

Sistemi Operativi

9 luglio 2013

Compito

Si risponda ai seguenti quesiti, giustificando le risposte.

Gli esercizi e le domande marcate con l'asterisco (*) devono essere svolti soltanto da chi ha in piano di studi l'esame di Sistemi Operativi da 9 o 12 CFU.

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:

(a) Lo scheduling della CPU può essere attivato nelle seguenti circostanze:

1. un processo viene creato ed entra nella coda dei pronti;
2. un processo passa dallo stato di esecuzione allo stato di attesa;
3. un processo passa dallo stato di esecuzione allo stato pronto;
4. un processo passa dallo stato di attesa allo stato pronto;
5. un processo termina.

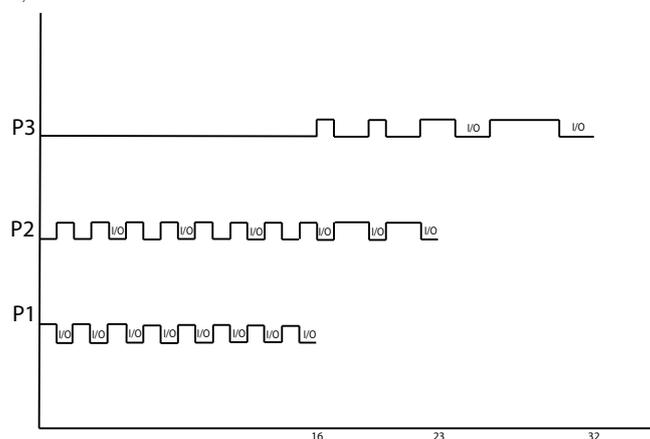
(b) Un algoritmo di scheduling si dice preemptive se può interrompere l'esecuzione di un processo a favore di un altro processo e può quindi essere attivato ogni volta che un processo passa nella coda dei pronti (coda ready), oltre che ovviamente anche in altri casi. I vantaggi di un algoritmo preemptive 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 prelazionato 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. I processi P_1, P_2, P_3 , di priorità decrescente, con richieste di CPU totali di 8, 12, 8 s rispettivamente, sono in coda ready. L'algoritmo di scheduling della CPU è basato su priorità con prelazione. Ogni processo P_i ha un comportamento ciclico: in ogni ciclo consuma c_i secondi di CPU ed esegue operazioni di I/O per o_i secondi, secondo la seguente tabella

	c_i	o_i	c_{tot}
P_1	1	1	8
P_2	2	1	12
P_3	4	2	8

L'algoritmo di scheduling viene eseguito ad intervalli di 1 secondo. Si calcolino i tempi di turnaround per i processi.

Risposta: Come si deduce dal diagramma seguente, i tempi di turnaround di P_1, P_2 e P_3 sono, rispettivamente, 16s, 23s e 32s.



3. (a) Cosa si intende per *race condition* (corsa critica)? Il verificarsi di corse critiche rappresenta un evento positivo o negativo per un sistema di calcolo?

Sistemi Operativi

9 luglio 2013

Compito

- (b) Descrivere due problemi che possono derivare dall'attesa attiva di un evento (busy wait).
- (c) I monitor ed i semafori sono costrutti applicabili anche a sistemi distribuiti? Motivare la risposta e in caso di risposta negativa descrivere un meccanismo alternativo adatto a tale scopo.

Risposta:

- (a) Si parla di race condition quando più processi accedono concorrentemente agli stessi dati e il risultato dipende dall'ordine di interleaving dei processi. Il verificarsi di corse critiche rappresenta un evento negativo per un sistema di calcolo in quanto introduce non determinismo e possibili inconsistenze nelle strutture dati condivise.
- (b) Un aspetto negativo dell'attesa attiva è il consumo "inutile" del tempo di CPU per controllare il valore della variabile. Un altro problema è dato dal fatto che l'attesa attiva di un processo a bassa priorità può ritardare/bloccare l'esecuzione di processi a maggiore priorità in attesa della stessa risorsa (inversione di priorità).
- (c) No, monitor e semafori necessitano di una forma di memoria condivisa e quindi non sono costrutti applicabili a sistemi distribuiti. Per questi ultimi si può utilizzare il modello dello scambio di messaggi in cui si utilizzano due primitive (chiamate di sistema o funzioni di libreria):
 - `send(destinazione, messaggio)`: spedisce un messaggio ad una certa destinazione; solitamente non bloccante.
 - `receive(sorgente, &messaggio)`: riceve un messaggio da una sorgente; solitamente bloccante (fino a che il messaggio non è disponibile).

Tali funzioni non necessitano di nessuna forma di condivisione di memoria e quindi sono applicabili anche per mettere in comunicazione dei sistemi distribuiti.

4. Si supponga che in una struttura vi sia un solo bagno e che sia necessario disciplinarne l'accesso tenendo presente quanto segue:
- non deve mai capitare che nel bagno siano presenti contemporaneamente uomini e donne;
 - un uomo (risp. una donna) prima di entrare deve controllare che nel bagno non siano presenti delle donne (risp. degli uomini): se la condizione non è verificata deve attendere.

Si completi la specifica del seguente monitor in modo da soddisfare i requisiti menzionati.

```
monitor Bathroom
...
integer nmen, nwomen;

procedure manEnter();
begin
    while(nwomen>0) do ...
    ...
end;

procedure manExit();
begin
    nmen := nmen-1;
    if(...) then ...
end;

procedure womanEnter();
begin
    ...
    ...
end;

procedure womanExit();
begin
    ...
```

Sistemi Operativi
9 luglio 2013
Compito

```
    ...  
    end;  
  
    nmen := 0;  
    nwomen := 0;  
end monitor;
```

Risposta:

```
monitor Bathroom  
  condition bathroomFree  
  integer nmen, nwomen;  
  
  procedure manEnter();  
  begin  
    while(nwomen>0) do wait(bathroomFree);  
    nmen := nmen + 1;  
  end;  
  
  procedure manExit();  
  begin  
    nmen := nmen-1;  
    if(nmen=0) then signal(bathroomFree);  
  end;  
  
  procedure womanEnter();  
  begin  
    while(nmen>0) do wait(bathroomFree);  
    nwomen := nwomen + 1;  
  end;  
  
  procedure womanExit();  
  begin  
    nwomen := nwomen-1;  
    if(nwomen=0) then signal(bathroomFree);  
  end;  
  
  nmen := 0;  
  nwomen := 0;  
end monitor;
```

5. Si descriva il funzionamento dell'algoritmo di rimpiazzamento delle pagine LRU. Quanti page fault si verificano sulla seguente stringa di riferimenti con l'algoritmo LRU, avendo a disposizione 4 frame fisici di memoria e 9 pagine?

6 0 5 7 8 7 0 5 1 6 4 2 1 3

Risposta: L'algoritmo LRU (Least Recently Used) "utilizza" il passato per "prevedere" il futuro, rimpiazzando la pagina che da più tempo non viene usata. Nel caso in esame si verificano 10 page fault come si vede dalla seguente simulazione:

Sistemi Operativi

9 luglio 2013

Compito

6	0	5	7	8	7	0	5	1	6	4	2	1	3
	6	0	5	7	8	7	0	5	1	6	4	2	1
		6	0	5	5	8	7	0	5	1	6	4	2
			6	0	0	5	8	7	0	5	5	5	6
							6	8	7	0	0	0	5
									8	7	7	0	
										8	8	7	
												8	
	P	P	P	P	P			P	P	P	P	P	

6. (*) Si descrivano brevemente le funzionalità offerte dal sottosistema di I/O del kernel (ovvero, dal device independent operating system software).

Risposta: Le funzionalità offerte dal sottosistema di I/O del kernel sono le seguenti:

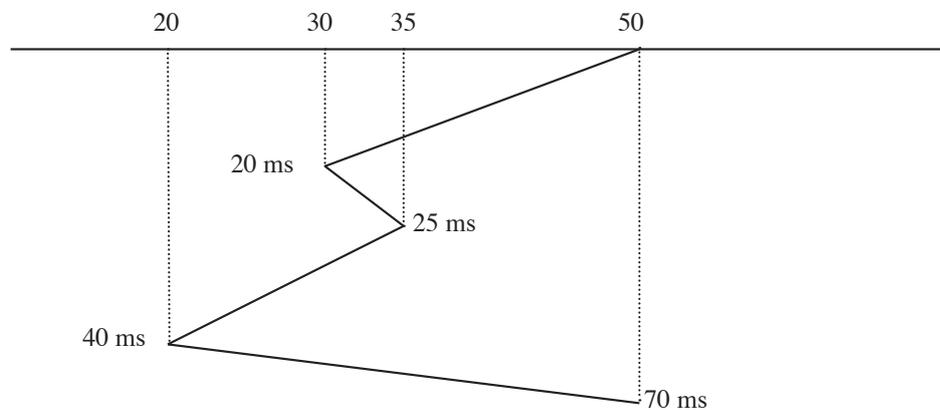
- scheduling: decide in che ordine le system call devono essere esaudite;
- buffering: serve a mantenere i dati in memoria mentre sono in transito, per gestire dispositivi a differenti velocità o con differenti dimensioni dei blocchi di trasferimento;
- caching: serve a migliorare le performance, mantenendo una copia dei dati più usati in una memoria più veloce;
- spooling: area che funge da buffer per dispositivi che non supportano I/O interleaved (es.: stampanti).
- accesso esclusivo: regola l'accesso ad alcuni dispositivi che possono essere usati solo da un processo alla volta;
- gestione degli errori: per proteggere il sistema operativo ed i processi dai problemi di funzionamento.

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 20, 30, 35, 50, rispettivamente agli istanti 0 ms, 10 ms, 20 ms, 30 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. Le richieste vengono soddisfatte nell'ordine: 30, 35, 20, 50, come risulta dal seguente diagramma:



Sistemi Operativi
9 luglio 2013
Compito

2. Il tempo di attesa medio per le quattro richieste in oggetto è
$$\frac{(20-10)+(25-20)+(40-0)+(70-30)}{4} = \frac{10+5+40+40}{4} = \frac{95}{4} = 23,75 \text{ ms.}$$

Punteggi:

- 9/12 CFU: 1a (2 pt.), 1b (2 pt.), 2 (4 pt.), 3a (2 pt.), 3b (2 pt.), 3c (2 pt.), 4 (4 pt.), 5 (3 pt.), 6 (4 pt.), 7.1 (4 pt.), 7.2 (2 pt.)
- 6 CFU: 1a (2 pt.), 1b (3 pt.), 2 (5 pt.), 3a (2 pt.), 3b (2 pt.), 3c (3 pt.), 4 (4 pt.), 5 (4 pt.), 7.1 (4 pt.), 7.2 (2 pt.)