
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS)
Nastavnik: dr Milo Tomašević, red. prof.; dr Marko Mišić, vanr. Prof.
Asistent: Matija Dodović, dipl. ing.
Ispitni rok: Prvi kolokvijum (Novembar 2023.)
Datum: 01.11.2023.

Kandidat:* _____

Broj Indeksa:* _____

*Kolokvijum traje 105 minuta, prvih sat vremena nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /15	<i>Zadatak 5</i>	_____ /15
<i>Zadatak 2</i>	_____ /15	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 3</i>	_____ /15	<i>Zadatak 7</i>	_____ /15
<i>Zadatak 4</i>	_____ /15		

Ukupno na kolokvijumu: _____ /100

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**. * popunjava student.

1. [15] Uporedno diskutovati nekadašnje i sadašnje principe projektovanja procesora.

2. [15] Opisati i kvantifikovano prikazati trendove tehnologije u izradi procesora..

3. [15] Definisati pojam programskog modela. Kako se oni realizuju? Koje su najvažnije projektne odluke koje paralelni programski model nameće?

4. [15] Objasniti programski model *paralelnih podataka*. Nacrtati i objasniti tipičnu paralelnu arhitekturu koja podržava ovaj model.

5. [15] Korišćenjem OpenMP tehnologije, paralelizovati deo koda u prilogu koji primenjuje Jakobijev metod za rešavanje sistema linearnih jednačina. Obratiti pažnju na efikasnost i korektnost paralelizacije. Smatrati da su sve promenljive ispravno deklarirane.

```
while (iterations < max_iterations) {

    for (int i = 0; i < size; ++i) {

        double sum = 0.0;

        for (int j = 0; j < size; ++j) {

            if (i != j)

                sum += A[i][j] * x[j];

        }

        new_x[i] = (b[i] - sum) / A[i][i];
    }

    double error = 0.0;

    for (int i = 0; i < size; ++i) {

        error += fabs(new_x[i] - x[i]);

        x[i] = new_x[i];
    }

    iterations++;

    if (error < tolerance)

        break;

}
```

6. [10] Objasniti upotrebu *schedule* odredbe kod OpenMP *for* direktive. Diskutovati mogućnosti upotrebe i razliku *schedule(static,1)* i *schedule(dynamic,1)* primenjenih na kod u prilogu.

```
#pragma omp parallel for
  for (int i = 0; i < n; i++)
    do();
```

7. [15] Neka se posmatra jedna aplikacija koja vrši klasterizaciju podataka. Aplikacija najpre učitava podatke za klasterovanje u obliku matrice dimenzije $m \times n$, gde m predstavlja pojedinačne odbirke, a n predstavlja njihove odlike nad kojima se vrši klasterovanje. Obrada se nezavisno može vršiti po redovima matrice. Nakon merenja performansi sekvencijalne implementacije posmatrane aplikacije pri uobičajenoj upotrebi, dobijeni su sledeći rezultati: aplikacija 70% vremena provodi obavljajući ulazno-izlazne operacije, a 30% vremena provodi u obradi podataka. Tipično vreme obrade jednog reda korišćenjem jednog jezgra je 10ms.

a) [7] Ukoliko se aplikacija paralelizuje za izvršavanje na SMP sistemu sa 8 jezgara na 2GHz sa 32GB memorije, navesti formulu za Amdalov zakon i odrediti maksimalno moguće ubrzanje koje se može postići za zadatak aplikaciju sa datom konfiguracijom.

b) [4] Navesti tipične način za dekompoziciju domena problema. Predložiti načine dekompozicije matrice podataka za problem naveden pod a).

c) [4] Diskutovati moguće načine za ublažavanje problema dugog vremena izvršavanja ulazno-izlaznih operacija u rešavanju problema navedenog pod a).