

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Primo Livello

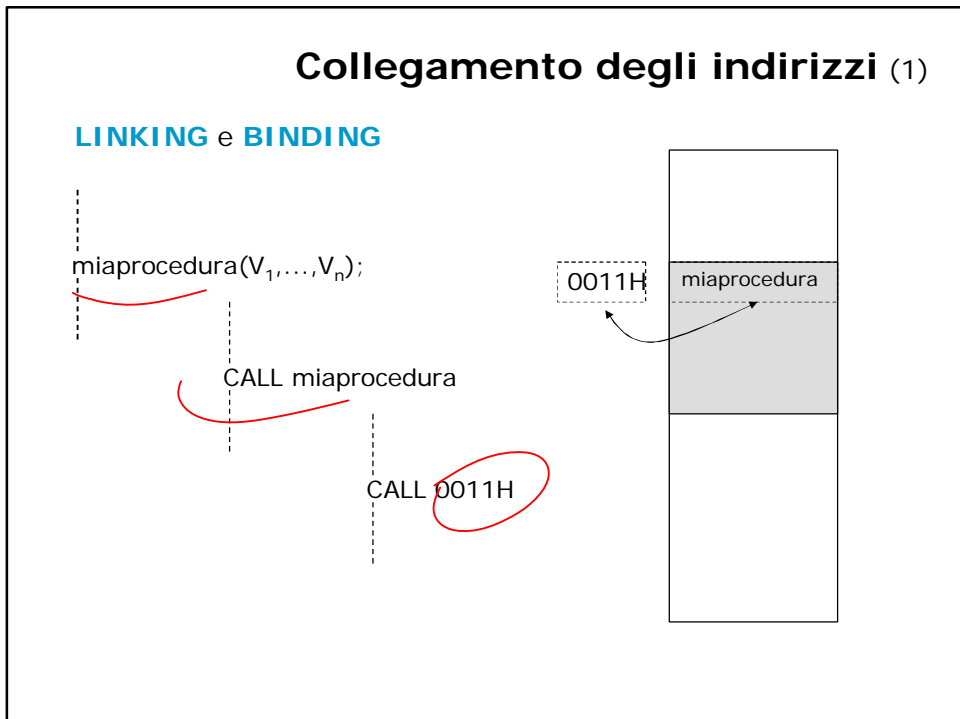
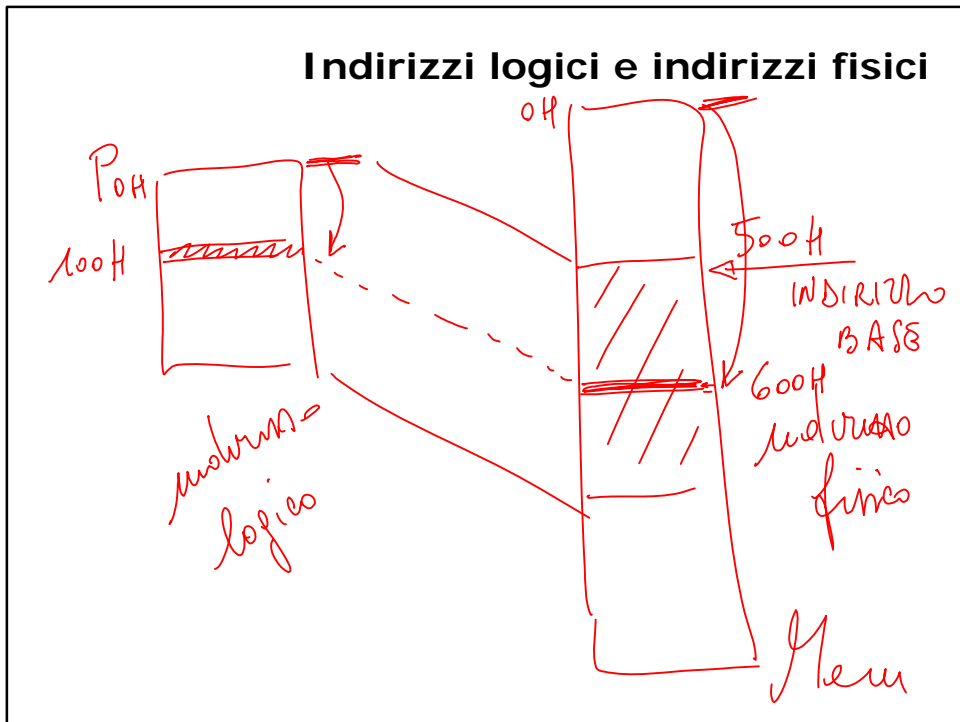
Lezione 1 – Indirizzamento in memoria centrale

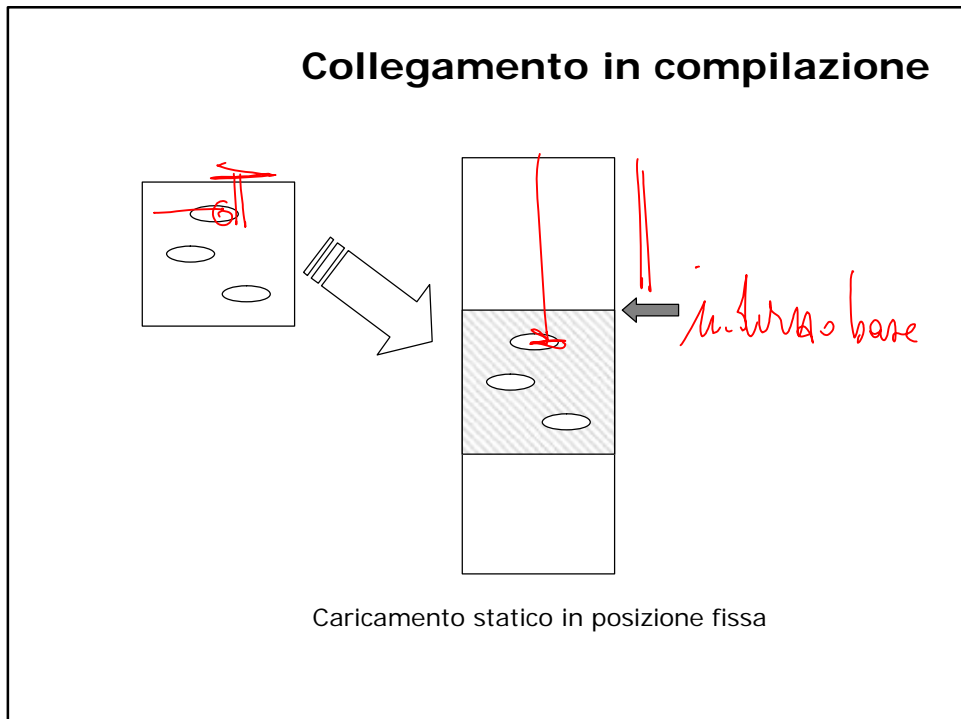
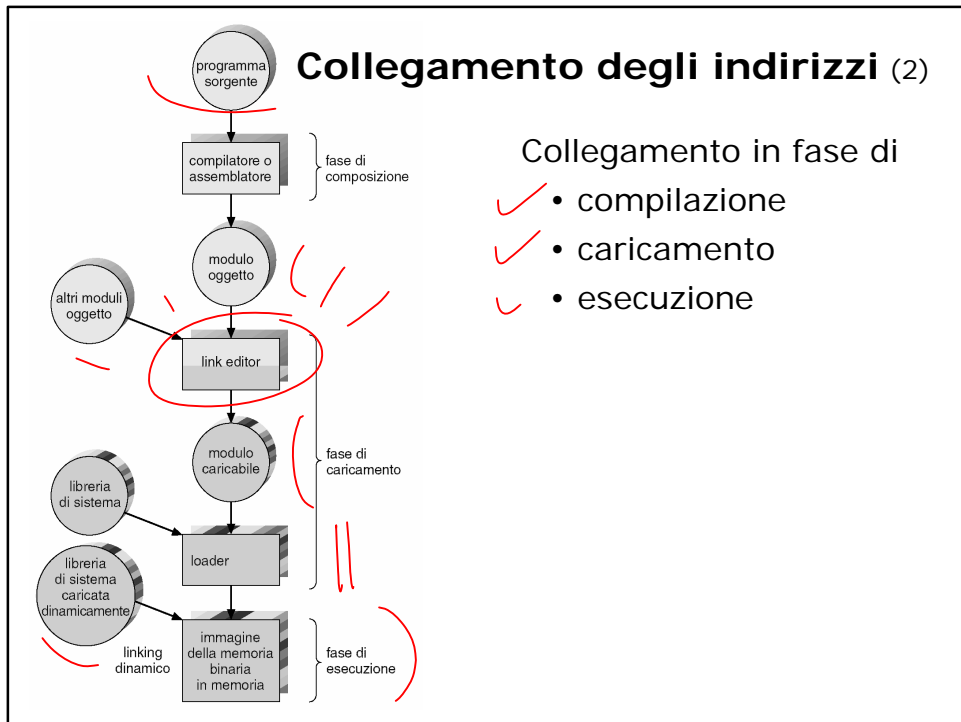
Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

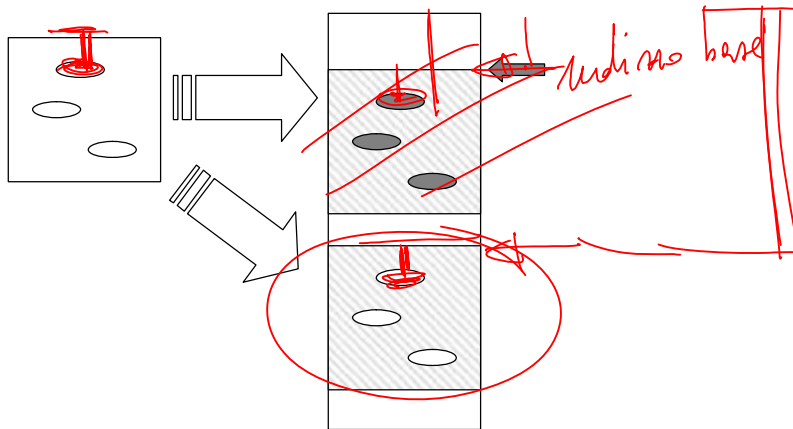
Sommario

- Indirizzi logici e indirizzi fisici
- Collegamento degli indirizzi logici agli indirizzi fisici
 - collegamento in compilazione
 - collegamento in caricamento
 - collegamento in esecuzione



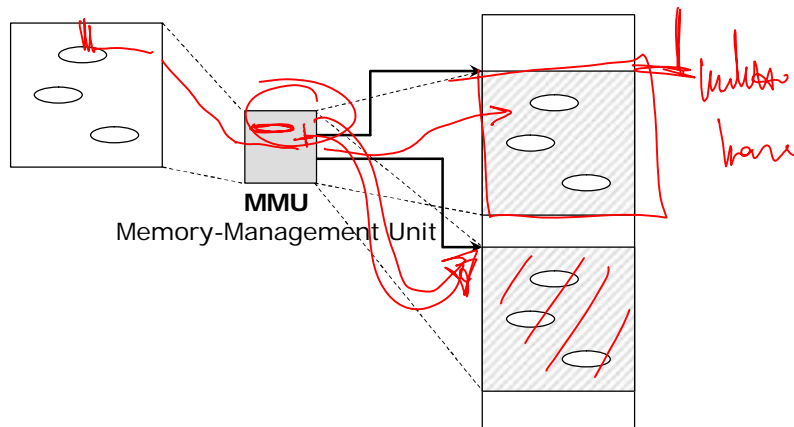


Collegamento in caricamento



Caricamento statico con rilocazione del codice durante il caricamento

Collegamento in esecuzione



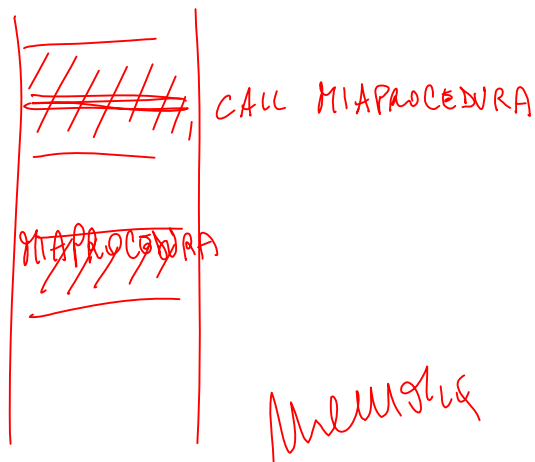
Caricamento statico con rilocazione del codice in esecuzione

Caricamento

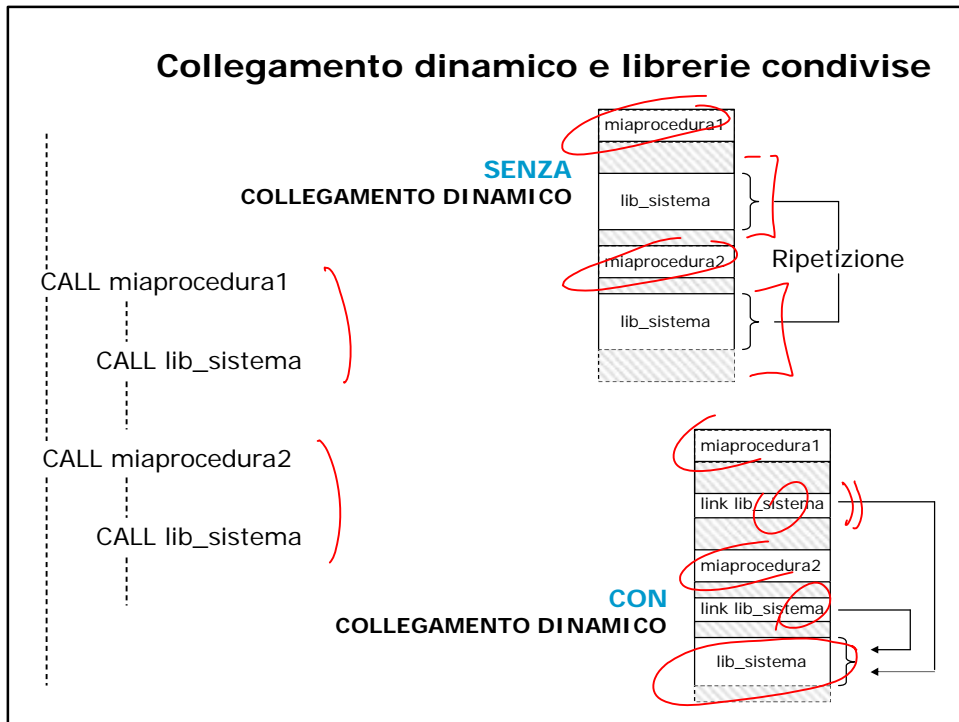
Perché occupare memoria caricando tutto il programma?

- Caricamento **statico**
- Caricamento **dinamico**

Collegamento in esecuzione



Caricamento dinamico



In sintesi

- Indirizzi fisici e indirizzi logici
- Collegamento degli indirizzi logici agli indirizzi fisici
 - collegamento in compilazione
 - collegamento in caricamento
 - collegamento in esecuzione

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Primo Livello

Lezione 2 – Partizionamento

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica del partizionamento della memoria centrale
- Obiettivi
- Partizioni
 - fisse (Statiche)
 - variabili (Dinamiche)
- Caratteristiche

Allocazione della memoria centrale

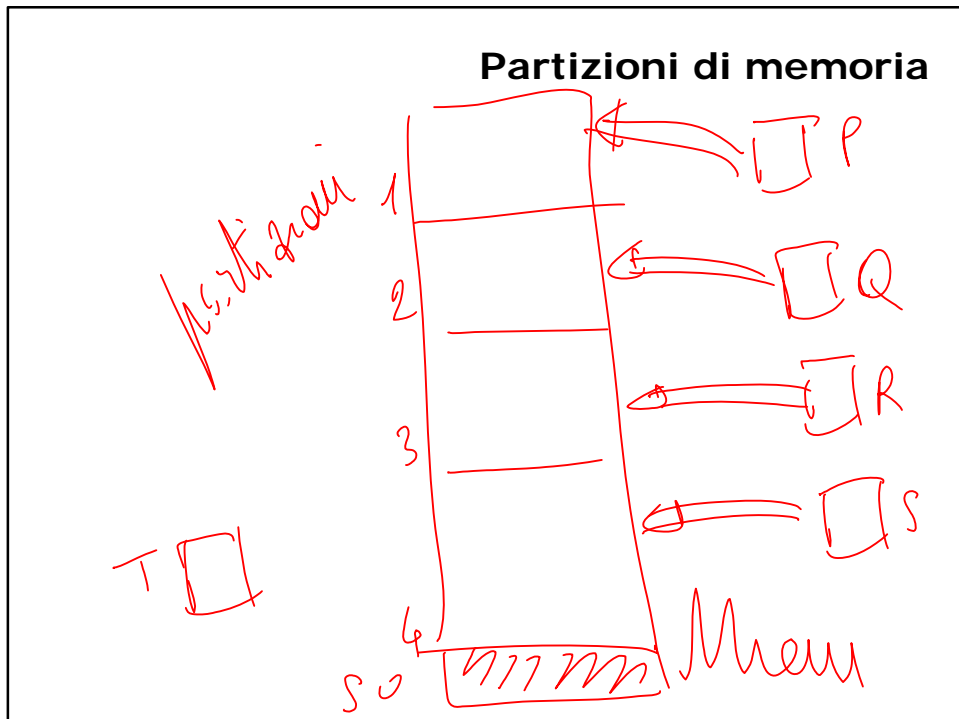
- Spazio di indirizzamento del processore e memoria centrale fisica installata
- La memoria centrale fisica ospita:
 - Sistema operativo
 - Processi utente
- Accessibilità della memoria centrale fisica da parte dei processi
- Lettura e scrittura della memoria centrale fisica da parte dei processi

Problema

- Realizzare la multiprogrammazione per supportare multitasking

Obiettivi

- Ripartire la memoria centrale fisica tra i processi ivi caricati (incluso il sistema operativo) per realizzare sistemi multiprogrammati
- Proteggere la memoria centrale da accessi non leciti da parte di un processo a zone occupate da altri processi o dal sistema operativo



Caratteristiche

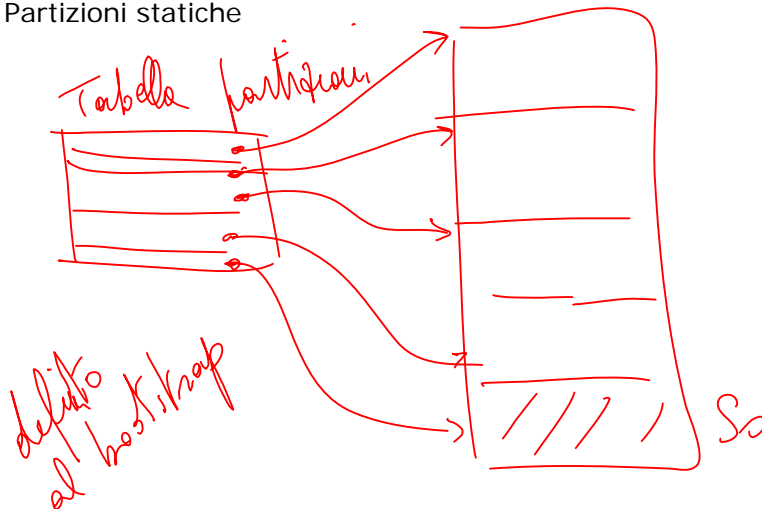
Somma dello spazio di indirizzamento dei processi
e del sistema operativo

=

Memoria centrale fisica

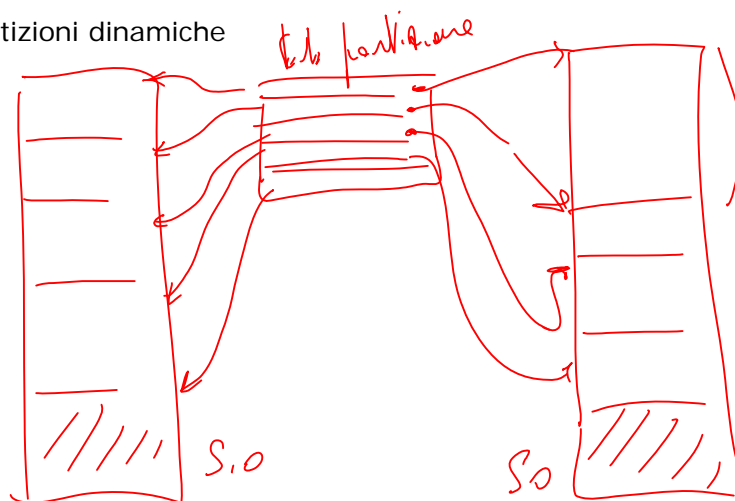
Partizioni fisse

Partizioni statiche

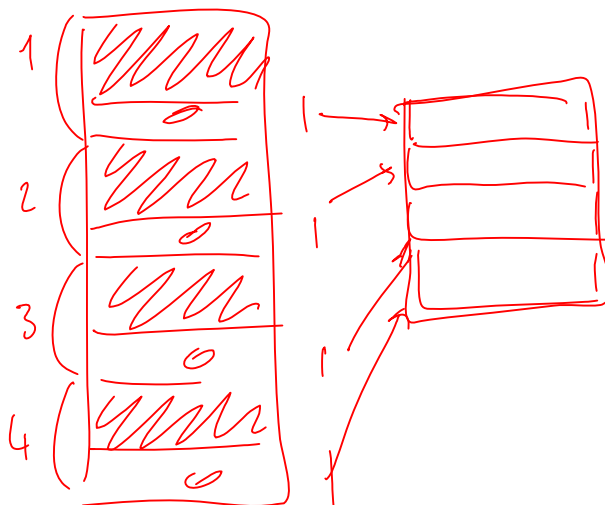


Partizioni variabili

Partizioni dinamiche



Frammentazione



In sintesi

- Abbiamo visto:
 - tecnica del partizionamento della memoria centrale
 - obiettivo
 - partizioni fisse e partizioni variabili
 - caratteristiche
- Notiamo che il partizionamento:
 - permette di ospitare
più processi in memoria centrale fisica
proteggendone lo spazio di ciascuno
 - **non aumenta lo spazio di memoria centrale**
a disposizione di ciascun processo
 - **deve essere configurato dal gestore del sistema**

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Primo Livello

Lezione 3 – Overlaying

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica dell'overlaying della memoria centrale
- Obiettivi
- Caratteristiche

Problemi

- La dimensione fisica della memoria centrale assegnata ad un processo può non essere sufficiente a contenere il suo spazio di indirizzamento logico
- Vincoli sui processi eseguibili dovuti allo spazio di memoria fisica disponibile
- Vincoli sull'eseguibilità variabili con lo specifico sistema di elaborazione considerato
- Non portabilità dei programmi dovuta a vincoli hardware non significativi

Obiettivo

- Permettere ad un processo di avere uno spazio di indirizzamento logico più grande della memoria centrale fisica ad esso allocata
 - caricando in memoria centrale fisica solo la porzione di spazio logico che serve per la computazione nell'immediato futuro

Overlaying (1)

- Identificare le porzioni di codice e di dati che sono usate:
 - sempre
 - in mutua esclusione per eseguire funzioni specifiche dell'applicazione (**overlay**)
- Creare in memoria centrale:
 - spazio per caricare le porzioni che devono essere sempre residenti in memoria centrale
 - spazio per caricare gli overlay

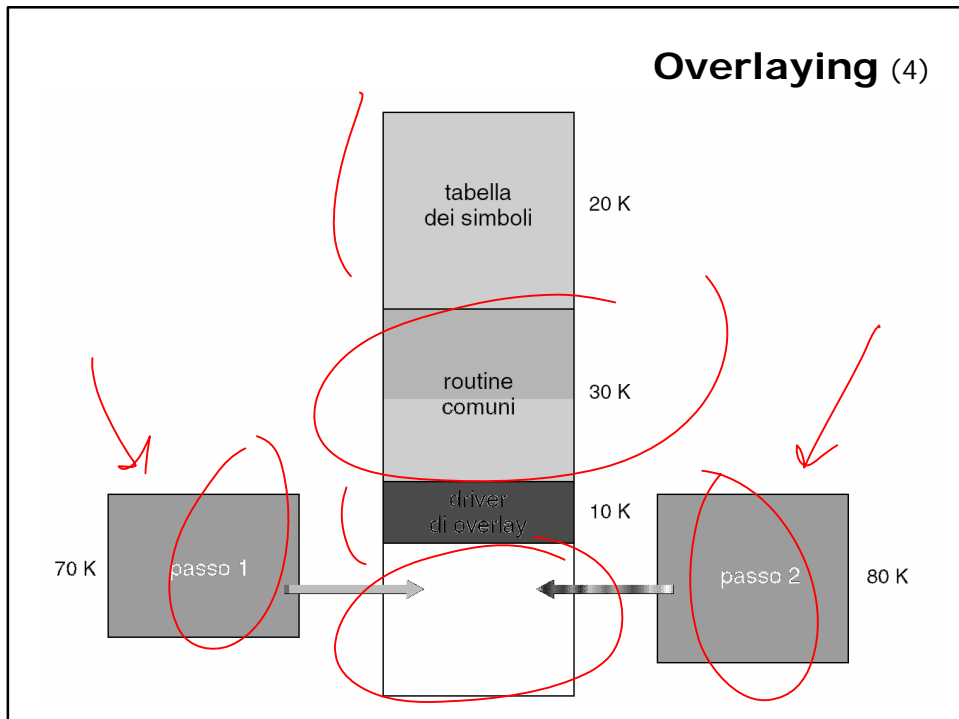
Overlaying (2)

- Caricare in memoria centrale fisica solo le porzioni di codice e dati che sono necessarie:
 - le porzioni usate sempre
 - le porzioni usate in mutua esclusione nell'immediato futuro
- Salvare in memoria temporanea gli overlay di dati che non servono per far spazio a nuovi overlay di dati richiesti dal programma

Overlaying (3)

- Sostituire le porzioni di codice e dati che non servono più in memoria centrale per rimpiazzarle con quelle richieste implicitamente dal programma con la chiamata di una funzione in un overlay non presente in memoria centrale

Overlaying (4)



Caratteristiche (1)

- **Ripiega lo spazio logico** occupato da un processo **in uno spazio fisico più piccolo** in memoria centrale
- Il programmatore deve identificare le porzioni di codice e dati sovrapponibili in overlay
- Omogeneità degli overlay per ridurre sfridi di memoria centrale non usata

Caratteristiche (2)

- Il compilatore introduce le chiamate alle funzioni di gestione per il caricamento e lo scaricamento di overlay
- Libreria di gestione dell'overlaying
- Overlay multipli
- Overlay gerarchici



In sintesi

- Abbiamo visto:
 - tecnica dell'overlaying della memoria centrale
 - obiettivo
 - gestione
 - caratteristiche
- Notiamo che l'overlaying
 - crea **spazio logico più grande dello spazio fisico** assegnato ad un processo nella memoria centrale fisica
 - deve essere **gestito dal programmatore applicativo** con supporto dell'ambiente di sviluppo e delle librerie del linguaggio di programmazione adottato

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Primo Livello

Lezione 4 – Swapping

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica dello swapping della memoria centrale
- Obiettivi
- Gestione
- Caratteristiche

Problemi

- Il partizionamento aumenta il numero di processi che sono caricati in memoria centrale
- Lo spazio fisico occupato da processi in stato di wait
rimane assegnato a tali processi
anche se non evolvono e
non è utilizzabile da processi che
potrebbero essere pronti all'esecuzione
- Spreco di memoria centrale ai fini dello sfruttamento del processore

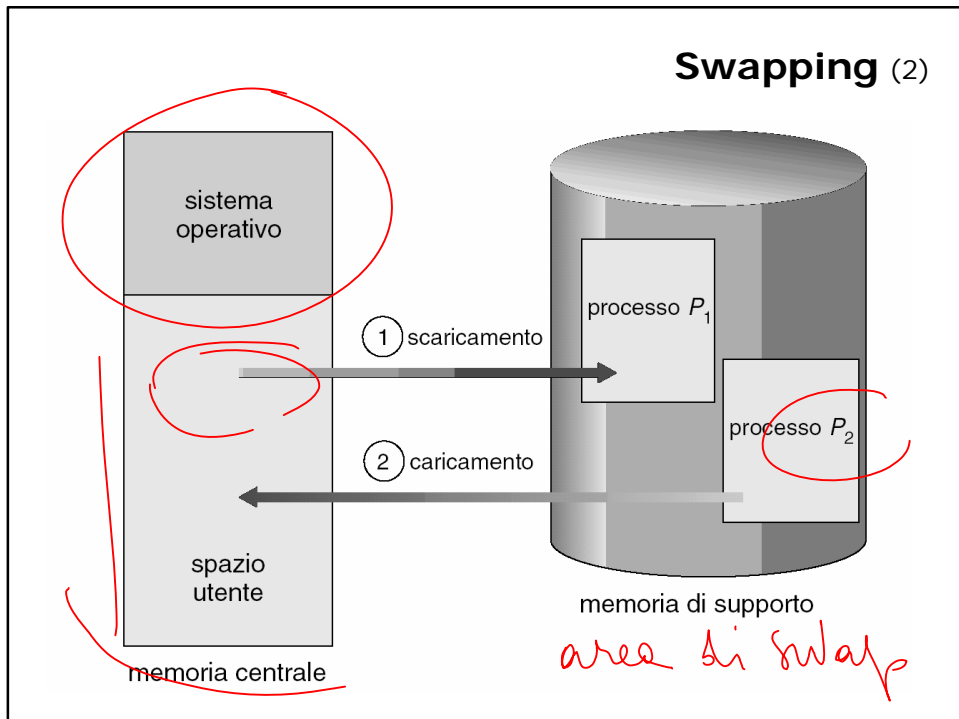
Obiettivi

- Liberare lo spazio di memoria centrale fisica occupato da processi in stato di wait per far posto a processi che possono essere eseguiti
- Aumentare il grado di multiprogrammazione
- Conservare in memoria centrale fisica solo processi negli stati di pronto e running

Swapping (1)

- Identificare i processi che non possono evolvere poiché sono in stato di wait
- Salvare in memoria temporanea i dati globali, lo heap e lo stack dei processi in stato di wait
- Rimuovere dalla memoria centrale fisica i processi che si trovano in stato di wait
- Caricare nello spazio libero della memoria centrale fisica processi che possono evolvere, prendendoli dall'area di memoria temporanea

Swapping (2)



Estensione dello swapping

Caricamento e scaricamento di processi

- nello stato di terminato
per liberare spazio in memoria centrale
- nello stato di pronto
per aumentare la turnazione di processi
("roll out/roll in" per sistemi
con schedulazione basata su priorità)

In sintesi

- Abbiamo visto:
 - tecnica dello swapping della memoria centrale
 - obiettivo
 - gestione
 - caratteristiche
- Notiamo che lo swapping:
 - **aumenta il numero di processi che usano la memoria centrale** e, quindi, possono evolvere
 - **non aumenta lo spazio fisico** di memoria centrale assegnato a ciascun processo
 - **gestito automaticamente dal sistema operativo**
 - può essere **lento** se applicato all'intero processo

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Secondo Livello

Lezione 1 – Paginazione

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica della paginazione della memoria centrale
- Obiettivi
- Gestione
- Supporti hardware

Problemi

- Caricare tutto il processo in memoria centrale è inutile per l'evoluzione della computazione
- Il caricamento e lo scaricamento di grosse porzioni di memoria richiedono tempi lunghi: lo swapping dell'intero processo è lento
- La gestione del caricamento e dello scaricamento effettuata dal programmatore nell'overlaying è complessa e può portare a sprechi di memoria

Obiettivi ⁽¹⁾

- Caricare e scaricare piccole porzioni di memoria
Velocità swapping e minimo sovraccarico
- Mantenere in memoria centrale solo le porzioni del processo che servono nell'immediato futuro
Minima occupazione di memoria centrale
- Porzioni di memoria di dimensione identica
Efficienza di gestione

Obiettivi (2)

- Possibile non contiguità delle porzioni di un processo nella memoria centrale fisica

Efficienza di gestione

- Gestione indipendente dal programmatore

Correttezza, sicurezza ed equità

Paginazione (1)

- Memoria centrale fisica divisa in **pagine fisiche** (frame)
- Spazio di indirizzamento del processo diviso in **pagine logiche** (pagine)
- Pagine logiche e pagine fisiche hanno la stessa dimensione



Paginazione (2)

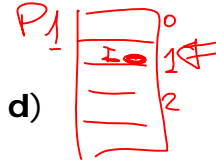
- La **Tabella delle pagine** di un processo definisce la corrispondenza tra pagine logiche e pagine fisiche del processo considerato

- TabellaPagine[PaginaLogica] =

PaginaFisica

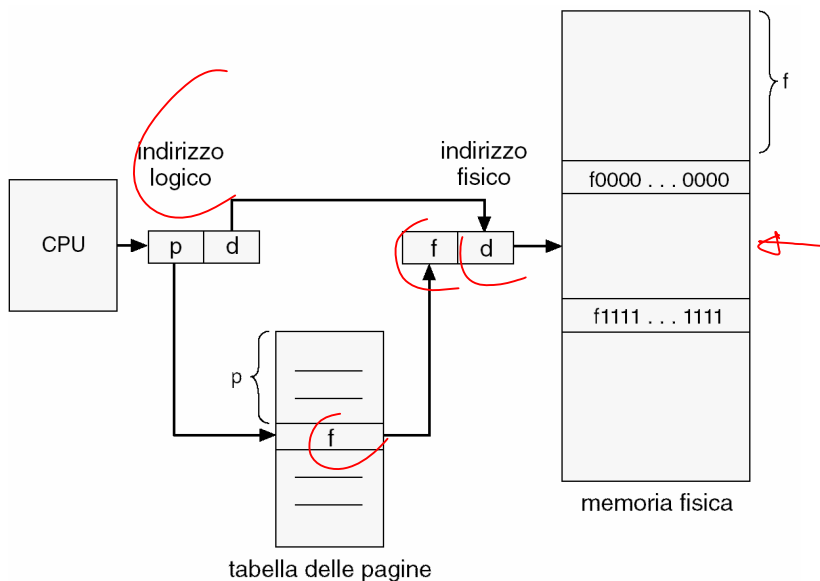
se caricata
 se non caricata

- Indirizzo logico =
 (numero di pagina logica **p**,
 spiazzamento nella pagina **d**)



- Indirizzo fisico =
 (numero di pagina fisica **f**,
 spiazzamento nella pagina **d**)

Paginazione (3)



Gestione della paginazione (1)

- Le pagine logiche
necessarie nell'immediato futuro
alla computazione di processi
nello stato di pronto
vengono caricate in pagine fisiche
- Le pagine logiche di un processo
possono essere caricate in pagine fisiche
non contigue in memoria centrale fisica

Gestione della paginazione (2)

- Le pagine logiche non caricate sono
conservate nell'area di swap
- Le pagine fisiche modificate vengono salvate
in area di swap prima di essere rimosse
dalle pagine fisiche

Gestione della paginazione (3)

- Il sistema operativo gestisce automaticamente
 - la selezione delle pagine da caricare in memoria centrale fisica
 - il caricamento in memoria centrale di pagine necessarie ma non presenti
 - la selezione delle pagine da scaricare dalla memoria centrale fisica
 - lo scaricamento delle pagine di memoria centrale non più necessarie

page fault

Supporto hardware: MMU

- Hardware dedicato per il supporto alla paginazione
Memory Management Unit
- Contiene la tabella delle pagine o il suo indirizzo in memoria centrale
- Traduce l'indirizzo logico in indirizzo fisico

Come gestire grandi tabelle delle pagine?

- Memoria ausiliaria di traduzione
- Tabella gerarchica delle pagine
- Tabella delle pagine con hashing
- Tabella invertita delle pagine

Memoria ausiliaria di traduzione

Translation Look-Aside Buffer

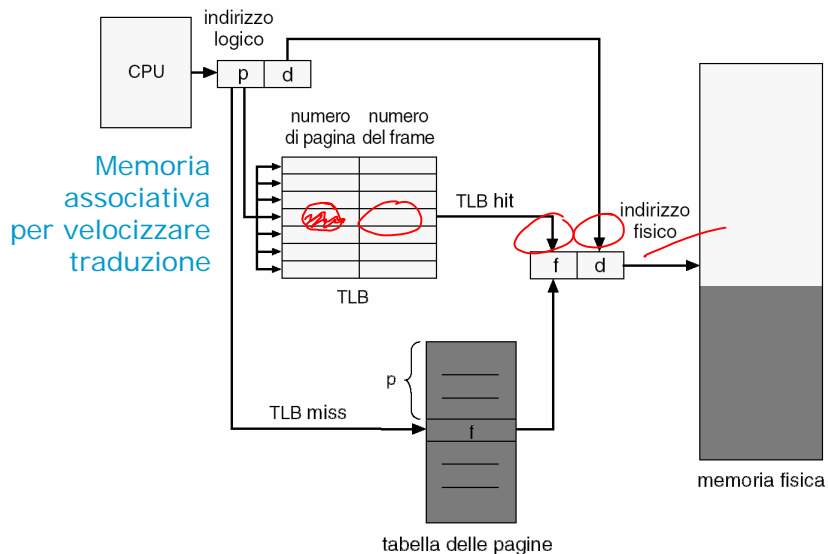


Tabella gerarchica delle pagine (1)

Tabella delle pagine a due livelli

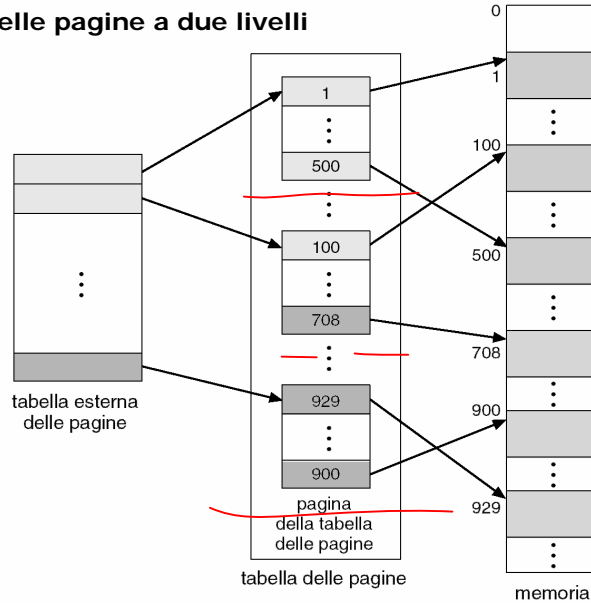
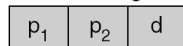


Tabella gerarchica delle pagine (2)

indirizzo logico



p_1

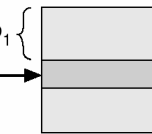
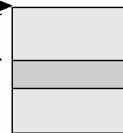


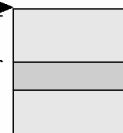
tabella esterna
delle pagine

p_2



pagina della tabella
delle pagine

d



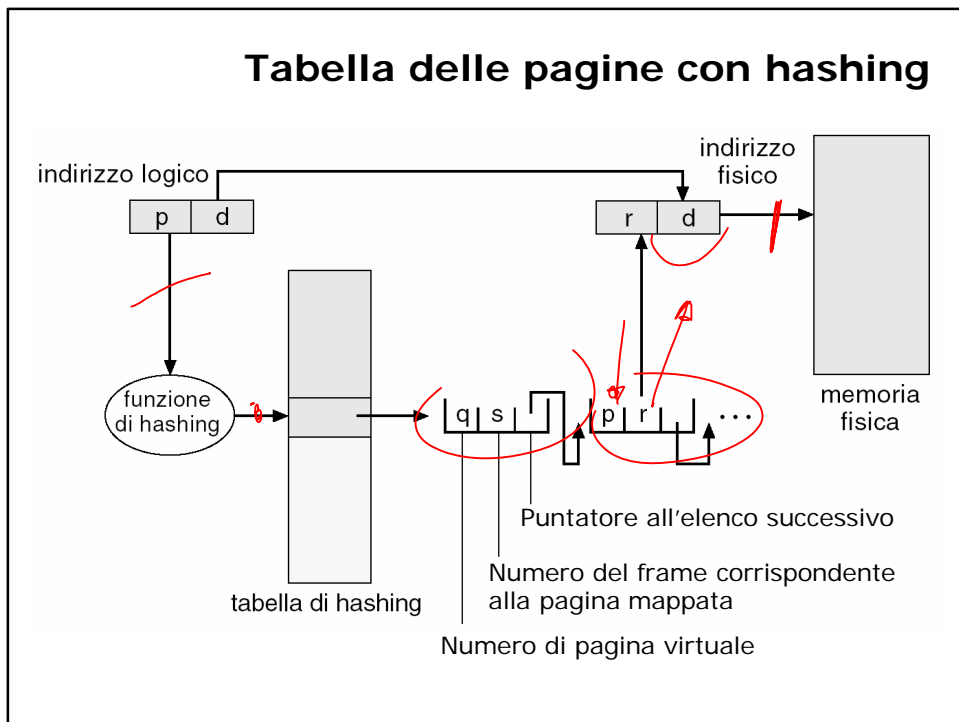


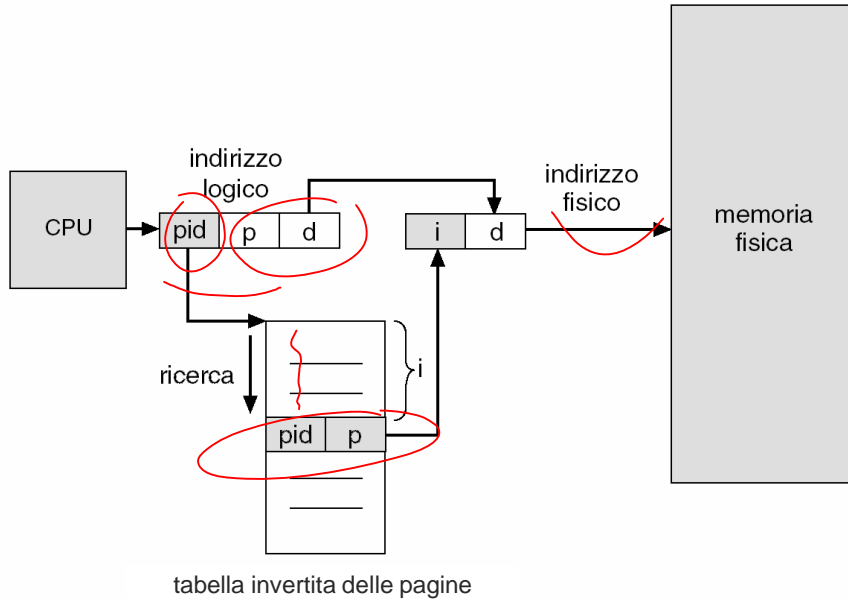
Tabella invertita delle pagine (1)

- Fornisce l'identificatore del processo e il numero di pagina logica caricata in una pagina fisica
- $TabellaInversaPagine[PaginaFisica] =$

Processo, PaginaLogica
se caricata

se non caricata
- Unica per il sistema di elaborazione

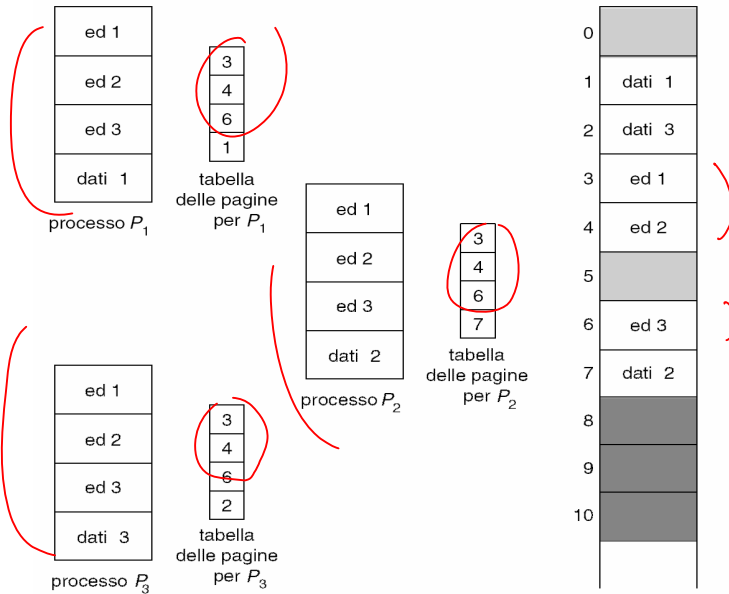
Tabella invertita delle pagine (2)



Protezione delle pagine

- Un processo può accedere solo alle sue pagine
 - La protezione dagli accessi di altri processi è implicita nella tabella delle pagine
- Bit di protezione
 - Permettono di definire pagine in
 - lettura/scrittura
 - sola lettura
 - sola esecuzione

Condivisione delle pagine



In sintesi

- Abbiamo visto:
 - tecnica della paginazione della memoria centrale
 - obiettivi
 - gestione
 - supporti hardware
 - protezione
- Notiamo che la paginazione
 - crea **spazio logico più grande dello spazio fisico** assegnato ad un processo nella memoria centrale fisica
 - è **gestito automaticamente dal sistema operativo**
 - è **efficiente** poiché sposta piccole porzioni di memoria

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Secondo Livello

Lezione 2 – Segmentazione

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica della segmentazione della memoria centrale
- Obiettivi
- Gestione
- Supporti hardware
- Frammentazione della memoria

Problemi

- Stessi problemi affrontati dalla paginazione
- La paginazione vede
lo spazio di indirizzamento logico
come un unico vettore di indirizzi
di memoria omogenei
- La paginazione non permette di tipizzare
le varie porzioni di spazio di indirizzamento logico
(ad esempio: codice, dati globali, heap, stack)
- Non permette la condivisione semplice
ed efficiente di porzioni di memoria
(ad esempio: codice)

Obiettivi (1)

- Stessi obiettivi della paginazione
(eccetto la dimensione identica delle porzioni)

Obiettivi (2)

- Supportare la visione dello spazio di indirizzamento dei processi dal punto di vista dell'utente
- Dare una visione semantica alle porzioni di spazio di indirizzamento dei processi
- Supportare la tipizzazione di porzioni dello spazio di indirizzamento logico e il controllo degli accessi e delle operazioni ammissibili in base al tipo

Obiettivi (3)

- Supportare la condivisione di porzioni dello spazio di indirizzamento tra diversi processi

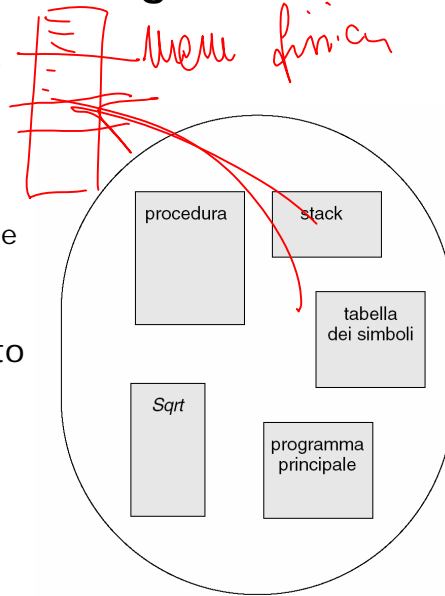
Segmentazione (1)

- Memoria centrale fisica divisa in **segmenti fisici** (frame)

struttura monodimensionale di memoria

- Spazio di indirizzamento del processo diviso in **segmenti logici** (segmenti)

struttura bidimensionale di memoria



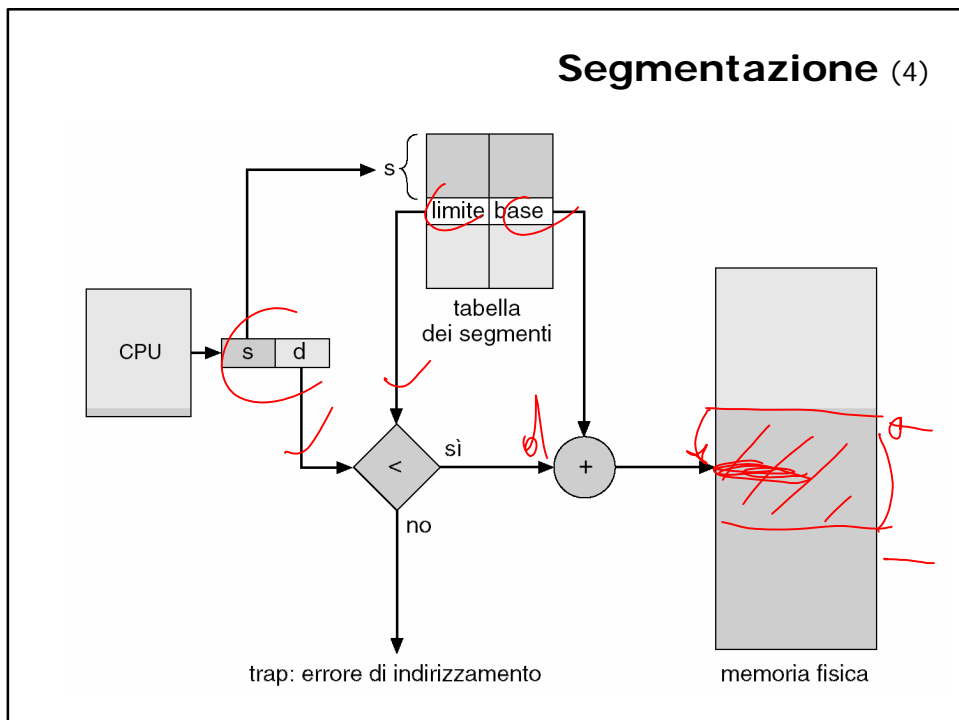
Segmentazione (2)

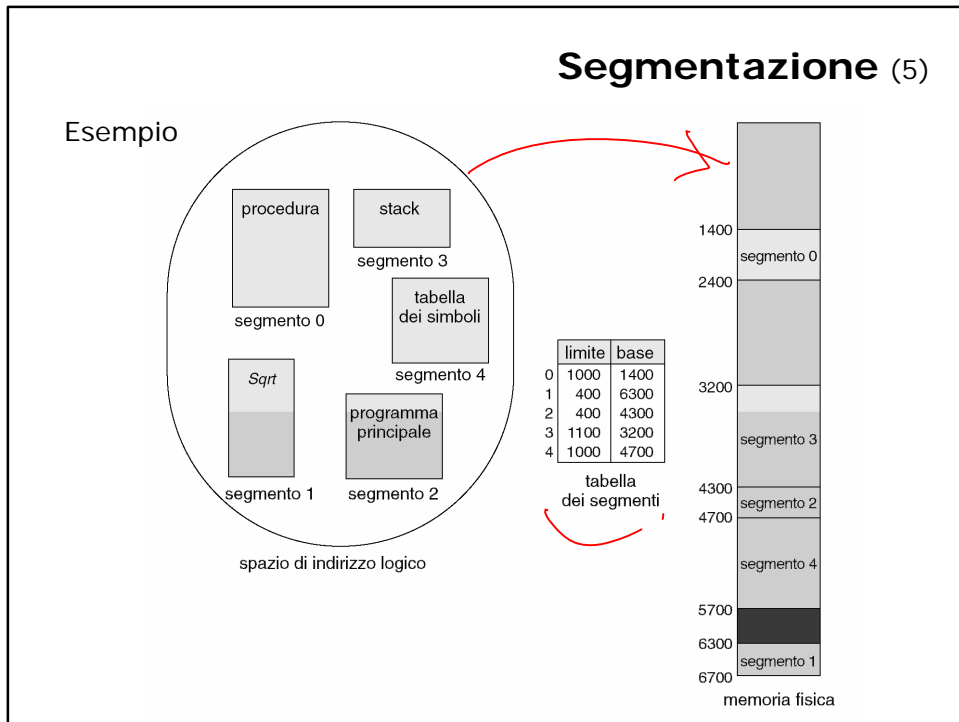
- I segmenti contengono informazioni di tipo diverso
(tipizzazione dei segmenti)
- I segmenti possono avere dimensioni diverse
- Un segmento è caricato in un frame di ugual dimensione

Segmentazione (3)

- La **Tabella dei segmenti** di un processo definisce la corrispondenza tra i segmenti e i frame del processo considerato
- TabellaSegmenti[Segmento] =
 (IndirizzoBaseFrame, DimensioneSegmento)
 --- se caricato
 --- se non caricato
- Indirizzo logico =
 (numero di segmento **s**,
 spiazzamento nel segmento **d**)
- Indirizzo fisico =
 (indirizzo di base del frame **bf**,
 spiazzamento nel frame **d**)

Segmentazione (4)





Gestione della segmentazione (1)

- I segmenti necessari nell'immediato futuro alla computazione di processi nello stato di pronto vengono caricati in frame
- I segmenti di un processo possono essere caricati in frame non contigui in memoria centrale fisica

Gestione della segmentazione (2)

- I segmenti non caricati sono conservati nell'area di swap
- I frame modificati vengono salvati in area di swap prima di essere rimossi dalla memoria centrale fisica

Gestione della segmentazione (3)

- Il programmatore deve configurare la divisione del processo in segmenti

La configurazione è effettuata implicitamente mediante la strutturazione del programma in moduli compilati separatamente e nell'uso di librerie

- Il compilatore e il linker generano
 - un segmento codice per modulo sorgente
 - un segmento dati globali per modulo
 - un segmento per la tabella dei simboli
 - un segmento di stack
 - un segmento di heap

Gestione della segmentazione (4)

- Il sistema operativo gestisce automaticamente
 - la selezione dei segmenti da caricare in memoria centrale fisica
 - il caricamento in memoria centrale dei segmenti necessari ma non presenti
 - la selezione dei frame da scaricare dalla memoria centrale fisica
 - lo scaricamento dei frame di memoria centrale non più necessari

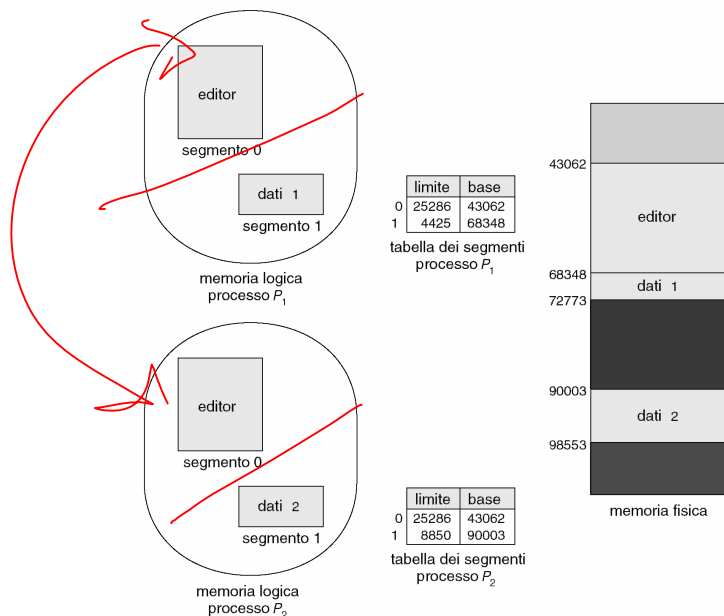
Supporto hardware: MMU

- Hardware dedicato per il supporto alla segmentazione
Memory Management Unit
- Contiene la tabella dei segmenti o il suo indirizzo in memoria centrale
- Traduce l'indirizzo logico in indirizzo fisico

Protezione dei segmenti

- Un processo può accedere solo ai suoi segmenti
 - La protezione dagli accessi di altri processi è implicita nella tabella dei segmenti
- Bit di protezione
 - Permettono di definire segmenti in
 - lettura/scrittura
 - sola lettura
 - sola esecuzione

Condivisione dei segmenti



Frammentazione della memoria

- Un segmento viene caricato in un frame di dimensione sufficiente a contenerlo
- Uno sfrido può rimanere inutilizzato nel frame
- Causa: dimensioni variabili dei segmenti
- Effetto: frammentazione esterna della memoria
- Soluzione: collezione degli sfridi di memoria (garbage collection) per creare frame liberi grandi, con eventuale rilocalizzazione dei segmenti caricati

In sintesi

- Abbiamo visto:
 - tecnica della segmentazione della memoria centrale
 - obiettivi
 - gestione
 - supporti hardware
 - protezione
 - frammentazione della memoria
- Notiamo che la segmentazione
 - crea **spazio logico più grande dello spazio fisico** assegnato ad un processo nella memoria centrale fisica
 - è **configurata implicitamente dal programmatore**
 - è **gestita automaticamente dal sistema operativo**
 - è **efficiente** poiché sposta piccole porzioni di memoria

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Tecniche di Base di Secondo Livello

Lezione 3 – Segmentazione con paginazione

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Tecnica della segmentazione con paginazione della memoria centrale
- Obiettivi
- Gestione
- Supporti hardware

Problemi

- Stessi problemi affrontati dalla segmentazione
- La segmentazione crea frammentazione della memoria centrale

Obiettivi

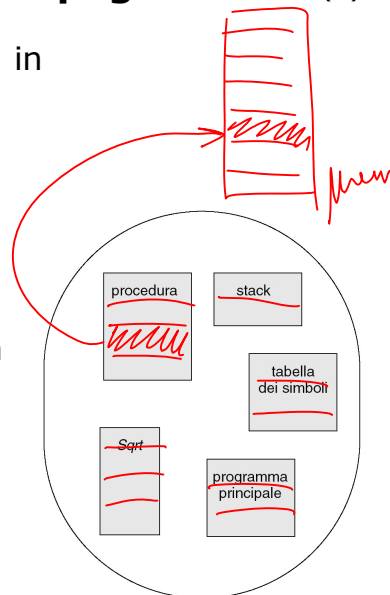
- Stessi obiettivi della segmentazione
- Evitare la frammentazione della memoria centrale

Segmentazione con paginazione (1)

- Fonde le caratteristiche di
 - gestione semplice ed efficiente
tipica della paginazione
 - identificazione dei frame liberi
 - scelta del frame libero in cui caricare una pagina
 - nessuna frammentazione
 - gestione tipizzata
tipica della segmentazione
 - verifica degli accessi e delle operazioni
 - condivisione di porzioni di memoria

Segmentazione con paginazione (2)

- Memoria centrale fisica divisa in **pagine fisiche** (frame)
- Spazio di indirizzamento del processo diviso in **segmenti logici** (segmenti),
ciascuno suddiviso in **pagine logiche**

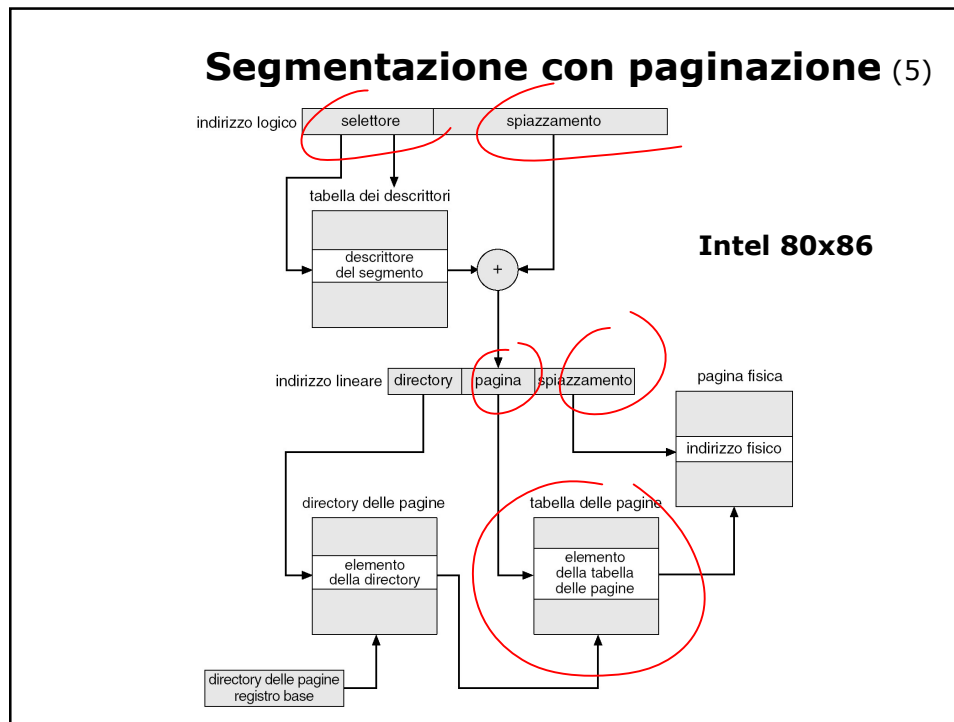


Segmentazione con paginazione (3)

- I segmenti contengono informazioni di tipo diverso
(tipizzazione dei segmenti)
- Le pagine di un segmento sono porzioni indifferenziate dello spazio di indirizzamento del segmento
- I segmenti possono avere dimensioni diverse
- Le pagine hanno tutte la stessa dimensione
- I frame hanno tutti la stessa dimensione, uguale alle pagine logiche
- Una pagina di un segmento è caricata in un frame

Segmentazione con paginazione (4)

- Indirizzo logico =
(numero di segmento **s**,
numero di pagina **p**,
spiazzamento nella pagina **d**)
- Indirizzo fisico =
(numero di frame **f**,
spiazzamento nel frame **d**)



Gestione della segmentazione con paginazione (1)

- Le pagine necessarie nell'immediato futuro alla computazione di processi nello stato di pronto vengono caricate in frame
- Le pagine di un segmento di un processo possono essere caricate in frame non contigui in memoria centrale fisica

Gestione della segmentazione con paginazione

(2)

- Le pagine non caricate sono conservate nell'area di swap
- I frame modificati vengono salvati in area di swap prima di essere rimossi dalla memoria centrale fisica

Gestione della segmentazione con paginazione

(3)

- Il programmatore deve configurare la divisione del processo in segmenti per supportare la tipizzazione

La configurazione è effettuata implicitamente mediante la strutturazione del programma in moduli compilati separatamente e nell'uso di librerie

Gestione della segmentazione con paginazione

(4)

- Il sistema operativo gestisce automaticamente
 - la selezione delle pagine da caricare in memoria centrale fisica
 - il caricamento in memoria centrale delle pagine necessarie ma non presenti
 - la selezione dei frame da scaricare dalla memoria centrale fisica
 - lo scaricamento dei frame di memoria centrale non più necessari

Supporto hardware: MMU

- Hardware dedicato per il supporto alla segmentazione con paginazione
Memory Management Unit
- Traduce l'indirizzo logico in indirizzo fisico

In sintesi

- Abbiamo visto:
 - tecnica della segmentazione con paginazione della memoria centrale
 - obiettivi
 - gestione
 - supporti hardware
- Notiamo che la segmentazione con paginazione
 - crea **spazio logico più grande dello spazio fisico** assegnato ad un processo nella memoria centrale fisica
 - è **configurata implicitamente dal programmatore**
 - è **gestita automaticamente dal sistema operativo**
 - è **efficiente** poiché sposta piccole porzioni di memoria
 - **evita frammentazione**

