

Multiprocesorski sistemi

Domaći zadatak 1

Pthreads – međusobno isključenje i sinhronizacija
(10 poena)

Uvod

Cilj prvog domaćeg zadatka je da studentima približi osnovne koncepte rada sa `Pthreads` nitima, te da ih obuči da upotrebljavaju konstrukte za međusobno isključenje i sinhronizaciju u okruženju koje koristi `Pthreads` niti.

Podešavanje okruženja

Preuzeti <http://mups.etf.rs/vezbe/pthreads/pthreads-win32.zip>. Prema uputstvima u priloženom fajlu podešiti okruženje za razvoj i izvršavanje programa sa `Pthreads` nitima. Alternativno, koristiti `gcc/g++` na računaru `rtidev4.etf.rs`.

Zadaci

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

Prilikom rešavanja zadatka voditi računa da se postigne maksimalni mogući paralelizam.

1. Sastaviti program koji određuje vrednost broja π . Glavna nit treba da stvori ostale niti, obavestiti ih o broju tačaka koje treba da obrade (unosi se sa glavnog ulaza) i prikupi podatke od njih po završenom računanju i ispiše krajnji rezultat. Glavna nit ne učestvuje u obradi tačaka. Sve niti dele samo jednu celobrojnu promenljivu i komuniciraju preko nje. Obezbediti logičku ispravnost programa međusobnim isključivanjem korišćenjem brava (`pthread_mutex_t`) i funkcije `pthread_join()`. [1]
2. Sastaviti program koji vrši određenu obradu nad dva ulazna niza celih brojeva. Potrebno je formirati novi niz od ulaznih nizova tako da bude zadovoljen uslov `c[i] = max(a[i], b[i])`. Glavna nit treba da učita veličinu i elemente niza, da svakoj od stvorenih niti prosledi početni i krajnji indeks dela niza koji data nit treba da obradi, te da i sama učestvuje u obradi. Po završenoj obradi niza, glavna nit ispisuje rezultujući niz. Raspodela posla između niti treba da bude što ravnomernija. [1]
3. Sastaviti program koji simulira komunikaciju niti unutar prstena (cikličkog grafa). Prsten treba da čine sve niti u programu, uključujući i glavnu nit. Svakoj niti je prilikom kreiranja potrebno dodeliti jedinstveni celobrojni identifikator iz opsega $[1, N]$, koji će omogućiti rastuće uređenje niti (identifikator glavne niti je 0). 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. Sastaviti program koji vrši proveru da li je uneti broj prost broj. Glavna nit treba da repetitivno učitava jedan broj, prosleđuje ga prvoj slobodnoj niti na obradu (proveru da li je broj prost ili ne) i ispisuje rezultat provere. Glavna nit upravlja stvorenim nitima i obavlja svu komunikaciju sa korisnikom. Obrada se završava kada se unese broj 0. Prilikom rešavanja zadatka koristiti koncept bazena niti. Glavna nit treba da stvori sve niti radnike pre početka obrade. [1]
5. Sastaviti program koji koji od početnog niza celih brojeva formira dva nova niza, tako da prvi sadrži samo pozitivne, a drugi samo negativne brojeve. Poredak elemenata u novoformiranim nizovima treba da bude identičan onom u početnom nizu. Glavna nit upravlja stvorenim nitima i obavlja svu komunikaciju sa korisnikom. Sve niti koje glavna nit stvara treba da ravnomočno učestvuju u obradi. Obezbediti da stvorene niti ne počnu sa radom pre nego glavna nit završi sa stvaranjem svih niti. Sinhronizaciju početka rada stvorenih niti ostvariti preko uslovnih promenljivih (`pthread_cond_t`). [1]

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

Napomena: Vrednost broja π se može statistički odrediti na više načina uz pomoć generatora pseudoslučajnih brojeva uniformne raspodele. Jedan od načina je generisanje tačaka u ravni sa obema koordinatama u realnom opsegu $[0,1]$. Vrednost broja π tada može biti određena na osnovu odnosa broja tačaka koje se nalaze u delu kruga poluprečnika 1 sa centrom u koordinatnom početku i broja tačaka koje pripadaju kvadratu stranice 1 koji obuhvata sve generisane tačke.