

Sistemi Operativi

16 maggio 2012

Compito A

Si risponda ai seguenti quesiti, giustificando le risposte.

- Una politica di scheduling a code multiple con feedback evita la starvation promuovendo un processo ad un livello di priorità più alto se ha speso 3 secondi nel suo attuale livello di priorità senza essere schedulato. Commentare vantaggi e svantaggi dei seguenti metodi di implementazione della promozione:

- (a) promuovere un processo al massimo livello di priorità;
- (b) promuovere un processo al successivo livello di priorità.

Risposta:

- (a) (3 punti) Con questo tipo di promozione si tende ad avere presto tutti i processi al livello massimo. Infatti più processi vanno al livello massimo, più essi bloccano altri processi che quindi andranno a loro volta al livello massimo. In pratica la soluzione degenera in FCFS.
- (b) (3 punti) Questa sembra una politica più equilibrata: può non essere sufficiente a permettere l'esecuzione di un processo, ma allora quest'ultimo verrà ulteriormente promosso.

- Si considerino due processi concorrenti P_i e P_j che eseguono la propria sezione critica in alternanza stretta. Il codice di P_i è il seguente:

```

while (TRUE) {
    while (turn ≠ i) no-op;
    sez. critica
    turn := j;
    sez. non critica
};

```

dove `turn` è una variabile inizializzata a i e il codice di P_j è analogo a quello di P_i .

Si scriva un codice alternativo per i processi P_i e P_j , che utilizza i semafori anziché la variabile `turn` per garantire l'alternanza stretta.

Risposta: (4 punti) Utilizziamo due semafori binari sem_i e sem_j , tali che

$$sem_i = \begin{cases} 1 & (\text{è il turno di } P_i) \\ 0 & (\text{non è il turno di } P_i) \end{cases}$$

$$sem_j = \begin{cases} 1 & (\text{è il turno di } P_j) \\ 0 & (\text{non è il turno di } P_j) \end{cases}$$

I due semafori vengono inizializzati come segue:

```

sem_i := 1
sem_j := 0
dopodichè i due processi eseguono il codice seguente:
Pi:
while (TRUE) {
    down(sem_i);
    sez. critica
    up(sem_j);
    sez. non critica
};
Pj:
while (TRUE) {
    down(sem_j);
    sez. critica
    up(sem_i);
    sez. non critica
};

```

- Nel seguente sistema:

<i>Richieste massime</i>			<i>Risorse allocate</i>			<i>Totali allocate</i>			<i>Totali esistenti</i>		
	R_1	R_2	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_1	3	6	8	2	2	3					
P_2	4	3	3	2	0	3	5	4	10	7	7
P_3	3	4	4	1	2	4					

Sistemi Operativi

16 maggio 2012

Compito A

- (a) È sicuro l'attuale stato di allocazione?
- (b) Sarebbero accettate dall'algoritmo del banchiere le seguenti richieste nello stato attuale?
1. Il processo P_1 chiede $(1, 1, 0)$.
 2. Il processo P_3 chiede $(0, 1, 0)$.
 3. Il processo P_2 chiede $(0, 1, 0)$.

Risposta: Per rispondere alle domande, calcoliamo la matrice R delle richieste (*Richieste massime – Risorse allocate*) ed il vettore A delle risorse disponibili (*Totali esistenti – Totali allocate*):

Richieste (R)

	R_1	R_2	R_3
P_1	1	4	5
P_2	2	3	0
P_3	2	2	0

$$A = (2, 3, 0)$$

- (a) (3 punti) L'attuale stato di allocazione è sicuro in quanto è possibile schedulare la sequenza di esecuzione sicura P_2, P_3, P_1 .
- (b) (3 punti)
1. Se venisse accettata la richiesta $(1, 1, 0)$ del processo P_1 , i valori di *Risorse allocate*, R e A sarebbero i seguenti:

<i>Risorse allocate</i>			<i>Richieste (R)</i>				
R_1	R_2	R_3	R_1	R_2	R_3		
P_1	3	3	3	P_1	0	3	5
P_2	2	0	3	P_2	2	3	0
P_3	1	2	4	P_3	2	2	0

$$A = (1, 2, 0)$$

La richiesta deve quindi essere respinta, dato che non esiste alcuna riga di R minore od uguale al valore di A (i.e., lo stato non è sicuro).

2. Se venisse accettata la richiesta $(0, 1, 0)$ del processo P_3 , i valori di *Risorse allocate*, R e A sarebbero i seguenti:

<i>Risorse allocate</i>			<i>Richieste (R)</i>				
R_1	R_2	R_3	R_1	R_2	R_3		
P_1	2	2	3	P_1	1	4	5
P_2	2	0	3	P_2	2	3	0
P_3	1	3	4	P_3	2	1	0

$$A = (2, 2, 0)$$

La richiesta verrebbe accettata, dato che esiste la sequenza di esecuzione sicura P_3, P_2, P_1 .

3. Se venisse accettata la richiesta $(0, 1, 0)$ del processo P_2 , i valori di *Risorse allocate*, R e A sarebbero i seguenti:

<i>Risorse allocate</i>			<i>Richieste (R)</i>				
R_1	R_2	R_3	R_1	R_2	R_3		
P_1	2	2	3	P_1	1	4	5
P_2	2	1	3	P_2	2	2	0
P_3	1	2	4	P_3	2	2	0

$$A = (2, 2, 0)$$

La richiesta verrebbe accettata, dato che esiste la sequenza di esecuzione sicura P_3, P_2, P_1 .

4. Si supponga che una macchina abbia indirizzi virtuali a 48 bit e indirizzi fisici a 32 bit.
 - (a) Se le pagine sono di 4 KB, quante voci ci sarebbero nella tabella delle pagine se fosse a un solo livello? Si dia una spiegazione.
 - (b) Si supponga che lo stesso sistema abbia un TLB con 32 voci. Inoltre, si assuma che un programma contenga istruzioni che stanno esattamente in una pagina e legga sequenzialmente elementi interi lunghi da un array che si estende su migliaia di pagine. Quanto è indicato il TLB in questo caso?

Risposta:

Sistemi Operativi
16 maggio 2012
Compito A

- (a) (2 punti) Se le pagine sono di 4 KB, servono 12 bit per rappresentare l'offset all'interno di una di esse ($2^{12} = 4096$). Quindi restano $48-12=36$ bit per il numero di pagina virtuale: conseguentemente ci saranno $2^{36} = 64G$ entry nella tabella delle pagine nel caso di una paginazione ad un livello.
- (b) (3 punti) È poco indicato perché la lettura (sequenziale) degli elementi dell'array coinvolge più di 32 voci e quindi si avranno continui fallimenti accedendo al TLB, rendendo di fatto inutile la presenza della memoria associativa se non per la singola voce facente riferimento alla pagina delle istruzioni.
5. Un piccolo computer ha 4 frame. Al primo ciclo del clock, i bit R sono 0111 (la pagina 0 è 0, le restanti 1). Ai cicli successivi i valori sono 1011, 1010, 1101, 0010, 1010, 1100 e 0001. Si forniscano i valori dei 4 contatori dopo l'ultimo intervallo, considerando l'uso dell'algoritmo LRU aging con un contatore a 8 bit.

Risposta: (5 punti) Applicando l'algoritmo di LRU aging (shiftando a destra i bit ad ogni ciclo del clock), si ottiene la seguente configurazione:

P_0	0	1	1	0	1	1	1	0
P_1	0	1	0	0	1	0	0	1
P_2	0	0	1	1	0	1	1	1
P_3	1	0	0	0	1	0	1	1

Verrà quindi scelta la pagina P_2 in quanto quella con la configurazione di bit che rappresenta il numero minore.

6. Per quanto riguarda l'implementazione del file system, spiegare come funziona la gestione dello spazio libero tramite bitmap.

Risposta: (3 punti) Utilizzando una bitmap per gestire lo spazio libero, ogni singolo bit rappresenta un blocco del disco: se il blocco è allocato ad un file il bit è 0, se il blocco è libero il bit è 1. Il vantaggio di questa codifica è essenzialmente l'efficienza nel trovare il primo blocco libero dato che la maggior parte delle CPU più diffuse mettono a disposizione delle istruzioni macchina che forniscono l'offset del primo bit a 1 in una parola. In questo modo il numero del primo blocco libero può essere calcolato come segue:

$$(\text{numero di bit in una parola}) \times (\text{numero delle parole con tutti i bit } 0) + \text{offset del primo bit a 1}$$

Lo svantaggio è che la bitmap deve essere tenuta in memoria per un utilizzo veloce e quindi può portare ad uno spreco di quest'ultima se il disco è di dimensioni ragguardevoli.

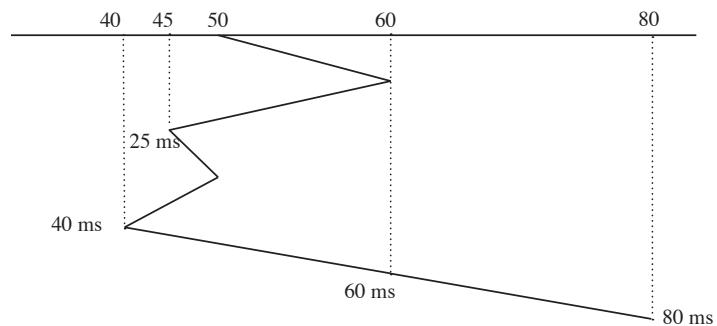
7. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 50; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 80, 45, 40, 60, rispettivamente agli istanti 0 ms, 10 ms, 30 ms, 40 ms. Si trascuri il tempo di latenza.

1. In quale ordine vengono servite le richieste?
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 quattro richieste in oggetto?

Risposta:

1. (3 punti) Le richieste vengono servite nell'ordine 45, 40, 60, 80:

Sistemi Operativi
16 maggio 2012
Compito A



2. (2 punti) Il tempo di attesa medio per le quattro richieste in oggetto è

$$\frac{(25-10)+(40-30)+(60-40)+(80-0)}{4} = \frac{15+10+20+80}{4} = \frac{125}{4} = 31,25 \text{ ms.}$$