

Concurrent and Distributed Systems, 2017

Turcu Gabriel-Virgil

Calculatoare Engleza, Anul 3, Grupa H1.B

1 Enunt Problema

Se considera o multime de masini care incearca in mod repetat sa treaca un pod in lucru ce are o singura banda de traversare. Fiecare masina se deplaseaza intr-un singur sens, fie de la stanga la dreapta, fie de la dreapta la stanga. Masinile care se deplaseaza de la stanga la dreapta formeaza convoiul din stanga , iar celelalte masini care se deplaseaza de la dreapta la stanga formeaza convoiul din dreapta . Capacitatea de traversare a podului este limitata, in sensul ca la orice moment de timp pot sa existe pe pod maximum masini. In plus, masinile se pot deplasa pe pod doar intr-o singura directie, nefiind posibil ca doua masini ce merg in directii opuse sa se intalneasca pe pod. Se cunoaste ca exista masini in convoiul din stanga si masini in convoiul din dreapta.

Sa se dezvolte un program concurent care simuleaza modul in care masinile traverseaza podul in mod repetat. Fiecare masina intra pe pod, traverseaza podul, si iese de pe pod, iar acest proces se repeta pentru fiecare masina in parte de un numar prestabilit de ori. Fiecare masina se va implementa printr-un fir de executie separat. Fiecare dintre activitatile de deplasare a unei masini in afara podului, respectiv pe pod, dureaza un interval de timp nenul. Masinile nu se pot depasi pe pod, in schimb exista posibilitatea depasirii in afara podului. Se cere:

- a. Sa se schiteze o solutie a problemei si sa se explice modul de functionare, independent de o implementare particulara. Se va avea in vedere corectitudinea solutiei. Acest lucru presupune formularea explicita a conditiilor de siguranta si vivacitate si asigurarea respectarii acestora.
- b. Sa se proiecteze si sa se implementeze un program concurent care simuleaza modul in care masinile din cele doua convoaie traverseaza in mod repetat podul. Se vor implementa doua metode diferite de sincronizare si coordonare folosind: i) semafoare; ii) zavoare.
- c. Sa se propuna o metoda de testare a corectitudinii functionarii programului dezvoltat. Se va argumenta de ce aceasta metoda ne asigura ca se respecta proprietatile de siguranta si vivacitate.

2 Descrierea Solutiei

- Programul creeaza mai multe procese care simuleaza masinile de pe pod si un proces pentru centrul de control. Procesul pentru centrul de control se ocupa de comutarea semafoarelor, semafoare care ghideaza ce masini au voie si ce masini nu au voie sa mearga pe pod la anumite intervale de timp. Threadurile care reprezinta masinile se aseaza la cele doua cozi de pe pod si apoi cheama functia `wait()` si asteapta sa fie chemate de centrul de control pentru a putea trece pe pod. Dupa ce trec pe pod se aseaza din nou la coada, lucru pe care il face de fiecare data cand se termina de un numar prestabilit de ori.
- Centrul de control face urmatoarele lucruri:
 - In functie de care semafor este Rosu si care este Verde, verifica daca sunt masini care asteapta la semafor si daca partea cealalta nu este infometata.
 - Daca conditiile de mai sus sunt adevarate, trece mai departe si verifica daca mai este loc de o masina pe pod. Daca mai este, o trezeste si aceasta merge pe pod.
 - Daca podul a atins limita de masini, masina de la inceputul podului asteapta sa iasa o masina de pe pod si apoi intra si ea.
 - Dupa ce trimite o masina pe pod, asteapta 1 secunda si apoi incearca sa trimita si urmatoarea masina daca mai poate.
 - Cand nu mai poate trimite masini din cauza infometarii sau daca nu mai sunt masini, trece semaforul pe Rosu si asteapta restul masinilor de pe pod sa ajunga in partea opusa ca sa poata sa faca semaforul Verde pentru partea opusa.
 - Se repeta procesul de mai sus pentru partea opusa.
 - Se repeta tot procesul pana trec toate masinile si nu mai exista nimeni sa stea la coada in nici-o parte.

3 Input-ul Problemei

1. Primul numar ce trebuie introdus este numarul de masini care o sa fie la inceput pe partea stanga.
2. Al doilea numar ce trebuie introdus este numarul de masini care o sa fie la inceput pe partea dreapta.
3. Al treilea numar ce trebuie introdus este capacitatea maxima a podului.

4 Output-ul Problemei

In output, o sa avem mai multe mesaje care ne arata cand diferite masini au fost aprobate pentru a merge pe pod, cand intra acestea cat si cand ies de pe pod. Pe langa mesaje legate de masini, o sa avem si mesaje de la centrul de control care o sa ne spuna cand se schimba semafoarele si cand nu mai exista masini intr-o parte sau cand nu mai exista masini in ambele parti.

5 Concluziile Personale

5.1 Realizari

Un lucru pe care l-am realizat in cadrul acestui proiect este exersarea si intelegerea mecanismului de sincronizare, a Proceselor si a functiilor de Wait si Notify din Java.

5.2 Viitoare directii pentru extinderea proiectului

O viitoare directie pentru extinderea proiectului ar fi implementarea acestuia intr-un proiect grafic, proiect care sa aiba ca scop vizualizarea in timp real a unor masini pe un pod simulat. O alta varianta ar fi extinderea proiectului intr-un simulator de trafic, sa cuprinda si intersectii de drumuri.

References

- [1] Thomas H. Cormen and Charles E. Leiserson and Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms*. MIT Press, 3rd Edition, 2009.
- [2] <https://www.journaldev.com/1037/java-thread-wait-notify-and-notifyall-example>
<https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html> https://www.tutorialspoint.com/java/java_multithreading.htm