

Qualche considerazione

- In teoria un processo puo' addirittura cominciare **senza pagine in memoria**, e ad ogni richiesta ne vengono caricate
- La **localita' di riferimento** dovrebbe aiutare
- **Una sola istruzione** puo' anche causare un **multiplo page fault**, uno per l'istruzione stessa e vari per i dati
- Un page fault su dati puo' portare a **rieseguire l'intera istruzione**

Operating System Concepts

9.10

Silberschatz and Galvin©1999

Prestazioni della paginazione su richiesta

- **Frequenza di page fault** $p: 0 \leq p \leq 1$
 - se $p = 0$ non ci sono page faults
 - se $p = 1$ ogni riferimento genera un page fault
- **Tempo di Accesso Effettivo (EAT)**

$$EAT = (1 - p) \times \text{accesso a memoria} + p \times \text{overhead di page fault}$$

Accesso a memoria = 100 nanosecondi

Overhead di page fault = 25 millisecondi

Include anche l'accesso a memoria



$$EAT = 100 + 24.999.900 \times p$$

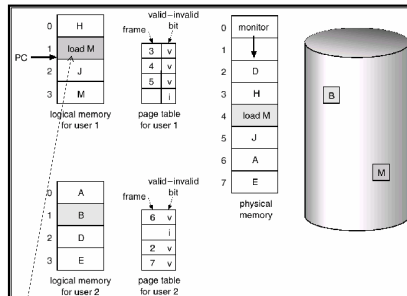
Un rallentamento entro il 10% implica al piu' un page fault ogni 2.500.000 accessi in memoria circa

Operating System Concepts

9.11

Silberschatz and Galvin©1999

Cosa succede se non ci sono frame liberi?!



Sovrallocazione della memoria

Si e' destinato un numero di frame ad ogni processo inferiore al numero di pagine che quel processo potrebbe voler utilizzare

Un'istruzione in questa pagina richiede il caricamento della pagina M

Operating System Concepts

9.12

Silberschatz and Galvin©1999

Soluzione al problema

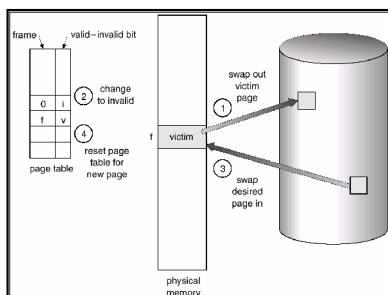
- Si opera la **sostituzione di una pagina** liberando cosi' il frame sul quale essa e' allocata
- Si complica quindi la procedura di page fault in quanto ora richiede un **doppio trasferimento di pagina**
- Si introduce il **dirty bit** per evitare di salvare una pagina su disco se non e' stata modificata

Operating System Concepts

9.13

Silberschatz and Galvin©1999

Quale pagina sara' la vittima?



Algoritmi di sostituzione pagina

Parametri input: stringa dei riferimenti, numero dei frame
Parametro di valutazione: frequenza di page fault

Operating System Concepts

9.14

Silberschatz and Galvin©1999

Algoritmi di sostituzione pagina

- First-In-First-Out
- Ottimale
- Least Recently Used (LRU)
- LRU approssimati

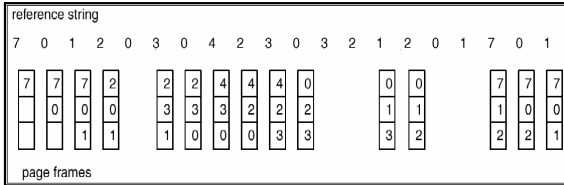


Operating System Concepts

9.15

Silberschatz and Galvin©1999

First-In-First-Out (FIFO)



Operating System Concepts

9.16

Silberschatz and Galvin©1999

FIFO - Anomalia di Belady :

piu' frames \Rightarrow meno page faults

• Stringa dei riferimenti : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

• 3 frames

9 page faults

1	4	5
2	1	3
3	2	4

Come si spiega?

• 4 frames

10 page faults

1	5	4	
2	1	5	
3	2		
4	3		

Anche la sequenza di
prima soffre di tale
anomalia?

Operating System Concepts

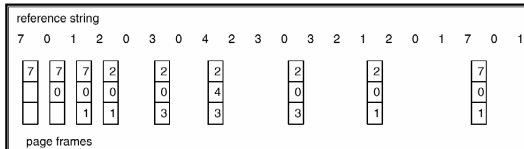
9.17

Silberschatz and Galvin©1999

Algoritmo Ottimale



IDEA : si rimpiazza la pagina che verra' utilizzata piu' tardi



Come possiamo conoscere in anticipo i riferimenti?
(analogia con SJF!)

Viene usata in effetti solo come *parametro di confronto* per la frequenza di page fault (ovviamente minima in questa!)

Operating System Concepts

9.18

Silberschatz and Galvin©1999

... e sull'esempio di prima

• Stringa dei riferimenti : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

• 3 frames

7 page faults

1		3
2		4
3	4	5

• 4 frames

6 page faults

1	4		
2			
3			
4	5		

Operating System Concepts

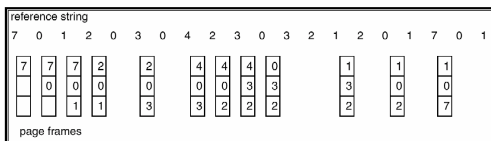
9.19

Silberschatz and Galvin©1999

Least Recently Used (LRU)

Via di mezzo tra il considerare l'istante di arrivo in memoria (il passato: FIFO) e l'utilizzazione della memoria (il futuro: OPT)

Ipotesi : futuro vicino = passato recente \Rightarrow
si sostituisce la pagina utilizzata meno di recente



E' ottimale tra quelle che guardano solo all'indietro !

Operating System Concepts

9.20

Silberschatz and Galvin©1999

Come tener traccia del recente utilizzo delle pagine?



Clock counter

• Ogni pagina ha un *registro associato*

• Ogni volta che si fa un riferimento alla pagina, il valore di un *clock counter logico* viene *copiato nel registro associato* alla pagina

• In caso di sostituzione di pagina si cerca e si seleziona quella con il *valore del registro piu' piccolo*

Operating System Concepts

9.21

Silberschatz and Galvin©1999

Come tener traccia del recente utilizzo delle pagine?

Stack

- Uno *stack di numeri di pagine* viene introdotto
- Ogni volta che si fa un riferimento alla pagina, il *numero di pagina emerge sul top dello stack*
- In caso di sostituzione di pagina si seleziona quella *sul fondo dello stack*
- Questo *evita di dover fare una ricerca di un valore* (come nel caso del clock counter) solo a condizione che una *struttura con doppi puntatori* venga implementata



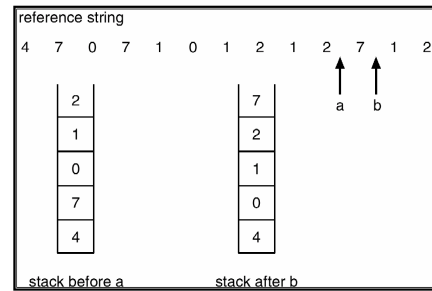
Operating System Concepts

9.22

Silberschatz and Galvin©1999

Come tener traccia del recente utilizzo delle pagine?

Stack



Operating System Concepts

9.23

Silberschatz and Galvin©1999