



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Paralelno računarstvo

Računarske vežbe

Letnji semestar 2020/2021.

Studijski program: Informacioni inženjering

OpenMP

Eksplicitna sinhronizacija

Sinhronizacija

- Implicitna barijera na kraju strukturiranog bloka `omp for` direktive
- Eksplicitna sinhronizacija:
 - direktiva `omp critical` – blok koda pod ovom direktivom se ponaša kao isključivi region
 - direktiva `omp barrier` – sve niti “se čekaju” kada dođu do ove direktive

Primer 6 – critical konstrukcija

```
1 /* ... */
2
3 int main() {
4
5     int proizvod = 1;
6
7     #pragma omp parallel for
8     for (int i = 1; i <= 5; i++)
9         #pragma omp critical
10        proizvod *= i;
11
12    cout << "Proizvod prvih pet elemenata je " << proizvod;
13    cout << ", a treba da bude 120." << endl;
14
15    return 0;
16 }
```

Primer 7 – barrier konstrukcija

```
1 /* ... */
2
3 int main() {
4
5     #pragma omp parallel
6     {
7         stringstream ss1;
8         ss1 << "Pre barijere (" << omp_get_thread_num() << ")." << endl;
9         cout << ss1.str();
10
11        #pragma omp barrier
12
13        stringstream ss2;
14        ss2 << "Posle barijere (" << omp_get_thread_num() << ")." << endl;
15        cout << ss2.str();
16
17    }
18
19    return 0;
20 }
```

Zadatak 8

Korišćenjem OpenMP direktiva, klauzula i funkcionalnosti OpenMP biblioteke implementirati paralelni C++ program za računanje vrednosti broja pi integraljenjem:

$$\int_0^1 \frac{4}{(1+x^2)}$$

Rešenje implementirati u funkciji `pi_openmp_parallel` datoteke `pi.cpp`.

Zadatak 9 - Linearna pretraga

Implementirati sekvencijalni C++ program za linearnu pretragu vektora. Postavka zadatka je data u datoteci `linearna_pretraga.cpp` gde je potrebno implementirati funkciju `linearna_pretraga_sekvencijalno`. Kao ulazne parametre funkcija prima vektor vrednosti i vrednost koja se traži u ulaznom vektoru. Vratiti indeks prve pronađene vrednosti u vektoru, odnosno -1 ukoliko tražena vrednost nije pronađena. Meriti vreme izvršavanja.

Zadatak 10 - Linearna pretraga

Implementirati paralelno OpenMP C++ rešenje za linearnu pretragu ulaznog niza. Funkcionalnost implementirati u funkciji `linearna_pretraga_openmp` datoteke `linearna_pretraga.cpp`. Meriti vreme izvršavanja. Uporediti vremena izvršavanja sekvencijalnog i paralelnog rešenja.

Zadatak 11 - Histogram

Napisati sekvencijalni C++ program za računanje histograma vrednosti iz ulaznog vektora. Vrednosti u ulaznom vektoru su nasumično generisani brojevi između 0 i 10. Postavka zadatka je data u datoteci `histogram.cpp` gde je potrebno implementirati funkciju `histogram_sekvencijalno`. Meriti vreme izvršavanja programa.

Zadatak 12 - Histogram

Implementirati paralelni OpenMP C++ program za računanje histograma za vektor ulaznih vrednosti. Implementirati rešenje u funkciji `histogram_openmp` datoteke `histogram.cpp`. Meriti vreme izvršavanja paralelnog rešenja i uporediti sa vremenom izvršavanja sekvencijalnog rešenja. Uporediti rezultate izvršavanja sekvencijalnog i paralelnog rešenja.

Zadatak 13 - Množenje matrice i vektora

Implementirati C++ rešenje za sekvencijalno množenje nekvadratne matrice i vektora. Rešenje implementirati u okviru funkcije `mnozenje_sekvencijalno` datoteke `mnozenje_matrica_vektor.cpp`.

Zadatak 14 – Množenje matrice i vektora

Implementirati OpenMP C++ rešenje za paralelno množenje nekvadratne matrice i vektora. Rešenje implementirati u okviru funkcije `mnozenje_openmp` datoteke `mnozenje_matrica_vektor.cpp`. Pre pokretanja rešenja otkomentarisati poziv funkcije iz glavnog koda i zakomentarisati poziv sekvencijalnog rešenja.

Zadatak 15 - Množenje matrica

Implementirati C++ rešenje za sekvencijalno množenje dve kvadratne matrice. Rešenje implementirati u okviru funkcije `mnozenje_sekvencijalno` datoteke `mnozenje_matrica.cpp`.

Zadatak 16 - Množenje matrica

Implementirati OpenMP C++ rešenje za paralelno množenje dve kvadratne matrice. Rešenje implementirati u okviru funkcije `mnozenje_openmp` datoteke `mnozenje_matrica.cpp`. Pre pokretanja rešenja otkomentarisati poziv funkcije iz glavnog koda i zakomentarisati poziv sekvencijalnog rešenja.

Dodatni materijali

- OpenMP
- Tim Mattson – Introduction to OpenMP
- Tim Mattson –
A “Hands-on” Introduction to OpenMP