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Memoria Virtuale

Lezione 1 – Concetti e tecniche fondamentali

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

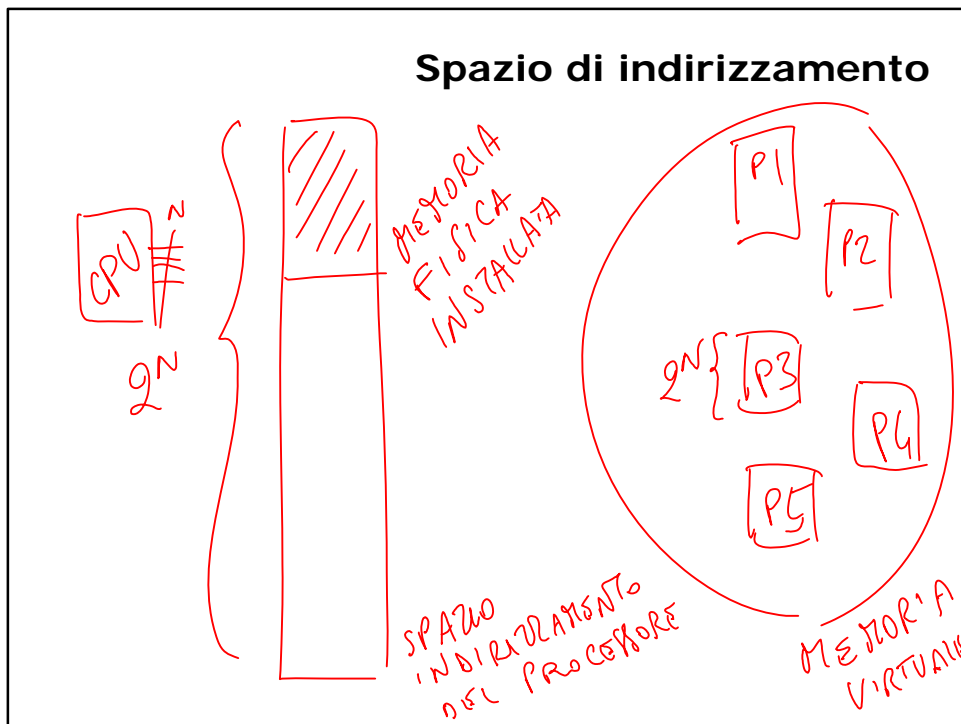
- Problemi di uso della memoria centrale fisica
- Obiettivi
- Concetto di memoria virtuale
- Tecniche per la realizzazione della memoria virtuale

Problemi

- Dimensioni ridotte della memoria centrale rispetto allo spazio di indirizzamento richiesto dai programmi
- Multiprogrammazione
- Condivisione della memoria centrale tra processi e protezione degli spazi assegnati
- Gestione efficiente dell'assegnamento della memoria centrale ai processi
- Supporto al time-sharing

Obiettivo

- Creare una **memoria centrale virtuale**
 - **astrazione**
 - più grande della memoria centrale fisica assegnata ad ogni processo
 - indipendente dalla memoria centrale fisica installata nel sistema
 - gestione efficiente della memoria
 - **virtualizzazione**
 - ogni processo può utilizzare tutto lo spazio di indirizzamento del processore
 - ogni processo ignora la presenza degli altri processi



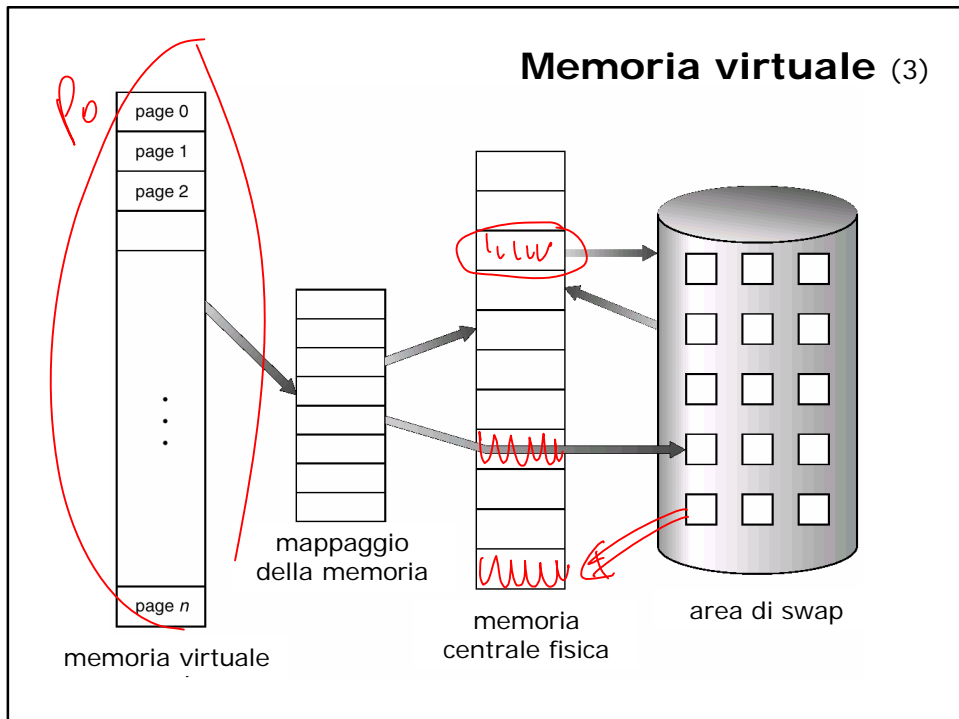
Memoria virtuale (1)

Insieme di tecniche e politiche che permettono di creare la visione di uno spazio di memoria centrale per ogni processo

- con dimensione pari allo spazio di indirizzamento del processore
- dedicato al singolo processo
- protetto dagli altri processi
- gestito in modo efficiente
- con porzioni eventualmente condivise con altri processi

Memoria virtuale (2)

- Divisione dello spazio di indirizzamento di un processo in porzioni
- Divisione della memoria centrale in frame
- Mappaggio protetto delle porzioni nei frame
- Caricamento in frame delle sole porzioni che servono nel prossimo futuro per la computazione dei processi che sono negli stati di "pronto" e "in esecuzione"
- Conservazione delle porzioni non caricate nell'area di swap



Tecniche per la memoria virtuale

- Paginazione o segmentazione o segmentazione con paginazione
- Swapping
- Meccanismi per il rilevamento di mancanza di frame
- Politiche di selezione dei frame da scaricare dalla memoria centrale fisica
- Meccanismi per il caricamento di frame
- Politiche di selezione delle porzioni da caricare nella memoria centrale fisica

In sintesi

- Concetto di memoria virtuale
- Obiettivi
- Tecniche per la realizzazione della memoria virtuale

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Memoria Virtuale

Lezione 2 – Tecniche di sostituzione della pagina

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

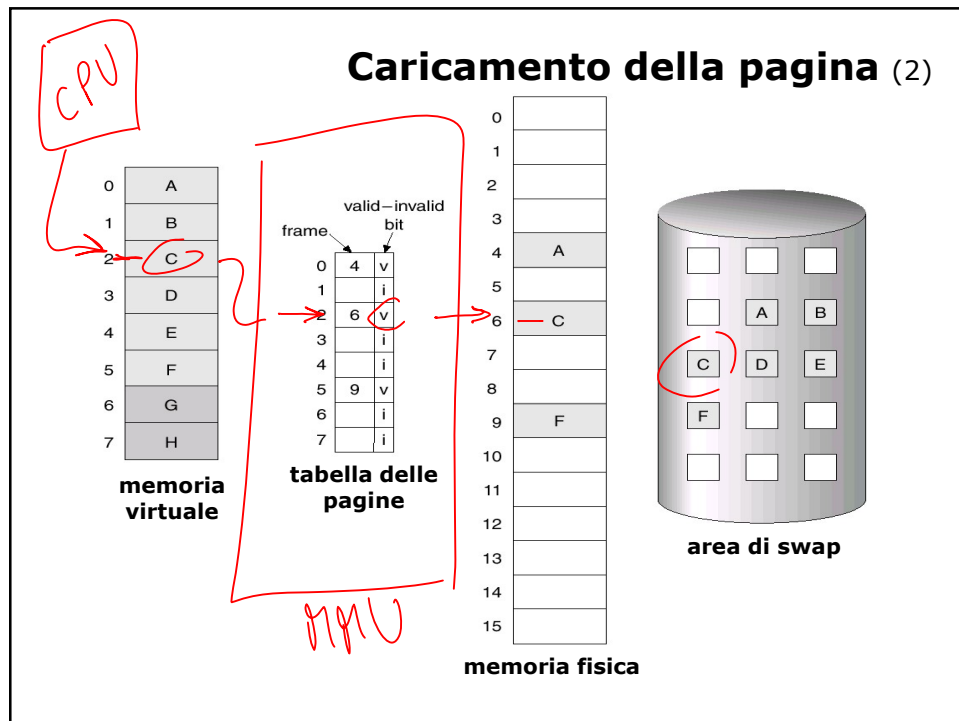
Sommario

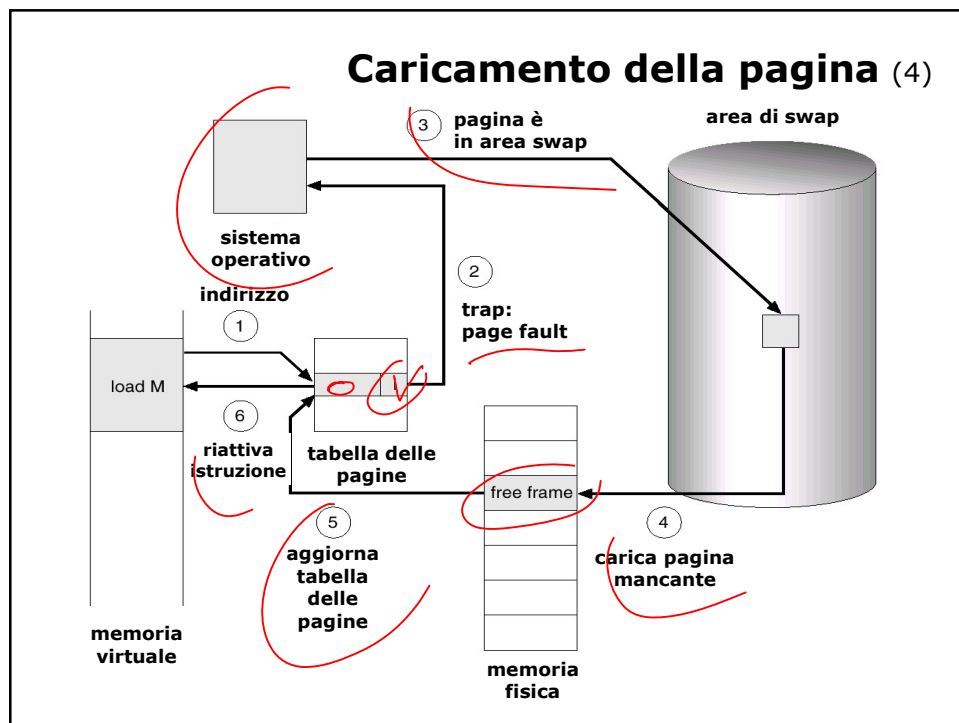
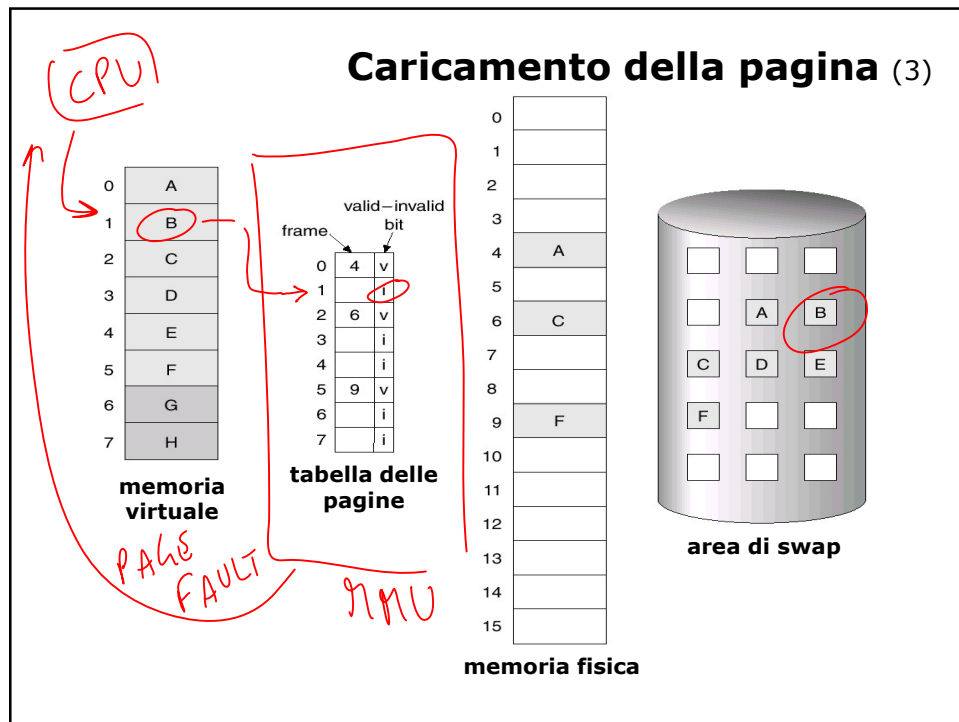
- Meccanismo di caricamento delle pagine
- Meccanismo di scaricamento delle pagine
- Sostituzione delle pagine
- Politiche di selezione delle pagine da scaricare
- Politiche di selezione delle pagine da caricare

