
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: Jul 2019.

Datum: 09.07.2019.

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>	_____ /15	<i>Zadatak 8</i>	_____ /10
<i>Zadatak 4</i>	_____ /5	<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 zašto se ne prave sve složeniji superskalarni procesori.

2. [10] Definisati paralelni programski model. Objasniti šta su implicitni, a šta eksplicitni paralelni programski modeli.

3. [15] Precizno objasniti sve akcije i prelaze između stanja koji se dešavaju prilikom upisa u MESI protokolu. Nacrtati odgovarajući deo dijagrama stanja.

4. [5] Objasniti kako migracija procesa utiče na performanse u sistemu sa održavanjem koherencije.

5. [10] Objasniti motivaciju za osnovnu tehniku smanjivanja širine u *directory* protokolima. Ukratko objasniti funkcionisanje ove tehnike, kao i njenu prostornu složenost.

6. [10] Objasniti i nacrtati interkonekcionu mrežu sa topologijom stabla. Dati vrednosti karakterističnih parametara. Koji je osnovni nedostatak i kako se prevazilazi?

7. [10] Da li postoje razlike u implementaciji rutina za komunikaciju tačno dva procesa i kolektivnih operacija kod MPI biblioteke? Objasniti na primeru slanja jednog podatka korišćenjem MPI_Bcast poziva i slanja tog istog podatka sa nekoliko MPI_Send operacija.

8. [10] Za priloženi kod koji vrši iteriranje kroz piksele zadate slike, napisati korišćenjem CUDA tehnologije kostur 2D jezgra za izvršavanje na GPU. Navesti poziv takvog jezgra.

```
for (uint16 pixelY = 0; pixelY < rectHeight; pixelY++) {
    for (uint16 pixelX = 0; pixelX < rectWidth; pixelX++) {
        pixelOffset = (pixelX * columnBytes + pixelY * rowBytes);
        doEffect(pixelOffset);
    }
}
```

9. [15] Korišćenjem OpenMP tehnologije, paralelizovati kod u prilogu. Obratiti pažnju na efikasnost i korektnost paralelizacije. Smatrati da su sve promenljive ispravno definisane i inicijalizovane, a memorija alocirana.

```
double epot, vir;
void
forces(int npart, double x[], double f[], double side, double rcoeff){
    int i, j;
    double sideh, rcoeffs, xi, yi, zi, fxi, fyi, fzi, xx, yy, zz;
    double rd, rrd, rrd2, rrd3, rrd4, rrd6, rrd7, r148;
    double forcex, forcey, forcez;
    vir = 0.0; epot = 0.0;
    sideh = 0.5*side; rcoeffs = rcoeff*rcoeff;

    for (i=0; i<npart*3; i+=3) {

        xi = x[i]; yi = x[i+1]; zi = x[i+2];
        fxi = 0.0; fyi = 0.0; fzi = 0.0;

        for (j=i+3; j<npart*3; j+=3) {

            xx = xi-x[j]; yy = yi-x[j+1]; zz = zi-x[j+2];
            rd = xx*xx+yy*yy+zz*zz;

            if (rd<=rcoeffs) {

                rrd = 1.0/rd;
                rrd2 = rrd*rrd;
                rrd3 = rrd2*rrd;
                rrd4 = rrd2*rrd2;
                rrd6 = rrd2*rrd4;
                rrd7 = rrd6*rrd;
                epot += (rrd6-rrd3);
                r148 = rrd7-0.5*rrd4;
                vir -= rd*r148;
                forcex = xx*r148;
                fxi += forcex;
                f[j] -= forcex;
                forcey = yy*r148;
                fyi += forcey;
                f[j+1] -= forcey;
                forcez = zz*r148;
                fzi += forcez;
                f[j+2] -= forcez;
            }
        }

        f[i] += fxi;
        f[i+1] += fyi;
        f[i+2] += fzi;
    }
}
```

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