

Sistemi Operativi
5 dicembre 2006
Compito A

Si risponda ai seguenti quesiti, giustificando le risposte.

1. Che cos'è la multiprogrammazione? Si può realizzare su un sistema monoprocesso?

Risposta: La nozione di multiprogrammazione prevede la possibilità di mantenere in memoria più processi contemporaneamente in modo da mantenere la CPU occupata il maggior tempo possibile. Infatti, non appena un processo richiede al sistema operativo di eseguire un'operazione di I/O, quest'ultimo può dedicare l'utilizzo della CPU ad un altro processo, invece di attendere passivamente il completamento della suddetta operazione. Questa tecnica è realizzabile anche su un sistema monoprocesso.

2. (a) Si dia la definizione di thread e si descriva come i thread sono rappresentati nei sistemi operativi.
(b) Si diano due esempi di situazioni in cui i sistemi a thread sono vantaggiosi rispetto ai processi tradizionali.

Risposta:

- (a) I thread sono unità di esecuzione all'interno di uno stesso processo; ogni thread è caratterizzato da un proprio PC, da un proprio insieme dei valori dei registri, da un proprio stack ed un proprio stato di esecuzione, mentre tutte le risorse rimanenti (spazio in memoria, file aperti ecc.) vengono condivise fra thread appartenenti allo stesso processo. Vi sono due possibili implementazioni dei thread: a livello utente ed a livello kernel. Nel primo caso ogni processo mantiene una tabella dei thread propria in cui viene tenuta traccia di tutti i thread del processo (PC, valori dei registri, stack, stato di esecuzione). Nel caso invece di un'implementazione a livello kernel viene tenuta, a fianco della tabella dei processi, una tabella dei thread nello spazio di memoria del nucleo del sistema operativo (quindi c'è un'unica tabella per tutti i thread in esecuzione nel sistema). Ovviamente nel primo caso si ha un'efficienza maggiore che nel secondo in quanto tutte le operazioni sui thread vengono eseguite in user space, per mezzo di un'apposita libreria, mentre nel secondo caso bisogna ricorrere ad opportune chiamate di sistema passando ogni volta da user mode a kernel mode.
- (b) Un esempio di situazione in cui sono vantaggiosi i thread, rispetto ai processi tradizionali, è nel caso di un word processor, in cui un thread si può occupare della gestione della GUI (e quindi dell'interazione con l'utente), mentre un altro thread può eseguire delle operazioni in background come, ad esempio, il salvataggio o la stampa del documento. In questo modo, mentre il documento viene salvato/stampato, l'utente può proseguire ad interagire con il word processor; senza l'utilizzo di thread separati invece l'utente sarebbe costretto ad attendere il completamento dell'operazione di I/O.

Un altro esempio dei vantaggi dei thread rispetto ai processi tradizionali si ha in programmi di tipo server come i web server in cui ogni richiesta viene presa in carico da un thread separato. Nel caso in cui si gestiscano le richieste con processi separati infatti si possono avere dei seri problemi di performance e di "spreco" di risorse all'aumentare del numero di richieste. Ciò non si verifica utilizzando i thread in quanto le operazioni di creazione di nuovi thread sono molto più veloci e meno onerose rispetto a quelle di creazione di nuovi processi.

3. Si consideri un sistema con scheduling della CPU 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.

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

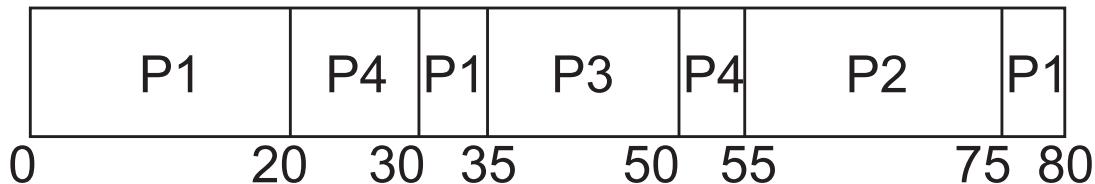
Si determini:

- (a) il diagramma di GANTT relativo all'esecuzione dei quattro processi;
- (b) il tempo di attesa medio;

Sistemi Operativi
5 dicembre 2006
Compito A

- (c) il tempo di reazione medio.

Risposta:



(a)

$$(b) \text{ tempo di attesa medio} = \frac{(30-20+75-35)+(55-5)+(35-15)+(50-30)}{4} = \frac{50+50+20+20}{4} = 35 \text{ ms};$$

$$(c) \text{ tempo di reazione medio} = \frac{0+(55-5)+(35-15)+0}{4} = \frac{50+20}{4} = 17,5 \text{ ms}.$$

4. Si consideri la seguente situazione, dove P_0, P_1, P_2 sono tre processi in esecuzione, C è la matrice delle risorse correntemente allocate, Max è la matrice del numero massimo di risorse assegnabili ad ogni processo e A è il vettore delle risorse disponibili:

	<u>C</u>			<u>Max</u>		
	<u>A</u>	<u>B</u>	<u>C</u>	<u>A</u>	<u>B</u>	<u>C</u>
P_0	0	3	1	0	5	1
P_1	2	2	2	2	4	3
P_2	1	3	0	2	4	1

	<u>Available (A)</u>		
	<u>A</u>	<u>B</u>	<u>C</u>
	x	4	1

(a) Calcolare la matrice R delle richieste.

(b) Determinare il minimo valore di x tale che il sistema si trovi in uno stato sicuro.

Risposta:

(a) Matrice delle richieste $R = Max - C$:

	<u>R</u>		
	<u>A</u>	<u>B</u>	<u>C</u>
	0	2	0
	0	2	1
	1	1	1

- (b) Il minimo valore di x tale da rendere lo stato sicuro è 0. Infatti $R_0 \leq A = (0, 4, 1)$ e quindi si può eseguire P_0 fino alla sua terminazione. A questo punto A diventa $A + C_0 = (0, 4, 1) + (0, 3, 1) = (0, 7, 2)$ e quindi $R_1 \leq A$. Dopo la terminazione di P_1 , A diventa $A + C_1 = (0, 7, 2) + (2, 2, 2) = (2, 9, 4)$. Quindi è possibile mandare in esecuzione anche P_2 (dato che $R_2 \leq A$) ed ottenere il valore finale di A , ovvero, $A + C_2 = (2, 9, 4) + (1, 3, 0) = (3, 12, 4)$. Ne segue che la sequenza sicura è $\langle P_0, P_1, P_2 \rangle$.

5. Si illustri la differenza fra partizionamento statico e dinamico della memoria. In quale dei due casi si può applicare la tecnica di *compattazione della memoria* e con quale scopo?

Risposta: Nel caso di partizionamento statico la memoria viene suddivisa in partizioni fisse (solitamente al momento del boot del sistema operativo) che possono avere dimensione uguale o diversa. Il SO tiene traccia delle partizioni allocate e di quelle libere, assegnando una di queste ultime (purché sia sufficientemente grande) ad ogni nuovo processo lanciato in esecuzione. Questa tecnica di gestione della memoria può portare a problemi di frammentazione interna.

Nel caso di partizionamento dinamico della memoria, quest'ultima viene suddivisa a tempo di esecuzione, allocando zone della dimensione esatta a contenere l'immagine del processo. Il sistema operativo tiene traccia delle porzioni allocate e di quelle libere (dette "buchi"). Quando viene lanciato in esecuzione un nuovo processo gli viene allocata una zona di memoria in un buco sufficientemente grande a contenerlo. In questo caso quindi, a lungo andare, vi possono essere dei problemi di frammentazione esterna, ovvero, la memoria complessivamente libera può essere sufficiente a contenere

Sistemi Operativi
5 dicembre 2006
Compito A

l'immagine di un nuovo processo, ma, essendo frammentata in piccoli buchi, risulta inutilizzabile. Per ovviare a questo problema è possibile utilizzare una tecnica di compattazione della memoria, ovvero, si riordinano le zone di memoria allocate in modo da agglomerare tutti i buchi in un unico buco di dimensioni sufficienti a soddisfare le richieste di un nuovo processo.

6. Si illustri la differenza fra i servizi orientati alla connessione e quelli non orientati alla connessione. Si illustri inoltre cosa si intende per *affidabilità del servizio*.

Risposta: I servizi orientati alla connessione stabiliscono una connessione, ovvero, un canale virtuale per tutta la durata della comunicazione in modo da assicurare il trasferimento dei dati in sequenza. Invece i servizi non orientati alla connessione trasferiscono singoli messaggi senza stabilire e mantenere una vera connessione. In particolare quindi il ricevente può ricevere i messaggi in ordine diverso da quello in cui sono stati spediti.

Per affidabilità del servizio si intende la garanzia che i dati trasmessi arrivino sempre a destinazione (quindi non vengano né persi né duplicati) nell'ordine in cui sono stati spediti.

7. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 40 e lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 80, 30, 31, 70, 33, rispettivamente agli istanti 0 ms, 30ms, 40 ms, 50 ms, 70 ms. Si trascuri il tempo di latenza. In quale ordine vengono servite le richieste?

Risposta: Le richieste vengono servite nell'ordine 80, 70, 33, 31, 30 come illustrato dal seguente diagramma:

