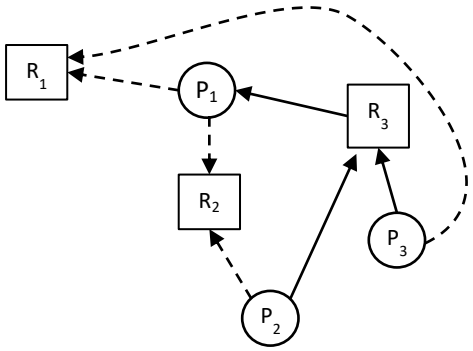
Sistemi Operativi

Seconda prova in itinere 11-06-2019

Esercizio 1 (10 punti)

Dato il seguente grafo di allocazione delle risorse:



Dove sono presenti archi di richiesta, assegnazione e rivendicazione. Determinare per ogni arco di rivendicazione se convertito in arco di assegnazione porta in uno stato sicuro o meno (giustificare il risultato). Costruire le strutture dati per l'algoritmo del banchiere relative al grafo sopra riportato e mostrare se l'assegnazione di R_2 a P_2 porta in uno stato sicuro o meno.

Esercizio 2 (20 punti)

Si vuole realizzare il seguente sistema:

- Sono presenti N thread Requester che richiedono servizi a M thread Worker tramite K thread Assigner
- Le richieste dei Requester sono inserite in una coda limitata di max R richieste dalla quale i thread Assigner prelevano le richieste e le assegnano ad uno degli M worker disponibili
- I thread Worker ricevono la richiesta, la elaborano e quindi devono restituire il risultato al Requester che ha fatto la richiesta originaria.
- Ogni thread Requester iterativamente:
    - richiede un numero progressivo ad un contatore condiviso tra tutti i Requester;
    - inserisce nella coda la sua richiesta con il valore del contatore e quindi
    - attende il risultato quindi stampa il valore inviato, quello ricevuto ed il tempo impiegato.
- Per semplificare il testing il thread Worker restituisce come risulato il valore inviato moltiplicato per 2.
- Come politica di assegnamento il thread Assigner deve assegnare il thread Worker usato il minor numero di volte.
- Il programma principale deve far partire i thread e dopo 10 secondi deve far terminare tutti i thread e stampare il numero di volte in cui ogni worker è stato assegnato.

Realizzare in java il sistema descritto usando i *metodi sincronizzati* per la sincronizzazione tra thread. Utilizzare il metodo statico *long System.nanoTime()* per ottenere il numero di nano secondi trascorsi da un istante di riferimento prefissato (es. avvio della JVM).