Soluzione di Peterson per la sezione critica

La soluzione di Peterson è una soluzione software per il problema della sezione critica. Questa soluzione potrebbe non funzionare bene sulle architetture moderne.

Sappiamo che per risolvere il problema della sezione critica dobbiamo raggiungere gli obbiettivi:

- Mutua esclusione
- Progress
- bounded waiting

Come funziona?
La soluzione è ristretta a Due processi che si alternano l'esecuzione della propria sezione critica.
Chiameremo i due processi: P_i e P_j.

Abbiamo bisogno di due variabili:

- int turn che indica di chi è il turno di entrare nella propria sezione critica.
  Se il turno è uguale a j => turn = j significa che il turno è di P_j

- boolean flag[2] Questa variabile indica che un processo è pronto ad entrare nella sua sezione critica. Quando, ad esempio la variabile flag è posta a j => flag[j] = true indica che j è pronto ad entrare nella sua sezione critica.

Piccola spiegazione preliminare

Quando un processo vuole entrare nella sua sezione critica, imposta il suo flag a true: flag[i] = true. Quando i vuole entrare nella sua sezione critica, imposterà la variabile turn sull'altro processo: turn = j.

L'intero meccanismo è come quando fai entrare una bella ragazza per prima in una porta, attendi che sia entrata e poi entri tu. codice a 2.08 wiki

Il codice

L'intero codice è posto in un loop infinito. Siamo nel caso in cui Pi vuole entrare nella sua sezione critica.

La prima cosa da fare è porre la propria variabile flag a true, in modo che poter entrare nonappena possibile.

Successivamente, si cede il posto all'altro processo con turn = j. Dopo aver settato le variabili, il processo attende finché sia flag[j] che turn == j sono VERE.

Nonappena il turno è di i, si può entrare nella sezione critica.

Una volta finita la sezione critica il flag va posto a False per dichiarare di voler uscire dalla sezione critica: flag[i] = false;

Tiriamo le somme

1. Mutua esclusione
Se il processo Pi sta eseguendo la sua sezione critica, allora nessun altro processo entrare nella sua.

2. Progress
Se nessun processo sta eseguendo la sua sezione critica, e qualche processo vorrebbe entrare nella sua, allora solo quei processi che non stanno eseguendo la loro sezione reminder possono partecipare alla decisione su chi entrerà nella S.C. per prossimo.
Questa decisione non può essere rimandata all'infinito.

Questo perché (guarda il codice) solo chi non è nella S.C. / r.s. può modificare le variabili flag e turn.

3. Bounded waiting
Esiste un limite sul numero di volte in cui un processo può entrare nella sua sezione critica, dopo che un processo ha richiesto di entrare nella S.C. e prima che questa richiesta sia accettata.