Sistemi Operativi

Compito 21-07-2021

Esercizio 1 (10 punti)

In un sistema sono presenti quattro processi P_1, P_2, P_3 ,P_4 che usano quattro tipi di risorsa A, B, C, D di cui sono presenti 1 unità per ciascun tipo. Il processo P_1 sta usando la risorsa A e B il processo P_2 sta usando la risorsa C e chiedendo le risorse B e D, il processo P_3 sta usando la risorsa D e il processo P_4 sta chiedendo accesso alle risorse A e C. Disegnare il grafo di allocazione delle risorse ed il rispettivo grafo d'attesa e dire se è una situazione di stallo o meno, giustificando il motivo. Per i processi P_1 e P_3 per ogni possibile singola richiesta di risorse non già possedute riportare il grafo di allocazione delle risorse ed il relativo grafo di attesa e indicare se è in stallo o meno.

Esercizio 2 (20 punti)

Si vuole realizzare il seguente sistema:

Sono presenti N *ClientThread* che producono delle richieste e M *WorkerThread* che le devono gestire e produrre un risultato. Le richieste sono inserite in una *RequestQueue* limitata a K posizioni.

Ogni *ClientThread* iterativamente genera un numero di richieste *n* tra 2 e 6 (compresi e diverso ad ogni iterazione) che verranno gestite dagli *WorkerThread* e aspetta i risultati di tutte le richieste, a questo punto integra tutti i risultati per produrre il risultato finale.

Ogni *WorkerThread* iterativamente preleva una richiesta e produce una risposta e per farlo usa attivamente la CPU per un tempo T1 (usare long *System. currentTimeMillis()*).

Per facilitare il testing il *ClientThread* invia nella richiesta i-esima il valore intero (i+1)*10 con i=0..n-1 e il *WorkerThread* restituisce il numero moltiplicato per 2. Il ClientThread somma tutti gli *n* valori prodotti dagli WorkerThread e stampa il valore di n ed il risultato.

Il programma principale deve far partire tutti i thread quindi dopo 10 secondi deve indicare ai client di non produrre più richieste, quando tutti i client hanno terminato interrompere gli worker thread e stampare per ogni worker il numero di richieste che ha servito.

Realizzare in java il sistema descritto usando i ***metodi sincronizzati*** per la sincronizzazione tra thread.