

Multiprocesorski sistemi

Domaći zadatak 2 Pthreads – sinhronizacija

Uvod

Cilj zadatka je da studente približi načinima sinhronizacije u okruženju koje koristi **Pthreads** niti.

Podešavanje okruženja

Preuzeti <http://mups.etf.rs/vezbe/threads/threads-win32.zip>. Prema uputstvima u priloženom fajlu podesiti okruženje za razvoj i izvršavanje programa sa **Pthreads** nitima. Alternativno, koristiti **gcc/g++**.

Zadaci

Ukoliko broj niti u programu (**ne** uključujući i glavnu nit, koja treba da kreira ostale, korisničke niti) nije određen tekstom zadatka, smatrati da je određen konstantom **NUM_OF_THREADS**. Minimalna vrednost ove konstante za koju programi treba ispravno da rade data je u uglastim zagradama na kraju svakog zadatka.

1. Napisati program koji će kreirati dve korisničke niti, koje će ciklično raditi sledeći posao: prva nit treba da od korisnika zatraži unos celog broja, a druga nit da ispiše uneti broj na standardnom izlazu. Niti treba da rade sve dok korisnik ne unese negativan broj, koji takođe treba ispisati. Unos narednog broja može započeti tek po ispisu prethodno unetog broja. Koristiti uslovne promenljive.
2. Napisati program koji će kreirati **NUM_OF_PRODUCERS** niti proizvođača i **NUM_OF_CONSUMERS** niti potrošača. Proizvođači proizvode pseudoslučajne brojeve i stavljaju ih u bafer kapaciteta **CAPACITY**, odakle ih potrošači uzimaju i ispisuju na konzoli. Konstante **NUM_OF_PRODUCERS** i **NUM_OF_CONSUMERS** su jednocifreni pozitivni celi brojevi.
3. Od svih niti u programu potrebno je stvoriti prsten (ciklički graf) na sledeći način: nitima dodeliti jedinstveni celobrojni identifikator iz opsega [1,N] (identifikator glavne niti je 0), koji će omogućiti rastuće uređenje niti. Glavna nit učitava sa standardnog ulaza broj i šalje narednoj niti. Svaka od ostalih niti treba da primi broj od prethodne, potom sa glavnog ulaza pročita ceo broj, sabere pročitani sa primljenim brojem i zbir pošalje narednoj niti. Poslednja nit šalje svoj rezultat glavnoj niti koja ispisuje taj broj (sumu svih unetih brojeva) na glavnom izlazu i završava program. Za sinhronizaciju je obavezno koristiti uslovne promenljive. Nije dozvoljeno koristiti funkciju **pthread_join()**. [1]
4. Rešiti prethodni problem primenom brava (**pthread_mutex**), bez korišćenja uslovnih promenljivih.
5. Napisati program koji vrši množenje dve dinamički alocirane matrice elemenata tipa **double**. Glavna nit učitava dimenzije i elemente matrica, stvara **NUM_OF_THREADS** niti i raspoređuje posao ostalim nitima na sledeći način: svaka stvorena nit dobija zadatak da izračuna jedan element rezultujuće matrice. Kada završi računanje tekućeg elementa, stvorena nit traži od glavne niti podatke potrebne za računanje narednog elementa, sve dok ima posla. Glavna nit treba da ispiše rezultujuću matricu. Smatrati da je broj niti manji od broja elemenata rezultujuće matrice. [1]

Važno: Ukoliko u nekom zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku i da nastavi da izgrađuje svoje rešenje na temeljima uvedene pretpostavke.