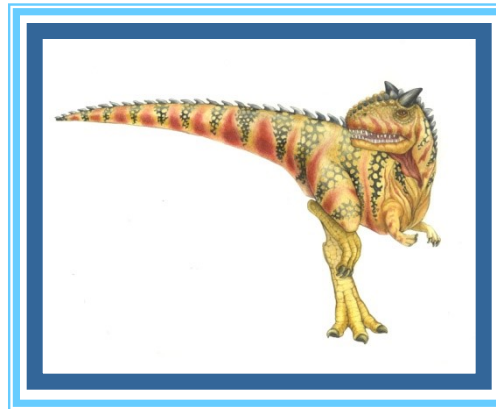


La memoria secondaria





Obiettivi

- ❖ Descrivere la struttura fisica dei diversi dispositivi di memoria secondaria e l'effetto della struttura di un dispositivo sul suo utilizzo
- ❖ Analizzare le prestazioni dei dispositivi di archiviazione di massa
- ❖ Valutare gli algoritmi di scheduling del disco
- ❖ Discutere i servizi forniti dal sistema operativo per la memorizzazione di massa





Sommario

- ❖ Struttura dei dispositivi di memorizzazione
- ❖ Scheduling dei dischi rigidi e dei dispositivi NVM
- ❖ Rilevamento e correzione di errori
- ❖ Gestione delle unità di memoria secondaria
- ❖ Gestione dell'area di swap
- ❖ Connessione dei dispositivi di memoria
- ❖ Strutture RAID





Memorie di massa e dispositivi di I/O

- ❖ I dispositivi collegati ad un calcolatore possono avere caratteristiche altamente variabili
- ❖ Rappresentano, nel loro insieme, la componente più lenta, voluminosa e variegata dell'elaboratore
- ❖ Il sistema operativo deve offrire alle applicazioni funzionalità che consentano l'accesso ai dispositivi mediante interfacce semplici e uniformi
- ❖ Deve inoltre garantire l'ottimizzazione dell'I/O, che costituisce il collo di bottiglia delle prestazioni del sistema di calcolo





Memorie di massa

- ❖ I dischi magnetici e i dispositivi di memoria non volatile (*nonvolatile memory*, **NVM**) costituiscono i supporti fondamentali di **memoria secondaria** nei computer attuali
 - **Dischi magnetici** (HDD – Hard Disk)
 - **Dischi a stato solido** (SSD – Solid State Disk)
- ❖ **Memorie terziarie**
 - Un solo drive, molti dispositivi rimovibili
 - ▶ Pen drive, memory card (flash)
 - ▶ Dischi ottici
 - Spazio di memorizzazione su cloud



hard disk esterni



dischi ottici



pen drive



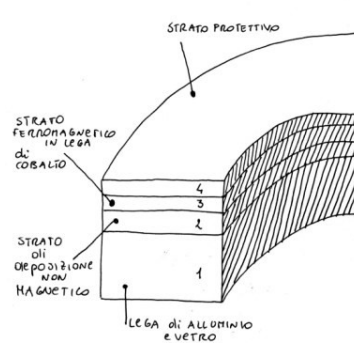
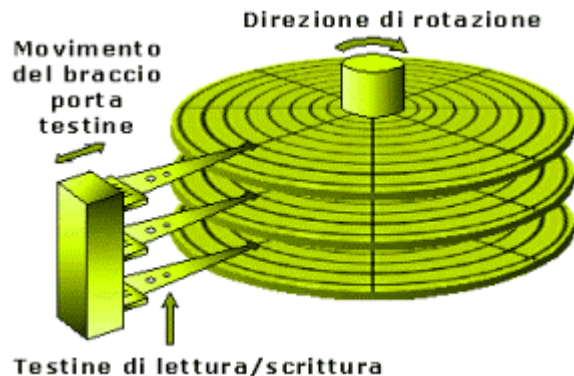
memory card





Dischi magnetici

- ❖ I **dischi magnetici** rappresentano ancora oggi un mezzo diffuso per la memorizzazione di massa
 - Costituiti da piatti, con un diametro che varia tra 1.8 e 3.5 pollici, sono rivestiti con materiale magnetico (ossido di ferro) ed erano originariamente in alluminio
 - La tecnologia attuale, viceversa, è orientata all'utilizzo del vetro:
 - ▶ Superficie più uniforme ⇒ maggiore affidabilità (errori di lettura/scrittura meno frequenti)
 - ▶ Più rigido
 - ▶ Più resistente agli urti
 - ▶ Permette di ridurre la distanza della testina dalla superficie





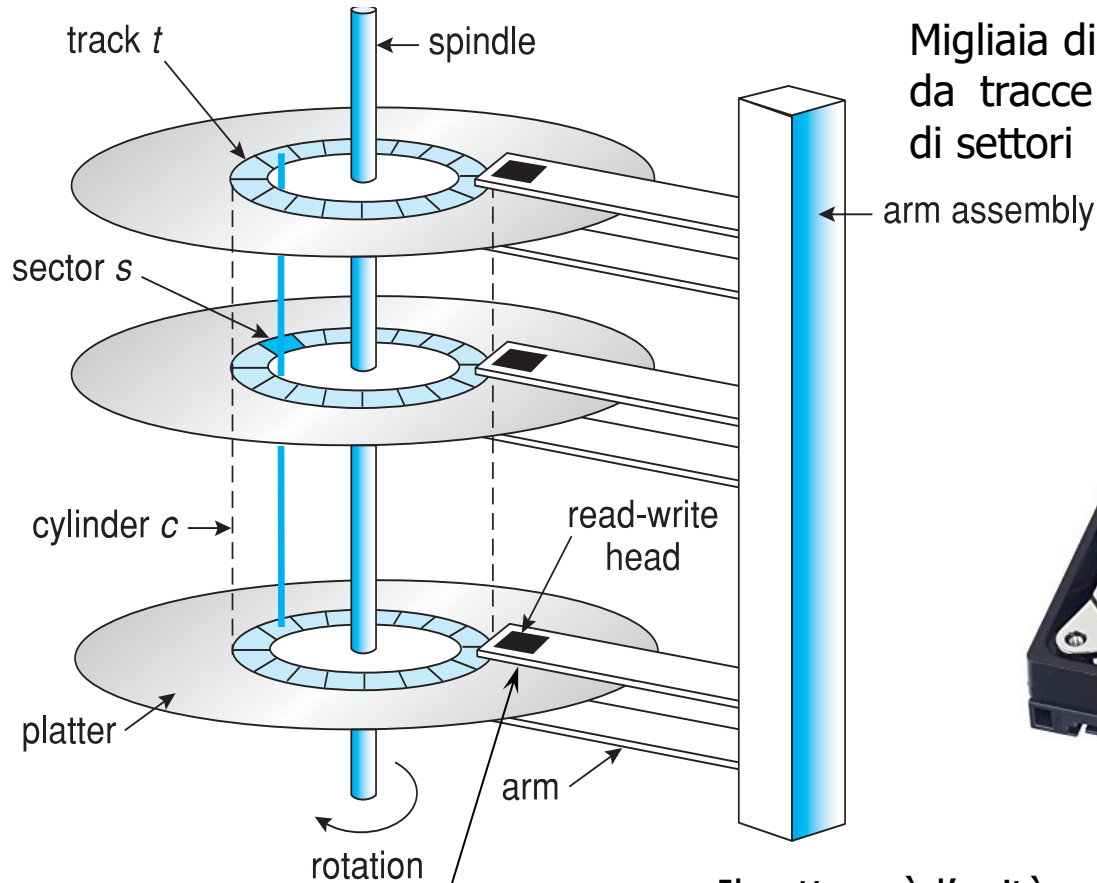
Dischi magnetici – Nomenclatura

- ❖ **Piatto** – un disco rigido si compone di uno o più dischi paralleli, in cui ogni superficie, detta “piatto” e identificata da un numero univoco, è destinata alla memorizzazione dei dati
- ❖ **Traccia** – su ogni piatto, vi sono numerosi anelli concentrici, detti tracce, ciascuno identificato da un numero univoco
- ❖ **Cilindro** – l’insieme di tracce poste alla stessa distanza dal centro e relative a tutti i dischi; corrisponde a tutte le tracce con lo stesso numero, ma giacenti su piatti diversi
- ❖ **Settore** – ogni traccia è suddivisa in settori circolari, ovvero in “spicchi” uguali, ciascuno identificato da un numero univoco (4KB)
- ❖ **Blocco** – l’insieme dei settori posti nella stessa posizione in tutti i piatti
- ❖ **Testina** – su ogni piatto è presente una testina di lettura/scrittura; la posizione di tale testina è solidale con tutte le altre sui diversi piatti (le testine sono attaccate al braccio del disco): se una testina è posizionata sopra una traccia, tutte le testine saranno posizionate sul cilindro a cui la traccia appartiene

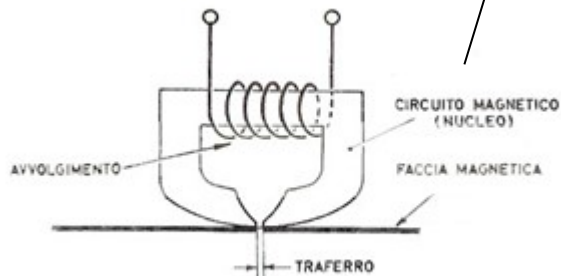




Dischi magnetici – Nomenclatura



Migliaia di cilindri concentrici costituiti da tracce che contengono centinaia di settori



Il settore è l'unità minima di trasferimento; originariamente costituito da 512 byte, dal 2010, la dimensione standard è 4KB





Dischi magnetici – Accesso

- ❖ I dischi ruotano ad una velocità compresa tra i 60 e i 250 giri al secondo
 - La **velocità di trasferimento** è la velocità con cui i dati fluiscono dall'unità a disco alla RAM (proporzionale alla velocità di rotazione)
 - Il **tempo di posizionamento** è il tempo necessario a spostare il braccio del disco in corrispondenza del cilindro desiderato (*seek time*), più il tempo necessario affinché il settore desiderato si porti sotto la testina (*latenza di rotazione*)
 - Il crollo della testina, normalmente sospesa su un cuscinetto d'aria di pochi micron (10^{-6} m), corrisponde all'impatto della stessa sulla superficie del disco
 - ▶ Di solito comporta la necessità di sostituire l'unità a disco
- ❖ I dischi possono essere rimovibili





Dischi magnetici – Lettura/Scrittura

- ❖ Memorizzazione e recupero dell'informazione tramite bobina conduttiva detta **testina** (head)
 - Durante la lettura/scrittura, la testina è stazionaria, mentre il disco ruota
 - **Scrittura**
 - ▶ la corrente, che fluisce nella bobina (nelle due possibili direzioni) produce un campo magnetico
 - ▶ le particelle aciculari dell'ossido di ferro si orientano in base al campo magnetico prodotto (0 e 1 memorizzati su disco)
 - **Lettura**
 - ▶ Il campo magnetico presente sul disco, muovendosi rispetto alla testina, induce corrente nella bobina





Caratteristiche degli hard disk

- ❖ Il raggio dei piatti variava, storicamente, fra 14 e 85 pollici
- ❖ I formati attualmente più comuni sono 3.5", 2.5", e 1.8"
- ❖ La capacità standard dei dischi (interni) si attesta fra 500GB e 20TB (fino a 30TB già annunciati da Seagate)
- ❖ Performance
 - Velocità di trasferimento (teorica): 6Gb/sec
 - Velocità di trasferimento (effettiva): 1Gb/sec
 - Seek time compreso fra 3msec e 12msec (9msec in media per i dischi presenti nei PC)
 - Tempo di latenza calcolato in base alla velocità di rotazione
 - ▶ $1 / (\text{RPM} / 60) = 60 / \text{RPM}$
 - Latenza media = $\frac{1}{2}$ giro

Velocità di rotazione [rpm]	Latenza media [msec]
15000	2
10000	3
7200	4.16
5400	5.55
4800	6.25



Prestazioni degli hard disk

- ❖ **Tempo di accesso medio** = seek time medio + latenza media
 - Per i dischi più veloci $\Rightarrow 3\text{msec} + 2\text{msec} = 5\text{msec}$
 - Per dischi lenti $\Rightarrow 9\text{msec} + 5.55\text{msec} = 14.55\text{msec}$
- ❖ Tempo medio di I/O = tempo medio di accesso + quantità di dati da trasferire/velocità di trasferimento + overhead
- ❖ Per esempio, per trasferire un blocco da 4KB su un disco con una velocità di rotazione pari a 7200 RPM, tempo medio di ricerca pari a 5msec, velocità di trasferimento di 1Gb/sec e con un overhead dovuto al controllore di 0.1msec, si ottiene:
 - Tempo di trasferimento = $4\text{KB}/1\text{Gb/s} = 0.031 \text{ msec}$
 - Tempo medio di I/O per un blocco da 4KB
 $= 5\text{msec} + 4.16\text{msec} + 0.1\text{msec} + \text{tempo di trasferimento}$
 $= 9.26\text{msec} + 0.031\text{msec} \cong 9.3\text{msec}$





Il primo hard disk commerciale



1956

Il computer IBM 305 RAMAC – che occupava una stanza di 9m×15m – includeva il primo disco magnetico nella storia dei calcolatori

- 5 milioni di caratteri da 7 bit più parità
- 50 dischi da 24"
- Tempo di accesso circa 600 msec





Dispositivi di memoria non volatile – 1

- ❖ Talvolta inseriti in chassis simili agli HDD, e perciò denominati **dischi a stato solido** (SSD), sono costituiti da un controllore e da diversi chip di memoria NAND flash
- ❖ Altre forme includono unità USB (pen drive, unità flash) e DRAM dotate di batteria di backup
- ❖ Negli smartphone, le NVM sono montate sulla scheda madre e rappresentano il dispositivo primario di archiviazione
- ❖ Le NVM possono essere più affidabili degli HDD
- ❖ Hanno costo al MB più elevato
- ❖ Possono avere vita più breve
- ❖ Sono molto più veloci e consumano meno energia
 - Nessuna parte meccanica in movimento, quindi nessun tempo di ricerca o latenza di rotazione e maggiore resistenza a sollecitazioni e urti (minor rumore e minore dispersione termica)





Dispositivi di memoria non volatile – 2

- ❖ Le caratteristiche dei semiconduttori **NAND** aprono a nuove sfide per l'affidabilità
- ❖ Letture e scritture con granularità di "pagina" (analogo del settore)
 - Impossibilità di cancellazione per "sovra-scrittura"
 - Il contenuto della pagina deve prima essere cancellato e le cancellazioni avvengono per "blocchi" (della dimensione di diverse pagine) ⇒ operazione costosa
 - Le NAND si deteriorano ad ogni ciclo di cancellazione e dopo un dato numero di cicli (dipendente dal particolare supporto), le celle non sono più in grado di mantenere l'informazione
 - Durata misurata in numero di scritture al giorno (DWPD, *Drive Writes per Day*)
 - ▶ Su una NAND da 1 TB di classe 5DWPD si possono scrivere 5 TB al giorno senza errori per il periodo di garanzia





Dispositivi di memoria non volatile – 3

- ❖ Algoritmi di gestione ottimizzata — attuati per arginare l'usura — implementati dal controllore e non di competenza del SO
- ❖ Il sistema operativo “si limita” a leggere e scrivere blocchi logici, mentre è il controller che si occupa dell'effettiva realizzazione delle operazioni

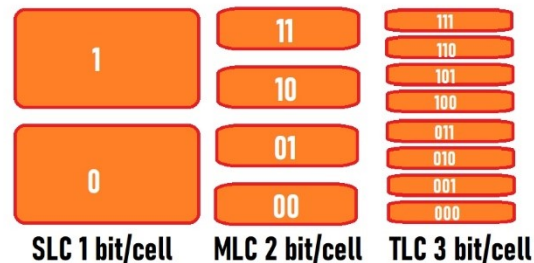




Memorie flash

- ❖ Sono memorie a stato solido (EEPROM)
 - In una memoria flash le informazioni vengono registrate in un array di **Floating Gate MOSFET**, una tipologia di transistor in grado di mantenere la carica elettrica per un tempo lungo
 - ▶ Ogni transistor costituisce una "cella di memoria" che conserva il valore di un bit
 - ▶ Le flash attuali utilizzano celle multilivello che permettono di registrare il valore di più bit in un solo transistor

Types of NAND Flash Memory





NAND flash – 1

- ❖ Senza sovrascrittura, i blocchi sono costituiti da un mix di pagine valide e non valide
- ❖ Per tenere traccia dei blocchi logici validi, il controllore mantiene la tabella **FTL** (*Flash Translation Layer*)
- ❖ Implementa anche la *garbage collection* per liberare spazio
 - Per cancellare dati non validi, un controller SSD di norma deve prima copiare tutti i dati validi (quelli che dovranno essere ancora utilizzati in futuro) nelle pagine vuote di un altro blocco
 - Quindi deve cancellare tutte le celle del blocco da liberare (eliminando sia i dati da cancellare che quelli copiati per poter essere successivamente riutilizzati) e solo a quel punto può iniziare a scrivere nuovi dati nel blocco che è stato così liberato





NAND flash – 2

- ❖ Normalmente, si mantiene un **overprovisioning** (7–20% di pagine) per fornire spazio di lavoro per la garbage collection
- ❖ Ogni cella ha durata di vita limitata, quindi il livello di usura deve essere mantenuto uniforme
 - L'overprovisioning è anche funzionale a garantire un numero adeguato di celle sostituibili a quelle che raggiungono il limite del ciclo di programmazione e cancellazione, così da prolungare la vita dello stesso SSD

valid page	valid page	invalid page	invalid page
invalid page	valid page	invalid page	valid page

Blocco NAND con pagine valide e non valide





Memoria volatile

- ❖ DRAM usata frequentemente come dispositivo di archiviazione di massa
 - Tecnicamente, non si può definire archiviazione secondaria perché volatile, ma può contenere file system, da utilizzare come storage secondario molto veloce
- ❖ Infatti, le unità RAM possono essere utilizzate come dispositivi a blocchi non formattati, ma più spesso contengono un file system
 - Supportate dai principali sistemi operativi
 - ▶ Linux: `/dev/ram`, `/tmp` (con file system temporaneo)
 - ▶ MAC OS: `diskutil` (creazione di dischi RAM)





Connessione – 1

- ❖ L'unità a disco è connessa al calcolatore per mezzo del bus di I/O
 - Diversi tipi: ATA (Advanced Technology Attachment), SATA (Serial ATA), USB (Universal Serial Bus), SAS (Serial Attached SCSI), FC (Fiber Channel)
- ❖ I bus standard possono essere troppo lenti per gli SSD
 - Collegamento al bus PCI di sistema con tecnologia NVM express (NVMe)
- ❖ Il trasferimento di dati in un bus è eseguito da speciali unità di elaborazione, dette **controllori**: gli **adattatori** sono i controllori posti all'estremità del bus relativa al calcolatore, i **controllori dei dischi** sono incorporati in ciascuna unità a disco





Connessione – 2

- ❖ Per eseguire un'operazione di I/O, si inserisce il comando opportuno nell'adattatore, generalmente mediante porte di I/O mappate in memoria
- ❖ L'adattatore invia il comando al controllore del disco, che gestisce l'hardware dell'unità, per portare a termine il compito richiesto
- ❖ Il trasferimento dei dati nell'unità a disco avviene tra la superficie del disco e la cache incorporata nel controllore
- ❖ Il trasferimento verso l'host avviene, a velocità elevate, tramite DMA





Mappatura degli indirizzi – 1

- ❖ Le unità a disco vengono indirizzate come giganteschi vettori monodimensionali di **blocchi logici**, dove il blocco logico rappresenta la minima unità di trasferimento
 - I blocchi logici sono creati all'atto della formattazione di basso livello
- ❖ L'array di blocchi logici viene mappato sequenzialmente nei settori del disco o sulle pagine di un blocco di una NVM
- ❖ Per esempio:
 - Il settore 0 è il primo settore della prima traccia del cilindro più esterno
 - La corrispondenza prosegue ordinatamente lungo la prima traccia, quindi lungo le rimanenti tracce del primo cilindro, e così via, di cilindro in cilindro, dall'esterno verso l'interno
 - I settori danneggiati vengono esclusi dall'operazione di mappatura e sostituiti con settori di riserva collocati in altre parti della stessa unità





Mappatura degli indirizzi – 2

- ❖ La mappatura da settori/pagine a blocchi logici trasforma tuple, $\langle \text{cilindro}, \text{traccia}, \text{settore} \rangle$ o $\langle \text{chip}, \text{blocco}, \text{pagina} \rangle$, in indirizzi progressivi lineari – **LBA** per *Logical Block Address* – più facili da utilizzare dal punto di vista algoritmico
- ❖ Nella pratica, una traduzione diretta è difficile; per esempio, nel caso dei dischi magnetici, hanno impatto significativo:
 - La presenza di settori difettosi
 - Il diverso numero di settori per traccia
 - La gestione interna al controllore (ed estranea al SO) dell'operazione di mappatura
- ❖ Tuttavia, gli algoritmi di gestione degli HDD tendono a presumere che gli indirizzi logici siano relativamente correlati agli indirizzi fisici, ovvero ad associare la crescita dell'indirizzo logico con la crescita dell'indirizzo fisico





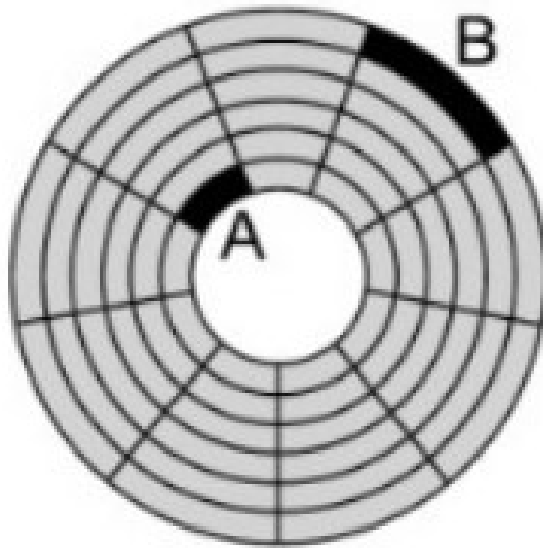
Struttura logica del disco magnetico – 1

- ❖ **CLV** – Constant Linear Velocity: densità dei bit per traccia uniforme
 - Tracce più lontane dal centro del disco sono più lunghe e contengono un maggior numero di settori (fino al 40% in più rispetto alle tracce vicine al centro di rotazione)
 - La velocità di rotazione aumenta spostandosi verso l'interno ($v = \omega r$), per mantenere costante la velocità lineare e, quindi, la quantità di dati che passano sotto le testine nell'unità di tempo
 - CD e DVD
 - Talvolta, si ha un'unica traccia a spirale
- ❖ **CAV** – Constant Angular Velocity: velocità di rotazione costante
 - La densità dei bit decresce dalle tracce interne alle più esterne per mantenere costante la quantità di dati che passano sotto le testine nell'unità di tempo
 - Dischi magnetici

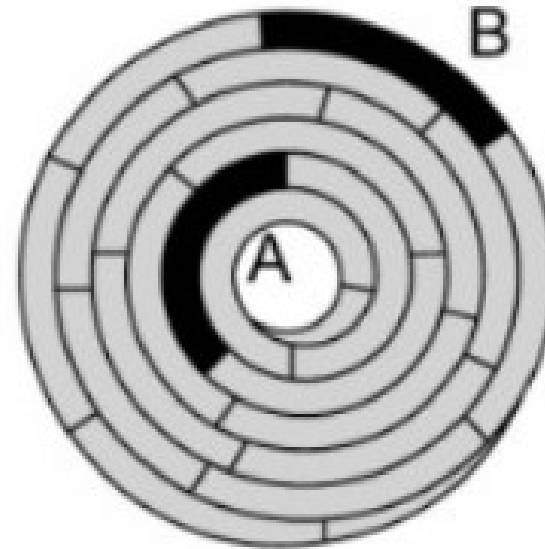




Struttura logica del disco magnetico – 2



Disco CAV a tracce concentriche



Disco CLV con un'unica traccia a spirale





Scheduling del disco – 1

- ❖ Il SO è responsabile dell'uso efficiente dell'hardware: per i dischi magnetici ciò significa garantire tempi di accesso contenuti e ampiezze di banda elevate
- ❖ Il *tempo di accesso* al disco si può scindere in due componenti principali:
 - *Tempo di ricerca* (*seek time*) — è il tempo impiegato per spostare la testina sul cilindro che contiene il settore desiderato
 - *Latenza di rotazione* (*rotational latency*) — è il tempo necessario perché il disco ruoti fino a portare il settore desiderato sotto la testina
- ❖ Per migliorare le prestazioni si può intervenire solo sul tempo di ricerca e si tenta quindi di minimizzarlo
- ❖ Seek time \propto distanza di spostamento fra le tracce





Scheduling del disco – 2

- ❖ L'ampiezza di banda del disco è il numero totale di byte trasferiti, diviso per il tempo trascorso fra la prima richiesta e il completamento dell'ultimo trasferimento
- ❖ Quando un processo (utente o di sistema) deve effettuare un'operazione di I/O relativa ad un'unità a disco, effettua una chiamata al SO
- ❖ La richiesta di servizio contiene:
 - Specifica del tipo di operazione (immissione/emissione di dati)
 - Indirizzo su disco relativamente al quale effettuare il trasferimento
 - Indirizzo nella memoria relativamente al quale effettuare il trasferimento
 - Numero di byte da trasferire





Scheduling del disco – 3

- ❖ Una richiesta di accesso al disco può venire soddisfatta immediatamente se unità a disco e controller sono disponibili; altrimenti la richiesta deve essere aggiunta alla coda delle richieste inevase per quella unità
- ❖ In passato, il SO era responsabile della gestione delle code, ovvero della schedulazione delle unità a disco
 - **Scheduling del disco** ora integrato ai controller dei dispositivi di archiviazione, che traducono direttamente gli indirizzi di blocco logico e gestiscono le code
 - Tuttavia, lo scheduling deve mantenersi equo e tempestivo e garantire il raggruppamento di accessi che appaiono in sequenza, poiché si ottengono prestazioni migliori con I/O "sequenziali"





Scheduling del disco – 4

- ❖ Gli algoritmi di scheduling del disco verranno testati sulla coda di richieste per i cilindri (0–199):
98, 183, 37, 122, 14, 124, 65, 67
La testina dell'unità a disco è inizialmente posizionata sul cilindro 53

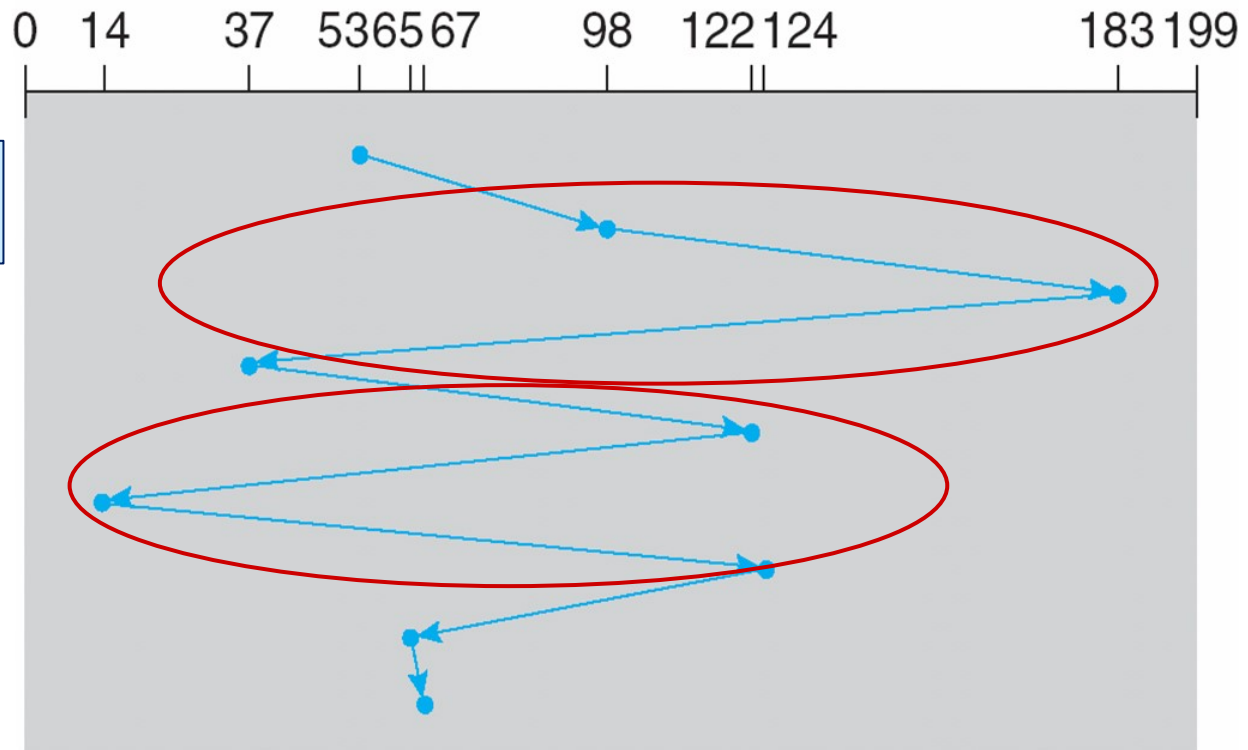




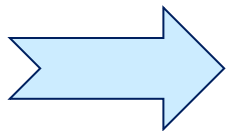
First-Come-First-Served

- ❖ **FCFS** – *First Come First Served* – è un algoritmo intrinsecamente equo
- ❖ Si produce un movimento totale della testina pari a 640 cilindri

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



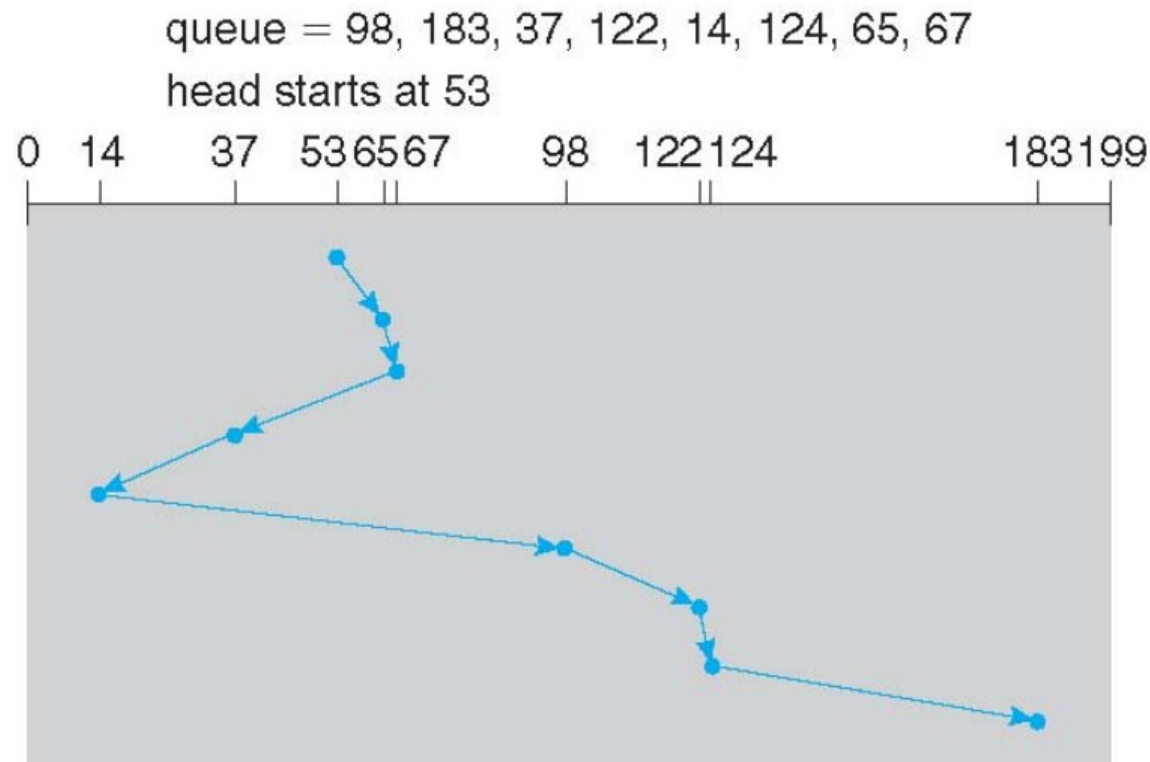
Si possono produrre grandi oscillazioni





Shortest-Seek-Time-First

- ❖ **SSTF** – *Shortest Seek Time First* – opta per servire le richieste minimizzando il movimento radiale della testina
- ❖ Non è necessariamente la scelta ottima e si può avere starvation e difficoltà nel servire le tracce periferiche



- ❖ Si ha un movimento totale pari a 236 cilindri





SCAN – 1

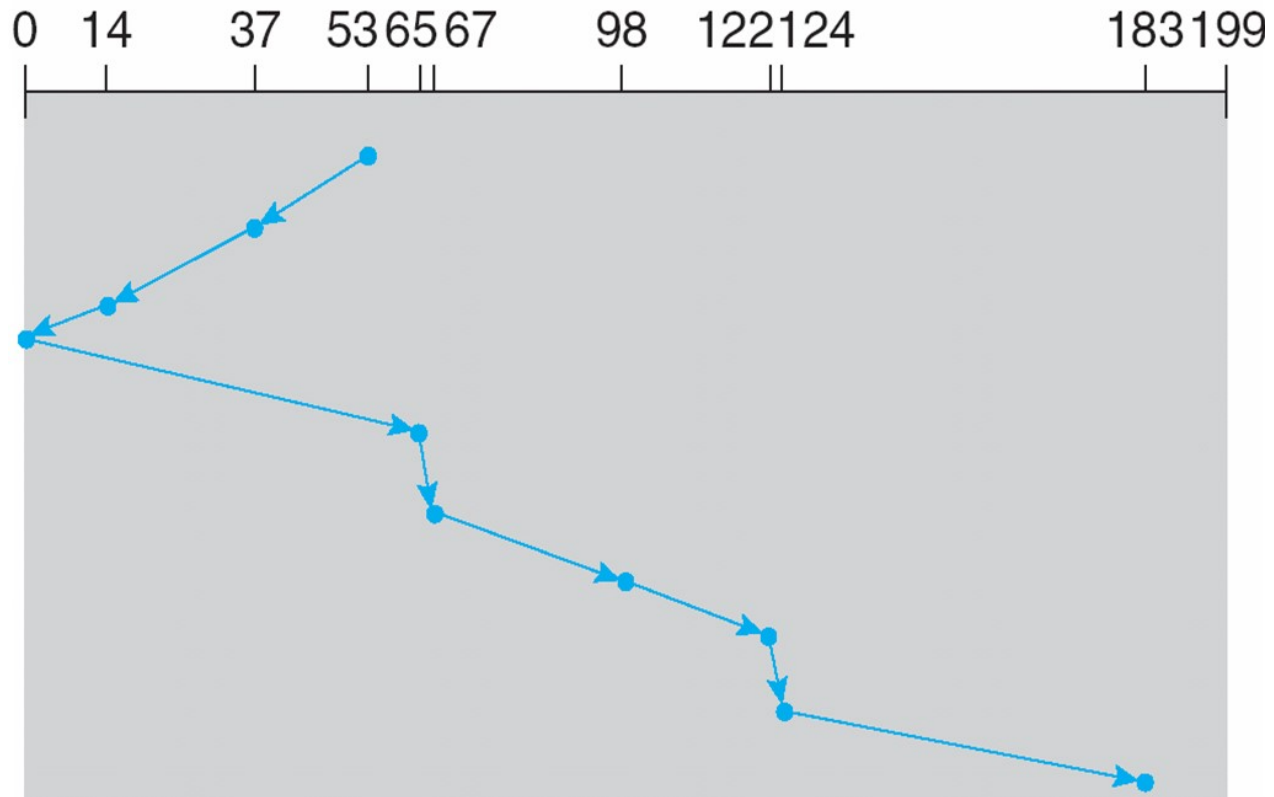
- ❖ Il braccio della testina si muove da un estremo all'altro del disco, servendo sequenzialmente le richieste; giunto ad un estremo inverte la direzione di marcia e, conseguentemente, l'ordine di servizio
- ❖ È chiamato anche **algoritmo dell'ascensore**
- ❖ Se gli accessi sono distribuiti uniformemente, quando la testina inverte il proprio movimento, la maggior densità di richieste si ha all'estremo opposto del disco
 - Tali richieste avranno anche i tempi più lunghi di attesa di servizio





SCAN – 2

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



❖ Si ha un movimento totale pari a 236 cilindri





C-SCAN – 1

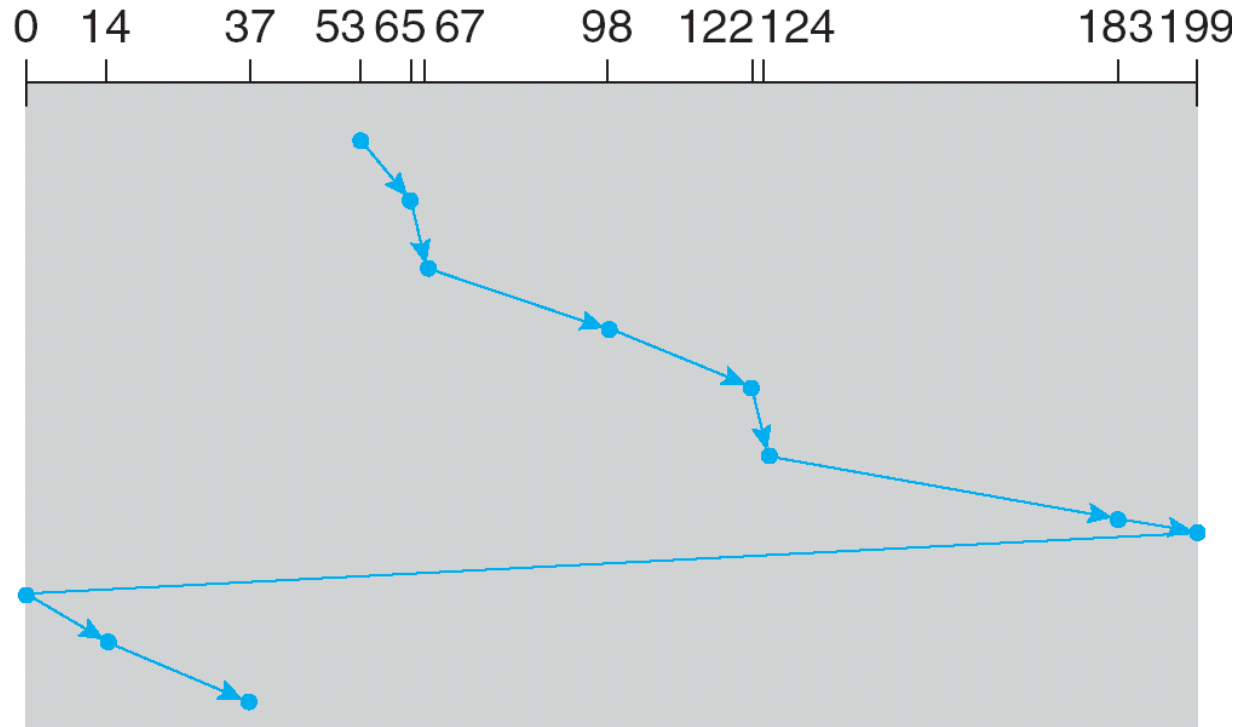
- ❖ Garantisce un tempo di attesa più uniforme rispetto a SCAN
- ❖ La testina si muove da un estremo all'altro del disco servendo sequenzialmente le richieste
- ❖ Quando raggiunge l'ultimo cilindro ritorna immediatamente all'inizio del disco, senza servire richieste durante il viaggio di ritorno
- ❖ Considera i cilindri come organizzati secondo una lista circolare, con l'ultimo cilindro adiacente al primo





C-SCAN – 2

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



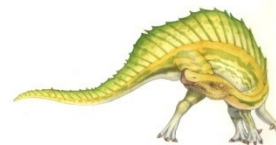
❖ Si ha un movimento totale pari a 383 cilindri





Scelta di un algoritmo di scheduling

- ❖ SCAN e C-SCAN forniscono buone prestazioni in sistemi che utilizzano intensamente le unità a disco
- ❖ Le prestazioni dipendono comunque dal numero e dal tipo di richieste
- ❖ Vengono normalmente implementati nella forma LOOK (e C-LOOK) ovvero non percorrendo tutto il disco, ma raggiungendo l'ultima traccia (in entrambe le direzioni) su cui è stata fatta richiesta di accesso
- ❖ Le richieste di I/O per l'unità a disco possono essere influenzate dal metodo di allocazione di file e directory





Esempio 1

- ❖ Un disco ha C cilindri. Un'operazione di ricerca richiede 6msec per lo spostamento tra un cilindro e l'altro, la latenza rotazionale media è di 10msec ed il tempo di trasferimento è di 0.25msec per blocco.
 - Quanto tempo è necessario per leggere un file costituito da 20 blocchi e memorizzato in modo tale che blocchi logicamente contigui nel file distino mediamente 13 cilindri l'uno dall'altro sul disco?
 - Quanto tempo è necessario per leggere un file con 100 blocchi mediamente distanti 2 cilindri?

❖ Soluzione

- Il tempo necessario per la lettura del file da 20 blocchi è

$$T_{L20} = 20 \times [13 \times 6 + 10 + 0.25] \text{ msec} = 1765 \text{ msec}$$

- Il tempo necessario per la lettura del file da 100 blocchi è

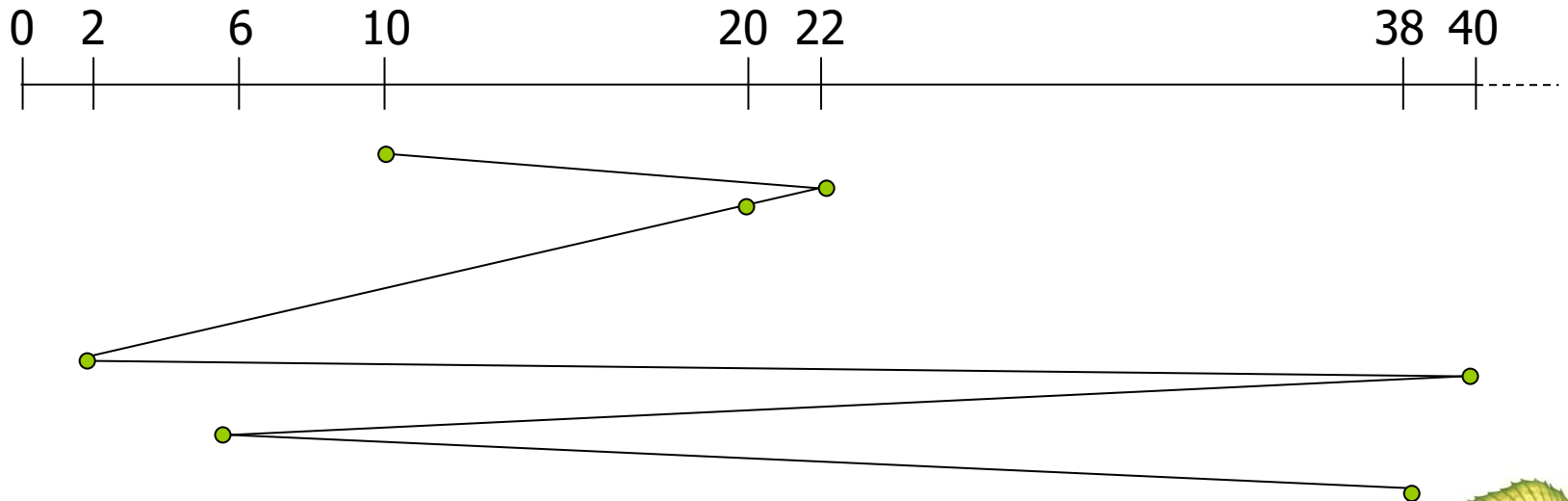
$$T_{L100} = 100 \times [2 \times 6 + 10 + 0.25] \text{ msec} = 2225 \text{ msec}$$



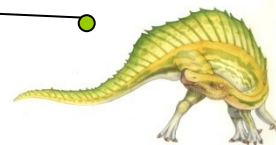


Esempio 2

- ❖ Al driver di un disco arrivano, nell'ordine, richieste per i cilindri 10,22,20,2,40,6,38. Uno spostamento da un traccia a quella adiacente richiede 6msec. Si stabilisca quanto tempo è necessario per servire le richieste con FCFS e C-LOOK (crescente). Si assuma, per tutti i casi, che il braccio si trovi inizialmente posizionato sul cilindro 10.
- ❖ **Soluzione FCFS**



$$T = 6 \times (12 + 2 + 18 + 38 + 34 + 32) \text{ msec} = 816 \text{ msec}$$

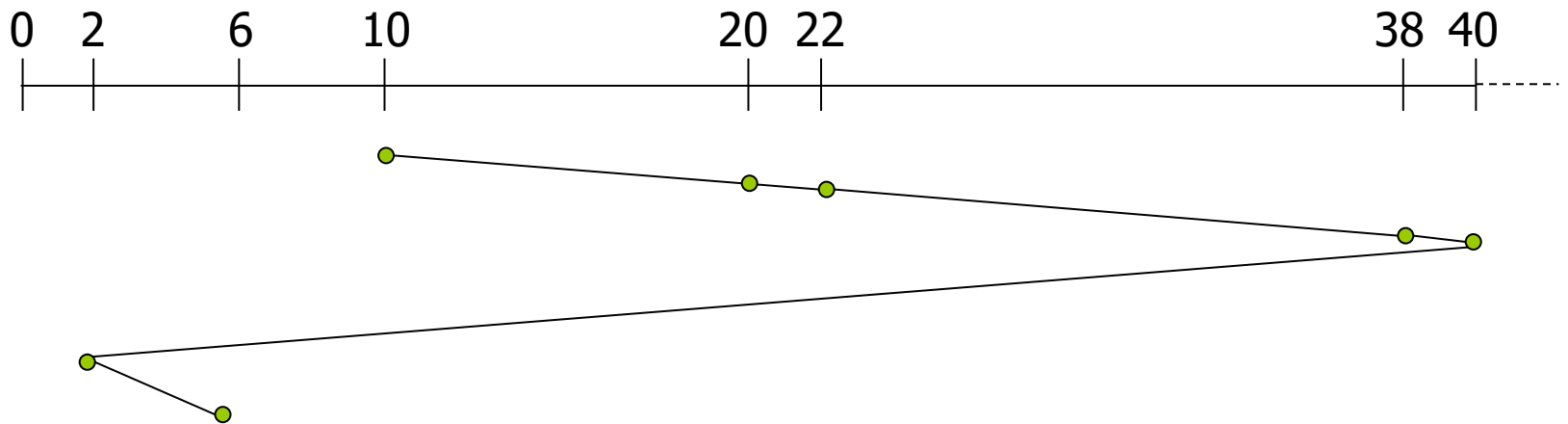




Esempio 2 (cont.)

❖ Soluzione C-LOOK crescente

10,22,20,2,40,6,38



$$T = 6 \times (10 + 2 + 16 + 2 + 38 + 4) \text{ msec} = 432 \text{ msec}$$





Esempio 3

❖ Un disco ha un tempo di seek di 0.5msec per ogni cilindro attraversato, un tempo di rotazione di 6msec e un tempo di trasferimento dei dati di un settore di $12\mu\text{sec}$. Inoltre, la testina è attualmente posizionata sul cilindro 12. Supponendo che al tempo attuale arrivino contemporaneamente le seguenti richieste di lettura di settori:

- 5 settori nel cilindro 14
- 3 settori nel cilindro 11
- 4 settori nel cilindro 19
- 1 settore nel cilindro 2
- 6 settori nel cilindro 31

Calcolare il tempo di completamento delle richieste nel caso in cui venga utilizzato l'algoritmo dell'ascensore (in modalità LOOK e con direzione iniziale verso i cilindri con numerazione crescente).





Esempio 3 (cont.)

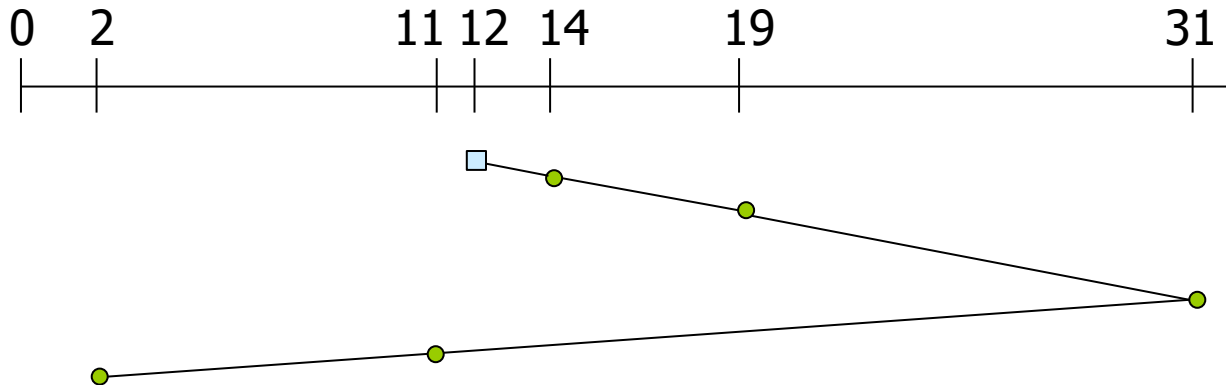
❖ Soluzione

- 5 settori nel cilindro 14
- 3 settori nel cilindro 11
- 4 settori nel cilindro 19
- 1 settore nel cilindro 2
- 6 settori nel cilindro 31

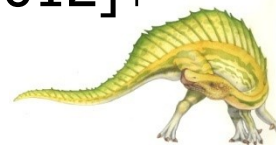
Tempo di seek = $500\mu\text{sec}$

Tempo di latenza (mezzo giro) = $3000\mu\text{sec}$

Tempo di trasferimento = $12\mu\text{sec}$



$$\begin{aligned} T &= \{[2 \times 500 + 5 \times 3012] + [5 \times 500 + 4 \times 3012] + [12 \times 500 + 6 \times 3012] + \\ &\quad [20 \times 500 + 3 \times 3012] + [9 \times 500 + 3012]\} \mu\text{sec} \\ &= \{24000 + 57228\} \mu\text{sec} = 81.228 \text{msec} \end{aligned}$$





Esempio 4

- ❖ Nella coda delle richieste di una unità a disco composta da 200 tracce, si trovano, nell'ordine, le seguenti richieste di accesso ai blocchi:

39700 304 115 2600 2120 270 321 0 760 20000

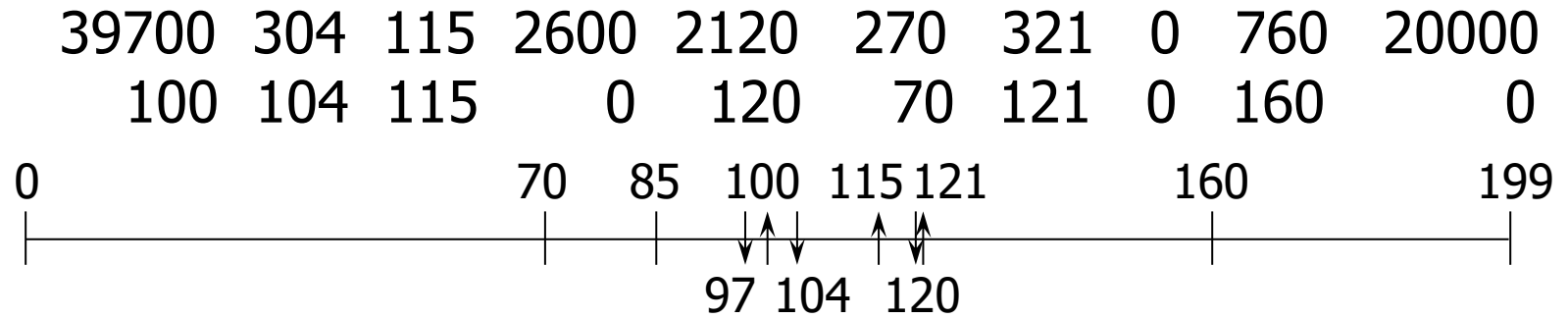
Si supponga che il blocco i -esimo sia memorizzato alla traccia $i \bmod 200$. La testina ha eseguito l'ultimo movimento portandosi dalla traccia 85 alla traccia 97. Si ipotizzi che lo spostamento da una traccia alla successiva richieda un tempo medio pari a $40\mu\text{sec}$ per traccia, che l'inversione della direzione di movimento della testina richieda mediamente $80\mu\text{sec}$ e che la velocità di rotazione sia pari a 7200 rpm. Determinare il tempo totale di servizio con scheduling C-SCAN considerando trascurabile il tempo di trasferimento.



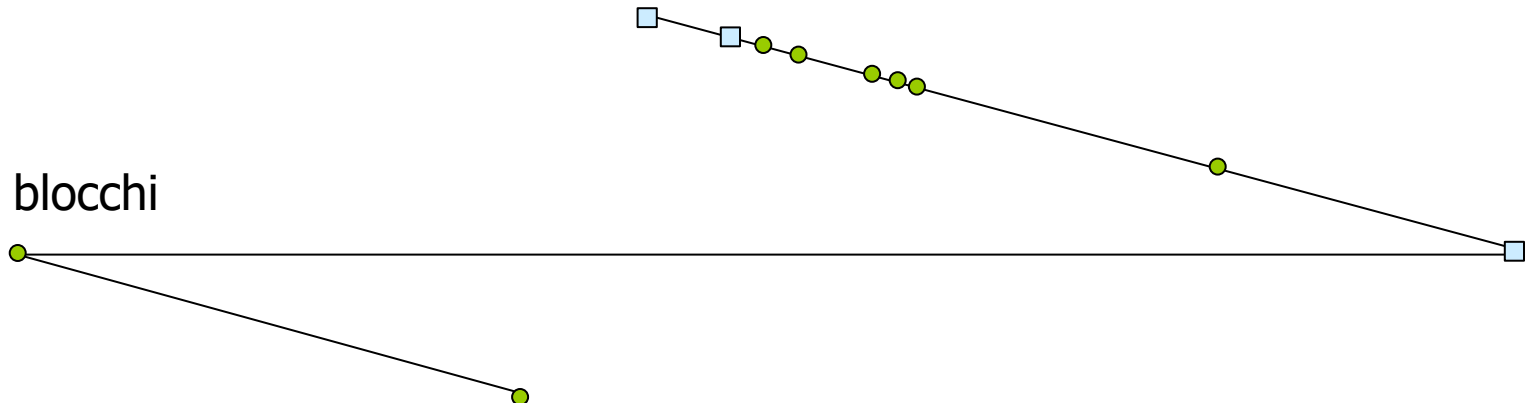


Esempio 4 (cont.)

❖ Soluzione



3 blocchi



$$T_{\text{Lat}} = 0.5 \times (60/7200) \text{sec} \approx 4.16 \text{msec} = 4160 \mu\text{sec}$$

$$T_{\text{Tot}} = [(3 \times 40 + 4160) + (4 \times 40 + 4160) + (11 \times 40 + 4160) + (5 \times 40 + 4160) + (40 + 4160) + (39 \times 40 + 4160) + (39 \times 40 + 80) + (199 \times 40 + 3 \times 4160) + 80 + (70 \times 40 + 4160)] \mu\text{sec}$$

$$= (371 \times 40 + 10 \times 4160 + 2 \times 80) \mu\text{sec} = 56.6 \text{msec}$$





Scheduling su NVM

- ❖ Nelle unità NVM, dove non esistono parti mobili, si utilizza di solito una politica FCFS
 - L'unica ottimizzazione possibile riguarda il servizio combinato di richieste (di scrittura) relative a indirizzi logici adiacenti
- ❖ Tuttavia, il vantaggio dei dispositivi NVM è meno sensibile in caso di accesso sequenziale — contrariamente al tempo di ricerca sugli HDD che, in questo caso, è ridotto al minimo
 - Prestazioni equivalenti o anche peggiori in caso di elevata usura del dispositivo
- ❖ Problema dell'amplificazione di scrittura, quando si attivano operazioni aggiuntive per la garbage collection





Rilevamento e correzione di errori

- ❖ Il rilevamento degli errori determina se si è verificato un problema (ad esempio un *bit flipping*)
 - A fronte del verificarsi di un errore, il sistema può interrompere l'operazione prima che l'errore venga propagato
 - Rilevazione eseguita frequentemente tramite **bit di parità**
- ❖ La parità è una forma di *checksum* che utilizza l'aritmetica modulare per calcolare, archiviare, confrontare valori su parole a lunghezza fissa
- ❖ Un altro metodo di rilevamento degli errori comune nelle reti è il **controllo di ridondanza ciclica** (CRC) che utilizza una funzione hash per rilevare errori su più bit
- ❖ Il codice di correzione degli errori (ECC) non solo rileva, ma può correggere alcuni errori
 - Errori soft correggibili, errori hard rilevati ma non corretti
 - Utilizzati a livello di settore/pagina





Gestione dell'unità a disco – 1

- ❖ **Formattazione di basso livello o fisica**
 - Si suddivide il disco in settori, che possono essere letti e scritti dal controllore del disco
 - Dimensione standard pari a 4KB
 - Nel caso di NVM devono essere inizializzate le pagine e creata la tabella **FTL** (Flash Translation Layer)
 - In entrambi i casi, la formattazione di basso livello inserisce nel dispositivo una speciale struttura dati per ogni "blocco" di memoria:
 - ▶ Intestazione, dati, coda
 - ▶ L'intestazione e la coda contengono informazioni (numero settore/pagina, codice ECC) ad uso del controllore
- ❖ La formattazione fisica è eseguita dal costruttore dell'unità come parte del processo produttivo





Gestione dell'unità a disco – 2

- ❖ Per poter impiegare un dispositivo per memorizzare i file, il SO deve mantenere le proprie strutture dati sul disco (HDD/SSD)
 - Si partiziona il dispositivo in uno o più gruppi di cilindri/pagine, ognuno dei quali rappresenta un “disco logico”
 - **Formattazione logica** o “creazione di un file system”
 - ▶ Strutture dati del SO per la descrizione dello spazio libero/occupato e creazione di una directory iniziale vuota
 - Per migliorare le prestazioni, la maggior parte dei file system accorpa i blocchi in gruppi, detti cluster
 - ▶ I/O su disco fatto per blocchi
 - ▶ I/O via file system fatto per cluster (accesso sequenziale)
 - ▶ File e metadati vicini su HDD per diminuire i movimenti della testina





Gestione dell'unità a disco – 3

- ❖ Un disco di avviamento o disco di sistema ha una partizione di boot
- ❖ Il primo blocco logico del dispositivo è il blocco di avvio o boot block
- ❖ La partizione di boot contiene il SO; altre partizioni possono contenere altri SO, altri file system o essere partizioni raw
 - Viene montata all'avvio del sistema
 - Altre partizioni possono essere montate automaticamente o manualmente (al boot o successivamente)
- ❖ Al momento del montaggio di ogni partizione, si verifica la coerenza del file system (controllando la correttezza dei metadati)
 - Si aggiorna la tabella di montaggio





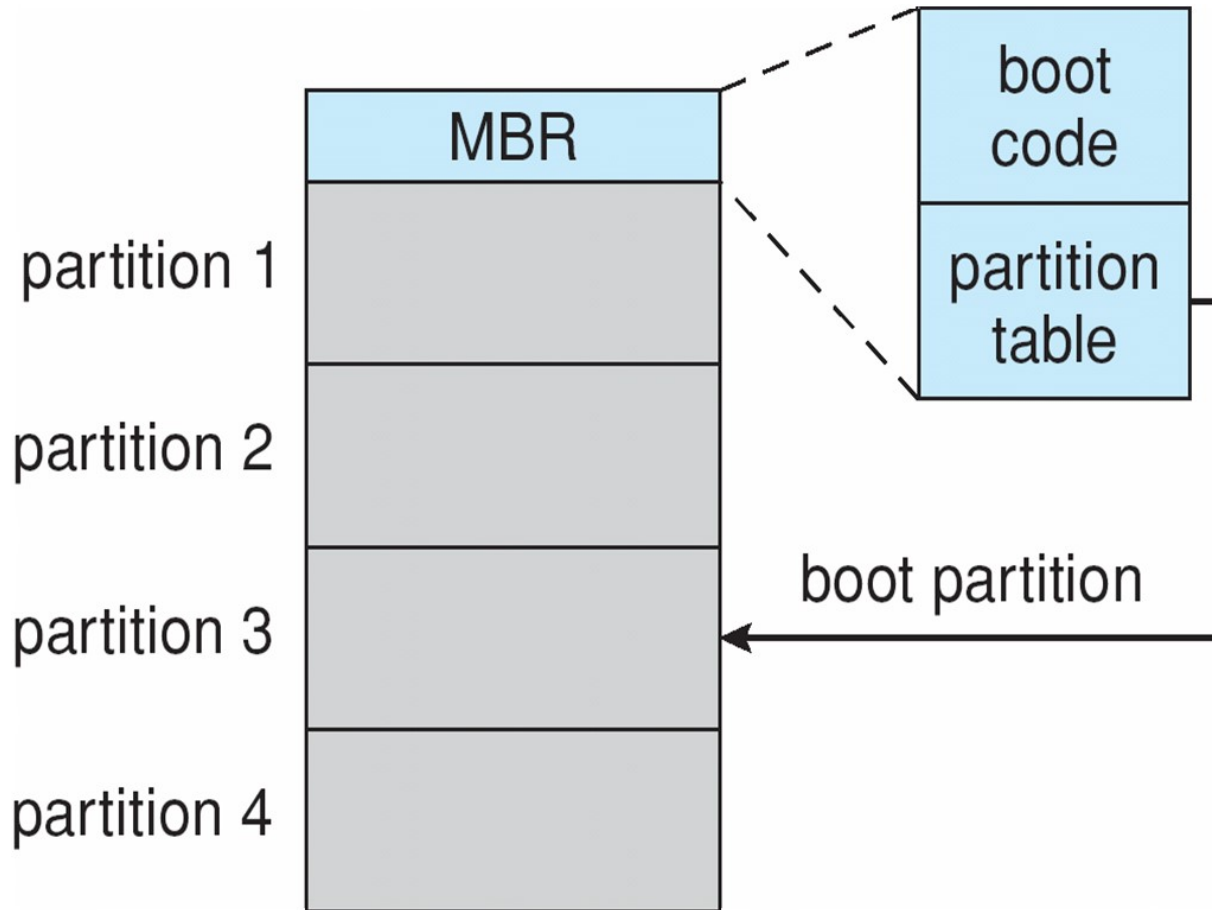
Gestione dell'unità a disco – 4

- ❖ Nel **boot block** sono contenute le informazioni necessarie all'inizializzazione del sistema
- ❖ In Windows si chiama **MBR** (Master Boot Record)
 - Esecuzione del codice del **bootstrap loader** contenuto nel firmware
 - Lettura/esecuzione del codice contenuto nell'MBR, che contiene anche una tabella delle partizioni, con un flag che identifica la partizione di boot
 - Caricamento, dalla partizione di boot, del kernel e dei sottosistemi del SO





Avviamento dal disco di Windows





Gestione dei blocchi difettosi

- ❖ I dischi magnetici sono strutturalmente pronti a malfunzionamenti, perché costituiti da parti mobili con basse tolleranze
- ❖ Si impiega l'**accantonamento dei settori** come modalità di gestione dei blocchi difettosi
 - Durante la formattazione fisica si mantiene un gruppo di settori di riserva non visibili al SO
 - Il controllore "è istruito" per sostituire, dal punto di vista logico, un settore difettoso con uno dei settori di riserva inutilizzati
- ❖ Anche i dispositivi NVM possono contenere pagine difettose, che vengono logicamente sostituite o con pagine accantonate o appartenenti alla riserva costituita dall'over-provisioning





Gestione dell'area di swap

- ❖ La memoria virtuale impiega lo spazio su disco come un'estensione della memoria centrale
 - Pratica attualmente meno comune, grazie all'incremento nella capacità delle memorie
- ❖ L'obiettivo principale nella progettazione e realizzazione dell'area di swap è di fornire la migliore produttività per il sistema di memoria virtuale
- ❖ Lo spazio di swap può essere ricavato all'interno del normale file system o, più comunemente, si può trovare in una partizione separata del disco
 - **Partizione raw:** adotta algoritmi ottimizzati rispetto alla velocità di accesso piuttosto che all'occupazione di spazio (frammentazione)





Gestione dell'area di swap in Linux – 1

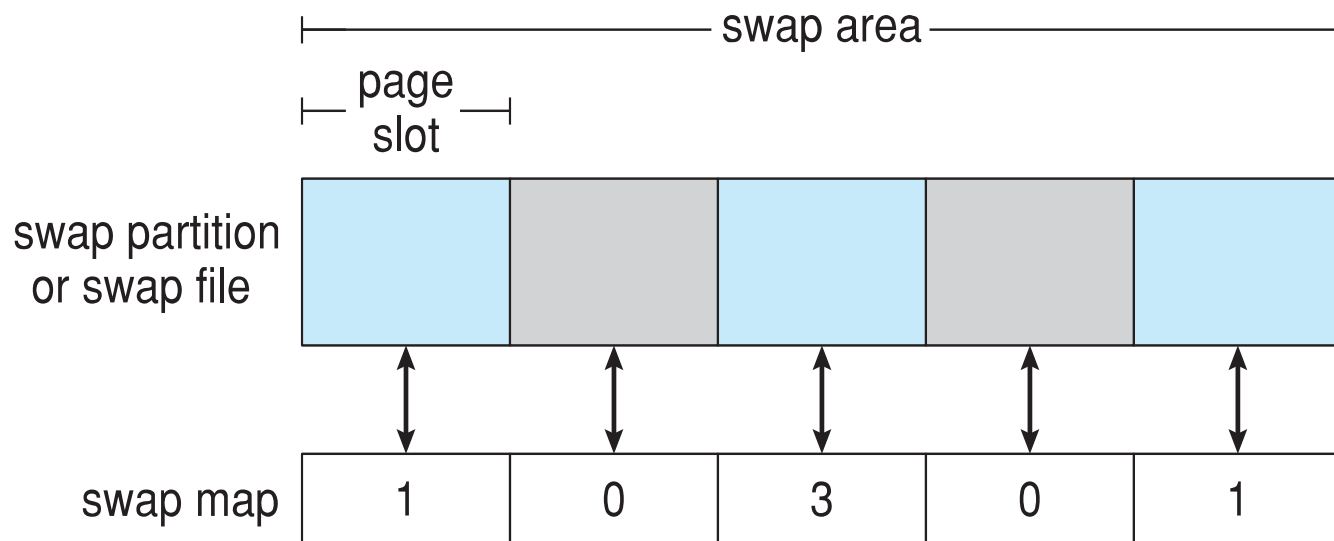
- ❖ L'area di swap, in Linux, è utilizzata solo per la memoria anonima, ovvero per dati che non corrispondono a file (stack, heap, dati non inizializzati)
- ❖ Linux permette l'istituzione di una o più aree di avvicendamento, sia in file che in una partizione raw, possibilmente situate su unità diverse per distribuire su più dispositivi il carico della paginazione
- ❖ Un'area di avvicendamento è formata da una serie di moduli di 4KB, detti **slot delle pagine**, la cui funzione è quella di conservare le pagine avvicendate





Gestione dell'area di swap in Linux – 2

- ❖ Ogni area dispone di una **mappa di avvicendamento**, un array di contatori interi, ciascuno dei quali corrisponde ad uno slot dell'area
 - Se un contatore vale 0, la pagina che gli corrisponde è disponibile; valori maggiori di 0 indicano che lo slot è occupato da una delle pagine avvicendate
 - ▶ Il valore del contatore indica il numero di collegamenti alla pagina; se, per esempio, vale 3, la pagina fa parte dello spazio degli indirizzi virtuali di tre processi distinti





Connessione dei dispositivi di memoria

- ❖ I calcolatori accedono alla memoria secondaria in tre modi:
 - Tramite un dispositivo collegato alla macchina
 - Tramite un dispositivo connesso alla rete
 - In cloud





Memoria secondaria connessa alla macchina

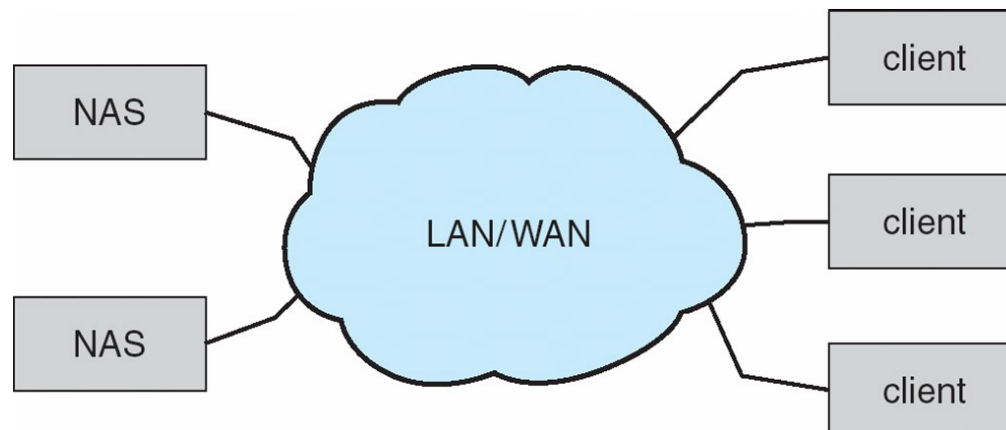
- ❖ Alla memoria secondaria connessa alla macchina si accede dalle porte locali di I/O che sono collegate al bus
 - Nei PC, con interfaccia SATA, due unità (al più) per ciascun bus di I/O
- ❖ Per accedere ad un maggiore spazio di archiviazione utilizzo di porte e cavi USB, FireWire o Thunderbolt
- ❖ **FC** (Fiber Channel) è un'architettura seriale ad alta velocità
 - Può gestire uno spazio d'indirizzi a 24 bit, che è alla base delle **storage area network** (SAN), nelle quali molti host sono connessi con altrettante unità di memorizzazione (*enterprise storage*)





