
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: Jun 2020.

Datum: 18.06.2021.

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]** Objasniti šta je ILP i navesti neki primer. Objasniti i obrazložiti potencijale i ograničenja u iskorišćenju ILP-a.

2. **[10]** Objasniti šta je labavo skaliranje? Koji zakon pretpostavlja ovakav vid skaliranja? Objasniti i izvesti izraz za ubrzanje po ovom zakonu i komentarisati ga.

3. [10] Kakvo poboljšanje donosi MESIF protokol u odnosu na MESI? Kakva je semantika novouvedenog stanja F? Objasniti aktivnosti protokola koje se odnose na ovo stanje i skicirati samo taj deo dijagrama stanja.

4. [10] Uporediti strategije invalidacije i ažuriranja, njihove prednosti i mane. Definisati osnovni kriterijum koji utiče na njihove performanse.

5. [10] Komparativno objasniti strukturu i način održavanja kataloga kod *Dir_i B* i *Dir_i NB* protokola sa grubim vektorom. Koji od njih je pogodniji za 'read-only' i 'read-mostly' podatake i zašto? Kojem od njih odgovara objavljivanje zamene ažurne kopije u keš memoriji?
6. [10] Nacrtati interkonekcionu mrežu *Butterfly* 8x8. Objasniti način povezivanja mreže i rutiranja poruka. Da li je mreže blokirajuća? Ako jeste, na slici ilustrovati dva prenosa koji bi izazvali blokiranje.

7. [10] Neka se posmatra isečak koda u prilogu paralelizovan putem OpenMP. Objasniti kakav problem navedeni kod rešava, kao i ulogu i neophodnost upotrebe *flush* direktive.

```
#define NUM_THREADS 256
int synch[NUM_THREADS]; float work[NUM_THREADS]; float result[NUM_THREADS];
int main() {
    int iam, neighbor;
    #pragma omp parallel private(iam,neighbor) shared(work,synch)
    {
        iam = omp_get_thread_num();
        synch[iam] = 0;
        #pragma omp barrier
        work[iam] = function_1(iam);
        #pragma omp flush(work,synch)
        synch[iam] = 1;
        #pragma omp flush(synch)
        neighbor = ...;
        while (synch[neighbor] == 0) {
            #pragma omp flush(synch)
        }
        #pragma omp flush(work,synch)
        result[iam] = function2(work[neighbor], work[iam]);
    }
}
```

8. [10] Neka se posmatra sledeći scenario u okviru jednog MPI programa. Procesi najpre rade nad nekim nizom n elemenata, svaki nad svojim delom. Svaki lokalno formira rezultujući niz od $p \ll n$ elemenata koji je potrebno da pošalje svim ostalim procesima, kako bi nastavili obradu. Koji mehanizmi (komunikacione rutine) na nivou MPI biblioteke su dostupne za ovakav scenario obrade? Diskutovati koje alternative mogu da se koriste?

9. [15] Korišćenjem CUDA tehnologije, paralelizovati kod u prilogu koji vrši detekciju ivica korišćenjem *Sobel* filtera. Obratiti pažnju na efikasnost i korektnost paralelizacije. Koristiti 2D organizaciju jezgra. Navesti poziv jezgra.

```
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 *MSI* 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. P0, R, A0	3. P1, W, A0	5. P1, W, A2	7. P0, R, A3
2. P1, R, A0	4. P0, R, A1	6. P0, R, A2	8. P0, W, A3

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:
