

## 5 – Memoria secondaria ottimizzazione delle prestazioni nei dischi

### Sommario

#### Introduzione

#### Evoluzione dei dispositivi di memoria secondaria

#### Caratteristiche dei dischi a testina mobile

#### Arrays Ridondanti di dischi Indipendenti (RAID)

#### Strategie di scheduling del disco

#### First-Come-First-Served (FCFS)

#### Shortest-Seek-Time-First (SSTF)

#### SCAN e varianti: C-SCAN, FSCAN e N-Step SCAN

#### LOOK e C-LOOK

#### Ottimizzazione rotazionale

#### Scheduling SLTF

#### Scheduling SPTF e SATF

#### Considerazioni sul sistema

#### Cache e Buffering del disco

#### Gestione degli errori

#### Software per I/O

#### Altre tecniche di miglioramento delle prestazioni

S. Balsamo – Università Ca' Foscari Venezia – SO.5.0

## Obbiettivi

- Realizzazione delle operazioni di input/output su disco
- Come si completa input/output
- Importanza dell'ottimizzazione delle prestazioni
- Ottimizzare la ricerca (seek) e la rotazione
- Strategie di scheduling del disco
- caching e buffering
- Altre tecniche per migliorar le prestazioni del disco
- Principali schemi per realizzare Array Ridondanti di Dischi Indipendenti (RAID)

S. Balsamo – Università Ca' Foscari Venezia – SO.5.1

## Introduzione

- La memoria secondaria è spesso uno collo di bottiglia
  - Dispositivi di memoria permanente, economico, riscrivibile, di lunga durata
    - Nastri – inadeguati se è richiesto un accesso rapido alle locazioni
    - Dischi – ad accesso 'casuale' (diretto)
    - Evoluzione di costo/prestazioni
    - Vincoli meccanici
  - Altri dispositivi I/O: tastiera, mouse, monitor
  - I miglioramenti delle prestazioni di memoria secondaria aumentano in modo significativo le prestazioni dell'intero sistema
  - Soluzioni possono essere basate sia su software e sia su hardware

S. Balsamo – Università Ca' Foscari Venezia – SO.5.2

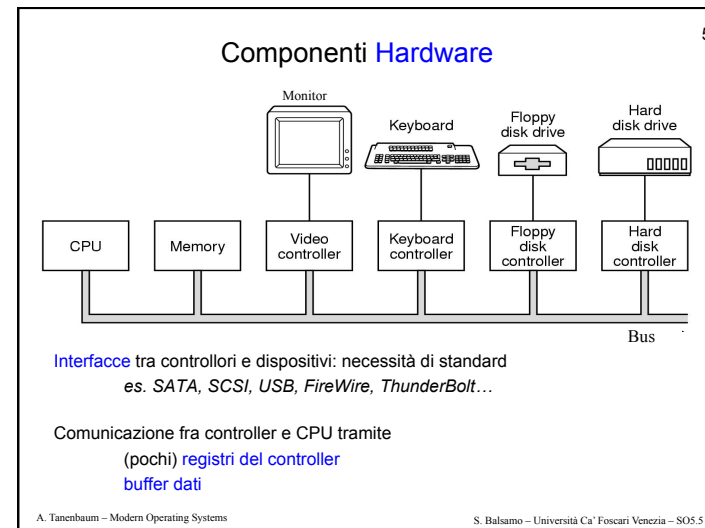
## Introduzione

- Dispositivi di I/O
  - a blocchi – di dimensione fissa
    - es. dischi, penne USB
  - a caratteri
    - es. stampanti, interfacce di rete, mouse
  - Alcune altre categorie
    - es. clock, touch screen
  - Diverse velocità dei dispositivi
  - Controllore del dispositivo o adattatore (componente elettronica)
  - Dispositivo (componente meccanica)