Memoria secondaria connessa alla rete

- ❖ Un dispositivo di memoria secondaria connessa alla rete (Network-Attached Storage, **NAS**) è un sistema di memoria specializzato al quale si accede in modo remoto attraverso la rete di trasmissione dati
- ❖ I client accedono alla memoria connessa alla rete tramite un'interfaccia RPC, supportata da protocolli quali NFS (UNIX) e CIFS (Windows)
- ❖ Le chiamate di procedura remota sono implementate per mezzo di TCP o UDP, sopra una rete IP, di solito la stessa rete che supporta tutto il traffico dei dati fra client e server





Memoria secondaria in cloud

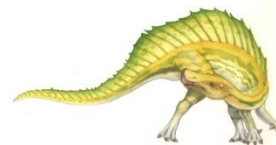
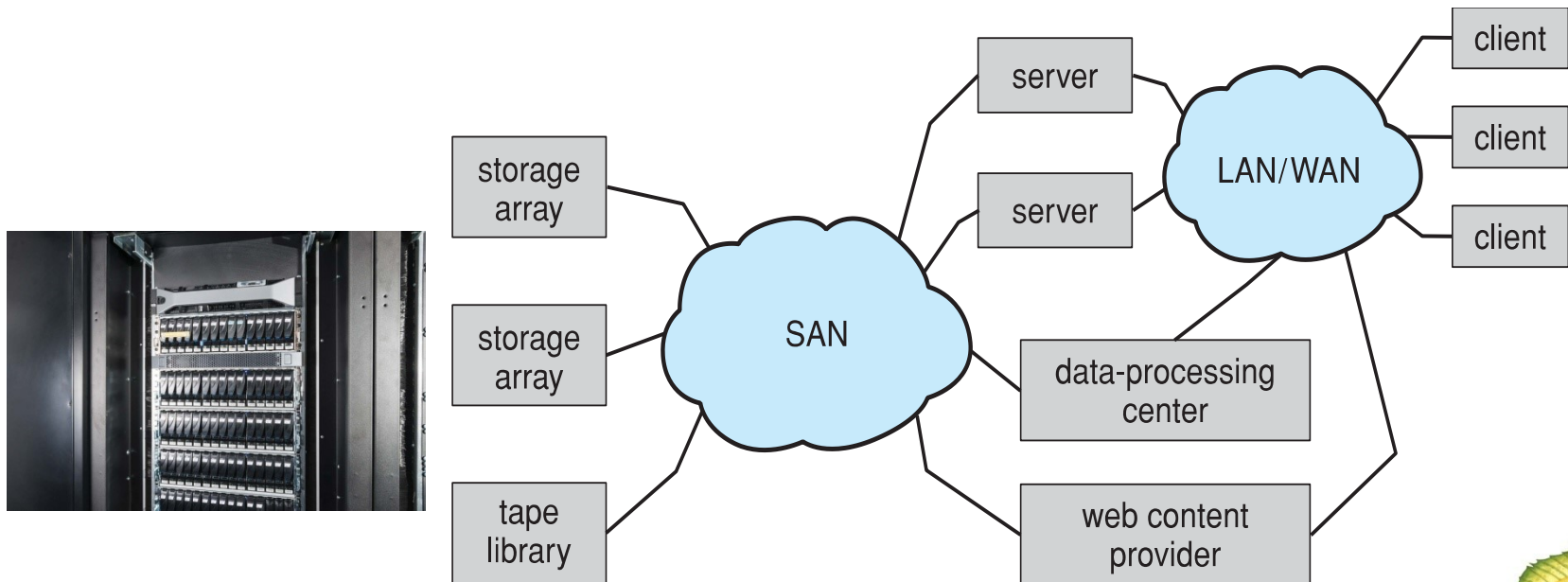
- ❖ Similmente al NAS, fornisce l'accesso allo storage tramite rete
 - A differenza del NAS, l'accesso al data center remoto avviene tramite Internet o WAN
- ❖ NAS si presenta come un altro file system, mentre lo storage cloud è basato su API, con programmi che utilizzano le API per fornire l'accesso
 - Esempi di cloud storage includono **Dropbox, Amazon S3, Microsoft OneDrive, Apple iCloud, Google drive**
 - Si impiegano le API a causa delle lunghe latenze e per i numerosi scenari di errore che sono comuni sulle WAN





Storage Area Network

- ❖ Reti private (che impiegano protocolli specifici per la memorizzazione) tra server e unità di memoria secondaria
- ❖ Flessibilità: si possono connettere alla stessa SAN molti calcolatori e molti storage array





Strutture RAID – 1

- ❖ RAID, Redundant Array of Independent Disks — l'affidabilità del sistema di memorizzazione viene garantita tramite la ridondanza
- ❖ Aumento del **tempo medio di guasto**
- ❖ Spesso affiancati dalla presenza di NVRAM per garantire la consistenza dei dati scritti “contemporaneamente” su dischi multipli e per migliorare le performance
- ❖ Inoltre... le tecniche per aumentare la velocità di accesso al disco implicano l'uso di più dischi cooperanti





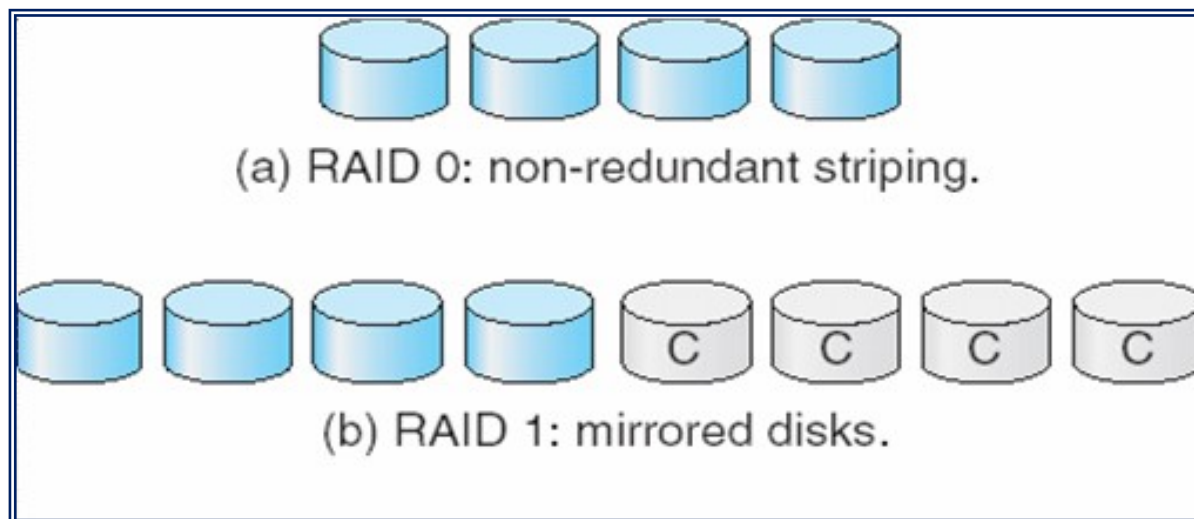
Strutture RAID – 2

- ❖ Il **sezionamento del disco** o **data striping** (RAID 0) tratta un gruppo di dischi come un'unica unità di memorizzazione:
 - Ogni "blocco" di dati è suddiviso in "sottoblocchi" memorizzati su dischi distinti (es.: i bit di ciascun byte possono essere letti "in parallelo" su 8 dischi)
 - Il tempo di trasferimento per rotazioni sincronizzate diminuisce proporzionalmente al numero dei dischi nella batteria
- ❖ Gli schemi RAID migliorano prestazioni ed affidabilità del sistema memorizzando dati ridondanti:
 - Il **mirroring** o **shadowing** (RAID 1) conserva duplicati di ciascun disco

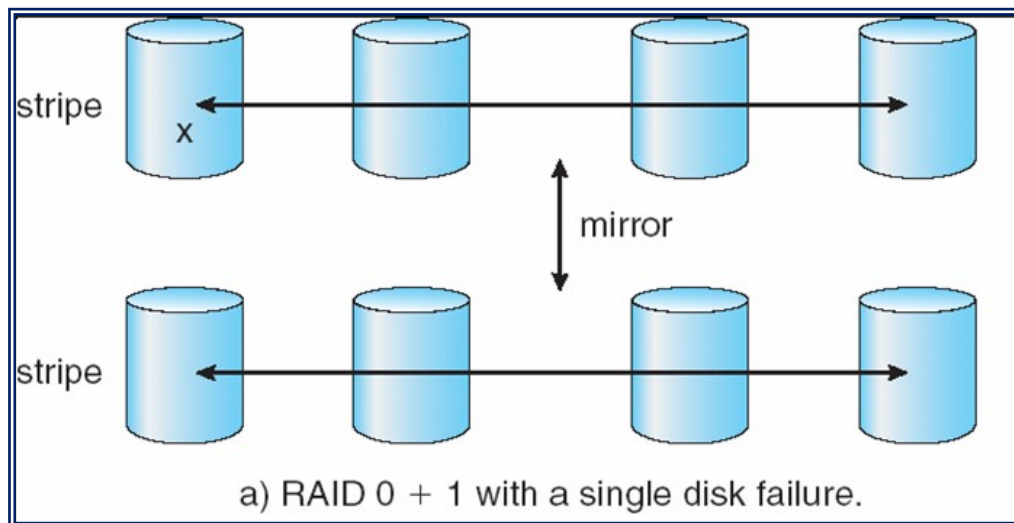




Alcuni livelli RAID



Il livello 0+1 consiste in una combinazione dei livelli RAID 0 e 1: il livello 0 garantisce le prestazioni e il livello 1 l'affidabilità



Fine del Capitolo 11

