

Petculescu Mihai-Silviu

Petculescu Mihai-Silviu

- 1. Se obține același rezultat la fiecare executare? Explicați.**

[illegible]

În teorie și în mare parte din cazurile practice se va obține același rezultat, dar nu este garantat. Această execuție are la bază două variabile, prima reprezentând timpul pe procesor disponibil la rulare programului, iar a doua reprezentând intervalul de timp dintre execuția primului thread și execuția celui de-al doilea și asumția că operația pe care o au de implementat se încadrează în acel interval de timp.

Ținând cont de natura sarcinii, afișarea pe consolă a unei valori de 100 de ori, este suficient de probabil ca threadul `t1` să termine înaintea threadului `t2` în situațiile în care timpul pe procesor este suficient pentru operațiunea indicată, iar, în caz contrar, se vor produce intercalări între cele două threaduri la afișare. Dacă, să zicem, creștem numărul de afișări de la 100 la 10000, cel din urmă comportament devine cel mai probabil.

Deoarece sarcina programului a fost delegată de la threadul principal (cel de pe care se execută funcția `main()`) la unele auxiliare, există întotdeauna riscul ca acesta să se finalizeze înainte ca threadurile adiacente adăugate să-și finalizeze operațiunea, rezultând de la lipsă unor porțiuni de informație așteptată, la zone de memorie rămase alocate chiar și după finalizarea programului inițial (sarcină de care poate sau nu să aibă grijă JVM-ul, dar în acest caz ne-am raporta la o proastă implementare a codului - căci să fim raționali, nu este de datoria sistemului să curețe zone de memorie adiționale, după ce, în prealabil, a eliminat spațiul alocat programului care le-a generat în primul rând).

Linia `while(t1.isAlive() || t2.isAlive()){} se asigură că funcția main() este activă pe durata operațiunii, astfel fiind disponibilă pentru a prelua informația procesată de threaduri.`