S. Balsamo – Università Ca' Foscari Venezia – SO.5.3

Dispositivo	Velocità di trasferimento dei dati
Tastiera	10 byte/s
Mouse	00 byte/s
Modem a 56 K	7 KB/s
Scanner a 300 dpi	1 MB/s
Videocamera digitale	3,5 MB/s
Disco Blu-ray 4x	18 MB/s
802.11n Wireless	37,5 MB/s
USB 2.0	60 MB/s
FireWire 800	100 MB/s
Gigabit Ethernet	125 MB/s
Disco fisso SATA 3	600 MB/s
USB 3.0	625 MB/s
Bus SCSI Ultra 5	640 MB/s
Bus PCIe 3.0 single lane	985 MB/s
Bus Thunderbolt 2	2,5 GB/s
Rete SONET OC-768	5 GB/s

A. Tanenbaum – Modern Operating Systems S. Balsamo – Università Ca' Foscari Venezia – SO.5.4

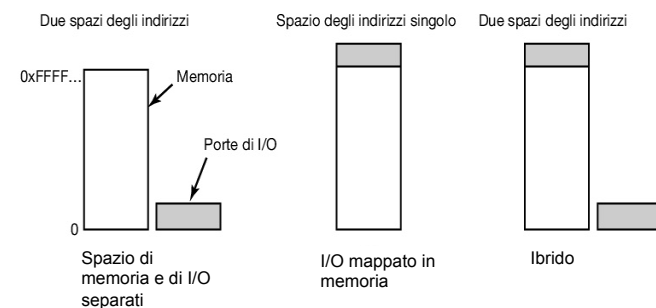


## Comunicazione con I/O

- Comunicazione fra CPU e dispositivi di I/O
  - registri di controllo ai quali è assegnata una porta di I/O (8 o 16 bit)
    - spazio delle porte: insieme delle porte
    - Protezione
  - I/O mappato in memoria
    - ad ogni registro è assegnato uno unico indirizzo di memoria, al quale non è assegnata memoria
  - ibrido
    - Due spazi separati, un buffer dai dati dei dispositivi IO mappati in memoria e porte IO separate per i registri di controllo

S. Balsamo – Università Ca' Foscari Venezia – SO.5.6

## Comunicazione con I/O



A. Tanenbaum – Modern Operating Systems

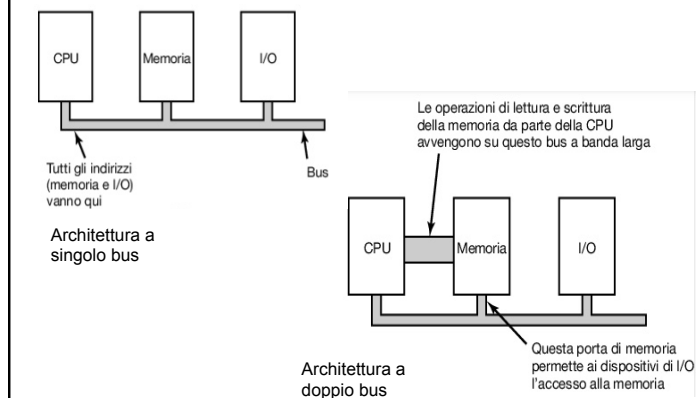
S. Balsamo – Università Ca' Foscari Venezia – SO.5.7

## I/O mappato in memoria

- Vantaggi
  - Il driver può essere scritto in **linguaggio** ad alto livello (es. C) e non in assembly – i registri sono solo variabili in memoria facilmente modificabili
  - **Protezione** semplice – controllo degli indirizzi
  - Le istruzioni possono riferirsi ai registri di controllo direttamente e semplificare la progettazione
- Svantaggi
  - Uso della **cache** non è possibile, va disabilitata selettivamente, azione potenzialmente complessa
  - Con bus separati i dispositivi di I/O potrebbero non poter vedere indirizzi di memoria spediti sul bus della memoria – alcune soluzioni

S. Balsamo – Università Ca' Foscari Venezia – SO.5.8

## I/O mappato in memoria



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.9

## DMA

DMA – **Direct Memory Access**

**Controllore** DMA, accesso diretto alla memoria

Accede al bus indipendentemente dalla CPU

Ha **molte** registri

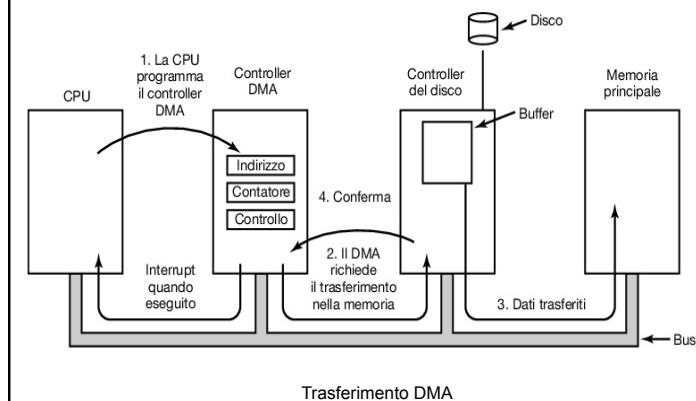
inclusi registri di memoria,  
di conteggio di byte  
di controllo

i registri di controllo contengono indicazioni delle porte I/O  
direzione di trasferimento  
dimensione dell'unità di trasferimento  
numero di byte da trasferire alla volta

Possibili trasferimenti multipli, con più registri di controllo, uno per canale e ogni trasferimento regolato da un controller di dispositivo

S. Balsamo – Università Ca' Foscari Venezia – SO.5.10

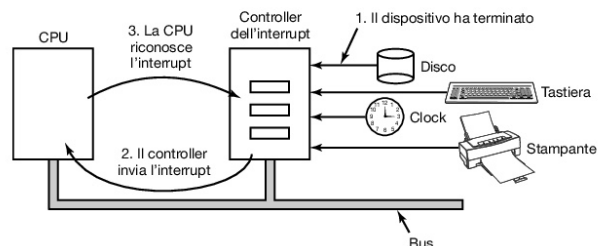
## DMA



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.11

### Gestione degli Interrupt

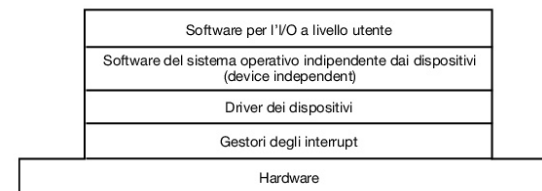


1. Il dispositivo causa l'Interrupt segnalato sulla linea del bus assegnatagli
2. Rilevato dal **controller dell'Interrupt** può essere trattato o momentaneamente ignorato.  
Il controller assegna linee di indirizzo al dispositivo e manda il segnale alla CPU
3. La CPU tratta l'Interrupt e usa le linee di indirizzo come indice della tabella (**vettore di Interrupt**) e preleva il nuovo PC
4. Inizia la **procedura** per il **trattamento** dell'Interrupt

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.12

### Livelli di software di I/O



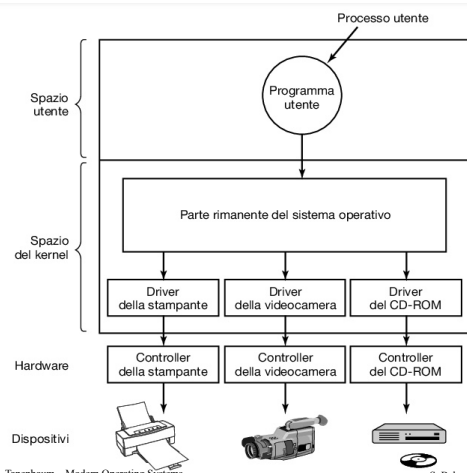
Gerarchia e livelli

- I gestori di interrupt di regola non sono visibili all'utente
- **Driver** del dispositivo: codice di controllo – solitamente nel nucleo dispositivi a blocchi / a caratteri
- **Software** indipendente dai dispositivi fornisce l'interfaccia al
- **Software** di livello utente

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.13

### Driver dei dispositivi di I/O



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.14

### Software **indipendente** dai dispositivi di I/O

Interfacciamento uniforme dei driver dei dispositivi

Buffering

Segnalazione degli errori

Allocazione e rilascio dei dispositivi dedicati

Dimensione dei blocchi indipendente dai dispositivi

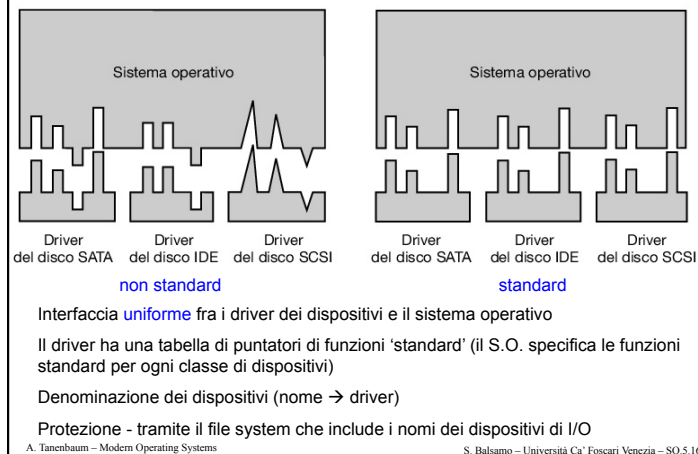
Alcune funzioni

**Interfaccia uniforme** a livello utente

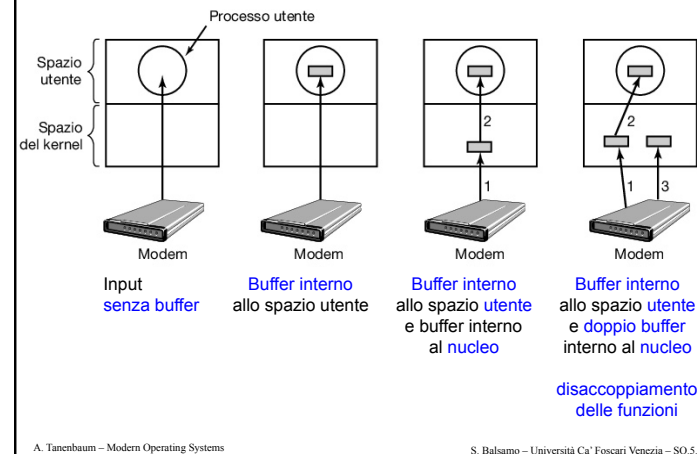
A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.15

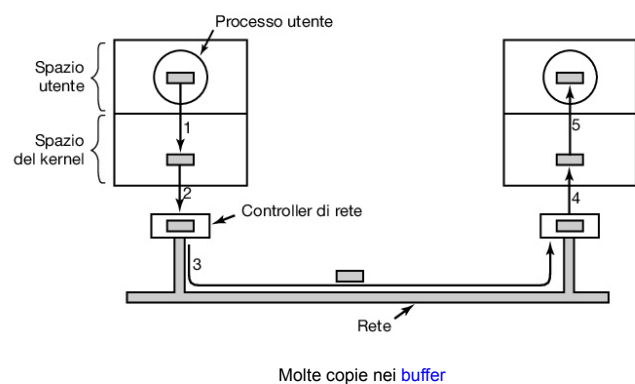
### Interfacce standard dei driver dispositivi di I/O



