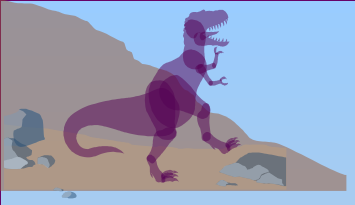
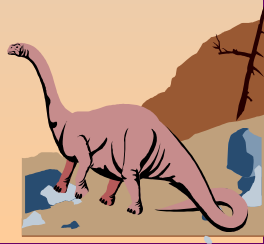


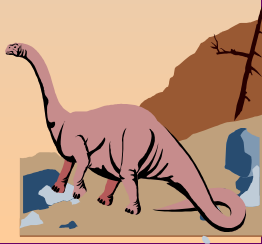
Capitolo 12: Memoria secondaria e terziaria

- Struttura dei dispositivi di memorizzazione
- Struttura dei dischi
- Connessione dei dischi
- Scheduling del disco
- Gestione dell'unità a disco
- Gestione dell'area di avvicendamento
- Strutture RAID
- Realizzazione della memoria stabile
- Strutture per la memorizzazione terziaria

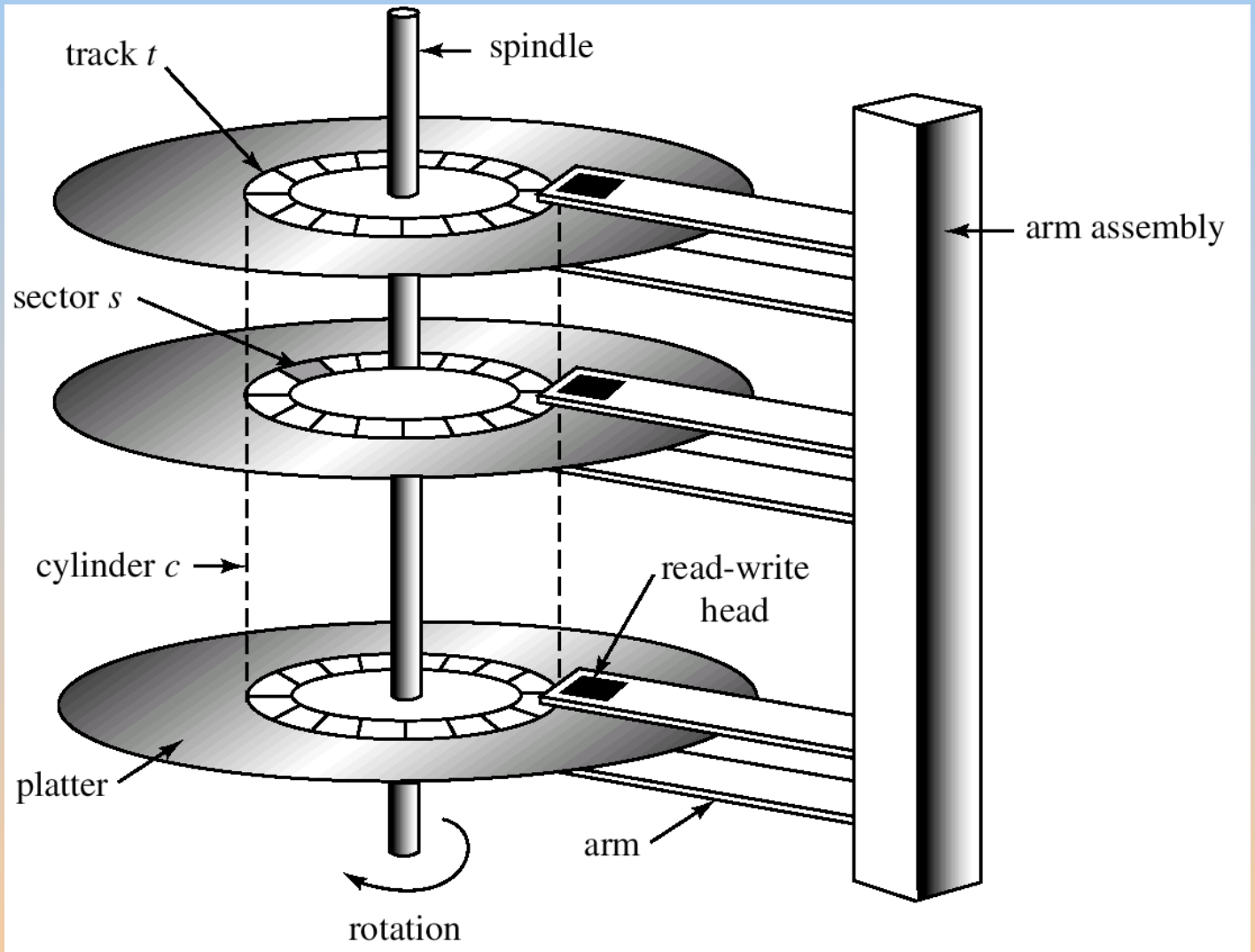


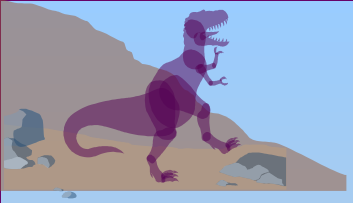
Struttura dei dischi

- I moderni dischi dal punto di vista dell'indirizzamento si considerano come un grande vettore monodimensionale di **blocchi logici**,
  - dove un blocco logico è la minima unità di trasferimento.
- La dimensione di un blocco logico è di solito 512 byte, ma alcuni dichi si possono **formattare a basso livello** allo scopo di ottenere una diversa dimensione dei blocchi logici (ad es. 1024 byte).
- Il vettore monodimensionale di blocchi logici corrisponde in modo sequenziale ai settori del disco:
  - Il settore 0 è il primo settore della prima traccia sul 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.



Schema funzionale di un disco



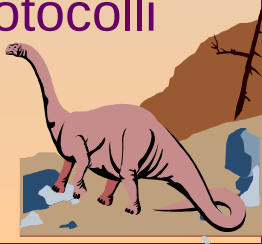


Scheduling del disco

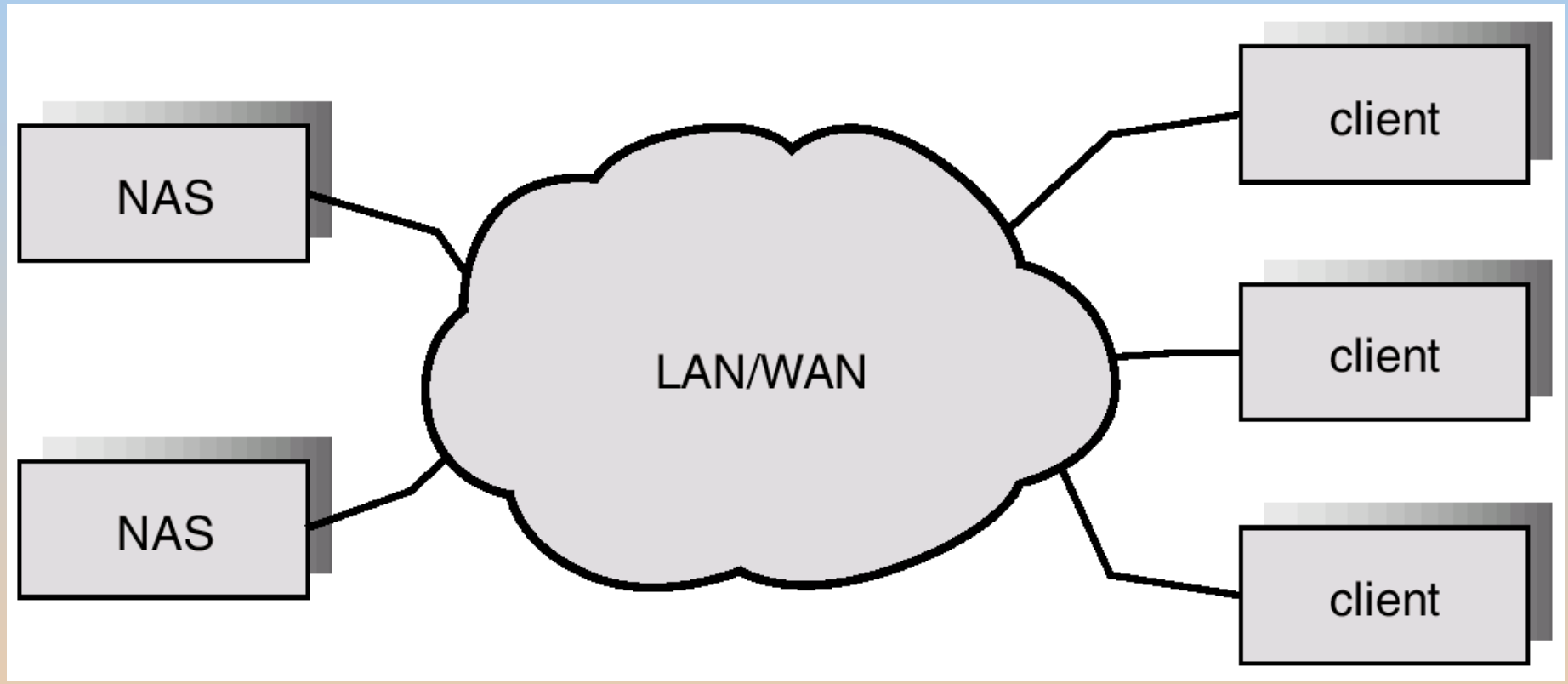
- La velocità di un disco nel trasferire i dati è un fattore di notevole importanza per tutto il sistema di elaborazione.
- Il sistema operativo è responsabile per l'utilizzo efficiente dell'hardware.
- Per quanto riguarda il disco questo significa l'avere
  - un **tempo di accesso** veloce ed
  - una grande **ampiezza di banda**
    - cioè il numero totale di byte trasferiti diviso il tempo totale intercorso tra la prima richiesta ed il completamento dell'ultimo trasferimento.
- Il sistema operativo può migliorare il tempo medio di servizio del disco pianificando le richieste di accesso al disco stesso attraverso un processo di scheduling.
- Il tempo di accesso ha due componenti principali
  - **tempo di ricerca**: il tempo necessario affinché il braccio dell'unità a disco sposti le testine fino al cilindro contenente il settore desiderato.
  - **latenza di rotazione**: tempo aggiuntivo necessario perché il disco ruoti finché il settore desiderato si trovi sotto la testina.
- Tramite lo scheduling del disco si possono migliorare entrambe.

Connessione dei dischi

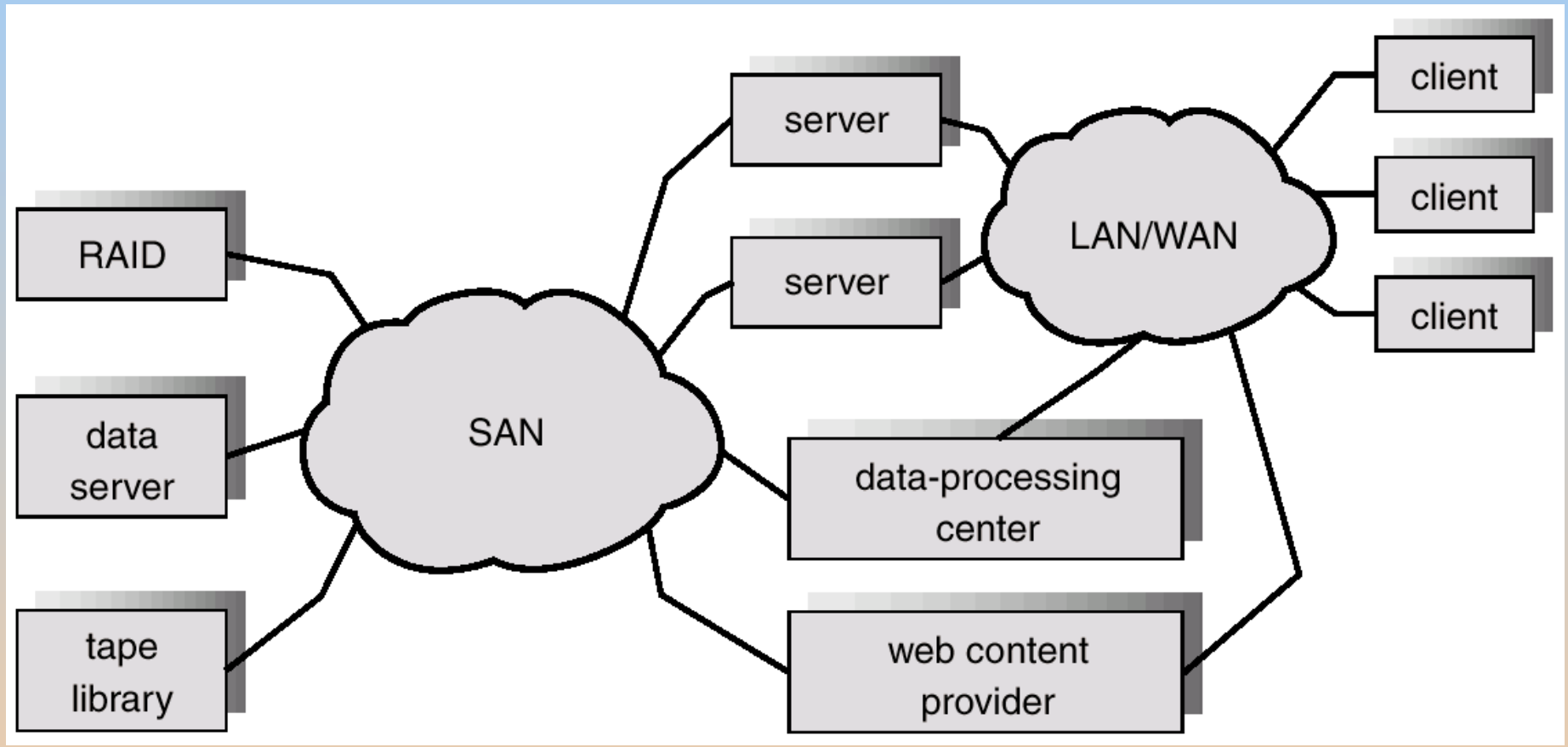
- L'accesso alla memoria secondaria può avvenire tramite:
  - le porte di I/O (**memoria secondaria connessa alla macchina**),
  - attraverso un file system distribuito (**memoria secondaria connessa alla rete**).
- Un dispositivo di memoria secondaria connessa alla rete (*Network Attached Storage* - **NAS**) è un sistema di memoria al quale si accede in modo remoto per mezzo di una rete di trasmissione dati.
- I client accedono alla memoria connessa alla rete tramite un interfaccia RPC, ad es. NSF nei sistemi Unix.
- Le chiamate di procedura remota (RPC) sono in genere realizzate per mezzo dei protocolli TCP o UDP su una rete IP.
- Uno svantaggio di una NAS è che le operazioni di I/O impiegano banda della rete e quindi aumentano la latenza di comunicazione.
- Una rete di memoria secondaria (*Storage Area Network* - **SAN**) è una rete privata che impiega protocolli specifici diversi dai protocolli di rete



Memoria secondaria connessa alla rete

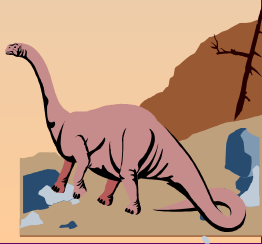


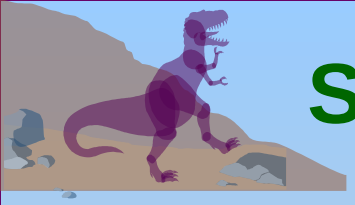
Rete di memoria secondaria



Scheduling del disco

- Ogni volta che si devono compiere operazioni di I/O con un'unità a disco un processo effettua una system call.
- La richiesta contiene diverse informazioni:
  - se l'operazione sia di immissione o di emissione di dati,
  - l'indirizzo nel disco rispetto al quale eseguire il trasferimento,
  - l'indirizzo di memoria rispetto al quale eseguire il trasferimento,
  - il numero di byte da trasferire.
- Se l'unità a disco è libera la richiesta si può immediatamente soddisfare, altrimenti le nuove richieste si aggiungono alla coda di richieste inevase rispetto a quell'unità.
- La coda relativa ad un'unità a disco può essere spesso lunga,
  - quindi S.O. dovrà scegliere quale fra le richieste inevase converrà servire prima.

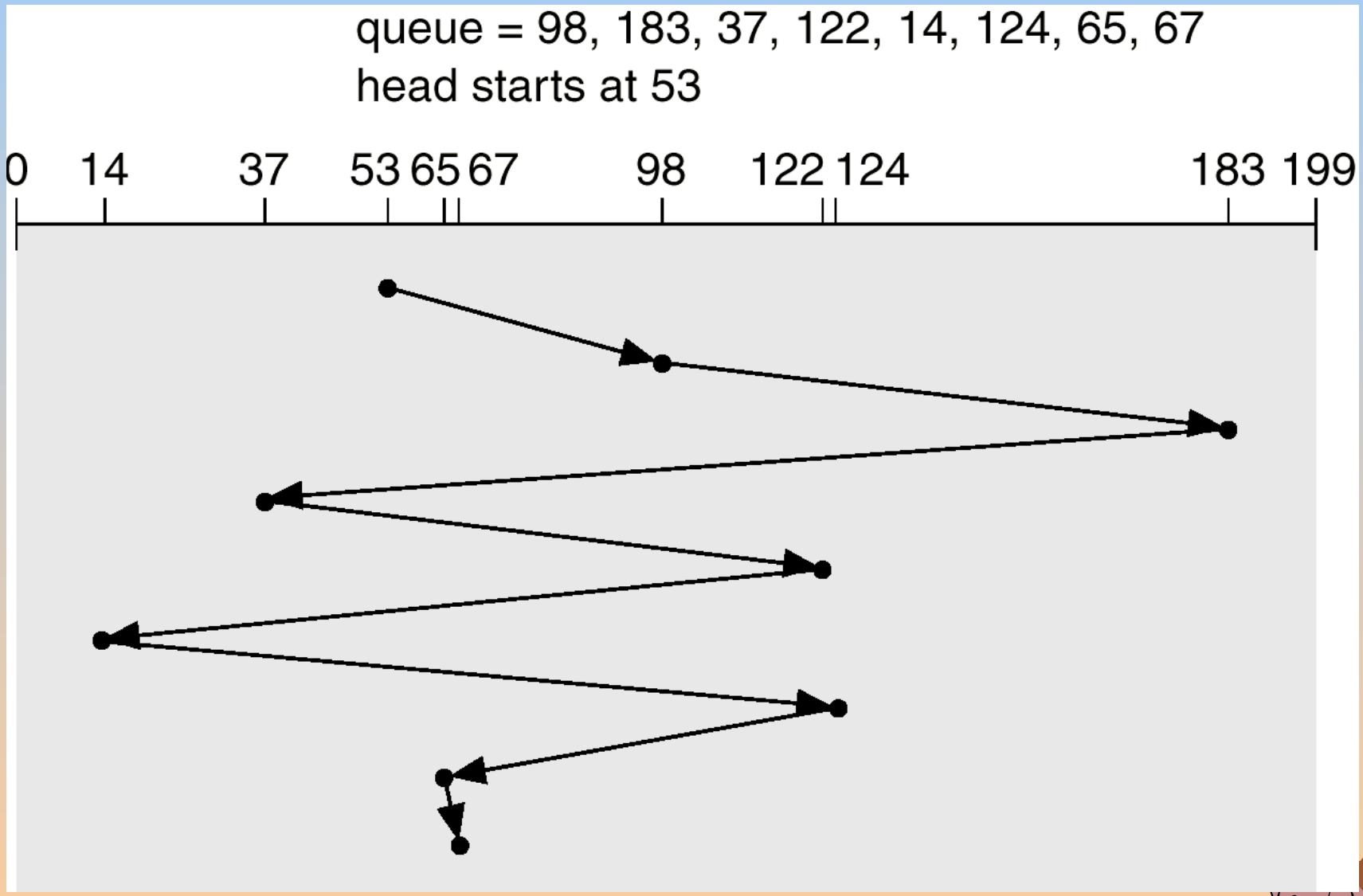




Scheduling in ordine di arrivo (FCFS)

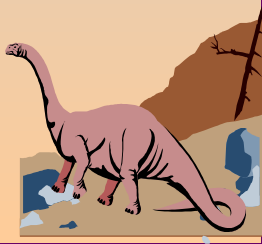
- La forma più semplice di scheduling è l'algoritmo di servizio secondo l'ordine di arrivo (FCFS).
- Nell'algoritmo di scheduling FCFS, la testina esegue le operazioni con lo stesso ordine di arrivo delle richieste.
- FCFS è un algoritmo semplice ma non è ottimale in termini di velocità del servizio.
- E' facile osservare infatti che le stesse richieste date in ordine diverso possono determinare tempi di servizio diversi.
- Tuttavia questo algoritmo viene utilizzato da quasi tutti i sistemi che hanno un carico basso:
  - in quanto i pochi benefici ottenuti dall'utilizzo di altri algoritmi, su un sistema a basso carico, non riescono a compensare i costi sostenuti per realizzarli.
- Si consideri ad esempio una coda di richieste per l'unità a disco che dia una lista di cilindri sui quali individuare i blocchi richiesti nel seguente ordine: **98, 183, 37, 122, 14, 124, 65, 67**.
- La testina è inizialmente posta al cilindro 53.

Scheduling FCFS

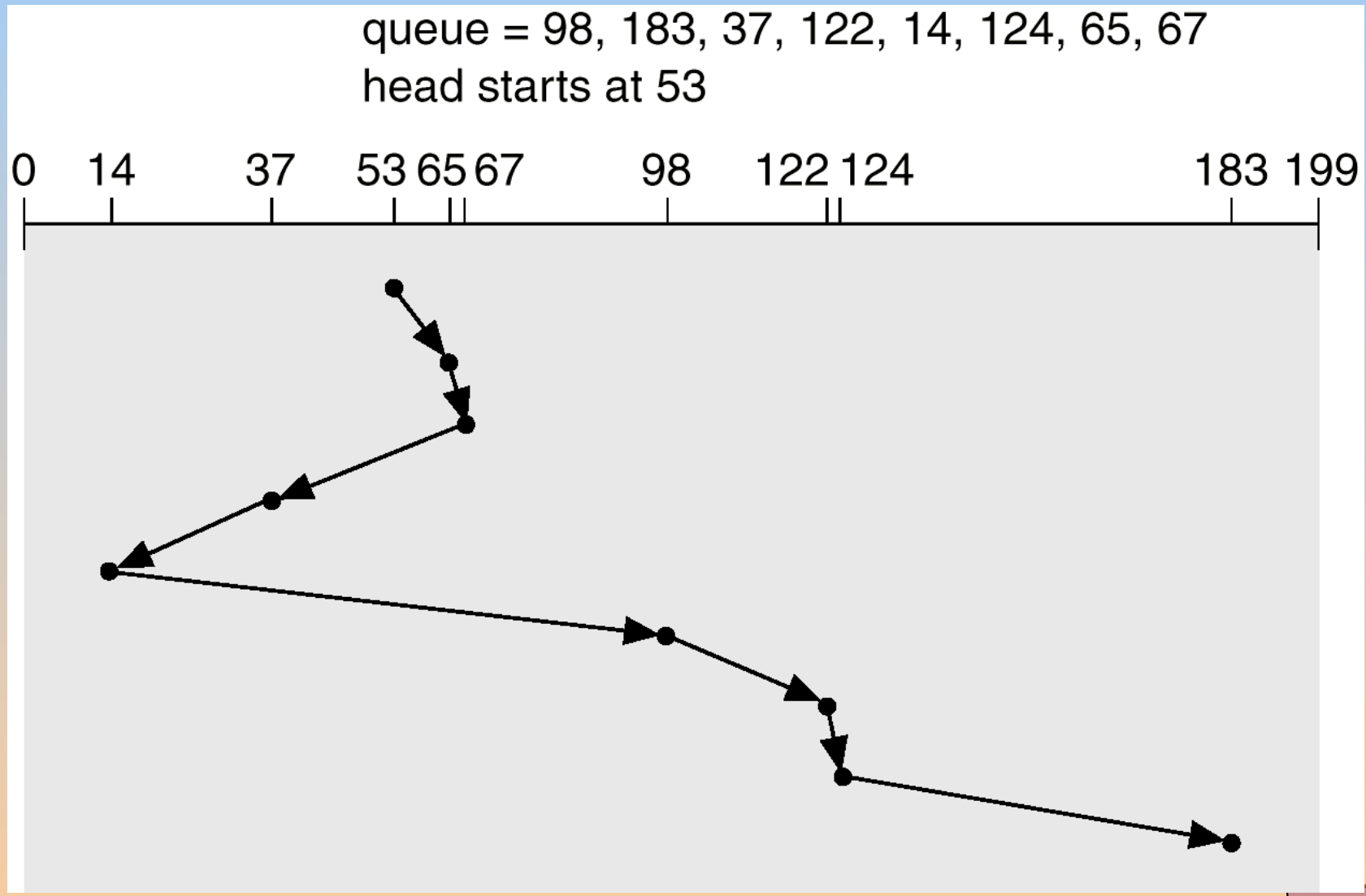


Scheduling per brevità (SSTF)

- L'algoritmo di scheduling per brevità (Shortest Seek Time First), a partire dalla posizione corrente seleziona la richiesta con tempo di ricerca minimo.
- In pratica la testina si sposta sempre sulla traccia più vicina, prima di allontanarsi per servire un'altra richiesta.
- Lo scheduling SSTF è essenzialmente una forma di scheduling SJF (Shortest-job-first), ed al pari di questo può portare a situazioni di starvation.
- Infatti poiché le richieste possono arrivare in qualunque momento, la richiesta di una traccia lontana dalla posizione iniziale potrebbe rimanere per un tempo indefinito in attesa.
- Pur avendo prestazioni migliori del FCFS, l'algoritmo SSTF non è ottimale in termini di velocità del servizio.

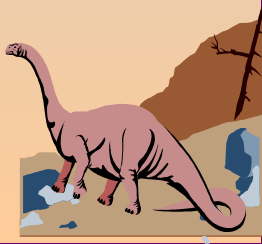


SSTF

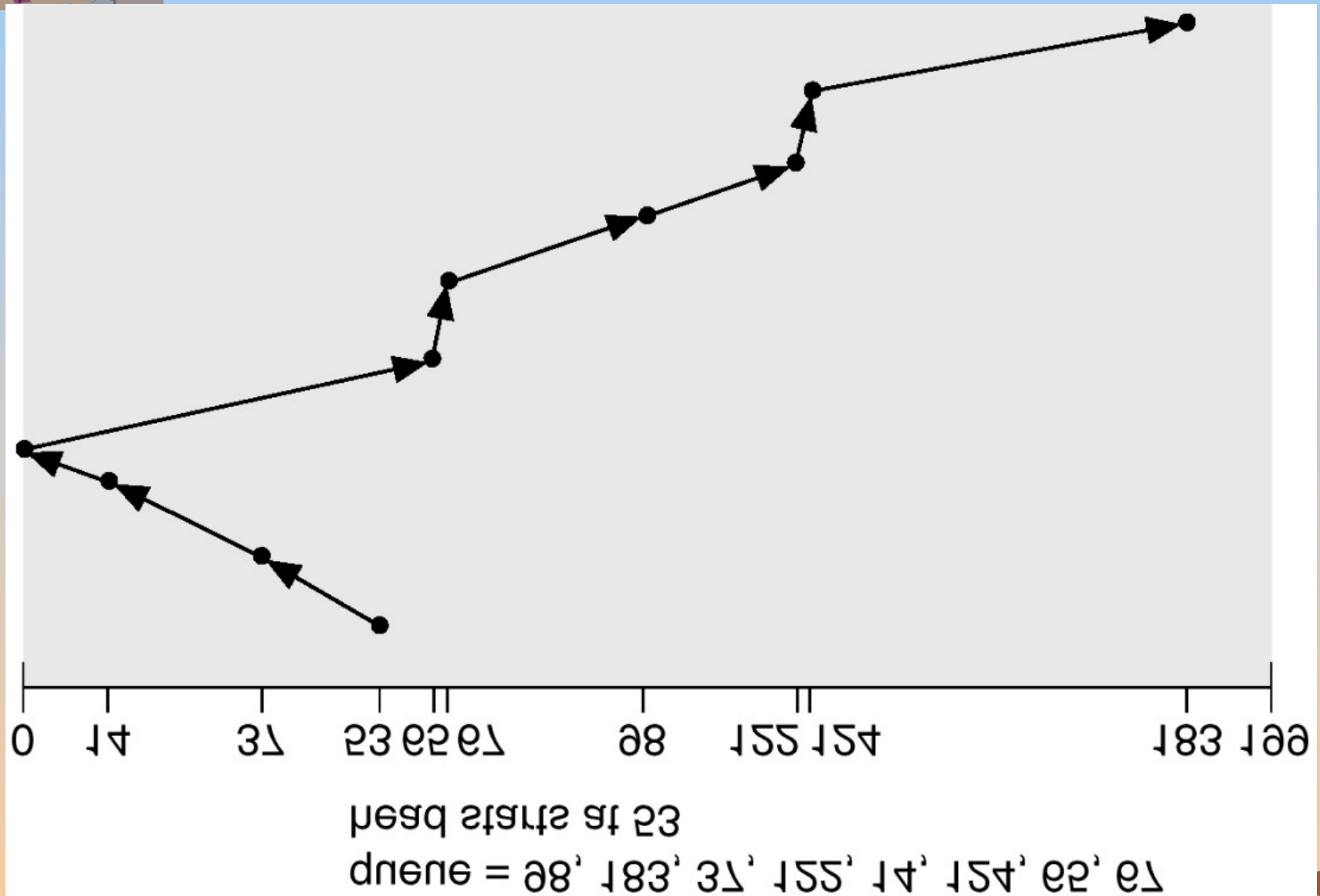

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Scheduling per scansione (SCAN)

- Secondo l'algoritmo di scheduling per scansione il braccio dell'unità a disco parte da un estremo del disco e si sposta mantenendo la stessa direzione,
  - servendo le richieste mentre attraversa i cilindri, fino a che non giunge all'altro estremo del disco.
- A questo punto il braccio inverte la marcia e la procedura continua.
- Le testine attraversano continuamente il disco nelle due direzioni.
- L'algoritmo SCAN viene talvolta chiamato algoritmo dell'ascensore,
  - perché il braccio del disco si comporta come un ascensore che serve prima tutte le richieste in salita e poi tutte quelle in discesa.
- Se nella coda delle richieste di I/O giunge una richiesta di lettura per una traccia molto vicina alla testina (rispetto al suo verso corrente), questa viene servita quasi immediatamente.
- Invece la richiesta di una traccia che si trova dietro la testina deve attendere fino a che la stessa non arrivi all'estremità del disco, inverta la direzione di spostamento e ritorni.



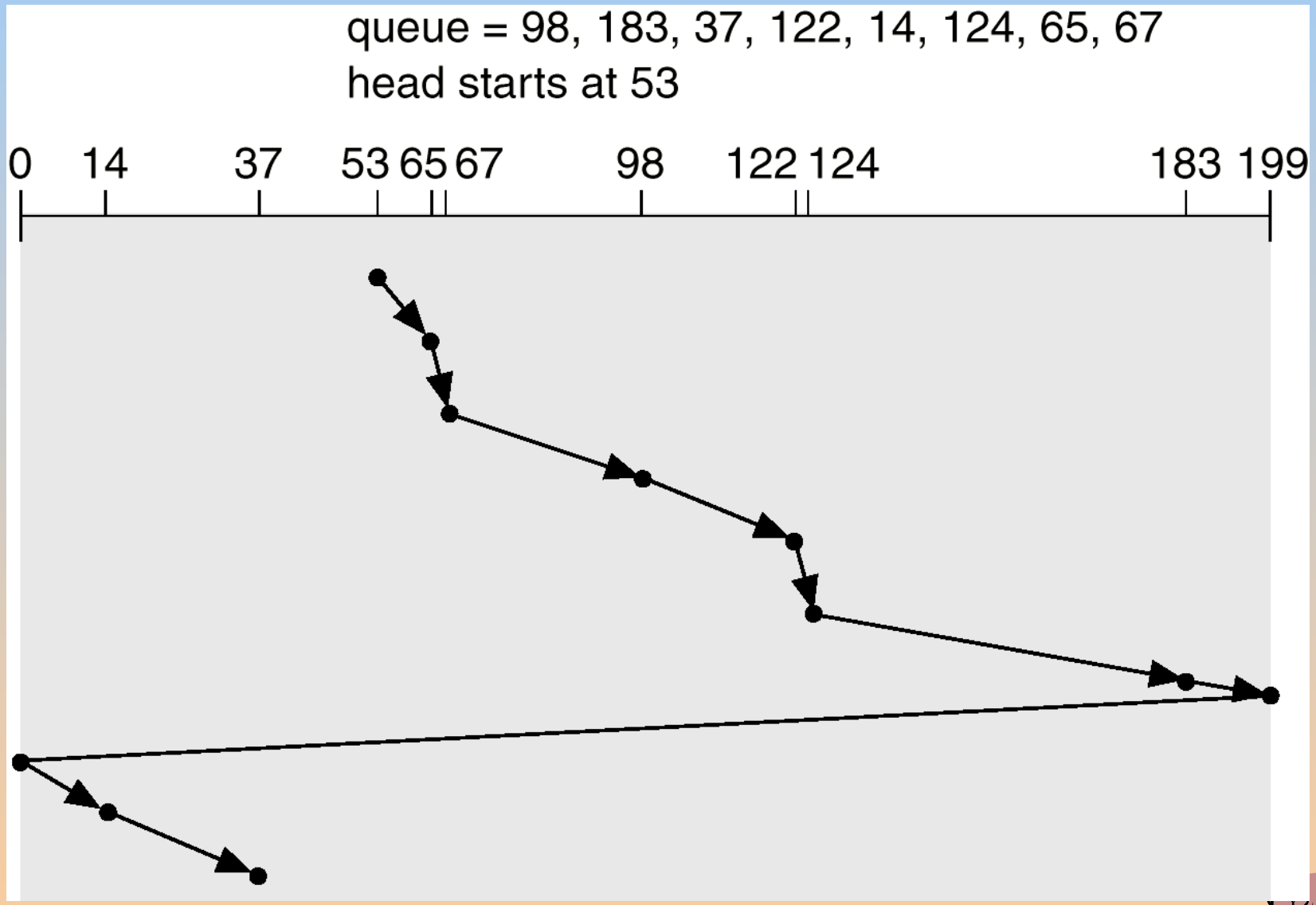
SCAN


0 14 37 53 65 67 98 122 124 183 199
head starts at 53
queue = 98, 183, 37, 122, 14, 124, 65, 67

Scheduling per scansione circolare (C-SCAN)

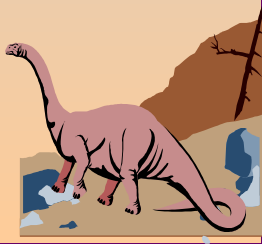
- L’algoritmo di scheduling per scansione circolare è una variante dell’algoritmo di scheduling SCAN progettato per fornire un tempo di attesa più uniforme.
- Ipotizzando una distribuzione uniforme per le richieste relative alle varie tracce, si consideri la densità di richieste che si presenta quando la testina raggiunge un'estremità e inverte la direzione di spostamento.
- A questo punto, dietro la testina sono presenti poche richieste, in quanto queste tracce sono state servite di recente.
- La maggiore densità di richieste è presente all'altra estremità del disco.
- Per ovviare a questa situazione quando la testina raggiunge l’estremità del disco, viene fatta tornare immediatamente all'inizio dello stesso senza servire le richieste che si trovano sul percorso di ritorno.
- Fondamentalmente, lo scheduling C-SCAN tratta il disco come una lista circolare, cioè come se il primo e l’ultimo cilindro fossero adiacenti.

C-SCAN

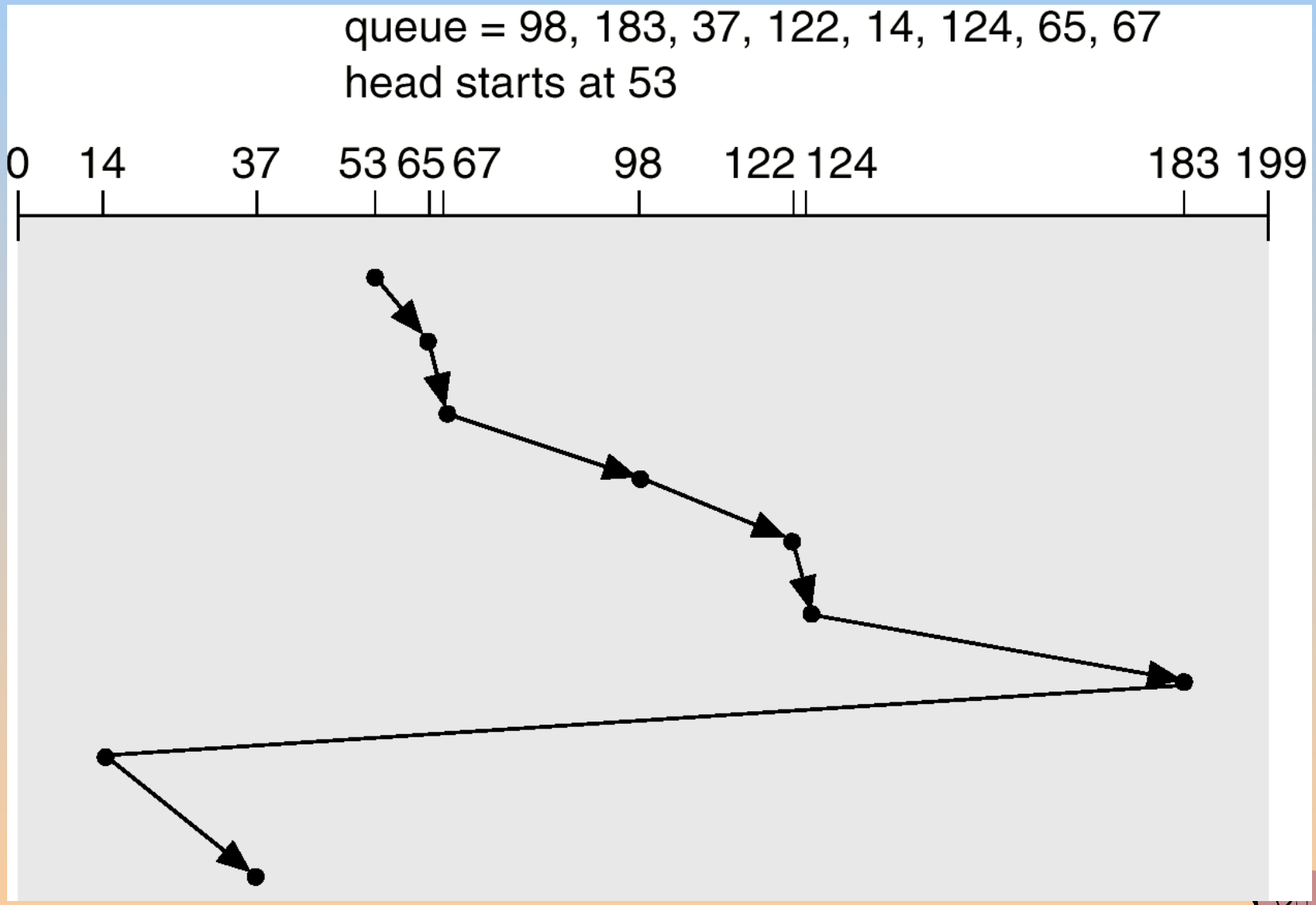

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Scheduling LOOK

- Gli algoritmi di scheduling LOOK e C-LOOK rappresentano un ulteriore passo in avanti rispetto agli algoritmi SCAN e C-SCAN.
- Infatti:
  - gli algoritmi del tipo SCAN effettuano sempre delle scansioni complete del disco,
  - gli algoritmi di tipo LOOK valutano ad ogni passo se vi sono altre richieste da servire in quella direzione, altrimenti viene immediatamente invertito il verso della testina .

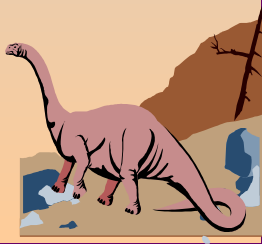


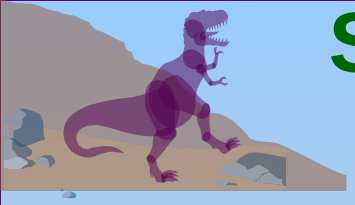
C-LOOK


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

Scelta di un algoritmo di scheduling

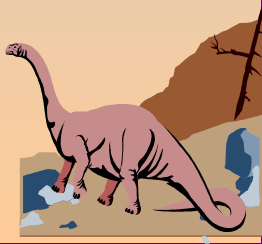
- Uno degli algoritmi più utilizzati è l'SSTF, poiché aumenta le prestazioni rispetto all'FCFS.
- SCAN e C-SCAN danno migliori prestazioni in sistemi che sfruttano molto le unità a disco, perché conducono con minor probabilità a situazioni di attesa indefinita.
- Definito l'insieme dei processi da schedulare, è possibile disegnare un algoritmo ad hoc ottimale,
  - ma il risparmio che ne deriva non e' sufficiente a compensare il calcolo necessario per ottenere l'ottimizzazione.
- E' chiaro che la prestazione dipende fortemente dal numero e dai tipi di richieste.
- Se c'è di solito solo una richiesta in sospeso, tutti gli algoritmi si equivalgono (anche FCFS).





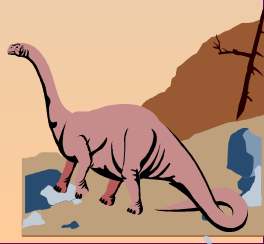
Scelta di un algoritmo di scheduling (II)

- Le richieste di servizio del disco sono influenzate anche dal metodo di allocazione dei file.
- Leggere un file memorizzato in modo contiguo genera parecchie richieste vicine, per cui il movimento della testina è limitato.
- Un file memorizzato tramite assegnazione concatenata o indicizzata usa meglio lo spazio, ma aumenta notevolmente gli spostamenti della testina durante l'accesso ai dati.
- L'algoritmo di schedulazione del disco dovrebbe essere scritto come un modulo separato del sistema operativo,
  - in maniera tale da permettere la propria riscrittura o sostituzione con un algoritmo diverso, ove necessario.



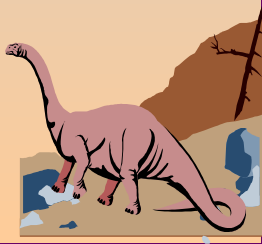
Scelta di un algoritmo di scheduling (III)

- E’ importante anche la locazione delle directory e dei blocchi indice.
- Per essere usati i file devono essere aperti, il che implica una ricerca nella struttura delle directory.
- Sono necessari quindi frequenti accessi a queste ultime.
- Se le directory vengono poste in tracce centrali del disco, lo spostamento della testina può essere ridotto.
- Ad esempio, se un elemento di directory si trova nella prima traccia e i dati di un file si trovano nell'ultima traccia, allora la testina deve spostarsi su tutta la larghezza del disco.



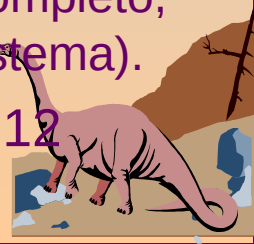
Gestione dell'unità a disco

- Il processo di organizzazione della superficie del disco in tracce e settori che possano essere letti dal controllore è chiamato **formattazione di basso livello** o **formattazione fisica**,
  - quasi tutti gli hard disk oggi vengono pre-formattati dalle case produttrici.
- La formattazione di basso livello riempie il disco con una specifica struttura di dati per ogni settore.
- In quasi tutti i sistemi, inclusi PC e Mac, i settori contengono in genere un'intestazione, un'area per i dati (in genere 512 bytes) ed una coda.
- L'intestazione e la coda contengono le informazioni usate dal controllore del disco,
  - ad es. il numero di settore ed un codice per la correzione degli errori.
- Per utilizzare un disco, S.O. deve:
  - **partizionare** il disco (in una o più partizioni, suddividendolo in gruppi di cilindri)
  - **formattarlo logicamente** (cioè creare il file system).

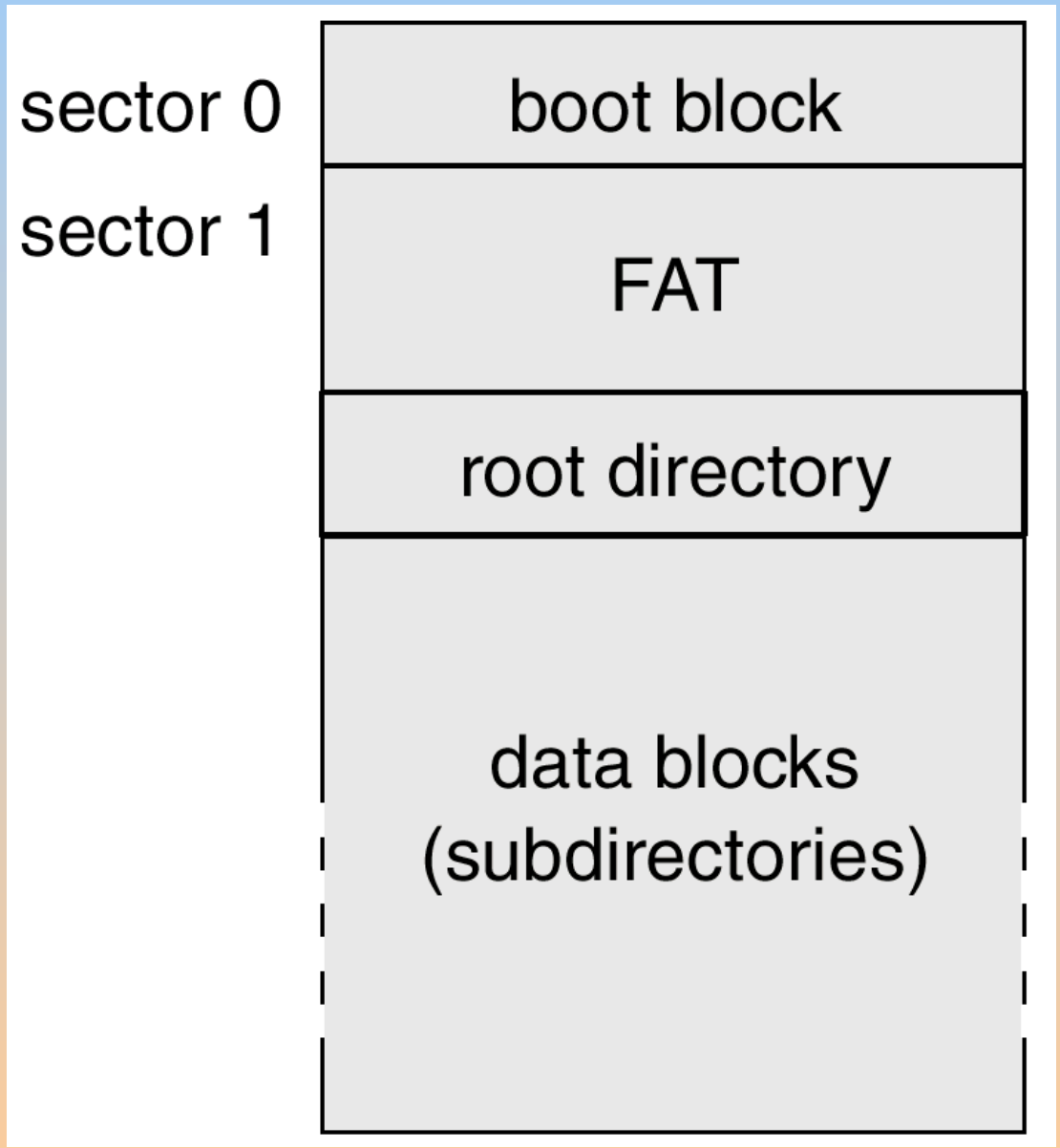


Blocco di avviamento

- Affinché un calcolatore possa entrare in funzione, è necessario che esegua un programma iniziale.
- Di solito il **programma di avviamento** iniziale è molto semplice:
  - inizializza il sistema in tutti i suoi aspetti, dai registri della CPU ai controllori dei dispositivi e al contenuto della memoria centrale, quindi avvia il S.O..
- Per far ciò il programma di avviamento trova il nucleo del sistema operativo sui dischi, lo carica nella memoria e salta ad un indirizzo iniziale per avviare l'esecuzione del sistema operativo.
- In genere il programma di avviamento è memorizzato in una memoria di sola lettura (ROM).
- Pero' cambiare programma di avviamento significherebbe dover cambiare la ROM.
- A causa di questo inconveniente molti sistemi memorizzano nella ROM un piccolo **caricatore di avviamento** (bootstrap loader),
  - il cui solo scopo è caricare da disco il programma di avviamento completo, registrato in una partizione del **disco di avviamento** (o disco di sistema).
- In MS-DOS il programma di avviamento è breve: un blocco di 512 byte.



Configurazione del disco nell'MS-DOS

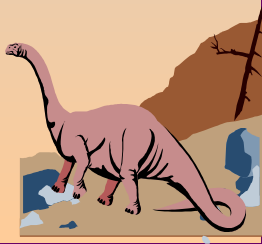


Blocchi difettosi

- Sebbene un ammontare straordinario di cura e sforzo da parte delle case produttrici vada nel costruire piatti per hard disk drives,
  - non è economicamente fattibile fabbricare platters al 100% liberi da errori.
- Perciò, tutti i moderni drives adottano nel controller una strategia per il management degli errori per assicurare spazio libero da **blocchi difettosi**.
- Alla fine del processo di manifattura, l'intera superficie del disco è analizzata per evidenziare eventuali difetti
  - e in questa fase il controllore del disco immagazzina una mappa delle loro ubicazioni.
- Quando il sistema operativo richiede che le informazioni debbano essere scritte in uno dei settori danneggiati,
  - il controllore del disco trasparentemente lo mappa ad uno dei settori di riserva inutilizzati (**accantonamento di settori**).
- Il controllore del disco aggiorna continuamente la mappa dei blocchi difettosi, per evitare che su nuovi settori danneggiati vengano memorizzate informazioni.

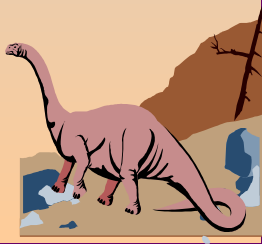
Gestione dell'area di avvicendamento

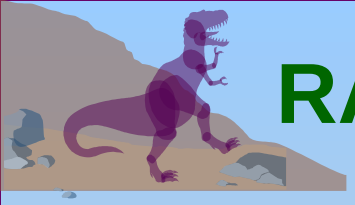
- La gestione dell'area di avvicendamento è un altro compito a basso livello del sistema operativo.
- La memoria virtuale usa lo spazio dei dischi come estensione della memoria centrale.
- I sistemi che adottano l'avvicendamento dei processi nella memoria possono usare l'area di avvicendamento (area di swap) per mantenere l'intera immagine del processo, inclusi i segmenti dei dati e del codice.
- I sistemi a paginazione possono semplicemente memorizzarvi pagine non contenute nella memoria centrale.
- Vi sono due possibili collocazioni per uno spazio di swap:
  - all'interno del file system
  - su una partizione indipendente del disco



Strutture RAID

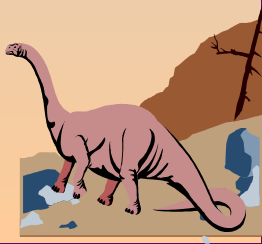
- La presenza di più dischi rende possibile l'aumento della frequenza con cui i dati si possono leggere o scrivere, e permette di migliorare l'affidabilità della memoria secondaria.
- Esistono tecniche per l'organizzazione dei dischi note con il nome comune di **batterie ridondanti di dischi** (Redundant Array of Independent Disks - RAID).
- Il RAID è un metodo di organizzazione dei dischi in un array che appare al calcolatore come una singola unità di memorizzazione.
- In passato strutture RAID composte da piccoli dischi economici erano viste come un'alernativa economicamente vantaggiosa rispetto a costosi dischi di grande capacità.
- Oggi si impiegano invece per la loro maggiore affidabilità e velocità di trasferimento dei dati.
- Quindi la I in RAID era originariamente letta come "Inexpensive".





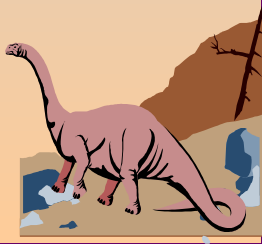
RAID: affidabilità tramite la ridondanza

- La possibilità che uno dei dischi in un insieme di *n* dischi si guasti è molto più alta della possibilità che uno specifico disco isolato presenti un guasto.
- Se si memorizzasse una sola copia dei dati, allora ogni guasto di un disco comporterebbe la perdita di una notevole quantità di dati.
- La soluzione al problema dell'affidabilità stà nell'introdurre una certa **ridondanza**,
  - cioè nel memorizzare informazioni che non sono normalmente necessarie ma che si possono usare in caso di guasto per ricostruire le informazioni perse.
- Ad esempio nella **copiatura speculare** (mirroring) ogni disco logico consiste di due dischi fisici e ogni scrittura si effettua su entrambi i dischi.



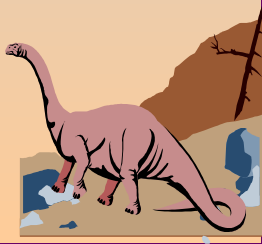
RAID: prestazioni tramite il parallelismo

- Con la copiatura speculare, se si può accedere in parallelo a più dischi, la frequenza con la quale si possono gestire le richieste di lettura raddoppia.
- Attraverso l'uso di più dischi è possibile anche migliorare la capacità di trasferimento distribuendo i dati in sezioni su più dischi.
- Ogni blocco di dati è frazionato in diversi sottoblocchi memorizzati in parallelo su dischi differenti.
- Nella sua forma più semplice questa distribuzione (**sezionamento dei dati**) consiste nel distribuire i bit di ciascun byte su più dischi (**sezionamento al livello dei bit**).
- Se distribuiamo i blocchi di un file su più dischi avremo il **sezionamento al livello dei blocchi**.
- Il fallimento di un singolo disco può però condurre a una grande perdita di dati.



Livelli RAID

- Sono stati proposti numerosi schemi per fornire ridondanza usando l'idea del sezionamento combinata con i bit di parità.
- Questi schemi realizzano diversi compromessi tra costi e prestazioni e sono stati classificati in più **livelli RAID**.
- **Raid 0:** batterie di dischi con ***sezionamento al livello dei blocchi***, ma senza ridondanza.
- **Raid 1: *copiatura speculare***,
  - ovvero tutti i dati sono memorizzati in due copie su due diversi dischi.
- **Raid 2: *organizzazione con codici per la correzione degli errori***.
  - I dati sono memorizzati sul disco in gruppi di bit.I bit del pattern dei dati sono memorizzati in parallelo su tutti i dischi dell'array.
  - Codici per la la correzione degli errori sono generati e scritti sui dischi di ECC per identificare e correggere errori.



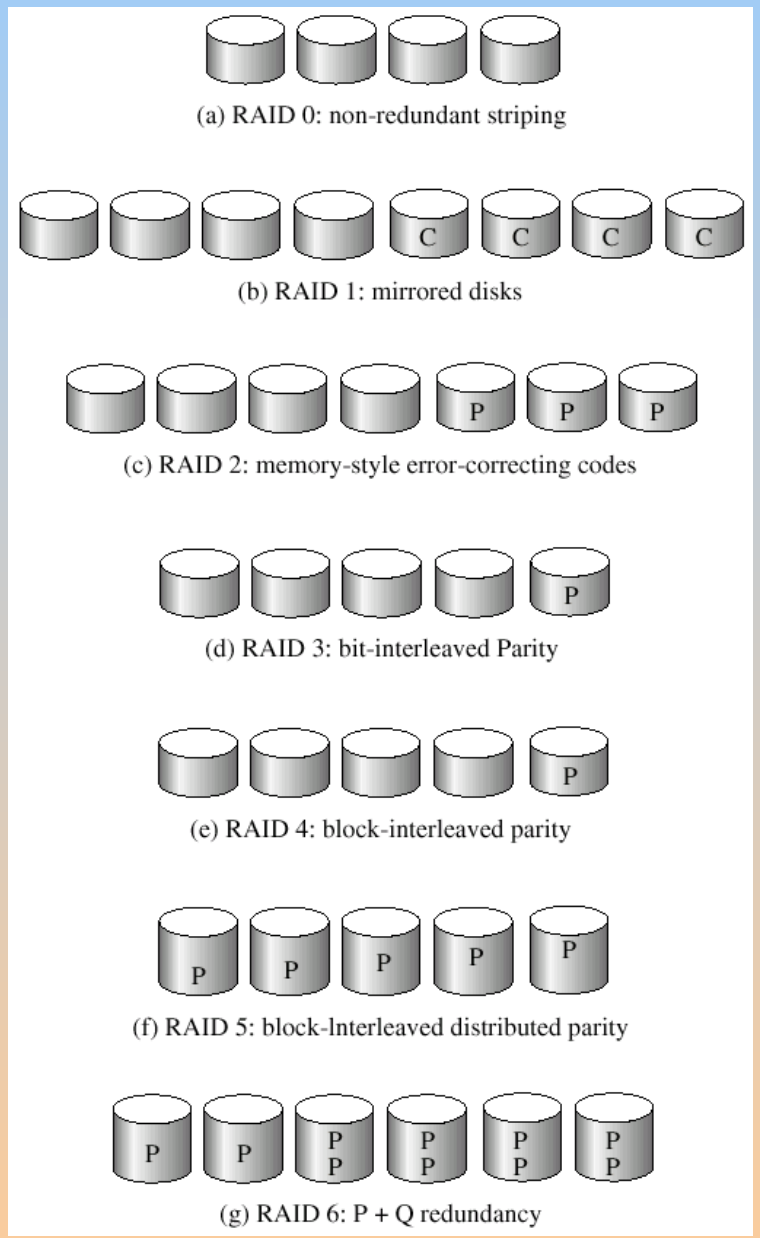
Livelli RAID (II)

- **Raid 3: *organizzazione con bit di parità intercalati*.**
  - Giacché i controllori dei dischi possono rilevare se un settore è stato letto correttamente, un unico bit di parità si può usare sia per individuare gli errori che per correggerli.
  - Se uno dei settori è danneggiato, per ogni bit del settore è possibile determinare il suo valore considerando tutti gli altri bit dell'ottetto memorizzati sugli altri dischi + il bit di parità.
- **Raid 4: *organizzazione con blocchi di parità intercalati*.**
  - Si impiega il sezionamento dei blocchi (RAID 0) e si tiene un blocco di parità in un disco separato.
  - Se uno dei dischi si guasta il blocco di parità viene utilizzato insieme agli altri blocchi corrispondenti per ripristinare il blocco perso.
- **Raid 5: *organizzazione con blocchi intercalati a parità distribuita*.**
  - Differisce dal livello 4 perché invece di memorizzare i dati in *n* dischi e la parità in un disco separato, i dati e la parità sono distribuiti tra gli *n+1* dischi.

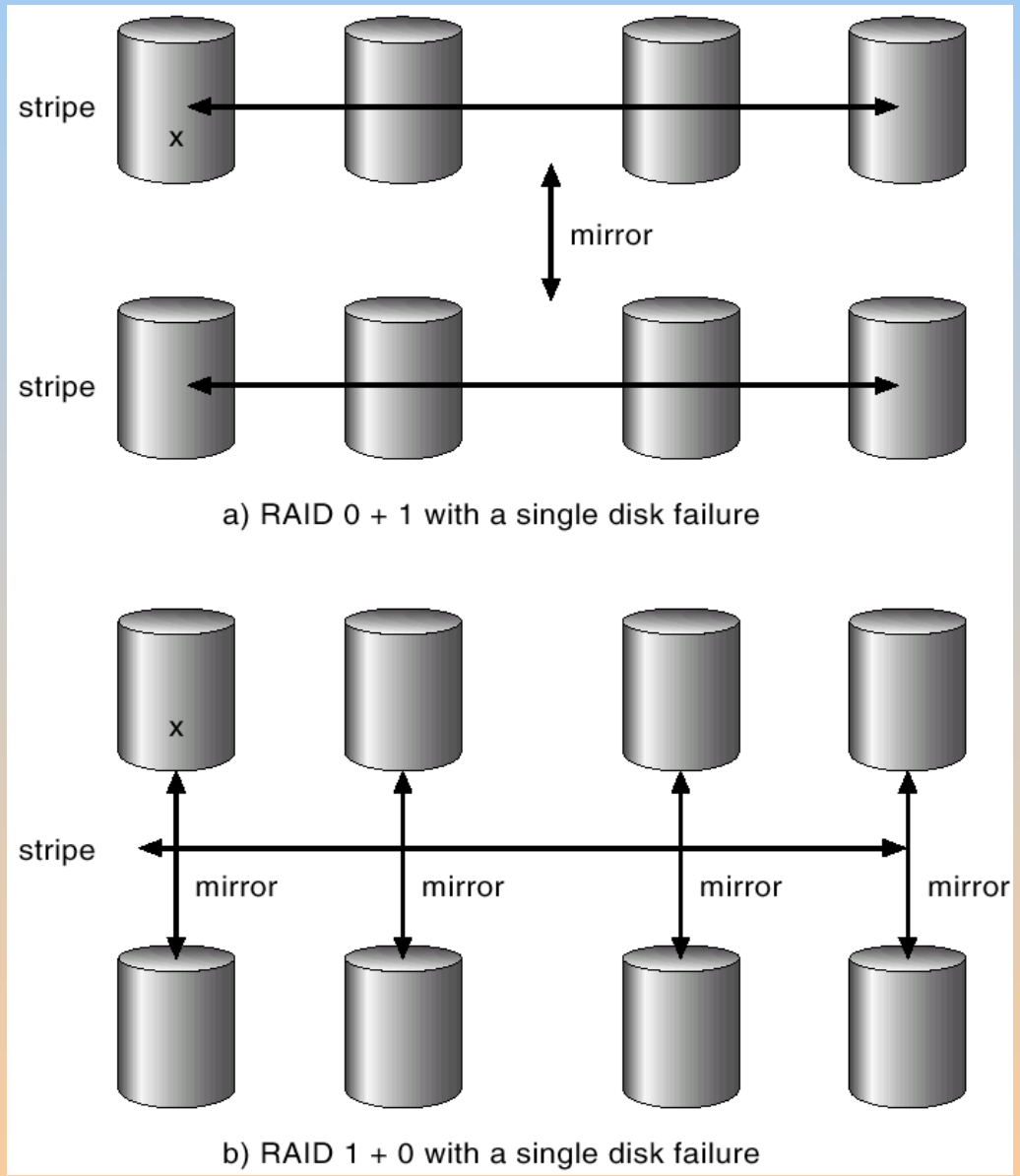
Livelli RAID (III)

- **Raid 5: *organizzazione con blocchi intercalati a parità distribuita*.**
  - Differisce dal livello 4 perché invece di memorizzare i dati in *n* dischi e la parità in un disco separato, i dati e la parità sono distribuiti tra gli *n+1* dischi.
- **Raid 6: *schema di ridondanza P+Q*.**
  - Molto simile al 5 ma memorizza ulteriori informazioni ridondanti per gestire guasti contemporanei di più dischi. Invece della parità si utilizzano i **codici di Reed-Solomon**.
- **Raid 0 + 1:**
  - è una combinazione dei livelli 0 e 1.
  - Si sezionano al livello dei blocchi i dati in un insieme di dischi e poi si duplica (indipendentemente ed eventualmente in modo diverso) ogni sezione con la tecnica della copiatura speculare.
- **Raid 1 + 0:**
  - si fa prima la copiatura speculare dei dischi a coppie e poi il sezionamento su queste coppie.
- Teoricamente in 0+1 se si guasta un disco l'intera sezione dati diventa inaccessibile, mentre in 1+0 si può utilizzare lo stesso disco rimanente per entrambe le sezioni.

Schematizzazione dei livelli RAID

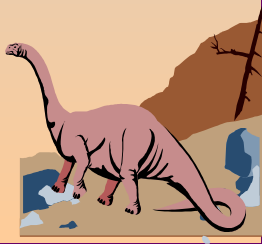


RAID (0 + 1) e (1 + 0)


stripe
x
mirror
stripe
a) RAID 0 + 1 with a single disk failure
x
stripe
mirror
mirror
mirror
mirror
b) RAID 1 + 0 with a single disk failure

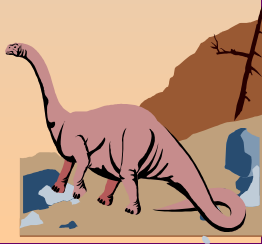
Scelta di un livello RAID

- Se un disco si guasta il tempo per ricostruire i dati in esso memorizzati può essere rilevante e può variare secondo il livello RAID impiegato.
- La ricostruzione più semplice si ha al livello 1, poiché i dati possono essere semplicemente copiati dal duplicato del disco.
- Per gli altri livelli è necessario accedere a tutti i dischi della batteria.
- Il RAID di livello 0 si usa nelle applicazioni ad alte prestazioni in cui le perdite di dati non sono critiche.
- Il RAID di livello 1 si usa nelle applicazioni che richiedono un'alta affidabilità ed un rapido ripristino.
- I RAID 0+1 e 1+0 si usano dove entrambe prestazioni ed affidabilità sono importanti, per esempio per piccole basi di dati.
- Per la memorizzazione di grandi quantità di dati, in genere si preferisce impiegare il RAID di livello 5, non essendo il livello 6 sempre disponibile.
- I concetti relativi ai sistemi RAID sono stati generalizzati ad altri dispositivi,
  - ad es. batterie di nastri o diffusione di dati in sistemi wireless.



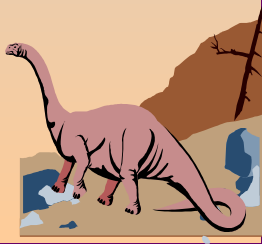
Strutture per la memorizzazione terziaria

- La caratteristica peculiare della memoria terziaria è il suo basso costo.
- In pratica consiste di **mezzi rimuovibili** come floppy, CD-ROM, nastri magnetici.
- I floppy disk sono un esempio di dischi rimovibili magnetici;
  - sono costituiti di un disco sottile e flessibile ricoperto di materiale magnetico racchiuso in un involucro protettivo di plastica.
- I comuni floppy disk hanno una capacità di memorizzazione di circa 1MB e possono operare a velocità confrontabili con quelle dei dischi.
- Il rischio che la loro superficie sia danneggiata da graffi è elevato in quanto le testine di lettura-scrittura sono a diretto contatto con la superficie del disco.



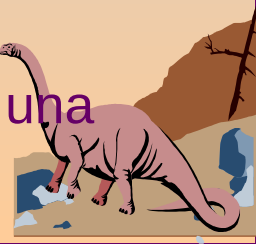
Dischi rimuovibili

- Altro esempio di dischi rimuovibili sono i dischi magneto-ottici,
- In questo caso i dati sono registrati su un disco ricoperto di materiale magnetico,
  - la tecnologia impiegata per la registrazione è diversa da quella dei dischi magnetici.
- La testina di un disco magneto-ottico è sospesa a una distanza dalla superfìcie del disco molto maggiore rispetto alla testina di un disco magnetico.
- Il materiale magnetico è protetto da uno spesso strato di plastica o di vetro.
- Di conseguenza, il disco è molto più resistente a eventuali collisioni con la testina.
- Un'altra categoria di dischi rimovibili è quella dei *dischi ottici,*
  - i quali non sfruttano affatto il magnetismo, ma usano dei materiali speciali che la luce laser è in grado di alterare in modo da creare punti relativamente chiari o scuri: ognuno di essi rappresenta un bit.



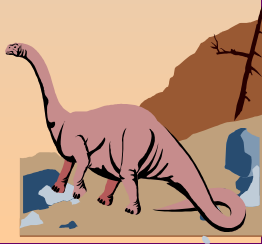
Dischi rimuovibili (II)

- I tipi di dischi fin qui descritti hanno la caratteristica di poter essere riutilizzati: per questo sono detti a lettura e scrittura.
- Per contro, i dischi *monoscrivibili* o WORM *(write once, read many limes)* costituiscono una categoria distinta.
- Uno dei modi di costruire un disco WORM è di inserire una pellicola di alluminio tra due piatti di plastica o di vetro.
- Per scrivere un bit l'unità usa un raggio laser per praticare un piccolo foro nell'alluminio:
  - visto che questo processo non è reversibile, su un qualunque settore del disco si può scrivere una sola volta.
- Sebbene sia possibile distruggere l'informazione contenuta in un disco WORM, ad esempio praticando fori dappertutto, è praticamente impossibile alterare i dati in esso contenuti,
  - perché l'unica azione possibile è quella di aggiungere fori, ed è probabile che il codice ECC associato a ogni settore rilevi le modifiche.
- I dischi a sola lettura, ad esempio i CD-ROM e i DVD sono commercializzati con un contenuto preregistrato, fanno uso di una tecnologia simile ai dischi WORM e sono assai durevoli.



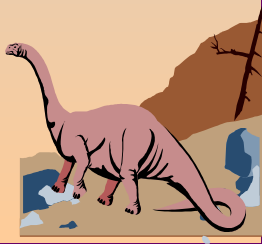
Nastri

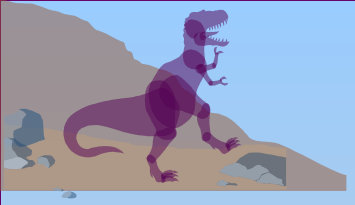
- I nastri sono un altro tipo di mezzo rimuovibile.
- In generale un nastro può contenere più dati rispetto ad un disco.
- La velocità di trasferimento è simile, ma l'accesso diretto è molto più lento.
- Un'unità a nastro è più costosa di un'unità a disco, ma una cartuccia a nastro è più economica di un disco magnetico della stessa capacità.
- I nastri si usano quindi per contenere copie di riserva dei dati presenti nei dischi.
- Il basso costo della memoria terziaria deriva dall'avere molte cartucce, che sono economiche, su pochi drive, che sono invece costosi.



Compiti del sistema operativo

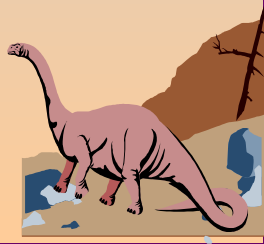
- Due tra gli obiettivi primari di un sistema operativo sono la gestione dei dispositivi fisici e la presentazione di una macchina virtuale alle applicazioni.
- Il sistema operativo realizza due astrazioni concernenti i dischi:
  - Il dispositivo a basso livello (il dispositivo visto come un semplice vettore di blocchi di dati)
  - Il file system





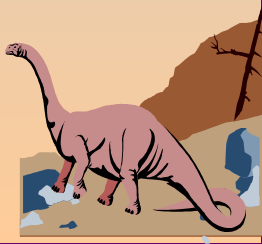
Interfaccia per le applicazioni

- La maggior parte dei sistemi operativi gestisce i dischi rimuovibili pressoché nella stessa maniera dei dischi fissi.
- Quando si inserisce un nuovo disco nella relativa unità esso deve essere formattato, quindi si crea su disco rimuovibile un file system vuoto che si usa proprio come il file system dei dischi fissi.
- Invece per i nastri il sistema operativo presenta un nastro come un mezzo di memorizzazione a basso livello:
  - una applicazione non apre un file su nastro, apre l'intera unità a nastro come dispositivo a basso livello.
- Quando un'unità a nastro è presentata come dispositivo a basso livello, S.O. non fornisce i servizi del file system, è l'applicazione che deve decidere come interpretare i blocchi letti.



Nomi dei file

- L’assegnazione dei nomi ai file su mezzi rimuovibili presenta alcune difficoltà,
  - soprattutto nel caso che il mezzo venga usato su più elaboratori diversi.
- I S.O. più diffusi in genere lasciano all’applicazione ed all’utente la chiave di lettura e di interpretazione dei dati.
- Alcuni tipi di mezzi rimuovibili (ad es. i CD o i DVD) sono ormai standardizzati,
  - nel senso che qualsiasi unità di lettura o sistema operativo sono in grado di gestirli.

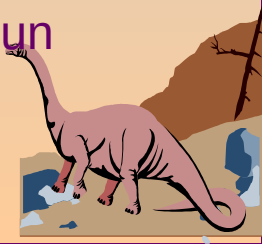


Gestione gerarchica della memoria

- Un sistema di gestione gerarchica della memoria estende la gerarchia di memorizzazione oltre la memoria centrale e secondaria comprendendo la memoria terziaria.
- Quest'ultima è di solito costituita da un **juke-box automatico** di nastri o di dischi rimuovibili.
- La tecnica più comune per estendere la gerarchia di memorizzazione fino alla memoria terziaria consiste nell'ampliare il file system.
- I file piccoli o frequentemente usati rimangono nei dischi magnetici.
- I file vecchi, ingombranti e raramente necessari si archiviano nel juke-box.
- La gestione gerarchica della memoria si trova di solito in centri di calcolo basati su supercalcolatori ed in altri grandi sistemi che posseggono grandi quantità di dati.

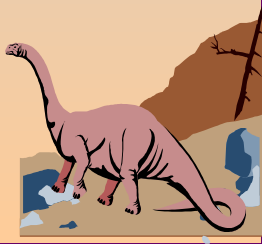
Prestazioni : Velocità

- I tre aspetti più importanti riguardanti le prestazioni della memorizzazione terziaria sono la velocità, l'affidabilità, i costi.
- La velocità è definita da due fattori: l'**ampiezza di banda** e la **latenza**.
- L'ampiezza di banda è di solito misurata in byte per secondo.
- **ampiezza di banda sostenuta**:
  - velocità media di trasferimento nel caso di una rilevante quantità di dati;
  - # di byte / tempo di trasferimento.
- **ampiezza di banda effettiva**:
  - # di byte trasferiti / tempo di I/O totale,
  - inclusi il tempo richiesto da una *seek* o una *locate* e l'attesa eventualmente dovuta a cambi di dischi o di nastri eseguiti dal juke-box.
- La **latenza d'accesso** è il tempo necessario a localizzare i dati.
  - Tempo di accesso per un disco < 35 millisecondi.
  - Tempo di accesso per un nastro: decine o centinaia di secondi.
  - In genere l'accesso casuale all'interno di un nastro è dell'ordine di un migliaio di volte dell'accesso casuale su disco.



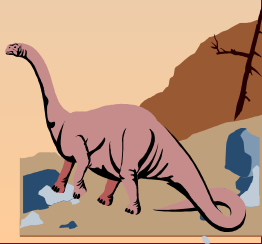
Affidabilità

- Un disco fisso è probabilmente più affidabile di un disco rimuovibile o di un'unità a nastro.
- Un disco ottico è più affidabile dei dischi e dei nastri magnetici.
- Anche le unità a disco fisso hanno punti deboli:
  - la collisione della testina col disco in genere distrugge i dati mentre il guasto di un'unità a nastro o a dischi ottici lascia spesso intatto il mezzo di memorizzazione.

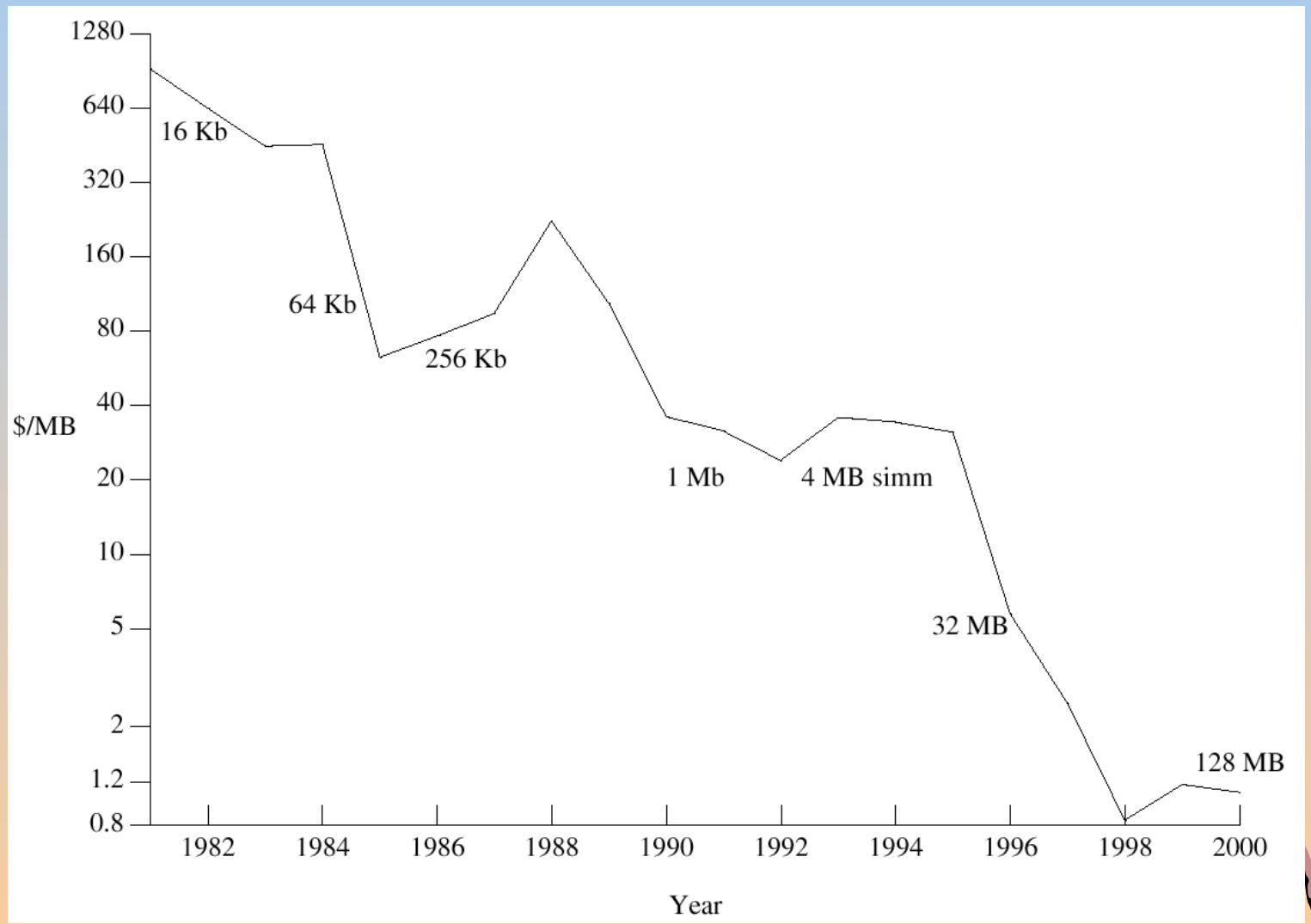


Costi

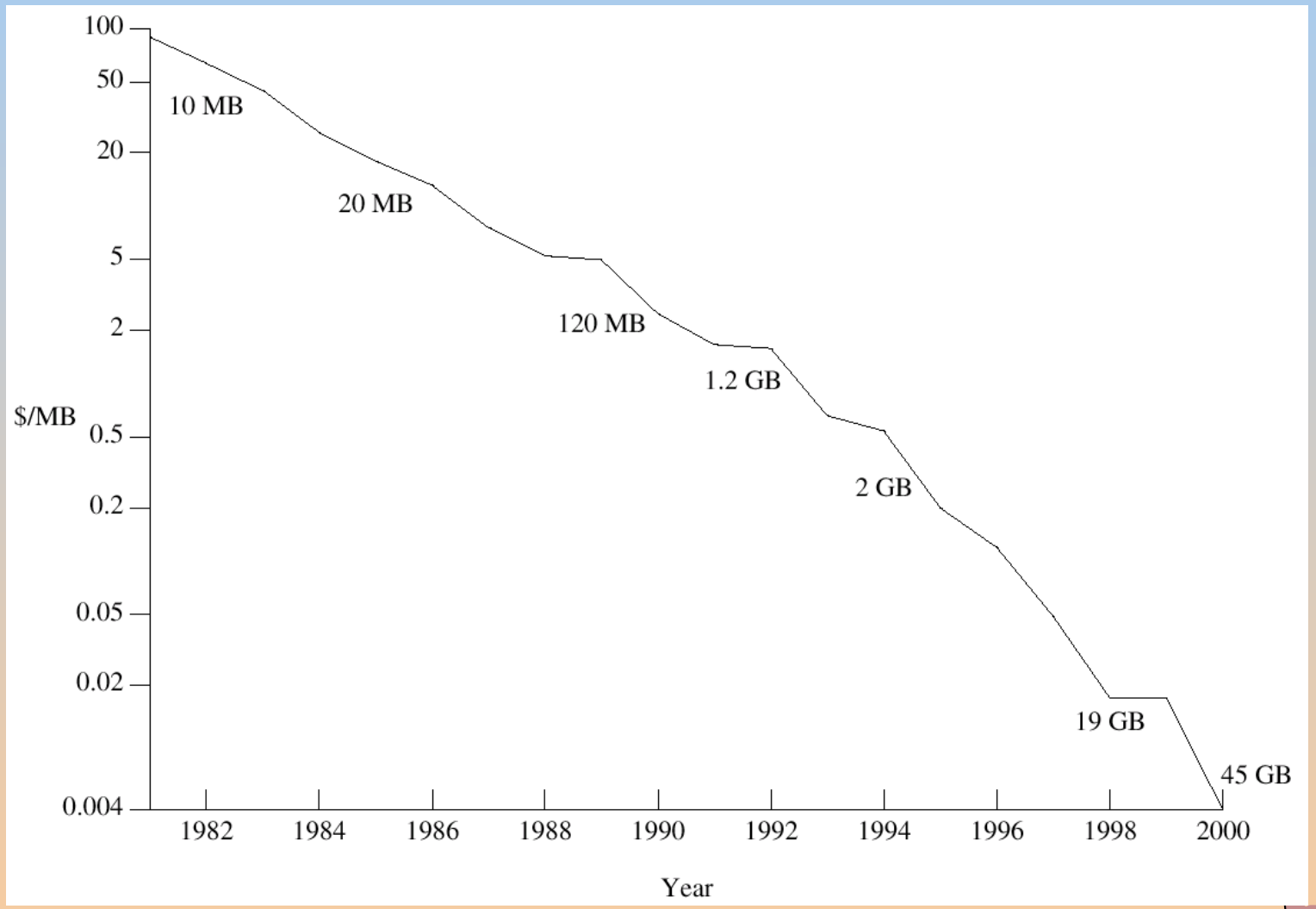
- Nella valutazioni delle prestazioni uno dei fattori più importanti è il costo,
  - in quanto il rapporto qualità–costo determina spesso il successo o il fallimento di un dispositivo.
- La memoria centrale è in genere più costosa della memoria secondaria.
- Il costo per megabitye della memoria su disco si avvicina ormai al prezzo dei nastri magnetici,
  - escludendo il costo delle unità a nastro.
- Quindi ormai gli archivi di nastri medi o piccoli hanno un costo di memorizzazione dei dati maggiore di un sistema a dischi di corrispondente capacità.
- Non si dispone quindi più di una tecnologia di memoria terziaria che sia di più ordini di grandezza più economica dei dischi magnetici.



Prezzo per Megabyte della memoria DRAM, dal 1981 al 2000



Prezzo per Mb delle unità a disco magnetico, dal 1981 al 2000



Prezzo per Mb delle unità a nastro, dal 1981 al 2000