

Capitolo 2: Strutture dei sistemi di calcolo

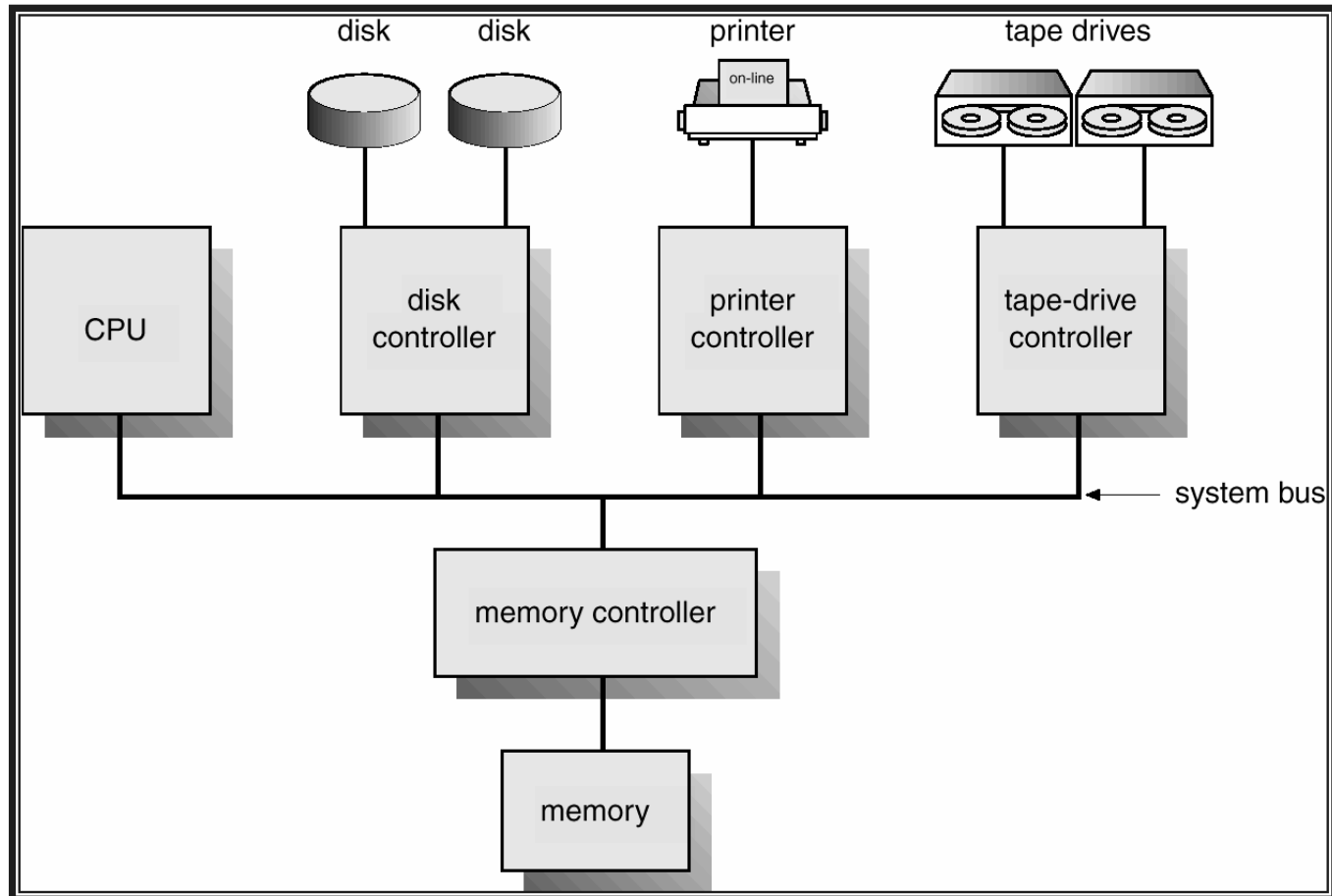
- Funzionamento dei sistemi di calcolo
- Struttura dell'I/O
- Struttura della memoria
- Gerarchie delle memorie
- Architetture di protezione
- Strutture delle reti di calcolatori

Introduzione

- Lo studio dei sistemi operativi presuppone una conoscenza generale della struttura di un calcolatore.
- Scopo della presente lezione è una trattazione introduttiva dei diversi elementi di questa struttura.
- Un sistema operativo deve assicurare il corretto funzionamento di un calcolatore.
- Per evitare che i programmi utenti interferiscano con le operazioni del sistema operativo, l'architettura deve fornire meccanismi appropriati.

Struttura di un sistema di calcolo

- Ogni calcolatore di uso generale è composto da una CPU e da un certo numero di dispositivi, i cui controllori sono connessi attraverso un bus comune alla memoria del sistema.



Funzionamento di un sistema di calcolo

- L'avviamento del sistema richiede un **programma di avviamento** (*bootstrap program*), conservato in una memoria a sola lettura.
- Esso inizializza i diversi componenti del sistema, carica in memoria centrale il nucleo del sistema operativo e ne avvia l'esecuzione.
- Il sistema operativo avvia il primo processo d'elaborazione e resta in attesa che si verifichi un evento.
- Tale evento può essere causato o da un dispositivo fisico o da un programma.
 - ◆ un'**interruzione** (*interrupt*) è un segnale che un controllore di un dispositivo (o un elemento hardware) invia alla CPU.
 - ◆ un'**eccezione** (*exception, trap*) può essere causata da un programma in esecuzione a seguito di un errore o una richiesta.

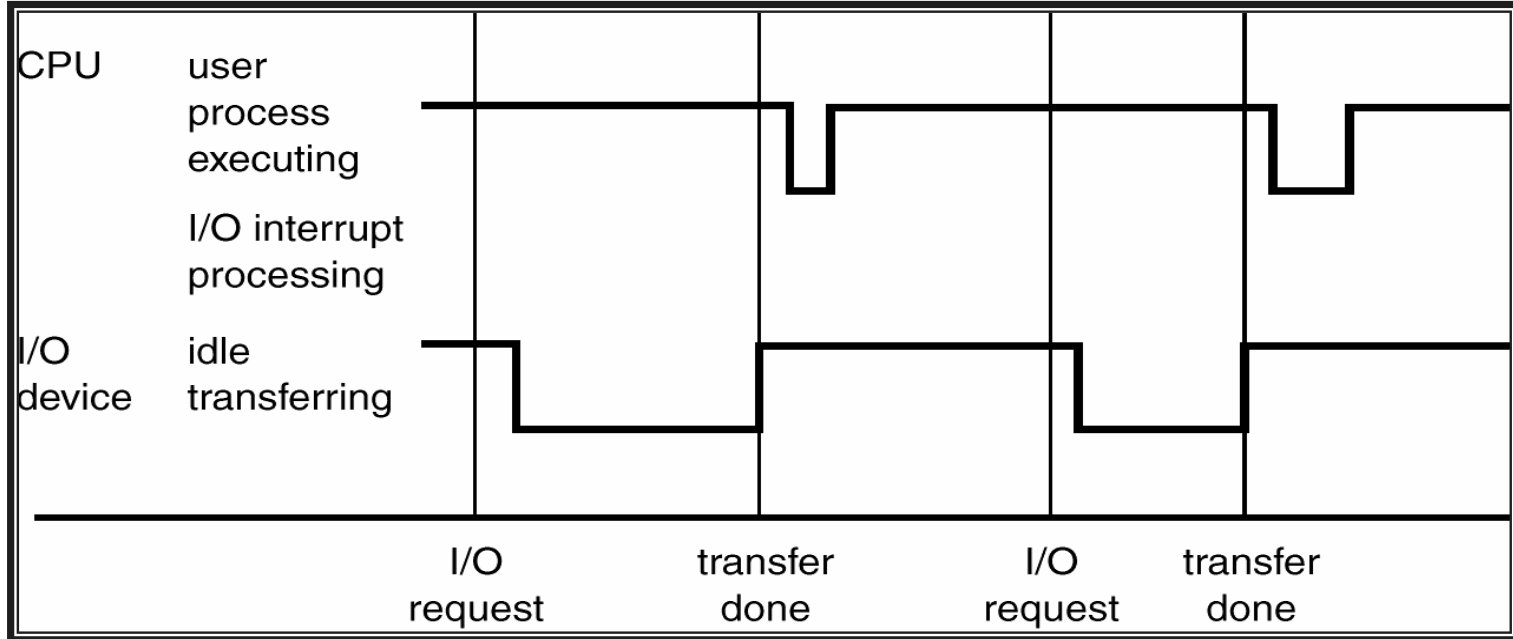
Funzionamento di un sistema di calcolo

- I dispositivi di I/O e la CPU possono eseguire in concorrenza.
- Ciascun controllore di dispositivo controlla un particolare dispositivo.
- Ciascun controller ha un buffer locale.
- La CPU muove i dati da/verso la memoria verso/da i buffer locali.
- I/O è da il dispositivo al buffer locale del controllore.
- Il controllore di un dispositivo informa la CPU che ha finito le sue operazioni causando un'**interruzione**.

Gestione delle interruzioni

- Un'interruzione causa il trasferimento dell'esecuzione ad una procedura di servizio, attraverso un *vettore delle interruzioni*, che contiene gli indirizzi delle procedure di servizio.
- L'architettura di gestione delle interruzioni deve anche salvare l'indirizzo dell'istruzione interrotta e il valore dei registri.
- Le interruzioni in arrivo sono *disabilitate* quando un'altra interruzione è in fase di elaborazione per evitare la *perdita delle interruzioni*.
- Un'eccezione è un'interruzione generata dal software causata o da un errore o da una richiesta di un utente.
- Un sistema operativo è **guidato dalle interruzioni**.

Diagramma temporale delle interruzioni per singolo processo



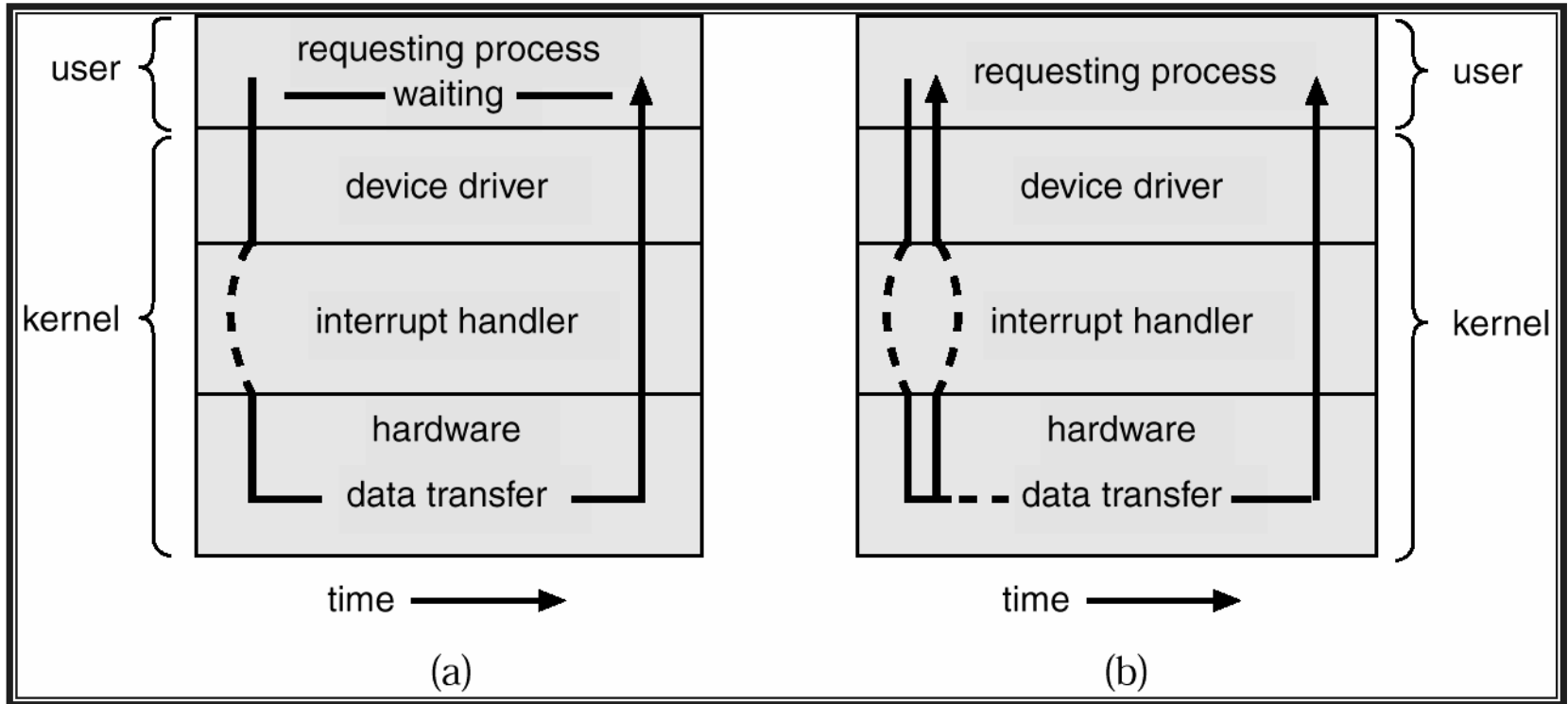
Struttura di I/O

- Per iniziare un'operazione di I/O la CPU inizializza i registri del controllore del dispositivo.
- Il controllore esamina il contenuto dei registri e determina l'operazione da effettuare.
- Ad esempio, se la CPU richiede una lettura, il controllore esegue il trasferimento dei dati dal dispositivo alla propria memoria di transito e quindi informa la CPU del completamento con un'interruzione.
- Se un processo utente effettua una richiesta di I/O può accadere:
 - ◆ il processo riacquista il controllo solo al completamento dell'operazione (**I/O sincrono**)
 - ◆ il controllo è immediatamente restituito al processo utente (**I/O asincrono**)

Metodi di I/O sincrono ed asincrono

Sincrono

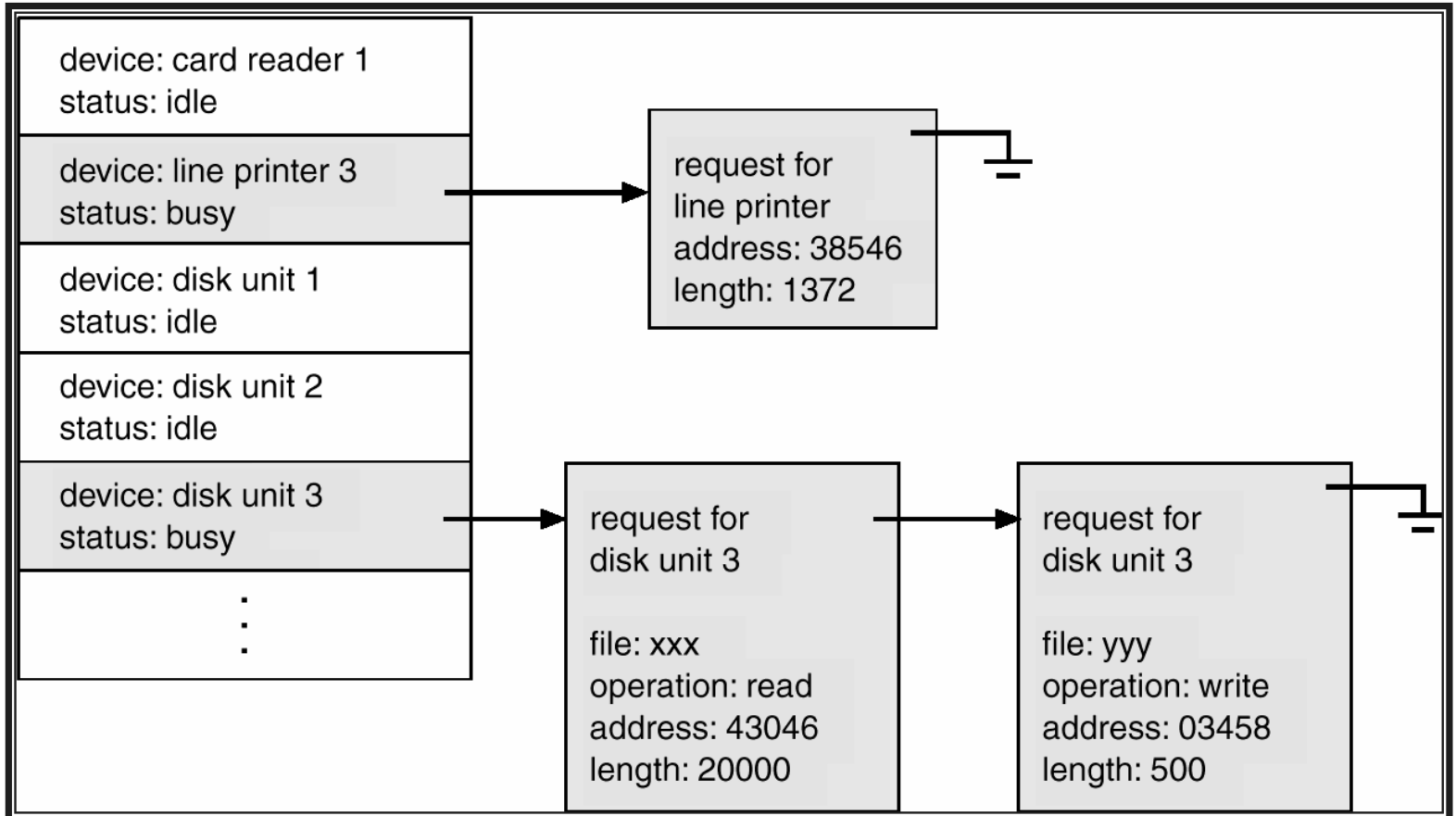
Asincrono



Struttura di I/O

- **Metodo sincrono:** dopo che un'operazione di I/O è iniziata, il controllo ritorna al programma utente solo dopo che l'I/O è completato.
 - ◆ L'istruzione `wait` lascia la CPU in attesa fino alla successiva interruzione
 - ◆ Ciclo d'attesa
 - Attesa: `JUMP Attesa`
 - ◆ Una sola richiesta alla volta può essere pendente
- **Metodo asincrono:** dopo che un'operazione di I/O è iniziata, il controllo ritorna al programma utente senza attendere il completamento
 - ◆ *Chiamata di sistema:* richiesta al sistema operativo di poter attendere il completamento dell'I/O.
 - ◆ *Tabella di stato dei dispositivi:* una riga per ciascun dispositivo di I/O (tipo, indirizzo, stato).
 - ◆ Ad ogni dispositivo è associata una coda d'attesa.

Tabella di stato dei dispositivi



Accesso diretto alla memoria

- Esempio: dispositivo immissione di dati collegato alla porta seriale
 - ◆ Ricevuto il carattere dal dispositivo, il controller della porta seriale invia un segnale d'interruzione alla CPU, che riceve tale interruzione prima di eseguire una nuova istruzione.
 - ◆ La CPU salva l'indirizzo dell'istruzione interrotta e trasferisce il controllo alla procedura di servizio del dispositivo.
 - ◆ La procedura di servizio verifica che non ci siano stati errori, preleva il carattere dal controller e lo memorizza in una locazione di memoria riservata al dispositivo.
 - ◆ La procedura di servizio predispone che nuovi caratteri in arrivo vengano memorizzati di seguito a quelli ricevuti
 - ◆ La procedura di servizio informa il sistema operativo del successo dell'operazione.
 - ◆ La procedura ripristina il valore dei registri e restituisce il controllo all'istruzione interrotta.

Accesso diretto alla memoria

- Se il dispositivo di immissione dati è lento rispetto alla memoria, le interruzioni avvengono con una frequenza tale che interruzioni più urgenti possono essere gestite prima.
- Per dispositivi con velocità paragonabili a quella della memoria, le interruzioni lasciano poco tempo all'esecuzione dei processi.
- Per risolvere questo problema è necessario che il controller stesso trasferisca blocchi di dati dalla memoria di transito alla memoria centrale, o viceversa, senza intervento della CPU.
- In questo modo viene generata una sola interruzione per blocco di dati, piuttosto che una per byte.
- E' il driver di dispositivo a impostare i registri del controllore di **accesso diretto alla memoria** (DMA), affinché impieghi gli appropriati indirizzi di partenza e destinazione e la lunghezza del blocco da trasferire.

Struttura della memoria

- **Memoria centrale:** è la sola area di memoria a cui la CPU può accedere direttamente.
 - ◆ L'interazione avviene mediante una sequenza di istruzioni **load** e **store**
- **Memoria secondaria:** estensione della memoria centrale che fornisce un'estesa capacità di memorizzazione non volatile.
- Altri tipi di memorie sono disponibili (cache, CD-ROM, nastri magnetici,..).

Memoria centrale

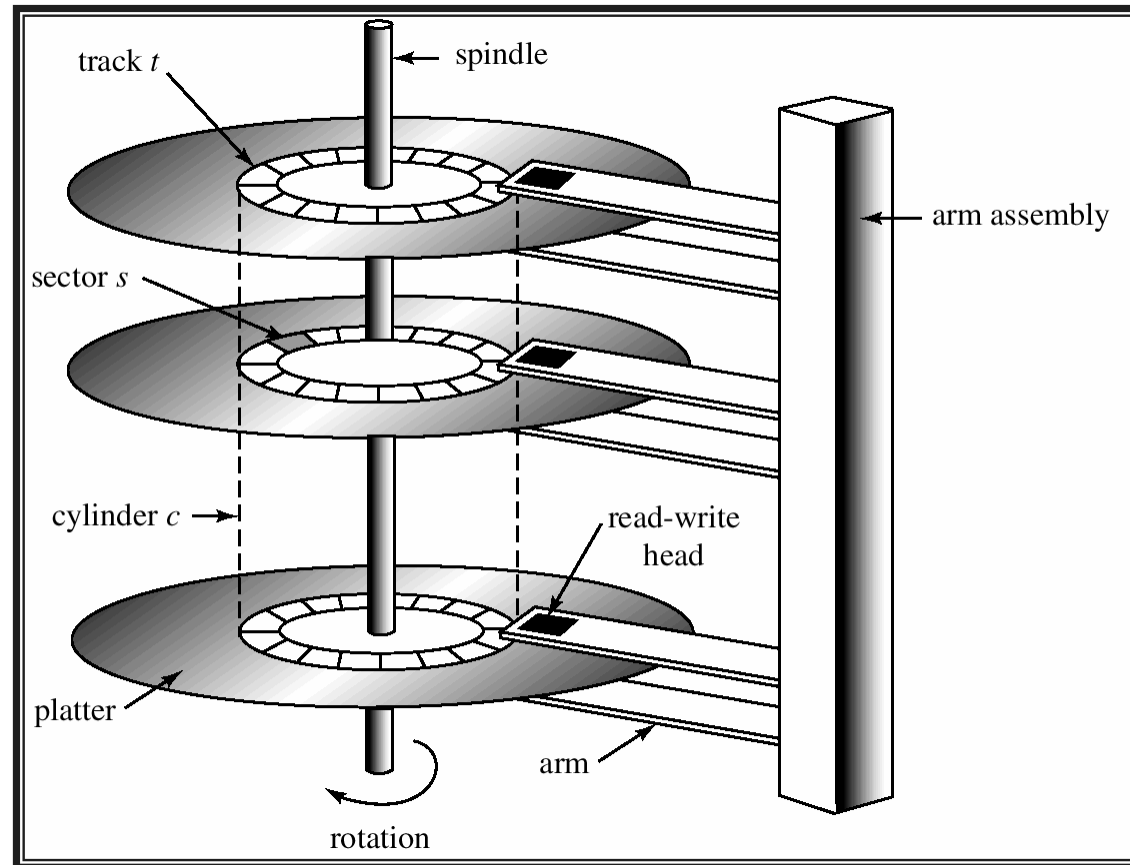
- Gli unici dispositivi di memoria direttamente accessibili dalla CPU sono la memoria centrale e i registri interni alla CPU stessa.
- Non esistono istruzioni macchina che accettino come argomenti indirizzi di un disco.
- Se i dati necessari ad una elaborazione non sono disponibili in memoria, essi devono essere trasferiti affinché la CPU possa operare su di essi.
- Ciascun controllore di I/O è dotato di registri nel quale mantenere i comandi e i dati da trasferire.
- Per rendere agevole l'accesso ai dispositivi, alcune architetture forniscono l'**I/O associato in memoria** (*memory-mapped I/O*), che consente di far corrispondere i registri dei dispositivi a indirizzi della CPU.
- Ogni riferimento a tali indirizzi da parte della CPU, comporta il un trasferimento di dati con i registri del dispositivo.

I/O mappato in memoria

- Questo metodo è particolarmente adatto a dispositivi con brevi tempi di risposta, come le schede video.
- Ad esempio, nei PC i controllori video dispongono di un'ampia memoria, detta memoria video, le cui locazioni corrispondono alle posizioni sullo schermo.
- Per inviare una sequenza di byte la CPU scrive un byte nel registro dei dati, quindi imposta un bit nel registro di controllo del dispositivo per indicare che il byte è disponibile.
- Il dispositivo riceve tale byte e cancella il bit di controllo per segnalare che è pronto a riceverne altri.
 - ◆ Se la CPU interroga ciclicamente (*polling*) il bit di controllo per verificare se il dispositivo è pronto, si chiama **I/O programmato**.
 - ◆ Se si attende un'interruzione che segnali che il dispositivo è pronto, si chiama **I/O guidato dalle interruzioni**.

Dischi magnetici

- **Dischi magnetici:** piatti metallici o vitrei rigidi coperti di materiale magnetico.
- Le testine sono attaccate a **braccia del disco**.
- La superficie del disco è logicamente divisa in **tracce** circolari, a loro volta divise in **settori**.
- L'insieme delle tracce corrispondenti ad una posizione del braccio sono dette **cilindri**.
- Il **controllore del disco** (*disk controller*) determina l'interazione tra il dispositivo e il computer.



Dischi magnetici

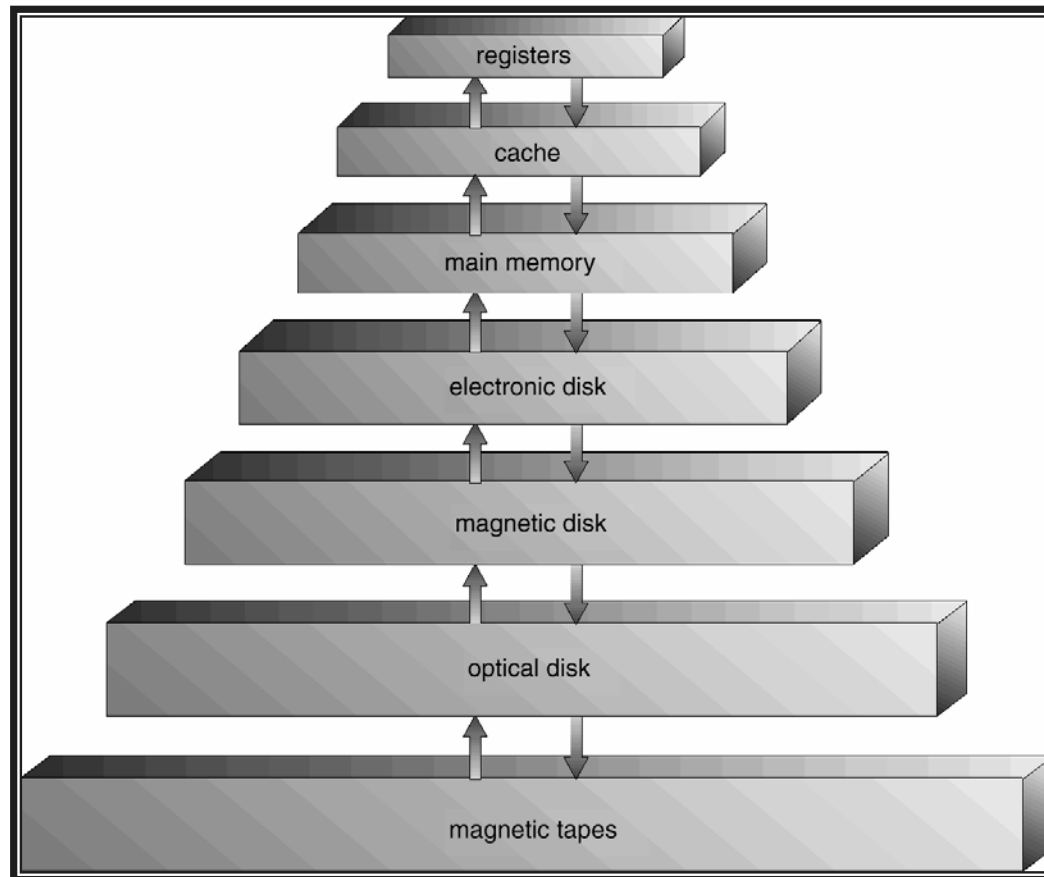
- L'efficienza di un disco è caratterizzata da tre valori:
 - ◆ la **velocità di trasferimento** è la velocità con cui i dati fluiscono dal disco al calcolatore;
 - ◆ il **tempo di posizionamento** è il tempo necessario a spostare il braccio del disco sul cilindro desiderato;
 - ◆ la **latenza di rotazione** è il tempo necessario affinché il settore desiderato si porti tramite la rotazione del disco sotto la testina.
- Un'unità a disco è connessa al calcolatore mediante un bus, quali **EIDE** (*enhanced integrated drive electronics*), **ATA** (*advanced technology attachment*) e **SCSI**

Nastri magnetici

- **Nastri magnetici:** un nastro avvolto in bobina su cui è possibile memorizzare permanentemente grandi quantità di dati
 - ◆ L'accesso sequenziale ai dati ne ha ristretto i campi di applicazione
 - ◆ Utilizzati principalmente per back-up e trasferimenti di dati

Gerarchia delle memorie

- I numerosi componenti di memoria di un sistema di calcolo si possono organizzare in una gerarchia.
 - ◆ Velocità, Costo e Volatilità

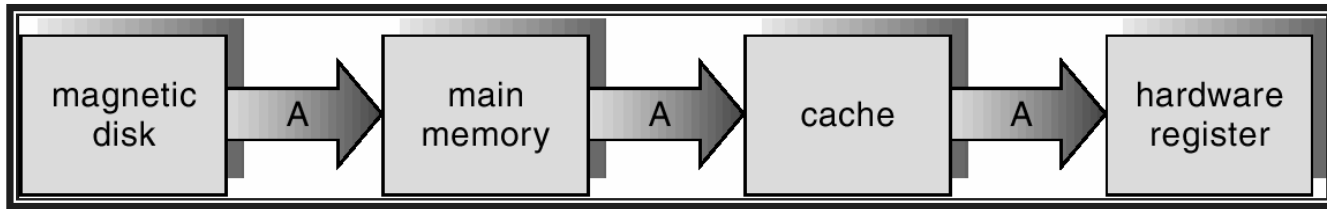


Caching

- **Cache:** Memorie ad alta velocità che vengono utilizzate per memorizzare dati recentemente utilizzati.
- Se la CPU deve accedere di nuovo a dati già utilizzati, essi possono essere prelevati direttamente dalla cache.
- I registri programmabili della CPU rappresentano una cache gestibile dal programmatore e/o dal compilatore.
- Richiede politiche di gestione della cache.
- Il caching introduce un altro livello nella gerarchia di memorizzazione. Ciò richiede che i dati simultaneamente memorizzati nei vari livelli siano *consistenti*

Coerenza

- Esempio: Incrementare di 1 il valore n contenuto nel file f , registrato su disco magnetico
 - ✦ copia in memoria centrale del blocco del file che contiene il valore n
 - ✦ copia di n nella cache
 - ✦ copia di n nei registri della CPU



- Dopo l'incremento il valore di n nei registri sarà differente dagli altri; le diverse copie avranno lo stesso valore solo quando n sarà memorizzato sul disco.
- Nei sistemi a partizione del tempo è necessario prestare attenzione che ogni processo che vuole accedere al valore n , acceda al valore più aggiornato.
- La situazione si complica nei sistemi multiprocessori e multicomputer.

Architetture di Protezione

- I primi sistemi di calcolo ammettevano la presenza di un solo utente alla volta ed erano gestite da un operatore che aveva il completo controllo dell'intero sistema.
- Con l'evoluzione dei programmi di sistema il controllo passò gradualmente al sistema operativo (monitor), che eseguiva essenzialmente operazioni di I/O.
- I sistemi operativi resero possibile la condivisione delle risorse del sistema tra più programmi e la gestione asincrona (*spooling*) delle operazioni di I/O, mentre altri processi erano in esecuzione.
- Se queste condivisioni incrementarono l'utilizzo del sistema, introdussero nuovi problemi da risolvere.
- Il verificarsi di un errore in un programma poteva influenzare il comportamento di altri programmi e del sistema operativo stesso.

Architetture di Protezione

- Senza un'adeguata protezione bisogna scegliere:
 - ◆ consentire l'esecuzione di un processo alla volta;
 - ◆ considerare sospetti tutti i risultati.
- Gli errori vanno riconosciuti direttamente dall'architettura e gestiti dal sistema operativo.
- Se un programma utente compie un'operazione non lecita:
 - ◆ l'architettura di protezione invia un segnale d'eccezione al sistema operativo;
 - ◆ il segnale determina il trasferimento del controllo al sistema operativo mediante il vettore delle interruzioni;
 - ◆ il gestore del segnale pone fine all'esecuzione del programma, eventualmente scrivendo in un file l'immagine della memoria (*dump*).

Duplici modo di funzionamento

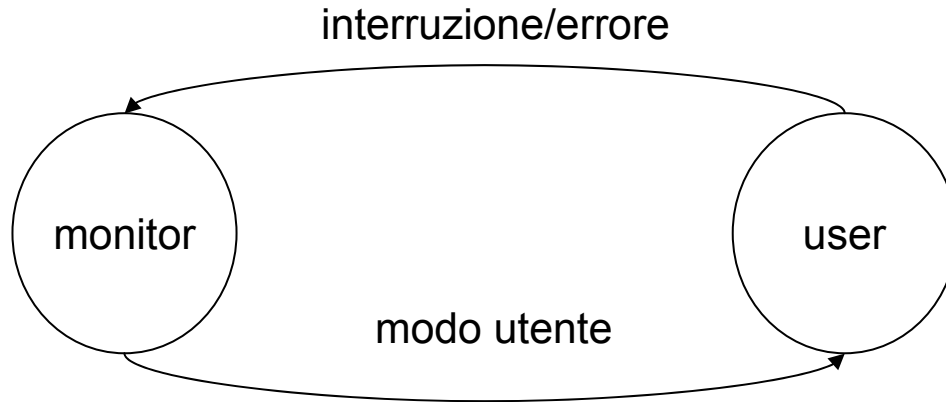
- La condivisione delle risorse richiede che il sistema operativo assicuri che un programma non corretto non possa causare errori in altri programmi.
- E' necessario che l'architettura del sistema fornisca un supporto hardware per differenziare almeno tra due modi di funzionamento:
 - ◆ **Modo utente:** l'esecuzione avviene per conto dell'utente.
 - ◆ **Modo di sistema:** l'esecuzione avviene per conto del sistema operativo.
- Per indicare quale modo è attivo l'architettura deve essere dotata di un **bit di modo** che consenta di stabilire per conto di chi è eseguita l'istruzione corrente.

Dupliche modo di funzionamento

- Il dupliche modo di funzionamento consente di proteggere il sistema operativo da programmi utente e programmi utente da altri programmi utente.
- La protezione si ottiene definendo come **istruzioni privilegiate** quelle istruzioni che possono causare danni, e permettere solo al sistema operativo di eseguirle.
- Le richieste di esecuzione di istruzioni privilegiate sono chiamate **chiamate di sistema** (*system call*).

Chiamate di sistema

- Quando un programma utente effettua una chiamata di sistema:
 - ◆ la CPU riceve un segnale di interruzione e, tramite il gestore delle interruzioni passa il controllo alla procedura di servizio e pone il bit di modo nel modo di sistema;
 - ◆ la procedura esamina l'istruzione che ha causato l'interruzione e interpreta il tipo di servizio richiesto;
 - ◆ soddisfatta la richiesta si restituisce il controllo all'istruzione immediatamente seguente la chiamata di sistema.



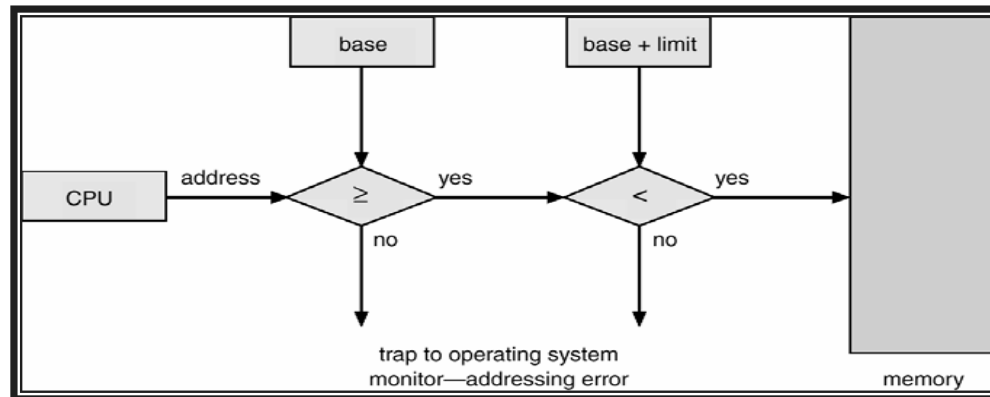
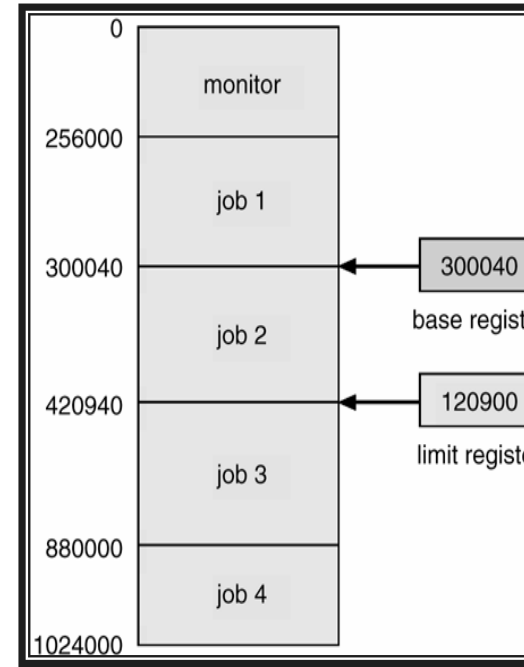
Le istruzioni privilegiate possono essere eseguite solo in modo di sistema

Protezione dell'I/O

- Per evitare che il funzionamento del sistema venga alterato da operazioni di I/O illegali, tutte le istruzioni di I/O sono istruzioni privilegiate.
- Se un utente riuscisse ad alterare il puntatore alla procedura di gestione nel vettore delle interruzioni, potrebbe mandare in esecuzione in modo di sistema un'arbitraria procedura.
- Naturalmente è possibile ottenere il controllo del sistema in molti altri modi.

Protezione della memoria

- Bisogna proteggere almeno il vettore delle interruzioni e le procedure di gestione.
- In generale, è necessario proteggere il sistema operativo dagli accessi dei programmi utente e i programmi utente dagli altri programmi.
- La protezione della memoria si ottiene aggiungendo due registri che determinano l'insieme degli indirizzi legali:
 - ◆ **Registro di base:** contiene il più piccolo indirizzo legale di memoria fisica.
 - ◆ **Registro limite:** contiene la dim. dell'intervallo
- La memoria al di fuori di questo intervallo è protetta.



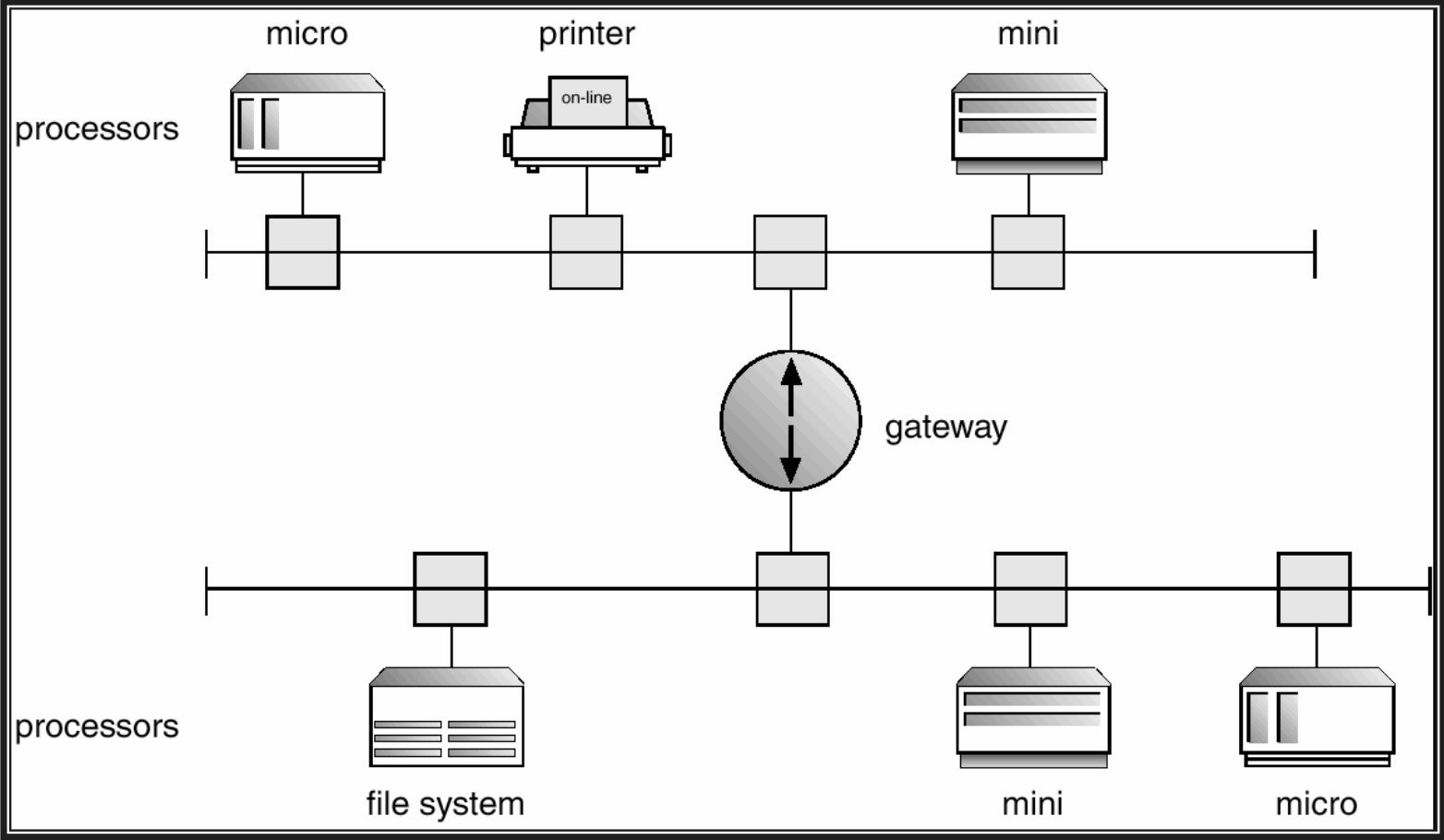
Protezione della CPU

- E' possibile che un programma utente entri in un ciclo infinito e non restituisca più il controllo al sistema operativo.
- A tale scopo si può usare un **temporizzatore** programmabile che invia interruzioni alla CPU a intervalli specificati, fissi o variabili:
 - ◆ Il temporizzatore è decrementato ad ogni clock tick.
 - ◆ Quando il temporizzatore raggiunge il valore 0, viene generata un'interruzione.
- La presenza di un temporizzatore assicura che nessun programma utente possa essere eseguito all'infinito.
- Essi sono impiegati per realizzare la partizione del tempo, inviando un'interruzione dopo ciascun **quanto di tempo**.
- Le operazioni necessarie per preparare l'esecuzione del programma successivo vengono dette **cambio di contesto**.

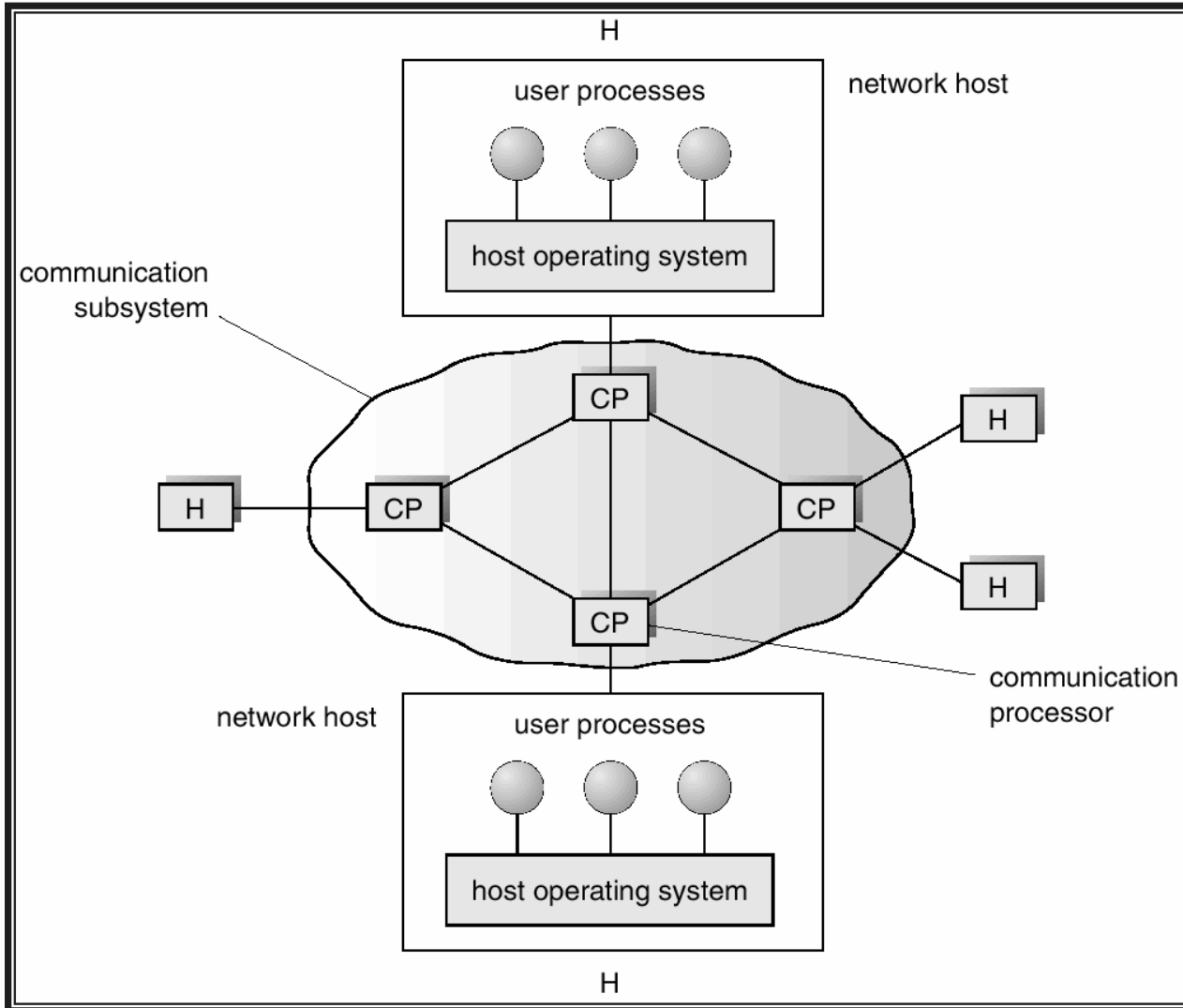
Struttura delle reti di calcolatori

- Esistono due tipi fondamentali di rete:
- **Reti locali** (*Local Area Networks - LAN*)
- **Reti geografiche** (*Wide Area Networks - WAN*)

Reti locali



Reti geografiche



Sommario

- I sistemi multiprogrammati e a partizione del tempo migliorano le loro prestazioni sovrapponendo le operazioni della CPU e quelle di I/O.
- I sistemi operativi moderni sono **guidati dalle interruzioni**: se non ci sono richieste di I/O da gestire, processi da eseguire o utenti con cui interagire, esso resta inattivo.
- I sistemi di memorizzazione di un calcolatore si possono organizzare in modo gerarchico secondo la velocità, costo e volatilità.
- Per garantire il corretto funzionamento, il sistema operativo si avvale di meccanismi forniti hardware quali il duplice modo di funzionamento, la protezione della memoria, dell'I/O e della CPU.
- Le reti di calcolatori locali e geografiche permettono di collegare calcolatori distanti per condividere risorse e scambiare informazioni.