
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13E114MUPS)

Nastavnik: dr Milo Tomašević, red. prof.

Asistent: doc. dr Marko Mišić

Ispitni rok: Jul 2020.

Datum: 10.07.2020.

Kandidat:* _____

Broj Indeksa:* _____

*Ispit traje 180 minuta, prvih sat vremena nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /5	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 2</i>	_____ /10	<i>Zadatak 7</i>	_____ /10
<i>Zadatak 3</i>	_____ /10	<i>Zadatak 8</i>	_____ /10
<i>Zadatak 4</i>	_____ /10	<i>Zadatak 9</i>	_____ /15
<i>Zadatak 5</i>	_____ /10	<i>Zadatak 10</i>	_____ /10

Ukupno na ispitu: _____ /100

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je u okviru (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. **[5]** Ukratko objasniti osnovne tehnološke trendove i njihov uticaj na performanse procesora.

2. **[10]** Opisati osnovne karakteristike programskog modela paralelnih podataka. Nacrtati i opisati tipičnu arhitekturu.

3. **[10]** Diskutovati prednosti i nedostatke zajedničke (deljene) keš memorije. Gde se ona obično primenjuje?

4. **[10]** Koji problemi se sagledavaju kod MSI protokola? Kako bi se oni mogli rešiti?

5. [10] Objasniti strukturu i način održavanja kataloga kod *Dir_i SW* protokola.

6. [10] Precizno objasniti transakcije između nivoa u dvonivoskoj keš hijerarhiji kada se primenjuje princip inkluzije.

7. [10] Neka se posmatra kod u prilogu koji vrši određenu obradu nad svakim elementom niza *mArr* i traži minimum tako obrađenih elemenata u tom nizu. Na koje sve načine se može sprovesti raspoređivanje i obrada podataka tako da se paralelizacija obavi efikasno korišćenjem MPI tehnologija? Skicirati odgovarajući deo koda za master i procese-radnike.

```
mArr[0] = f(mArr[0]);
min = mArr[0];
for (j = 0; j < n; j++) {
    mArr[j] = f(mArr[j]);
    if (min > mArr[j]) {
        min = mArr[j];
    }
}
```

8. [10] Na koji način se može rešiti problem atomičnog inkrementiranja brojača kod određivanja histograma na GPU? Kako takvo rešenje utiče na propusni opseg i na koji način tehnika privatizacije izlaza utiče na povećanje propusnog opsega?

9. [15] Korišćenjem OpenMP biblioteke, paralelizovati kod u prilogu koji vrši detekciju ivica korišćenjem *Sobel* filtera. Obratiti pažnju na efikasnost i korektnost paralelizacije.

```
void sobelFiltering(unsigned char **image1, unsigned char **image2,
    int weight[3][3], int x_size, int y_size, double min, max) {
    int x, y, i, j;

    double pixel_value;

    for (y = 1; y < y_size - 1; y++) {

        for (x = 1; x < x_size - 1; x++) {

            pixel_value = 0.0;

            for (j = -1; j <= 1; j++) {

                for (i = -1; i <= 1; i++) {

                    pixel_value += weight[j + 1][i + 1] * image1[y + j][x + i];

                }

            }

            pixel_value = MAX_BRIGHTNESS * (pixel_value - min) / (max - min);

            image2[y][x] = (unsigned char)pixel_value;
        }
    }
}
```

10. [10] Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *Firefly* protokol za održavanje koherencije keš memorije. Svaka keš memorija ima po 2 ulaza, koji su veličine jedne reči. Preslikavanje je **direktno**. Početne vrednosti podataka su 0. Svaki upis uvećava vrednost izmenjenog podatka za 1. Na početku su sve keš memorije prazne. Data je sledeća sekvenca pristupa memoriji:

1. P1, W, A0	3. P0, W, A0	5. P2, R, A0	7. P1, W, A1
2. P2, R, A0	4. P0, R, A2	6. P1, W, A2	8. P3, W, A1

Napisati stanja koherencije u svim procesorima i stanje memorije posle svake promene i skicirati opisani sistem u trenutku 8. [8 poena]

Da li procesori pristupaju memoriji i kada? Za svaki pristup navesti razlog. [2 poena]

Trenutak 1											
P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 2											
P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 3											
P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 4											
P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 5

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 6

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 7

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:

Trenutak 8

P0			P1			P2			P3		

Memorija	
A0	
A1	
A2	
A3	

Pristupi memoriji:
