

Approssimazioni ad LRU

Bit supplementari di riferimento

- Ad ogni pagina viene associato un **bit di riferimento**
- Un **vettore di scorrimento di bit** (p.es., di taglia 8) viene anche mantenuto per ogni pagina
- Quando si fa un riferimento alla pagina **il bit viene settato a 1**
- Ad intervalli regolari tutti i **bit vengono trasferiti nel vettore** mediante uno **shift verso destra**
- In caso di sostituzione di pagina si seleziona quella con **il piu' piccolo valore di vettore**

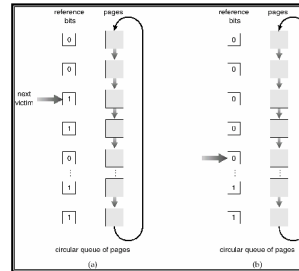
Operating System Concepts

9.24

Silberschatz and Galvin©1999

Approssimazioni ad LRU

Algoritmo di seconda chance



Si tratta in pratica di una strategia **FIFO** che, quando trova il bit di riferimento a 1, **da' una seconda chance** a quella pagina rimettendo il bit a 0 e cercandone un'altra

Si puo' introdurre anche un bit di modifica in modo che le classi di pagine diventino quattro

Operating System Concepts

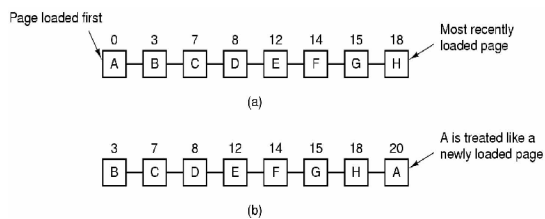
9.25

Silberschatz and Galvin©1999

Approssimazioni ad LRU

Algoritmo di seconda chance

Altro modo di vederla



Operating System Concepts

9.26

Silberschatz and Galvin©1999

Algoritmi con conteggio

- Si tiene un **contatore del numero di riferimenti** che sono stati effettuati per ogni pagina
- Tali contatori devono essere **periodicamente shiftati a destra** per introdurre un fattore di tempo



- **LFU**: rimpiazza la pagina con il piu' piccolo contatore
- **MFU**: rimpiazza la pagina con il piu' grande contatore in quanto **quelle con contatori piccoli potrebbero essere arrivate in memoria da poco** (piuttosto che essere state utilizzate poco!)

Operating System Concepts

9.27

Silberschatz and Galvin©1999

Supporto agli algoritmi di rimpiazzamento pagine

- Si puo' introdurre un **pool di frame liberi** per virtualizzare la cancellazione di una pagina...

... tale pool puo' essere anche utilizzato per ritrovare una pagina appena cancellata



- Si potrebbe anche pensare di **memorizzare sulla memoria di massa a intervalli regolari** tutte le pagine modificate e aggiornarne i bit, in modo da evitare tale overhead in caso di sostituzione

Operating System Concepts

9.28

Silberschatz and Galvin©1999

Allocazione di frame

Ogni processo ha dei **limiti sul numero di frame** che possono essere allocati ad esso

Numero massimo: determinato dall'architettura dell'elaboratore e dal grado di multiprocessing

Numero minimo: determinato dal set di istruzioni

dipende da ---> Numero massimo di frame ai quali una singola istruzione puo' fare riferimento...
... e quindi in parte dai livelli di indirizzione delle istruzioni

Operating System Concepts

9.29

Silberschatz and Galvin©1999

Criteri di allocazione di frame

- Uguale/Fissa
- Proporzionale sulla taglia
- Proporzionale sulla priorit 

Tutte prevedono una **dinamica riallocazione** del totale numero di frame in caso di **creazione o cancellazione di un processo**

Operating System Concepts

9.30

Silberschatz and Galvin 1999

Un importante parametro da tenere in conto nell'allocazione



Sostituzione globale o locale delle pagine ?!

- **Globale** : il frame puo' essere selezionato tra tutti, anche tra quelli di altri processi...
- (... la **taglia di allocazione** di un processo **puo' variare**, ma non si controlla la frequenza di page fault perche' influenzata da fattori esterni)

- **Locale** : il frame deve essere selezionato solo tra quelli allocati al processo stesso...
- (... la **taglia di allocazione** di un processo **e' fissa**, ma si potrebbero non "vedere" all'occorrenza pagine poco usate)

Operating System Concepts

9.31

Silberschatz and Galvin 1999

Thrashing

Se un processo non ha abbastanza pagine, la frequenza di page fault e' molto alta. Questo porta a :

- **Bassa utilizzazione** della CPU
- OS pensa che **si deve aumentare il grado di multiprogrammazione**, e quindi aggiunge un altro processo

... e cos  via...



Thrashing = i processi perdono piu' tempo per la paginazione che per l'esecuzione

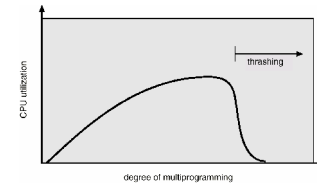


Operating System Concepts

9.32

Silberschatz and Galvin 1999

Diagramma di utilizzazione CPU



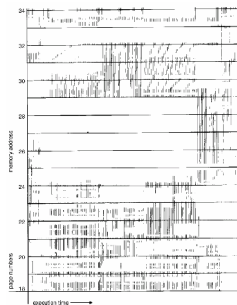
- L'utilizzo della CPU **cresce al crescere del grado di multiprogrammazione** fino ad una soglia
- Superata questa soglia i processi tenderanno a generare una quantita' sempre crescente di page fault e quindi **staranno tutti accodati ad attendere il pager** e l'utilizzo della CPU scendera' molto rapidamente

Operating System Concepts

9.33

Silberschatz and Galvin 1999

Come ci aiuta il concetto di localita' di un processo



Basarsi sulla **localita' dei processi** per prevedere il numero di pagine che occorrono onde evitare sostituzioni frequenti

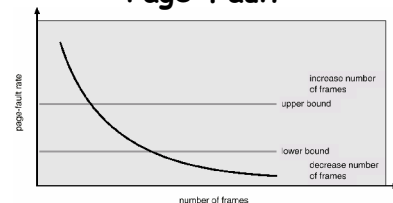
Il **numero di frame** allocati ad un processo puo' quindi **variare nel tempo**

Operating System Concepts

9.34

Silberschatz and Galvin 1999

Modello a controllo di frequenza dei Page-Fault



Si stabilisce un intervallo di **frequenze di page fault accettabile** :

- Se la frequenza corrente e' troppo bassa il processo perde frame
- Se la frequenza corrente e' troppo alta il processo guadagna frame

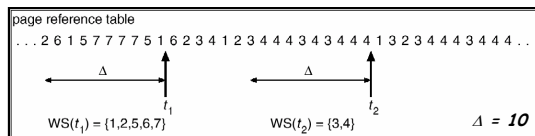
Operating System Concepts

9.35

Silberschatz and Galvin 1999

Modello Working-Set

- $\Delta \equiv$ *finestra* di working set \equiv numero prefissato di riferimenti di pagina, per esempio 10,000 istruzioni
- *Working set* del processo P_i = lista di pagine referenziate nelle ultime Δ istruzioni
- WSS_i = *taglia del working set* di P_i (puo' variare nel tempo pur non variando Δ !)



Operating System Concepts

9.36

Silberschatz and Galvin©1999

Qualche considerazione sul modello

$$D = \sum WSS_i \equiv \text{totale domanda di frame}$$

$$m = \text{numero totale di frame liberi}$$

$$D > m \Rightarrow \text{Thrashing}$$



Per evitare si potrebbe, per esempio, *sospendere uno dei processi* con lo scheduler a medio termine

Operating System Concepts

9.37

Silberschatz and Galvin©1999

Qualche considerazione sul modello

La scelta di Δ

Δ *troppo piccolo*: non include tutta la corrente localita'

Δ *grande*: sovrappone differenti localita'

$\Delta = \infty$: include l'intera sequenza di riferimenti del processo

Operating System Concepts

9.38

Silberschatz and Galvin©1999

Come tenere traccia del Working Set

- Δ e' una finestra in movimento \Rightarrow si approssima
- *Bit di riferimento + timer*
- *Esempio*: $\Delta = 10,000$ riferimenti
 - Il timer interrompe ogni 5000 riferimenti
 - In memoria 2 bits supplementari per ogni pagina
 - Ad ogni interrupt del timer si copiano i bit di riferimento di ogni pagina in uno di quelli supplementari e si resettano a 0
 - Tutte le pagine che hanno almeno uno dei 2 bits a 1 stanno nel working set
- Si puo' migliorare la tecnica *aumentando il numero di bit supplementari e rendendo piu' frequenti le interruzioni del timer* (ex. 10 bits e interruzioni ogni 1000 riferimenti)

Operating System Concepts

9.39

Silberschatz and Galvin©1999