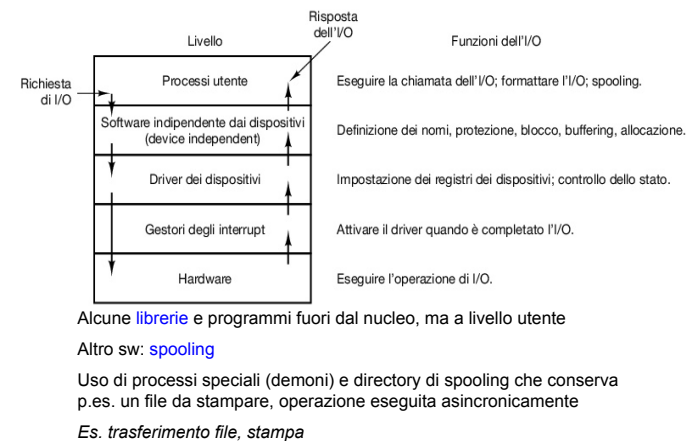
### Buffering per dispositivi di I/O



### Buffering per dispositivi di I/O e su rete



### Software di I/O nello spazio utente



## Obbiettivi del software di I/O

- Progettazione del software **indipendente** dal dispositivo  
definizione **uniforme** dei nomi (di file, di dispositivi,...)
- **Affidabilità**, correzione degli errori gestiti preferibilmente hardware
- **Tipi** di trasferimento - **comunicazione** dati CPU-I/O con operazioni  
sincrone (bloccanti) o  
asincrone (con interrupt)
- Gestione dei **buffer** nel trasferimento dei dati  
prestazioni, (es. operazioni real-time)
- **Condivisione**  
dispositivi condivisi da più utenti/processi  
dispositivi dedicati

A. Tanenbaum - Modern Operating Systems

S. Balsamo - Università Ca' Foscari Venezia - SO.5.20

## Tipi di software per I/O

Metodi di gestione software dell'I/O

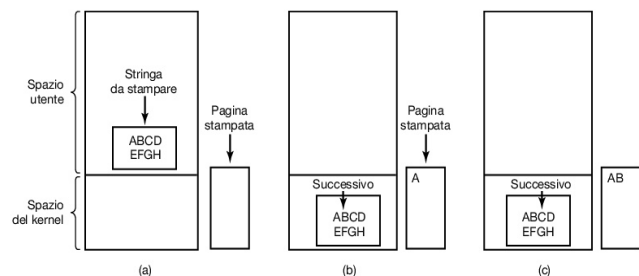
- I/O **programmato**  
delega alla CPU il controllo dell'operazione  
**busy waiting** della CPU  
semplice, ma potenzialmente inefficiente
- I/O guidato dall'**interrupt**  
la CPU può eseguire altri processi mentre un processo è bloccato  
uso di interrupt  
maggiore utilizzazione della CPU, ma molti interrupt
- I/O su **DMA**  
il controllore DMA, indipendente dalla CPU, interagisce con il dispositivo  
uso di hardware speciale, ma maggiore concorrenza e utilizzo CPU  
riduce il numero di interrupt, ma può essere più lento

A. Tanenbaum - Modern Operating Systems

S. Balsamo - Università Ca' Foscari Venezia - SO.5.21

## Tipi di software per I/O

Esempio di I/O **programmato**



- (a) Buffer nello spazio utente - chiamata di sistema e copia nello spazio kernel  
(b) Controllo della stampa un carattere per volta - uso del registro dati della stampante  
(c) Avanzamento dopo il controllo che la stampante sia disponibile

A. Tanenbaum - Modern Operating Systems

S. Balsamo - Università Ca' Foscari Venezia - SO.5.22

## Tipi di software per I/O

Esempio di I/O **programmato**

Stampa di una stringa  
p buffer nel nucleo  
un carattere per volta tramite il buffer  
CPU controlla

```
copy_from_user(buffer, p, count);          /* p è il buffer del kernel */
For (i = 0; i < count; i++){                /* ripeti per tutti i caratteri */
    while ("printer_status_reg != READY");  /* ripeti finché lo stato della stampante
                                                non è READY */
    *printer_data_register = p[i];          /* output di un carattere */
}
return_to_user( );
```

A. Tanenbaum - Modern Operating Systems

S. Balsamo - Università Ca' Foscari Venezia - SO.5.23

## Tipi di software per I/O

Esempio di I/O guidato dall'**interrupt**

Stampa di una stringa  
p buffer nel nucleo  
un carattere per volta via via che arrivano  
gestione dell'interrupt

```
copy_from_user(buffer, p, count);
enable_interrupts();
while ("printer_status_reg != READY) ;
*printer_data_register = p[0] ;
scheduler();
```

