

# Sistemi Operativi

Laurea in Ingegneria Informatica

Università Roma Tre

Docente: Romolo Marotta

## I/O e file management

# I/O management

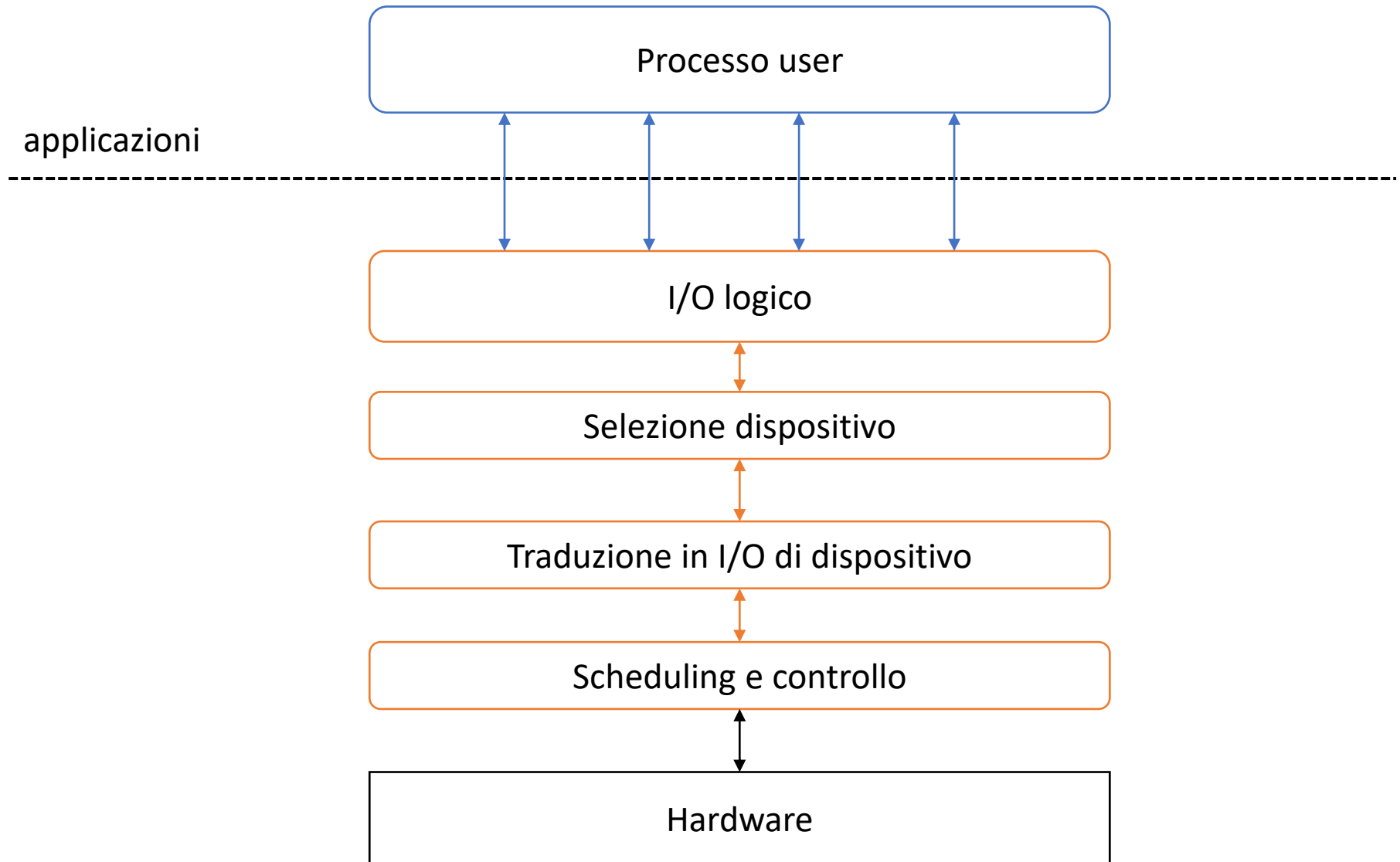
- Diversi tipi di dispositivi
  - Interazione con l'essere umano
  - Interazione con la macchina
  - Comunicazione tra macchine
- Caratteristiche differenti:
  - Applicazioni
  - Data rate
  - Controllo
  - Unità del trasferimento di dati
  - Rappresentazione del dato
  - Condizioni di errore
  - Gestione dei consumi

