Capitolo 4

Gestione della Memoria

4.1 Introduzione alla gestione della memoria
4.2 Swapping
4.3 Memoria virtuale
4.4 Implementazione
4.5 Algoritmi di sostituzione
4.6 Criteri di progetto per la paginazione
4.7 Case study: Unix
4.8 Case study: Windows 2000

Algoritmi di Sostituzione

- Il Page fault forza la scelta su quale pagina deve essere rimossa
  - Libera memoria per la pagina da caricare
- Pagine modificate devono essere salvate
  - Quelle non modificate vengono semplicemente sovrascritte
- Deve evitare di selezionare una pagina riferita spesso
  - Potrebbe essere necessario ricaricarla in breve tempo

Algoritmo di Sostituzione Ottimo

- Sostituisce la pagina che sarà riferita nell'istante più lontano nel tempo
  - Ottimo ma non realizzabile

- In alternativa:
  - Si stima l'ordine di caricamento delle pagine in esecuzioni precedenti del processo
  - Neanche questa soluzione è pratica

Algoritmo di Sostituzione NRU (Not Recently Used)

- Ogni pagina ha un bit “Riferita”, e un bit “Modificata”
  - I bit sono settati quando la pagina è riferita o modificata
- Le pagine sono classificate come:
  1. Non riferita, non modificata
  2. Non riferita, modificata
  3. Riferita, non modificata
  4. Riferita, modificata
- NRU rimuove una pagina a caso dalla classe non vuota di indice minimo

Algoritmo di Sostituzione FIFO

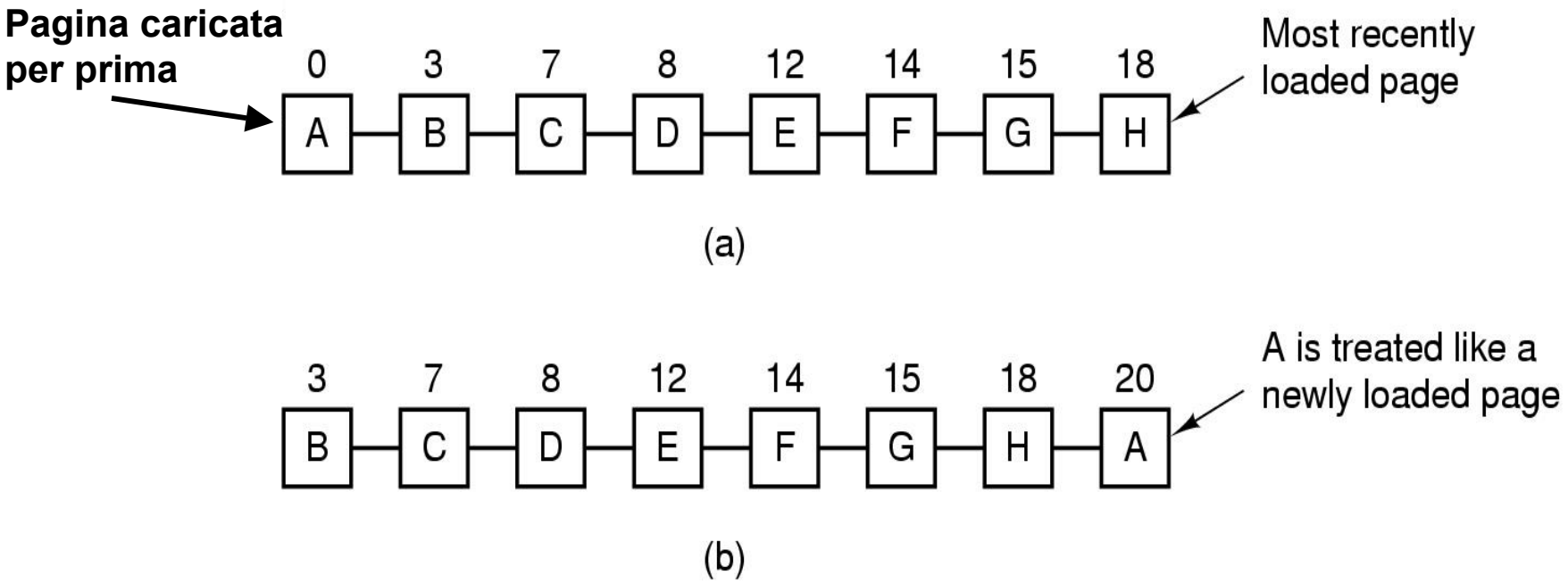
- Mantiene una lista di tutte le pagine
  - Nell'ordine in cui le pagine vengono caricate in memoria
- La prima pagina della lista viene sostituita

