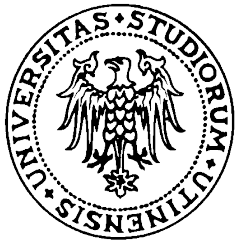
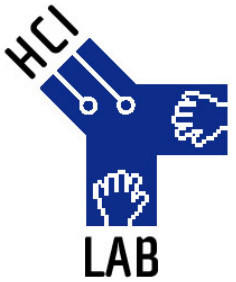
Memoria secondaria



Fabio Buttussi

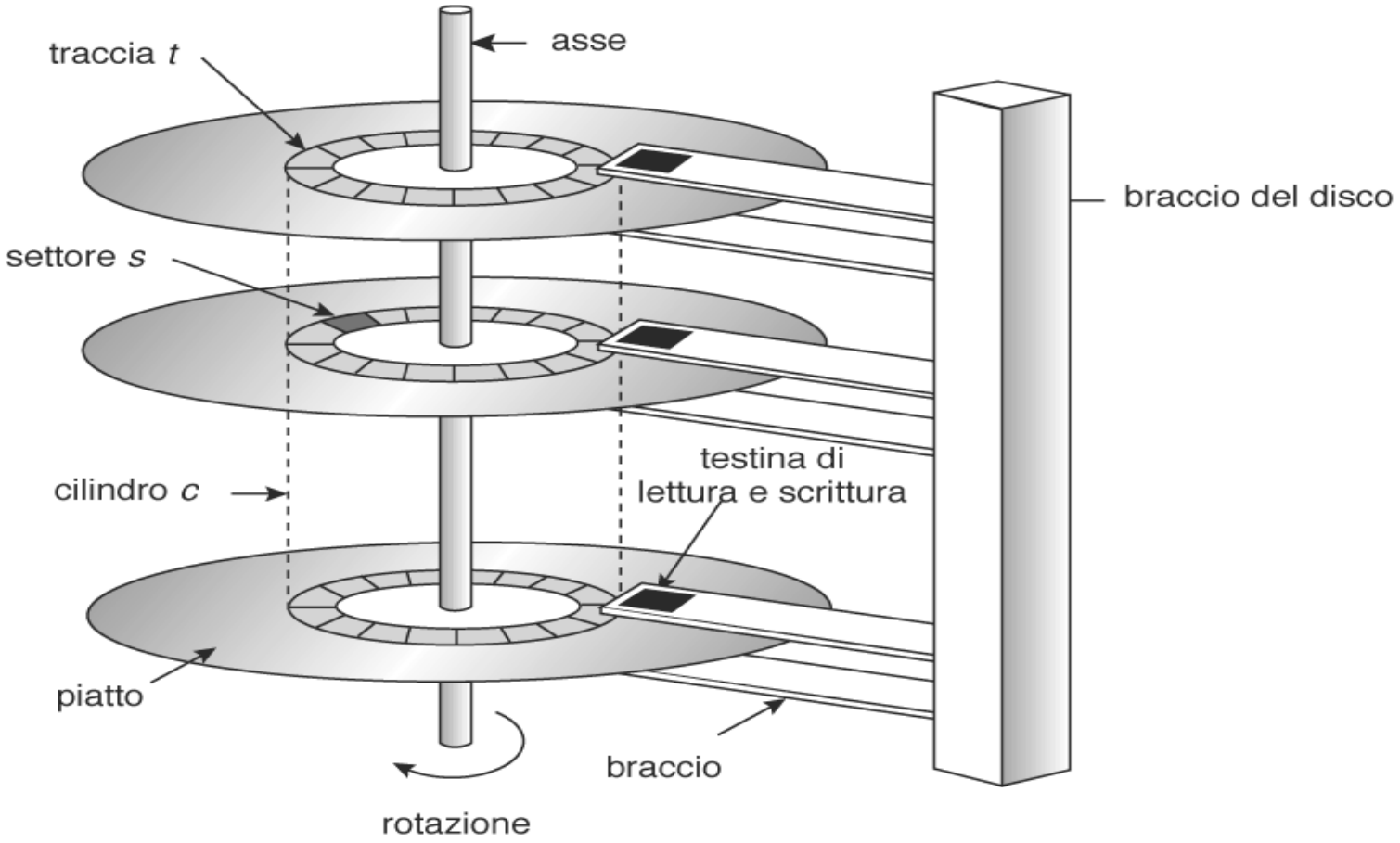
HCI Lab
Dept. of Math and Computer Science
University of Udine
ITALY

www.dimi.uniud.it/buttussi



Struttura di un disco

- Da un punto di vista logico, rappresenta il **livello più basso** del file system



