

Esercizio: memoria virtuale

Si consideri un sistema di gestione della memoria virtuale basato su paginazione.

Si discuta, in non più di 5 righe, la congettura secondo la quale l'aumento di capacità della memoria fisica comporti una diminuzione delle occorrenze di accesso a pagina mancante (page fault). Si discuta ora ciò che accade nella situazione descritta di seguito e lo si metta in relazione con la congettura di cui sopra:

Una memoria fisica M1, inizialmente vuota, ha capacità pari a 3 pagine, a fronte di una memoria virtuale V di ampiezza 5 pagine, numerate da 0 a 4. Il gestore della memoria virtuale adotta la politica di rimpiazzo detta LRU (least recently used) implementata in versione a basso costo (lazy), dove cioè non si aggiorna la lista ordinata per frequenza d'accesso a seguito di un accesso avvenuto con successo (hit).

L'esecuzione del programma causa la sequenza di chiamata di pagine:

0,1,2,3,0,1,4,0,1,2,3,4.

Si mostri l'effetto di tale sequenza sul contenuto di M1, indicando ove essa provochi page fault. Si mostri poi l'effetto della medesima sequenza sulla memoria fisica M2, inizialmente vuota, e di capacità pari a 4 pagine.

Soluzione

Per quanto intuitivamente plausibile, non è sempre il caso che all'aumento di capacità della memoria fisica corrisponda una diminuzione del numero di eccezioni di pagina: un certo numero di fattori concomitanti (quali la politica di rimpiazzo e la sequenza di chiamata) deve concorrere affinché la congettura sia verificata.

Questa incongruenza è nota come anomalia di Belady, dal nome dello studioso che per primo la rilevò.

Come mostrato di seguito, la sequenza indicata causa 9 eccezioni di pagina (marcate con '*') su M1, meno capiente, e 10 su M2, più capiente:

(VEDI LUCIDO SUCCESSIVO)

Soluzione

pagina richiesta	0	1	2	3	0	1	4	0	1	2	3	4
accesso più recente	0	1	2	3	0	1	4	4	4	2	3	3
		0	1	2	3	0	1	1	1	4	2	2
accesso meno recente			0	1	2	3	0	0	0	1	4	4
eccezione di pagina	*	*	*	*	*	*	*	*	*	*	*	*
mancato riordinamento di lista								↑	↑			↑
					<i>M1</i>							
pagina richiesta	0	1	2	3	0	1	4	0	1	2	3	4
accesso più recente		0	1	2	3	3	4	0	1	2	3	4
			0	1	2	2	3	4	0	1	2	3
accesso meno recente				0	1	1	2	3	4	0	1	2
eccezione di pagina	*	*	*	*	*	*	*	*	*	*	*	*
mancato riordinamento di lista					↑	↑						
					<i>M2</i>							

Si verifichi, per esercizio, cosa succede nel caso si utilizzi una implementazione rigorosa della LRU

Esercizio: architettura di CPU

L'architettura di instruction set a registri generali di tipo registro-registro è generalmente nota con il nome Load/Store. Si illustrino, in modo sintetico, le caratteristiche salienti di tale architettura e la relazione esistente tra le due denominazioni.

Risposta:

Nella classe di architetture di instruction set a registri generali detta registro-registro tutte le istruzioni operative (ossia quelle operanti su dati) agiscono solo su registri della CPU. Ciascuna sequenza operativa si apre con il prelievo dell'operando da memoria (Load) e si conclude con il deposito del risultato in memoria (Store). Tale coppia di istruzioni è la sola abilitata ad operare sulla memoria. Per questo motivo la denominazione Load/Store descrive (per complemento) la classe registro-registro.

Esercizio: indirizzamento

Si illustrino le differenze sintattiche e semantiche tra la modalità di indirizzamento detta "immediata" e quella detta "a registri". Si rappresenti l'effetto della prima mediante una sequenza di istruzioni (pseudo)assembler che utilizzino la seconda modalità.

Risposta:

La modalità di indirizzamento detta "immediata" è del tipo:

add R1, k **! [R1]+k → [R1]**

Quella detta "a registri" è invece del tipo:

add R1, R2 **! [R1]+[R2] → [R1]**

L'effetto della prima tramite la seconda è ottenibile mediante la seguente sequenza di istruzioni:

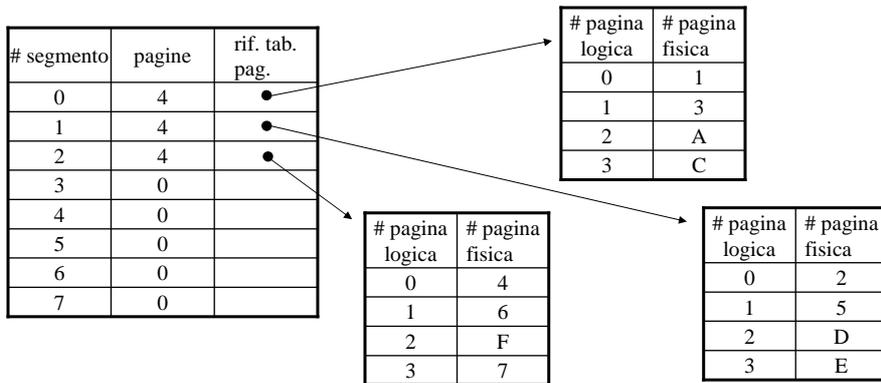
load k, R2 **! k → [R2]**
add R1, R2 **! [R1]+[R2] → [R1]**

Esercizio: memoria virtuale

Si assuma una memoria segmentata con paginazione, dove ogni indirizzo logico può fare riferimento a non più di 8 segmenti. Ogni segmento può contenere non più di 2047 pagine, ognuna di 4KB. L'unità di indirizzamento logico è 1B e la memoria fisica è di 16MB.

Dati i seguenti due indirizzi logici, in esadecimale, 1001EC3 e 080281B e le tabelle dei segmenti e delle pagine mostrate nel lucido successivo, si dica a quali indirizzi fisici corrispondono, assumendo che la prima pagina fisica sia memorizzata a partire dall'indirizzo 0.

Esercizio: memoria virtuale



Esercizi

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 26

Soluzione

Per l'indirizzamento dello spazio logico occorrono 3 bit per individuare il segmento, 11 bit per selezionare la pagina in un segmento, e 12 per individuare la locazione in una pagina di 4KB. Quindi in totale abbiamo 26 bit. Un indirizzo fisico ha bisogno di 24 bit (per indirizzare tutte le 16M locazioni da 1B), di cui i 12 meno significativi per individuare la locazione fisica all'interno della pagina fisica (offset).

Se rappresentiamo in binario (su 26 bit) l'indirizzo 1001EC3 otteniamo:

01 0000 0000 0001 1110 1010 0011

di cui i primi 3 bit rappresentano il segmento (quindi 2), i successivi 11 bit rappresentano la pagina nel segmento (quindi 1) e i rimanenti 12 bit rappresentano l'offset.

Dobbiamo generare un indirizzo fisico di 24 bit. I primi bit sono generati usando l'indirizzo fisico alla riga di pagina 1 della tabella puntata dal segmento 2 in tabella dei segmenti (quindi 6 esadecimale) e completandolo con l'offset (quindi EC3 esadecimale), ottenendo in binario (su 24 bit)

0000 0000 0110 1110 1010 0011

Di seguito riportiamo la traduzione in esadecimale per i due indirizzi logici

indirizzo logico		indirizzo fisico
1001EC3	segmento 2, pagina 1	006EC3
080281B	segmento 1, pagina 2	00D81B

Esercizi

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 27

Esercizio: processi

Sia dato un gestore di processi con ordinamento a divisione di tempo, con quanto di tempo fissato in 4 ms, e selezione a priorità, dove il processo a priorità più elevata ha precedenza sugli altri processi pronti. Siano dati i seguenti processi, dei quali si indicano, rispettivamente, la priorità, il tempo di esecuzione richiesto in ms ed il tempo di attivazione relativo al tempo 0 di inizio delle attività:

	priorità	tempo esec.	tempo attivaz.
A	3	13	0
B	5	9	6
C	1	21	2

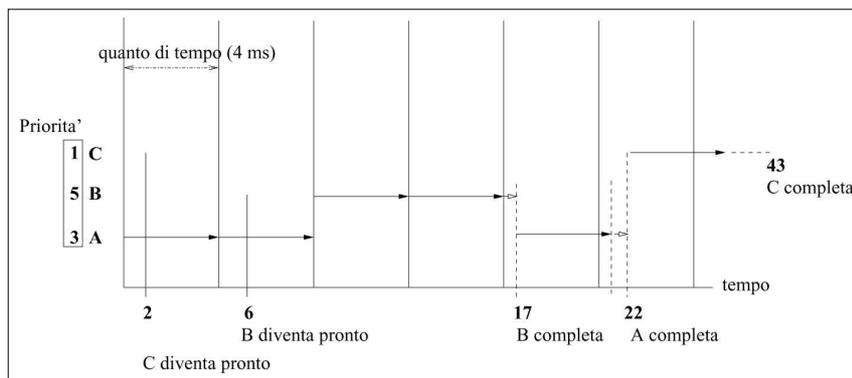
Si mostri l'ordinamento di tali processi come effettuato dal suddetto gestore, assumendo una gestione che eviti lo spreco di quanto di tempo assegnati ma utilizzati solo parzialmente.

Esercizi

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 28

Soluzione



Esercizi

Architettura degli Elaboratori - 1 - A. Sperduti

Pagina 29