

Tutorato di sistemi operativi



Hardware di base

Dove risiedono le istruzioni in esecuzione?

Hardware di base

Dove risiedono le istruzioni in esecuzione?

- Registri (1 ciclo di clock)
- Memoria centrale (diversi cicli di clock)

Separazione dei processi

Come vengono separati i processi?

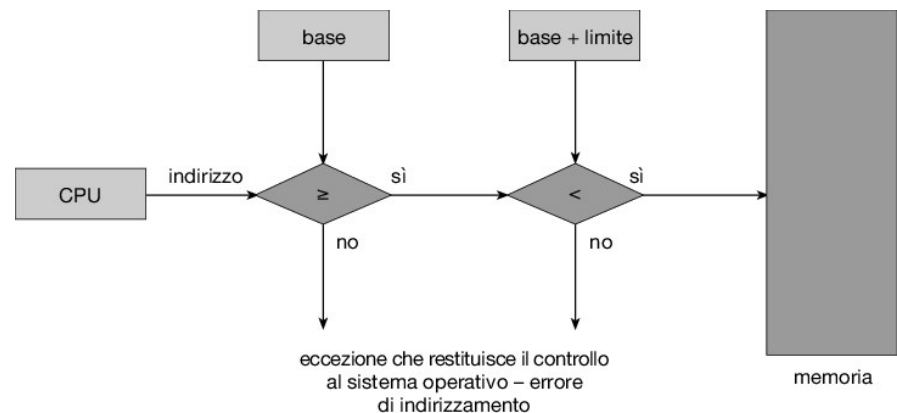
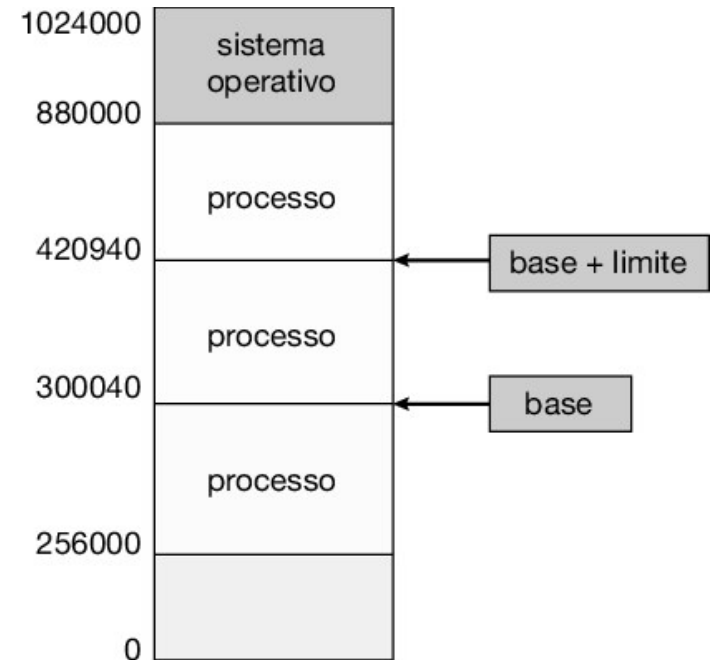
Separazione dei processi

Come vengono separati i processi?

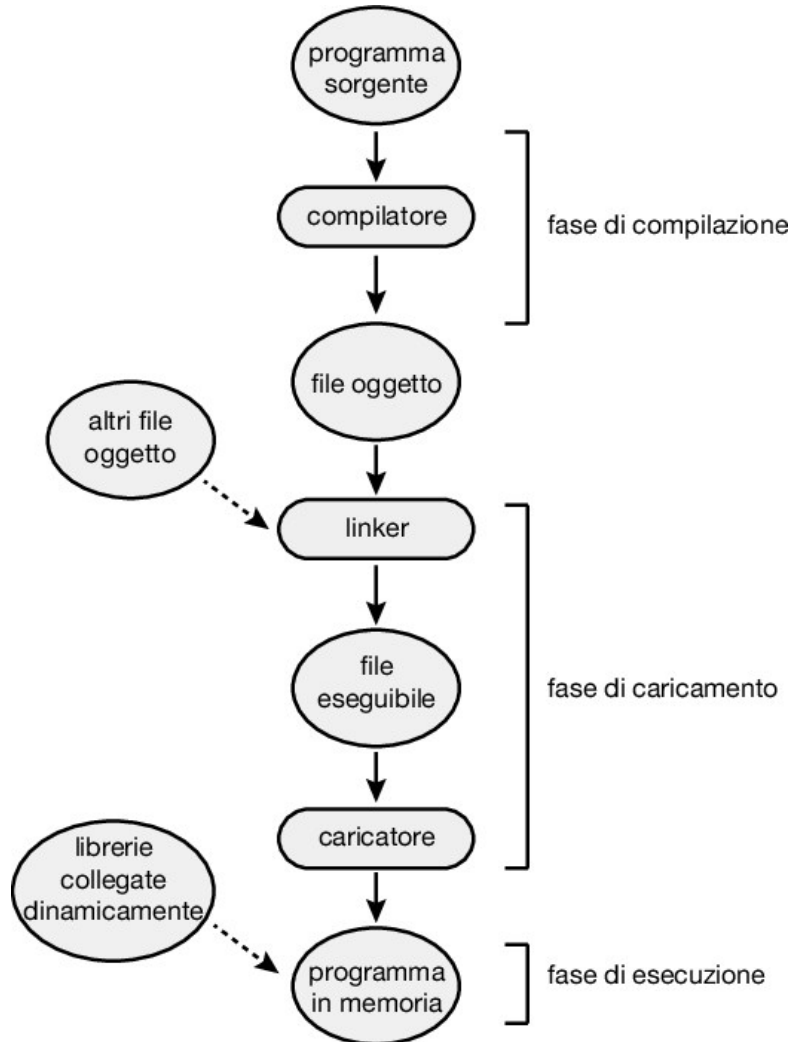
Viene delimitato l'intervallo degli indirizzi a cui un processo può accedere (legalmente).

Una possibile implementazione del meccanismo di protezione avviene tramite l'utilizzo di due registri:

- Base
- Limite



Associazione degli indirizzi



Se si conosce già dove il processo risiederà.

Se non si conosce già la posizione del codice viene generato del codice rilocabile.

Se il processo può essere spostato in fase di esecuzione.

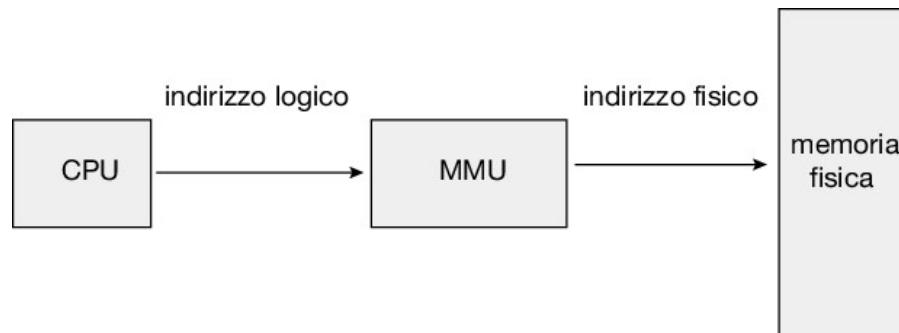
Spazi di indirizzi

Logici

- Generato dalla cpu.
- Gli indirizzi corrispondono se l'associazione avviene a fase di compilazione o di caricamento.
- Gli indirizzi non corrispondono se l'associazione avviene a fase di esecuzione. In questo caso ci si riferisce agli indirizzi logici come indirizzi virtuali.

Fisici

- Visto dall'unità di memoria.



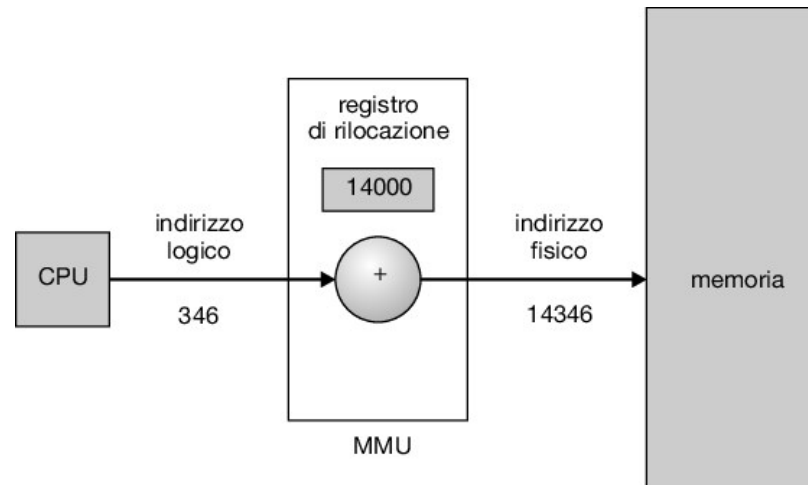
Spazi di indirizzi

Logici

- Generato dalla cpu.
- Gli indirizzi corrispondono se l'associazione avviene a fase di compilazione o di caricamento.
- Gli indirizzi non corrispondono se l'associazione avviene a fase di esecuzione. In questo caso ci si riferisce agli indirizzi logici come indirizzi virtuali.

Fisici

- Visto dall'unità di memoria.



Caricamento dinamico

Perché si ricorre al caricamento dinamico?

Caricamento dinamico

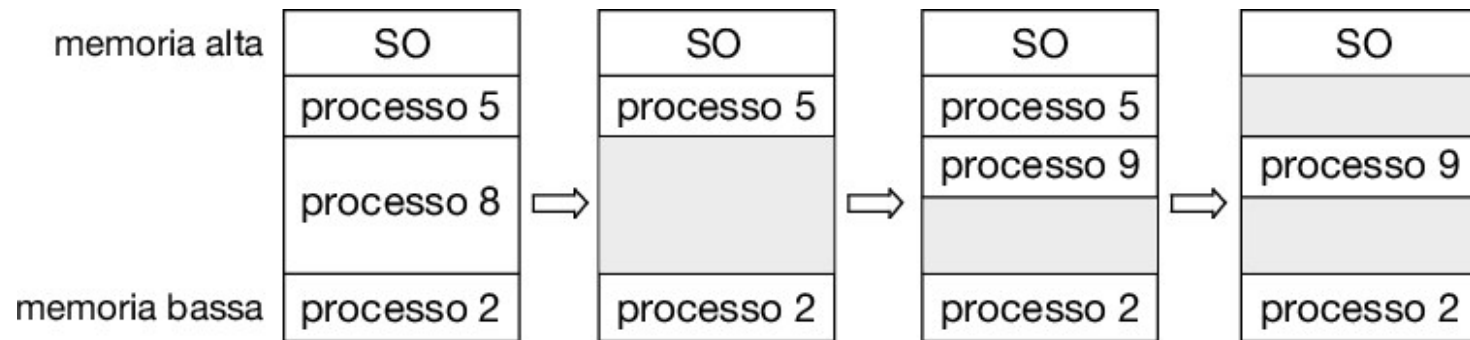
Perché si ricorre al caricamento dinamico?

- Per migliorare l'utilizzo della memoria
- Le procedure vengono caricate in memoria solo quando vengono chiamate.

Le librerie di sistema sono, tipicamente, caricate dinamicamente.

Allocazione della memoria

- L'allocazione della memoria consiste nella sua suddivisione in partizioni di dimensioni variabili.
- Ciascuna partizione può contenere un solo processo.
- Il sistema operativo conserva una tabella in cui sono tracciate le partizioni disponibili e occupate.
- Scelta del primo buco libero tramite i criteri first-fit, best-fit, worst-fit.



Frammentazione

Cos'è la frammentazione e come la risolvo?

Frammentazione

Cos'è la frammentazione e come la risolvo?

Caricando e rimuovendo i processi dalla memoria si frammenta lo spazio libero in tante piccole parti.

Sono presenti 2 tipi di frammentazione:

- Interna, si verifica quando la memoria allocata a un processo è maggiore di quella effettivamente necessaria.
- Esterna, si verifica quando la memoria libera è suddivisa in piccoli blocchi non contigui.

Come soluzione si potrebbe:

- Tracciare i buchi di memoria, causando troppo overhead.
- Utilizzare la compattazione, che non è sempre possibile.

Paginazione

Cos'è la paginazione?

Paginazione

Cos'è la paginazione?

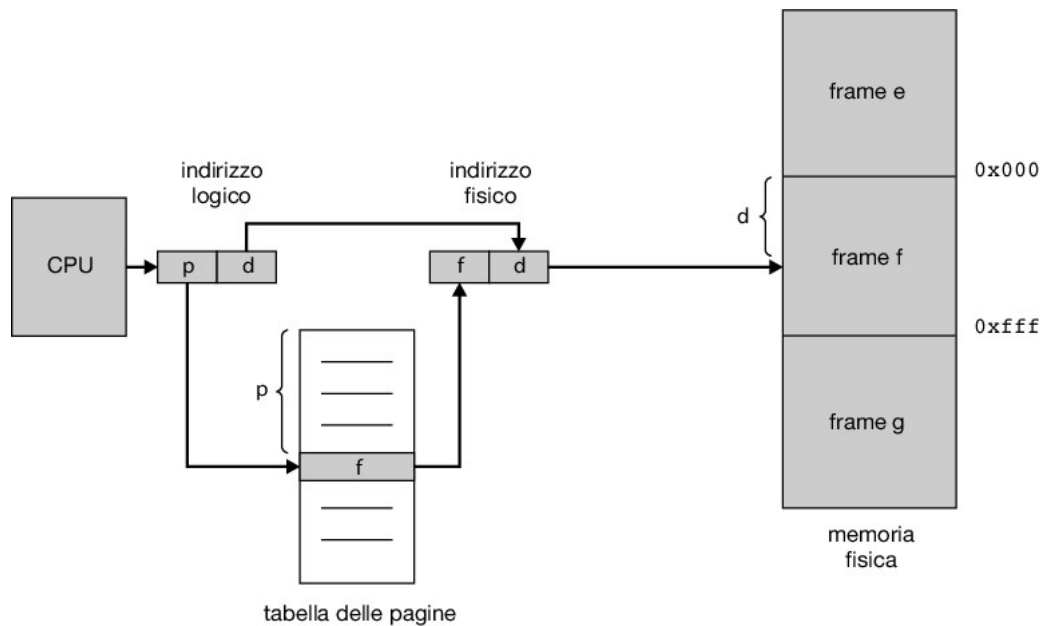
Uno schema di gestione della memoria che permette allo spazio di indirizzi fisici di non essere contiguo.

Consiste nel suddividere la memoria fisica in blocchi di dimensione fissa (frame) e nel suddividere la memoria logica in blocchi di pari dimensione (pagine).

In questo modo lo spazio degli indirizzi logici è completamente separato da quello degli indirizzi fisici.

Paginazione

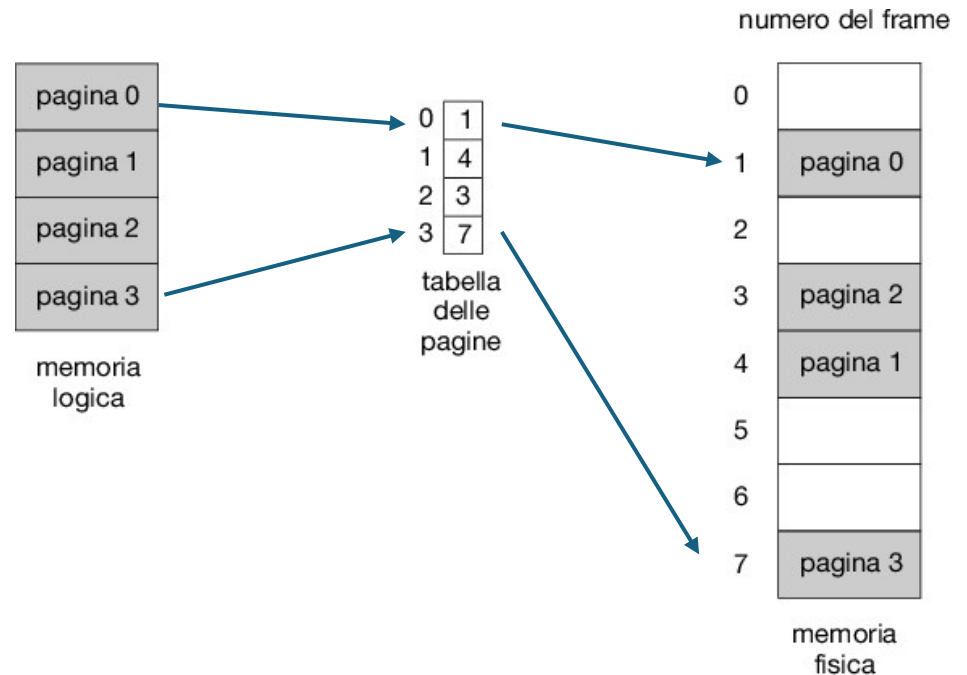
Ogni indirizzo generato dalla cpu è diviso in due parti: un numero di pagina (p), e un offset di pagina (d).



Traduzione dell'indirizzo logico

Passaggi per la traduzione:

1. Estrazione del numero di pagina p e utilizzo come indice nella tabella delle pagine.
2. Estrazione del numero di frame f corrispondente dalla tabella delle pagine.
3. Sostituzione del numero di pagina p nell'indirizzo logico con il numero di frame f .

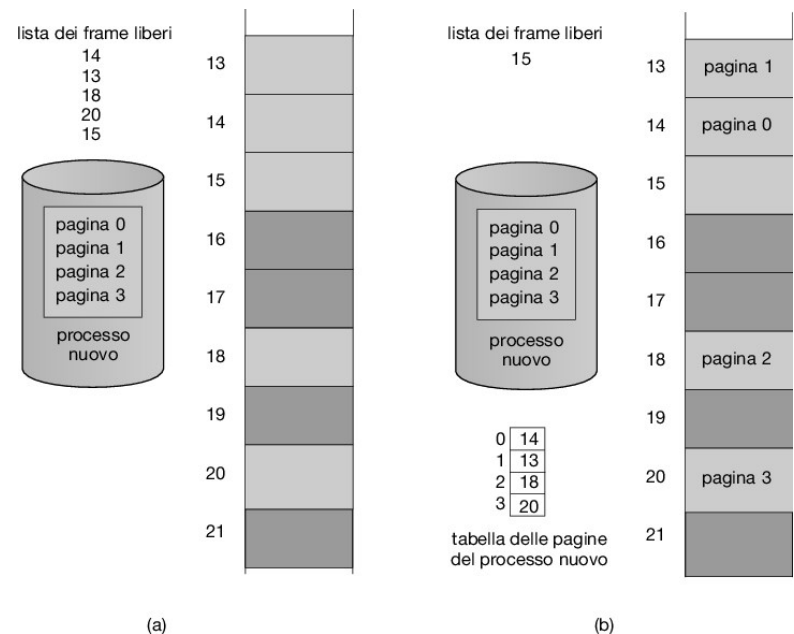


Paginazione

Un aspetto importante della paginazione è la netta distinzione tra la memoria vista dal programmatore e l'effettiva memoria fisica.

Il programmatore vede la memoria come un unico spazio contiguo, contenente solo il programma stesso

In realtà, il programma utente è sparso in una memoria fisica contenente anche altri programmi.



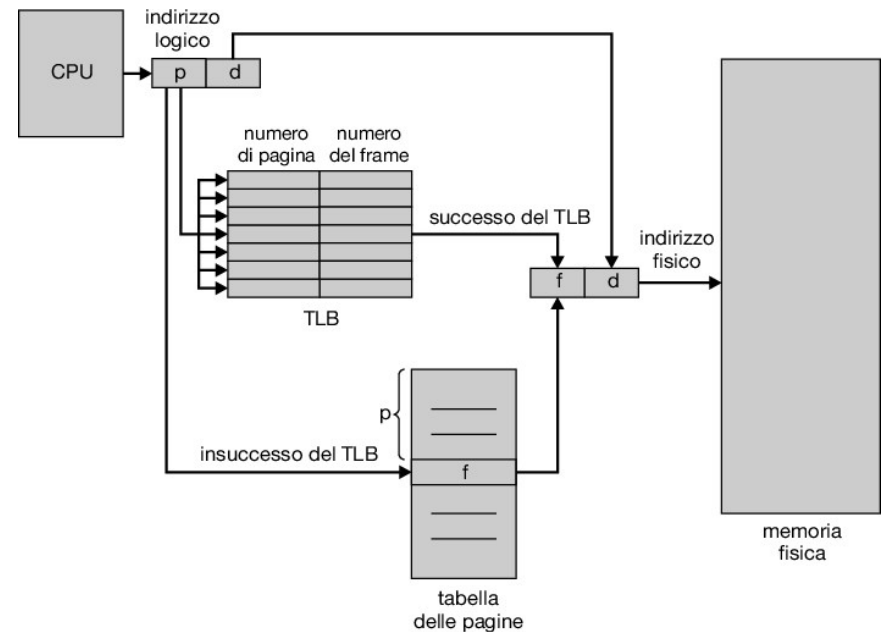
Tlb

La memorizzazione della tabella delle pagine nella memoria principale può favorire cambi di contesto più rapidi, ma può anche comportare tempi di accesso alla memoria più lenti.

La soluzione tipica a questo problema consiste nell'impiego del tlb (translation lookaside buffer).

Il tlb è:

- Una piccola cache hardware.
- Una memoria associativa ad alte prestazioni.



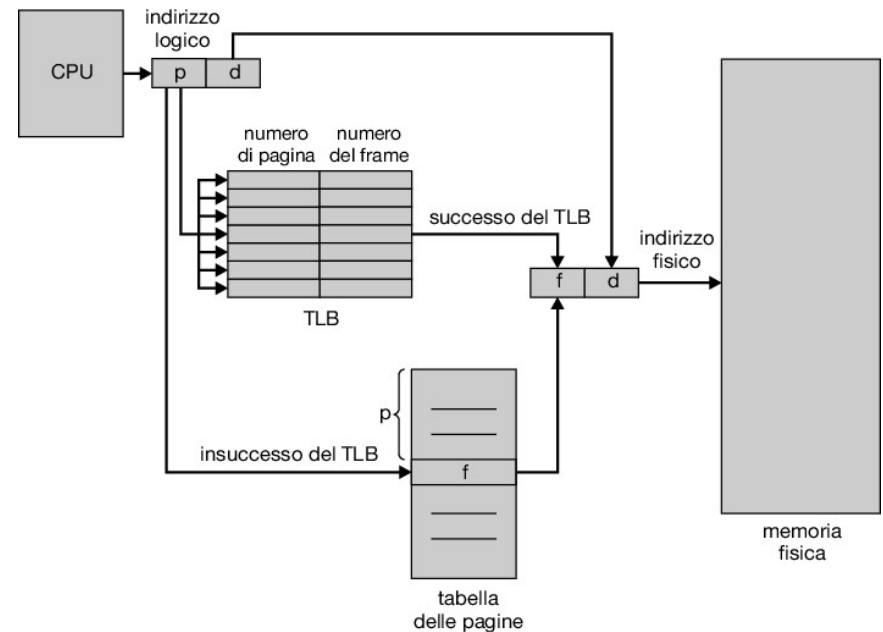
Tlb

Quando si presenta un elemento, la memoria associativa lo confronta contemporaneamente con tutte le chiavi.

Se si trova una corrispondenza, riporta il valore correlato.

La ricerca è molto rapida e in un hardware moderno è parte della pipeline delle istruzioni, quindi non induce dunque nessuna penalizzazione in termini di prestazioni.

Alcune cpu implementano tlb dati ed istruzioni separate.



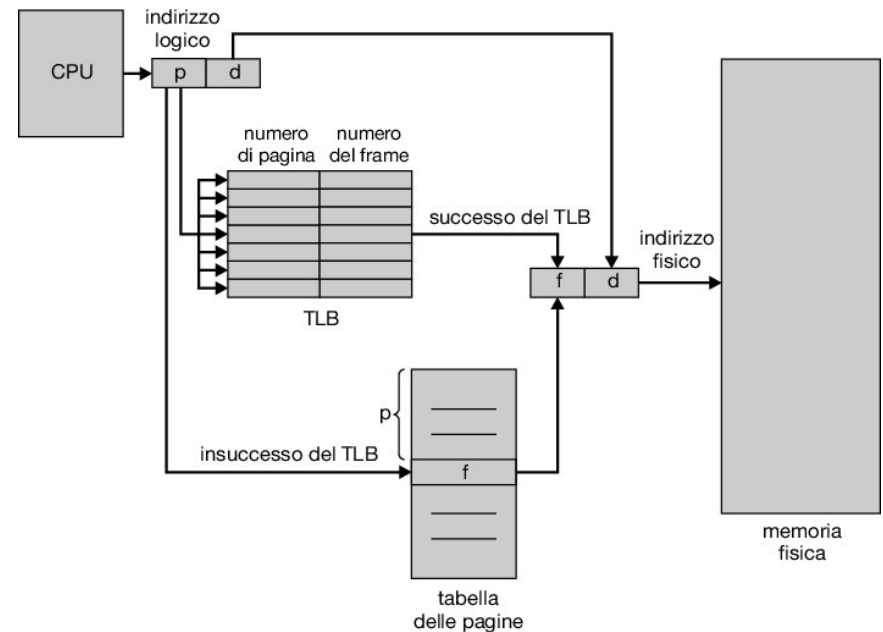
Tlb

Il tlb deve essere di dimensioni ridotte, in genere contenute tra le 32 e le 1.024 voci.

Non tutti i numeri di pagina vengono memorizzati nel tlb.

Questo avvenimento viene considerato come insuccesso del tlb e si parla di 'tlb miss'.

Se il tlb è già pieno d'elementi, occorre sceglierne uno per sostituirlo; alcuni elementi non possono essere rimossi perché sono 'vincolati' (wired down).



Struttura della tabella delle pagine

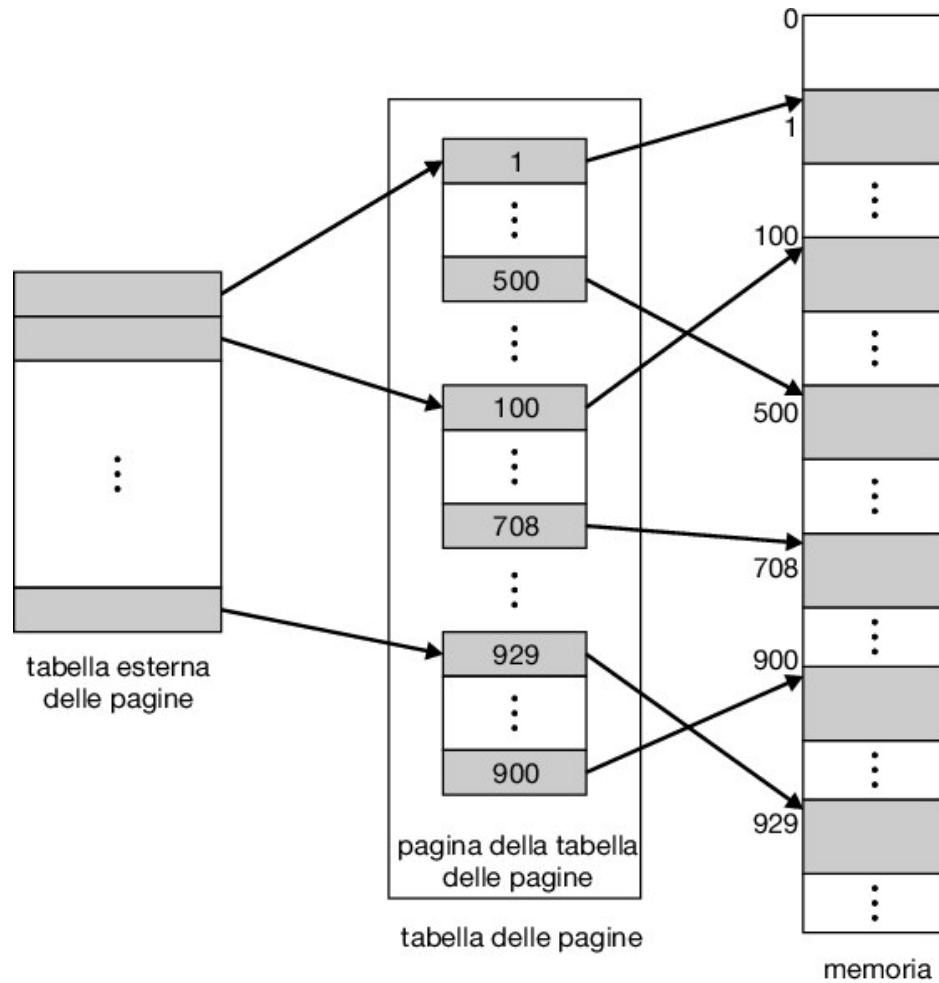
Quali tecniche abbiamo a disposizione per strutturare la tabella delle pagine?

Struttura della tabella delle pagine

Quali tecniche abbiamo a disposizione per strutturare la tabella delle pagine?