- Svantaggi
  - La pagina sostituita potrebbe essere quella usata più spesso

Least Recently Used (LRU)

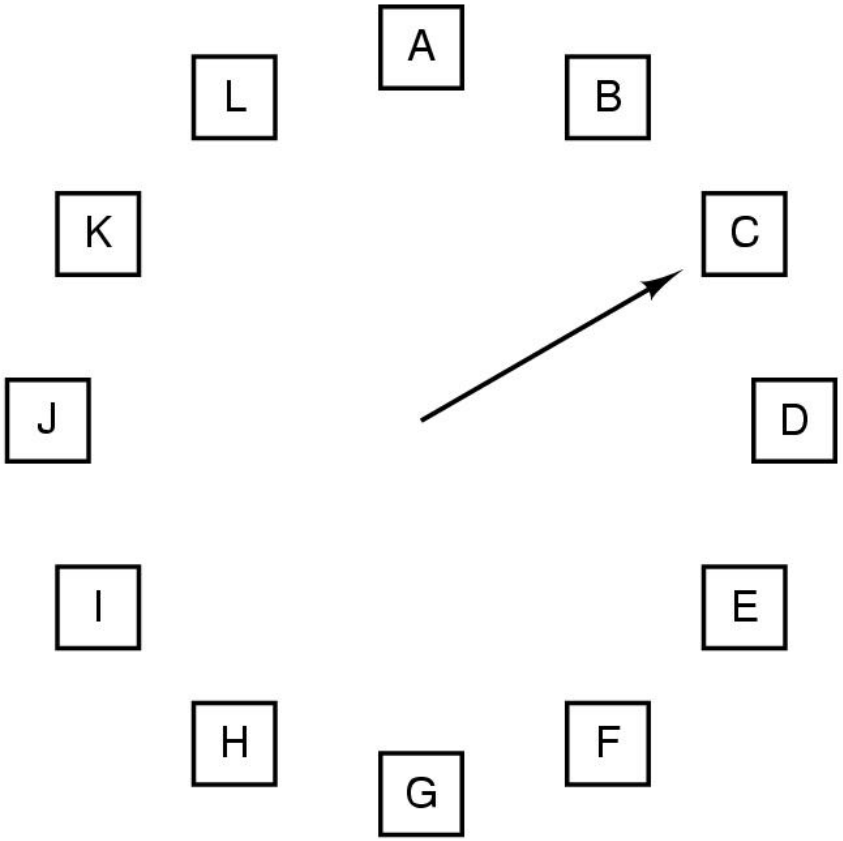
- Assume che le pagine usate di recente siano riferite di nuovo in breve tempo
  - Scarica le pagine inutilizzate da più tempo
- Mantiene una lista di pagine
  - Le pagine usate più di recente in cima
  - Aggiorna la lista ad ogni riferimento della memoria!!
- In alternativa mantiene un contatore per ogni record della tabella delle pagine
  - Scarica la pagine con il più piccolo valore nel contatore
  - Periodicamente azzera i contatori

Algoritmo di Sostituzione "Second Chance"



- Operazioni di un algoritmo "second chance"
  a) Pagine disposte in ordine FIFO
  b) Lista delle pagine se il fault di pagina avviene all'istante 20, *A* ha il bir *R* settato
  (I numeri sopra le pagine indicano il tempo di caricamento)

