
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi
(13E114MUPS, 13S114MUPS, 13M114MUPS)

Nastavnik: dr Marko Mišić, vanr. prof.

Asistenti: Matija Dodović, mast. ing.

Ispitni rok: Septembar 2026.

Datum: 15.09.2026.

Kandidat:* _____

Broj Indeksa:* _____

*Ispit traje 120 minuta, prvih sat vremena nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /10	<i>Zadatak 5</i>	_____ /10
<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	<i>Zadatak 8</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.

- 2/8

3. [15] Objasniti glavne zaključke analize načina deljenja podataka u tipičnim paralelnim aplikacijama, kao i konsekvence ove analize na širini i visinu kataloga u *directory* protokolima. Navesti dva primera protokola gde se ovi zaključci koriste.
4. [15] Objasniti topologiju interkonekcionih mreža tipa stabla i njene osnovne osobine. Kolike su vrednosti karakterističnih parametara? Koji je osnovni problem i kako se rešava?

5. [10] U prilogu je dato CUDA jezgro koje vrši konverziju slike iz RGB kolor modela u nijanse sive.

```
__global__ void rgb2gray(float *grayImage, float *rgbImage, int channels,
                        int width, int height) {
    int x = threadIdx.x + blockIdx.x * blockDim.x;
    int y = threadIdx.y + blockIdx.y * blockDim.y;
    if (x < width && y < height) {
        int grayOffset = y * width + x;
        int rgbOffset = grayOffset * channels;
        float r = rgbImage[rgbOffset];
        float g = rgbImage[rgbOffset + 1];
        float b = rgbImage[rgbOffset + 2];
        grayImage[grayOffset] = 0.21f * r + 0.71f * g + 0.07f * b;
    }
}
```

a) [5] Kakav je aritmetički intenzitet ovog jezgra u odnosu na pristupe memoriji? Komentarisati.

b) [5] Da li su pristupi memoriji sjedinjeni (*coalesced*) i na koji način bi se to moglo postići?

6. **[10]** Za šta služi struktura `Status` prilikom prijema poruke korišćenjem `MPI_Recv` poziva? Na koji način se može iskoristiti ova struktura, ukoliko je poruka primljena korišćenjem džoker simbola `MPI_ANY_TAG` kao labele poruke? Navesti primer.

7. [15] Korišćenjem OpenMP tehnologije, paralelizovati funkciju koja je data u prilogu. Funkcija vrši jednu iteraciju rešavača jedne vrste parcijalnih diferencijalnih jednačina. Obratiti pažnju na efikasnost paralelizacije.

```
void iterate ( int nx, int ny, double dx, double dy, double f[NX][NY],
               int itold, int itnew, double u[NX][NY], double unew[NX][NY])
{
    int i, it, j;

    for ( it = itold + 1; it <= itnew; it++ ) {

        for ( j = 0; j < ny; j++ ) {

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

                u[i][j] = unew[i][j];
            }
        }

        for ( j = 0; j < ny; j++ ) {

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

                if ( i == 0 || j == 0 || i == nx - 1 || j == ny - 1 ) {

                    unew[i][j] = f[i][j];
                }

                else {

                    unew[i][j] = 0.25 * (u[i-1][j] + u[i][j+1] + u[i][j-1] +
                                         u[i+1][j] + f[i][j] * dx * dy );
                }
            }
        }
    }
    return;
}
```

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

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	
												A1	
												A2	
												A3	

Pristupi memoriji:

Trenutak 2												Memorija	
P0			P1			P2			P3			A0	
												A1	
												A2	
												A3	

Pristupi memoriji:

Trenutak 3												Memorija	
P0			P1			P2			P3			A0	
												A1	
												A2	
												A3	

Pristupi memoriji:

Trenutak 4												Memorija	
P0			P1			P2			P3			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:
