

## Gestione della memoria

---

## Gestione della memoria

---

- In un sistema multiprogrammato il numero di processi è  $>$  del numero di processori, ciò implica:
  - unità di elaborazione virtuale
  - memoria virtuale
- Un gestore di risorsa, deve creare le risorse virtuali e gestire le risorse reali.
- Memoria virtuale: insieme di blocchi su memoria di massa di dimensioni sufficienti a contenere le informazioni di un processo, quando questo non è allocato in memoria fisica.

## Gestore della risorsa reale

- Mantenere aggiornato lo stato della risorsa:
  - nel caso della memoria deve registrare quale parte della memoria è libera, occupata e da chi.
- Allocare e revocare la risorsa:
  - nel caso della memoria deve eseguire le procedure di swap-in e swap-out.
- Decidere per quanto tempo e a chi allocare la risorsa:

## Differenze tra gestione della CPU e della memoria

- Parti diverse di memoria possono essere allocate a diversi processi contemporaneamente.
  - è necessaria una struttura dati più complessa di quella usata per la CPU.
- Lo stesso processo può essere allocato in parti diverse della memoria in tempi diversi.
  - necessità di un meccanismo di rilocazione dei programmi.
- La memoria può essere allocata sia dinamicamente sia staticamente.
  - diverse tecniche di allocazione.

## Compiti del gestore della memoria

- Tenere traccia di quali parti della memoria sono libere e quali occupate.
- Allocare memoria ai processi che ne hanno bisogno.
- Deallocare la memoria di un processo quando questi non ne fa più uso.
- Gestire le operazioni di trasferimento tra memoria di massa e memoria centrale:
  - swap-in
  - swap-out

## Memoria virtuale: tipi di indirizzi

- Indirizzi simbolici:
  - programma sorgente.
- Indirizzi logici (virtuali)
  - dopo la fase di link
- Indirizzi fisici
  - indirizzi per l'esecuzione, necessità di rilocare gli indirizzi virtuali in indirizzi fisici quando un programma viene caricato in un indirizzo fisico diverso da 0.

## Rilocalazione statica

- Effettuata dal caricatore rilocante in fase di caricamento.
- Le informazioni dipendenti dalla locazione vengono modificate sommandoci l'indirizzo iniziale di caricamento (selettore di caricamento).
- Un programma una volta caricato e rilocato in memoria fisica, non può essere spostato in un'altra area.

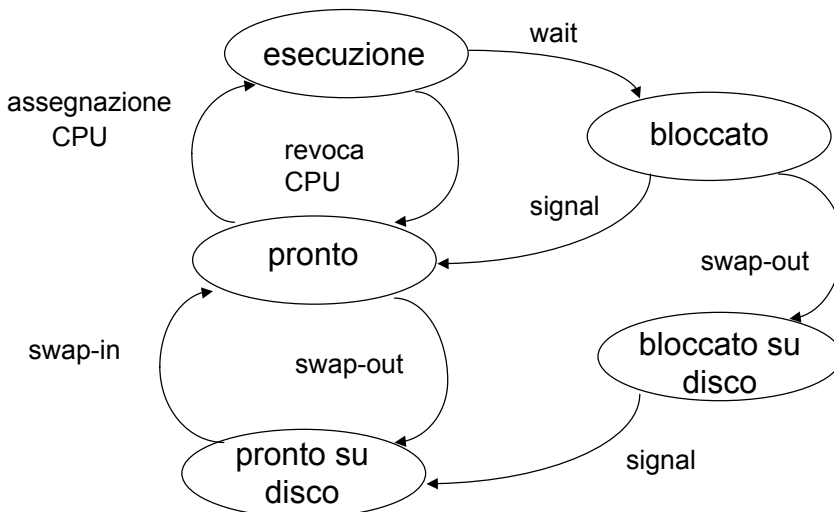
## Rilocalazione dinamica

- Necessità di un meccanismo che traduca da indirizzi virtuali a indirizzi fisici:
  - $y=f(x)$   
x indirizzo virtuale, y corrispondente indirizzo fisico.
- Un programma può essere caricato in un'area di memoria fisica e poi successivamente spostato in un'altra area.
- Modificando le informazioni contenute nel meccanismo hardware (MMU) che realizza la funzione di rilocalazione, è possibile spostare un programma in aree di memoria differenti.

# Tecniche di allocazione della memoria

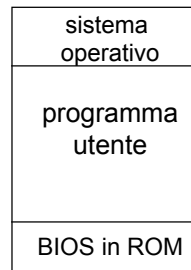
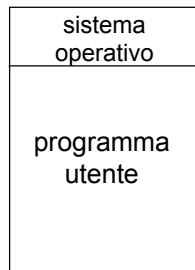
- Tecniche che fanno riferimento a meccanismi di rilocalizzazione statica della memoria:
  - ad un processo viene assegnata memoria quando viene creato e revocata quando termina.
- Tecniche supportate da meccanismi di rilocalizzazione dinamica:
  - ad un processo viene allocata e revocata memoria più volte durante la sua "vita".

# Allocazione dinamica

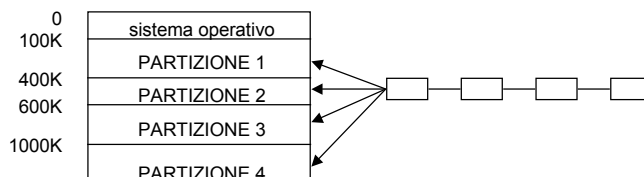
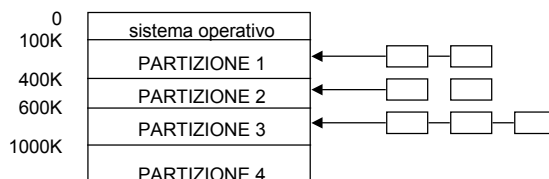
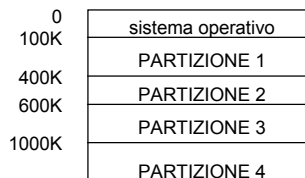


# Allocazione della memoria in sistemi monoprogrammati

- Hardware di supporto: registro limite di protezione.
- Per programmi di dimensione maggiore della memoria fisica: necessità di adottare tecniche di caricamento parziale (overlay).



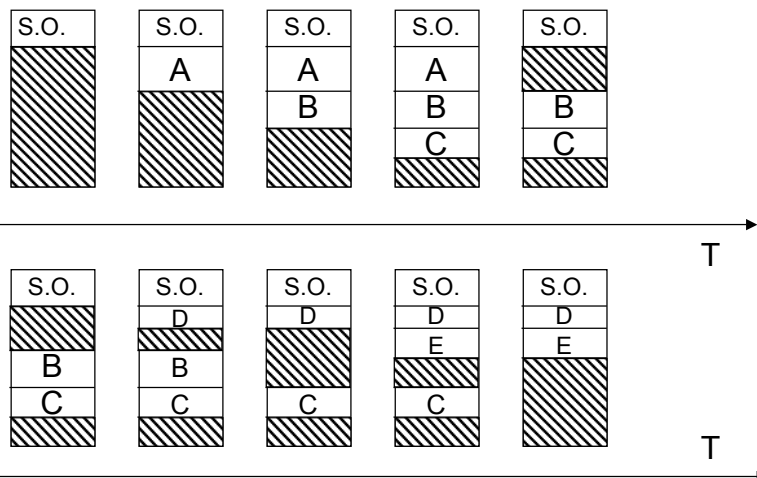
# Tecnica delle partizioni fisse



## Tecnica delle partizioni fisse

- Aspetti positivi: semplicità
- Inconvenienti:
  - frammentazione interna
  - necessità di tecniche di overlay
- Meccanismi hardware:
  - meccanismo di protezione:
    - registri di frontiera
    - chiavi di protezione

## Tecnica delle partizioni variabili





## Tecnica delle partizioni variabili

---

- **Aspetti positivi:**
  - migliore sfruttamento della memoria rispetto alle partizioni fisse.
- **Inconvenienti:**
  - frammentazione esterna
  - necessità di tecniche di overlay.
  - maggiore overhead (gestione delle partizioni libere).
- **Meccanismi hardware:**
  - meccanismo di protezione
    - registri di frontiera
    - chiavi di protezione