Codice eseguito al tempo di  
chiamata di sistema per la stampa

```
If count == 0) {
    unblock_user();
} else {
    *printer_data_register = p[0] ;
    count = count - 1;
    i = i + 1;
}
acknowledge-interrupt();
return_from_interrupt( );
```

Procedura di gestione dell'interrupt per la stampa

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.24

## Tipi di software per I/O

Esempio di I/O su **DMA**

Stampa di una stringa tramite DMA  
gestione tramite accesso diretto alla stampante

```
copy_from_user(buffer, p, count);
setup_DMA_controller();
scheduler();
```

Codice eseguito al tempo di  
chiamata di sistema per la stampa

```
acknowledge-interrupt();
unblock_user( );
return_from_interrupt( );
```

Procedura di gestione dell'interrupt per la stampa

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.25

## Elaborazione di un interrupt per I/O

Alcuni passi del S.O. per trattare un interrupt I/O

- **Salvataggio** dei **registri** non ancora salvati dall'*hardware* (es PSW)
- **Caricamento contesto** per la procedura di gestione dell'interrupt
- **Impostazione stack**
- **Avviso al controllore degli interrupt** (o *riabilitazione interrupt*)
- **Copia** dei **registri** salvati nella *tabella dei processi*
- **Esecuzione** della procedura di **gestione** dell'interrupt, che recupera le informazioni dai registri del controllore del dispositivo
- **Scelta** del prossimo processo da eseguire
- **Impostazione** del **contesto** della MMU per il prossimo processo, eventualmente anche della TLB
- **Caricamento** dei nuovi **registri** del processo, compreso PSW
- **Avvio** dell'esecuzione del **nuovo** processo

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.26

## Evoluzione dei sistemi di memoria secondaria

- La maggior parte dei dispositivi di memorizzazione secondaria si basano su **supporti magnetici**
  - Accesso ai dati con una **testina di lettura-scrittura**
  - I primi tecnologie utilizzavano **memoria sequenziale**
    - Informazioni accessibili in modo ordinato uno per volta
    - Inefficiente per applicazioni ad accesso diretto
- Memorizzazione ad **accesso casuale**
  - Anche detto memoria **ad accesso diretto**
  - Accesso ai record in qualsiasi ordine

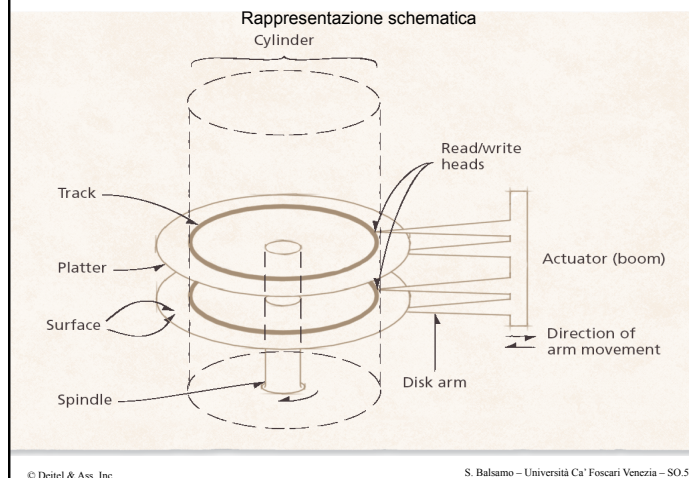
S. Balsamo – Università Ca' Foscari Venezia – SO.5.27

### Caratteristiche dei dischi a tesina mobile

- Struttura **fisica** di unità disco
  - Insieme di **dischi** (piatti) magnetici
    - Che **ruotano** su un perno (rotore)
    - Alta velocità
    - Composto da **tracce**, che a loro volta contengono **settori**
    - **Cilindri**: formati da gruppi verticali di tracce
    - Testina di lettura-scrittura molto vicina (micron)
    - **Braccio mobile** collegato ad un attuttore (*boom*)
    - Movimento della testina fra i cilindri
    - Ricerca del cilindro (**seek**)

S. Balsamo – Università Ca' Foscari Venezia – SO.5.28

### Caratteristiche dei dischi a tesina mobile



S. Balsamo – Università Ca' Foscari Venezia – SO.5.29

### Caratteristiche dei dischi a tesina mobile

Parametro	Floppy disk IBM 360 KB	Disco fisso WD 3000 HLFS
Numero dei cilindri	40	36.481
Tracce per cilindro	2	255
Settori per traccia	9	63 (media)
Settori per disco	720	586.072.368
Byte per settore	512	512
Capacità del disco	360 KB	300 GB
Tempo di ricerca (cilindri adiacenti)	6 ms	0,7 ms
Tempo di ricerca (situazione media)	77 ms	4,3 ms
Tempo di rotazione	200 ms	6 ms
Tempo di stop/avvio del motore	250 ms	1 ms
Tempo per trasferire 1 settore	22 ms	1 μs

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.30

### Caratteristiche dei dischi a tesina mobile

- Indici di prestazione
  1. **Tempo di ricerca (seek)**
    - Tempo per la testina di lettura-scrittura per spostarsi nuovo cilindro
  2. **Latenza rotazionale**
    - Tempo di ritardo dovuto alla rotazione, perché i dati ruotino dalla posizione attuale alla testina di lettura-scrittura
  3. **Tempo di trasmissione**
    - Tempo di **trasferimento** perché tutti i dati cercati ruotino sotto la testina di lettura-scrittura

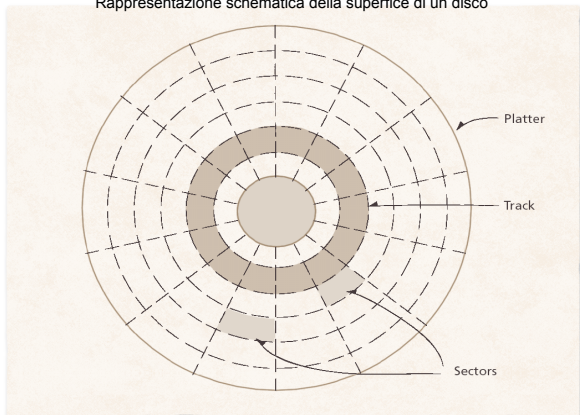
S. Balsamo – Università Ca' Foscari Venezia – SO.5.31



### Caratteristiche dei dischi a tesina mobile

Dischi dividono le tracce in molti settori, ognuno tipicamente contiene 512 byte

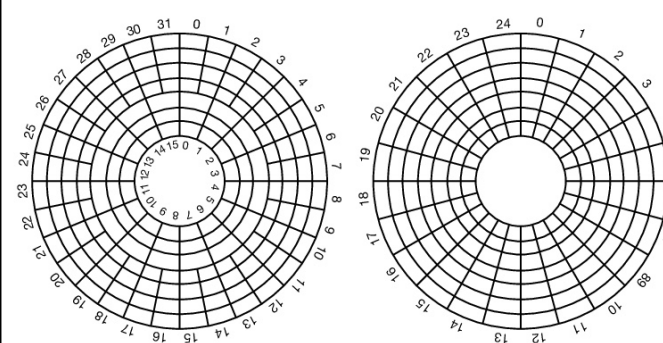
Rappresentazione schematica della superficie di un disco



© Deitel & Ass. Inc.

S. Balsamo – Università Ca' Foscari Venezia – SO.5.32

### Caratteristiche dei dischi a tesina mobile



Numero di settori per cilindro – **geometria fisica** e **geometria virtuale**

I dischi moderni supportano un sistema di **indirizzamento logico** dei blocchi con numerazione dei settori senza considerare la geometria del disco

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.33

### RAID – Redundant array of inexpensive disks

Uso di meccanismi di  
ridondanza per aumentare l'**affidabilità**  
parallelismo per migliorare le **prestazioni** dei dischi

RAID è **visto** dal sistema come **una** sola grande unità disco

La progettazione include

- la gestione di un **insieme di dischi** a cui si accede in parallelo
- la gestione di molte **copie** di parti di dati
- la **distribuzione** delle parti dei dati su più dischi

