

UNIVERSITA' DEGLI STUDI DI BOLOGNA - CORSO DI LAUREA IN INFORMATICA  
 CORSO DI SISTEMI OPERATIVI - ANNO ACCADEMICO 2002/2003  
 GESTIONE RISORSE - 9 giugno 2003

**Esercizio -1:** essersi iscritti correttamente per svolgere questa prova.

**Esercizio 0:** Scrivere correttamente nome, cognome e numero di matricola prima di svolgere ogni altro esercizio seguente.

**Esercizio 1:** Un docente che riceve al secondo piano del dipartimento di informatica dell'Università di Bologna è preoccupato per le lunghe code che si creano in orario di ricevimento. Suggerisce allora agli studenti di organizzarsi in modo tale da minimizzare il tempo di attesa (degli studenti), basandosi su queste informazioni: ad ogni istante  $t$ , l'orario di arrivo e il tempo richiesto degli studenti arrivati fino all'istante  $t$ . L'ordine di arrivo e il tempo previsto di servizio degli studenti è il seguente:

<i>Studente</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S5</i>	<i>S6</i>	<i>S7</i>	<i>S8</i>	<i>S9</i>
Orario arrivo	14.00	14.00	14.00	14.20	14.30	14.30	14.30	14.50	14.50
Tempo richiesto (m)	18	12	25	15	25	5	13	3	15

a) Si consideri il caso in cui il docente sia non-preemptive. Mostrate il diagramma di Gantt del docente. Qual è l'algoritmo seguito? Qual è il tempo di attesa massimo, medio e minimo?

b) Si consideri il caso in cui il docente sia preemptive. Mostrate il diagramma di Gantt del docente. Qual è l'algoritmo seguito? Qual è il tempo di attesa massimo, medio e minimo? Ci sono differenze? Se sì, spiegare perchè. Se no, spiegare perchè.

**Esercizio 2:** Si consideri un disco con 100 cilindri con una velocità di seek di 1ms per cilindro. (La velocità non è realistica per almeno un ordine di grandezza ma è utile per fare i calcoli in modo semplice). Si mostri una sequenza di richieste (minimale ma non banale) nella quale SCAN impieghi meno di un quarto del tempo necessario a FCFS per soddisfare lo stesso insieme di richieste.

**Esercizio 3:** Sia dato un sistema nel quale esistono  $N$  classi di risorse e un numero uguale  $N$  di processi. Ogni classe di risorsa contiene  $N-1$  istanze. Ad un certo istante il sistema si trova in una situazione di deadlock determinata dalla situazione seguente: ogni processo  $i$  ha una risorsa di ogni classe di indice diverso da  $i$  e richiede una risorsa della classe  $i$ .

A) si vuole risolvere il deadlock aggiungendo istanze di risorse

A1) è sufficiente aggiungere una sola istanza in una classe scelta casualmente

A2) occorre aggiungere una risorsa ad ogni classe

A3) occorre aggiungere  $N-1$  risorse in una classe scelta a caso

A4) occorre aggiungere  $N-1$  risorse in ogni classe

A5) non è possibile uscire dalla situazione di deadlock aggiungendo istanze di risorsa

Motivare la risposta

B) si vuole risolvere il deadlock facendo preemption (liberando assegnazioni di risorse)

B1) è sufficiente fare preemption di una singola assegnazione scelta a caso fra le  $(N^2-N)$  presenti

B2) occorre fare preemption di tutte le assegnazioni di un processo preso a caso

B3) occorre fare preemption di una assegnazione per ogni processo scelta casualmente

B4) occorre fare preemption di una assegnazione per ogni processo scelta secondo uno schema predefinito

B5) solo facendo preemption di tutte le assegnazioni si può eliminare il deadlock

Motivare la risposta