# I/O management

## Obiettivi:

- Efficienza
  - Tipicamente l'interazione con dispositivi di I/O è il collo di bottiglia
  - Multiprogrammazione allevia il problema, ma richiede swapping
  - Swapping richiede operazioni di I/O
- Generalità
  - Necessità di trattare i dispositivi in modo **uniforme**
  - Fornire servizi di I/O con interfacce standard (indipendenti dal tipo di dispositivo)
  - Progettazione gerarchica e modulare tesa a nascondere dettagli di basso livello

# Modello di organizzazione moduli di I/O



# I/O management

- Evoluzione delle funzioni di I/O
  - Programmed I/O
    - Processore **attende** la terminazione di comandi e **copia** dati da I/O a memoria
  - Interrupt-driven I/O
    - Processore **non attende** la terminazione di comandi e **copia** dati da I/O a memoria
  - Direct memory access
    - Processore **non attende** la terminazione di comandi e **non copia** dati da I/O a memoria

# I/O buffering

L'I/O diretto su memoria dei processi ha implicazioni non banali:

- area destinata all'I/O non è swappabile
  - sottoutilizzo delle risorse
- area destinata all'I/O è swappabile
  - Deadlock
    - Processo bloccato in attesa di I/O e poi swappato
    - I/O in attesa che il processo sia riattivato
- I/O viene effettuato su memoria riservata al sistema operativo chiamato **buffer**
- L'utilizzo di buffer è in generale utile al fine di risolvere altre criticità:
  - Appianare la differenza di velocità tra produttore e il consumatore del dato
  - Appianare la differenza della taglia del dato che può essere maneggiata dal produttore e dal consumatore
  - Supportare la semantica di copia per l'I/O

# I/O buffering

- No buffering

User process



- Single buffer

User process

Operating system



- Double buffer

User process

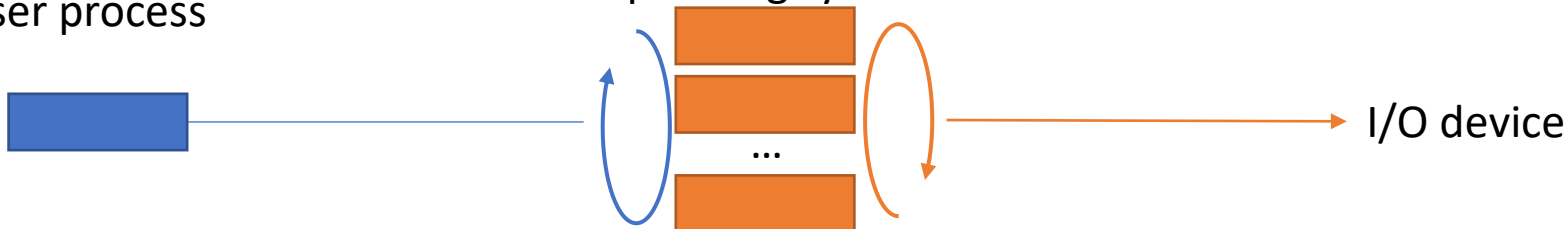
Operating system



- Circular buffer

User process

Operating system



# I/O scheduling

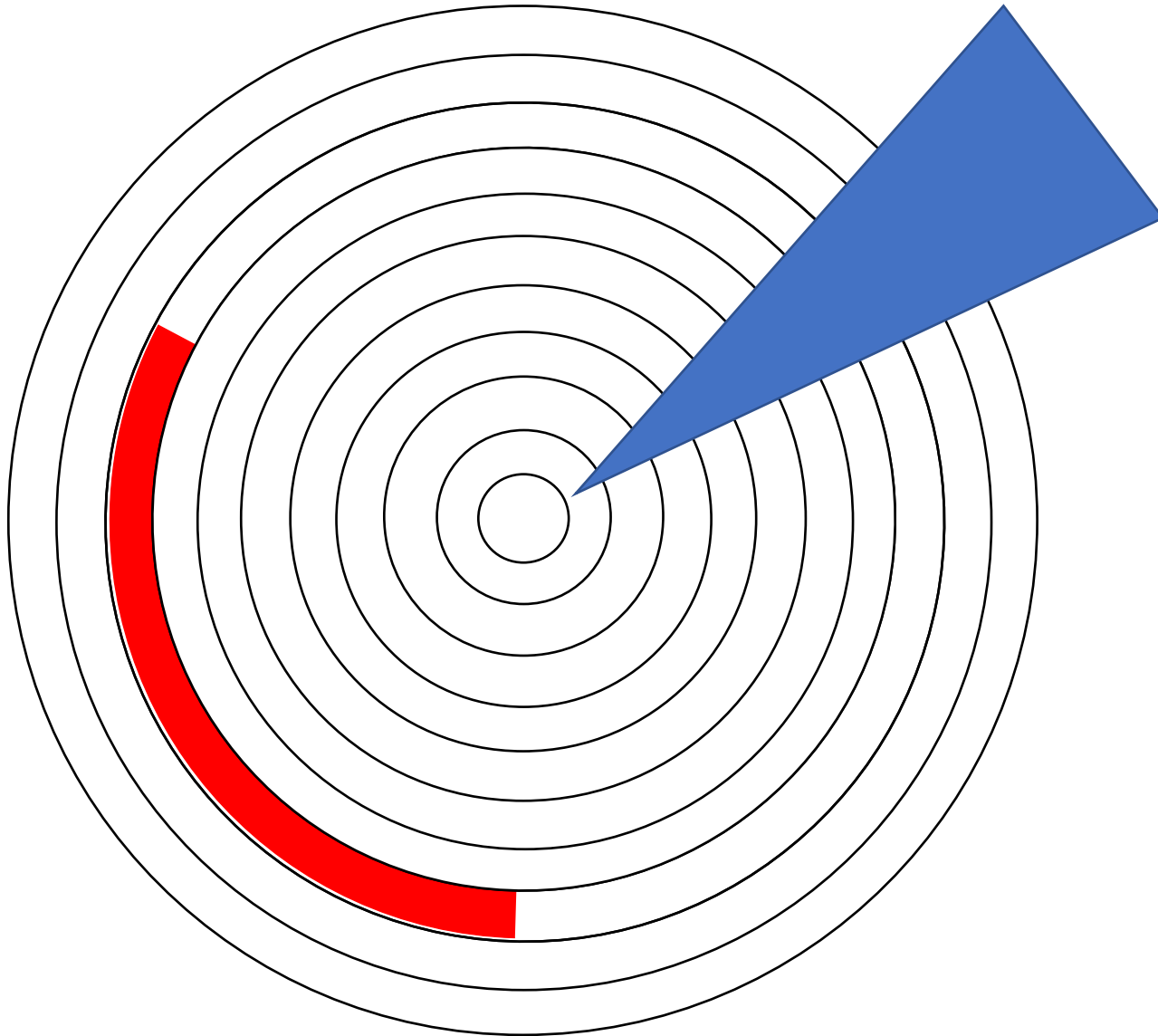
- Definisce la pianificazione per cui un dispositivo di I/O viene attivato per le sue operazioni
- In alcuni casi, mantenere l'ordine tra richieste di I/O e le effettive operazioni può essere necessario (e.g., terminale)
- FCFS non è necessariamente la soluzione più efficiente
  - Forte dipendenza dalle peculiarità dell'hardware caratteristico del dispositivo di I/O



# Hard disk – caratteristiche salienti

- Ogni blocco è accessibile in lettura/scrittura ad ogni istante di tempo
- L'usura del dispositivo è essenzialmente legata all'usura delle parti meccaniche coinvolte
  - Non ci sono relazioni dirette tra operazioni di scrittura/lettura ed usura del dispositivo
  - Relazioni indirette tra operazioni ed usura
    - Per leggere/scrivere è necessario muovere la testina
- Tempo di accesso
  - Tempo di accodamento della richiesta
  - Tempo di acquisizione del canale di I/O (potrebbe essere condiviso con altri dispositivi)
  - Seek time: tempo per spostare la testina sulla traccia corretta
  - Ritardo di rotazione: tempo di rotazione per allineare l'inizio del settore di interesse alla testina
  - Tempo di trasferimento: tempo speso a ruotare il disco affinché tutto il settore venga letto dalla testina