Efficienza di un disco

- L'efficienza di un disco è caratterizzata da:
  - **Velocità di trasferimento** dei dati dal disco al calcolatore
  - **Tempo di posizionamento** (o d'accesso): somma di
    - **tempo di ricerca** (**seek** time) e
    - **latenza di rotazione**

Dischi e operazioni di I/O

- Connessione al calcolatore attraverso un **bus di I/O**

- Trasferimento dati mediato da **controllori**
  - Adattatori dalla parte del calcolatore
  - Controllori dei dischi dalla parte dei dischi

- **Operazioni di I/O**
  - Calcolatore inserisce un comando nell'adattatore
  - Adattatore invia comando al controllore del disco
  - Controllore agisce sugli elementi elettromeccanici

Scheduling del disco

- Il **sistema operativo** deve garantire **tempi d'accesso contenuti** e **ampiezze di banda elevate**
- Le richieste di I/O effettuate dai processi vengono tipicamente aggiunte ad una **coda di richieste**
- Le richieste in coda sono soggette a **scheduling**

Scheduling FCFS e SSTF

- **First Come First Served** (FCFS):
  - Le richieste vengono soddisfatte nell'**ordine** in cui arrivano
  - Algoritmo **equo**, ma **inefficiente**

- **Shortest Seek Time First** (SSTF):
  - Sceglie la richiesta che necessita del **minimo tempo di ricerca** rispetto alla posizione corrente della testina
  - Può portare ad **attesa indefinita**
  - Più **efficiente** di FCFS, ma **non ottimale**

Scheduling SCAN e C-SCAN

- **SCAN**
  - Il braccio dell'unità a disco parte da un'estremità del disco e si sposta in un'**unica direzione**
  - Le richieste vengono servite mentre si attraversano i cilindri
  - All'altra estremità del disco si inverte la direzione
  - La densità di richieste (e il loro **tempo d'attesa**) è **maggiore** all'aumentare della **distanza** dalla posizione corrente
- **C-SCAN**
  - **Variante** di SCAN che tratta il disco come una **coda circolare** per minimizzare il tempo d'attesa

Scheduling LOOK e C-LOOK

- **Varianti** di SCAN e C-SCAN che **spostano** la **testina** in una direzione **fino** all'**ultima richiesta** presente
- Permettono una **maggior efficienza**
- Scheduling migliori quando si utilizzano molto le unità a disco (no attesa indefinita)

Scelta di un algoritmo di scheduling

- In generale, l'algoritmo migliore dipende dal **numero e tipo di richieste**
- Il **tipo di allocazione** influenza il tipo di richiesta di I/O (spostamenti limitati per allocazione contigua, spostamenti sparsi per allocazione concatenata o indicizzata)
- **Posizione** delle **directory** e dei **file indice** influenzano gli spostamenti delle testine
- Algoritmi considerati non tengono conto del tempo di latenza
- Lo scheduling può essere **parzialmente affidato** ai **controllori** dei dischi

Formattazione di un disco

- **Formattazione di basso livello**: suddivide il disco in **settori**; ogni settore consiste di un'intestazione, un'area dati e una coda
- **Intestazione** e **coda** contengono **informazioni** utili al **controllore** (es. numero del settore)
- Per utilizzare il disco, il SO vi registra le proprie strutture dati in due passi:
  - Suddivisione in **partizioni**
  - **Formattazione logica** (creazione di un file system)
- Alcuni SO possono impiegare **partizioni senza file system**

Blocchi difettosi

- Nei dischi gestiti da un controllore IDE, la gestione dei blocchi difettosi avviene **manualmente**
  - In DOS, durante la formattazione logica, i blocchi difettosi individuati vengono segnalati con un valore apposito all'interno della FAT
  - Durante l'uso del sistema, i blocchi difettosi vengono indiviuati grazie a programmi appositi (chkdsk) e vengono gestiti tramite la FAT
- Nei sistemi più complessi, si utilizzano **strategie più avanzate**
  - Lista di blocchi malfunzionanti
  - Accantonamento di settori
  - Traslazione dei settori

Gestione dell'area di avvicendamento

- Area utilizzata per **swapping** e memoria virtuale
- Collocazione all'**interno del file system**
  - soluzione semplice ma inefficiente
- Collocazione in una **partizione apposita non formattata**
  - si usa un gestore specifico che adotta algoritmi ottimizzati rispetto alla velocità
  - frammentazione interna non problematica
- In Linux, l'area di avvicendamento (swap file o swap area) è usata solo per memoria anonima (pila, heap) e regioni di memoria condivise
  - una mappa d'avvicendamento associata ad ogni area di avvicendamento tiene traccia del numero di collegamenti (processi) ad ogni slot delle pagine