
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Multiprocesorski sistemi (13S114MUPS)

Nastavnik: dr Marko Mišić, vanr. prof.

Asistent: Matija Dodović, mast. ing.

Ispitni rok: Prvi kolokvijum (Oktobar 2024.)

Datum: 30.10.2024.

Kandidat:* _____

Broj Indeksa:* _____

*Kolokvijum traje 90 minuta. Prvih 60 minuta nije dozvoljeno napuštanje kolokvijuma.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____/15	<i>Zadatak 5</i>	_____/15
<i>Zadatak 2</i>	_____/15	<i>Zadatak 6</i>	_____/15
<i>Zadatak 3</i>	_____/15	<i>Zadatak 7</i>	_____/10
<i>Zadatak 4</i>	_____/15		

Ukupno na kolokvijumu: _____/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.

- 2/6

3. **[15]** Definirati pojam labavog skaliranja aplikacija, a zatim izvesti model ubrzanja kod *Gustafson-Barsis*-ovog zakona, Kada je ovaj zakon primenljiv?
4. **[15]** Objasniti pojam kolektivne komunikacije kod programskog modela *prenosa poruka*. Objasniti tipične primere komunikacije koje podržava ovaj model i odgovor ilustrovati slikom.

5. [15] Korišćenjem OpenMP tehnologije, paralelizovati kod koji je dat u prilogu. Kod vrši određene proračune u simulaciji prostiranja talasa na žici. Obratiti pažnju na efikasnost paralelizacije. Smatrati da su sve promenljive već definisane i inicijalizovane.

```
for (i = 1; i <= nsteps; i++) {  
  
    for (j = 1; j <= tpoints; j++) {  
  
        if ((j == 1) || (j == tpoints))  
  
            newval[j] = 0.0;  
  
        else  
  
            newval[j] = (2.0 * values[j]) - oldval[j] +  
                        (sqtau * (values[j-1] - (2.0 * values[j]) +  
                        values[j+1]));  
  
    }  
  
    for (j = 1; j <= tpoints; j++) {  
  
        oldval[j] = values[j];  
  
        values[j] = newval[j];  
  
    }  
  
}
```

6. **[15]** Sinhronizacija kod OpenMP tehnologije

a. **[5]** Nabrojati i kratko opisati konstrukte za sinhronizaciju kod OpenMP tehnologije.

b. **[10]** Na primeru koda sa slike, označiti deo koda nad kojim je potrebno izvršiti sinhronizaciju i predložiti odgovarajuće konstrukte.

```
#pragma omp parallel for private(k, f_part_k, len, len_3, mg, fact)
for (k = part+1; k < n; k++) {
    /* Compute force on part due to k */
    len_3 = len*len*len;
    f_part_k[X] = (curr[part].s[X] - curr[k].s[X]) * mg / len_3 * fact;
    f_part_k[Y] = (curr[part].s[Y] - curr[k].s[Y]) * mg / len_3 * fact;
    len = sqrt(f_part_k[X]*f_part_k[X] + f_part_k[Y]*f_part_k[Y]);
    /* Add force in to total forces */
    forces[part][X] += f_part_k[X];
    forces[part][Y] += f_part_k[Y];
    forces[k][X] -= f_part_k[X];
    forces[k][Y] -= f_part_k[Y];
}
```

7. **[10]** Koje sve direktive za podelu posla postoje kod OpenMP tehnologije. Diskutovati njihove razlike i ilustrovati na primeru upotrebe pojedinačnih direktiva.