
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13E114MUPS)
Nastavnik: dr Milo Tomašević, red. prof.; dr Marko Mišić, vanr. prof.
Asistent: Matija Dodović, dipl. ing.
Ispitni rok: Jun 2024.
Datum: 13.06.2024.

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 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. [5] Objasniti fenomen *power wall*. Diskutovati njegove konsekvence na principe projektovanja procesora.

2. [10] Definisati pojam programskog modela. Kako se oni realizuju? Nabrojati tri osnovna paralelna programska modela i identifikovati na kom nivou sistema se ostvaruje komunikacija i sinhronizacija.

3. **[10]** Objasniti motivaciju za uvođenje stanja E u protokolu *MESI*. Precizno objasniti akcije i prelaze u dijagramu stanja koje izazivaju upisi.

4. **[10]** Opisati 4C model promašaja u keš memorijama. Objasniti kako bi se broj promašaja svake pojedine vrste mogao smanjiti.

5. [10] Opisati šta se dešava pri izbacivanju nekog bloka iz keš memorije kod *directory* protokola. Posebno diskutovati alternative pri izbacivanju bloka čiji je sadržaj isti kao u memoriji. Identifikovati situacije i protokole za koje su pojedine alternative pogodne.

6. [10] Objasniti princip organizacije višestepene sprežne mreže tipa *Omega*, kao i način rutiranja poruka. Objasniti da li je mreža blokirajuća. Ako jeste, nacrtati u mreži 8x8 primer rutiranja dve poruke koje izazivaju blokiranje i označiti mesto gde dolazi do blokiranja.

7. [10] Objasniti na koji način se vrši raspoređivanje niti *streaming* multiprocesorima u okviru grafičkog procesora prilikom njihovog izvršavanja. Na koji način je omogućena skalabilnost izvršavanja u zavisnosti od karakteristika dostupnog hardvera? Odgovor ilustrovati slikom.
8. [10] Definisati pojam jednostrane komunikacije kod MPI biblioteke i koncept prozora. Kada ima smisla koristiti ovaj vid komunikacije? Napisati skelet odgovarajućih kodova za inicijalizaciju, postavljanje i dohvaćanje putem jednostrane komunikacije poruka iz zadatog lokalnog bafera *msg* dužine jednog 32-bitnog celog broja. Prosesi koji učestvuju u komunikaciji imaju rangove definisane konstantama *ORIGIN* i *TARGET*.

9. [15] Koristeći OpenMP tehnologiju, dopuniti i paralelizovati funkciju koja je data u prilogu. Funkcija računa histogram zadate slike, a delimično je već prilagođena za efikasnu paralelizaciju putem OpenMP. Obratiti pažnju na efikasnost i korektnost paralelizacije.

```
void histogram(int *data, int n, int *bins, int c, int k){
    int num_bins = c / k;

    int number_of_threads = _____;

    int local_bins[number_of_threads][1024];

    memset(bins, 0, sizeof(bins));

    memset(local_bins, 0, sizeof(local_bins));

    {
        int id = _____;
        int i, j;

        for(i = 0 ; i < n ; ++i)

            local_bins[id][data[i] / k]++;

        for(i = 0 ; i < num_bins; ++i)
        {

            for(j = 0 ; j < number_of_threads; ++j)

                bins[i] += local_bins[j][i];

        }
    }
}
```

10. [10] Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *Dragon* 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,A2	3. P0,R,A2	5. P2,W,A1	7. P0,W,A0
2. P0,R,A1	4. P1,W,A2	6. P0,W,A1	8. P1,R,A2

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:
