

# Elementi di Sistemi Operativi

Bioinformatica - Tiziano Villa

18 Marzo 2008

Nome e Cognome:

Matricola:

Posta elettronica:

problema	punti massimi	i tuoi punti
problema 1	8	
problema 2	8	
problema 3	8	
problema 4	6	
totale	30	

1. Rispondere in modo preciso ma conciso alle seguenti domande.

(a) A che cosa servono le chiamate di sistema ?

Traccia di soluzione.

Le chiamate di sistema permettono ai processi degli utenti di richiedere servizi del sistema operativo.

(b) Quali sono le attività principali di un sistema operativo rispetto alla gestione dei processi ?

Traccia di soluzione.

Si veda la Sez. 1.6 del testo di Silberschatz

- i. Creazione e cancellazione di processi dell'utente e del sistema
- ii. Sospensione e riattivazione di processi
- iii. Gestione della concorrenza (sincronizzazione e comunicazione) tra processi
- iv. Gestione delle situazioni di stallo

(c) Quali sono le attività principali di un sistema operativo rispetto alla gestione della memoria principale ?

Traccia di soluzione.

Si veda la Sez. 1.7 del testo di Silberschatz Il sistema operativo e' responsabile delle seguenti attività connesse alla gestione della memoria centrale:

- i. tenere traccia di quali parti di memoria sono attualmente usate e da chi
- ii. decidere quali processi e dati devono essere caricati in memoria o trasferiti altrove
- iii. assegnare e revocare lo spazio di memoria secondo necessità

(d) Quali sono le attività principali di un sistema operativo rispetto alla gestione della memoria secondaria ?

Traccia di soluzione.

Si veda la Sez. 1.8 del testo di Silberschatz Il sistema operativo e' responsabile delle seguenti attività connesse alla gestione dei dischi:

- i. gestione dello spazio libero
- ii. assegnazione dello spazio
- iii. schedulazione del disco

2. Le domande che seguono si riferiscono a come evitare le situazioni di stallo con l'algoritmo del banchiere.

- (a) Si descrivano brevemente le strutture dati dell'algoritmo del banchiere relative a: numero delle istanze disponibili per ogni tipo di risorsa, richiesta massima di ciascun processo, numero delle istanze di ogni risorsa assegnate a ogni processo, necessita' residua di risorse di ogni processo.  
Traccia di soluzione.

Si veda per i particolari la Sez. 7.5 del testo di Silberschatz.

*Disponibili*[ $j$ ] =  $k$  significa che sono disponibili  $k$  istanze del tipo di risorsa  $R_j$ .

*Massimo*[ $i,j$ ] =  $k$  significa che il processo  $P_i$  puo' richiedere un massimo di  $k$  istanze del tipo di risorsa  $R_j$ .

*Assegnate*[ $i,j$ ] =  $k$  significa che al processo  $P_i$  sono correntemente assegnate  $k$  istanze del tipo di risorsa  $R_j$ .

*Necessita'*[ $i,j$ ] =  $k$  significa che il processo  $P_i$  puo' avere bisogno di altre  $k$  istanze del tipo di risorsa  $R_j$  per completare il suo compito.

(b) Si presenti l'algoritmo del banchiere per verificare se un sistema sia in uno stato sicuro.

Traccia di soluzione.

Si veda la Sez. 7.5.3.1 per l'algoritmo che verifica se uno stato sia sicuro (cioè che esista una successione di esecuzione dei processi, in cui ciascuno ha a disposizione tutte le risorse necessarie).

Idea: Se le risorse necessarie a un processo sono disponibili, tale processo può essere eseguito e poi rilasciare le sue risorse ad altri processi. Se è possibile iterare per tutti i processi, si costruisce una sequenza di esecuzione dei processi che garantisce a tutti le risorse necessarie.

(c) Si presenti l'algoritmo del banchiere per determinare se una richiesta di risorse possa essere soddisfatta.

Traccia di soluzione.

Si veda la Sez. 7.5.3.2 per l'algoritmo che verifica se una richiesta possa essere soddisfatta mantenendo lo stato sicuro.

Idea: Se le risorse richieste sono disponibili, si simula la loro assegnazione modificando lo stato del sistema e verificando che sia ancora sicuro con l'algoritmo precedente.



(d) Si applichi l'algoritmo del banchiere ad un sistema nel seguente stato:

	Assegnate				Massimo				Disponibili			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

- i. Si calcoli la matrice *Necessita'* (necessita' residua di risorse relativa a ogni processo).

Traccia di soluzione.

I valori di *Necessita'* per i processi  $P_0, \dots, P_4$  sono:

	Necessita'			
	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

Essa si ottiene per differenza tra i valori corrispondenti delle matrici *Massimo* e *Assegnate*.

- ii. Si verifichi se il sistema sia in uno stato sicuro.

Traccia di soluzione.

Si. Essendo il vettore *Disponibili* = (1 5 2 0), si potra' eseguire il processo  $P_0$  oppure il processo  $P_3$ . Dopo che il processo  $P_3$  avra' finito, esso rilascerà le sue risorse che permetteranno l'esecuzione di tutti gli altri processi.

- iii. Si determini se la richiesta (0 4 2 0) da parte del processo  $P_1$  possa essere soddisfatta immediatamente.

Traccia di soluzione.

Si. Soddisfacendo tale richiesta, il risultato e' che il vettore *Disponibili* diventa (1 1 0 0) (per differenza con il vettore della richiesta).

Soddisfacendo tale richiesta si raggiunge questo nuovo stato:

	Assegnate				Necessita'				Disponibili			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	0	0	1	1	0	0
P1	1	4	2	0	0	3	3	0				
P2	1	3	5	4	1	0	0	2				
P3	0	6	3	2	0	0	2	0				
P4	0	0	1	4	0	6	4	2				

Un ordine dei processi che garantisce la terminazione e'  $P_0, P_2, P_3, P_1, P_4$ .



3. Si supponga di avere 5 processi con i seguenti profili esecutivi:

**P1** 20ms CPU, I/U, 80ms CPU, I/U, 40ms CPU, I/U, 30ms CPU

**P2** 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU, I/U, 1ms CPU

**P3** 300ms CPU, I/U, 400ms CPU, I/U, 150ms CPU, I/U, 100ms CPU

**P4** 20ms CPU, I/U, 80ms CPU, I/U, 40ms CPU, I/U, 30ms CPU

**P5** 1ms CPU, I/U, 2ms CPU, I/U, 5ms CPU, I/U, 2ms CPU, I/U, 1ms CPU

Si assuma che i processi arrivino tutti al tempo 0 alla coda dei pronti nell'ordine  $P_1, P_2, P_3, P_4, P_5$ . Si assuma inoltre che ogni accesso ad I/U (ingresso/uscita) richieda 40ms.

Si determini il tempo di completamento di ogni processo se la strategia di schedulazione e' quella RR (Round Robin, cioe' assegnamento circolare), con un quanto di tempo di 50 ms. All'uopo si simuli la traccia esecutiva mostrando l'asse dei tempi annotato con la coda dei processi pronti e con gli eventi ai relativi istanti (esempio: quale processo va in esecuzione sul processore a un certo istante, quale processo e' scaricato dal processore per ingresso/uscita a un certo istante, quale processo termina definitivamente a un certo istante). Infine si calcoli il tempo medio di completamento.

Traccia di soluzione.

Gli eventi di attivazione e disattivazione dei processi sono mostrati nella Tav. 2, la cui sintesi e' mostrata nella tavola nella Tav. 1. Si noti che, quando

processo	tempo di completamento
P1	524
P2	525
P3	1426
P4	605
P5	606

Table 1: Tempo di completamento dei processi.

tutti gli altri processi hanno finito,  $P3$  ha accesso illimitato al processore.

Il tempo medio di completamento per RR e' di 737,2 ms (somma dei tempi di completamento di tutti i processi diviso per il numero di processi).

tempo	evento	coda pronti
0	tutti pronti, esegui P1	P2 P3 P4 P5
20	P1 si blocca, esegui P2	P3 P4 P5
21	P2 si blocca, esegui P3	P4 P5
60	P1 pronto	P4 P5 P1
61	P2 pronto	P4 P5 P1 P2
71	P3 fuori tempo, esegui P4	P5 P1 P2 P3
91	P4 si blocca, esegui P5	P1 P2 P3
92	P5 si blocca, esegui P1	P2 P3
131	P4 pronto	P2 P3 P4
132	P5 pronto	P2 P3 P4 P5
142	P1 fuori tempo, esegui P2	P3 P4 P5 P1
143	P2 si blocca, esegui P3	P4 P5 P1
183	P2 pronto	P4 P5 P1 P2
193	P3 fuori tempo, esegui P4	P5 P1 P2 P3
243	P4 fuori tempo, esegui P5	P1 P2 P3 P4
245	P5 si blocca, esegui P1	P2 P3 P4
275	P1 si blocca, esegui P2	P3 P4
276	P2 si blocca, esegui P3	P4
285	P5 pronto	P4 P5
315	P1 pronto	P4 P5 P1
316	P2 pronto	P4 P5 P1 P2
326	P3 fuori tempo, esegui P4	P5 P1 P2 P3
356	P4 si blocca, esegui P5	P1 P2 P3
361	P5 si blocca, esegui P1	P2 P3
396	P4 pronto	P2 P3 P4
401	P1 si blocca, esegui P2, P5 pronto	P3 P4 P5
402	P2 si blocca, esegui P3	P4 P5
441	P1 pronto	P4 P5 P1
442	P2 pronto	P4 P5 P1 P2
452	P3 fuori tempo, esegui P4	P5 P1 P2 P3
492	P4 si blocca, esegui P5	P1 P2 P3
494	P5 si blocca, esegui P1	P2 P3
524	P1 termina, esegui P2	P3
525	P2 termina, esegui P3	-
532	P4 pronto	P4
534	P5 pronto	P4 P5
575	P3 fuori tempo, esegui P4	P5 P3
605	P4 termina, esegui P5	P3
606	P5 termina, esegui P3	-
656	P3 si blocca	-
696	P3 pronto, esegui P3	-
1096	P3 si blocca	-
1136	P3 pronto, esegui P3	-
1286	P3 si blocca	-
1326	P3 pronto, esegui P3	-
1426	P3 termina	-

Table 2: Sequenza temporale degli eventi.

4. Si consideri un programma che genera la seguente successione di accessi a pagina:

A A A A B B C A C B B D D A E B C E A E D E.

Si assuma che il programma abbia a sua disposizione tre pagine fisiche ("frames").

(a) Quante assenze di pagina ("page faults") si generano usando lo schema FIFO per la sostituzione di pagina ?

Traccia di soluzione.

accessi:

A A A A B B C A C B B D D A E B C E A E D E

esecuzione algoritmo FIFO:

pagina fisica 1:

A A A A A A A A A A A D D D D B B B B E E E  
^ ^ ^ ^

pagina fisica 2:

- - - - B B B B B B B B B A A A C C C C D D  
^ ^ ^ ^

pagina fisica 3:

- - - - - C C C C C C C C E E E E A A A A  
^ ^ ^ ^

Ci sono 11 assenze di pagina (indicate dal segno ^).

(b) Quante assenze di pagina ("page faults") si generano usando lo schema ottimale MIN per la sostituzione di pagina ?

Traccia di soluzione.

accessi:

A A A A B B C A C B B D D A E B C E A E D E

esecuzione algoritmo MIN:

pagina fisica 1:

A A A A A A A A A A A A A A A A A A A D D  
^ ^

pagina fisica 2:

- - - - B B B B B B B B B B B B C C C C C C  
^ ^

pagina fisica 3:

- - - - - C C C C C D D D E E E E E E E E  
^ ^ ^

Ci sono 7 assenze di pagina (indicate dal segno ^).