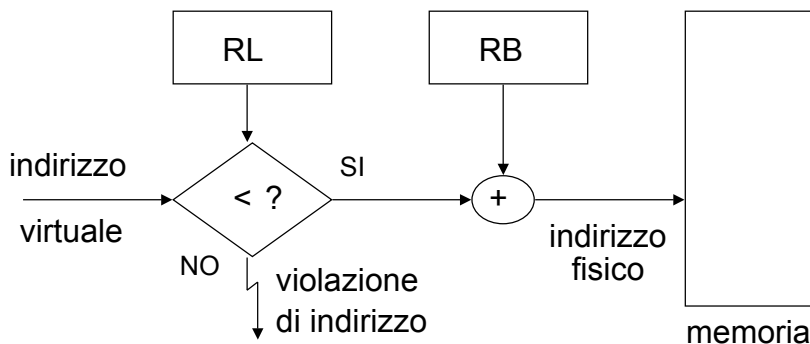
## Strategie di allocazione

---

- first fit
- best fit
- worst fit

## Tecnica delle partizioni rilocabili

- Rilocazione dinamica mediante una coppia di registri: registro base RB e registro limite RL.
- $Y=F(X)=X+RB$



## Strategie di allocazione

- Stesse strategie di allocazione delle partizioni variabili.
- SWAP-IN e SWAP-OUT
- Strategia di compattamento per eliminare la frammentazione.
- Necessità di due coppie di registri base/limite per consentire la condivisione di codice tra processi (l'area dati è separata).
- Notevole overhead per gestire la frammentazione.

## Tecnica della paginazione

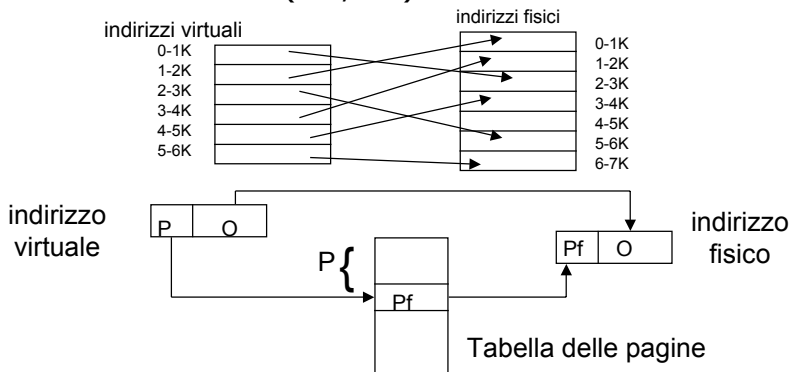
- Meccanismo di rilocazione dinamica che consente di allocare uno spazio virtuale contiguo in aree di memoria fisica non necessariamente contigue.
- Ciò si ottiene suddividendo sia lo spazio virtuale sia lo spazio fisico in parti (pagine) di dimensioni fisse ed allocando ad ogni processo un numero di pagine fisiche esattamente uguale al numero di pagine virtuali del processo.

## Indirizzo virtuale

- Supponiamo di avere pagine di dimensioni di:  $S=1024$  byte
- L'indirizzo virtuale  $X$  può essere espresso come:  $X=\text{coppia}(P,O)$ 
  - $P$ : pagina a cui appartiene  $X$
  - $O$ : offset di  $X$  nell'ambito della pagina.
- Esempio:  $X=2051$ 
  - $P=X \text{ DIV } S = 2051 \text{ DIV } 1024 = 2$
  - $O=X \text{ MOD } S = 2051 \text{ MOD } 1024 = 3$

# Traduzione degli indirizzi

- Esempio: sapendo che la pagina logica N. 2 è stata caricata nella pagina fisica N. 5:
  - l'indirizzo virtuale  $X=2051 = (P:2,O:3)$  corrisponde a
  - indirizzo fisico  $Y=(P:5,O:3) = 5*1024+3$



# Demand paging

- Caricamento parziale delle pagine di un processo
- Se l'indirizzo virtuale da tradurre corrisponde ad una pagina non presente in memoria si genera una interruzione (page fault).
- Gestione delle interruzioni per caricare la pagina richiesta da disco.

pagina logica	indice della pagina fisica	bit di presenza
0	2	1
1	0	1
2	-	0
3	1	1
4	-	0
5	6	1

→

## Caratteristiche della paginazione

- Paginazione su domanda: allocazione dinamica.
- Possibilità di eseguire processi con memoria virtuale maggiore della memoria fisica.
- Il problema della frammentazione viene eliminato automaticamente.
- Protezione automatica fra spazi virtuali diversi.

## Caratteristiche della paginazione

- Tabella della memoria: tabella con tante entry (righe) quante sono le pagine fisiche e ogni entry indica se la pagina è libera o occupata e da chi.
- Nel descrittore di un processo ci deve essere l'indirizzo della sua tabella delle pagine (in memoria e su disco).
- Quando si verifica una commutazione di contesto deve essere commutata anche la tabella delle pagine con cui effettuare la traduzione degli indirizzi.

## Procedura di page fault

- Interruzione di page fault -> ingresso nel nucleo.
- Salvataggio dello stato del processo.
- Sospensione del processo che ha provocato il page fault in attesa del trasferimento.
- Individuazione del numero della pagina richiesta.
- Verifica dell'esistenza di una pagina fisica libera; altrimenti selezione di una pagina da rimpiazzare.
- Se la pagina da rimpiazzare è stata modificata bisogna salvarla su disco.
- Caricamento in memoria della pagina richiesta.

## Procedura di page fault

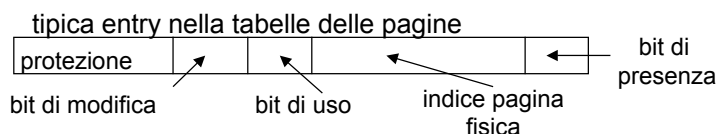
- Aggiornamento della tabella delle pagine all'arrivo dell'interruzione di fine caricamento.
- Aggiornamento del registro puntatore all'istruzione corrente (IP) per poter rieseguire l'istruzione che ha provocato un page fault.
- Il processo sospeso può essere messo nello stato di pronto.

## Aspetti critici della paginazione

- Necessità di minimizzare il tempo di traduzione degli indirizzi da virtuali a fisici (parte della tabella è mantenuta in una piccola memoria associativa nella MMU).
- Necessità di risolvere il problema dell'allocazione in memoria della tabella delle pagine (paginazione a più livelli).
- Necessità di realizzare un algoritmo di rimpiazzamento delle pagine tale da minimizzare il numero dei page fault (problema del trashing)

## Algoritmi di rimpiazzamento delle pagine

- FIFO
  - la prima pagina caricata in memoria è quella che viene scaricata in caso di necessità.
- Least recently used:
  - viene scaricata la pagina usata meno di recente.
  - è complicato capire da quanto tempo una pagina è in memoria.
- Clock algorithm:
  - seleziono come vittima la prima pagina con il bit di uso a 0; se trovo una pagina con il bit di uso ad uno lo azzerò.



## Frammentazione interna e tabella della memoria

---

- Normalmente un processo non occupa un numero intero di pagine virtuali; l'ultima pagina è mediamente occupata per metà.
- Frammentazione interna= parte inutilizzata della memoria fisica allocata all'ultima pagina virtuale.
- La tabella della memoria è costituita da un un array con un numero di elementi pari al numero di pagine fisiche.
- Ogni elemento contiene lo stato di allocazione della corrispondente pagina fisica e informazioni utili per realizzare l'algoritmo di sostituzione.

## Protezione e dimensione delle pagine

---

- La protezione dello spazio fisico allocato ad un processo è garantita dal meccanismo di rilocazione.
- Attributi di protezione associata alle pagine:
  - distinzione tra pagine read only e read write.
- La frammentazione interna aumenta aumentando la dimensione delle pagine.
- La dimensione della tabella delle pagine aumenta diminuendo la dimensione delle pagine.
- Valori tipici di dimensioni di pagine sono compresi tra 512 e 4K

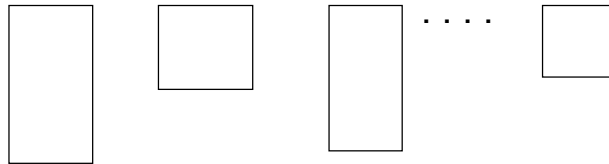
## Tecnica della segmentazione

- Il modo più naturale per un programmatore di concepire la memoria, non è quello di vedere una serie di locazioni contigue, bensì quello di immaginare la memoria divisa in una serie di segmenti, indipendenti tra di loro, contenente ognuno una parte logica del programma: funzioni, strutture dati, stack etc.

main program

struttura dati

stack



## Spazio virtuale segmentato

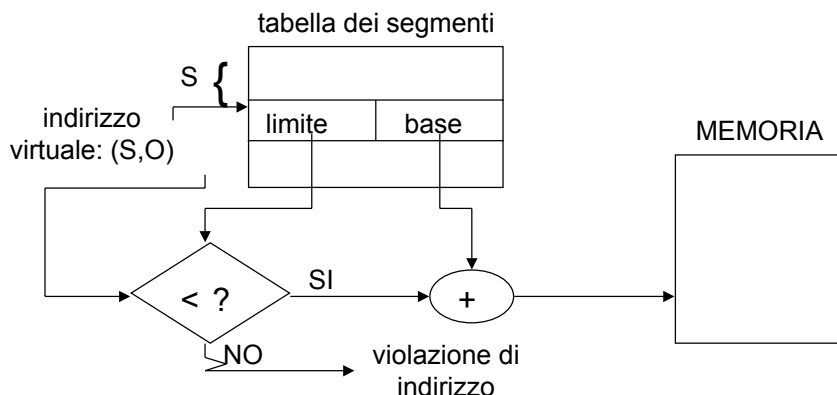
- Ogni segmento è un sottospazio lineare contiguo di locazioni comprese tra zero e un valore massimo.
- Ogni segmento è indipendente dagli altri.
- Differenti segmenti possono avere differenti lunghezze.
- Un segmento può avere dimensioni variabili all'interno delle sue dimensioni massime.
- Indirizzo composto dal numero di segmento e dallo scostamento all'interno del segmento.
- Per ogni segmento possono essere specificati attributi di protezione.
- L'ultimo indirizzo di un segmento non è consecutivo al primo indirizzo del segmento successivo.

## Spazio virtuale segmentato

- In un sistema segmentato lo spazio virtuale è ancora creato in fase di linking.
- Il compilatore crea un segmento separato per ogni componente del programma: (main, stack, etc.)
- Il linker assegna un diverso valore numerico ad ogni segmento (a partire da zero) ed effettua il collegamento traducendo i riferimenti intersegmento.

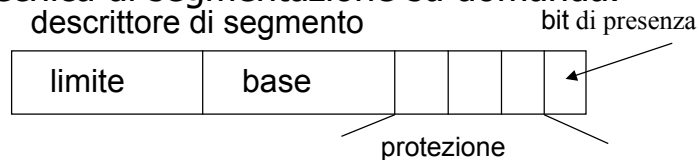
## Allocazione di memoria

- La rilocalizzazione in memoria di ogni segmento avviene tramite una coppia di registri base - limite.



## Tecnica della segmentazione

- Ogni elemento della tabella dei segmenti (coppia base-limite) è nota con il nome di descrittore di segmento.
- Due o più processi possono condividere un segmento rientrante (che non si automodifica).
- Attributi di protezione possono essere associati ad ogni segmento.
- Mediante un bit di presenza è possibile realizzare una tecnica di segmentazione su domanda.



## Esempio: Intel 80286

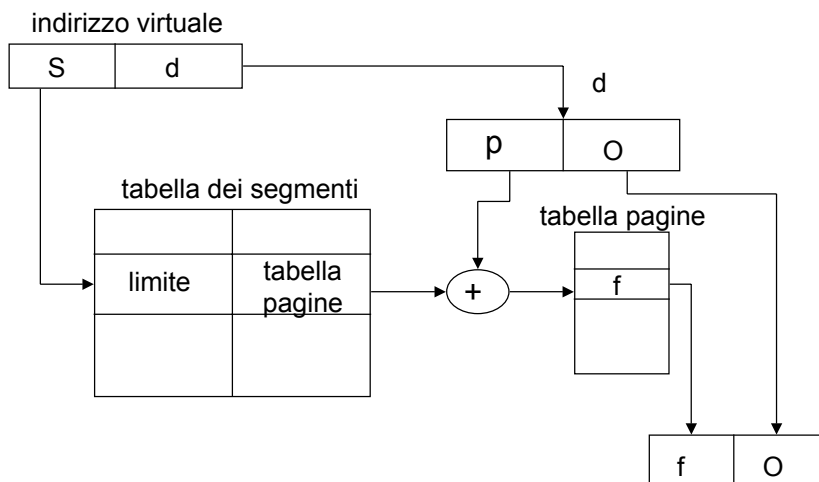
- 16384 ( $2^{14}$ ) segmenti di dimensioni variabili fino ad un massimo di 64KB. Lo spazio virtuale può quindi contenere fino ad un massimo di 1 GB.
- I segmenti sono divisi in due gruppi di 8192 segmenti ciascuno (un sottospazio globale di sistema ed uno locale proprio ad ogni processo).

	15	8	7	0
1	limite (0-15)			0
3	base (0-15)			2
5	controllo	base (16-23)		4
7	non usati			6

## Segmentazione paginata

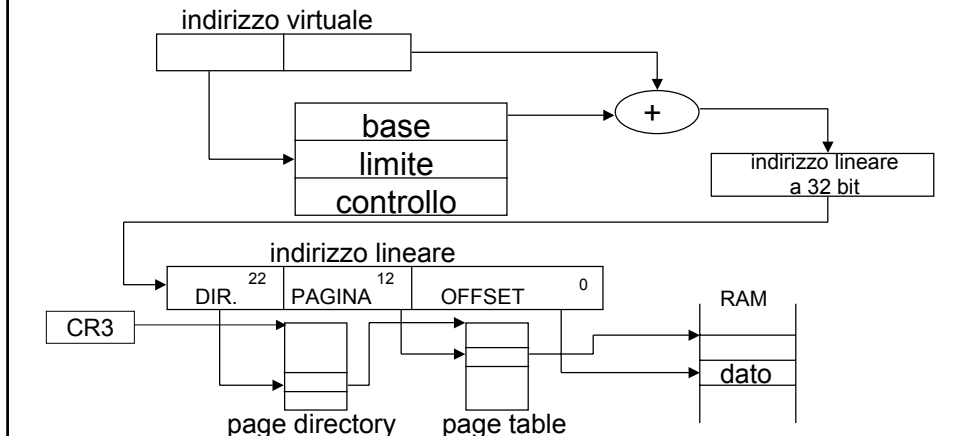
- La tecnica più sofisticata di gestione della memoria è costituita dall'unione della paginazione e della segmentazione.
- La segmentazione consente di ottenere tutti i vantaggi di una memoria virtuale non contigua.
- La paginazione viene applicata per ridurre ulteriormente la frammentazione della memoria fisica.

## Segmentazione paginata



## Esempio: Intel 80386

- Descrittore di segmento analogo al 286, cambia leggermente il formato.
- I segmenti possono contenere fino a 4GB.



## Unix: gestione della memoria

- Nelle versioni più recenti di Unix la memoria viene allocata con la tecnica della paginazione su domanda.
- Le pagine fisiche libere sono mantenute in una lista concatenata (core map).
- Un processo di sistema (processo N. 2 page daemon) viene attivato periodicamente per verificare che ci sia un numero minimo di pagine libere.
- Se le pagine libere diminuiscono il page daemon esegue il rimpiazzamento con strategia clock algorithm.



## Unix: gestione della memoria

- La strategia di swap viene realizzata dal processo N. 0 (swapper).
- Lo swapper viene attivato ogni 4 secondi per vedere se c'è una pagina da scaricare su disco o da caricare in memoria.
- Un processo è candidato ad essere scaricato se è sospeso o se è in memoria da molto tempo o se è di notevoli dimensioni.
- Per evitare il fenomeno di trashing un processo deve stare in memoria almeno un tempo minimo.
- Se si verifica una situazione di sovraccarico un processo può essere completamente scaricato.