

# Sistemi Operativi

Laurea in Ingegneria Informatica

Università Roma Tre

Docente: Romolo Marotta

## I/O e file management

# I/O management

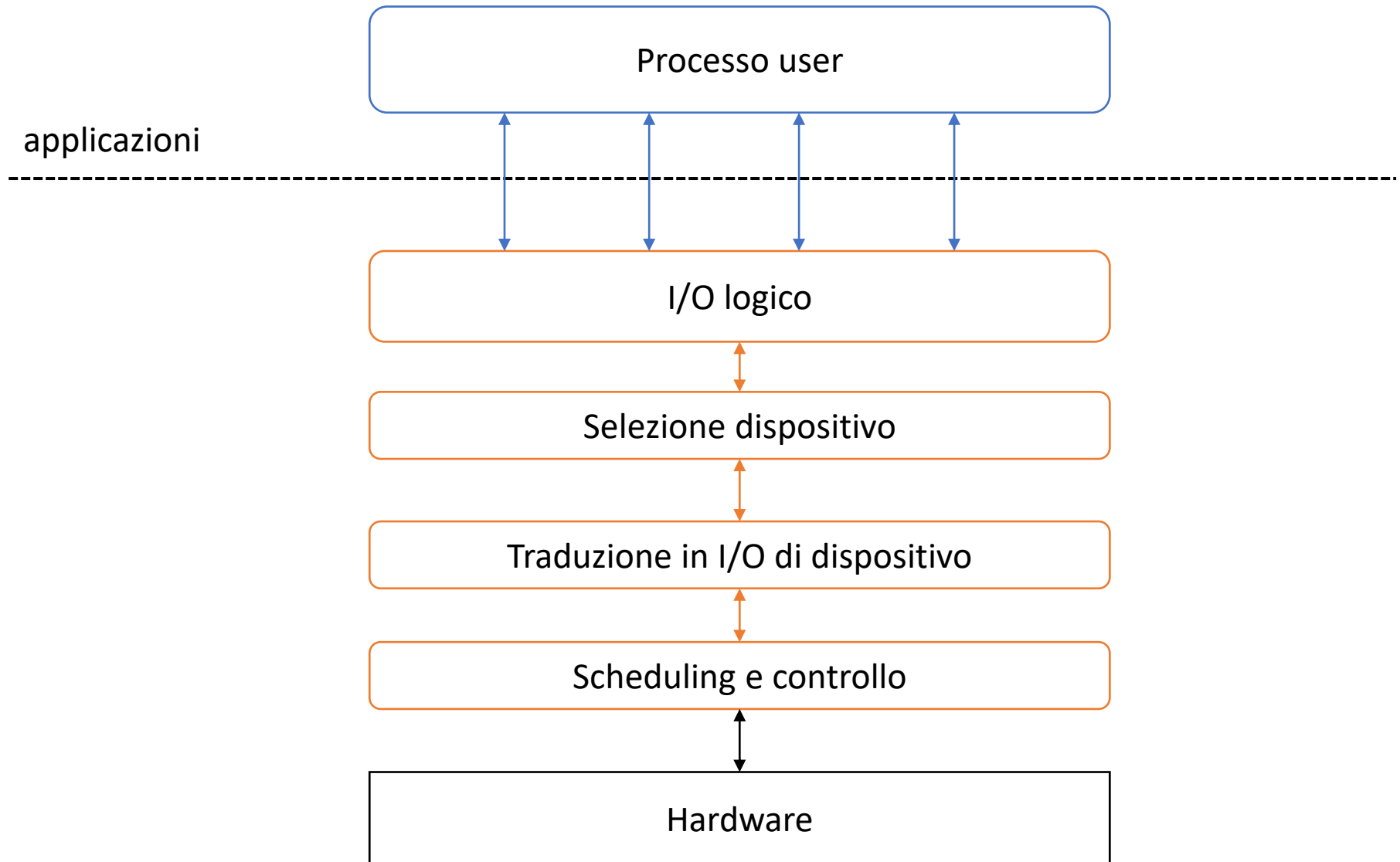
- Diversi tipi di dispositivi
  - Interazione con l'essere umano
  - Interazione con la macchina
  - Comunicazione tra macchine
- Caratteristiche differenti:
  - Applicazioni
  - Data rate
  - Controllo
  - Unità del trasferimento di dati
  - Rappresentazione del dato
  - Condizioni di errore
  - Gestione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

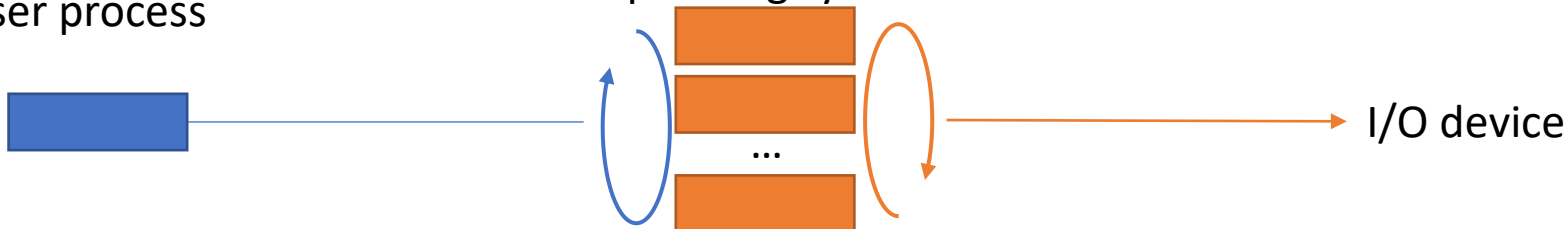
Operating system



- Circular buffer

User process

Operating system



# I/O scheduling

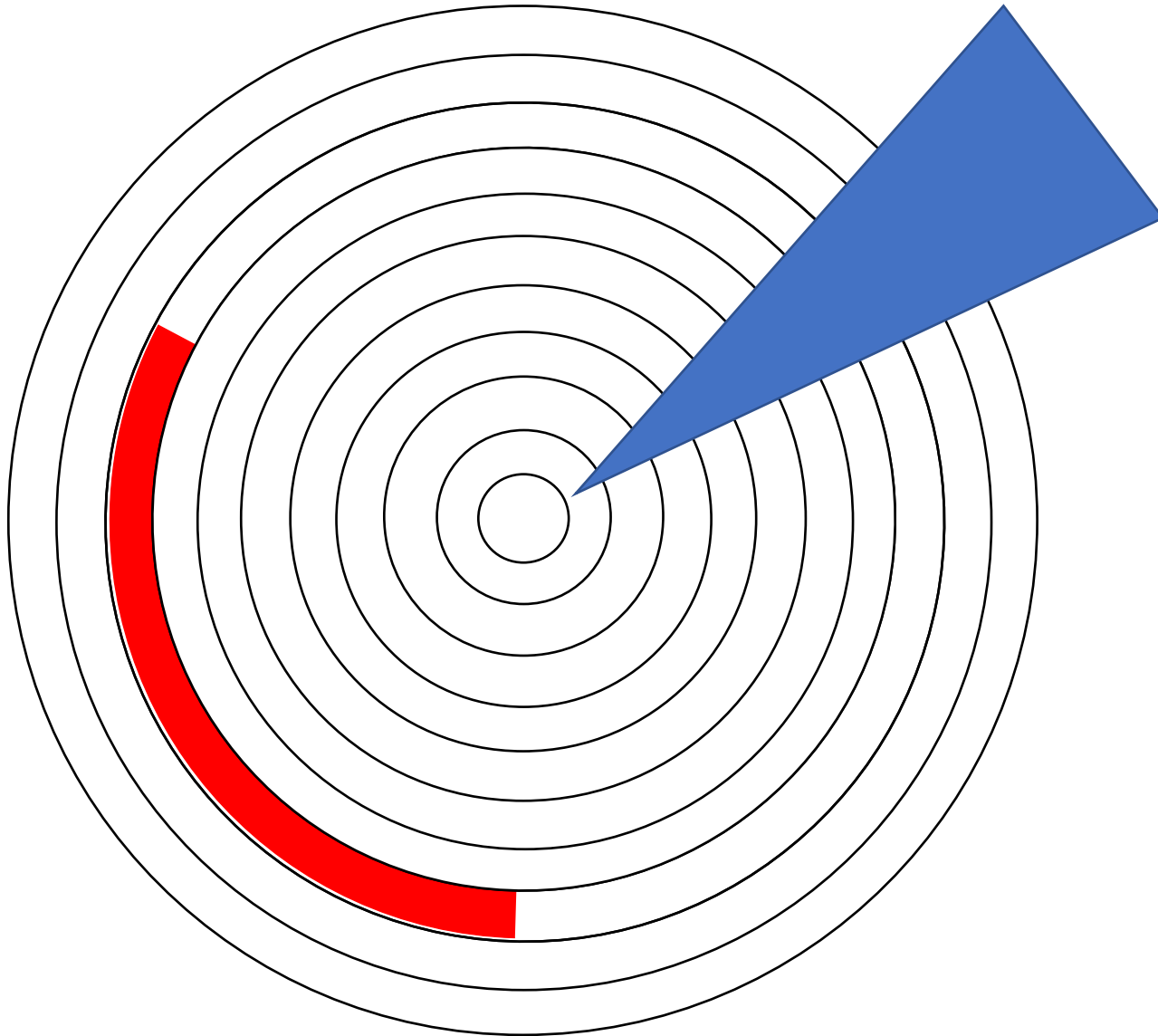
- Definisce la pianificazione per cui un dispositivo di I/O viene attivato per le sue operazioni
- In alcuni casi, mantenere l'ordine tra richieste di I/O e le effettive operazioni può essere necessario (e.g., terminale)
- FCFS non è necessariamente la soluzione più efficiente
  - Forte dipendenza dalle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

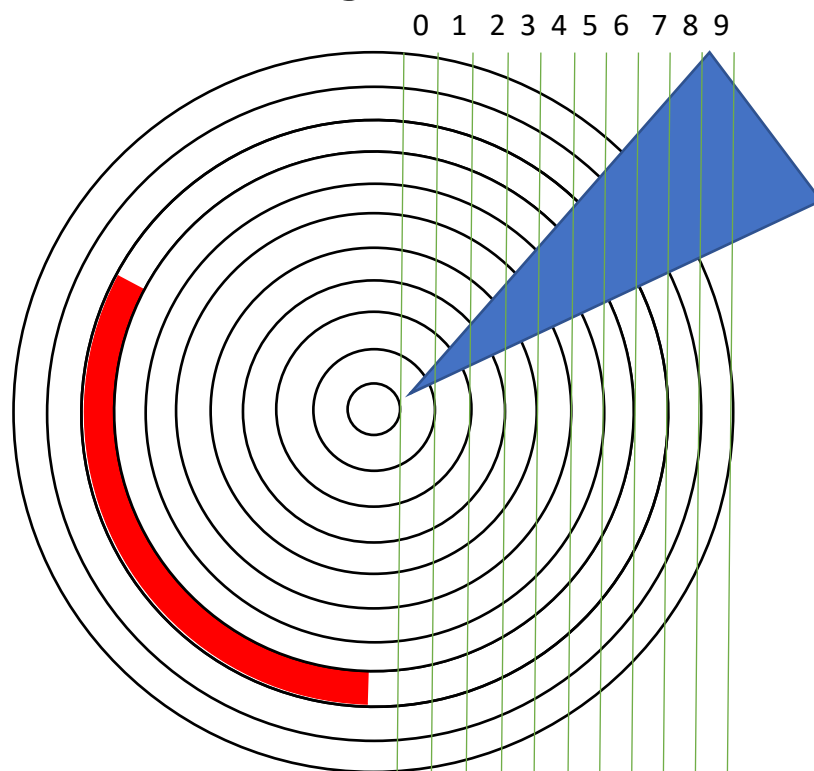
- Esempio
  - 500 settori per traccia
  - Average seek time = 4ms
  - Velocità di rotazione = 15000rpm = 250rps
  - Seek time per tracce adiacenti trascurabile
- Dati posizionati su 5 tracce consecutive
  - Tempo di rivoluzione = 4ms
  - Tempo di trasferimento =  $5 \times (\text{tempo di rivoluzione}) = 20\text{ms}$
  - 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 =  $20\text{ms} + 10\text{s} + 5\text{s} = 15.02\text{s}$

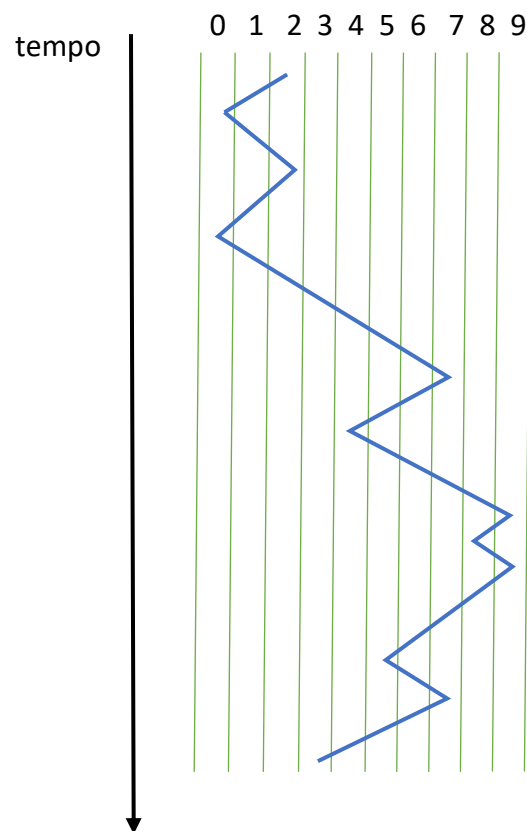
# Disk scheduling

- 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



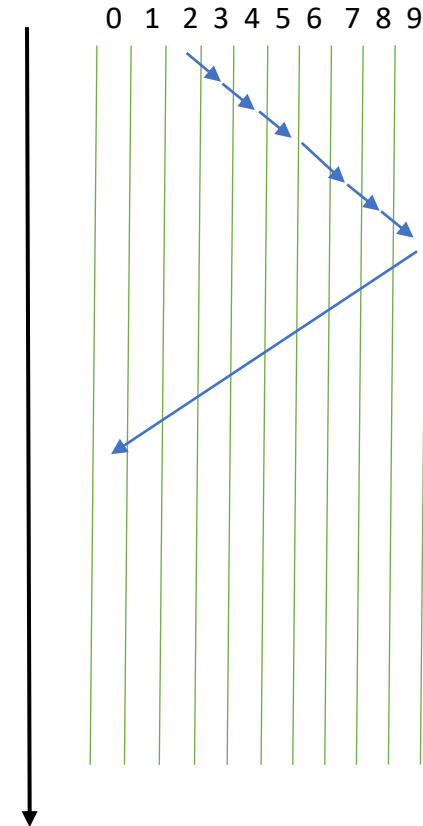
# Disk scheduling

- 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

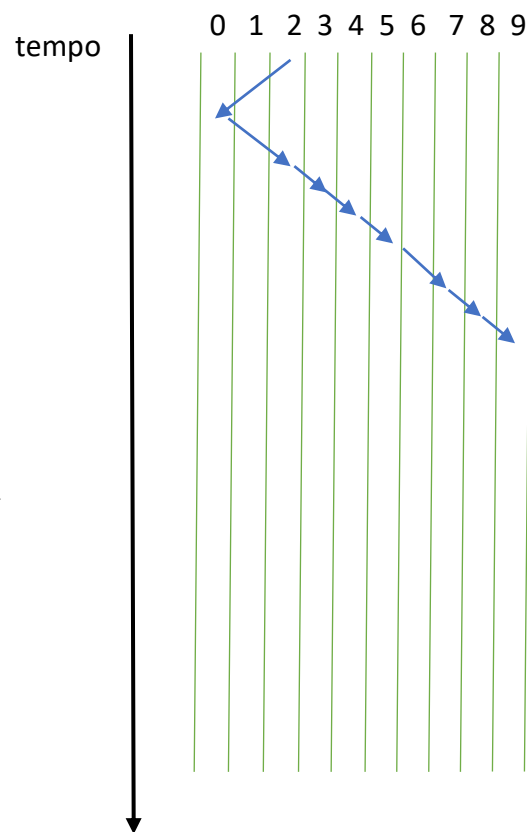
- Richieste:
  - 0,2,0,7,4,9,8,9,5,7,3
  - Posizione iniziale: 2
- Shortest-seek-time-first (SSTF)<sup>tempo</sup>
  - Seleziona la traccia successiva che minimizza il seek time
  - 2,3,4,5,7,7,8,9,9,0,0
  - Movimenti totali della testina: 16
  - Starvation
  - Non minimizza seek time





# 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