

Sistemi Operativi

AAF - Secondo anno - 3CFU

A.A. 2023/2024

Corso di Laurea in Matematica

La Gestione della Memoria - Parte 4

Annalisa Massini

Dipartimento di Informatica
Sapienza Università di Roma

Argomenti trattati

- 1 Memoria virtuale
 - Memoria virtuale e sistema operativo

Gestione della Memoria: Decisioni

Progettare la porzione di SO che si occupa della gestione della memoria comporta una serie di decisioni

- Usare o no la memoria virtuale?
- Usare solo la paginazione, segmentazione o entrambe?
- Che algoritmi usare per gestire i vari aspetti della gestione della memoria?

I primi due punti dipendono dall'hardware disponibile, mentre l'ultimo punto è responsabilità del SO

Gestione della Memoria: Decisioni

- In ogni caso si vuole minimizzare il tasso di page fault, perchè causano un notevole overhead di tempo
- L'overhead include:
 - la decisione di quali pagine rimpiazzare
 - la scelta del processo da mandare in esecuzione
 - la gestione del process switch
- Quindi, è importante minimizzare la probabilità che il processo in esecuzione faccia riferimento a un'istruzione o un dato non presente in memoria principale generando page fault

Elementi centrali per il progetto del SO

La progettazione della gestione della memoria riguarda principalmente le seguenti politiche:

- Politica di prelievo (*fetch policy*)
- Politica di posizionamento (*placement policy*)
- Politica di sostituzione (*replacement policy*)

Non è possibile stabilire quale delle politiche possibili sia migliore in assoluto

Fetch Policy

- La **politica di prelievo** determina quando una data pagina debba essere portata in memoria principale
- Si usano principalmente due politiche:
 - paginazione su richiesta (*demand paging*)
 - prepaginazione (*prepaging*)
- **N.B.** Quando un processo viene sospeso e swappato in memoria secondaria, tutte le sue pagine vengono spostate e al ritorno tutte le pagine vengono ricaricate

Demand Paging e Prepaging

● Demand paging

- una pagina viene portata in memoria principale nel momento in cui un qualche processo la richiede
- molti page fault nei primi momenti di vita del processo
- man mano che pagine vengono caricate i fault diminuiscono per il principio di località

● Prepaging

- vengono portate in memoria principale più pagine di quelle richieste
- ovviamente, si tratta di pagine vicine a quella richiesta (si può fare efficientemente sul disco)
- non è una politica efficiente se poi le pagine caricate non vengono utilizzate

Placement policy

- La **politica di posizionamento** serve a decidere dove mettere una pagina in memoria principale, *quando c'è almeno un frame libero*
 - se non ci sono frame liberi, allora **replacement policy**
- La pagina può essere messa ovunque, grazie all'hardware per la traduzione degli indirizzi
- Tipicamente, la pagina viene messa nel primo frame libero
 - dove per **primo** si intende il frame con indirizzo più basso

Algoritmi di Sostituzione

I principali algoritmi (di base) per la selezione della pagina da sostituire sono:

- Sostituzione ottima (**Optimal**)
- Sostituzione della pagina usata meno di recente (**LRU: Least Recently Used**)
- Sostituzione a coda (**FIFO: First In First Out**)
- Sostituzione ad orologio (**clock**)

Algoritmi di Sostituzione

- Gli esempi riportati nel seguito usano tutti la stessa sequenza di richieste a pagine:

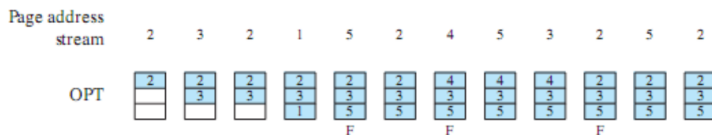
2 3 2 1 5 2 4 5 3 2 5 2

- Si suppone inoltre che ci siano solo 3 frame in memoria principale

Sostituzione Ottima

- Con la politica di sostituzione **ottima** si sostituisce la pagina che verrà richiesta più in là nel tempo
- Ovviamente, non è implementabile perchè il SO non ha conoscenza di ciò che avverrà nel futuro
- È utilizzata come standard per valutare altre politiche
- Infatti visto che nessuna politica può fare meglio dell'ottimo serve a valutare quanto si sia distanti

Sostituzione ottima sull'esempio



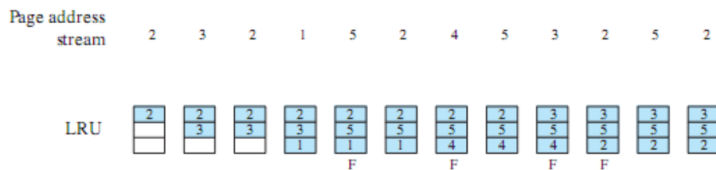
F = page fault occurring after the frame allocation is initially filled

Risultato: 3 page faults

Sostituzione LRU

- Con la politica **LRU** si sostituisce la pagina cui non è stato fatto riferimento per il tempo più lungo
- Basandosi sul principio di località, dovrebbe essere la pagina che ha meno probabilità di essere usata nel prossimo futuro
- L'implementazione è difficile:
 - occorre etichettare ogni frame con il tempo dell'ultimo accesso
 - e poi confrontare tutti i tempi
 - anche per la cache si usa questa tecnica ma è implementata in hardware, cosa che per la memoria secondaria non si può fare

Sostituzione LRU sull'esempio



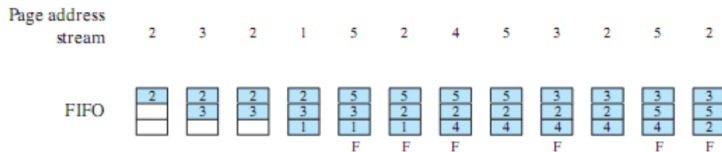
F = page fault occurring after the frame allocation is initially filled

Risultato: 4 page faults (vicino all'ottimo)

Sostituzione FIFO

- Con la politica **FIFO** i frame allocati ad un qualche processo sono trattati come una coda circolare
- Da questa coda, le pagine vengono rimosse a turno (*round robin*)
- L'implementazione è semplice
- Si rimpiazzano le pagine che sono state in memoria per più tempo
 - però non è detto che non servano più: magari alcune di loro hanno molti accessi

Sostituzione FIFO sull'esempio



F = page fault occurring after the frame allocation is initially filled

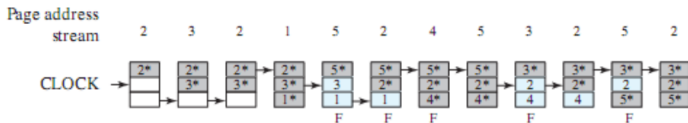
Risultato: 6 page faults

N.B. FIFO non *vede* che le pagine 2 e la 5 sono molto richieste

Sostituzione dell'orologio

- La politica del **clock** è un compromesso tra LRU e FIFO
- Si usa uno **use bit** per ogni frame, per indicare se la pagina caricata nel frame è stata riferita
- Il bit è settato ad 1 quando la pagina viene caricata in memoria principale, e poi rimesso ad 1 per ogni accesso
- Quando occorre sostituire una pagina, il SO cerca il frame come nella FIFO, partendo dall'ultima posizione utilizzata:
 - Se incontra un frame la cui pagina ha lo **use bit** a 1, lo mette a 0 e passa alla successiva
 - Se incontra un frame la cui pagina ha lo **use bit** uguale a 0, seleziona tale frame mettendo lo **use bit** a 1

Sostituzione dell'orologio sull'esempio

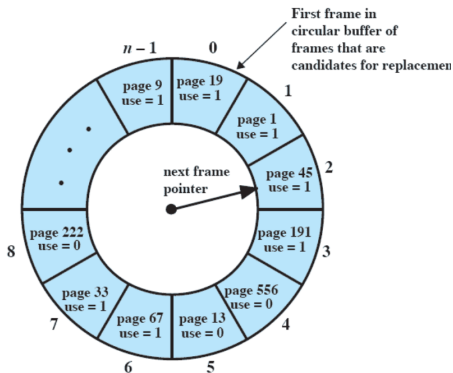


F = page fault occurring after the frame allocation is initially filled

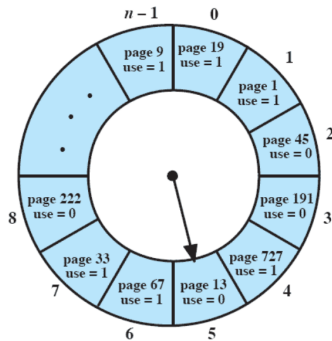
Risultato: 5 page faults

- N.B. La richiesta per pagina 3 (quarto fault) comporta che tutta la coda venga attraversata partendo dal primo frame e tutti gli *use bit* siano messi a 0. Quando si torna al primo frame questo può essere selezionato e lo *use bit* è messo a 1
- N.B. La politica dell'orologio si accorge che le pagine 2 e 5 sono molto richieste

Politica dell'Orologio



(a) State of buffer just prior to a page replacement



(b) State of buffer just after the next page replacement

Algoritmi di sostituzione sull'esempio

Page address stream

2 3 2 1 5 2 4 5 3 2 5 2

OPT

2	2	2	2	2	2	4	4	4	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
			1	5	5	5	5	5	5	5	5

F

F

F

LRU

2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	5	5	5	5	5	5	5	5
			1	1	1	4	4	4	2	2	2

F

F

F

F

FIFO

2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	3	2	2	2	2	2	5	5
			1	1	1	4	4	4	4	4	2

F

F

F

F

F

F

F

CLOCK

2*	2*	2*	2*	5*	5*	5*	5*	3*	3*	3*	3*
	3*	3*	3*	3	2*	2*	2*	2	2*	2	2*
			1*	1	1	4*	4*	4	4	5*	5*

F

F

F

F

F

F = page fault occurring after the frame allocation is initially filled