

Sistemi Operativi

23 giugno 2011

Compito B

Si risponda ai seguenti quesiti, giustificando le risposte.

1. (a) In quali situazioni può essere attivato lo scheduling della CPU?
- (b) Quando un algoritmo di scheduling è preemptive? Quali sono i vantaggi e gli svantaggi di un algoritmo preemptive?

Risposta:

1. (3 punti) Lo scheduling della CPU può essere attivato nelle seguenti circostanze:
 - (a) un processo viene creato ed entra nella coda dei pronti;
 - (b) un processo passa dallo stato di esecuzione allo stato di attesa;
 - (c) un processo passa dallo stato di esecuzione allo stato pronto;
 - (d) un processo passa dallo stato di attesa allo stato pronto;
 - (e) un processo termina.
2. (3 punti) Un algoritmo di scheduling si dice preemptive se può interrompere l'esecuzione di un processo a favore di un altro processo e può quindi essere attivato ogni volta che un processo passa nella coda dei pronti (coda ready), oltre che ovviamente anche in altri casi. I vantaggi di un algoritmo preemptive sono essenzialmente dei tempi di risposta migliori e la garanzia che nessun processo riesca a monopolizzare la CPU senza rilasciarla. Gli svantaggi sono relativi alla condivisione dei dati fra processi; infatti, se un processo sta manipolando dei dati utilizzati anche da altri processi e viene prelazonato c'è il rischio che questi rimangano in uno stato inconsistente e generino così degli errori. Per evitare tutto ciò è necessario un attento uso di primitive come mutex e semafori per garantire un accesso esclusivo e corretto alle risorse condivise.
2. Si consideri un sistema con scheduling a priorità con tre code, A, B, C, di priorità decrescente, con prelazione tra code. Le code A e B sono round robin con quanto di 10 e 15 ms, rispettivamente; la coda C è FCFS. Se un processo nella coda A o B consuma il suo quanto di tempo, viene spostato in fondo alla coda B o C, rispettivamente.
 - (a) Di quale tipo di algoritmo di scheduling si tratta?
 - (b) Nelle code A, B, C entrano i seguenti processi:

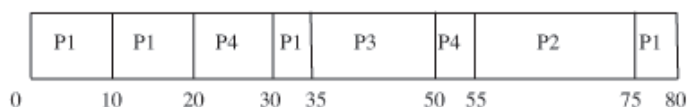
	coda	arrivo	burst
P_1	A	0	30ms
P_2	C	5	20ms
P_3	B	15	15ms
P_4	A	20	15ms

Si determini:

1. il diagramma di GANTT relativo all'esecuzione dei quattro processi;
2. il tempo di attesa medio;
3. il tempo di reazione medio.

Risposta:

- (a) (2 punti) Si tratta di un algoritmo di scheduling a code multiple con feedback negativo.
- (b) (4 punti) Assunzione: un processo prelazonato viene posto all'inizio della sua coda.
 1. Il diagramma di Gantt è il seguente:



2. Tempo di attesa medio = $\frac{50+50+20+20}{4} = \frac{140}{4} = 35$ ms.
3. Tempo di reazione medio = $\frac{0+50+20+0}{4} = \frac{70}{4} = 17,5$ ms.

Sistemi Operativi

23 giugno 2011

Compito B

3. Si descriva il fenomeno del *thrashing* e si indichino tecniche per ovviare a tale fenomeno.

Risposta: (4 punti) Quando la memoria fisica libera (e quindi il numero di frame liberi) è insufficiente a contenere il working set corrente di un processo, quest'ultimo comincerà presto a generare parecchi page fault, rallentando considerevolmente la propria velocità d'esecuzione. Quando parecchi processi cominciano ad andare in thrashing, ovvero a spendere più tempo per la paginazione che per l'esecuzione, il sistema operativo potrebbe erroneamente essere indotto a dedurre che sia necessario aumentare il grado di multiprogrammazione (dato che la CPU rimane per la maggior parte del tempo inattiva a causa dell'intensa attività di I/O). In questo modo vengono avviati nuovi processi che però, a causa della mancanza di frame liberi, cominceranno a loro volta ad andare in thrashing: in breve le prestazioni del sistema collassano fino ad indurre l'operatore a dover terminare forzatamente alcuni processi.

Per ovviare a tale fenomeno si può aumentare la memoria fisica oppure cercare di approssimare correttamente le località dei processi controllandone il working set ed allocando loro frame sufficienti per coprirlo. Quindi, alla creazione di un nuovo processo, questo viene ammesso nella coda ready solo se ci sono frame liberi sufficienti per coprire il suo working set. Altrimenti si sospende uno dei processi per liberare la sua memoria per il nuovo processo (lo scheduling di medio termine diminuisce il grado di multiprogrammazione). In questo modo si impedisce il thrashing, massimizzando nel contempo l'uso della CPU.

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

7 5 6 7 7 6 6 1 1 5 3

Cosa è $WS(10, 8)$, ossia dopo l'ultimo accesso?

- (c) Nel precedente esempio quanti page fault ci sono complessivamente con $\Delta = 3$ (supponendo che in ogni istante si mantenga in memoria esattamente il solo working set)?

Risposta:

- (a) (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 $[(t - \Delta + 1), t]$.
 (b) (2 punti) $WS(10, 8) = \{1, 3, 5, 6, 7\}$.
 (c) (3 punti) Si verificano 6 page fault, come risulta dalla seguente simulazione dell'algoritmo:

7	5	6	7	7	6	6	1	1	5	3
7	5	6	7	7	6	6	1	1	5	3
		7	5	6	6	7	7	6	6	1
			7	5						1
p	p	p					p	p		p

5. Si spieghi come funziona l'allocazione contigua dei blocchi di un disco. Che problemi può comportare?

Risposta: (4 punti) Lo schema di allocazione contigua prevede che ogni file occupi un insieme di blocchi contigui sul disco. I vantaggi sono la semplicità in quanto basta conoscere il blocco iniziale e la lunghezza per accedere al file; inoltre l'accesso random è facile da implementare. I difetti principali sono l'impossibilità per i file di crescere (a meno di deframmentazione) e la frammentazione esterna.

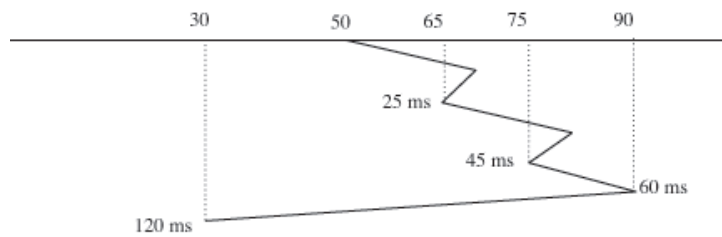
6. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 50; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 90, 65, 75, 30, rispettivamente agli istanti 0 ms, 20 ms, 40 ms, 55 ms. Si trascuri 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?

Sistemi Operativi
23 giugno 2011
Compito B

Risposta:

1. (3 punti) Le richieste vengono servite nell'ordine 65, 75, 90, 30:



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

$$\frac{(60-0)+(25-20)+(45-40)+(120-55)}{4} = \frac{60+5+5+65}{4} = \frac{135}{4} = 33,75 \text{ ms.}$$