- Paginazione gerarchica.
- Tabella delle pagine di tipo hash.
- Tabella delle pagine invertita.

Paginazione gerarchica



Paginazione gerarchica

numero di pagina		offset di pagina
p_1	p_2	d
10	10	12

indirizzo logico

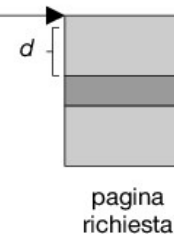
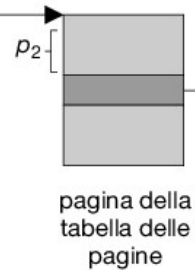
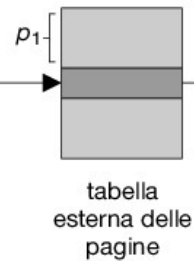
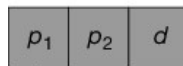


Tabella delle pagine di tipo hash

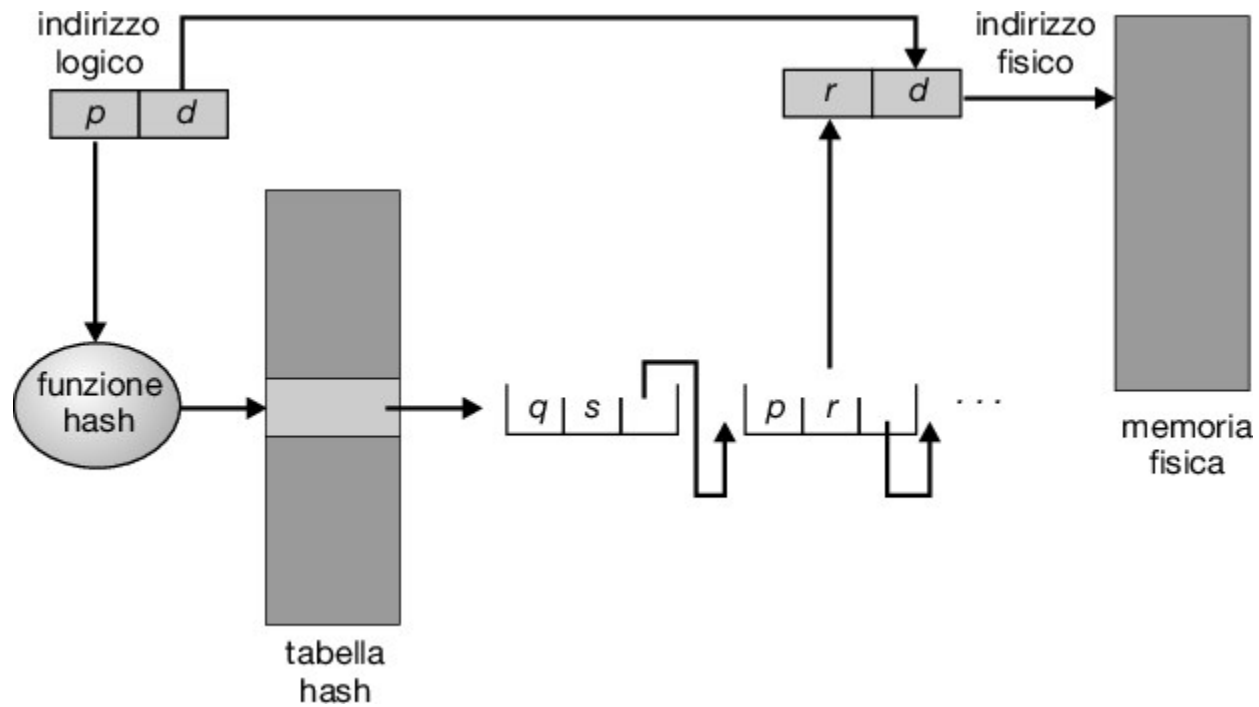


Tabella delle pagine invertita

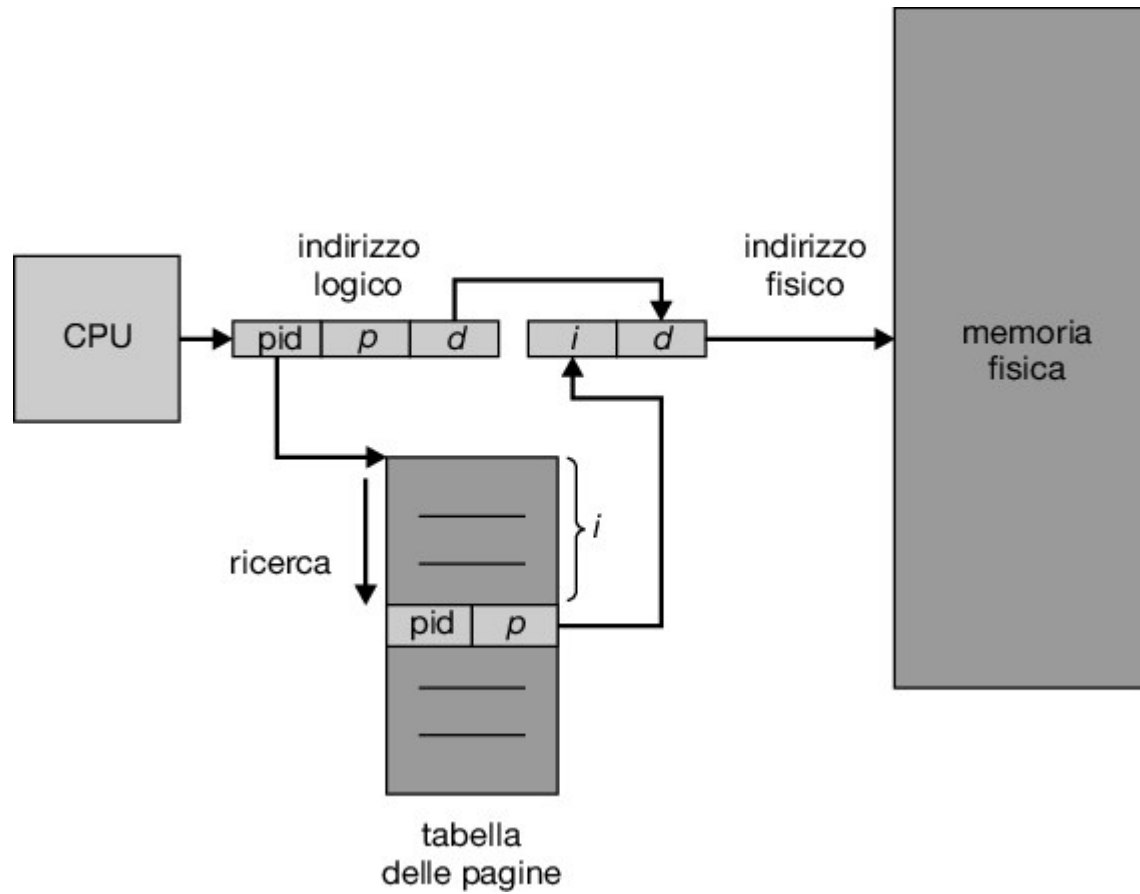
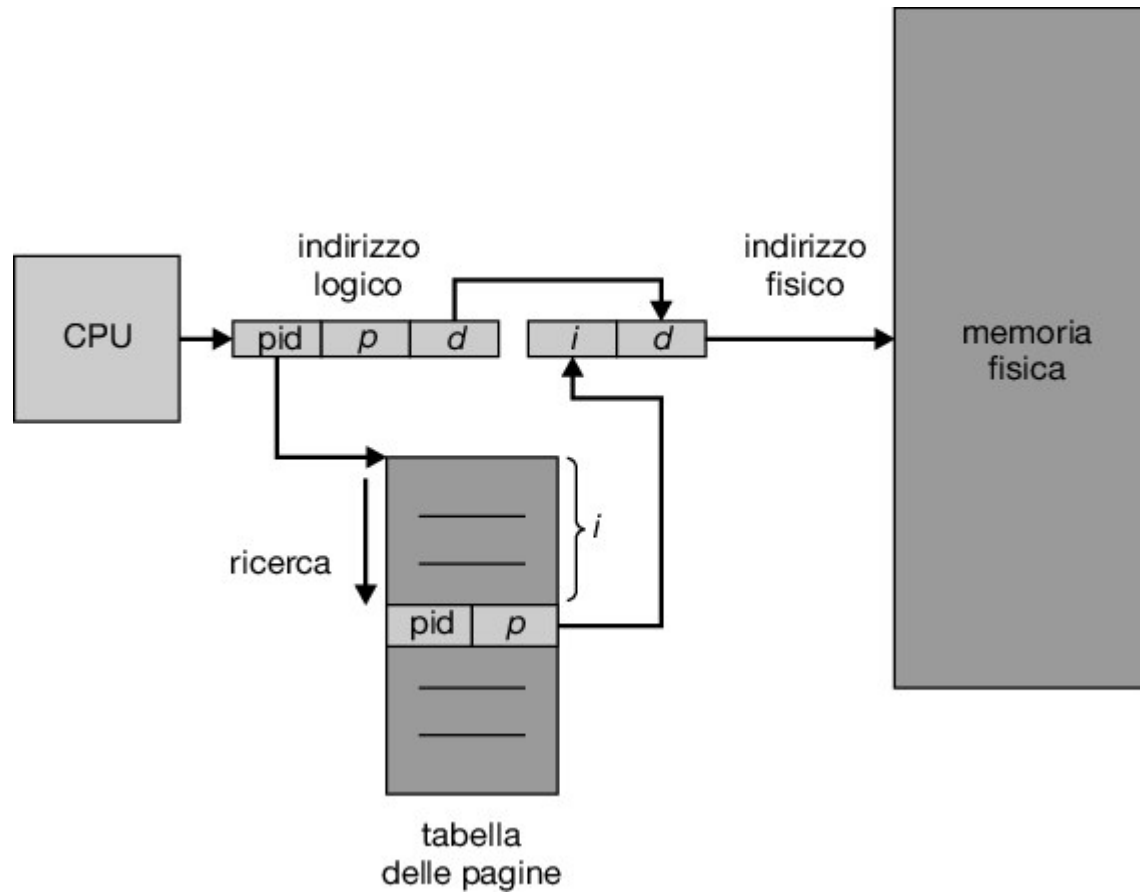


Tabella delle pagine invertita



Memoria virtuale

Qual è il concetto fondante della memoria virtuale?

Memoria virtuale

Qual è il concetto fondante della memoria virtuale?

La memoria virtuale si fonda sulla separazione della memoria logica percepita dall'utente dalla memoria fisica.



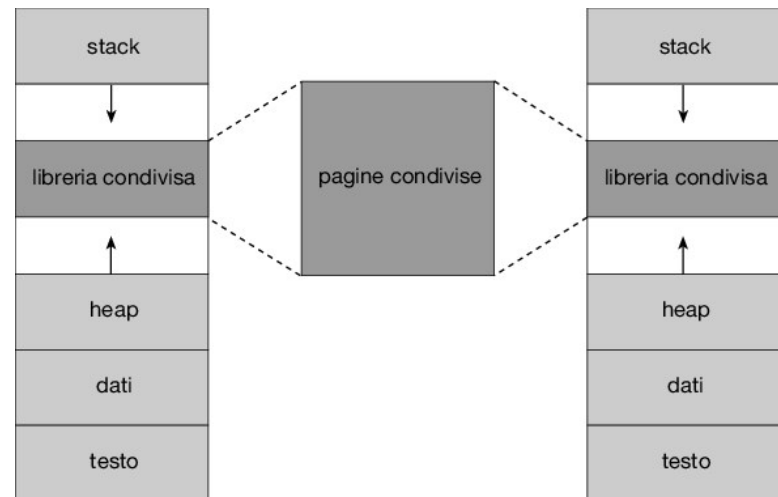
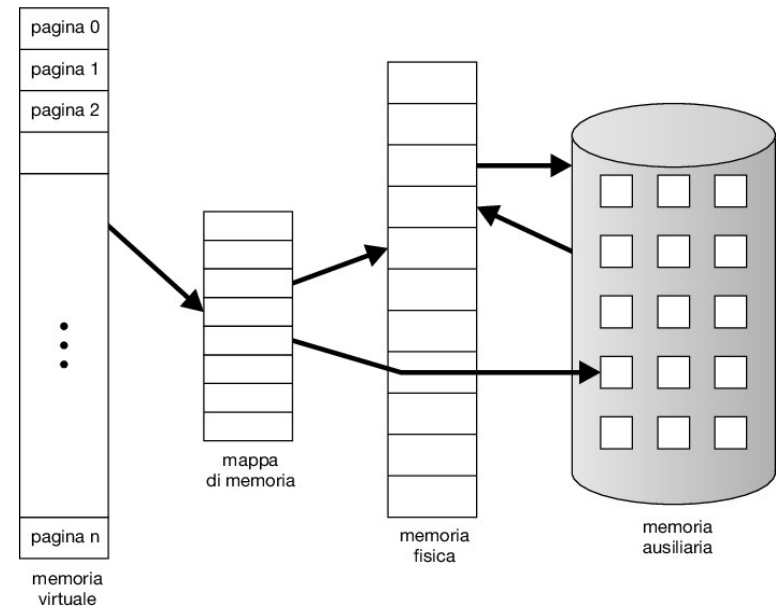
Memoria virtuale

Quali vantaggi comporta?

Memoria virtuale

Quali vantaggi comporta?

- Permette di offrire ai programmatori una memoria virtuale molto ampia, anche se la memoria fisica disponibile è più piccola.
- Facilita la programmazione, poiché il programmatore non deve preoccuparsi della quantità di memoria fisica disponibile, ma può concentrarsi sul problema da risolvere con il programma.
- Permette di condividere i file e la memoria fra due o più processi, mediante la condivisione delle pagine



Paginazione su richiesta

**Quale concetto sta alla base della
paginazione su richiesta?**

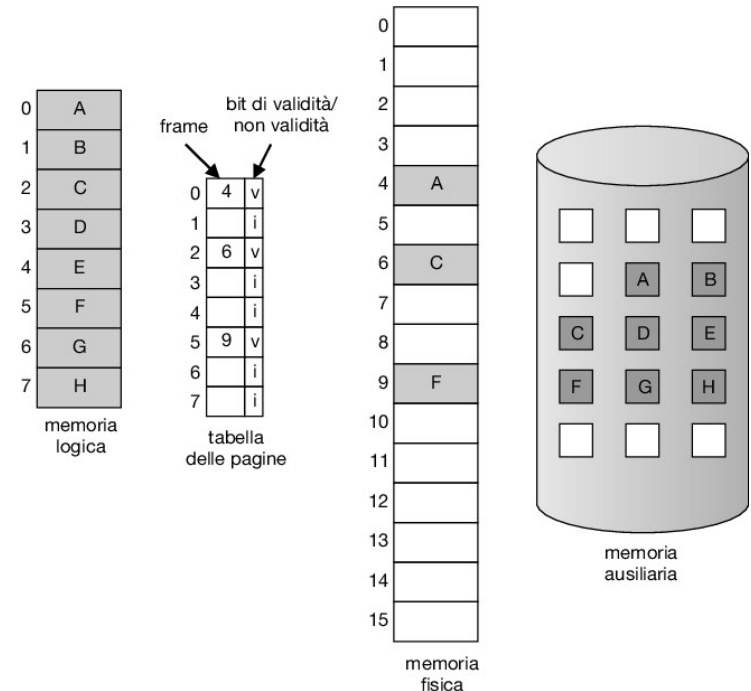
Paginazione su richiesta

Quale concetto sta alla base della paginazione su richiesta?

Caricare una pagina in memoria solo quando è necessaria, quindi mentre un processo è in esecuzione, una parte delle pagine sarà in memoria principale e una parte sarà in memoria secondaria.

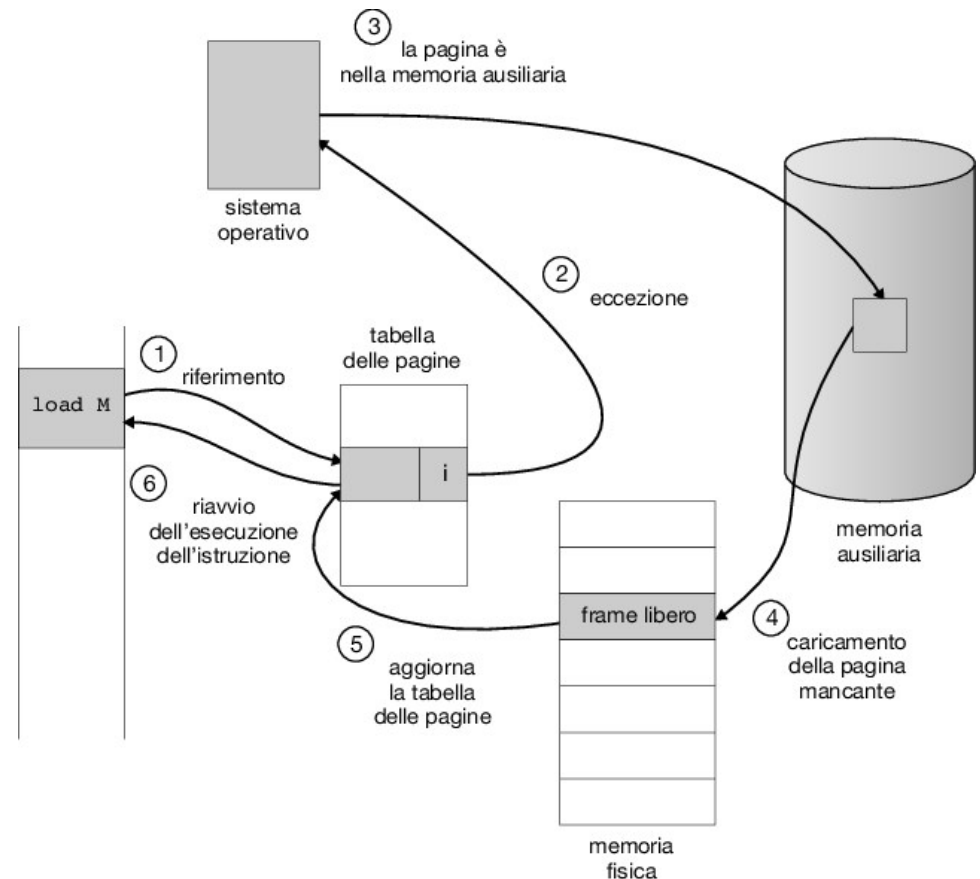
È necessario un supporto hardware per distinguere i due casi; a tale scopo si può utilizzare uno schema basato sul bit di parità.

Se un processo tenta di accedere ad una pagina contrassegnata non valida causa un'eccezione definita 'page fault'.



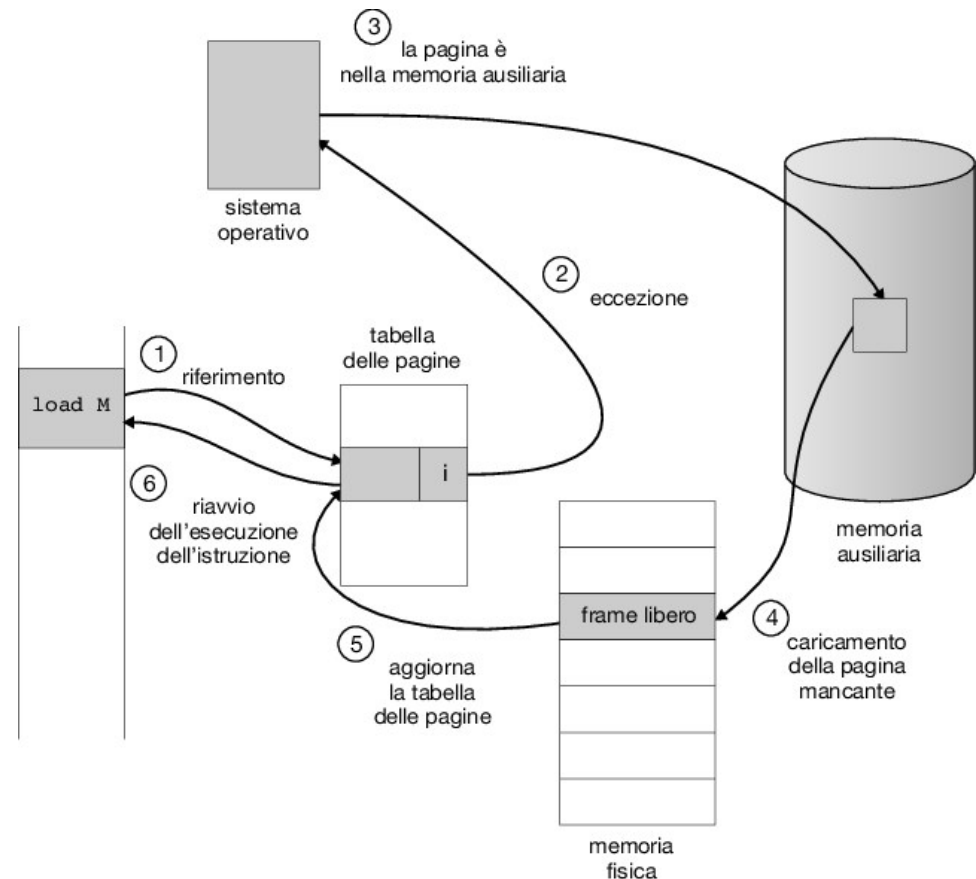
Paginazione su richiesta

Se un processo tenta di accedere ad una pagina contrassegnata non valida causa un'eccezione definita 'page fault'.



Gestione dell'eccezione del page fault

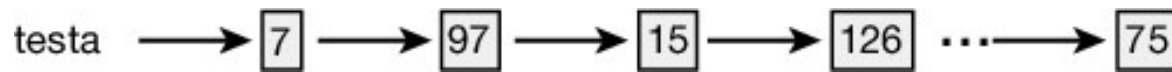
1. Controllo della tabella interna per verificare se l'accesso alla memoria è valido.
2. Terminazione del processo se l'accesso è invalido; caricamento della pagina se è valido ma non ancora in memoria.
3. Ricerca di un frame libero dalla lista dei frame liberi.
4. Trasferimento della pagina nel frame assegnato tramite operazione su disco.
5. Aggiornamento delle tabelle interne e delle pagine per indicare che la pagina è in memoria.
6. Riavvio dell'istruzione interrotta dall'eccezione.



Lista dei frame liberi

Quando si verifica un page fault, il sistema operativo deve spostare la pagina desiderata dalla memoria secondaria alla memoria principale.

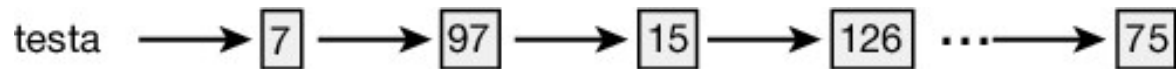
La maggior parte dei sistemi operativi mantiene una lista dei frame liberi (frame disponibili e utilizzabili per soddisfare le richieste).



Lista dei frame liberi

Quando si verifica un page fault, il sistema operativo deve spostare la pagina desiderata dalla memoria secondaria alla memoria principale.

La maggior parte dei sistemi operativi mantiene una lista dei frame liberi (frame disponibili e utilizzabili per soddisfare le richieste).



L'allocazione dei frame liberi avviene, generalmente, tramite una tecnica nota come zero-fill-on-demand (“riempimento con zeri su richiesta”).

I frame vengono “azzerati” su richiesta prima di essere allocati, cancellando così il loro precedente contenuto

Sostituzione di pagina

Cos'è la sostituzione di pagina?

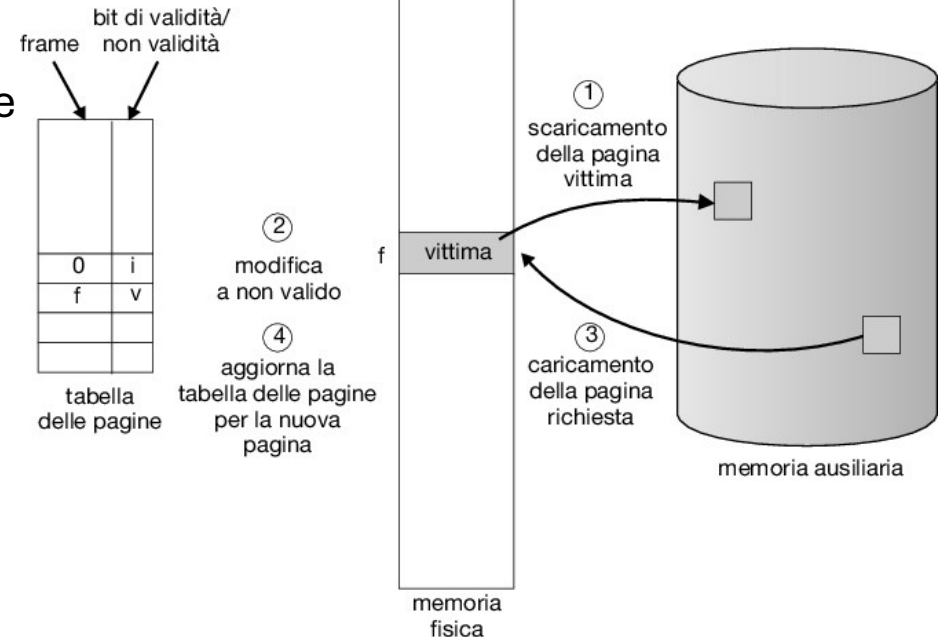
Sostituzione di pagina

Cos'è la sostituzione di pagina?

La sostituzione di pagina è una tecnica utilizzata nei sistemi operativi per gestire la memoria virtuale.

Quando un processo richiede una pagina che non è attualmente in memoria (page fault), il sistema operativo deve caricare quella pagina dalla memoria secondaria nella memoria principale.

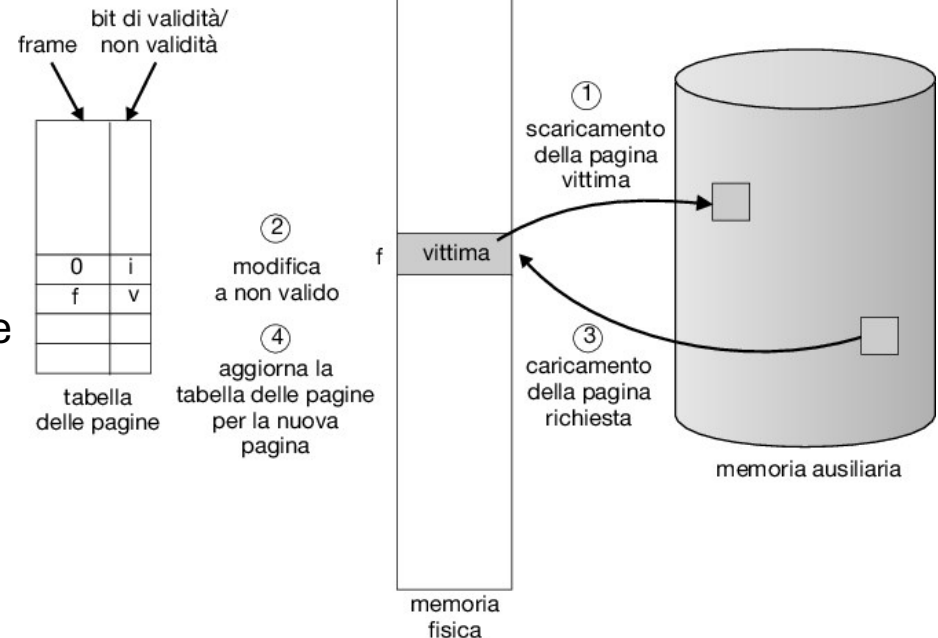
Se la memoria principale è piena, il sistema operativo deve liberare spazio sostituendo una delle pagine attualmente in memoria con quella nuova.



Gestione sostituzione di pagina

L'algoritmo di gestione dell'eccezione del page fault viene modificato per seguire i seguenti passaggi:

1. Individuazione della locazione sul disco della pagina richiesta.
2. Ricerca di un frame libero; se esiste lo si usa, altrimenti si sceglie un 'frame vittima' tramite un algoritmo e si scrive la pagina 'vittima' sul disco.
3. Scrittura della pagina richiesta nel frame liberato.
4. Ripresa del processo dal punto del page fault.



Algoritmi di sostituzione delle pagine

Quali algoritmi abbiamo a disposizione?

Algoritmi di sostituzione delle pagine

Quali algoritmi abbiamo a disposizione?

- Sostituzione FIFO.
- Sostituzione ottimale.
- Sostituzione LRU (least recently used).
- Sostituzione LFU (least frequently used).
- Sostituzione MFU (most frequently used).
- Eventuali variazioni degli algoritmi sopra citati.

Valutazione di un algoritmo

Quali algoritmi abbiamo a disposizione?

- Un algoritmo si valuta effettuandone l'esecuzione su una particolare successione di riferimenti alla memoria e calcolando il numero di page fault.
- La successione dei riferimenti alla memoria è detta, 'successione dei riferimenti'.