Caricamento della pagina (1)

L'esecuzione del programma richiede la presenza in memoria centrale dei frame contenenti le istruzioni e i dati su cui deve operare in quel momento

Stringa di riferimento delle pagine
sequenza delle pagine richieste dal processo





Caricamento della pagina (5)

Prestazioni della richiesta di paginazione

- p** probabilità di mancanza di pagina
ma tempo di accesso alla memoria centrale fisica
spf tempo di servizio dell'interruzione di mancanza di pagina

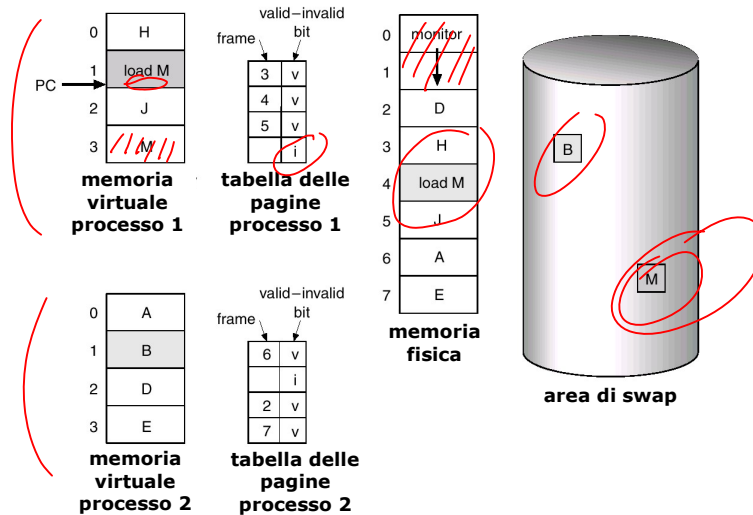
$$\text{tempo di accesso effettivo} = (1 - p) \times ma + p \times spf$$

Scaricamento della pagina

- Frame non modificata
→ rimuovere frame da memoria centrale
- Frame modificata
→ salvare frame su area di swap e rimuoverla da memoria centrale
→ salvare frame in buffer delle pagine e rimuoverlo da memoria centrale;
salvare poi il buffer su area di swap e liberarlo
- Frame residenti
→ non possono mai essere rimosse dalla memoria centrale

Sostituzione della pagina (1)

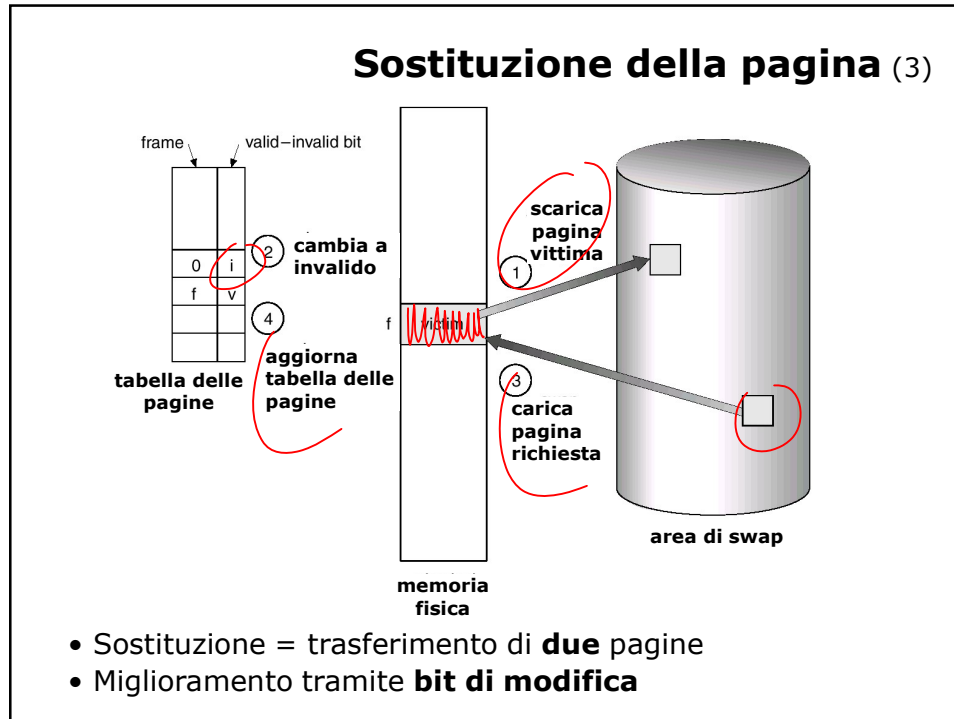
Scaricamento di una pagina
per caricare un'altra pagina



Sostituzione della pagina (2)

Identificazione del frame da scaricare
(pagina o frame vittima)

- Sostituzione locale
Il processo seleziona un frame da scaricare **solo** tra i frame ad esso assegnati
- Sostituzione globale
Il processo può selezionare un frame da scaricare tra tutti i frame, inclusi quelli di altri processi



Politiche di sostituzione della pagina

Politiche di selezione del frame da scaricare dalla memoria centrale

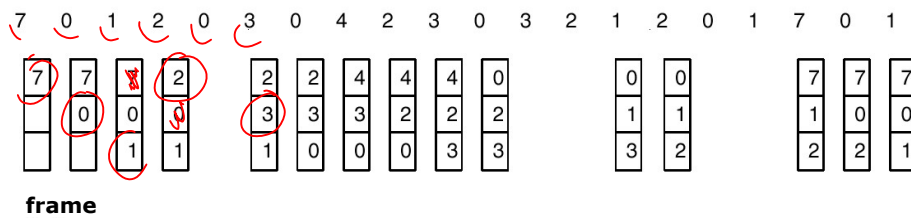
- First In / First Out
- Sostituzione ottima
- Least Recently Used (usato meno recentemente)
- Reference Bits (bit di riferimento)
- Second Chance (seconda possibilità)
- Least Frequently Used
(usata meno frequentemente)
- Most Frequently Used
(usata più frequentemente)

Politiche di sostituzione della pagina: FIFO

First In / First Out

Il frame da scaricare è il frame più vecchio
(quello caricato per primo)
tra i candidati allo scaricamento

stringa di riferimento

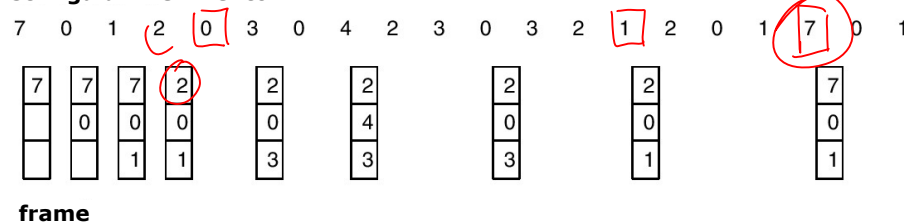


Politiche di sostituzione della pagina: SO

Sostituzione Ottima

Il frame da scaricare è quello che non sarà usato
per il più lungo periodo di tempo

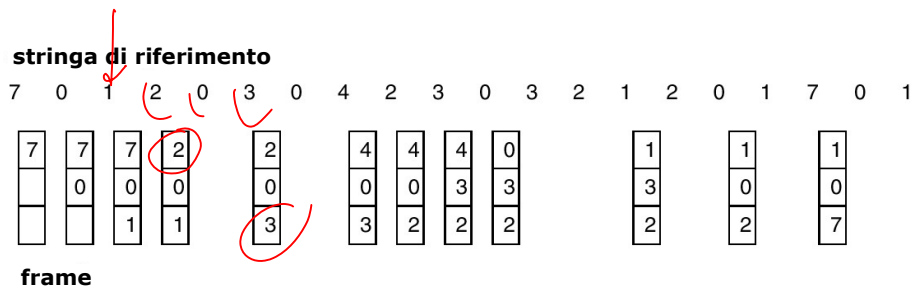
stringa di riferimento



Politiche di sostituzione della pagina: LRU (1)

Least Recently Used

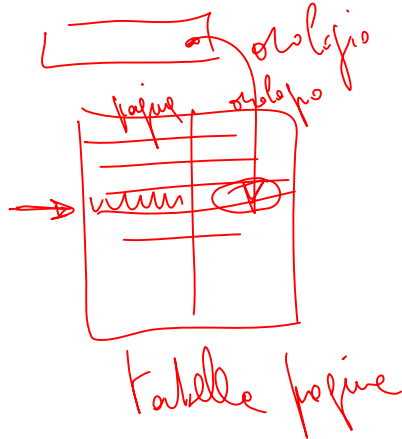
Il frame da scaricare è quello che è stato usato meno recentemente



Politiche di sostituzione della pagina: LRU (2)

Realizzazione

Orologio o Contatore



Stack

stringa di riferimento

4 7 0 7 1 0 1 2 1 2 7 1 2

2
1
0
7
4

stack prima di a

7
2
1
0
4

stack dopo b

a b

Politiche di sostituzione della pagina: RB (1)

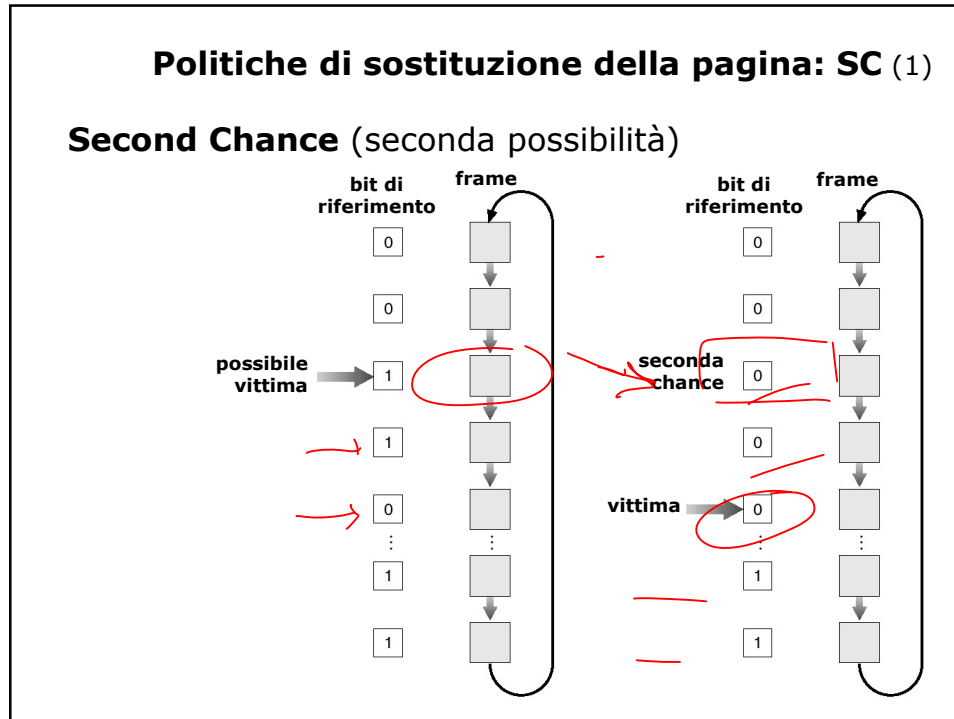
Reference Bits (bit di riferimento)

- Bit di riferimento per ogni pagina
 - 1 \Rightarrow pagina referenziata
 - 0 \Rightarrow pagina non referenziata
- Il sistema operativo azzera periodicamente i bit di riferimento
- Accesso ad una pagina \rightarrow bit di riferimento posto a 1
- Dopo un periodo di tempo T , bit di riferimento = 0 sono quelli dei frame a cui non si è acceduto nel periodo di tempo T
- Il frame da scaricare è uno con bit di riferimento = 0

Politiche di sostituzione della pagina: RB (2)

Estensione con storia più recente

- Gruppi di N bit di riferimento per ogni pagina
- Allo scadere di ogni periodo:
 - Bit di riferimento shiftati verso bit meno significativi
 - Bit più significativo inizializzato a 0
- Gruppo di bit conserva la storia più recente (ultimi N periodi)
- Il frame da scaricare è quello con gruppo di bit di riferimento avente valore minore



Politiche di sostituzione della pagina: SC (2)

Bit di riferimento e bit di modifica

- Bit di riferimento: accesso al frame
- Bit di modifica: scrittura nel frame
- Frame con
 - R=0, M=0: né usato né modificato recentemente
miglior frame da sostituire
 - R=0, M=1: non usato recentemente, ma modificato
seconda scelta: bisogna salvare frame
 - R=1, M=0: recentemente usato, ma non modificato
probabilmente sarà usato presto
 - R=1, M=1: recentemente usato e modificato
probabilmente sarà usato presto e
deve essere salvato

Politiche di sostituzione della pagina: LFU

Least Frequently Used (meno frequentemente usato)

- Contatore degli accessi al frame
- Contatori inizializzati a zero
- Il contatore di frame incrementato ad ogni accesso
- Il frame da scaricare è quello
con il contatore **minore**
- Problema: frame usati intensamente in un passato remoto rimangono in memoria
Soluzione: decadimento dell'uso mediante divisione per 2 dei contatori a intervalli regolari

Politiche di sostituzione della pagina: MFU

Most Frequently Used (più frequentemente usato)

- Contatore degli accessi al frame
- Contatori inizializzati a zero
- Il contatore di frame incrementato ad ogni accesso
- Il frame da scaricare è quello
con il contatore **maggiore**

Politiche di selezione delle pagine da caricare

- Pagina richiesta
- Previsione delle pagine richieste
 - Pagina richiesta + N pagine successive adiacenti
 - N pagine della stringa di riferimento predetta a partire dalla pagina richiesta

In sintesi

- Meccanismo di caricamento delle pagine
- Meccanismo di scaricamento delle pagine
- Sostituzione delle pagine
- Politiche di selezione delle pagine da scaricare
- Politiche di selezione delle pagine da caricare

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Memoria Virtuale

Lezione 3 – Tecniche di allocazione dei frame

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Allocazione dei frame ai processi
- Algoritmi di allocazione

Problema

- Il numero di frame allocati ad un processo influenza la frequenza di page fault di tale processo

maggiore è il numero di frame

- minore è l'occorrenza dei page fault
- minore è il tempo di accesso medio alla memoria centrale

- Quanti frame allocare a ciascun processo?
- Allocazione omogenea o eterogenea?

Vincoli per l'allocazione dei frame

Numero minimo di frame per ogni processo

dipende dall'architettura

Frame disponibili

dipende dalla memoria centrale fisica installata

Frame allocati

f_i frame allocati al processo P_i

r_j frame riutilizzati nella condivisione dai processi del gruppo G_j

d frame disponibili

$$S_i f_i - S_j r_j = d$$

Insieme dei frame da cui effettuare l'allocazione

Allocazione globale

Allocazione locale

Allocazione omogenea

Ogni processo ha la stessa quantità di frame

m frame totali

n processi

f_i frame allocati al processo P_i

$$f_i = m/n$$

Allocazione proporzionale alla dimensione

Il numero di frame allocate a un processo
è proporzionale alla dimensione del processo

m frame totali

s_i dimensione dello spazio virtuale occupato
dal processo P_i

f_i frame allocati al processo P_i

$$f_i = m \times s_i / \sum s_i$$

Allocazione proporzionale alla priorità

Il numero di frame allocate a un processo
è proporzionale alla priorità del processo

m frame totali

p_i priorità del processo P_i (logica positiva)

f_i frame allocati al processo P_i

$$f_i = m \times p_i / \sum p_i$$

In sintesi

- Problema dell'allocazione dei frame ai processi
- Algoritmi di allocazione

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Memoria Virtuale

Lezione 4 – Thrashing

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Il fenomeno del thrashing
- Tecniche di gestione

Il fenomeno del thrashing

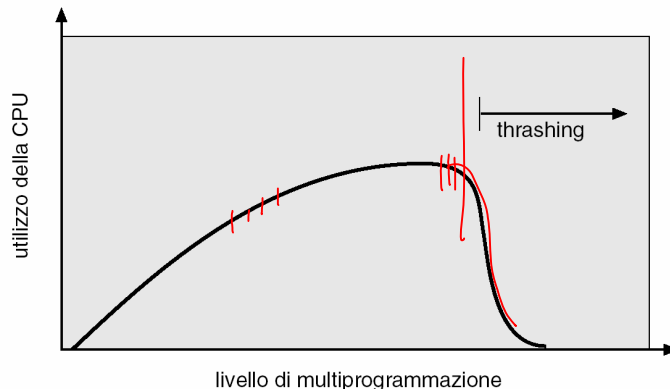
Se un processo ha troppo pochi frame allocati, i page fault occorrono così frequentemente che viene trascorso più tempo nella paginazione che nell'esecuzione del processo

Thrashing = paginazione spazzatura

Grave perdita di prestazioni

Cause

- Multiprogrammazione
- Algoritmo di schedulazione a lungo termine introduce nuovi processi per incrementare sfruttamento processore

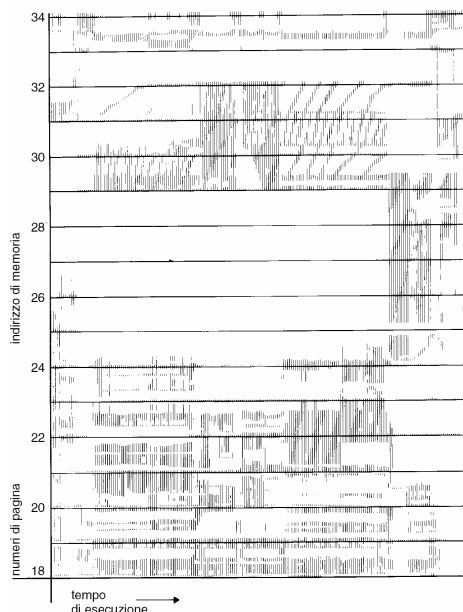


Evitare il thrashing

- Politica di schedulazione che impedisca il thrashing limitando il caricamento di nuovi processi quando il numero di frame allocati ai processi diminuisce eccessivamente
- Politica di allocazione dei frame che impedisca il thrashing restringendo l'insieme dei frame eligibili per lo scaricamento in caso di mancanza di frame liberi (allocazione locale)

Prevenzione del thrashing (1)

**Modello di
località
di esecuzione
del processo**



Prevenzione del thrashing (2)

- Identificare il numero di frame capaci di contenere la località del processo durante tutta la sua esecuzione

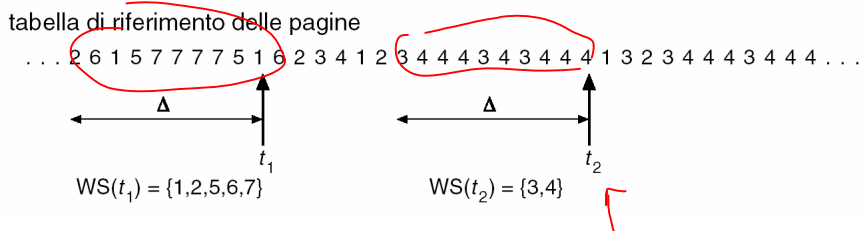
Approssimazioni

- **working set** (insieme delle pagine di lavoro)
- **page-fault frequency** (frequenza dei page fault)

Prevenzione del thrashing (3)

Working Set (insieme delle pagine di lavoro)

- Finestra del working set: ?
- Working Set (**WS**) è l'insieme delle ? pagine più recenti usate dal processo
- Se una pagina è in uso attivo, allora è nel WS; altrimenti sarà tolta dal WS dopo ? periodi dal suo ultimo utilizzo



Prevenzione del thrashing (4)

Working Set

m frame totali

f_i frame allocati al processo P_i

- Ad ogni nuovo processo P_i da caricare in memoria centrale si allocano i frame:

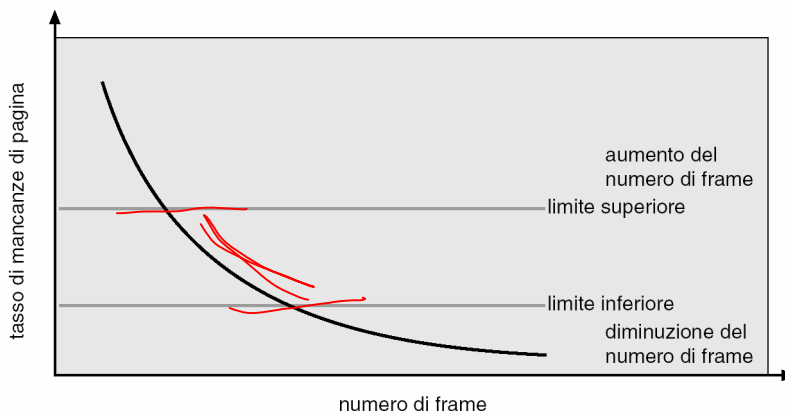
$$f_i = \max_t(WS_i(t))$$

- Se $\sum_{k=1..i} f_k > m$
viene selezionato un processo vittima P_v
che viene scaricato per evitare thrashing

Prevenzione del thrashing (5)

Page-Fault Frequency

(frequenza delle mancanze di pagina)



Prevenzione del thrashing (5)

Page-Fault Frequency

- Frequenza di page fault troppo elevata
→ si allocano frame al processo
- Frequenza di page fault troppo bassa
→ si deallocano frame al processo
- Mancanza di frame liberi
→ si sospende un processo per liberare frame

In sintesi

- Il fenomeno del thrashing
- Tecniche di gestione

SISTEMI OPERATIVI

Gestione della Memoria Centrale
Memoria Virtuale

Lezione 5 – Ottimizzazione delle prestazioni

Vincenzo Piuri

Università degli Studi di Milano - SSRI - CDL ONLINE

Sommario

- Prepaginazione
- Dimensione della pagina
- Translation Look-aside Buffer
- Tabella invertita delle pagine
- Strutturazione del programma
- Pagine residenti per dispositivi di I/O
- Pagine residenti per processi in tempo reale

Prepaginazione

Caricare pagine in memoria centrale
in anticipo rispetto al loro uso

Prevenire un elevato numero di page fault
all'attivazione iniziale del processo o
alla riattivazione dopo il rientro in memoria centrale

Portare la località del processo in memoria centrale

Nel modello del working set, si caricano
tutte le pagine del working set corrente

Dimensione della pagina

Qual è la dimensione ottimale di una pagina?

- **Pagine grandi**

- meno pagine
- dimensione minore della tabella delle pagine
- maggiore frammentazione interna
- minore tempo di I/O per caricamento e scaricamento
- minore risoluzione
- maggiore memoria caricata ma non usata
- possibile maggior numero di page fault

- **Pagine piccole**

- più pagine
- dimensione maggiore della tabella delle pagine
- minore frammentazione interna
- maggiore tempo di I/O per caricamento e scaricamento
- aumento della località
- maggiore risoluzione
- minore memoria caricata ma non usata
- possibile minor numero di page fault

Translation Look-aside Buffer

Estensione della TLB

$$= (\text{Dimensione TLB}) \times (\text{Dimensione pagina})$$

- maggiore estensione
 - minori page fault
 - minore tempo di accesso medio alla memoria
- aumentare dimensione TLB
- aumentare dimensione pagina
- pagine con dimensione eterogenea

Tabella invertita delle pagine

TabellaInvertita[PaginaFisica]

$$= (\text{Processo}, \text{PaginaLogica})$$

- Ridurre la quantità di memoria fisica necessaria per tradurre indirizzi virtuali in indirizzi fisici
- Tabelle esterne delle pagine referenziate quando c'è un page fault

Strutturazione del programma

I page fault e il working set possono essere ridotti se il programma ha una forte località

La forte località è ottenibile usando una forte strutturazione del programma

- figure strutturali
- modularità
- strutture dati

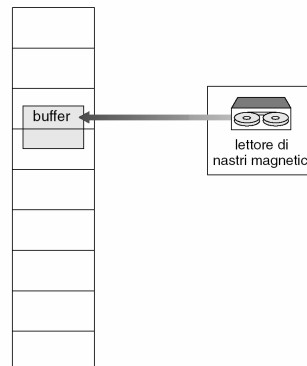
Compilatori e linker evitano di produrre codice che può soffrire di frequenti page fault

Pagine residenti per dispositivi di I/O

Page fault possono essere evitati lasciando le pagine per i buffer dei dispositivi di I/O residenti in memoria centrale

Buffer per dispositivi di I/O in

- spazio indirizzi del sistema operativo con copiatura nelle variabili del processo
- spazio indirizzi del processo con pagine residenti in memoria



Pagine residenti per processi in tempo reale

Processi in tempo reale
spesso non possono rispettare
il vincolo temporale
se viene utilizzata la memoria virtuale
a causa del tempo di gestione della paginazione

Soluzione:

lasciare residenti in memoria centrale
le pagine critiche
in modo da non dover effettuare
sostituzioni di pagina per caricarle

In sintesi

- Prepaginazione
- Dimensione della pagina
- Translation Look-aside Buffer
- Tabella invertita delle pagine
- Strutturazione del programma
- Pagine residenti per dispositivi di I/O
- Pagine residenti per processi in tempo reale