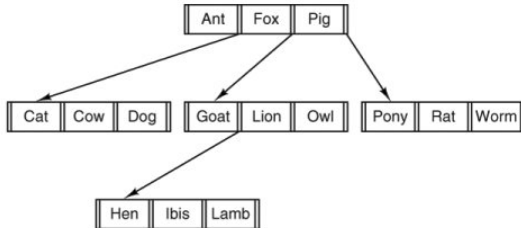
```
struct partition {  
char active;  
char begin[3];  
char type;  
char end[3];  
int start;  
int length;  
};
```



(a)



(b)



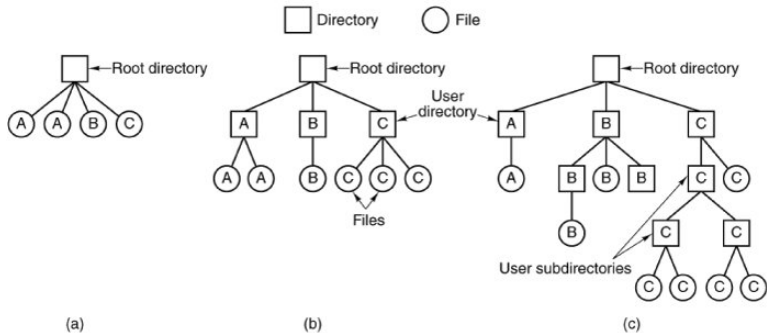
(c)

# Directory



Sistemi Operativi

Bruschi Monga Re



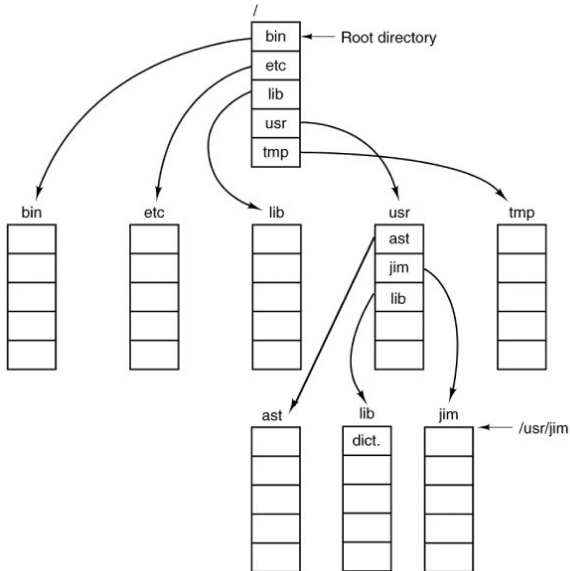
Memorie di massa

Tempi di lettura e scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio





# Disk layout



Sistemi Operativi

Bruschi  
Monga Re

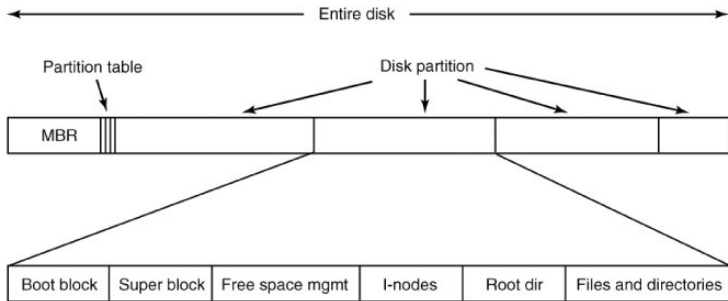
Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

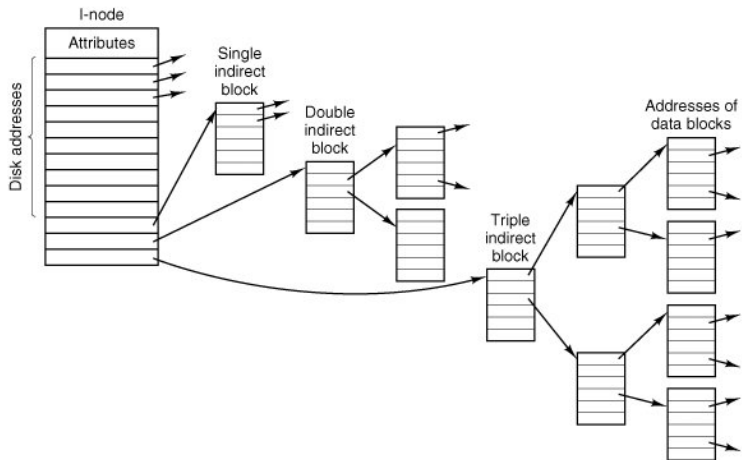




Una **partizione** può essere usata per ospitare un *file system*.

- Un file system va *creato* (`mkfs`)
- Un file system va *montato* (`mount`)
- Corrispondentemente va *smontato* (`umount`)
- Ogni file è caratterizzato da un *i-node* e conosciuto tramite uno o più **link** o nomi (`ln`)

# I-node



Sistemi Operativi

Bruschi Monga Re

Memorie di massa

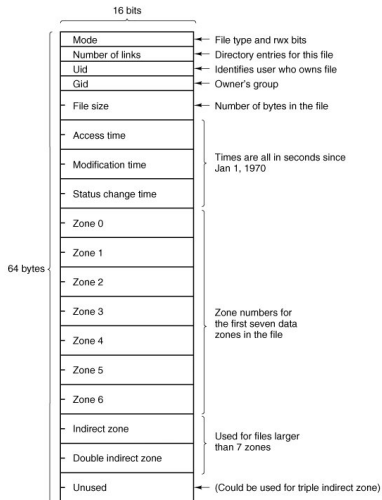
Tempi di lettura e scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

# Un esempio di i-node



Con blocchi da 1KB, zone da 32 bit (default con EXT-2, che però ha 12 zone dirette):

- Qual è la dimensione massima di un file con un solo blocco di overhead?
- Qual è la dimensione massima di un file?
- Quanti blocchi di overhead sono necessari per un file da 100MB di dati?

Sistemi Operativi

Bruschi Monga Re

Memorie di massa

Tempi di lettura e scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

# Aggiungere un disco in QEmu



Sistemi  
Operativi

Bruschi  
Monga Re

Memorie di  
massa

Tempi di lettura e  
scrittura

L'astrazione del s.o.

Dischi virtuali

Esercizio

Fuori dalla macchina virtuale

```
qemu-img create disco.img 100M
```

Poi può essere usato aggiungendo `-hdb disco.img`

In generale (anche al di là delle macchine virtuali) un file può facilmente essere usato come “disco”: i *loop device* servono proprio per utilizzare un file (che è uno stream di caratteri) come *device a blocchi*

```
dd if=/dev/zero of=prova.img seek=10M bs=1 count=0  
/sbin/mkfs.ext2 prova.img
```

```
sudo mount -o loop prova.img /mnt
```

```
echo ciao > /mnt/pippo
```

```
sudo umount prova.img
```



- 1 Creare un disco virtuale
- 2 Partizionare il disco
- 3 Creare il file system
- 4 Montare il file system