

Sistemi Operativi

7 luglio 2011

Compitino II B

Si risponda ai seguenti quesiti, giustificando le risposte.

- Si spieghi la differenza fra politica di sostituzione locale e politica di sostituzione globale in merito alla gestione della memoria virtuale.

Risposta: (4 punti) Con una politica di sostituzione locale ogni processo può rimpiazzare solo i propri frame. Quindi viene mantenuto fisso il numero di frame allocati ad un processo (anche se ci sono frame liberi). Di conseguenza il comportamento di un processo non è influenzato da quello degli altri processi.

Invece con una sostituzione globale un processo sceglie un frame tra tutti i frame del sistema. Quindi un processo può “rubare” un frame ad un altro. In questo modo si sfrutta meglio la memoria fisica, ma il comportamento di un processo dipende da quello degli altri.

- Si descriva il concetto di *working set* $WS(t, \Delta)$, all'istante t con intervallo Δ .
 - Si consideri la seguente stringa di riferimenti (partendo con $t = 0$):

6 4 2 7 6 5 5 5 1 6 4

Cosa è $WS(10, 3)$, ossia dopo l'ultimo accesso?

- Nel precedente esempio quanti page fault ci sono complessivamente con $\Delta = 4$ (supponendo che in ogni istante si mantenga in memoria esattamente il solo working set)?

Risposta:

- (3 punti) Il working set è un'approssimazione della località del processo, ossia è l'insieme di pagine “attualmente” riferite. In generale $WS(t, \Delta)$ = insieme delle pagine riferite negli accessi $[(t - \Delta + 1), t]$.
- (3 punti) $WS(10, 8) = \{1, 4, 6\}$.
- (3 punti) Si verificano 8 page fault, come risulta dalla seguente simulazione dell'algoritmo:

6	4	2	7	6	5	5	5	1	6	4
6	4	2	7	6	5	5	5	1	6	4
	6	4	2	7	6	6	6	5	1	6
		6	4	2	7	7		5	1	
			6	4	2				5	
p	p	p	p		p			p	p	p

- Si consideri un processo che generi la seguente stringa di riferimenti alle pagine virtuali:

0 2 1 0 4 0 2 1 1 0 5 3 2

- Se il processo ha 3 frame, gestiti LRU, quanti page fault vengono generati?
- Qual è il numero minimo di frame necessario per minimizzare i page fault?

Risposta:

- (3 punti) Simuliamo il funzionamento di LRU nel caso della reference string data:

	0	2	1	0	4	0	2	1	1	0	5	3	2
		0	2	1	0	4	0	2	2	1	0	5	3
			0	2	1	1	4	0	0	2	1	0	5
					2	2	1	4	4	4	2	1	0
											4	2	1
												4	4
	P	P	P		P		P	P			P	P	P

Si verificano quindi nove page fault.

- (3 punti) Il minimo numero di page fault è 6 page fault (perché il processo accede a 6 pagine). Per determinare il numero minimo di frame per avere solo 6 page fault, si può sfruttare la *distance string*, che nel caso in questione risulta essere la seguente:

$\infty \infty \infty 3 \infty 2 4 4 1 3 \infty \infty 5$

Sistemi Operativi

7 luglio 2011

Compitino II B

Si ricorda che la *distance string* rappresenta la distanza fra la posizione di una pagina nel modello e la prima posizione, ovvero, quella nella prima riga della matrice (contando anche la casella di partenza) nel momento in cui la pagina stessa viene riferita. Se una pagina non è presente nella matrice, allora la sua distanza, quando viene riferita è ∞ .

Indichiamo ora con C_i il numero di volte che il numero i compare nella *distance string*; nel caso in questione abbiamo: $C_1 = 1, C_2 = 1, C_3 = 2, C_4 = 2, C_5 = 1, C_\infty = 6$. Indicando poi con m il numero di frame e con n il numero più grande che compare nella *distance string*, indichiamo con $F_m = \sum_{k=m+1}^n C_k + C_\infty$ il numero di page fault che si verificano con m frame e con la *reference string data*. L'intuizione è la seguente: se ho a disposizione m frame i page fault saranno provocati dai riferimenti a pagine che "distanza" almeno $m + 1$ dal top della matrice e dal numero di ∞ (ovvero da riferimenti a pagine non ancora presenti nel modello). Nel nostro caso abbiamo: $F_1 = 12, F_2 = 11, F_3 = 9, F_4 = 7, F_5 = 6$; quindi il numero minimo di frame che minimizza i page fault è 5.

