Gestione della memoria di massa

Programma – Sistemi Operativi

- Introduzione ai sistemi operativi
- Gestione dei processi
- Sincronizzazione dei processi
- Gestione della memoria centrale
- Gestione della memoria di massa
- File system
- Sicurezza e protezione

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 (flash)
- CD, DVD, Blu-ray

Dischi magnetici

I dischi magnetici rappresentano ancora oggi un mezzo molto diffuso per la memorizzazione di massa

- Rivestiti con materiale magnetico (ossido di ferro), erano originariamente in alluminio
- La tecnologia attuale, viceversa, è orientata all'utilizzo del vetro:
  - Con una superficie più uniforme si ottiene maggiore affidabilità (gli errori di lettura/scrittura sono meno frequenti)
  - Più rigido e più resistente agli urti
  - Permette di ridurre la distanza della testina dalla superficie

Dischi magnetici

https://youtu.be/kdmLvl1n82U

Dischi magnetici

Piatto (platter)

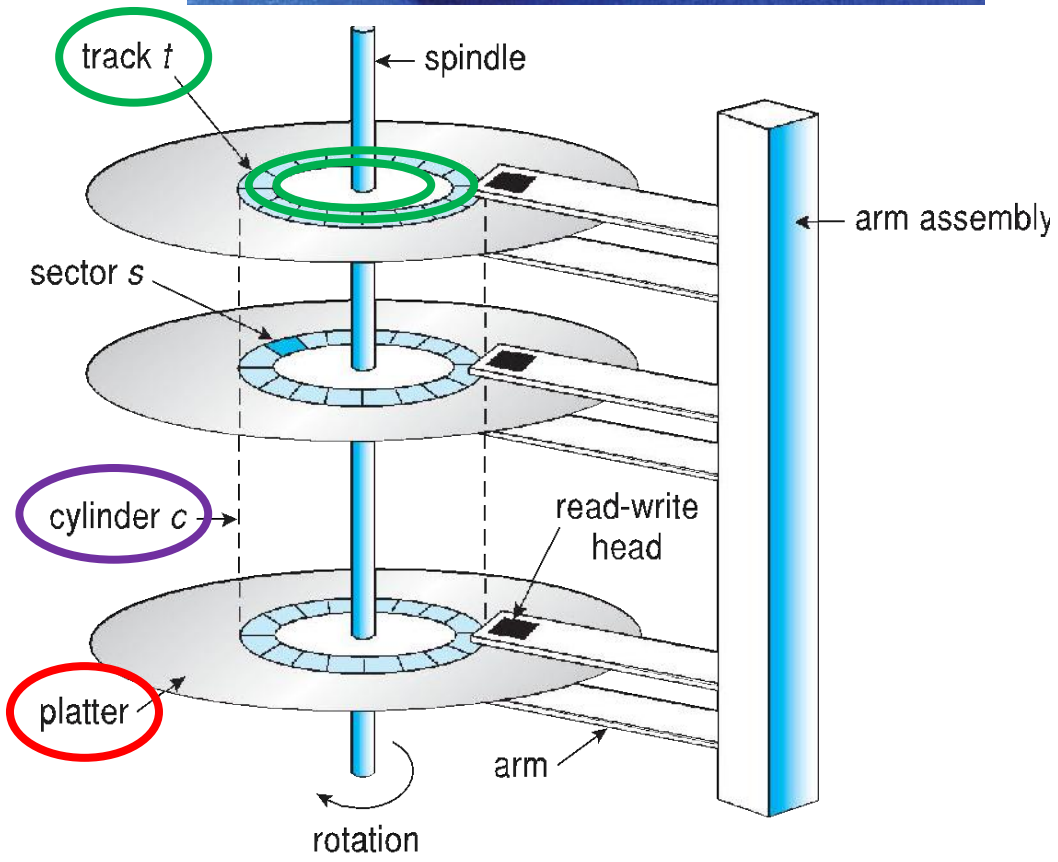
un disco rigido si compone di uno o più dischi paralleli, in cui ogni superficie, detta “piatto” e identificata da un numero univoco, è destinata alla memorizzazione dei dati

Traccia (track)

su ogni piatto, vi sono numerosi anelli concentrici, detti tracce, ciascuno identificato da un numero univoco

Cilindro (cylinder)

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



Dischi magnetici

Settore (sector)

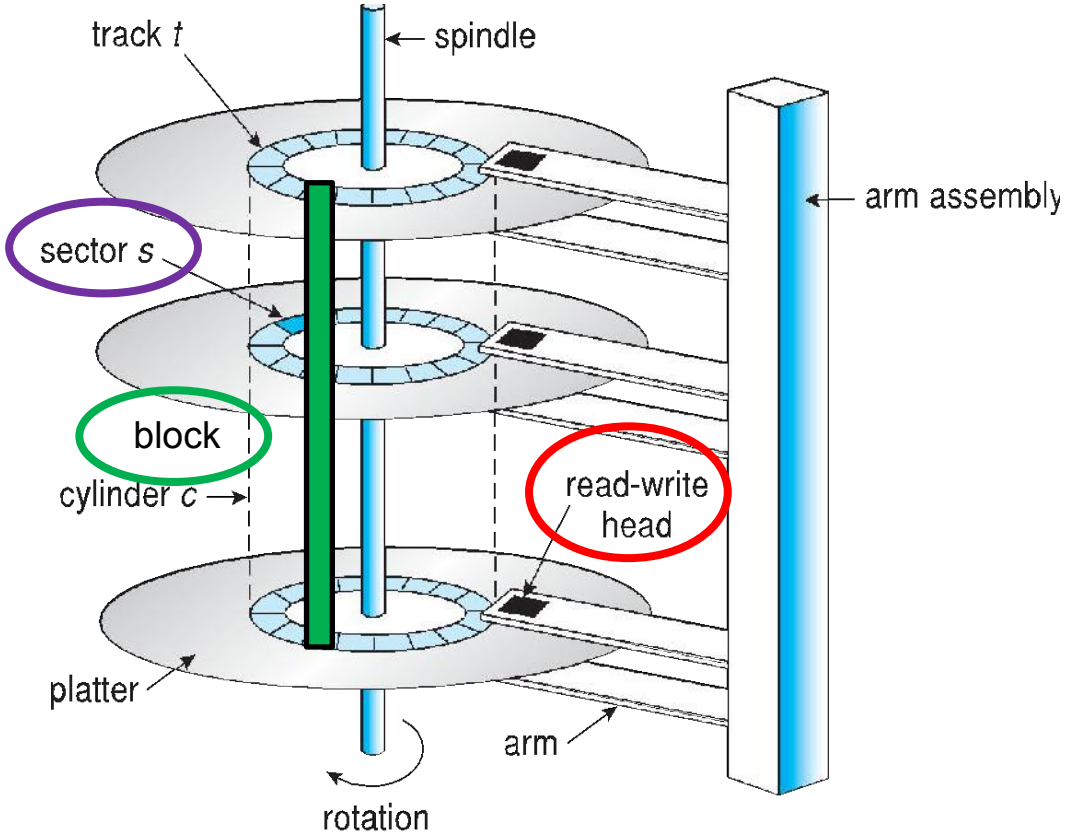
ogni traccia è suddivisa in settori, cioè in “spicchi” uguali, ciascuno identificato da un numero univoco

Blocco (block)

l’insieme dei settori posti nella stessa posizione in tutti i piatti

Testina (read-write head)

su ogni piatto è presente una testina di lettura/scrittura; la posizione di tale testina è solidale con tutte le altre sui diversi piatti: se una testina è posizionata sopra una traccia, tutte le testine saranno posizionate sul cilindro a cui la traccia appartiene



Dischi magnetici – Tempi di 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
- 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, 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)

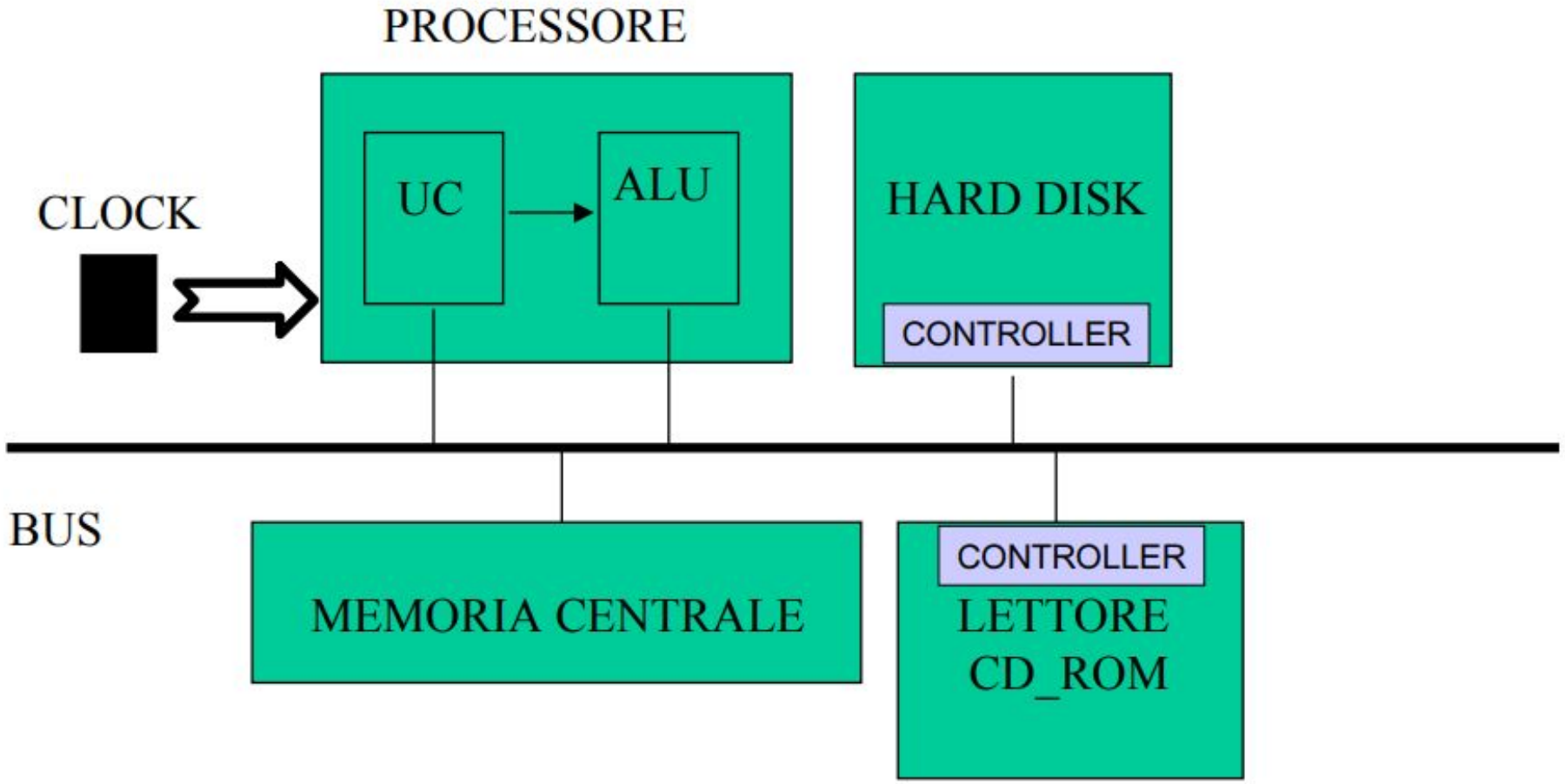
- Durante la lettura/scrittura, la testina è stazionaria, mentre il disco ruota
- Scrittura
  - la corrente, che fluisce nella bobina (nelle due possibili direzioni) produce un campo magnetico
  - le particelle aciculari dell'ossido di ferro si orientano in base al campo magnetico prodotto (0 e 1 memorizzati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

Dischi magnetici – Connessione

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), Fiber Channel, SCSI (Small Computer System Interface)

- 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



http://www.di.unito.it/~rossano/DIDATTICA/suism-0203/lezione7.pdf

Dischi magnetici – Controller

https://commons.wikimedia.org/wiki/File:Serial_ATA_hard_disk_connected.jpg

Dischi magnetici – I/O

- 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 agisce sugli elementi elettromeccanici 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 dei dati tra la cache e l’adattatore avviene alla velocità propria dei dispositivi elettronici

Dischi magnetici – Caratteristiche

- 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 attuale dei dischi si attesta fra 30GB e 15TB

- 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 / (RPM / 60) = 60 / RPM
  - Latenza media = ½ giro

Dischi magnetici – Tempi di accesso

- Tempo di accesso medio = seek time medio + latenza media

  Per i dischi più veloci si avrà 3msec + 2msec = 5msec

  Per dischi lenti si avrà 9msec + 5.55msec = 14.55msec

- Tempo medio di I/O = tempo medio di accesso +
  (quantità di dati da trasferire /
  velocità di trasferimento) +
  overhead

Dischi magnetici – Tempi di accesso

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 = 4KB/1Gb/s = 0.031 msec

Tempo medio di I/O per un blocco da 4KB =
5msec + 4.16msec + 0.1msec + tempo di trasferimento =
9.27msec + 0.031msec = 9.301msec

Il primo HD commerciale

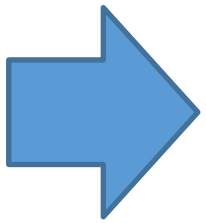
1956

Il computer IBM 305 RAMAC 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 NVM

- Spesso inseriti in chassis simili agli HDD e perciò denominati dischi a stato solido (SSD)
- 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 un costo al MB più elevato
- Possono avere vita più breve



Dispositivi NVM

- Hanno minore capacità, ma 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)
- I bus standard possono essere troppo lenti
- Collegamento al bus PCI di sistema con tecnologia NVM express

Dispositivi NVM

Le caratteristiche dei semiconduttori NAND aprono a nuove sfide per l'affidabilità

Le NAND si deteriorano ad ogni ciclo di cancellazione e dopo circa 100.000 cicli 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 5 DWPD si possono scrivere teoricamente 5 TB al giorno senza errori

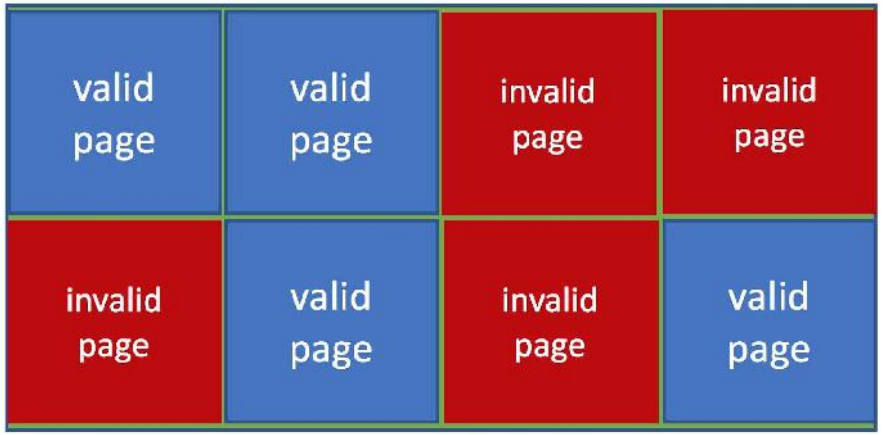
Dispositivi NVM

Letture e scritture avvengono con granularità di “pagina” (analogo del settore)

- Impossibilità di cancellazione per “sovra-scrittura”
- La scrittura è costosa: Il contenuto della pagina deve prima essere cancellato e le cancellazioni avvengono per “blocchi” (della dimensione di diverse pagine)

Dispositivi NVM

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

Garbage collection

- 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 nel frattempo copiati per poter essere riutilizzati) e solo a quel punto può iniziare a scrivere nuovi dati nel blocco che è stato così liberato

Overprovisioning

- In alternativa, assegna un overprovisioning (7-20%) 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

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`**

Blocchi logici

- Le unità a disco vengono indirizzate come giganteschi vettori monodimensionali di blocchi logici, dove il blocco logico rappresenta la minima unità di trasferimento

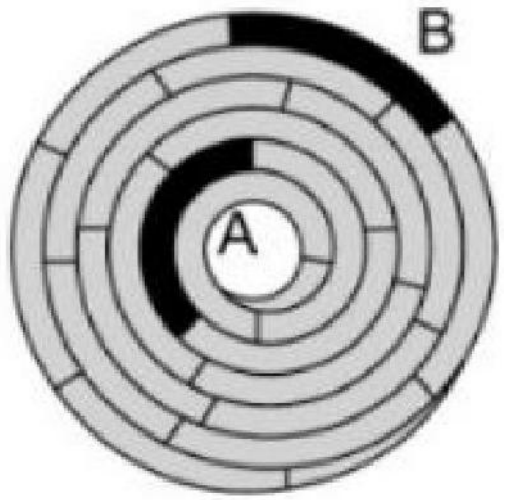
- I blocchi logici sono creati all'atto della formattazione di basso livello

Mappatura degli indirizzi

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 spo l'interno ($v = \omega r$), per mantenere co lineare e, quindi, la quantità di dati c le testine nell'unità di tempo
- CD e DVD
- Talvolta, si ha un'unica traccia a spira

Disco CLV con un'unica traccia a spirale



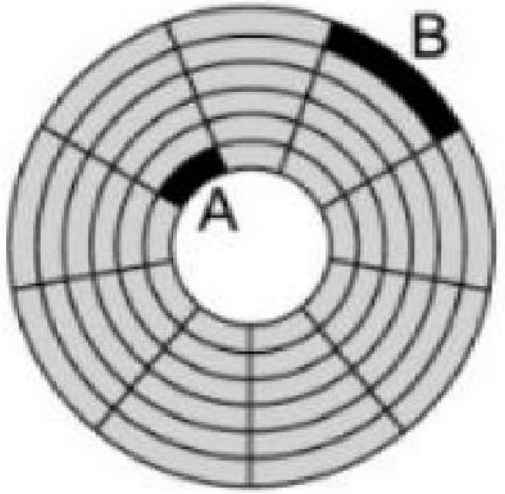
Mappatura degli indirizzi

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

Disco CAV a tracce concentriche



Scheduling del disco

Il SO è responsabile dell'uso efficiente dell'hardware: per i dischi 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

Scheduling del disco

Per migliorare le prestazioni si può intervenire solo sul tempo di ricerca e si tenta quindi di minimizzarlo

Il seek time è proporzionale alla distanza di spostamento fra le tracce

seek time ≈ seek distance

Bandwidth

L’ampiezza di banda (bandwidth) del disco è il numero totale di byte trasferiti, diviso per il tempo trascorso fra la prima richiesta e il completamento dell’ultimo trasferimento

Operazioni di I/O su disco

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

Algoritmi di scheduling del disco

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

Esistono diversi algoritmi di scheduling del disco per gestire la coda di richieste alla memoria secondaria

Algoritmo di scheduling del disco FCFS

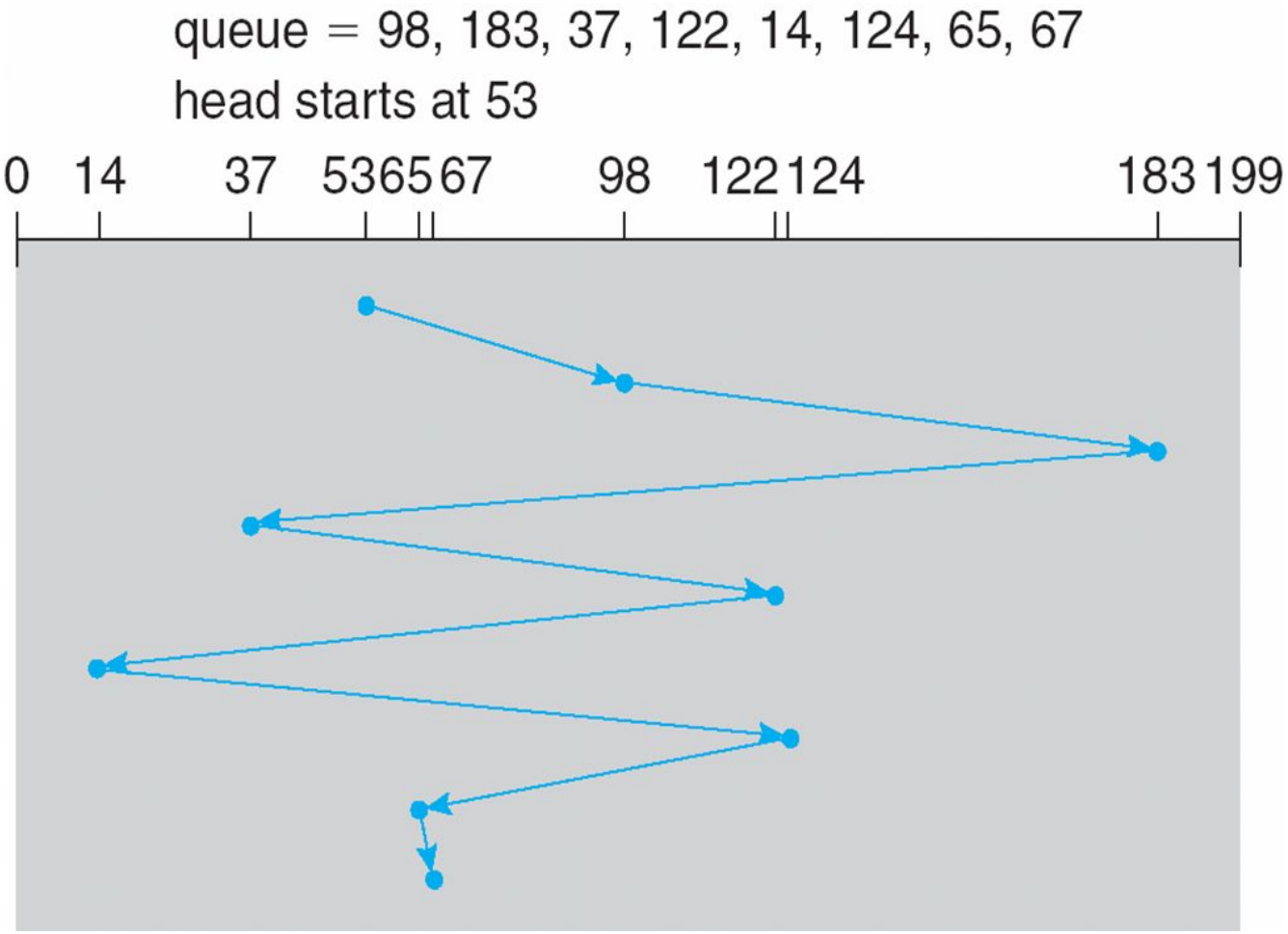
FCFS First Come First Served

Lista di richieste da esaudire:
98, 183, 37, 122, 14, 124, 65, 67
Testina correntemente sul cilindro 53, cilindri totali 200 (0-199)

La distanza (in cilindri) da coprire per la testina sarà
$|53 – 98| + |98 – 183| + |183 – 37| + |37 – 122| +$
$|122 – 14| + |14 – 124| + |124 – 65| + |65 – 67| =$
$45+85+96+35+108+110+59+2 = 640$

Algoritmo di scheduling del disco FCFS

FCFS First Come First Served



Viene generato un movimento totale della testina pari a 640 cilindri

Algoritmo di scheduling del disco SCAN

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

Algoritmo di scheduling del disco SCAN

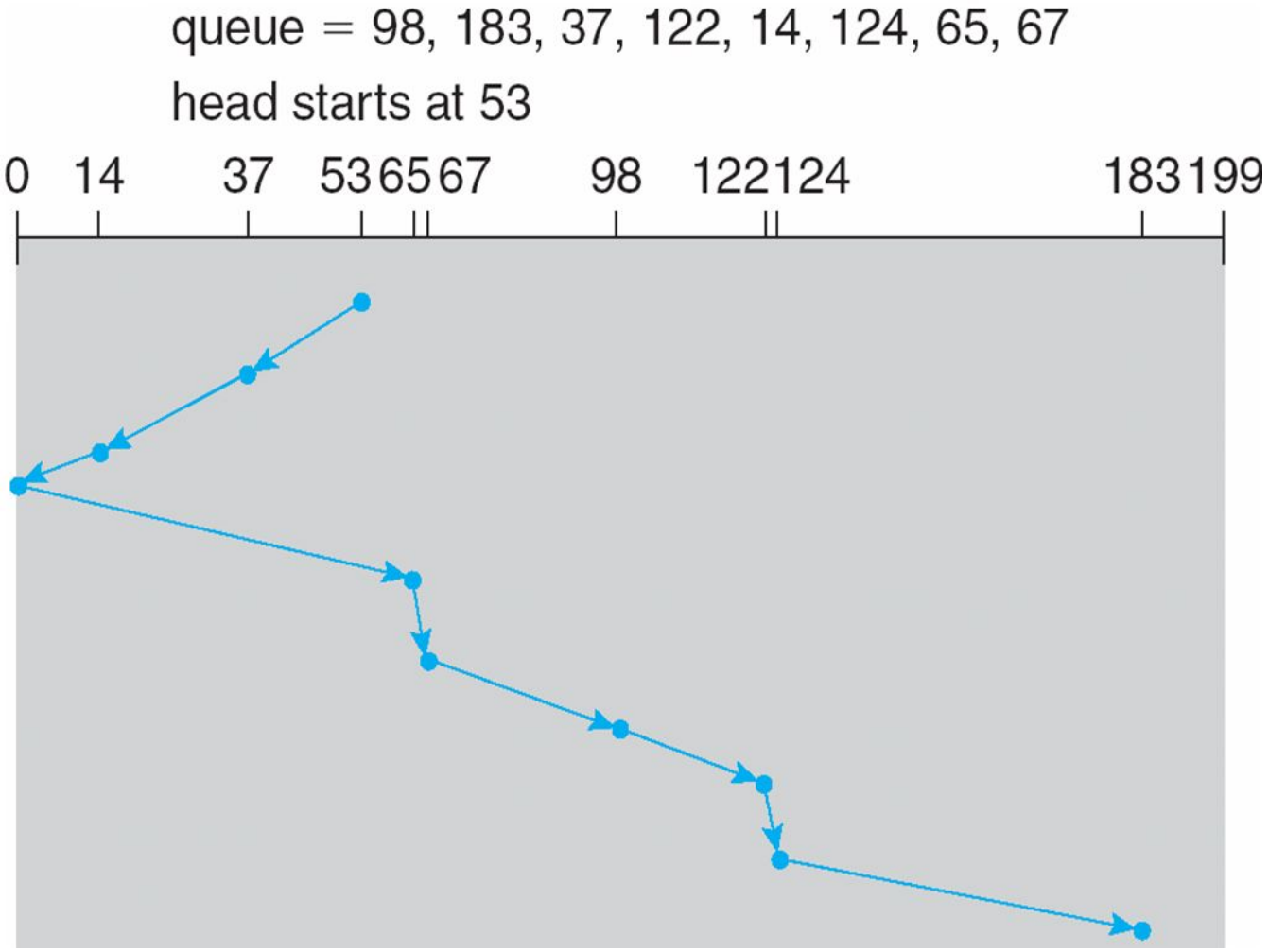
Lista di richieste da esaudire:
98, 183, 37, 122, 14, 124, 65, 67
Testina correntemente sul cilindro 53, cilindri totali 200 (0-199)

Scheduling SCAN (si supponga che il movimento sia verso il cilindro 0)
37, 14, 0, 65, 67, 98, 122, 124, 183

La distanza (in cilindri) da coprire per la testina sarà
$|53 – 37| + |37 – 14| + |14 – 0| + |0 – 65| + |65 – 67| +$
$|67 – 98| + |98 – 122| + |122 – 124| + |124 – 183| =$
$16+23+14+65+2+31+24+2+59 = 236$

Algoritmo di scheduling del disco SCAN

SCAN



Algoritmo di scheduling del disco SCAN

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

Algoritmo di scheduling del disco C-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 garantisce un tempo di attesa più uniforme rispetto a SCAN

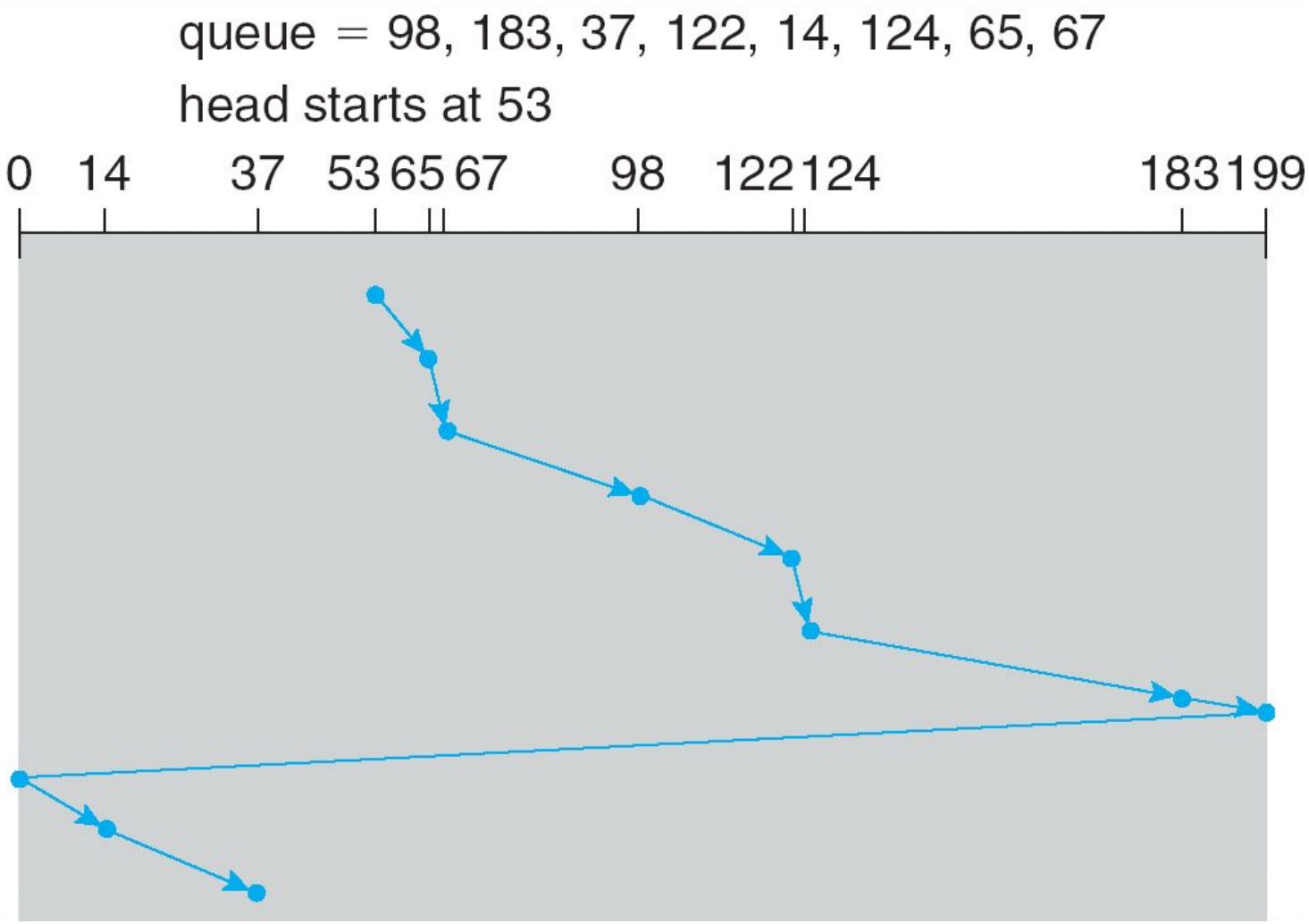
Algoritmo di scheduling del disco C-SCAN

Lista di richieste da esaudire:
98, 183, 37, 122, 14, 124, 65, 67
Testina correntemente sul cilindro 53, cilindri totali 200 (0-199)

Scheduling C-SCAN (supponendo movimento verso il cilindro 199)
65, 67, 98, 122, 124, 183, 199, 0, 14, 37

Algoritmo di scheduling del disco C-SCAN

C-SCAN



Scelta di un algoritmo di scheduling del disco

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

Le richieste di I/O per l'unità a disco possono essere influenzate dal metodo di allocazione di file e directory

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 lettura) relative a indirizzi logici adiacenti

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

Rilevamento e correzione di errori

- 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

- Gli errori correggibili sono detti soft
- Gli errori rilevati ma non corretti sono definiti hard

Formattazione

- Con formattazione di basso livello (o fisica) si intende la suddivisione del disco in settori che possono essere letti e scritti dal controllore del disco

- Salvataggio su disco di una struttura dati per ogni settore (intestazione/coda/ECC) con dimensione standard pari a 512 byte

Partizionamento

- Per poter impiegare un disco per memorizzare i file, il SO deve mantenere le proprie strutture dati sul disco

- Si partiziona il disco in uno o più gruppi di cilindri, ognuno dei quali rappresenta un “disco logico”

- Formattazione logica o “creazione di un file system”

- 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

Partizione di boot

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, si verifica la coerenza del file system (controllando la correttezza dei metadati)

- Si aggiorna la tabella di montaggio
- Il blocco di avvio può puntare al volume di avvio o all'insieme di blocchi contenenti il caricatore di avvio, ovvero codice sufficiente per caricare il kernel dal file system

Partizione di boot

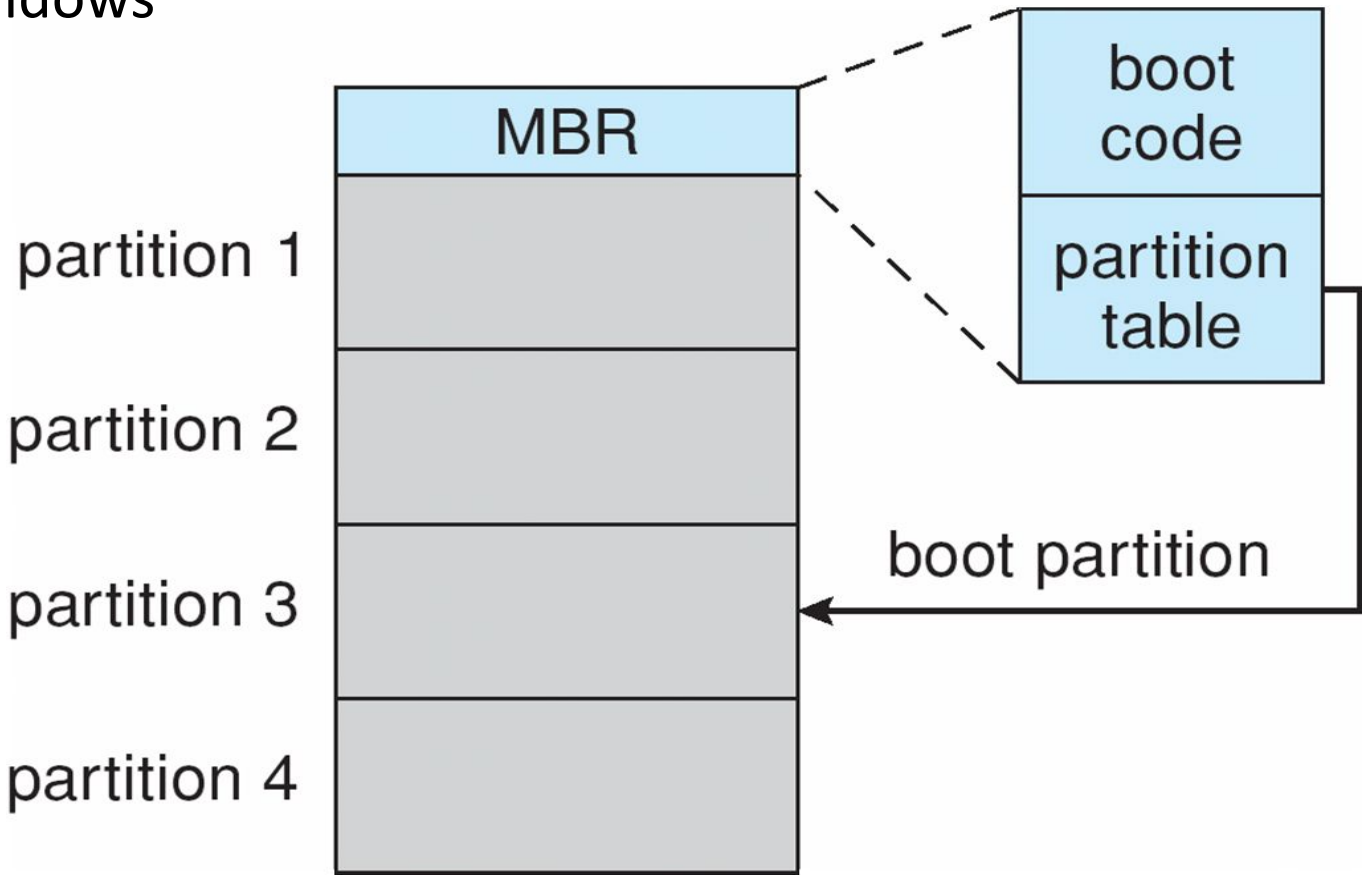
Nel boot block sono contenute le informazioni necessarie all'inizializzazione del sistema

Il bootstrap loader è memorizzato nella ROM

Il bootstrap completo è memorizzato in posizione fissa nell'hard disk.
Per esempio, nel primo settore del disco di avviamento (o disco di sistema)

Boot in Windows

Booting from secondary storage in Windows



Accantonamento dei settori

- 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

Gestione dell’area di swap

- La memoria virtuale impiega lo spazio su disco come un’estensione della memoria centrale

- L’obiettivo principale nella progettazione e realizzazione dell’area di swap è di fornire la migliore produttività per il sistema di memoria virtuale

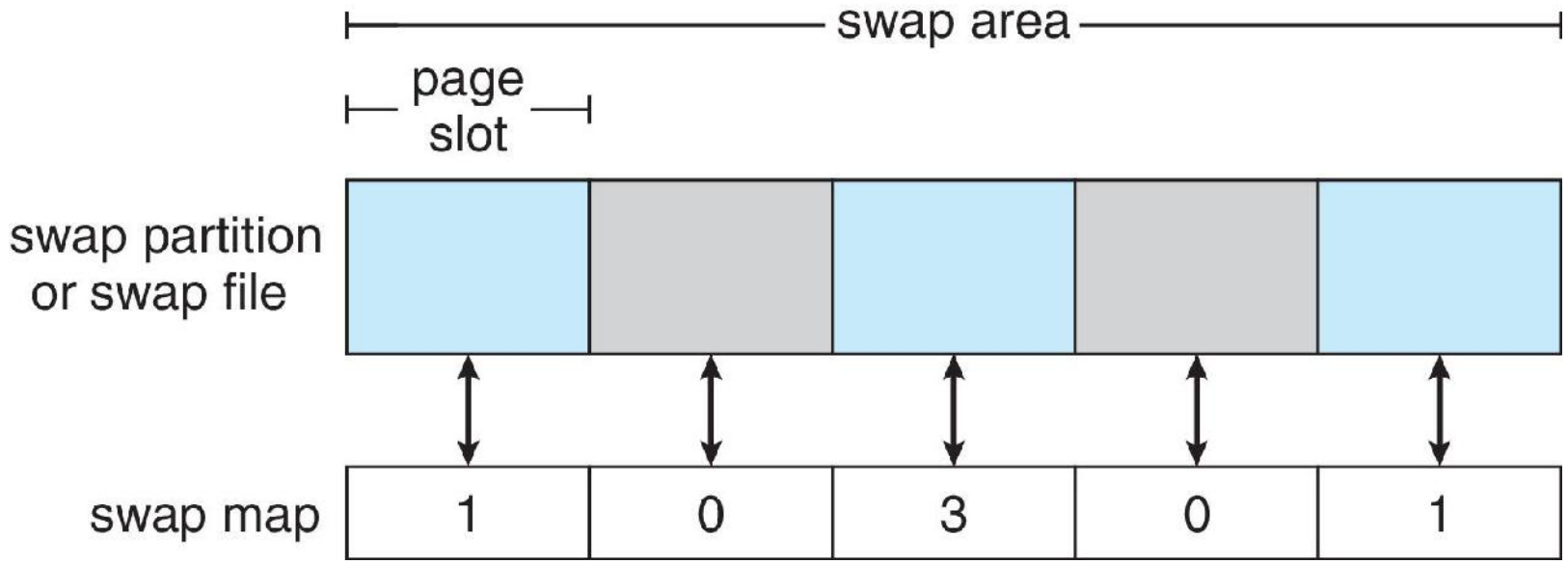
- Lo spazio di swap può essere ricavato all’interno del normale file system o, più comunemente, si può trovare in una partizione separata del disco

Area di swap in Linux

- L’area di swap, in Linux, è utilizzata solo per la memoria anonima, ovvero per dati che non corrispondono a file
- Linux permette l’istituzione di una o più aree di avvicendamento, sia in file che in una partizione raw
- 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

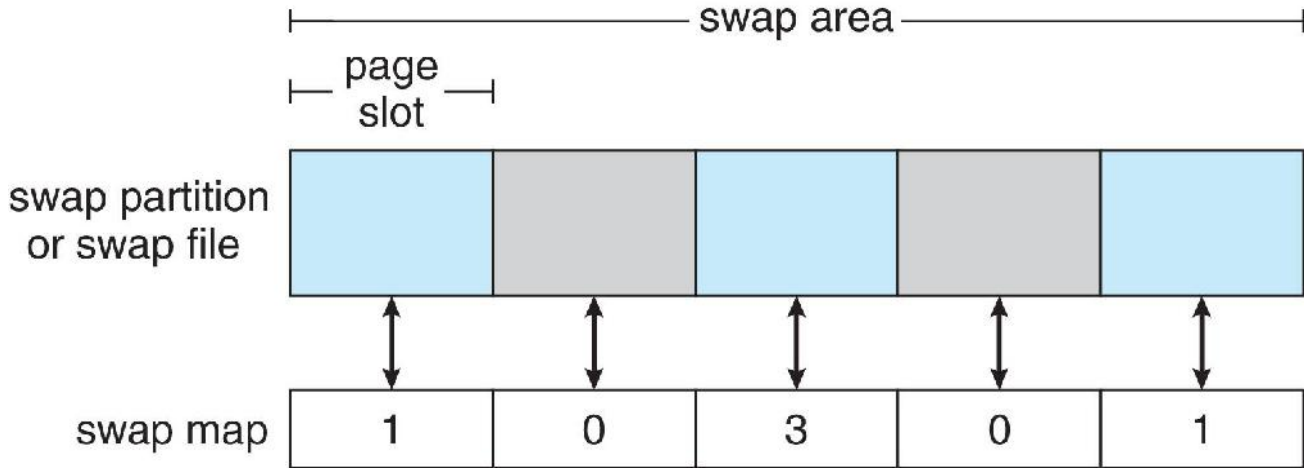
Swap map

Ogni swap area dispone di una mappa di avvicendamento (swap map), un array di contatori interi, ciascuno dei quali corrisponde ad uno slot dell'area



Swap map

- 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 (host-attached)
- Tramite un dispositivo connesso alla rete (network-attached)
- In cloud

Host-attached storage

Alla memoria secondaria connessa alla macchina si accede dalle porte locali di I/O che sono collegate al bus

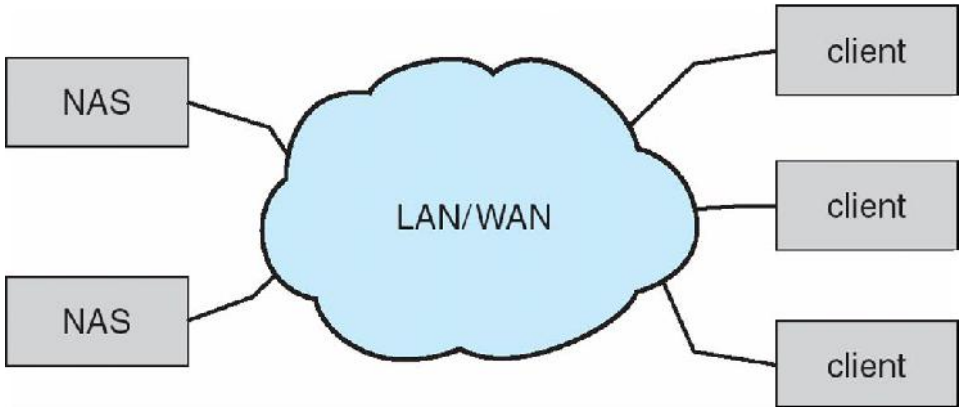
- Nei PC, con interfaccia ATA o SATA, due unità (al più) per ciascun bus di I/O
- La tecnologia SCSI è un'interfaccia standard progettata per realizzare il trasferimento di dati, che permette la connessione di un massimo di 16 device
- FC (Fiber Channel) è un'architettura seriale ad alta velocità
  - Può gestire uno spazio di indirizzi a 24 bit, che è alla base delle storage area network (SAN), nelle quali molti host sono connessi con altrettante unità di memorizzazione

Network-Attached Storage

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 di dati

- I client accedono alla memoria connessa alla rete utilizzando specifici protocolli quali NFS (UNIX) e CIFS (Windows)

- L'implementazione avviene via remote procedure calls (RPCs) tra host e memoria tipicamente usando TCP o UDP su rete IP



Cloud storage

- In maniera simile 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
- Si impiegano le API a causa delle lunghe latenze e per i numerosi scenari di errore

Cloud storage

Esempi di cloud storage

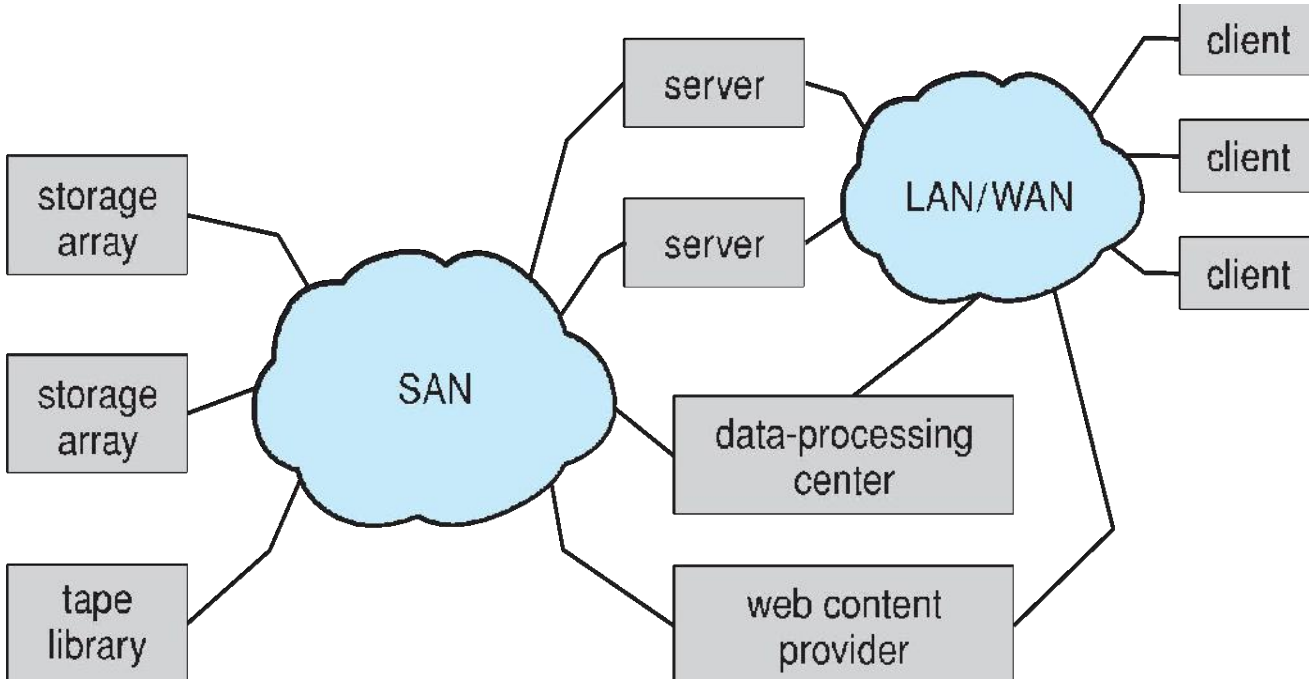
- Dropbox
- Amazon S3
- Microsoft OneDrive
- Apple iCloud
- Google drive

Google Drive

iCloud

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



Storage array

Dispositivo costruito appositamente che può includere porte SAN, porte di rete o entrambe.

Contiene

- Unità per la memorizzazione dati
- Controllore (CPU + memoria + software per le funzionalità dell’array)

Strutture RAID

- RAID, Redundant Array of Independent Disks
  L'affidabilità del sistema di memorizzazione viene garantita tramite la ridondanza

- Aumento del tempo medio di guasto

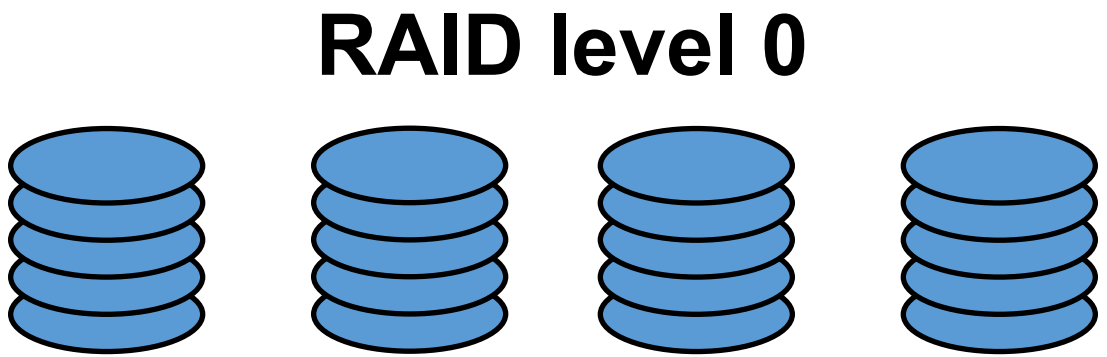
- Spesso affiancati dalla presenza di NVRAM per garantire la consistenza dei dati scritti "contemporaneamente" su dischi multipli e per migliorare le performance

- Inoltre… le tecniche per aumentare la velocità di accesso al disco implicano l'uso di più dischi cooperanti

RAID 0

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



Sezionamento senza ridondanza

Vantaggi dello Striping

- Aumento del throughput per accessi multipli a pagine in memoria
- Diminuzione del tempo di risposta per l’accesso a grandi quantità di dati

Svantaggi dello Striping

- Non aumenta l'affidabilità

- Se il tempo medio di guasto di una unità disco è pari a 100000 ore, allora il tempo medio di guasto per una batteria di dischi con 100 unità sarà 100000/100 = 1000 ore, davvero poco!

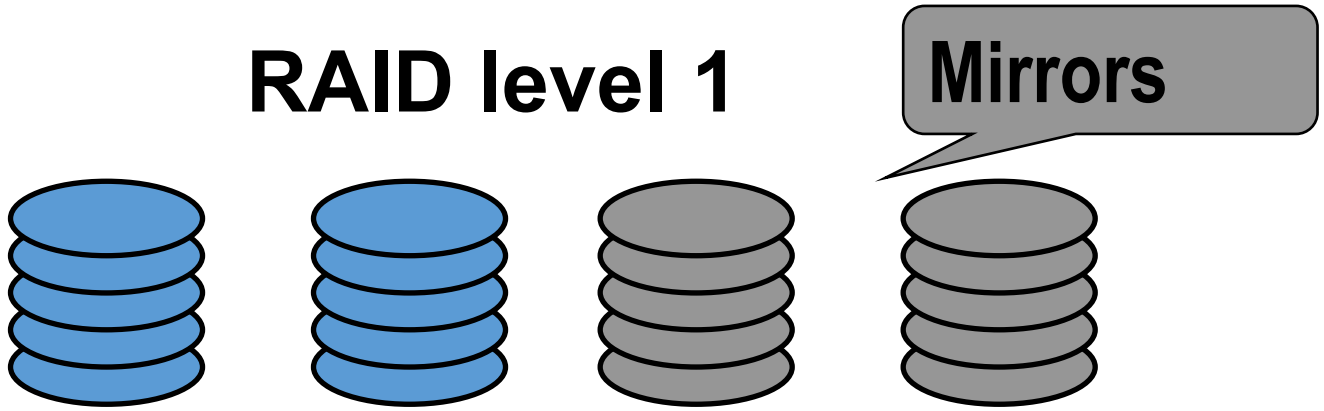
Mirroring

- Soluzione al problema dell'affidabilità
- Due copie di ogni blocco di dischi
- Vantaggi:
  - Semplice da implementare
  - Resistente ai guasti
- Svantaggio:
  - Richiede il doppio della capacità rispetto ad un normale sistema

RAID 1

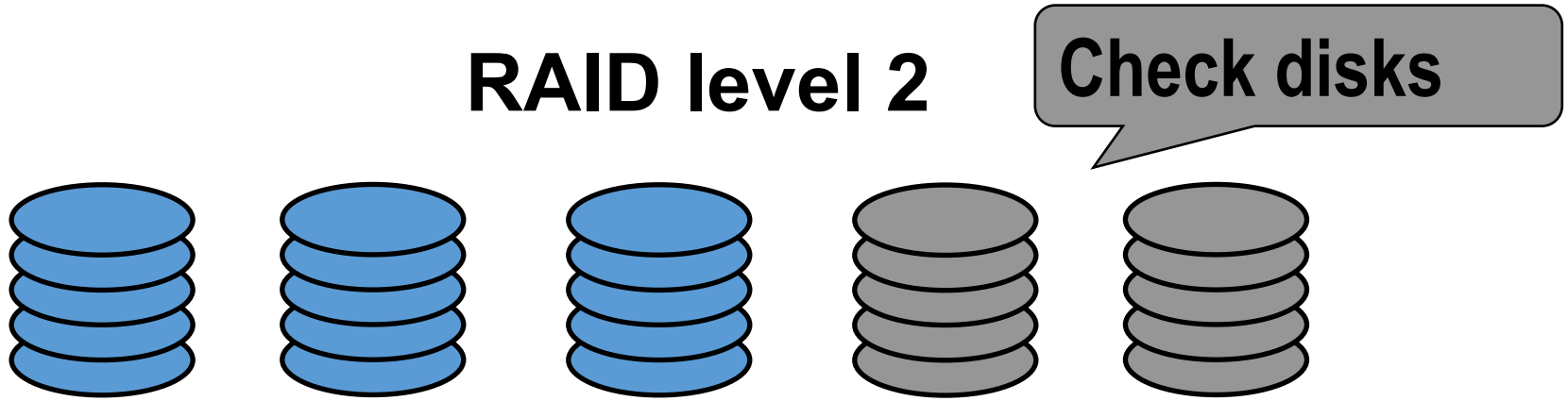
- Gli schemi RAID migliorano prestazioni e affidabilità del sistema memorizzando dati ridondanti

- Il mirroring o shadowing (RAID 1) conserva duplicati di ciascun disco



RAID 2

bit level striping

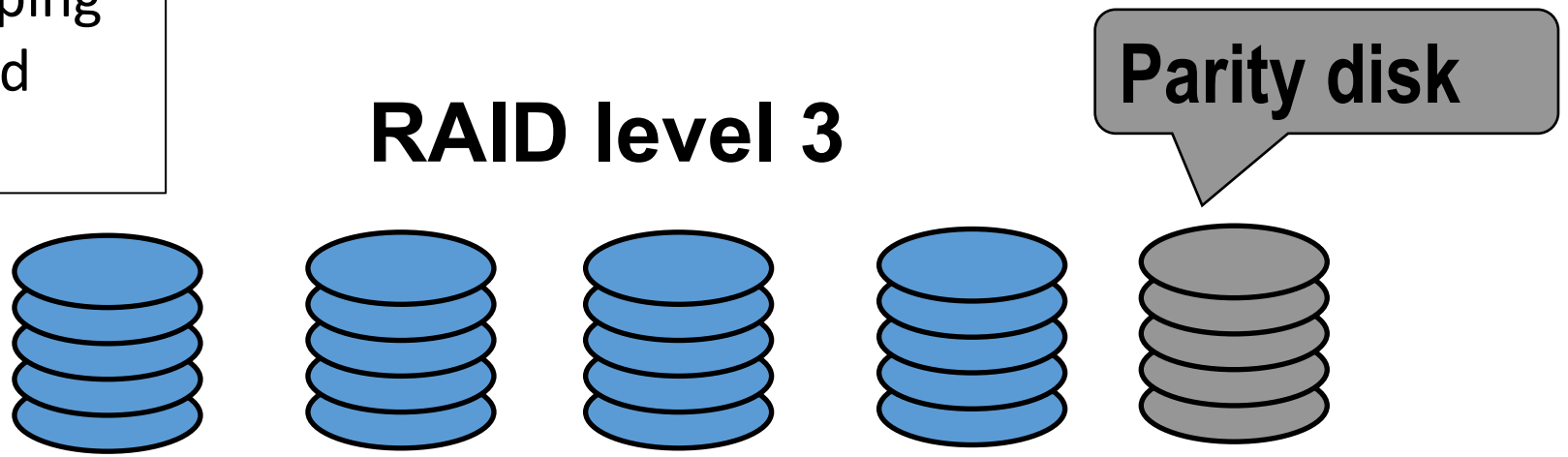


- Si usa il codice di Hamming per correggere gli errori
- E’ necessario sincronizzare i dischi per far sì che la testina di ciascun disco sia nella stessa posizione in ogni disco
- Poiché i moderni HDD hanno sistemi a correzione di errori integrati, RAID2 è considerato obsoleto

RAID 3

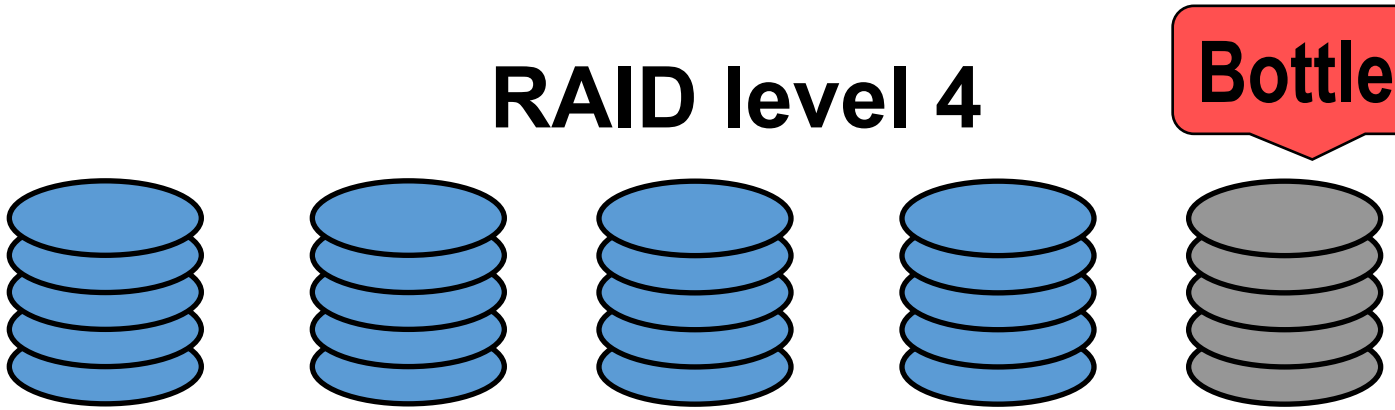
- Dati salvati in stripe della lunghezza di 1 byte
- Il byte di parità si determina per ogni riga di dati e viene salvato nel cosiddetto “parity disk”
- RAID-3 è poco usato perché non può eseguire richieste multiple simultaneamente dato che ogni singolo blocco di dati è memorizzato in modo distribuito tra tutti i dischi del RAID

byte level striping with dedicated parity disk



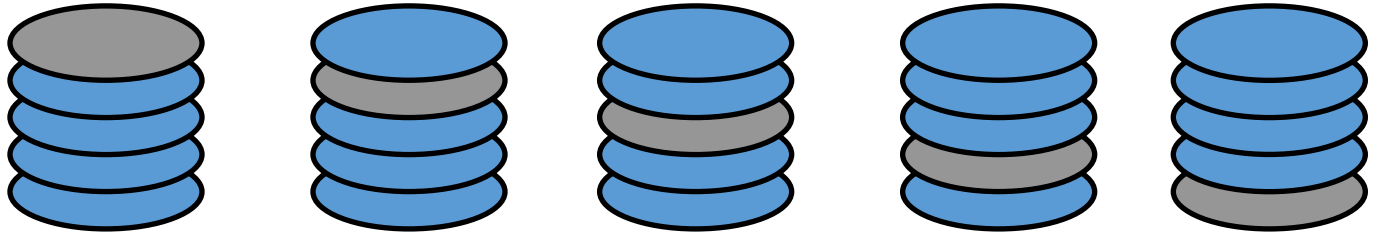
RAID 4 e RAID 5

RAID level 4



block-level striping with a dedicated parity disk

RAID level 5

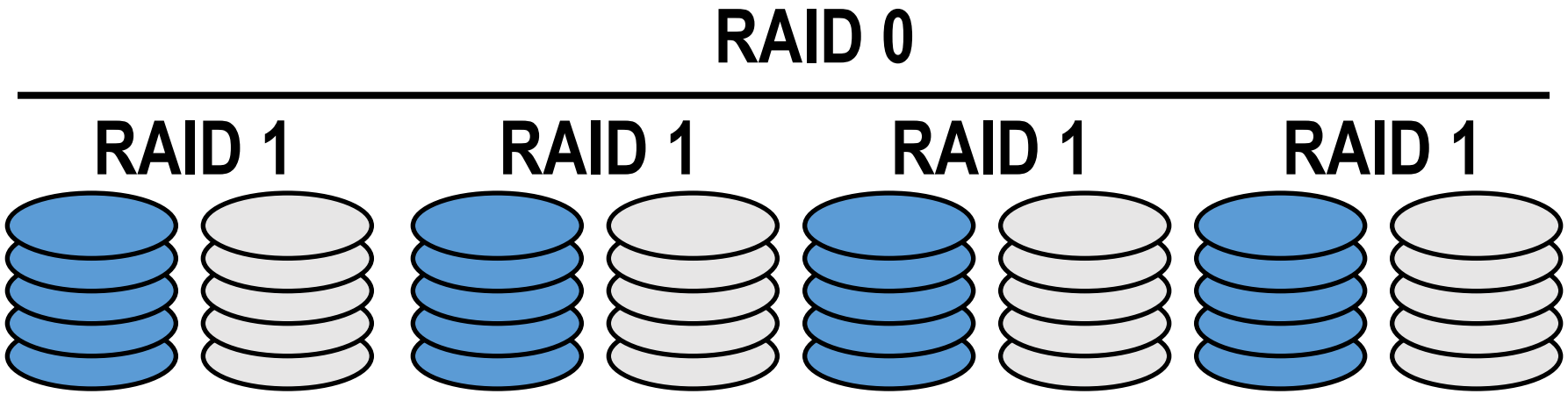


distributed parity

RAID 10

Anche noti come RAID 1+0

I dati sono divisi in stripe (come in RAID 0) su coppie di dischi duplicati (RAID 1)



Sistema operativo e I/O

Il ruolo di un sistema operativo nell'I/O è quello di gestire e controllare le operazioni e i dispositivi di I/O

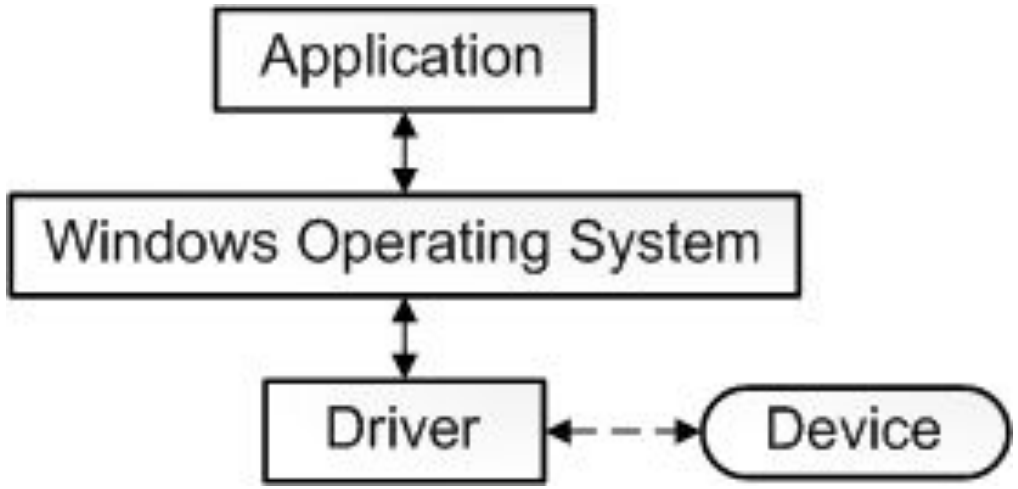
Sottosistema di I/O

- I dispositivi di I/O possono essere molto diversi per funzioni e velocità, quindi necessitano di diversi sistemi di controllo

- Il sottosistema di I/O del kernel separa il resto del kernel dalla complessità di gestione dei dispositivi di I/O

Driver di dispositivo

I driver dei dispositivi offrono al sottosistema di I/O una interfaccia uniforme per l'accesso ai dispositivi di I/O



https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/what-is-a-driver-

Hardware di I/O

Se più dispositivi condividono un insieme di fili, la connessione è detta *bus*.

Un **bus** è un insieme di fili e un protocollo rigorosamente definito che specifica l'insieme dei messaggi che si possono inviare attraverso i fili.

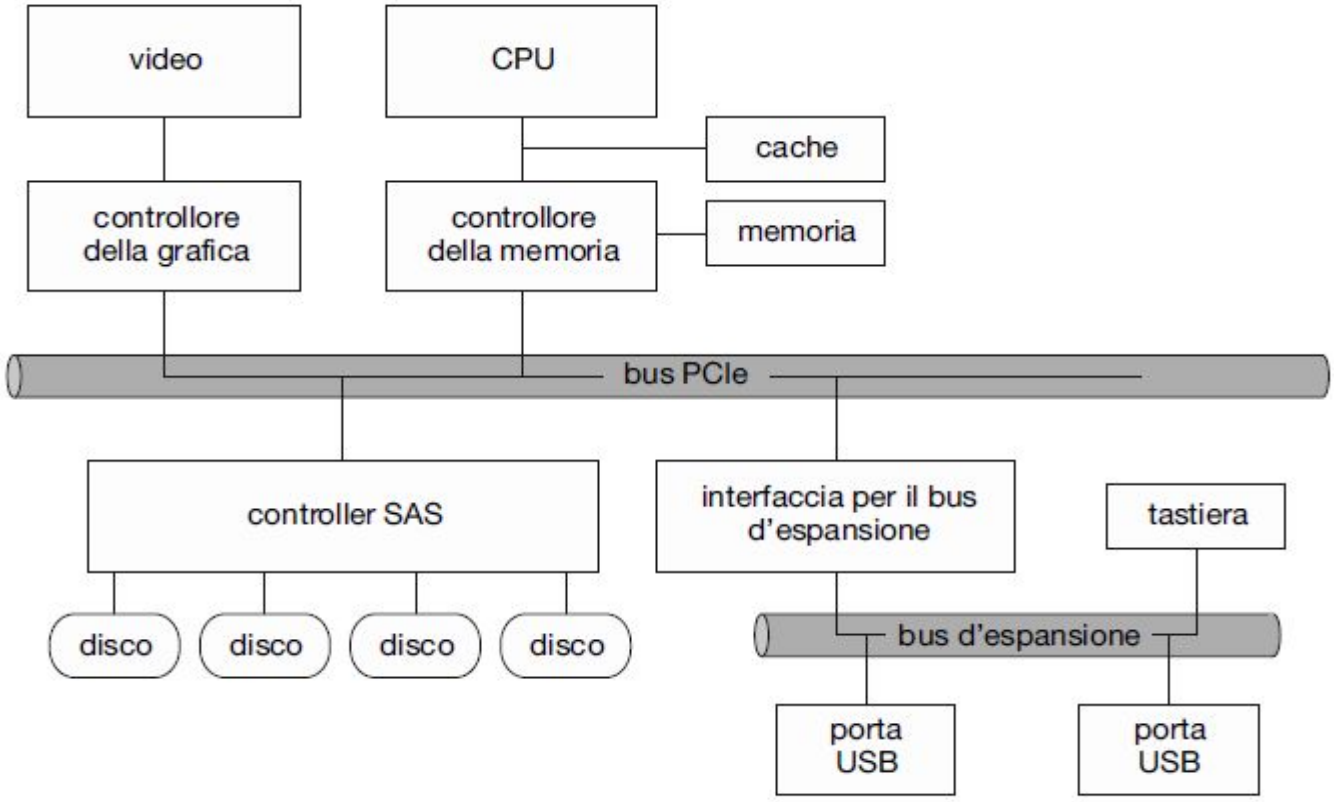


Figura 12.1 Tipica struttura del bus di un PC.

Memory mapped I/O

Il controllore di dispositivo può supportare l'**I/O memory mapped** *(I/O mappato in memoria).*

- I registri di controllo del dispositivo sono mappati in un sottoinsieme dello spazio di indirizzi della CPU

- La CPU esegue le richieste di I/O leggendo e scrivendo i registri di controllo del dispositivo alle locazioni di memoria fisica a cui sono mappati

Memory mapped I/O

In passato, i PC usavano spesso istruzioni di I/O per controllare alcuni dispositivi e l'I/O memory mapped per controllarne altri.

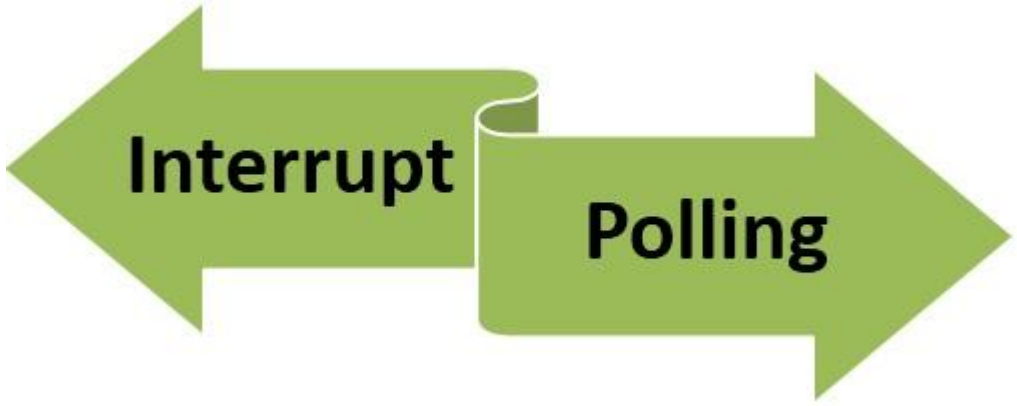
indirizzi per l'I/O (in esadecimale)	dispositivo
000-00F	controllore DMA
020-021	controllore delle interruzioni
040-043	timer
200-20F	controllore dei giochi
2F8-2FF	porta seriale (secondaria)
320-32F	controllore del disco
378-37F	porta parallela
3D0-3DF	controllore della grafica
3F0-3F7	controllore dell'unità a dischetti
3F8-3FF	porta seriale (principale)

Figura 12.2 Indirizzi delle porte dei dispositivi di I/O nei PC (elenco parziale).

Polling vs. interrupt

Interrupt e polling sono le due modalità con cui gli eventi generati dai dispositivi connessi al PC possono essere gestiti dalla CPU

- Nella gestione con **polling**, la CPU tiene traccia delle comunicazioni dei dispositivi di I/O a intervalli regolari
- Nella gestione con **interrupt**, il dispositivo di I/O interrompe la CPU comunicando ad essa che ha bisogno di andare in esecuzione



https://techdifferences.com/difference-between-interrupt-and-polling-in-os.html

Interruzioni

Le **interruzioni** sono usate diffusamente dai sistemi operativi moderni per gestire eventi asincroni e per eseguire procedure in modalità supervisore nel kernel.

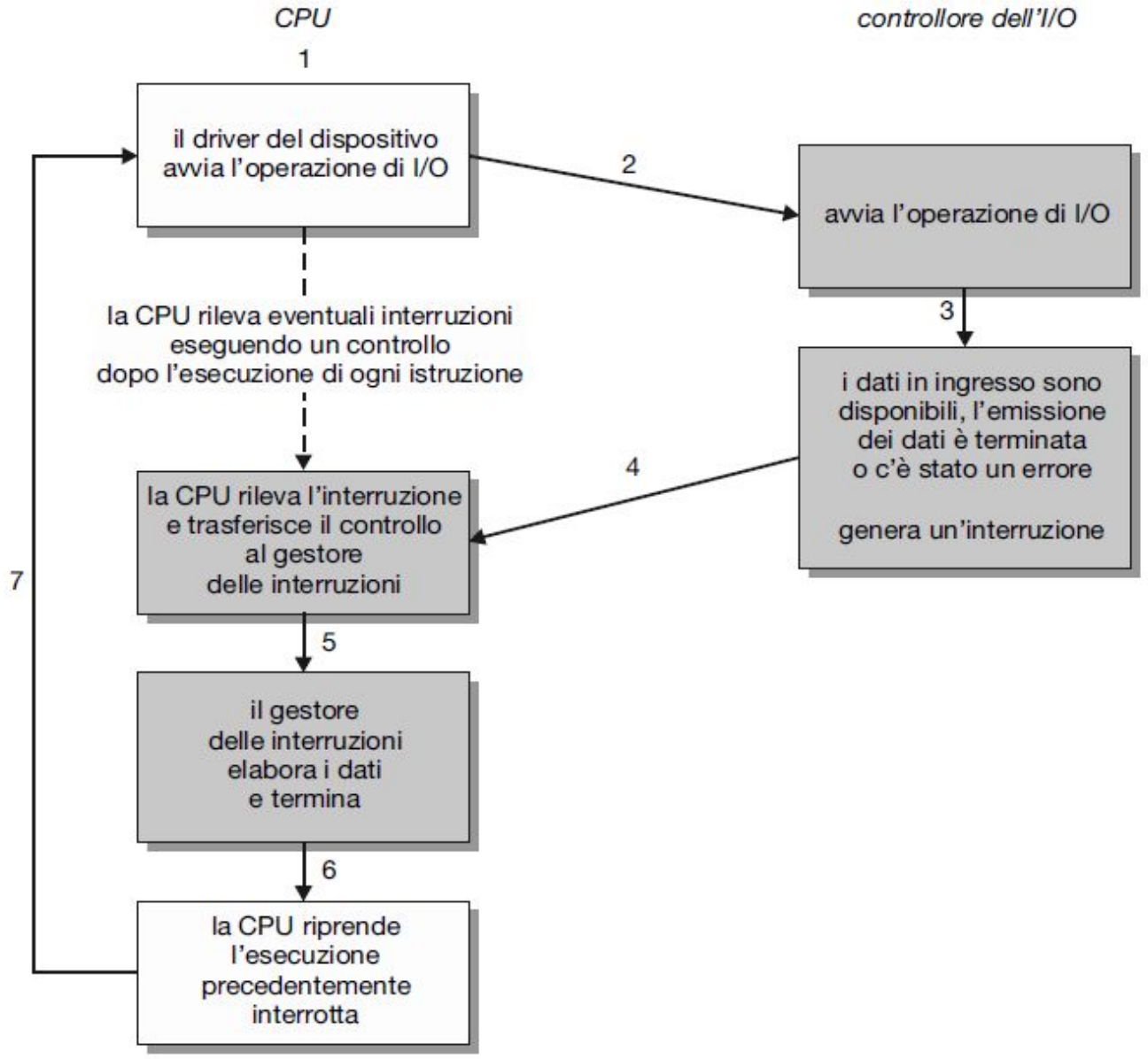


Figura 12.3 Ciclo di I/O basato sulle interruzioni.

Interruzioni

- Per far sì che i compiti più urgenti siano portati a termine per primi, i calcolatori moderni usano un **sistema di priorità delle interruzioni.**

- I controllori dei dispositivi, i guasti hardware e le chiamate di sistema generano interruzioni al fine di innescare l'esecuzione di procedure del kernel.

- Poiché le interruzioni sono usate in modo massiccio per affrontare situazioni in cui il tempo è un fattore critico, è necessario avere un'efficiente gestione delle interruzioni per ottenere buone prestazioni del sistema.

Interruzioni

Anche i moderni sistemi monoutente gestiscono centinaia di interruzioni al secondo e i server ne gestiscono persino centinaia di migliaia al secondo

La schermata mostra l'output del comando latency su macOS, rivelando che in dieci secondi un computer desktop senza particolari carichi di lavoro ha eseguito quasi 23.000 interrupt.

```
Fri Nov 25 13:55:59                                    0:00:10
                      SCHEDULER      INTERRUPTS
-----------------------------------------------
total_samples                13           22998

delays <  10 usecs           12           16243
delays <  20 usecs            1            5312
delays <  30 usecs            0             473
delays <  40 usecs            0             590
delays <  50 usecs            0              61
delays <  60 usecs            0             317
delays <  70 usecs            0               2
delays <  80 usecs            0               0
delays <  90 usecs            0               0
delays < 100 usecs            0               0
total  < 100 usecs           13           22998
```

Figura 12.4 Il comando `latency` di Mac OS X.

Interruzioni mascherabili e non mascherabili

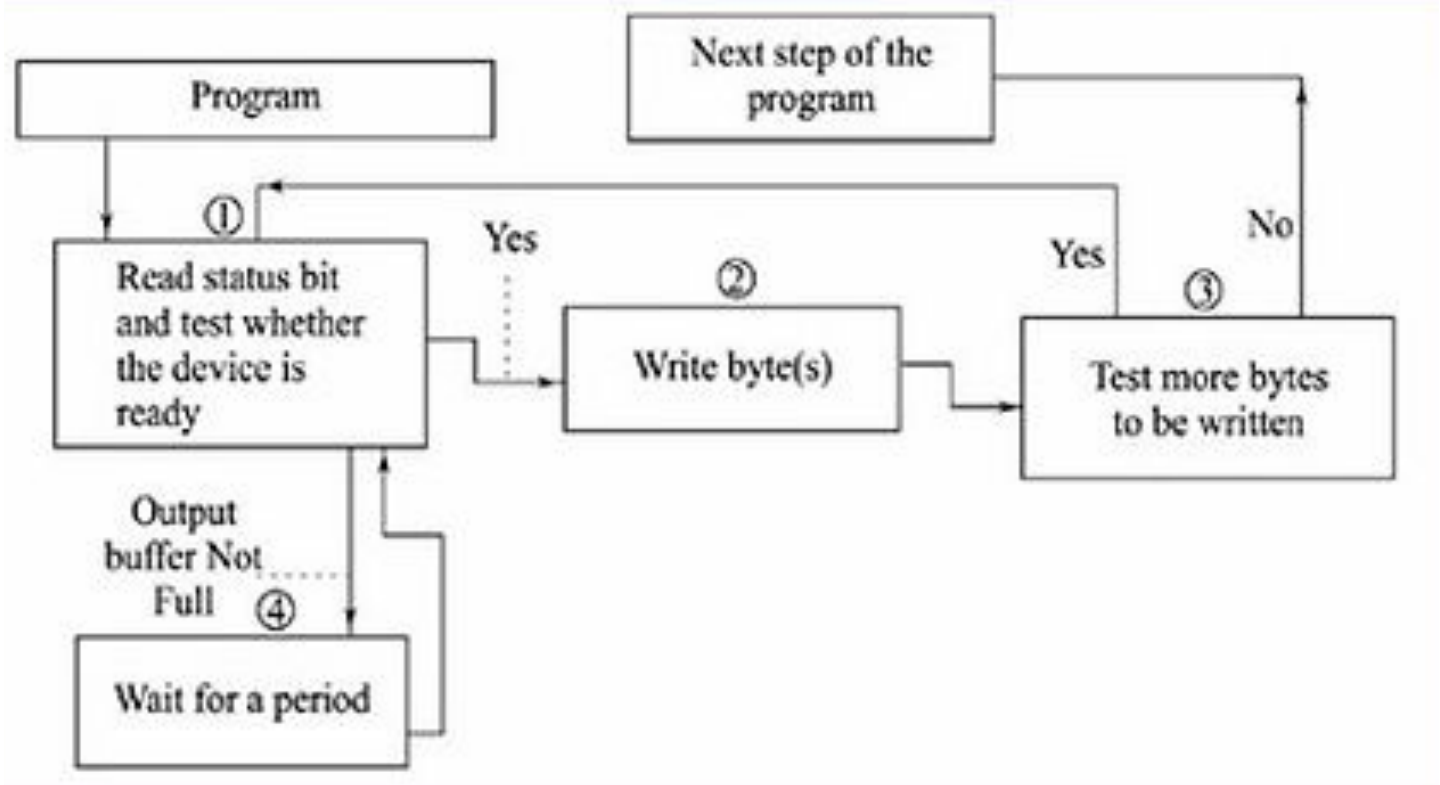
Gli eventi da 0 a 31, non mascherabili, si usano per segnalare varie condizioni d'errore; quelli dal 32 al 255, mascherabili, si usano, per esempio, per le interruzioni generate dai dispositivi → **livelli di priorità delle interruzioni**

indice del vettore	descrizione
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19-31	(Intel reserved, do not use)
32-255	maskable interrupts

Figura 12.5 Vettore delle interruzioni della CPU Intel Pentium.

Programmed I/O (PIO)

Nell'I/O programmato, la CPU scrive i dati nel registro del controllore di dispositivo un byte alla volta



http://inputoutput5822.weebly.com/programmed-io.html

Svantaggi del PIO

- Per il trasferimento di grandi quantità di dati tale tecnica risulta inefficiente poiché può sovraccaricare la CPU

- Per evitare di sovraccaricare la CPU, si assegnano i compiti di trasferimento dati a un processore specializzato, detto controllore dell'accesso diretto in memoria

Direct memory access (DMA)

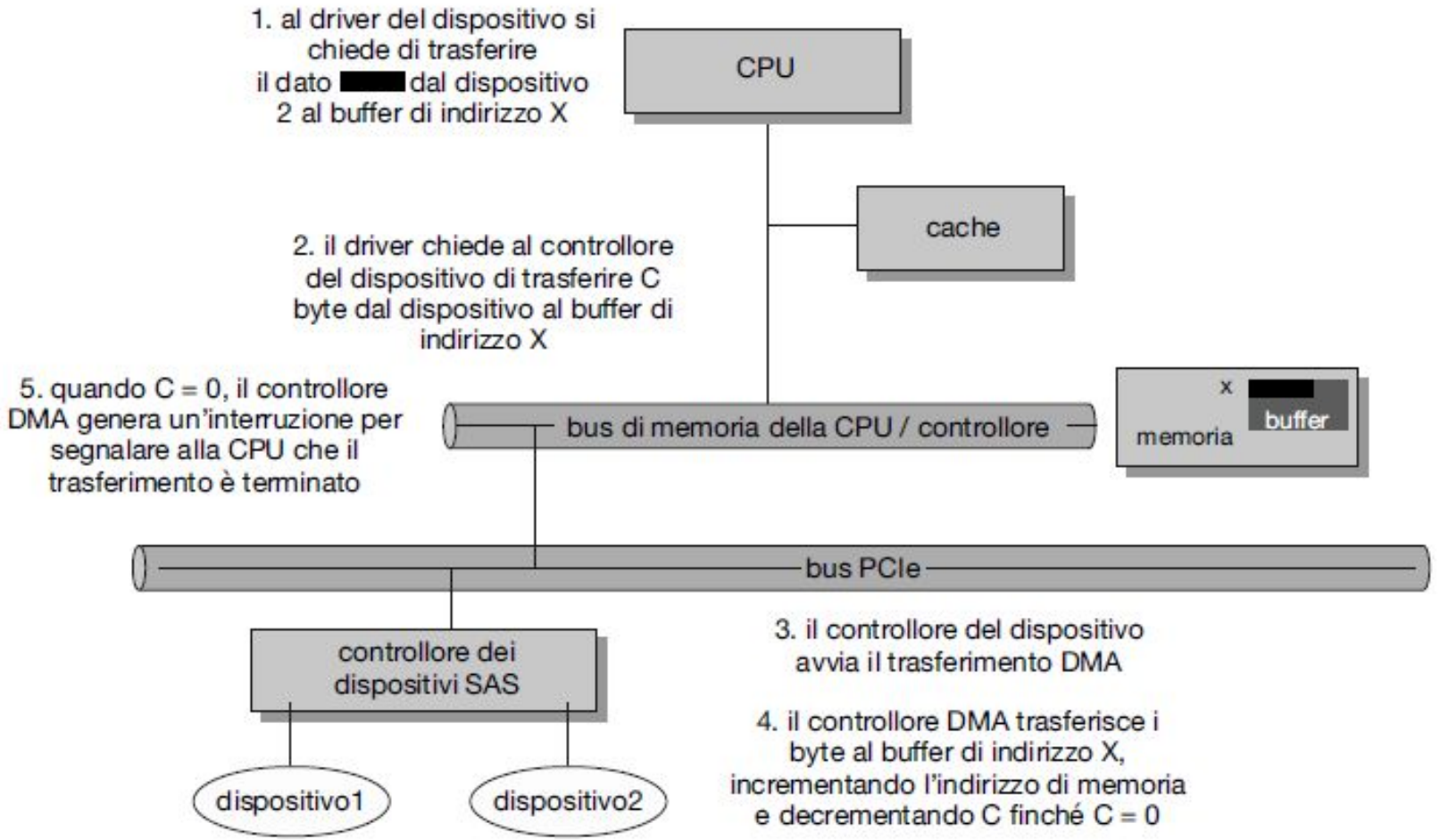


Figura 12.6 Passi di un trasferimento DMA.

Riassumendo

bus

controllore

porta di I/O e suoi registri

procedura di handshaking tra la CPU e il controllore di un dispositivo

esecuzione dell'handshaking per mezzo del polling o delle interruzioni

delega dell'I/O a un controllore DMA nel caso di trasferimenti di grandi quantità di dati.

Interfaccia di I/O delle applicazioni

La figura a lato illustra la divisione in strati software di quelle parti del kernel che riguardano la gestione dell'I/O

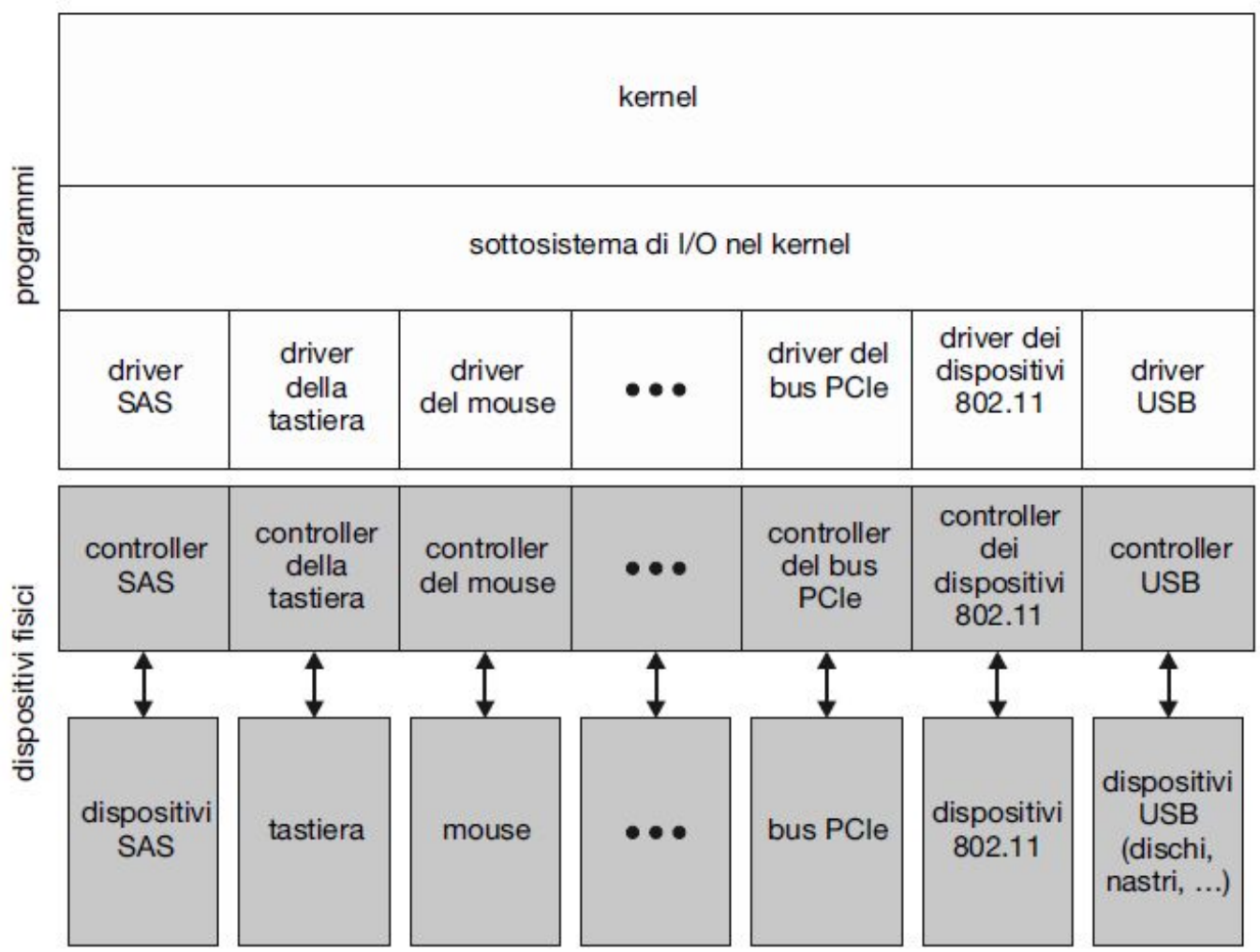


Figura 12.7 Struttura relativa all'I/O nel kernel.

Trasferimento a flusso di caratteri

Trasferimento a flusso di caratteri o a blocchi

Chiamate di sistema:

`get()` per acquisire un carattere

`put()` per inviare un carattere

La tastiera è un esempio di dispositivo al quale si accede tramite una interfaccia a flusso di caratteri.

Altri esempi sono stampanti e schede audio

Dispositivo sincrono

Dispositivi sincroni o asincroni

Un dispositivo sincrono trasferisce dati con un tempo di risposta prevedibile, in maniera coordinata rispetto al resto del sistema

Esempi di comunicazione sincrona:

- videoconferenza
- telefonata

Dispositivo asincrono

Dispositivi sincroni o asincroni

Un dispositivo asincrono ha tempi di risposta irregolari o non prevedibili, non coordinati con altri eventi del computer

Esempi di comunicazione asincrona:

- email
- chat

Dispositivi sequenziali

Dispositivi sequenziali o ad accesso diretto

Un dispositivo sequenziale trasferisce dati secondo un ordine fisso dipendente dal dispositivo

Esempio di dispositivo sequenziale:
La CPU esegue una sequenza di operazioni, una alla volta, in successione.
La CPU è anche un dispositivo di tipo sincrono (usa un clock per gestire la sincronizzazione)

Dispositivi ad accesso diretto

Dispositivi sequenziali o ad accesso diretto

L’utente di un dispositivo ad accesso diretto può richiedere l’accesso a una qualunque delle possibili locazioni di memorizzazione

Esempi di dispositivi ad accesso diretto:

- CD
- HDD
- USB flash drive

Interfaccia di I/O delle applicazioni

aspetto	variazione	esempio
modalità di trasferimento dei dati	a caratteri a blocchi	terminale unità a disco
modalità d'accesso	sequenziale casuale	modem lettore di CD-ROM
prevedibilità dell'I/O	sincrono asincrono	unità a nastro tastiera
condivisione	dedicato condiviso	unità a nastro tastiera
velocità	latenza tempo di ricerca velocità di trasferimento attesa fra le operazioni	
direzione dell'I/O	solo lettura solo scrittura lettura e scrittura	lettore di CD-ROM controllore della grafica unità a disco

Figura 12.8 Caratteristiche dei dispositivi per l'I/O.

I/O sincrono/asincrono

Una possibile alternativa alle chiamate di sistema non bloccanti è costituita dalle chiamate di sistema asincrone. Esse restituiscono immediatamente il controllo al chiamante, senza attendere che l'I/O sia stato completato.

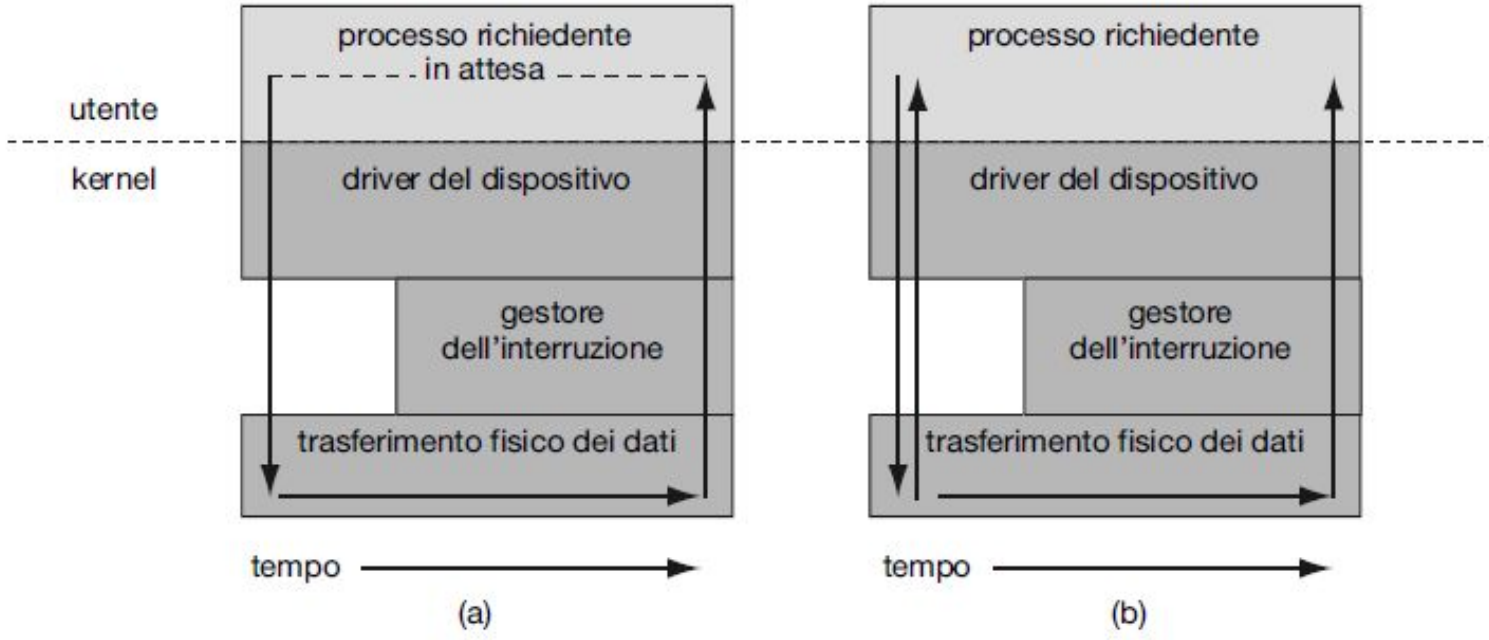


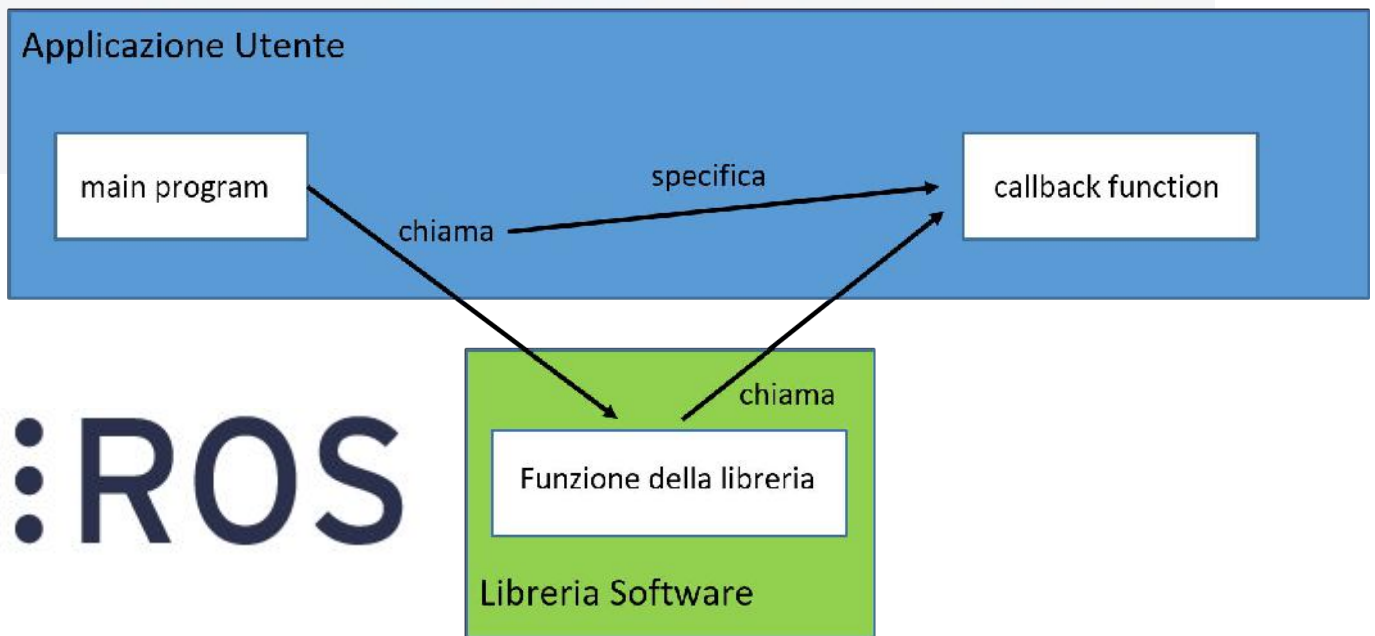
Figura 12.9 Due metodi per l'I/O; (a) sincrono e (b) asincrono.

Callback

Nelle chiamate di sistema asincrone l'applicazione continua ad essere eseguita e il completamento dell'I/O è successivamente comunicato all'applicazione:

- Per mezzo dell'impostazione del valore di una variabile nello spazio di indirizzi dell'applicazione
- Tramite un interrupt software
- Tramite una callback eseguita fuori del normale flusso lineare di elaborazione dell'applicazione

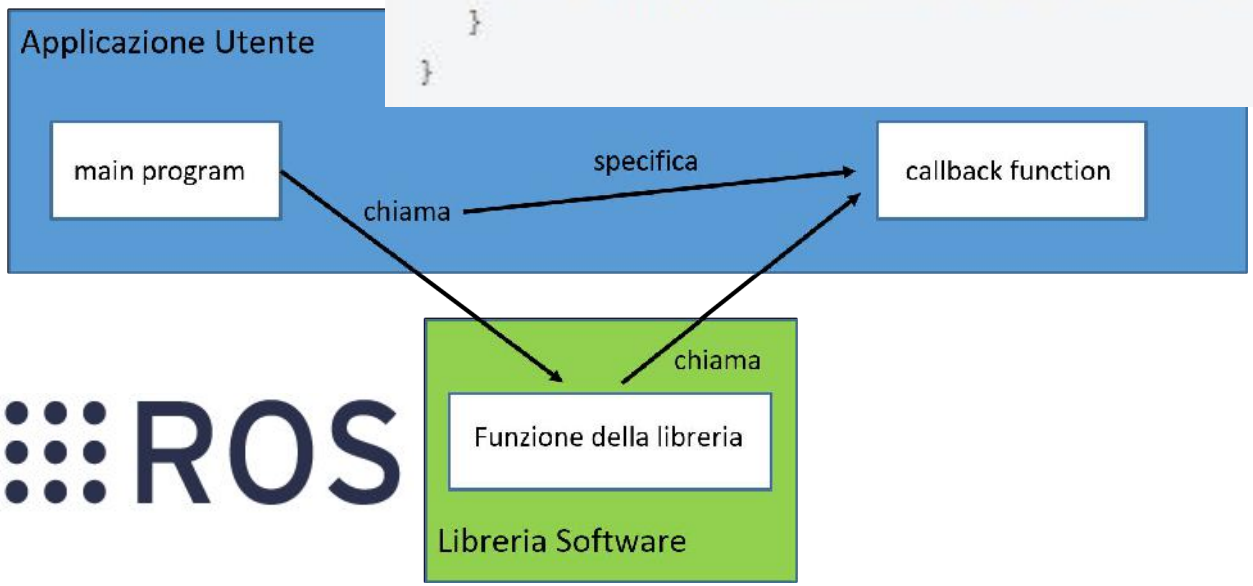
```
int main(int argc, char **argv)
{
  ros::init(argc, argv, "image_listener");
  ros::NodeHandle nh;
  cv::namedWindow("view");
  cv::startWindowThread();
  image_transport::ImageTransport it(nh);
  image_transport::Subscriber sub = it.subscribe("camera/image", 1, imageCallback);
  ros::spin();
  cv::destroyWindow("view");
}
```



ROS

http://wiki.ros.org/image_transport/Tutorials/SubscribingToImages

```
void imageCallback(const sensor_msgs::ImageConstPtr& msg)
{
  try
  {
    cv::imshow("view", cv_bridge::toCvShare(msg, "bgr8")->image);
    cv::waitKey(30);
  }
  catch (cv_bridge::Exception& e)
  {
    ROS_ERROR("Could not convert from '%s' to 'bgr8'.", msg->encoding.c_str());
  }
}
```



ROS

http://wiki.ros.org/image_transport/Tutorials/SubscribingToImages

```
it.subscribe("camera/image", 1, imageCallback);
```

```
ros::spin();
```

Sottosistema di I/O del kernel

Il kernel fornisce molti servizi riguardanti l'I/O; i seguenti servizi sono offerti dal sottosistema di I/O del kernel e sono realizzati a partire dai dispositivi e dai relativi driver.

- scheduling
- gestione del buffer
- gestione delle cache
- gestione delle code di spooling
- riservazione dei dispositivi
- gestione degli errori → protezione dell'I/O

Tabella dello stato dei dispositivi

Gli elementi della **tabella dello stato dei dispositivi** – uno per ogni dispositivo di I/O – indicano il *tipo*, l'*indirizzo* e lo *stato del dispositivo*: non funzionante, inattivo o occupato.

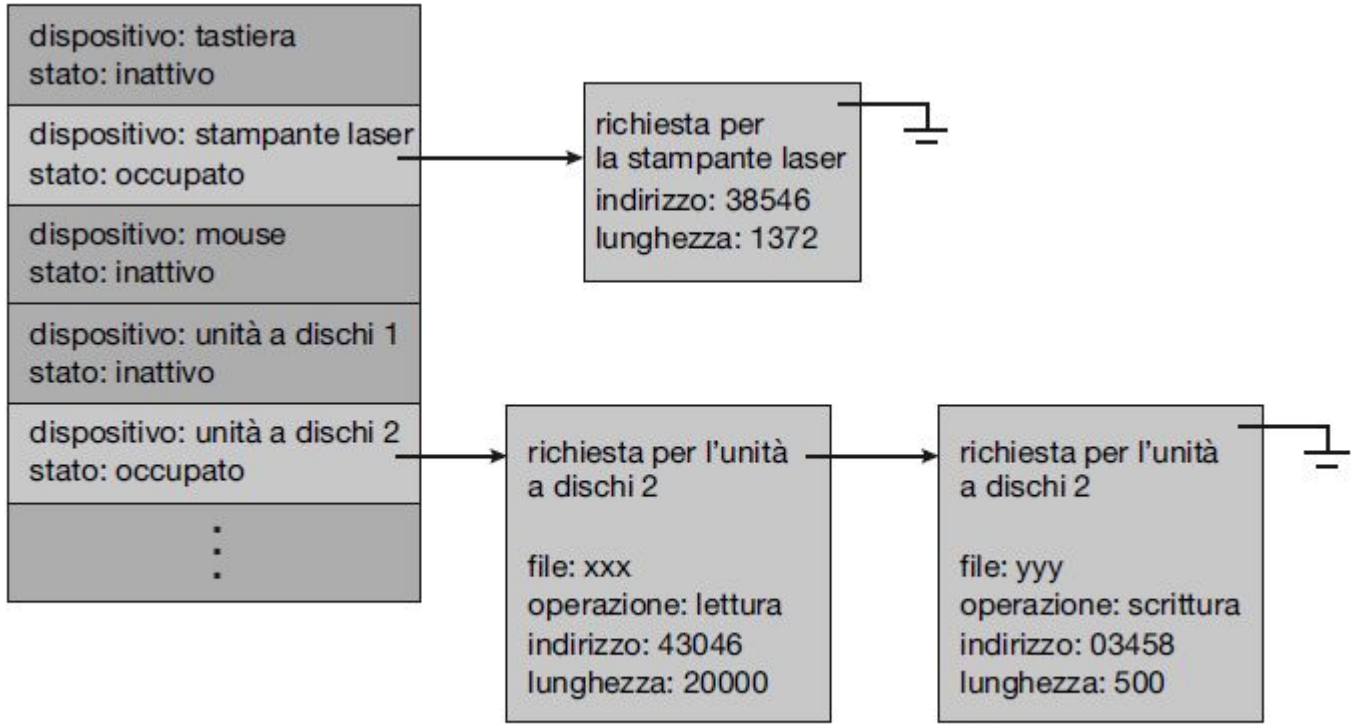


Figura 12.10 Tabella dello stato dei dispositivi.

Gestione dei buffer

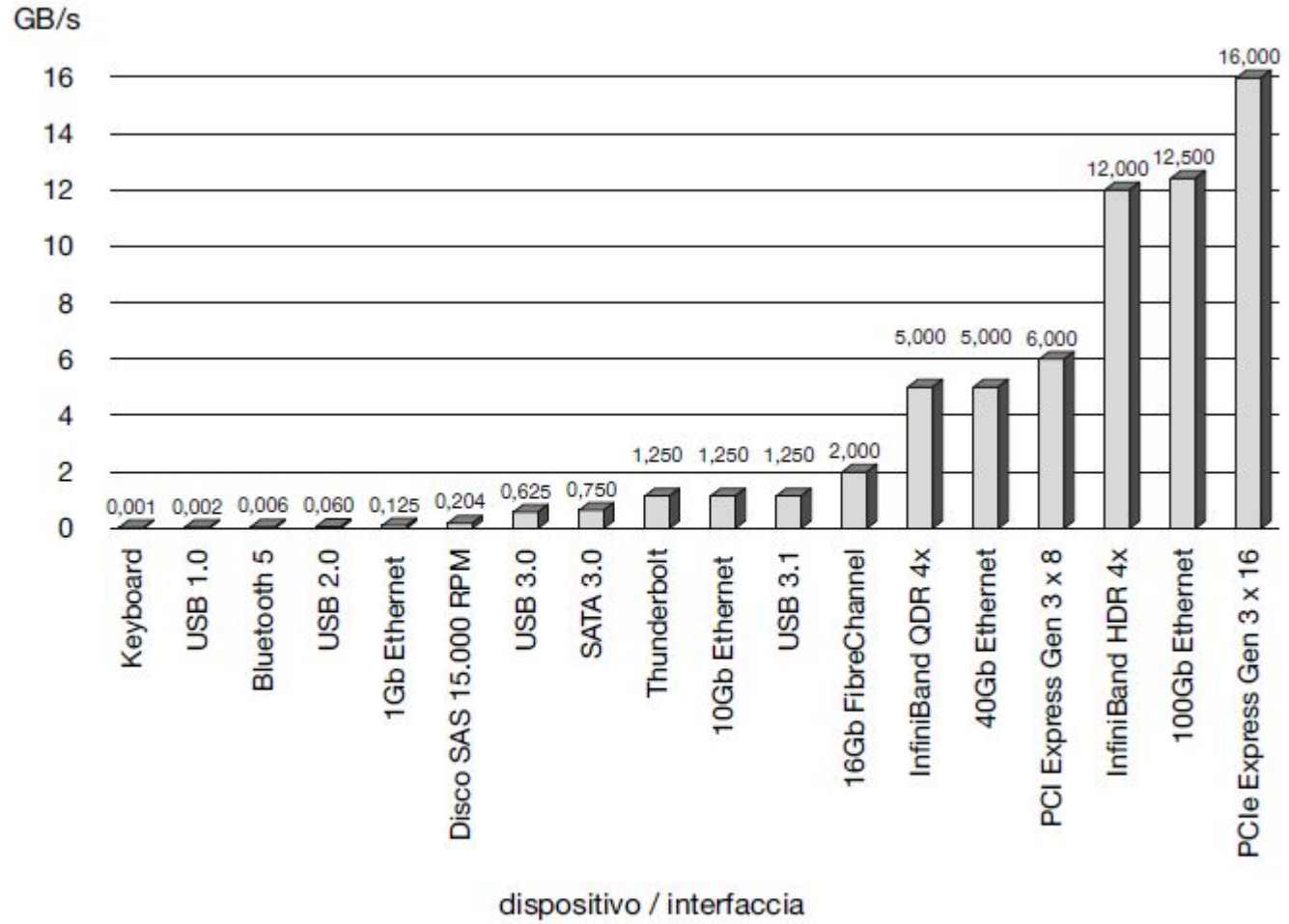


Figura 12.11 Dispositivi di I/O utilizzati in PC e data center e velocità dell'interfaccia.

Cache

La differenza tra un buffer e una cache consiste nel fatto che il primo può contenere dati di cui non vi è altra copia, mentre una cache, per definizione, mantiene su un mezzo più efficiente una copia di informazioni memorizzate altrove.

Protezione dell'I/O

Un programma utente, per eseguire l'I/O, invoca una chiamata di sistema per chiedere al sistema operativo di svolgere una data operazione nel suo interesse.

Il sistema, passando alla modalità privilegiata, verifica che la richiesta sia valida e, in tal caso, esegue l'operazione; esso trasferisce quindi il controllo all'utente.

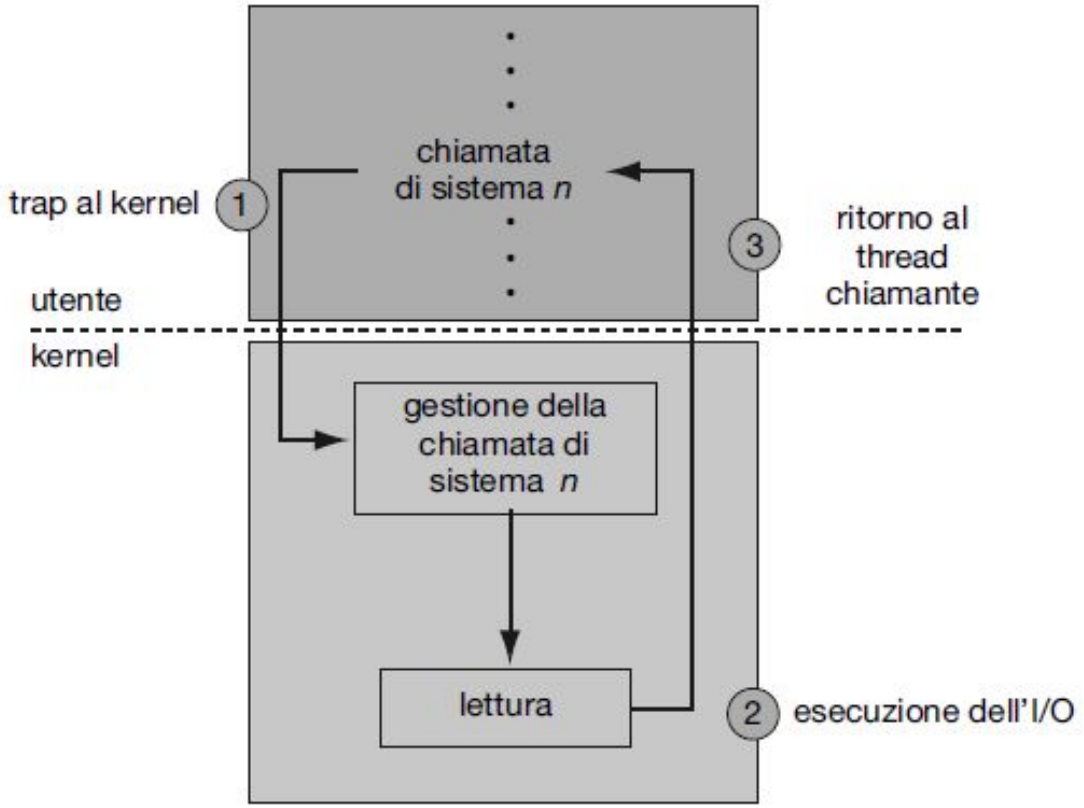


Figura 12.12 Uso delle chiamate di sistema per eseguire I/O.

Strutture dati del kernel

Servono a mantenere informazioni sullo stato dei componenti di I/O

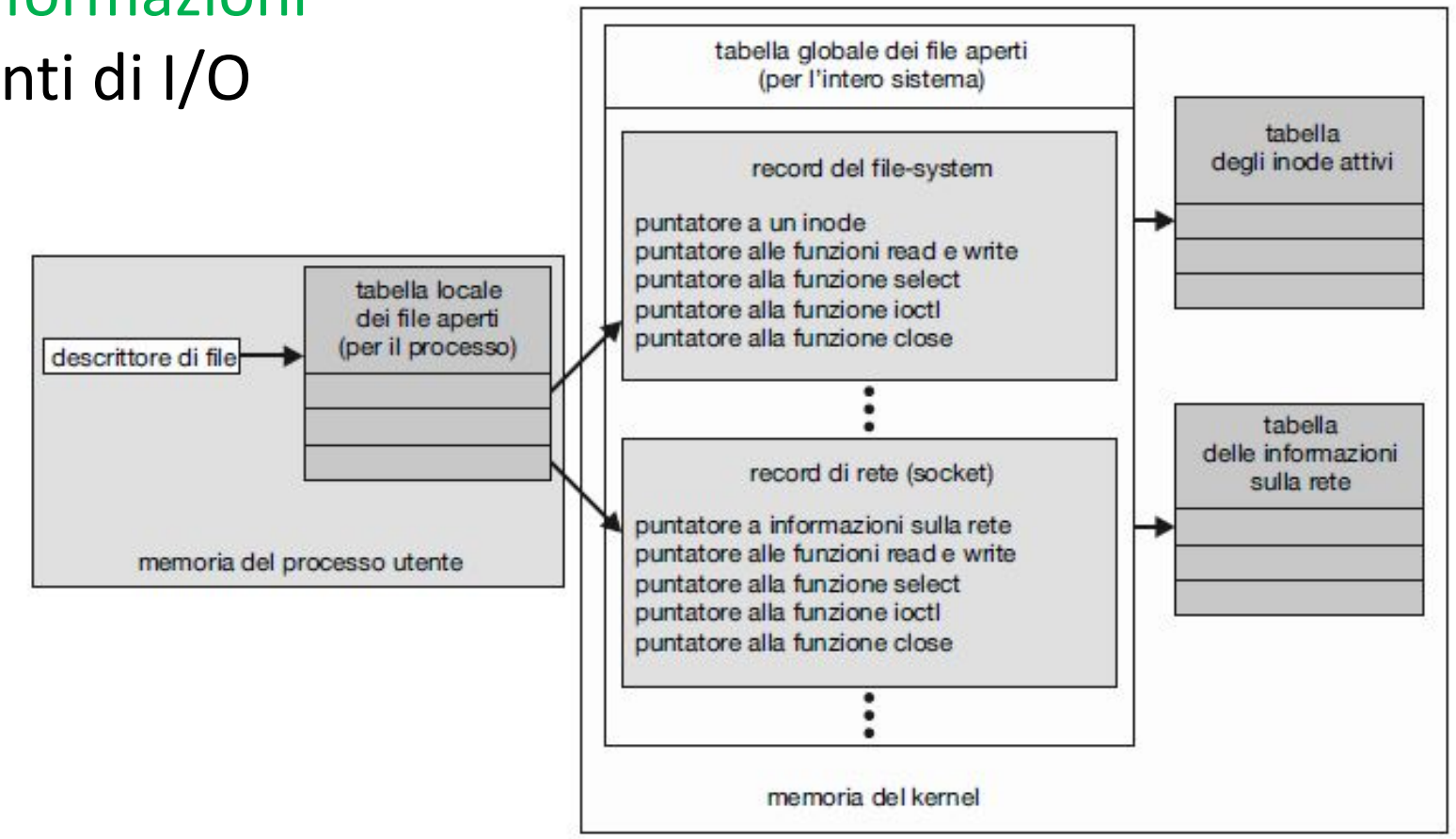


Figura 12.13 Struttura dell'I/O nel kernel di UNIX.

Riassumendo

Il sistema per l'I/O coordina un'ampia raccolta di servizi disponibili per le applicazioni e per altre parti del kernel:

- gestione dello spazio dei nomi per file e dispositivi
- controllo dell'accesso ai file e ai dispositivi
- controllo delle operazioni
- allocazione dello spazio per il file system
- allocazione dei dispositivi
- gestione dei buffer, delle cache e delle code di spooling
- scheduling dell'I/O
- controllo dello stato dei dispositivi, gestione degli errori e procedure di ripristino
- configurazione e inizializzazione dei driver dei dispositivi
- gestione energetica dei dispositivi dell'I/O

Esecuzione di una richiesta di I/O

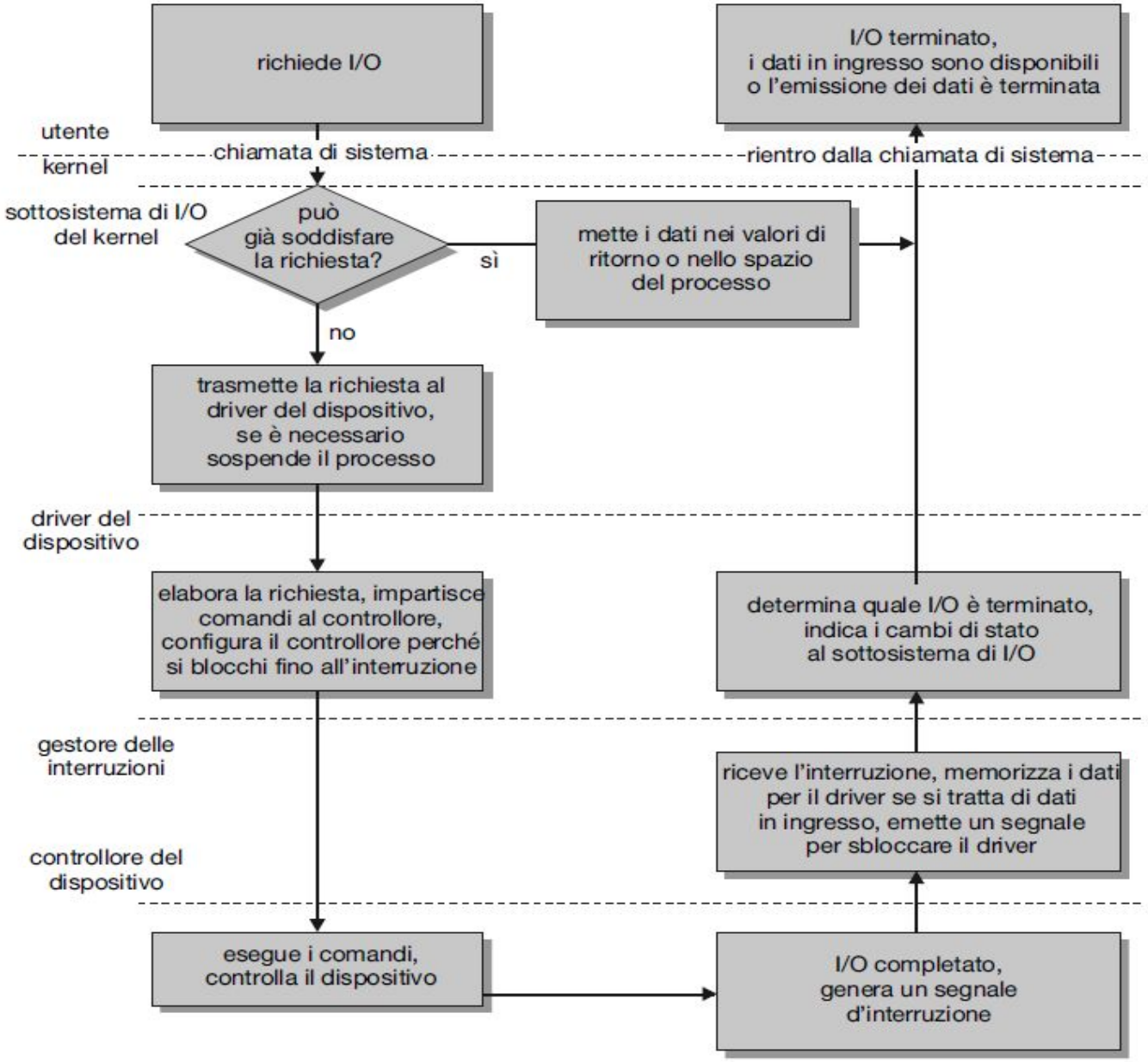


Figura 12.14 Schema d'esecuzione di una richiesta di I/O.

STREAMS

STREAMS è una metodologia che permette di sviluppare in modo modulare e incrementale i driver e i protocolli di rete.

Utilizzando gli *stream*, i driver possono essere organizzati in una catena, attraverso cui passano i dati in maniera sequenziale e bidirezionale per l'elaborazione

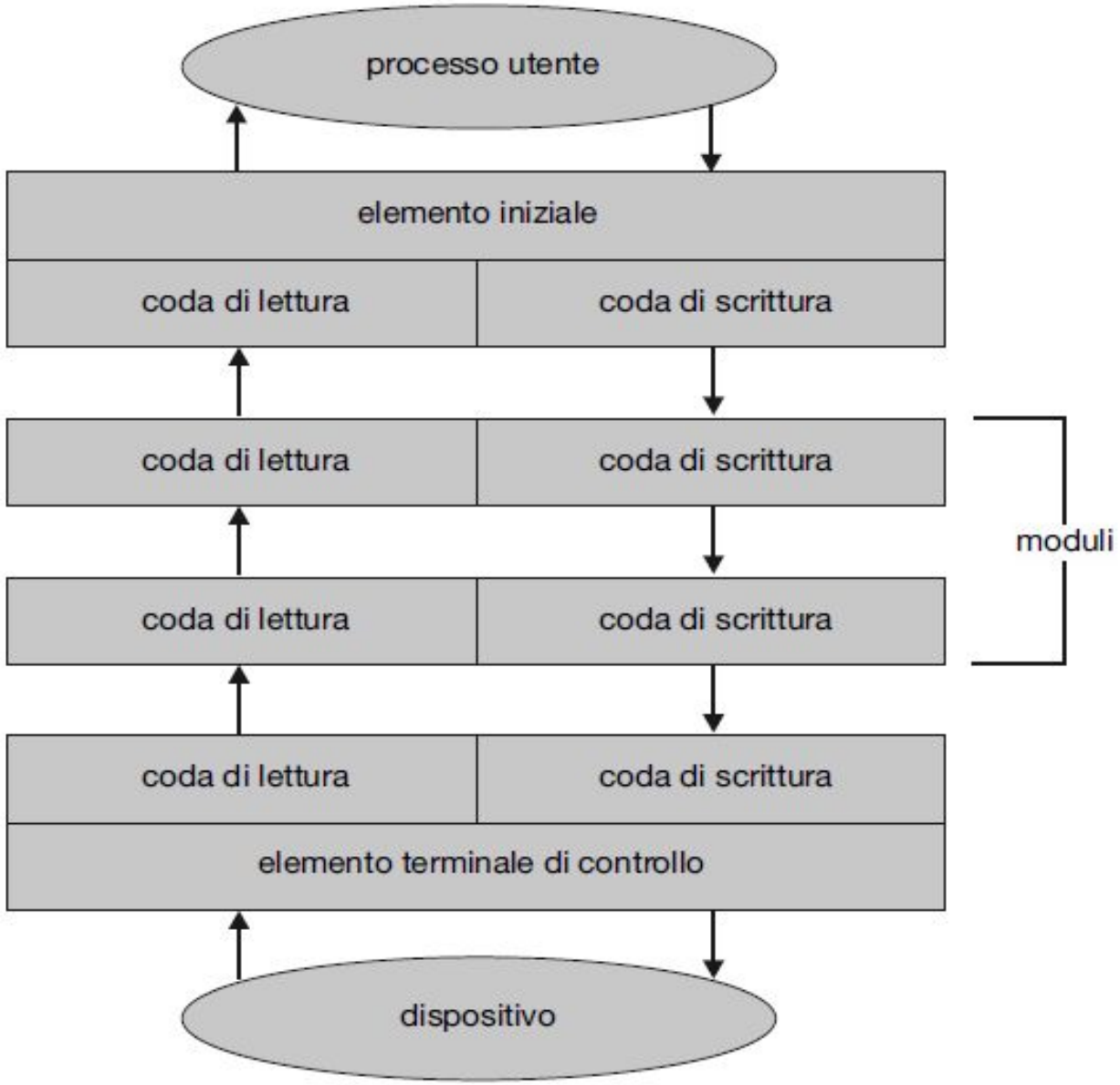


Figura 12.15 Struttura di STREAMS.

`ioctl`

La system call `ioctl` permette di interagire con il driver di un dispositivo generico, per esempio una webcam.

Tramite la `ioctl` sarà possibile ricavare e settare i parametri di tale dispositivo, per esempio ricavare la risoluzione della webcam o settarne la tipologia di acquisizione dati.

Per configurare dispositivi seriali a flusso di caratteri, per esempio un terminale, è possibile usare le API incluse nella interfaccia `termios`. Tramite questa, avremo accesso a tutte le informazioni relative al dispositivo, per esempio baudrate, echo, etc...

Prestazioni

A causa dei molti strati di software presenti fra un dispositivo fisico e l'applicazione, le chiamate di sistema per l'I/O sono onerose in termini di utilizzazione della CPU.

Anche il traffico di una rete può portare a un alto numero di cambi di contesto; si consideri, per esempio, il login remoto da un calcolatore a un altro.

Prestazioni

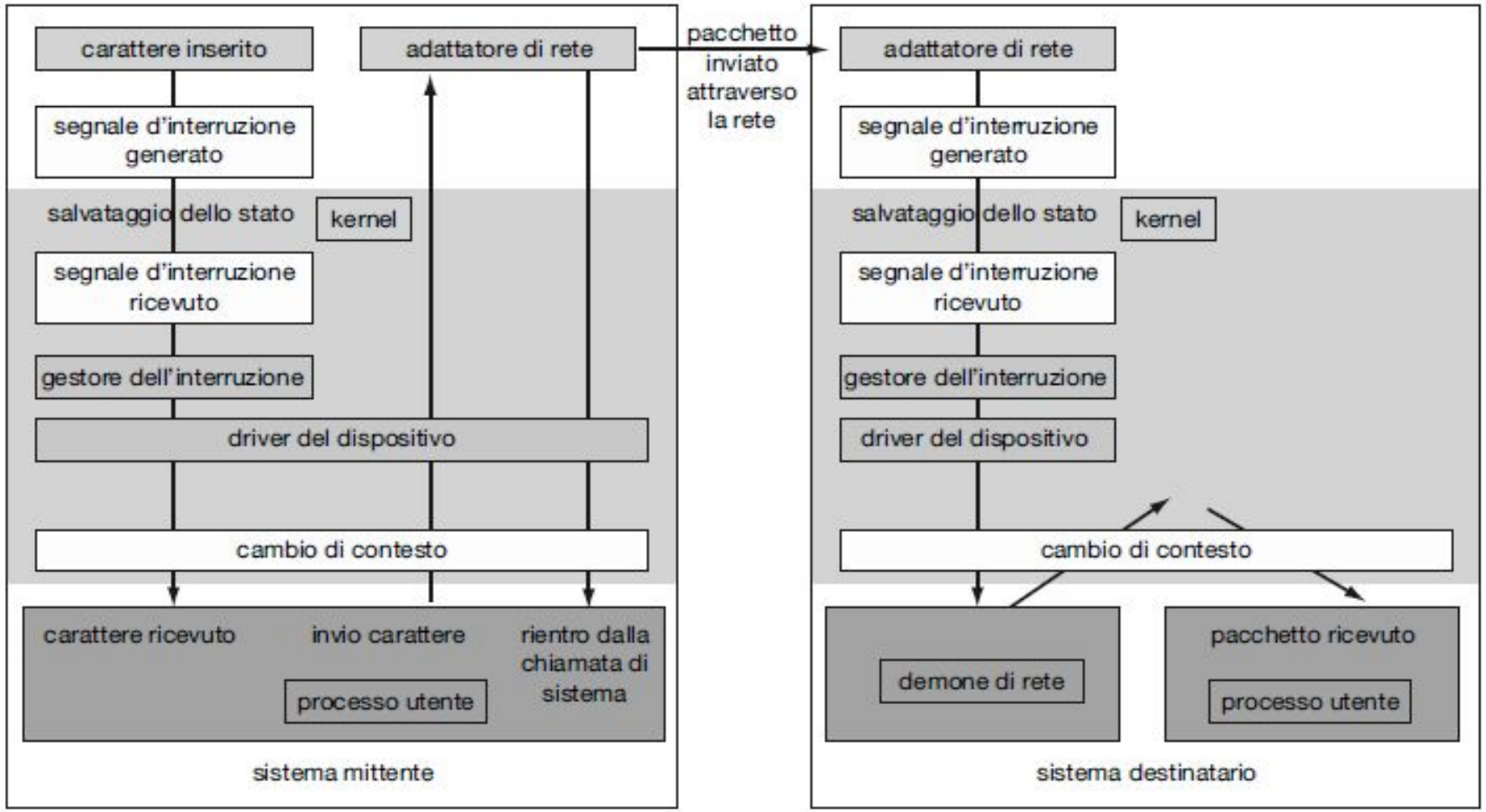


Figura 12.16 Comunicazione tra calcolatori.

Implementazione dei servizi di I/O

Ci si può chiedere se i servizi di I/O si debbano implementare nei dispositivi hardware, nei loro driver, o nelle applicazioni. Talvolta si può osservare (Figura 12.17) la seguente successione.

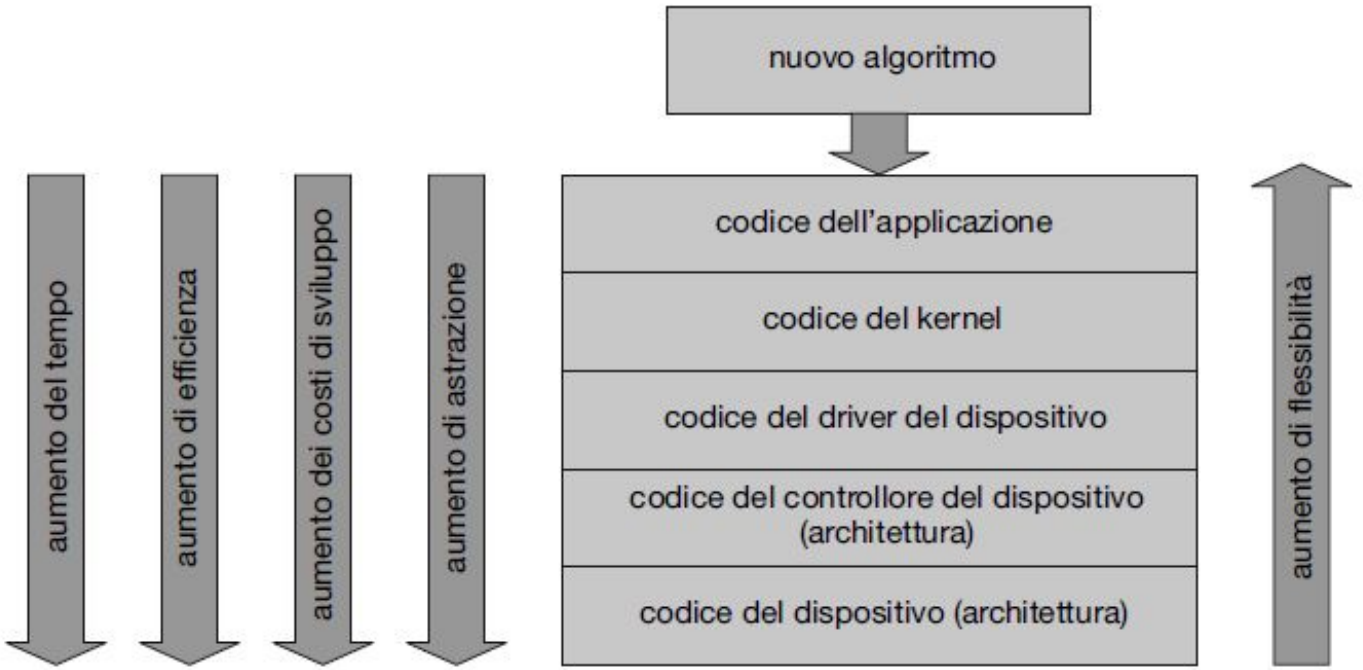


Figura 12.17 Successione delle funzionalità dei servizi di I/O.

Capacità e latenza

La Figura 12.18 mostra CPU e dispositivi di memoria in un grafico dove le due dimensioni rappresentano la capacità e la latenza delle operazioni di I/O. Inoltre, la figura mostra una rappresentazione della latenza di rete, utile per rivelare il tributo aggiuntivo imposto dal networking in termini di prestazioni.

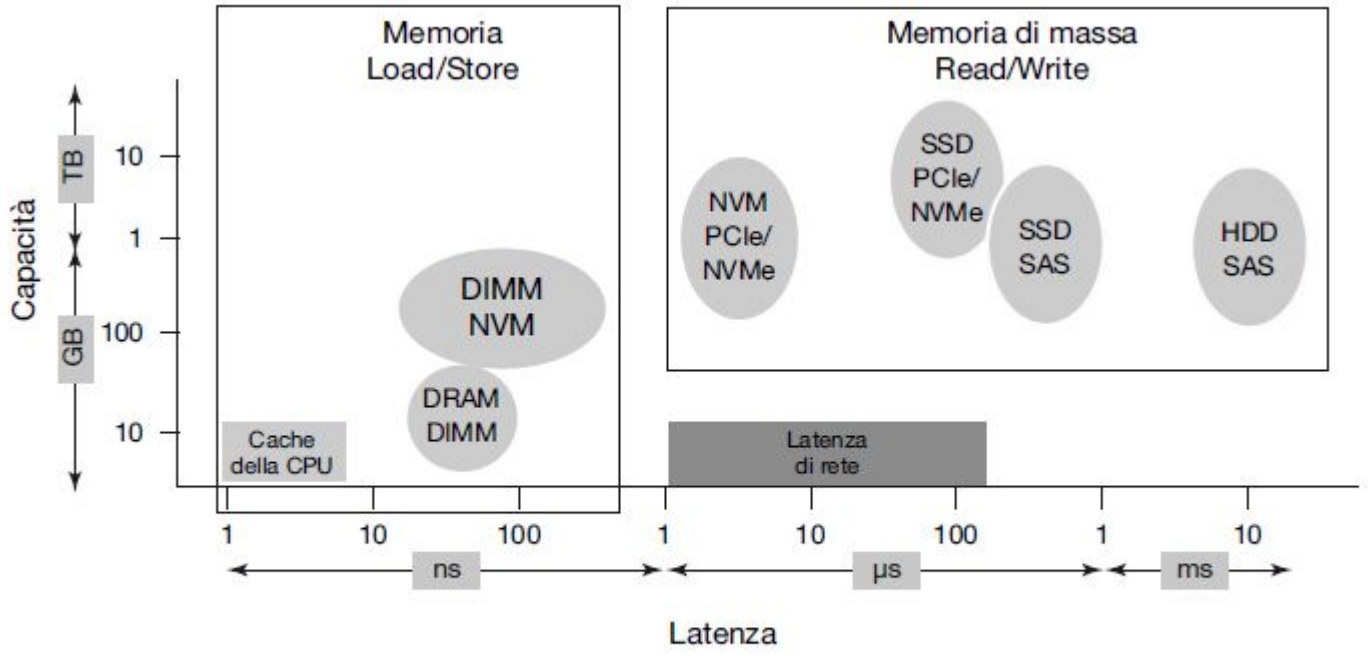


Figura 12.18 Prestazioni di I/O dei dispositivi di memorizzazione (e latenza di rete).