

# Multiprocesorski sistemi

## Pismeni ispit, 02.09.2009.

Literatura nije dozvoljena.

Ispit traje 180 minuta.

1. Objasniti razloge koji uslovljavaju trend ka *multicore* procesorima. [5 poena]
2. Šta je procesorska lokalnost? Uporedno diskutovati prednosti i nedostatke strategija invalidacije i ažuriranja? [5 poena]
3. Navesti implikacije na pisanje paralelnih programa. [5 poena]
4. Za protokol MESI objasniti stanja, transakcije na magistrali i akcije protokola. Nacrtati i objasniti dijagram stanja i prelaza. [15 poena]
5. Definisati svojstvo inkluzije kod keš memorija? Objasniti kako ona utiče na održavanje koherencije. Objasniti osnovne akcije između L1 i L2. [10 poena]
6. Objasniti organizaciju informacija o koherenciji i akcije *DirCV* protokola sa "grubim" vektorom? [10 poena]
7. Koristeći POSIX niti napisati na jeziku C ili C++ program koji radi upis blokova celobrojnih podataka na disk. Postoji nekoliko niti koje proizvode celobrojne podatke pseudoslučajne vrednosti od 0 do 20. Postoji jedna nit koja čeka da se sakupi 32 KiB podataka i onda ih upiše na disk jednim pozivom neke postojeće U/I funkcije. Kada naprave ukupno 32 KiB podataka, niti-proizvođači prestaju sa radom dok nit-potrošač ne obavi upis. Postupak treba ponavljati beskonačno. Glavni program treba da stvori i pokrene opisane niti. Sinhronizaciju između proizvođača i potrošača obaviti preko uslovnih promenljivih. [15 poena]
8. Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *Firefly* za održavanje koherencije keš memorije. Svaka keš memorija ima po 2 ulaza, koji su veličine jedne reči. Preslikavanje je direktno. Na početku su sve keš memorije prazne. Svaki upis povećava vrednost izmenjenog podatka za 1. Početne vrednosti podataka u memoriji su 0. Data je sledeća sekvenca pristupa memoriji:

1. P0, R, A0	3. P1, W, A0	5. P2, R, A0	7. P2, R, A2
2. P2, R, A0	4. P3, W, A0	6. P2, R, A1	8. P2, R, A3

- 8.1. Koliko puta koji od procesora pristupa memoriji? Za svaki pristup navesti razlog. [3 poena]
- 8.2. Koliki je Hit Rate za svaki od procesora (brojati i čitanje i upis, prikazati zbirno)? [3 poena]
- 8.3. Napisati stanja koherencije u svim procesorima i stanje memorije nakon svakog koraka (dovoljno je stanja ispisivati samo posle promene). [9 poena]
9. Sastaviti MPI program na jeziku C ili C++ koji određuje najveći i najmanji element niza brojeva. Pretragu obavljaju dve grupe procesa. Prva grupa sadrži procese sa parnim rangom u osnovnom komunikatoru i određuje najmanji element. Druga grupa sadrži procese sa neparnim rangom u osnovnom komunikatoru i određuje najveći element. Svaki proces treba da obradi isti broj elemenata kao i svi ostali. Proces sa rangom 0 u osnovnom komunikatoru treba da obavi potrebnu komunikaciju sa korisnikom. Ako dimenzija niza nije odgovarajuća broju procesa, prekinuti program. [20 poena]

### Napomena:

U zadacima pretpostaviti da funkcije koje obavljaju potrebne ulazne i izlazne radnje već postoje, tako da za njih samo treba navesti prototipove i pozvati ih na odgovarajućim mestima u programskom kodu. Pretpostaviti da korisnik unosi sintaksno ispravne podatke.

Ukoliko u bilo kom pitanju ili zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi bila lakše prepoznata prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke.