
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS, 13E114MUPS)

Nastavnik: dr Milo Tomašević, red. prof.

Asistent: doc. dr Marko Mišić

Ispitni rok: Avgust 2021.

Datum: 27.08.2021.

Kandidat:* _____

Broj Indeksa:* _____

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

<i>Zadatak 1</i>	_____ /10	<i>Zadatak 6</i>	_____ /10
<i>Zadatak 2</i>	_____ /5	<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 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. **[10]** Ukratko opisati osnovne klase računara i njihove karakteristike.

2. **[5]** Objasniti i uporediti implicitne i eksplicitne paralelne programske modele.

3. [10] Objasniti sve akcije i promene stanja koje iniciraju upisi od strane procesora u protokolu *Dragon*. Nacrtati i objasniti odgovorajuće delove dijagama stanja i prelaza.

4. [10] Objasniti motivaciju za adaptivne protokole, kao i osnovne principe njihovog rada.

5. [10] Objasniti strukturu kataloga i funkcionisanje *Dir_i DP* protokola sa dinamičkim pointerima.

6. [10] Objasniti strukturu, karakteristike i način funkcionisanja krosbar interkonekcijske mreže.

7. [10] Objasniti da li se može koristiti *reduction* odredba u okviru *task* direktive OpenMP biblioteke? Diskutovati alternative. Korigovati kod u prilogu koji vrši određenu vrstu redukcije tako da predstavlja korektno rešenje.

```
int index, n = width * height;
float mean = 0.0;
#pragma omp parallel
{
    #pragma omp single
    {
        #pragma omp task
        for (index = 0; index < n; index++)
            mean += process(inputImage[index]);

        std = sqrtf(var);
        mean /= (float)n;
    }
    #pragma omp for
    for (index = 0; index < n; index++)
        outputImage[index] = (inputImage[index] - mean)/std;
}
```

8. [10] Zbog čega je bitan pristup memoriji u transakcijama na grafičkom procesoru i kako on utiče na performanse? Kada se dešava pristup u više transakcija? Odgovor ilustrovati slikom.

9. [15] Korišćenjem MPI biblioteke, napisati deo koda procesa-gospodara za paralelizaciju koda koji je dat u prilogu i vrši sinusnu transformaciju signala. Proces-gospodar treba da raspodeli posao, učesvuje u obradi i prikupi rezultate.

```
double *sine_transform_data (double d[], int n) {
    double angle, pi = 3.141592653589793, *s;
    int i, j;
    s = new double[n];
    for (i = 0; i < n; i++) {
        s[i] = 0.0;
        for (j = 0; j < n; j++) {
            angle = pi * (double) ((i + 1) * (j + 1)) / (double) (n + 1);
            s[i] = s[i] + sin(angle) * d[j];
        }
        s[i] = s[i] * sqrt (2.0 / (double) (n + 1));
    }
    return s;
}
```

10. [10] Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *MOESI* 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, R, A0	3. P2, W, A0	5. P1, W, A0	7. P1, R, A0
2. P0, R, A0	4. P2, R, A2	6. P0, W, A0	8. P0, 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												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 2												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 3												Memorija	
P0			P1			P2			P3			A0	

Pristupi memoriji:

Trenutak 4												Memorija	
P0			P1			P2			P3			A0	

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:
