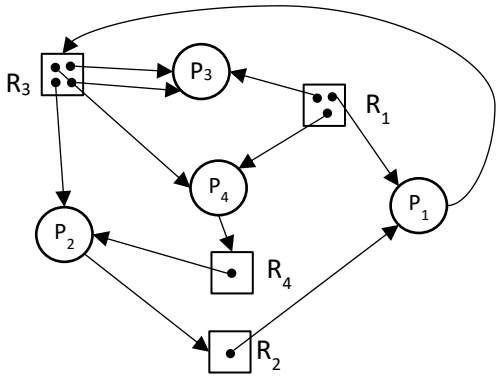
Sistemi Operativi

Compito del 13-01-2025

Esercizio 1 (10 punti)

Dato il seguente grafo di allocazione delle risorse:



Costruire le corrispondenti tabelle di allocazione delle risorse, delle richieste e delle risorse disponibili, quindi usando l'algoritmo di **rilevazione dello stallo** stabilire se il sistema è in stallo o meno. Per ogni singolo processo rivalutare la presenza di uno stallo se il processo richiede anche una risorsa non posseduta e non già in eventuale richiesta.

Esercizio 2 (20 punti)

Si vuole realizzare il seguente sistema:

Sono presenti K diversi tipi di risorse e per ogni tipo sono presenti S diverse istanze utilizzabili dagli *WorkerThread*.

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 client quando riceve il risultato stampa il valore inviato, il risultato ricevuto ed il numero di risorse usate per ogni tipo.

Sono presenti M WorkerThread dove ognuno iterativamente: preleva dalla coda un messaggio e richiede per ognuna dei K tipi di risorsa 0, 1 o 2 istanze e deve attendere se tutte le istanze richieste non sono disponibili e le acquisirà solo quando tutte saranno disponibili, quindi elabora il messaggio in un tempo [T, T + TA), rilascia le risorse acquisite ed infine segnala al ClientThread che ha inviato la richiesta che l'elaborazione è stata fatta.

La coda dei messaggi è limitata e può contenere al massimo X messaggi.

Il programma principale deve far partire i thread necessari e dopo 10 secondi deve interrompere gli WorkerThread i quali se stanno elaborando una richiesta devono rilasciare le risorse e indicare al ClientThread il fallimento della richiesta. I ClientThread che ricevono un fallimento non fanno più richieste e terminano. Quando tutti gli WorkerThread hanno finito si possono terminare i ClientThread eventualmente ancora attivi.

Quindi si stampi il numero di risorse presenti per ogni tipo e per ogni ClientThread si stampi il numero di richieste che hanno inviato e ricevuto e per ogni WorkerThread si stampi quanti messaggi hanno elaborato e quante richieste sono rimaste nella coda.

Per facilitare il testing il client genera numeri progressivi che partono da 1 ed il WorkerThread restituisce il valore moltiplicato per 2, inoltre il worker thread indica in modo casuale il numero di risorse da acquisire per ogni tipo.

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