

Multiprocesorski sistemi

Pismeni ispit, 15.09.2008.

Literatura nije dozvoljena.
Ispit traje 210 minuta.

1. Definisati sekvencijalnu konzistenciju kao i ograničenja koja ona nameće. [5 poena]
2. Uporediti programske modele zajedničke memorije i prenosa poruka. [10 poena]
3. Zašto je neophodno *Shared-modified* stanje u protokolu *Dragon*? Da li bi se ono moglo izbeći i kako? [5 poena]
4. Za *full-map directory* protokol objasniti organizaciju informacije o koherenciji, transakcije u mreži i akcije protokola. Nacrtati i objasniti dijagram stanja i prelaza. [15 poena]
5. Definisati svojstvo inkluzije kod keš memorija. Koji su problemi kod održavanja inkluzije i koji su potrebni uslovi za njeno održavanje. [5 poena]
6. Ukratko opisati logiku i realizaciju adaptivnog RWB protoka. Koji je njegov nedostatak? [10 poena]
7. Koristeći POSIX niti napisati program na jeziku C ili C++ koji simulira bolje uređivanje novca raspoređenog po džakovima. Pretpostaviti da je džak predstavljen zbirkom brojeva, od kojih svaki predstavlja nominalnu vrednost odgovarajuće novčanice iz džaka. Nit zadužena za brojanje treba da obradi dodeljeni džak, tako što će svaku novčanicu postavljati u odgovarajući svežanj. Čim svežanj bude kompletiran, nit zadužena za slaganje treba da preuzme svežanj i postavi ga u sef. Glavni program treba da sačeka da sve niti završe svoj posao, te da pre kraja svog izvršavanja ispiše koliko je svežnjeva uređeno, koliko je njihov iznos i koji je ukupni iznos u svim džakovima. [20 poena]
8. Dat je multiprocesorski sistem sa 4 identična procesora, koji koristi *Dragon* za održavanje koherencije keš memorije. Svaka keš memorija ima po 2 ulaza, koji su veličine jedne reči. Preslikavanje je direktno. Data je sledeća sekvenca pristupa memoriji:

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

Na početku su sve keš memorije prazne. Inicijalna vrednost svih memorijskih lokacija je 0. Instrukcija upisa povećava vrednost podatka za 1.

- 8.1. Skicirati opisani sistem posle trenutka 8. [3 poena]
 - 8.2. Koliko puta koji od procesora pristupa memoriji? Za svaki pristup navesti razlog. [2 poena]
 - 8.3. Ponoviti prethodnu tačku za *Firefly* protokol. Objasniti razliku. [2 poena]
 - 8.4. Napisati stanja koherencije u svim procesorima (samo posle promene) [8 poena].
9. Sastaviti MPI program na jeziku C ili C++ koji pronalazi najjeftinije trešnje, dinje i breskve na pijaci. Pijaca je predstavljena kao niz tezgi, od kojih svaka ima informaciju o ceni svake od tri namirnice. Organizovati procese u tri grupe sa što sličnijim brojem procesa. Svaka grupa treba da traži jednu od namirnica. Proces sa rangom 0 u svakoj grupi treba da redni broj tezge sa najpovoljnijom cenom pošalje procesu sa rangom 0 u glavnom komunikatoru, koji treba da učita podatke o tezgama i ispiše dobijene redne brojeve. [15 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.