**Distribuzione**

**Partizione**

Insiemi (pila) di dischi SCSI o SATA

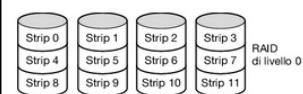
Compatibilità dei driver

Diversi schemi di organizzazione dei RAID, detti livelli

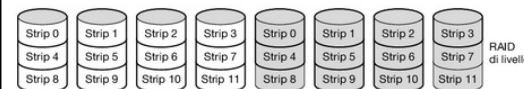
A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.34

### RAID – Redundant array of inexpensive disks



**Striping**  
ciascuna di k settori



**Parallelismo**  
**Prestazioni**



**Ridondanza** per la  
tolleranza i guasti



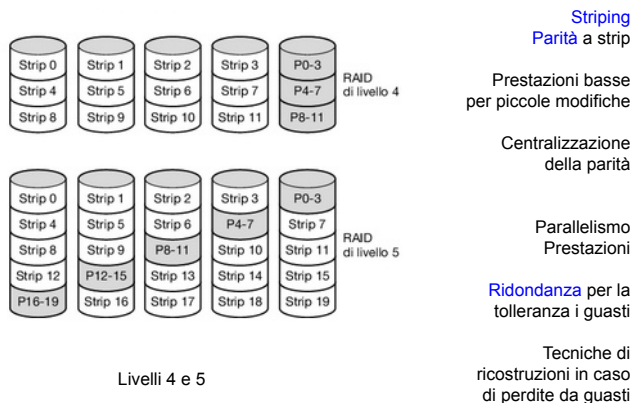
**Parità** a parole o a  
strip  
Sincronizzazione dei  
dischi

**Livelli 0,1,2,3**

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.35

## RAID – Redundant array of inexpensive disks



A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.36

## Formattazione dei dischi

**Formattazione a basso livello** dei piatti del disco, via sw  
Tracce concentriche con i settori



Settore di un disco

ECC informazioni ridondanti per recupero di errori, es. 16 byte

Lo spazio nel disco formattato si riduce es. circa del 20%

**Cylinder Skew**

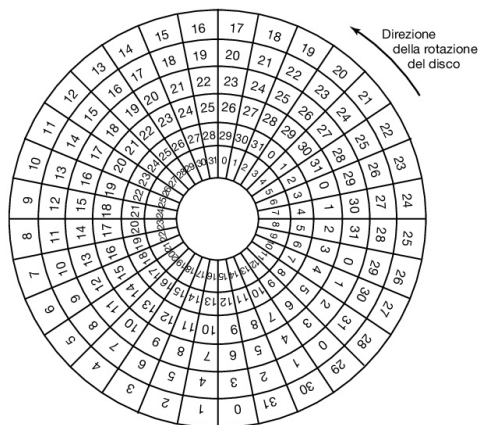
per ogni traccia il settore 0 è spostato rispetto alla traccia precedente per migliorare le prestazioni

**Partizionamento del disco:** tabella delle partizioni e dimensione di ogni partizione**Formattazione ad alto livello** di ogni partizioneDefinisce: il blocco di avvio, gestione dello spazio libero, directory principale (*root*) e il file system (vuoto)

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.37

## Formattazione dei dischi

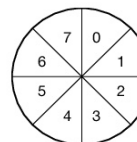


Cylinder Skew

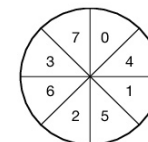
A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.38

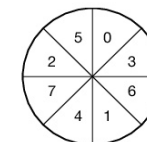
## Formattazione dei dischi



Nessun interleaving



Interleaving singolo



Interleaving doppio

Lettura di due settori consecutivi

Permette al controller di copiare un settore dal buffer in memoria principale

- Interleaving singolo
- Interleaving doppio (per copie lente)

A. Tanenbaum – Modern Operating Systems

S. Balsamo – Università Ca' Foscari Venezia – SO.5.39