

Multiprocesorski sistemi (SI4MPS)

Treći kolokvijum, 22.1.2010.

Literatura nije dozvoljena.
Kolokvijum traje 90 minuta.

1. Objasniti vrste promašaja u keš memorijama multiprocesorskih sistema i načine njihovog smanjivanja. [15 poena]
2. Razmotriti tipične načine deljenja u aplikacijama sa aspekta broja invalidacija. Koje su implikacije u odnosu na organizaciju kataloga? [10 poena]
3. Šta se postiže primenom "cache-based" protokola? Objasniti organizaciju kataloga i izračunati njegovu veličinu ako je m – veličina memorije, c – veličina keš memorije po procesoru, b - veličina bloka i n – broj procesora. Koje su prednosti i nedostaci ovakve organizacije. [20 poena]
4. Diskutovati prednosti i nedostatke organizacije čvora kao manjeg multiprocesorskog sistema u većim sistemima? [15 poena]
5. Multiprocesorski sistem ima 32 procesora i ukupno 1 GiB deljene memorije. Jedna linija keša ima 128 B. Proračunati koliko je dodatne memorije potrebno za čuvanje kataloga. Proračun napraviti za Dir_iB šemu sa $i=4$. [15 poena]
6. Napisati program na programskom jeziku C ili C++ koji traži minimalni i maksimalni element matrice. Obradu paralelizovati i realizovati korišćenjem MPI, uz podelu procesa po grupama. Minimum treba da traži posebna grupa procesa, koja ima onoliko procesa koliko ima vrsta matrice. Svaki proces treba da obradi jednu vrstu matrice. Isto važi i za maksimum. Proces sa rangom 0 (gospodar) ne treba da učestvuje u obradi, treba da po završenoj obradi prikupi rezultate obrade od procesa sa rangom 0 u svakoj od namenskih grupa i ispiše rezultate. Ako korisnik zada broj procesa manji od potrebnog, prekinuti program. Ako korisnik zada broj procesa veći od potrebnog, prekobrojne procese rasporediti u posebnu grupu, u kojoj će svaki proces samo ispisati poruku da je bez posla. [25 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.