Sistemi Operativi

Compito del 19-07-2024

Esercizio 1 (10 punti)

In un sistema sono presenti quattro processi P_1, P_2, P_3 e P_4 che usano quattro tipi di risorsa A, B, C, D di cui sono presenti 2 unità per ciascun tipo. I processi hanno la seguente matrice di allocazione delle risorse:

	Allocazione			
	A	**B**	**C**	**D**
P_1	1	1	0	0
P_2	0	1	0	1
P_3	1	0	1	0
P_4	0	0	1	1

Inoltre ogni processo può richiedere, per ogni risorsa di un tipo che non possiede già, al massimo una risorsa.

Usando l'**algoritmo di rilevazione dello stallo** stabilire in quale caso il sistema è in stallo (giustificare il risultato).

Esercizio 2 (20 punti)

Si vuole realizzare il seguente sistema:

sono presenti N *ClientThread* dove iterativamente ognuno: inserisce in una coda un messaggio con un dato da elaborare ed attende che il messaggio sia stato elaborato da un WorkerThread. Il messaggio oltre al dato da elaborare contiene anche una priorità (0=alta, 1=media, 2=bassa) e ogni client genera messaggi con la stessa priorità. Il client quando riceve il risultato stampa il valore inviato, la priorità, il risultato ricevuto ed il tempo impiegato in ms tra inserimento in coda e ricezione risultato.

Sono presenti M WorkerThread dove ognuno iterativamente: preleva dalla coda il messaggio a priorità più alta e lo elabora in un tempo T e quindi segnala al ClientThread che ha prodotto il messaggio che l'elaborazione è stata fatta.

La coda dei messaggi è limitata e può contenere al massimo K messaggi per ogni priorità.

Il programma principale deve far partire i thread necessari e dopo 10 secondi deve indicare ai ClientThread di non generare più messaggi e terminare. Quando tutti i ClientThread hanno finito si possono terminare gli WorkerThread.

Per ogni ClientThread si vuole sapere il numero di messaggi che hanno inviato ed il tempo medio in millisecondi impiegato tra l'inserimento in coda e l'istante di ricezione del risultato. Inoltre per ogni WorkerThread si vuole sapere quanti messaggi hanno elaborato.

Per facilitare il testing il client i (i=0..N-1) genera numeri progressivi che partono da i con priorità 0 per i=0 o 1, priorità 1 per i=2 o 3 e priorità 2 altrimenti ed il WorkerThread restituisce il valore moltiplicato per 2.

Realizzare in java il sistema descritto usando i ***semafori*** per la sincronizzazione tra thread (sarà considerato errore usare polling o attesa attiva).