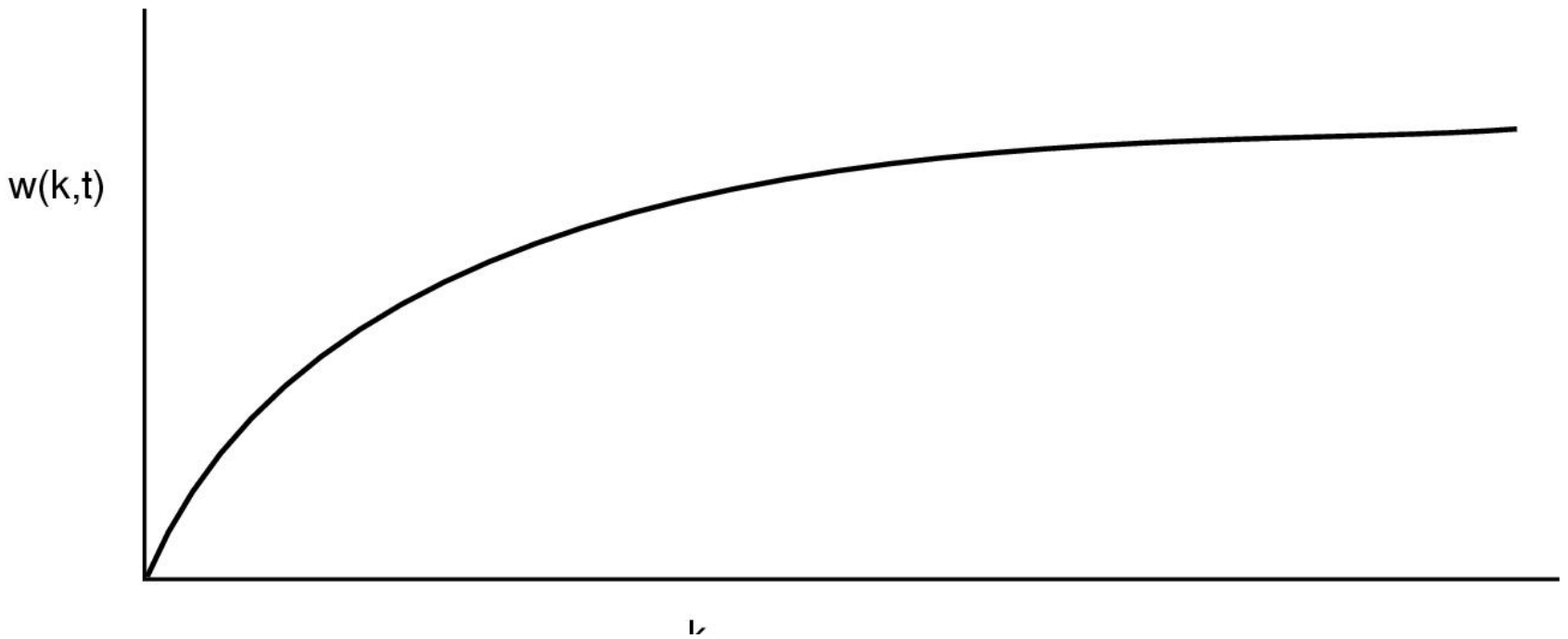
Algoritmo di Sostituzione “Clock”



Quando avviene un fault di pagina, la pagina indicata dal puntatore viene analizzata. L’azione dipende dal bir R:

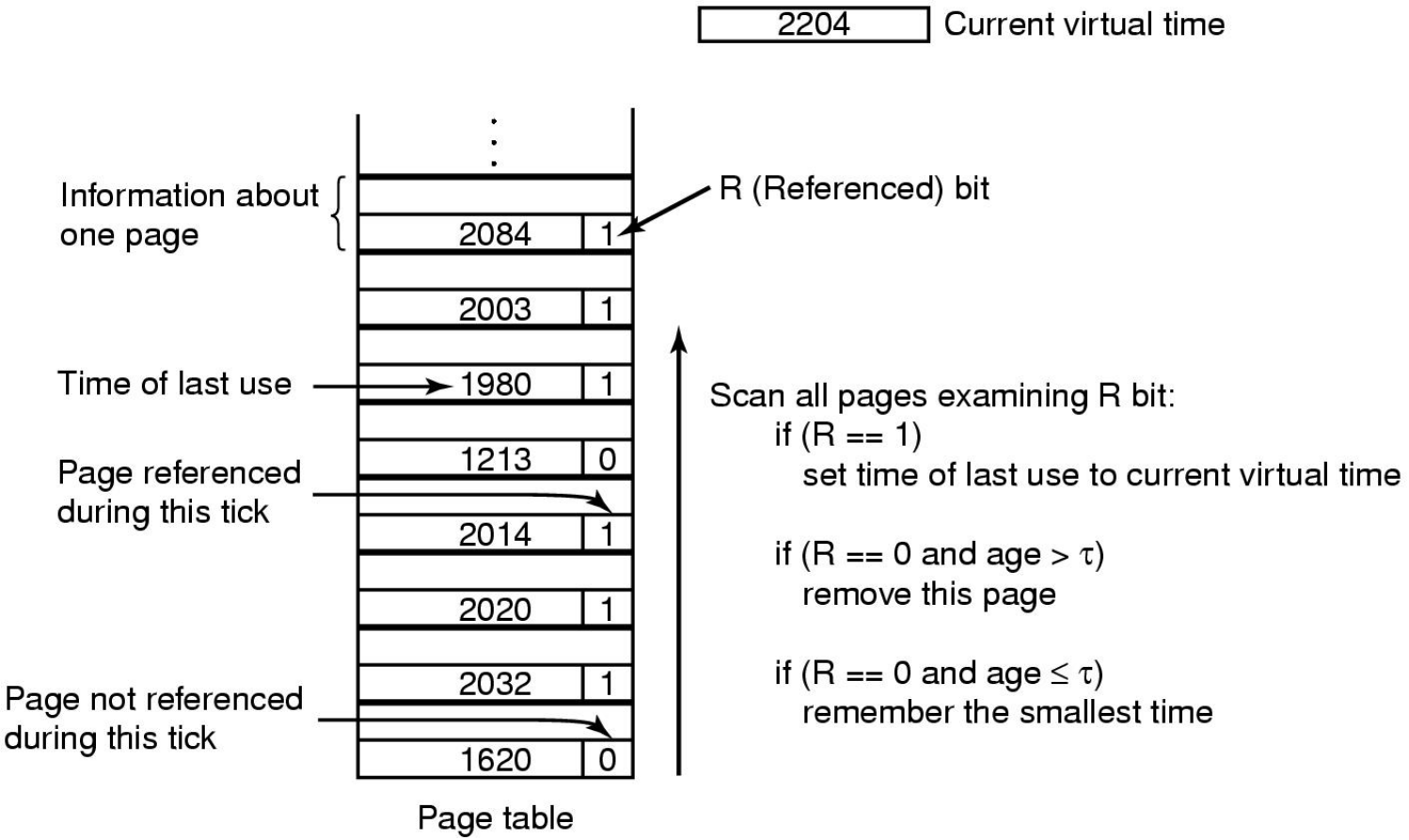
- R=0: elimina la pagina
- R=1: resetta R e avanza il puntatore

Algoritmo di Sostituzione “Working Set” (1)



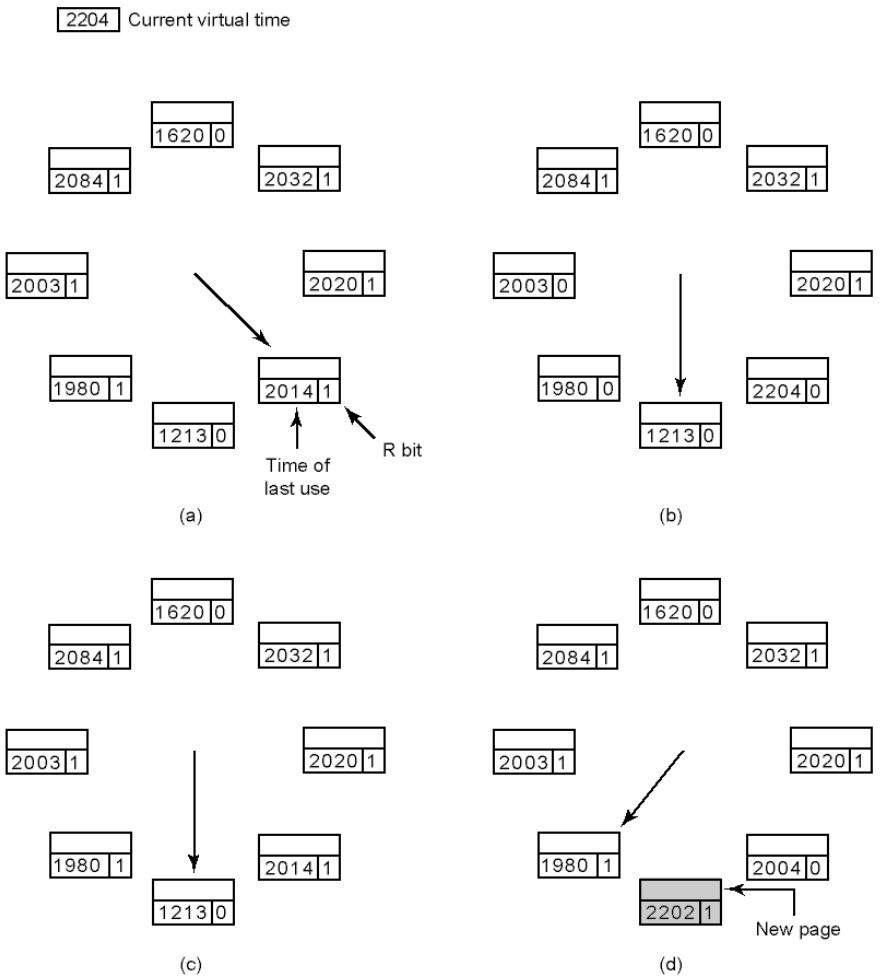
- Il “working set” è l’insieme di pagine riferite negli ultimi *k* accessi in memoria
- w(*k*,*t*) è la dimensione del “working set” al tempo *t*

Algoritmo di Sostituzione “Working Set”(2)



L’algoritmo “working set”

L'Algoritmo di Sostituzione "WSClock"