4. Quali sono le operazioni compiute dal driver delle interruzioni?

Risposta: (4 punti) Il driver delle interruzioni è un componente fondamentale del software di un sistema operativo nei sistemi time-sharing e con I/O interrupt driven; infatti esso permette di gestire le interruzioni generate dai dispositivi di I/O, attivando le opportune routine di servizio. I passi eseguiti dal driver delle interruzioni sono i seguenti:

- salvare i registri della CPU,
- impostare un contesto per la procedura di servizio (inizializzare TLB, MMU, stack ecc.),
- inviare un segnale di *acknowledge* al controllore degli interrupt (per avere interrupt annidati),
- copiare la copia dei registri nel PCB,
- eseguire la procedura di servizio che accede al dispositivo,
- eventualmente, cambiare lo stato a un processo in attesa (e chiamare lo scheduler di breve termine),
- organizzare un contesto (TLB, MMU ecc.) per il processo successivo,
- caricare i registri del nuovo processo dal suo PCB,
- continuare l'esecuzione del processo selezionato.

5. Si spieghi come funziona l'allocazione concatenata dei blocchi di un disco. Che problemi possono verificarsi?

Risposta: (4 punti) Lo schema di allocazione concatenata prevede che ogni blocco dati abbia una parte di esso riservata alla memorizzazione dell'indirizzo del blocco successivo (l'ultimo blocco viene marcato con un indirizzo non valido, e.g., -1). In questo modo l'implementazione di un file consiste in una lista di blocchi concatenati ed è sufficiente memorizzare per ogni file nella directory entry il blocco iniziale e quello finale. I pregi consistono essenzialmente nella semplicità dell'implementazione, mentre gli aspetti negativi sono la fragilità dello schema (se si corrompe un puntatore della catena, si perde tutta la parte del file da quel punto in poi) e la mancanza di supporto dell'accesso diretto (seek); infatti per accedere al blocco i -esimo bisogna prima scorrere la catena di puntatori dei precedenti $i-1$ blocchi.

Questo schema non soffre di frammentazione esterna, dato che i blocchi assegnati ad un file possono essere sparsi ovunque nel disco (non devono necessariamente essere contigui); infatti, per aggiungere un blocco ad un file è sufficiente aggiornare il puntatore dell'ultimo blocco in modo che punti ad uno dei blocchi liberi su disco (marcando l'indirizzo di quest'ultimo come indirizzo finale, e.g., -1 ed aggiornando la directory entry corrispondente).

6. Si consideri un disco gestito con politica LOOK. Inizialmente la testina è posizionata sul cilindro 40; lo spostamento ad una traccia adiacente richiede 2 ms. Al driver di tale disco arrivano richieste per i cilindri 90, 45, 40, 60, 55, rispettivamente agli istanti 0 ms, 20 ms, 30 ms, 40 ms, 80 ms. Si trascuri il tempo di latenza.

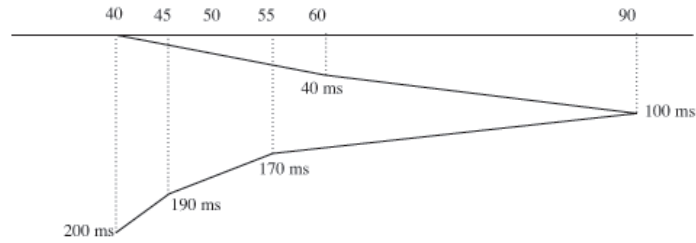
1. In quale ordine vengono servite le richieste?

Sistemi Operativi
7 luglio 2011
Compitino II B

2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le cinque richieste in oggetto?

Risposta:

1. (3 punti) Le richieste vengono servite nell'ordine 60, 90, 55, 45, 40:



2. (2 punti) Il tempo di attesa medio per le cinque richieste in oggetto è
$$\frac{(100-0)+(190-20)+(200-30)+(40-40)+(170-80)}{5} = \frac{100+170+170+0+90}{5} = \frac{530}{5} = 106 \text{ ms.}$$