# Hard disk



# Hard disk – Tempi di accesso

- Tempo di trasferimento
  - $B$ =byte da trasferire
  - $N$ =byte per traccia
  - $R$ =tempo di rivoluzione
  - $BR/N$
- Ritardo di rotazione
  - in media occorre metà giro per posizionarsi sul settore corretto
  - Tipicamente il disco gira ad una velocità costante (da 5400rpm a 15000rpm)
  - Da 5.5ms a 2ms
- Seek time
  - in media occorre un terzo del full seek time (dalla traccia più interna a quella più esterna)
  - tipicamente nell'ordine dei millisecondi

# Hard disk – Tempi di accesso

## Ritardo di rotazione

- Perché metà giro è la distanza attesa per allinearsi al settore di interesse?
- Una volta che la testina raggiunge la traccia di interesse, l'inizio del settore può trovarsi ad una qualsiasi distanza angolare dalla testina
- In altre parole, la distanza angolare  $X$  è uniformemente distribuita tra 0 e l'angolo giro ( $2\pi$ )
  - $X \sim f(x) = \frac{1}{2\pi}$  per  $x \in [0, 2\pi]$ , 0 per  $x \notin [0, 2\pi]$
  - $\mathbb{E}(X) = \int_{-\infty}^{+\infty} xf(x) dx = \int_0^{2\pi} \frac{x}{2\pi} dx = \left[ \frac{x^2}{4\pi} \right]_0^{2\pi} = \frac{4\pi^2}{4\pi} = \pi$

# Hard disk – Tempi di accesso

## Seek time

- Perché un terzo del full seek time è la distanza attesa per raggiungere la traccia di interesse?
- Si calcola come il rapporto tra la somma di tutte le distanze possibili tra due tracce ( $A$ ) e il numero di movimenti distinti che la testina può effettuare ( $B$ )
- Sia  $N$  il numero di tracce del disco
- La traccia di partenza può essere scelta tra  $N$
- La traccia destinazione può essere scelta tra  $N$
- $B = N^2$

# Hard disk – Tempi di accesso

- Data la traccia  $i$ , il numero di tracce  $D(i, j)$  da attraversare per raggiungere una traccia  $j$  è

$$D(i, j) = |i - j|$$

- La somma di tutte le possibili distanze tra una specifica traccia  $i$  e una generica traccia  $j$  è

$$A_i = \sum_{j=1}^N D(i, j) = \sum_{j=1}^N |i - j|$$

- La somma di tutte le possibile distanze  $A$  è

$$A = \sum_{i=1}^N A_i = \sum_{i=1}^N \sum_{j=1}^N |i - j|$$

$$\sum_{i=1}^N \left[ \sum_{j=1}^i (i - j) + \sum_{j=i+1}^N (j - i) \right]$$

# Hard disk – Tempi di accesso

Seek time

$$\sum_{i=1}^N \left[ \sum_{j=1}^i (i-j) + \sum_{j=i+1}^N (j-i) \right]$$

$$\sum_{i=1}^N \left[ \sum_{j=1}^i (i-j) + \sum_{j=1}^N (j-i) - \sum_{j=1}^i (j-i) \right]$$

$$\sum_{i=1}^N \left[ \sum_{j=1}^i 2(i-j) + \sum_{j=1}^N (j-i) \right]$$

$$\sum_{i=1}^N \left[ 2i^2 - i(i+1) + \frac{N(N+1)}{2} - Ni \right]$$

$$\sum_{i=1}^N \left[ i^2 - i + \frac{N(N+1)}{2} - Ni \right]$$

$$\frac{N(N+1)(2N+1)}{6} - \frac{N(N+1)}{2} + \frac{N^2(N+1)}{2} - \frac{N^2(N+1)}{2}$$

$$\frac{2N^3 + 3N^2 + N}{6} - \frac{N(N+1)}{2}$$

$$\frac{N^3}{3} - \frac{N}{12}$$

# Hard disk – Tempi di accesso

## Seek time

- Perché un terzo del full seek time è la distanza attesa per raggiungere la traccia di interesse?
- Il seek time atteso è

$$\frac{A}{B} = \frac{\frac{N^3}{3}}{N^2} = \frac{N}{3}$$



# Hard disk – Accesso sequenziale

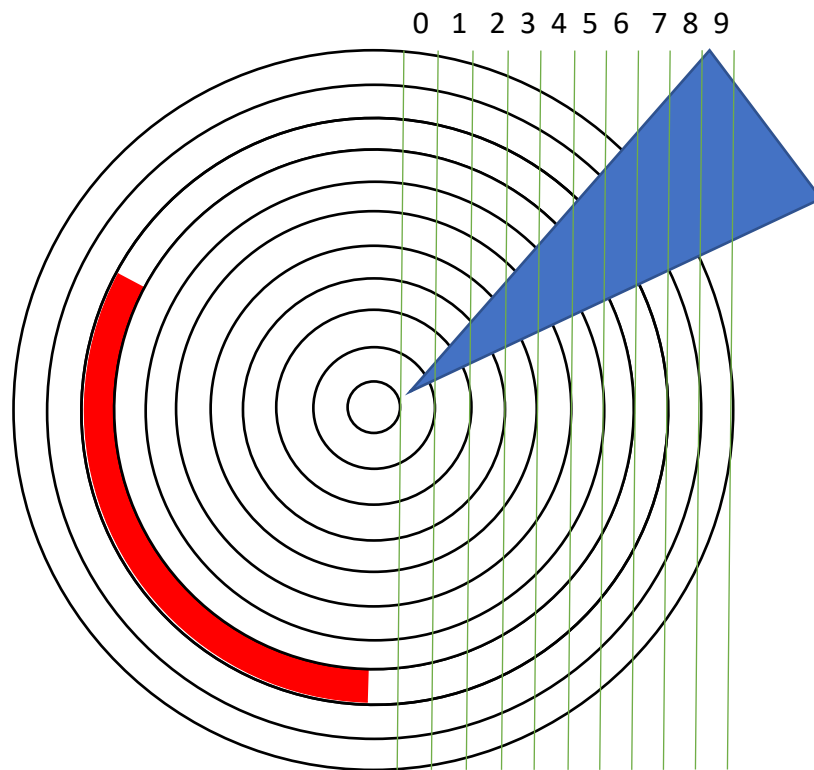
- Esempio
  - 500 settori per traccia
  - Average seek time = 4ms
  - Velocità di rotazione = 15000rpm = 250rps
  - Seek time per tracce adiacenti trascurabile
- Dati posizionati su 5 tracce consecutive
  - Tempo di rivoluzione = 4ms
  - Tempo di trasferimento =  $5 \times (\text{tempo di rivoluzione}) = 20\text{ms}$
  - Tempo di seek = 4ms (da considerare una sola volta)
  - Ritardo di rotazione complessivo =  $5 \times (\text{tempo di rivoluzione}) / 2 = 10\text{ms}$
  - Tempo totale per l'accesso =  $20\text{ms} + 10\text{ms} + 4\text{ms} = 34\text{ms}$

# Hard disk – accesso casuale

- Esempio
  - 500 settori per traccia
  - Average seek time = 4ms
  - Velocità di rotazione = 15000rpm = 250rps
  - Seek time per tracce adiacenti trascurabile
- Dati posizionati su 2500 settori non consecutivi
  - Tempo di rivoluzione = 4ms
  - Tempo di trasferimento =  $5 \times (\text{tempo di rivoluzione}) = 20\text{ms}$
  - Tempo totale di seek =  $2500 \times (\text{tempo di seek}) = 10\text{s}$
  - Ritardo di rotazione complessivo =  $2500 \times (\text{tempo di rivoluzione}) / 2 = 5\text{s}$
  - Tempo totale per l'accesso =  $20\text{ms} + 10\text{s} + 5\text{s} = 15.02\text{s}$

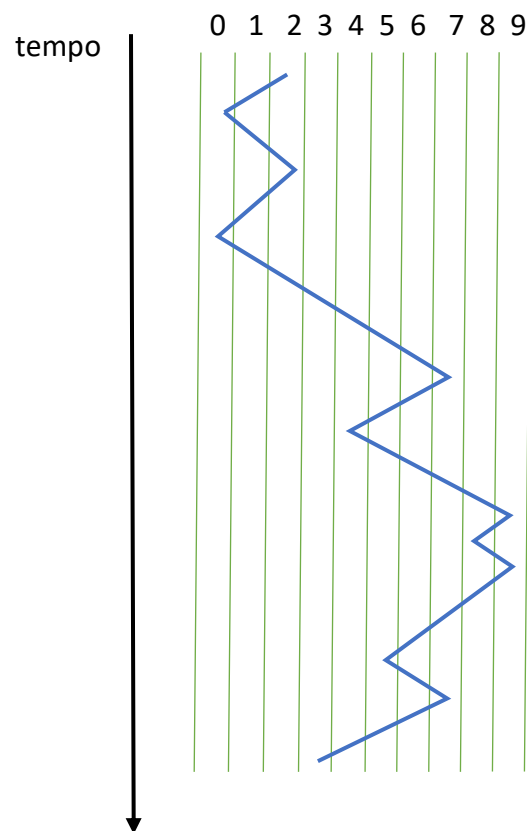
# Disk scheduling

- Minimizzare i movimenti meccanici è un aspetto cruciale per massimizzare le performance di accesso a disco
- Seek time è oggetto di numerosi algoritmi di ottimizzazione
- Metodologia
  - Esecuzione sintetizzata come una sequenza di richieste identificate dal numero di traccia
- Metrica
  - numero di tracce complessivamente attraversate



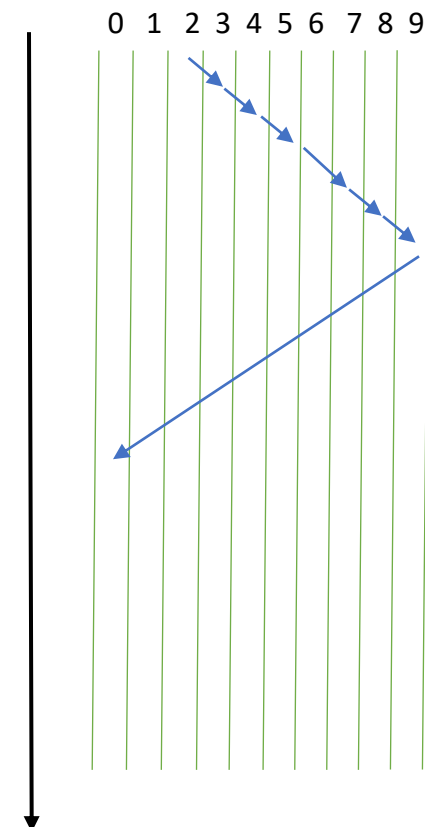
# Disk scheduling

- Richieste:
  - 0,2,0,7,4,9,8,9,5,7,3
  - Posizione iniziale: 2
- FCFS
  - Richieste servite nell'ordine di arrivo
  - No starvation
  - Non minimizza seek time
  - Movimenti totali della testina: 32



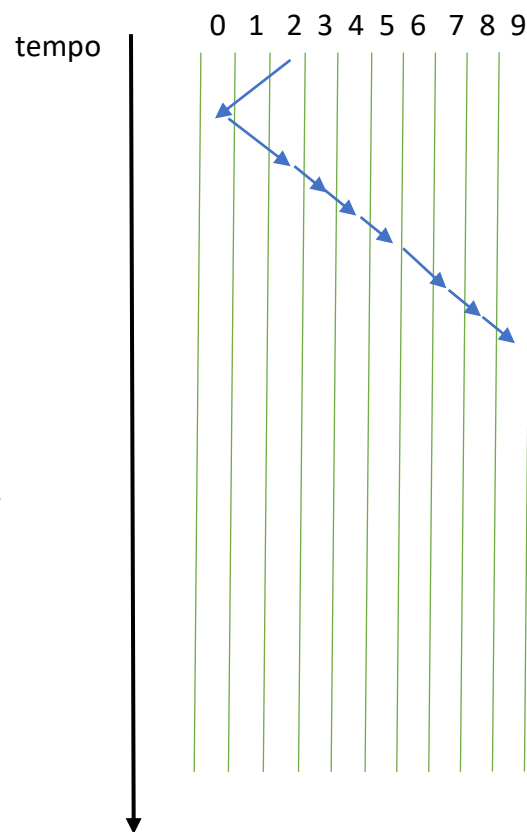
# Disk scheduling

- Richieste:
  - 0,2,0,7,4,9,8,9,5,7,3
  - Posizione iniziale: 2
- Shortest-seek-time-first (SSTF)<sup>tempo</sup>
  - Seleziona la traccia successiva che minimizza il seek time
  - 2,3,4,5,7,7,8,9,9,0,0
  - Movimenti totali della testina: 16
  - Starvation
  - Non minimizza seek time



# Disk scheduling

- Richieste:
  - 0,2,0,7,4,9,8,9,5,7,3
  - Posizione iniziale: 2, direzione decrescente
- SCAN (elevator)
  - Il seek avviene in una data direzione fino a che non ci sono più richieste lungo in quella direzione
  - 0,0,2,3,4,5,7,7,8,9,9
  - Movimenti totali della testina: 11
  - Performance simili a SSTF
  - Sfavorisce tracce servite in precedenza
  - Favorisce tracce agli estremi
  - Starvation



# Disk scheduling

- SCAN (elevator)
  - Sfavorisce tracce servite in precedenza
  - Favorisce tracce agli estremi
  - Starvation
- CSCAN
  - Le richieste vengono soddisfatte in un'unica direzione
  - Starvation
- FSCAN
  - Due code per la gestione delle richieste
  - Una coda viene usata per lo scheduling secondo SCAN
  - Nuove richieste vengono inserite in una coda di richieste pendenti
  - Quando la coda di scheduling corrente è vuota, le due code vengono scambiate

# Disk scheduling

- Gli algoritmi mostrati si focalizzano sul seek time
- Nei dischi moderni
  - il seek time è comparabile con il tempo di rotazione
  - non mostrano la posizione fisica del dato
- I produttori implementano algoritmi di scheduling nel controller del disco tesi a minimizzare sia seek time che ritardi di rotazione
- Il sistema operativo può delegare al controller se l'unico obiettivo è la performance
- Tuttavia, esistono scenari in cui alcune richieste sono prioritarie rispetto ad altre



# Solid state drives

- Basati su tecnologia flash
- Le scritture deteriorano le celle flash
  - Erase
  - Program
  - La vi
- La tecnologia NAND:
  - notevole densità (byte per chip)
  - Asimmetria tra unità di erase (blocco=N pagine) e program (pagina)
- Le pagine vengono gestite dal controller all'interno dell'unità SSD al fine di garantire **wear leveling**
  - I dati vengono spostati al fine di controllare operazioni di erase
- Nascondono al sistema operativo la posizione fisica dei dati
- I sistemi operativi:
  - Spesso adottano politiche di scheduling semplici (FCFS)
  - Devono notificare la cancellazione dei dati

# Operazioni di I/O logico

- Necessità di astrarre dalle peculiarità del dispositivo
- Si basano su modelli di riferimento
  - Stream I/O
  - Block I/O
- Le interfacce (system call) esistono in relazione ad oggetti logici
- Per interagire con un oggetto è necessario aprire un canale di I/O
- Il sistema operativo mantiene informazioni di sessione che permettono di relazionare operazioni susseguenti sull'oggetto
- Il trasferimento di dati da/verso oggetti logico **può** riflettersi su attività che coinvolgono dispositivi hardware (e.g., disco o interfacce di rete)

# **File system**

# Concetto di file

- File: minima unità informativa archiviabile
- Il ciclo di vita di un file è scorrelato dal ciclo di vita del processo che lo ha creato
- Se il file è archiviato su un dispositivo di memoria non volatile (e.g., disco, ssd), allora il ciclo di vita del file può includere molteplici riavvii della macchina
- Un file è costituito da un insieme di record
- Un record è l'unità minima manipolabile da un'applicazione all'interno di un file

# File system

- Il file system è il modulo di sistema operativo per gestire file
- Associa ad un file una serie di attributi:
  - Nome
  - Timestamp (creazione, modifica)
  - Protezione (chi può accedervi e con quali modalità)
- Gli attributi sono memorizzati in un Record di Sistema disgiunto dal file

# Operazioni su file

## Creazione

- Allocazione del record di sistema e inizializzazione del file

## Eliminazione

- Deallocazione del record di sistema e rilascio di tutti i record allocati per il file di interesse

## Apertura

- Inizializzazione di una sessione e di un relativo indice per accessi al file nella sessione

## Chiusura

- Distruzione della sessione, dei suoi metadati e dell'indice

## Scrittura/Lettura

- Accesso a partire dal record a cui l'indice fa riferimento

## Riposizionamento (seek)

- Aggiornamento dell'indice

## Troncamento

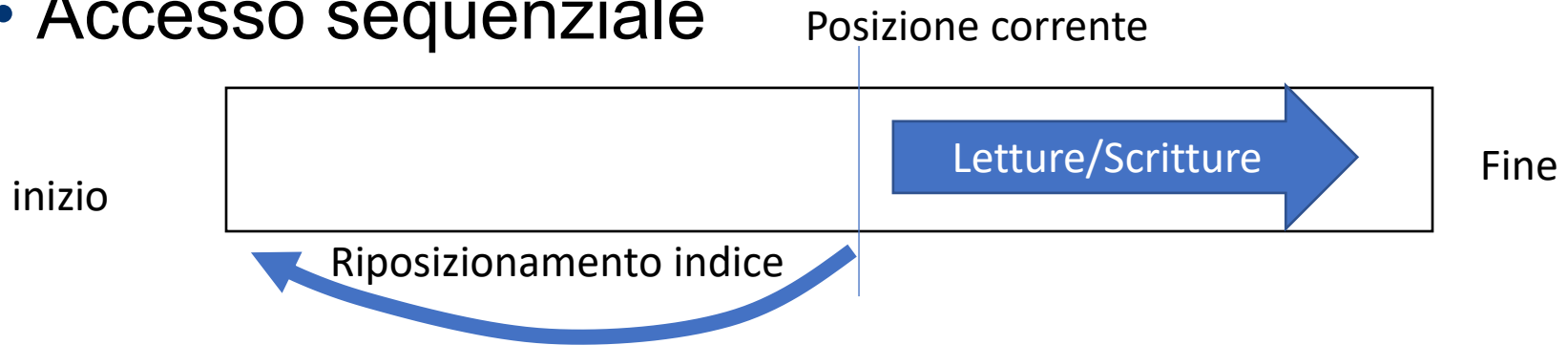
- Distruzione di record del file

# Indice di accesso

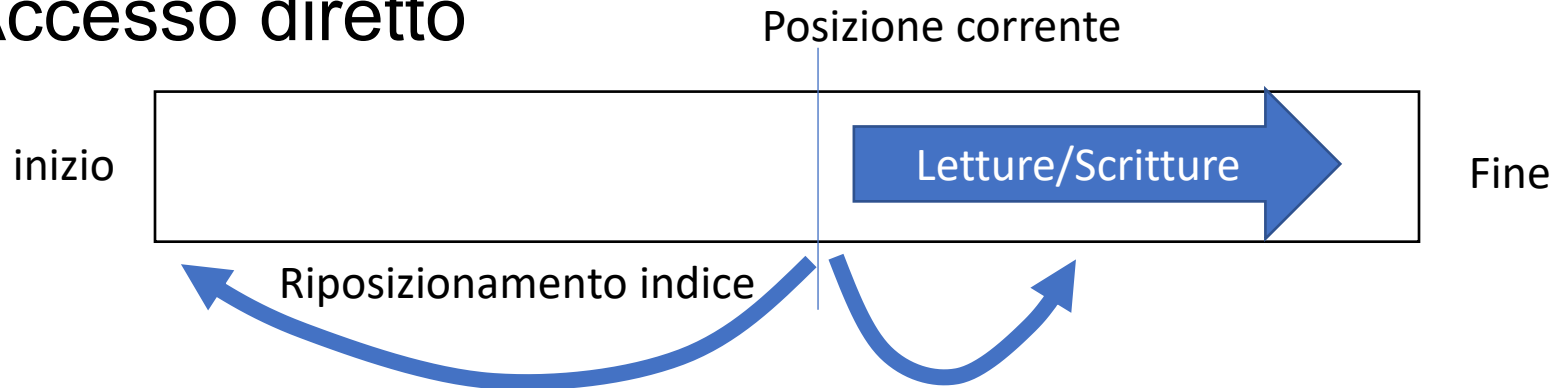
- Anche chiamato file pointer
- È un concetto di sessione
  - Non fa parte del record di sistema
- Una sessione può essere condivisa tra processi
  - Non è incluso nell'immagine di processo
- È memorizzato in una immagine di sessione
  - Struttura dati utilizzata per memorizzare tutti i metadati necessari alla gestione di accessi nell'ambito della sessione
- La modalità di aggiornamento dell'indice dipende dalle modalità di accesso ai record supportati dal file system

# Metodi di accesso

- Accesso sequenziale



- Accesso diretto



- Accesso indicizzato

- Grazie all'accesso diretto è possibile costruire indici per individuare specifici blocchi identificati da una chiave di ricerca all'interno del file