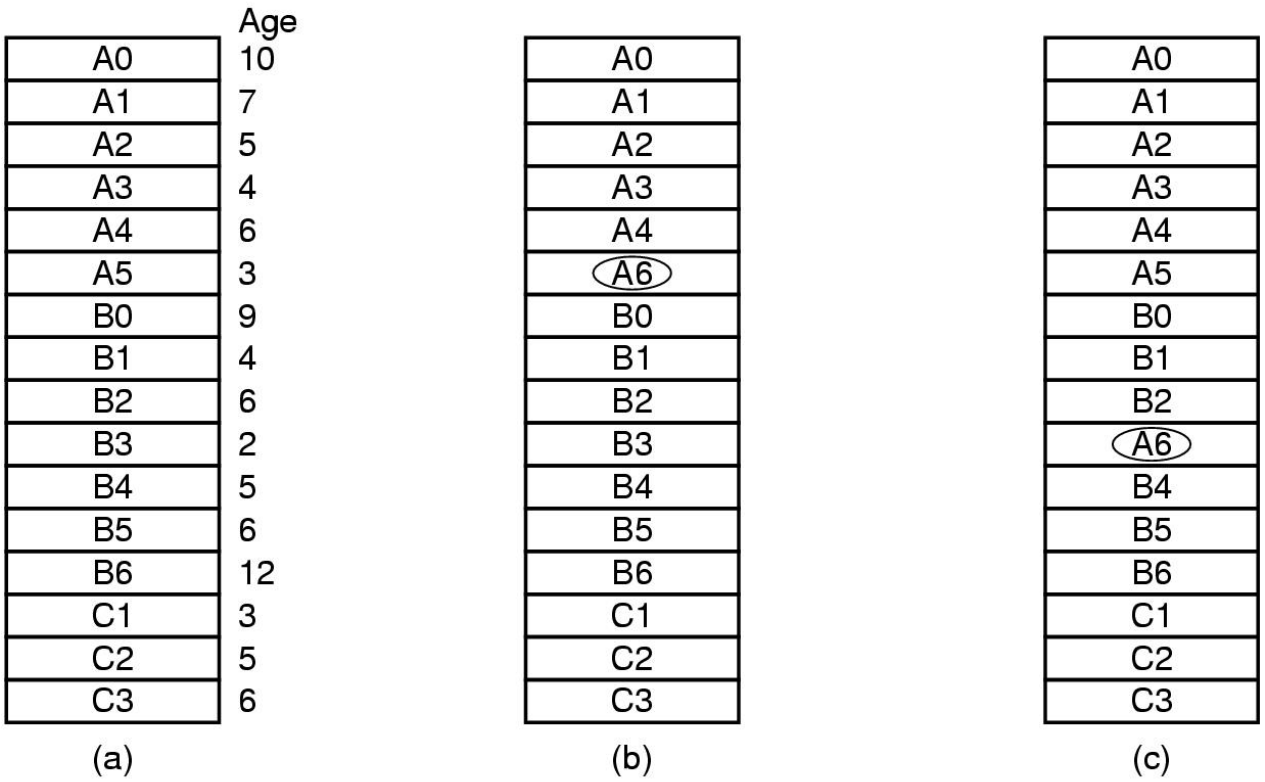


Funzionamento dell'algoritmo "WSClock"

Riassunto degli Algoritmi di Sostituzione

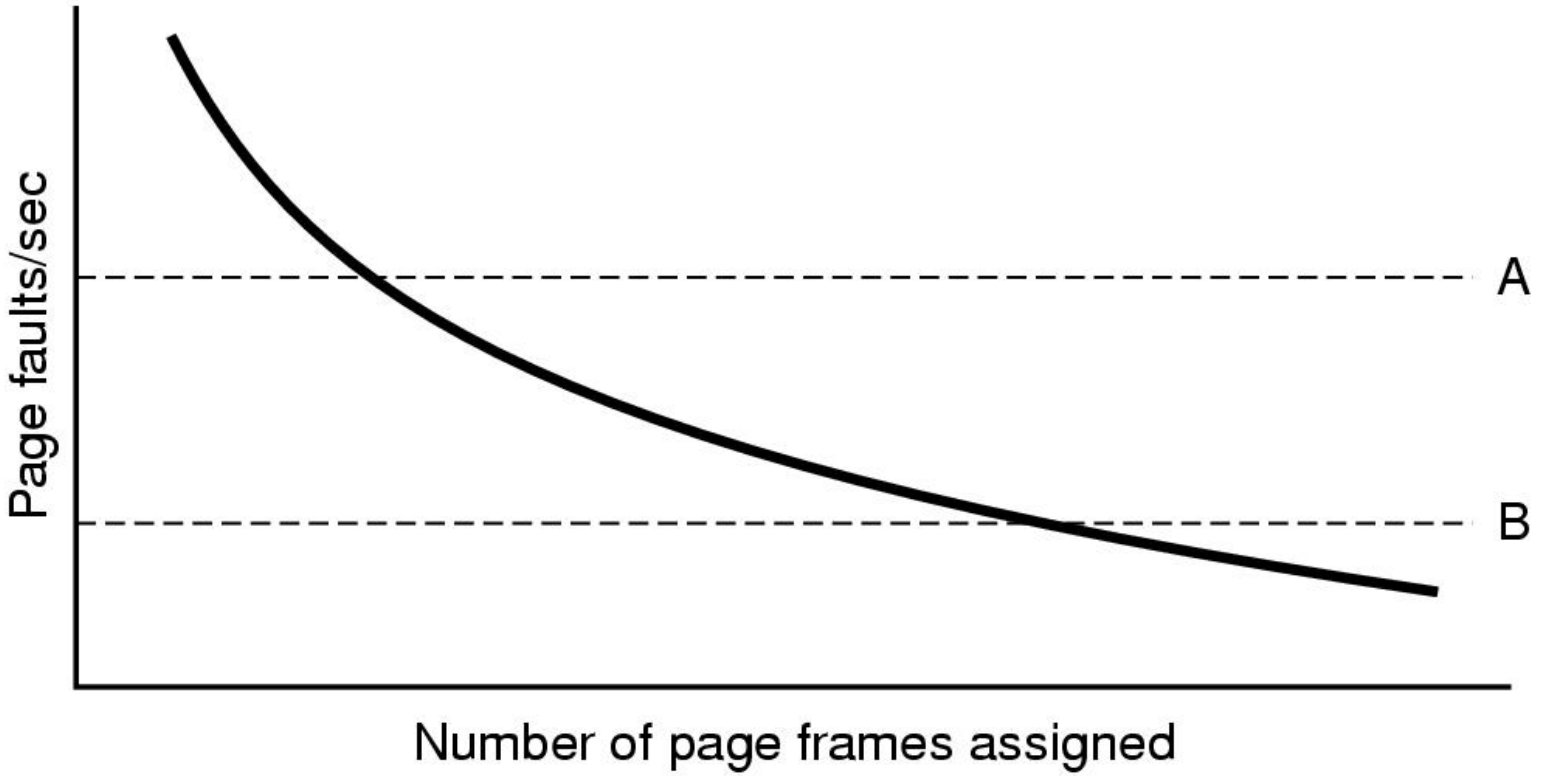
Algoritmo	Commento
Ottimo	Non implementabile, si usa come benchmark
NRU (Not Recently used)	Molto elementare
FIFO	Potrebbe scaricare pagine importanti
LRU (Least Recently Used)	Eccellente ma difficile da implementare
Second Chance	Migliora sensibilmente il FIFO
Clock	Realistico
Working Set	Costoso da implementare
WSClock	Buono ed efficiente

Criteri di progetto per la paginazione
Politiche di Allocazione Locali VS Globali (1)



- Configurazione originale
- Sostituzione con politica locale
- Sostituzione con politica globale

Politiche di Allocazione Locali VS Globali (2)



Tasso di page fault in funzione del numero di pagine fisiche assegnate

Controllo del carico

- A prescindere dalla bontà dello schema adottato, il sistema può comunque andare in thrashing

- Quando l'algoritmo di page-fault frequency (PFF) indica che
  - Alcuni processi necessitano di più memoria
  - E che nessun processo necessita di meno memoria

- Soluzione :
  Ridurre il numero di processi che competono per la memoria
  - Fare lo swap di qualche processo su disco
  - Ridurre il grado di multiprogrammazione

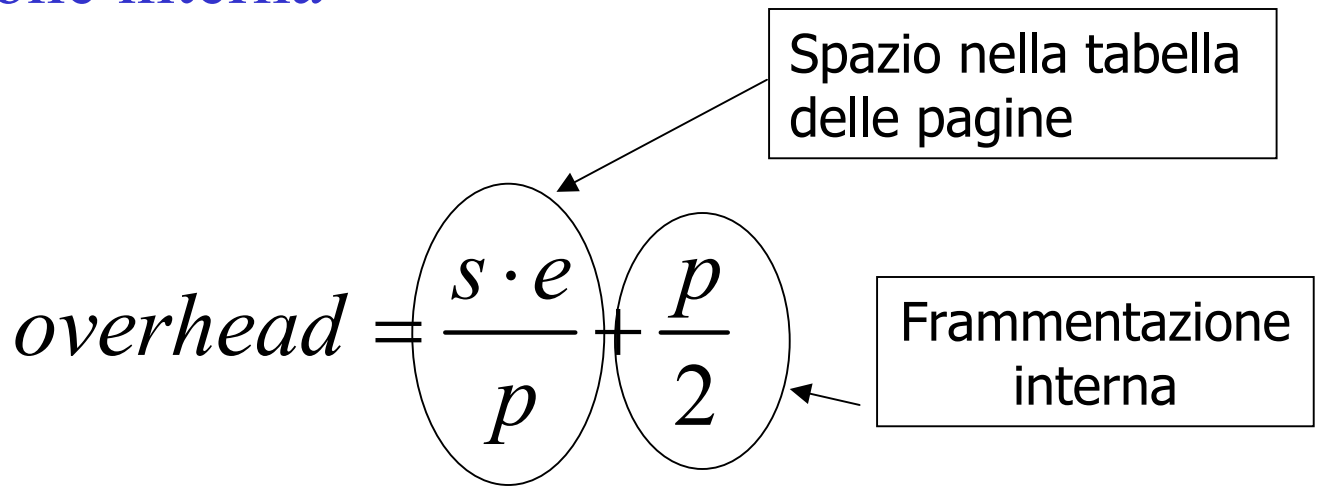
Dimensione delle Pagine (1)

Pagine piccole

- Vantaggi
  - Riduce frammentazione interna
  - si adatta meglio a varie strutture dati e sezione di codice
  - Meno programmi inutilizzati in memoria
- Svantaggi
  - I programmi necessitano di parecchie pagine, tabelle più grandi

Dimensione delle Pagine (2)

- Overhead dovuto alla tabella delle pagine e alla frammentazione interna

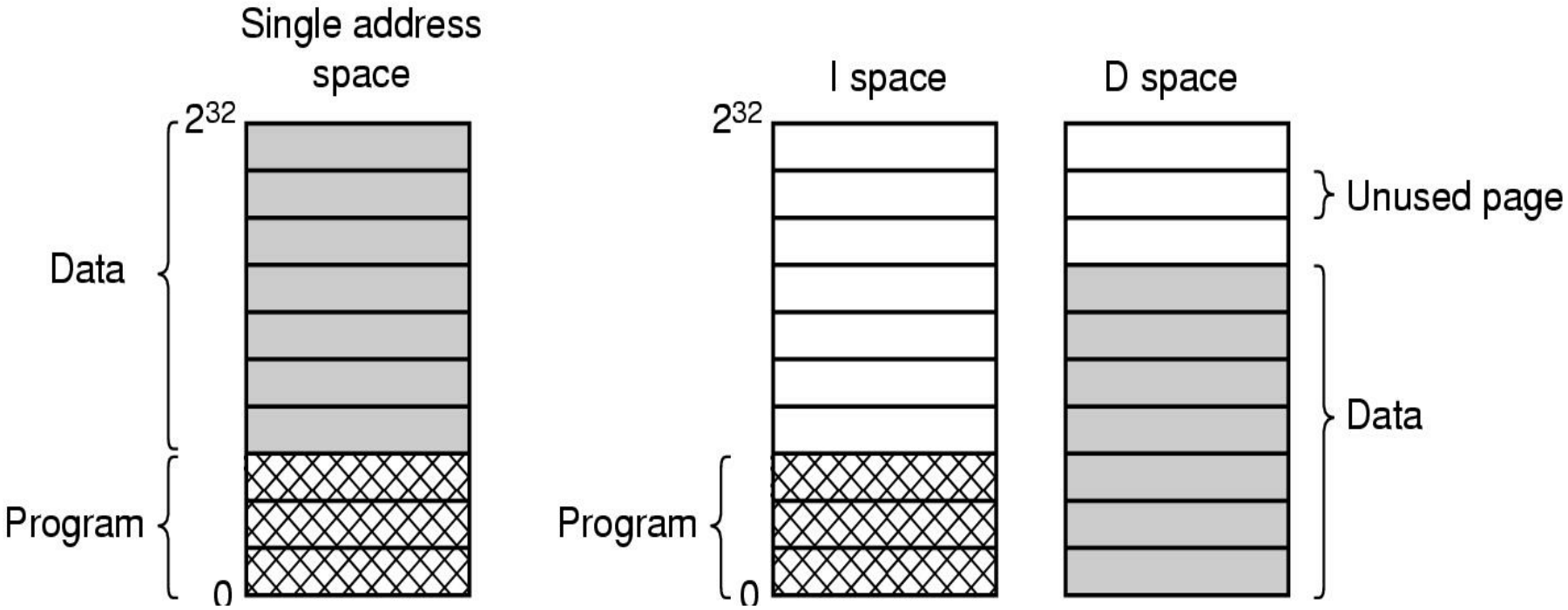


- Dove
  - s = dimensione media di un processo (in bytes)
  - p = dimensione della pagina (in bytes)
  - e = descrittore di pagina (in bytes)

Ottimizzata quando

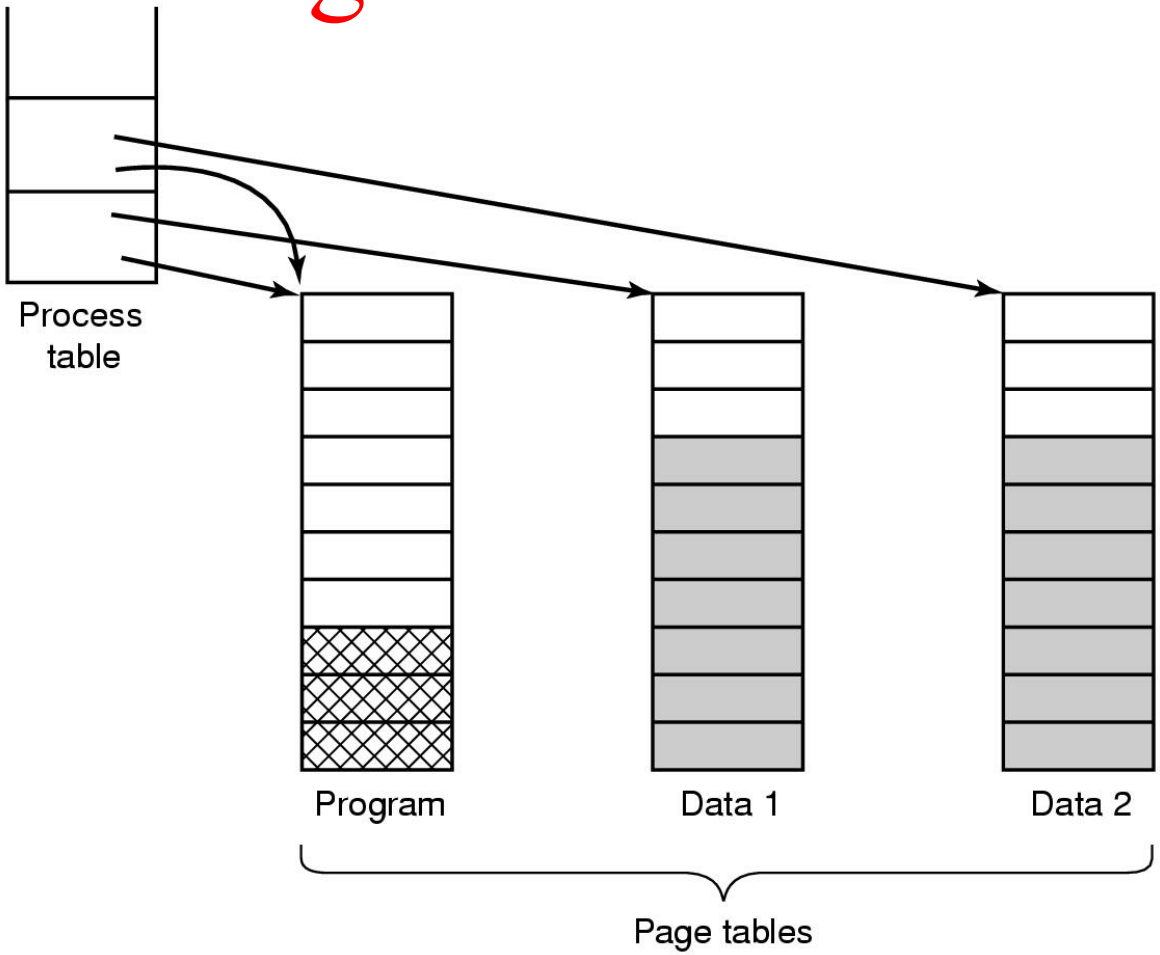
$$p = \sqrt{2se}$$

Separare Istruzioni e Dati



- Uno spazio di indirizzamento
- Spazi separati per istruzioni e dati

Pagine Condivise



Due processi possono condividere lo stesso programma tramite la condivisione dei record della tabella delle pagine associate al programma

Cleaning Policy

- Necessità di un processo in background (demone di paginazione)
  - Analizza periodicamente lo stato della memoria
- Quando troppe poche pagine fisiche sono libere
  - Seleziona una pagina da scaricare usando un'algoritmo di sostituzione