

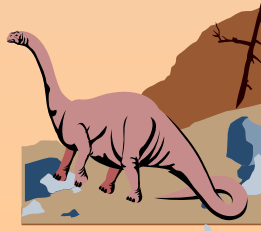
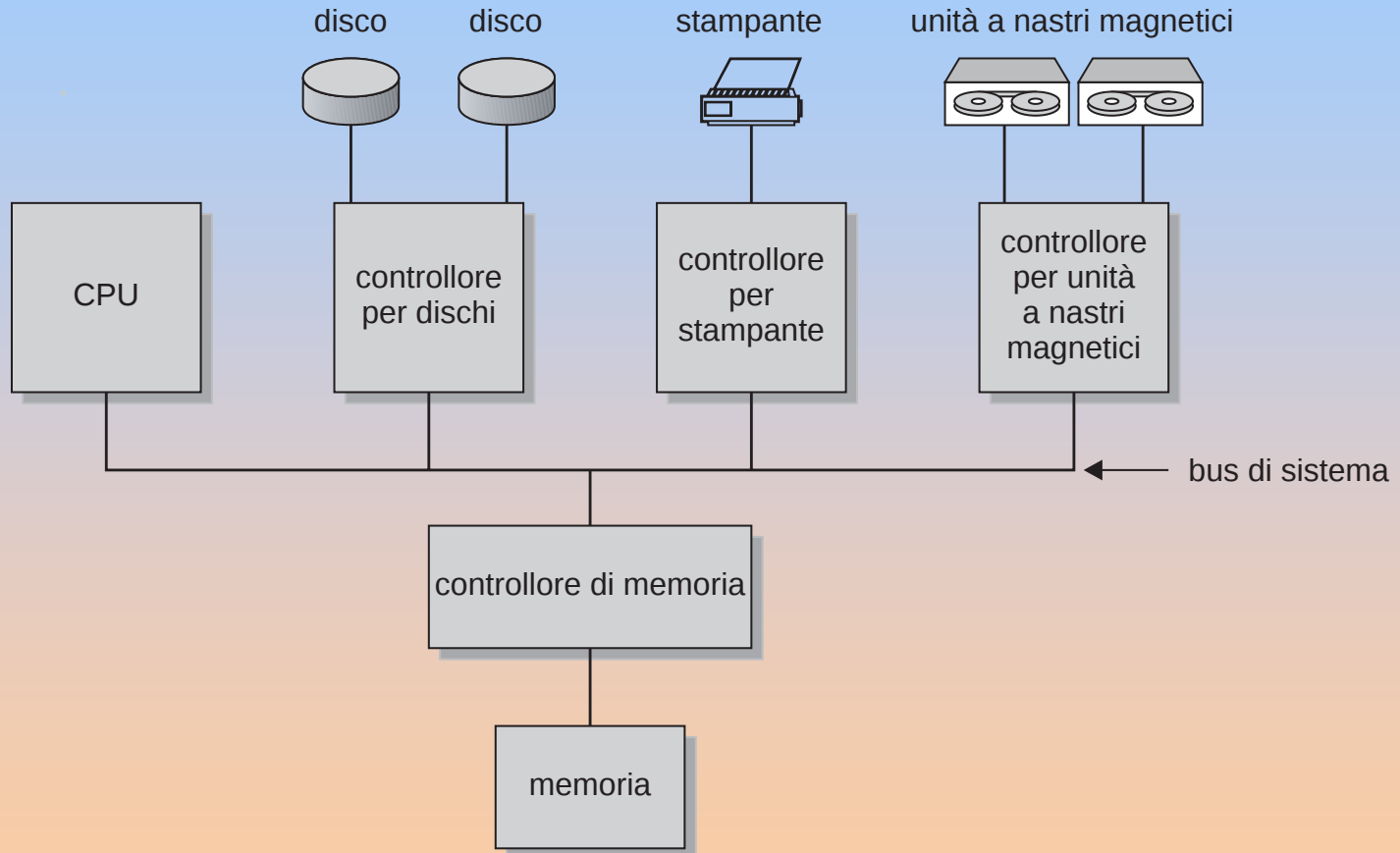
Strutture dei sistemi di calcolo

- Funzionamento di un sistema di calcolo
- Struttura di I/O
- Struttura della memoria
- Gerarchia delle memorie
- Architetture di protezione





Architettura di un sistema di calcolo





Funzionamento di un sistema di calcolo

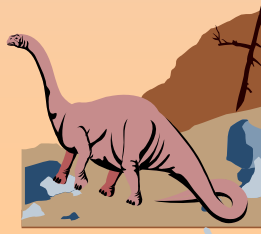
- I dispositivi di I/O e la CPU possono operare in modo concorrente.
- Ciascun controllore si occupa di un particolare tipo di dispositivo fisico (es. unità a disco, dispositivi audio, ...).
- Ciascun controllore ha un buffer locale.
- La CPU sposta i dati da/verso la memoria principale da/verso i buffer locali.
- L'I/O avviene dal dispositivo al buffer locale del controllore.
- Il controllore informa la CPU di aver terminato un'operazione causando un **segnale d'interruzione** (*interrupt*).





Funzioni comuni dei segnali d'interruzione

- Un segnale d'interruzione deve causare il trasferimento del controllo all'appropriata procedura di servizio dell'evento a esso associato.
- L'architettura di gestione delle interruzioni deve anche salvare l'indirizzo dell'istruzione interrotta.
- Un **segnale d'eccezione** (*trap*) può essere causato da un programma in esecuzione a seguito di un evento eccezionale oppure a seguito di una richiesta specifica effettuata da un programma utente.
- Un moderno sistema operativo è detto **guidato dalle interruzioni** (*interrupt driven*).





Gestione dell'interruzione

- Il sistema operativo memorizza l'indirizzo di ritorno nella pila (*stack*) di sistema.
- Determina quale tipo di interruzione si è verificato:
 - ☞ *polling*
 - ☞ *vectored* interrupt system
- Segmenti separati di codice determinano quale azione debba essere intrapresa per ciascun tipo di interruzione.

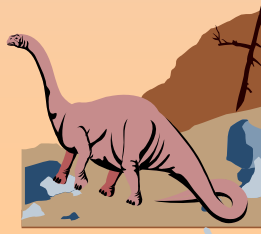
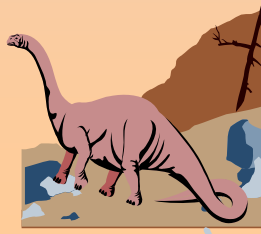
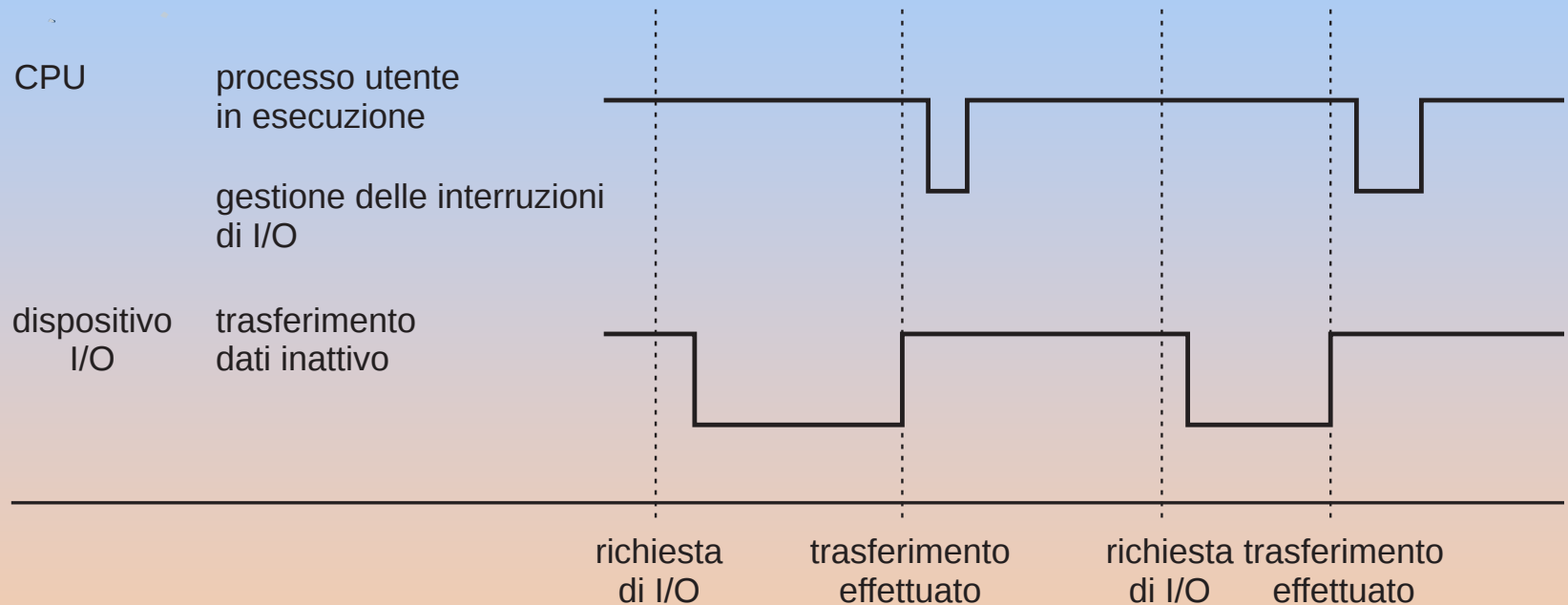




Diagramma temporale delle interruzioni per un singolo processo che emette dati





Struttura di I/O

- Una volta iniziata l'operazione di I/O, si restituisce il controllo al processo utente solo dopo il completamento dell'operazione di I/O.
 - ☞ L'istruzione `wait` sospende la CPU fino al successivo segnale d'interruzione.
 - ☞ Nei calcolatori che non prevedono un'istruzione del genere, si può generare in ciclo d'attesa
 - ☞ Si ha al più una richiesta pendente alla volta.
- Una volta iniziata l'operazione di I/O, si restituisce immediatamente il controllo al processo utente, senza attendere il completamento dell'operazione di I/O.
 - ☞ *System call* – richiesta al sistema operativo di consentire al programma utente di attendere il completamento dell'operazione.
 - ☞ La *tabella di stato dei dispositivi* contiene elementi per ciascun dispositivo di I/O che ne specificano il tipo, l'indirizzo e lo stato.
 - ☞ Il sistema operativo individua il controllore del dispositivo che ha emesso il segnale d'interruzione, quindi accede alla tabella dei dispositivi, risale allo stato in cui il dispositivo si trova e modifica l'elemento della tabella indicando l'occorrenza dell'interruzione.

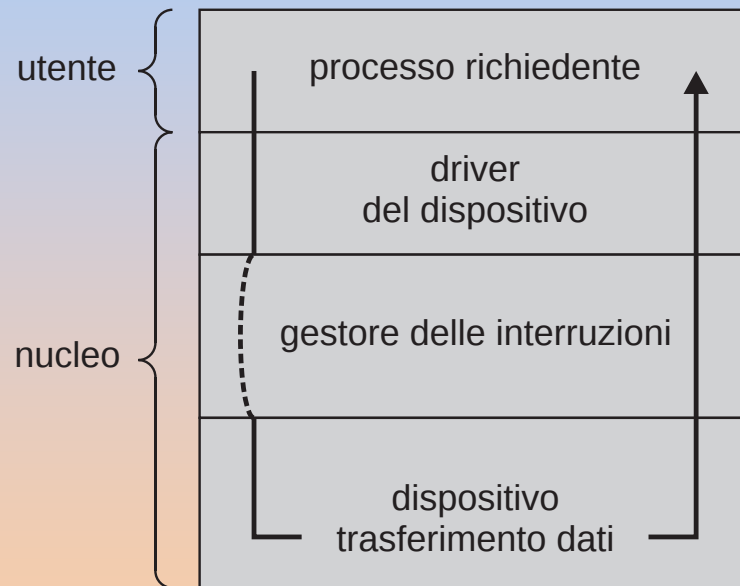




Due metodi di I/O

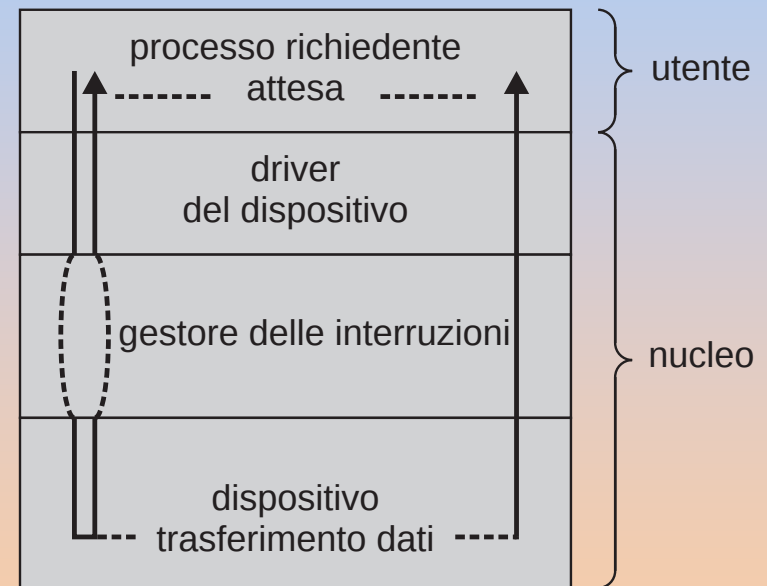
sincrono

asincrono



tempo →

(a)



tempo →

(b)

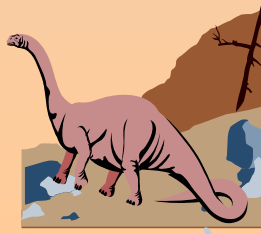
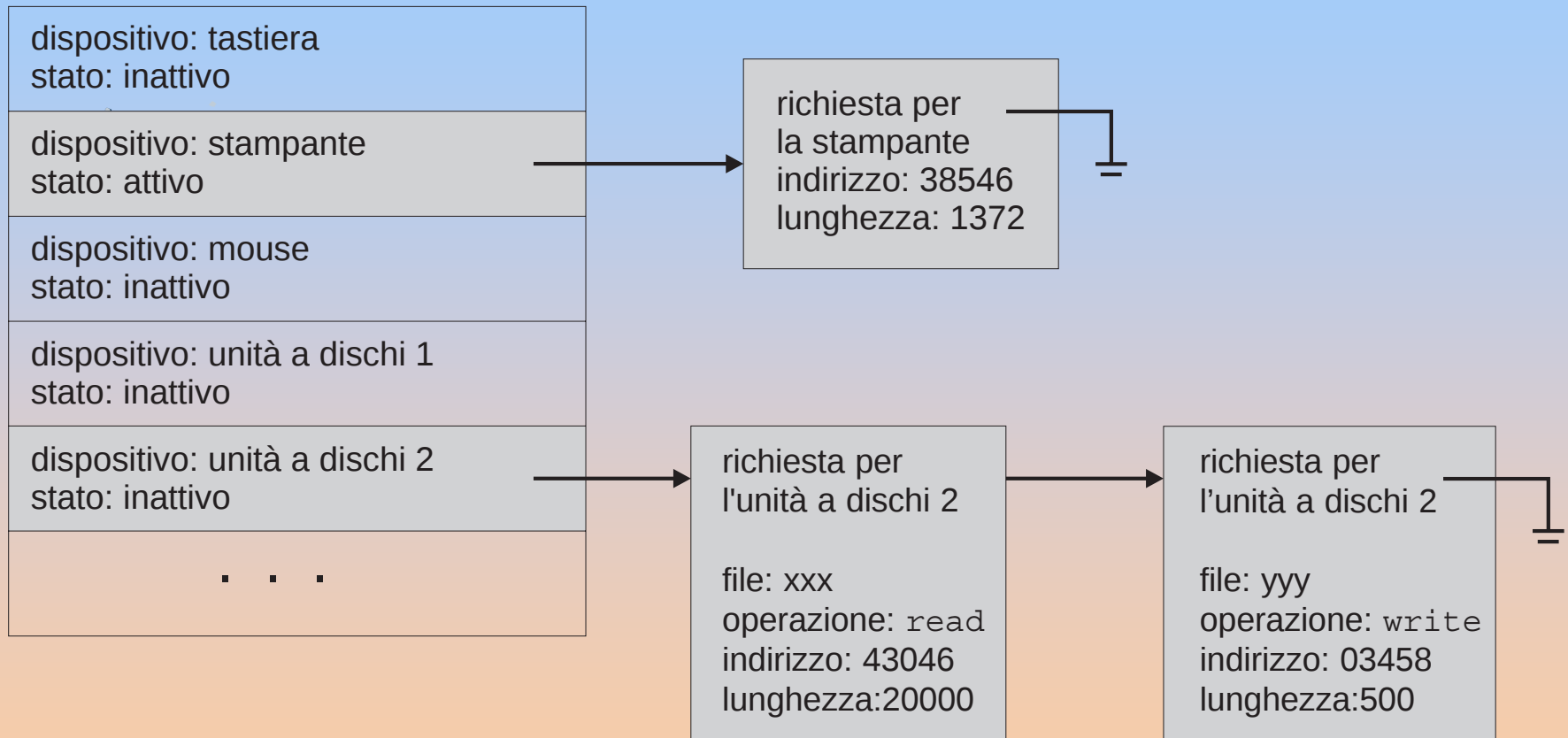




Tabella di stato dei dispositivi





Accesso diretto alla memoria

- Tecnica usata con dispositivi di I/O veloci.
- Il controllore trasferisce un intero blocco di dati dalla propria memoria di transito direttamente nella memoria centrale, o viceversa, senza alcun intervento da parte della CPU.
- Il trasferimento richiede una sola interruzione per ogni blocco di dati trasferito, piuttosto che per ogni byte (o parola).



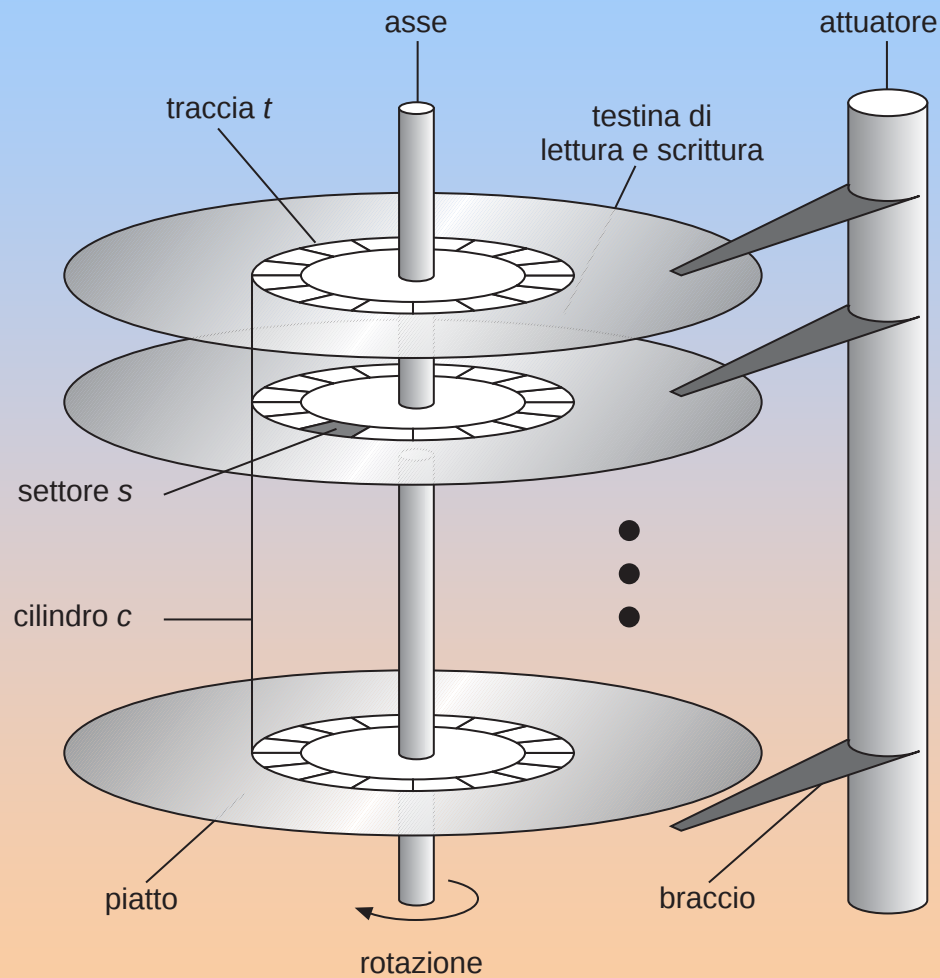


Struttura della memoria

- **Memoria centrale:** dispositivo di memoria direttamente accessibile dalla CPU.
- **Memoria secondaria:** estensione della memoria centrale capace di conservare in modo permanente grandi quantità di informazioni.
- **Disco magnetico:** piatto rigido di metallo o vetro ricoperto di materiale magnetico
 - ☞ La superficie del disco è divisa logicamente in *tracce* circolari a loro volta suddivise in *settori*.
 - ☞ I *controllori dei dischi* determinano l'interazione logica tra il dispositivo e il calcolatore.



Schema funzionale di un disco





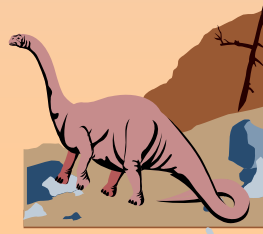
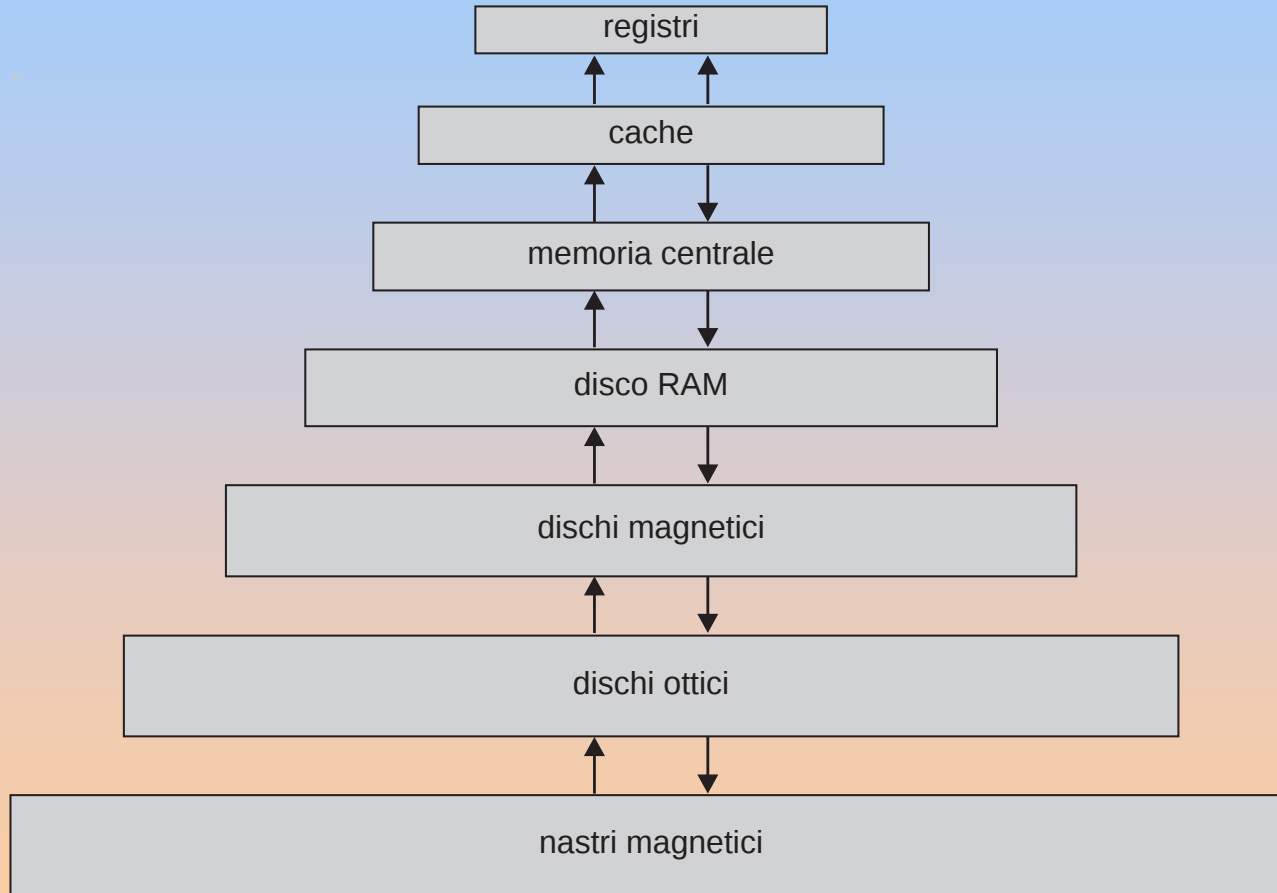
Gerarchia delle memorie

- I componenti di memoria di un sistema di calcolo possono essere organizzati in una struttura gerarchica in base a:
 - ☞ velocità
 - ☞ costo
 - ☞ volatilità
- **Cache:** copia temporanea di informazioni in un'unità più veloce; la memoria centrale si può considerare una cache per la memoria secondaria.





Gerarchia dei dispositivi di memoria





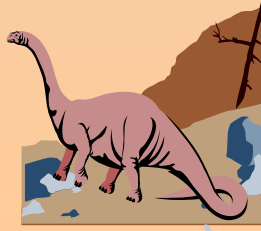
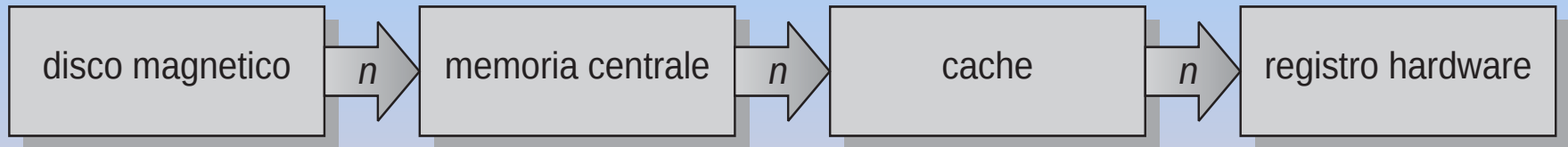
Cache

- Utilizzo di una memoria ad alta velocità per registrare dati ai quali si è avuto accesso recentemente (e che si prevede che presto serviranno ancora).
- Richiede una politica di **gestione della cache**.
- La cache introduce un altro livello nella gerarchia delle memorie. Ciò richiede che i dati che sono memorizzati contemporaneamente su più livelli siano *coerenti*.





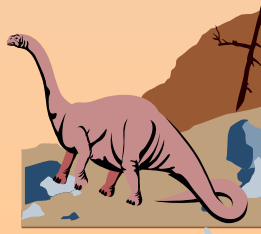
Migrazione di un intero n da un disco a un registro





Architetture di protezione

- Duplice modo di funzionamento (*dual-mode*)
- Protezione dell'I/O
- Protezione della memoria
- Protezione della CPU





Duplica modo di funzionamento

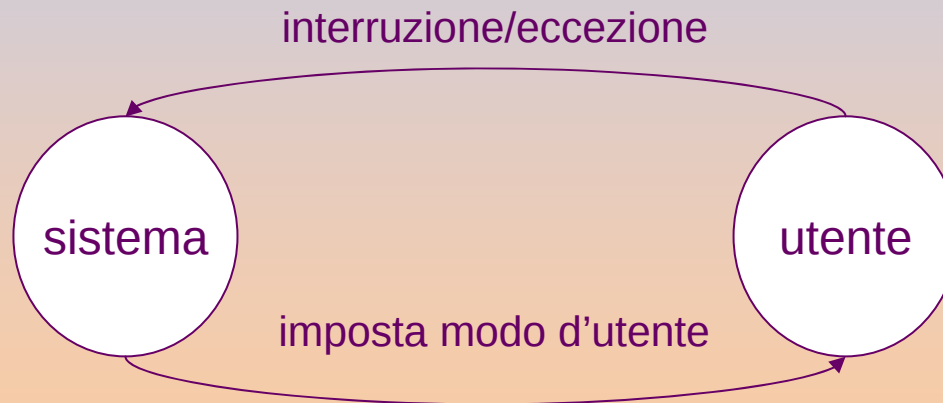
- La condivisione delle risorse di sistema rende necessario che il sistema operativo garantisca che un programma malfunzionante non causi una scorretta esecuzione di altri programmi.
- Specifiche caratteristiche dell'architettura di sistema consentono di gestire almeno due modi di funzionamento.
 1. **Modo d'utente** (*user mode*): l'istruzione corrente si esegue per conto di un utente.
 2. **Modo di sistema** (*monitor mode*, detto anche *modo del supervisore*, *modo monitor* o *modo privilegiato*): l'istruzione corrente si esegue per conto del sistema operativo.



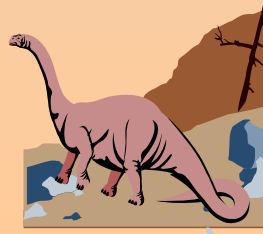


Duplice modo di funzionamento (Cont.)

- Il **bit di modo** (*mode bit*) indica quale modo è attivo: di sistema (0) o d'utente (1).
- Ogni volta che si verifica un'interruzione o un'eccezione si passa dal modo d'utente al modo di sistema, cioè si pone a 0 il bit di modo.



La CPU consente l'esecuzione di **istruzioni privilegiate** (*privileged instruction*) soltanto nel modo di sistema.





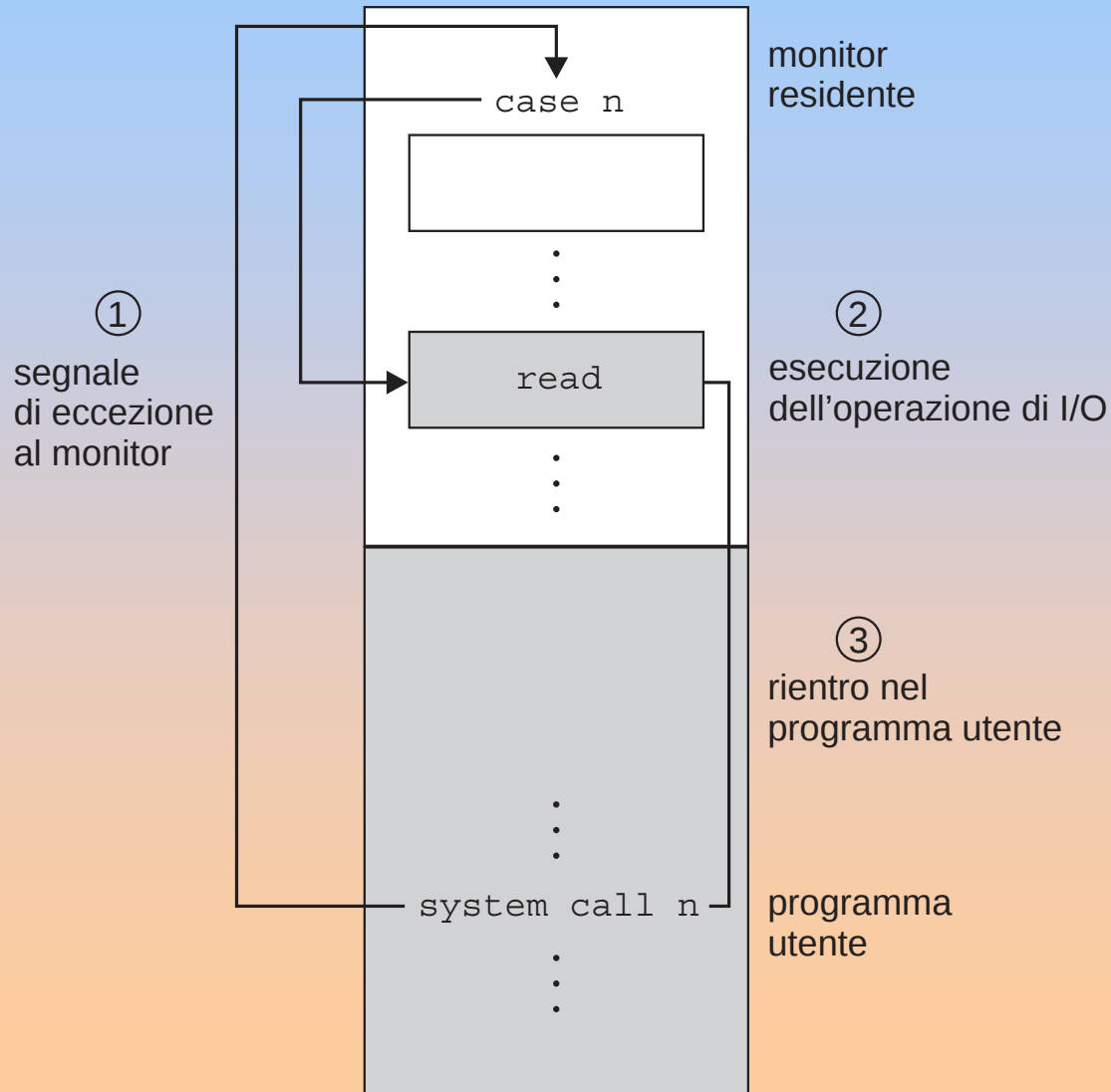
Protezione dell'I/O

- Tutte le istruzioni di I/O sono istruzioni privilegiate.
- Affinché la protezione dell'I/O sia totale, è necessario evitare che l'utente possa in qualsiasi modo ottenere il controllo del calcolatore quando questo si trova nel modo di sistema (es. un programma utente che, come parte della sua esecuzione, memorizza un nuovo indirizzo nel vettore delle interruzioni).





Uso di una chiamata del sistema per l'esecuzione di una chiamata di I/O





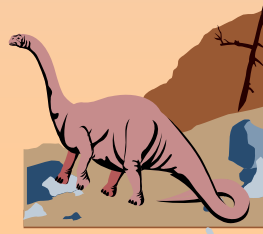
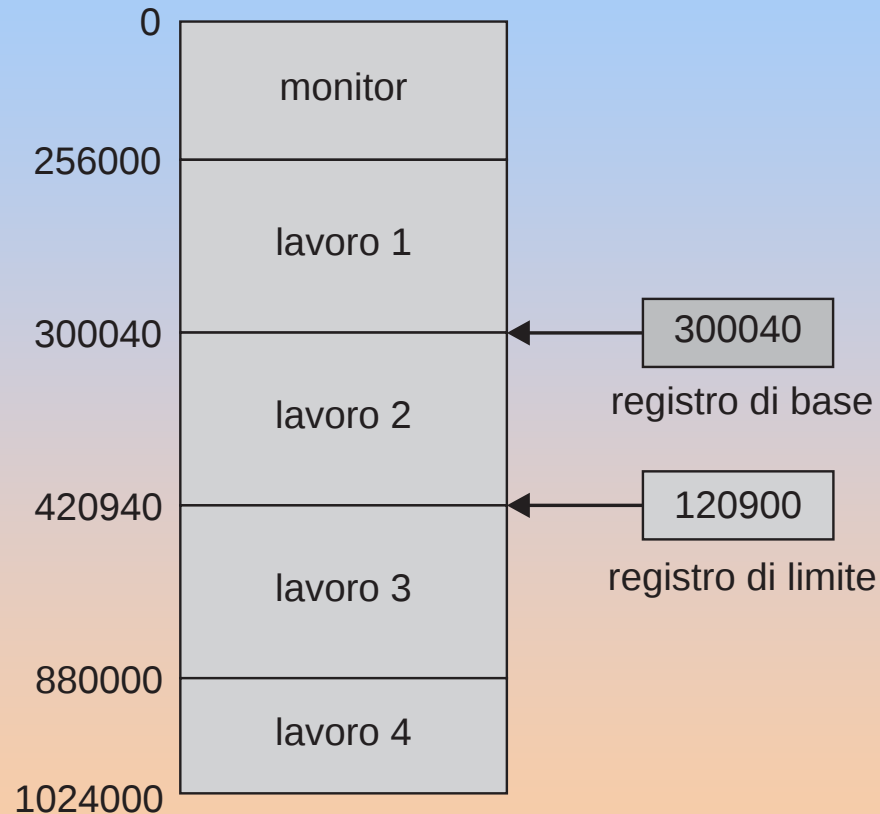
Protezione della memoria

- È necessario fornire protezione della memoria almeno per il vettore delle interruzioni e le relative procedure di servizio contenute nel codice del sistema operativo.
- Questo tipo di protezione si realizza impiegando due registri che contengono l'intervallo degli indirizzi validi cui un programma può accedere:
 - ☞ **Registro di base:** contiene il più basso indirizzo della memoria fisica al quale il programma dovrebbe accedere.
 - ☞ **Registro di limite:** contiene la dimensione dell'intervallo.
- Le aree di memoria al di fuori dell'intervallo stabilito sono protette.



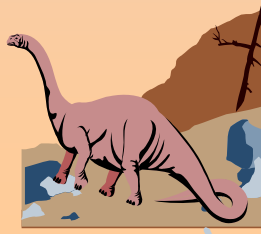
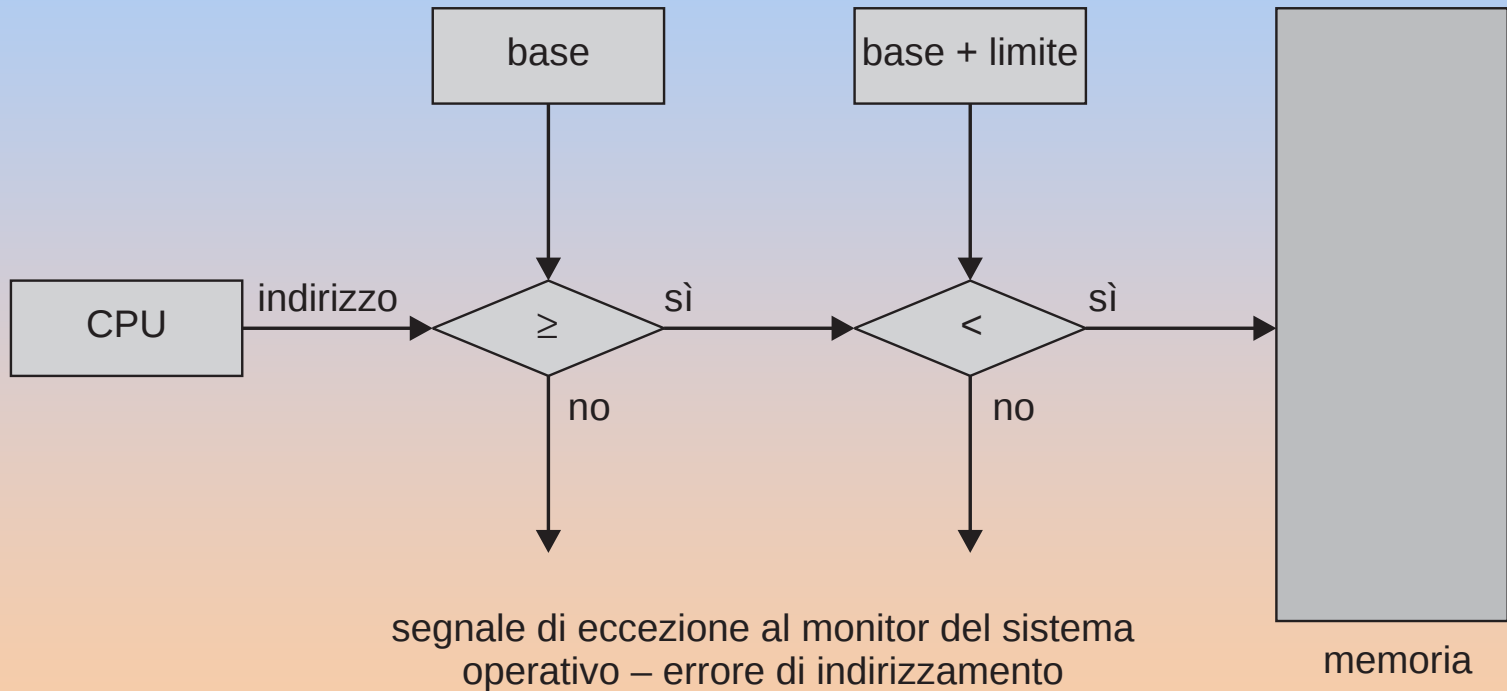


Uso di un registro di base e un registro di limite





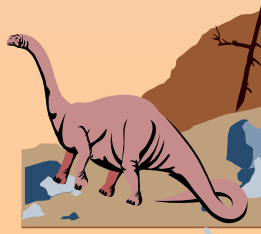
Architettura di protezione degli indirizzi





Protezione hardware

- Funzionando nel modo di sistema, il sistema operativo ha la possibilità di accedere indiscriminatamente sia alla memoria a esso riservata sia a quella riservata agli utenti.
- Questo privilegio consente al sistema di caricare i programmi utenti nelle relative aree di memoria.





Protezione della CPU

- **Temporizzatore** (*timer*): invia un segnale d'interruzione alla CPU a intervalli di tempo specificati per assicurare che il sistema operativo mantenga il controllo dell'elaborazione.
 - ☞ Il timer si decrementa a ogni impulso.
 - ☞ Quando raggiunge il valore 0, si genera un segnale d'interruzione.
- I temporizzatori si usano comunemente per realizzare la partizione del tempo d'elaborazione (*time sharing*).
- Un altro impiego dei temporizzatori è il calcolo dell'ora corrente.
- Load-timer è un'istruzione privilegiata.

