

# Multiprocesorski sistemi

## Laboratorijska vežba 4 (MPI – virtuelne topologije)

### Uvod

Cilj vežbe je da studente obučiti da koriste MPI pozive za rad sa virtuelnim topologijama.

### Podešavanje okruženja

Preuzeti [http://mups.etf.rs/vezbe/mpl/code/mpl\\_virtual\\_topologies.zip](http://mups.etf.rs/vezbe/mpl/code/mpl_virtual_topologies.zip). U arhivi je VS 2005 rešenje sa dva MPI projekta. Prema uputstvima u priloženom `readme.txt` fajlu podesiti okruženje za razvoj i kontrolisano izvršavanje (engl. debugging) MPI programa na lokalnom računaru.

### Zadaci

Svaki od programa napisati tako da može biti izvršen sa bilo kojim od broja procesa određenih vrednostima navedenih u uglastim zagradama na kraju postavke zadatka. Broj N označava maksimalan mogući broj procesa u trenutno dostupnom MPI klasteru. Za programe koji će biti izvršavani samo na lokalnom računaru, pretpostaviti da važi  $N=10$ . Svi programi treba da vrše proveru da li je broj procesa odgovarajući postavci zadatka. U slučaju da broj procesa nije odgovarajući, prekinuti izvršavanje korišćenjem MPI poziva `Abort`.

1. Sastaviti program koji učitava niz realnih brojeva (broj elemenata je određen brojem raspoloživih procesa), uvezuje procese u `Cart` komunikator i ispisuje rangove levog i desnog procesa-suseda u `Cart` komunikatoru kao i vrednosti susednih elementa niza [2..N].
2. Isto kao i prethodni program, samo što je virtuelna topologija prsten umesto vektora [2..N].
3. Sastaviti program koji učitava dimenzije i elemente matrice, uvezuje raspoložive procese u `Cart` komunikator i ispisuje rangove gornjeg, donjeg, levog i desnog suseda u `Cart` komunikatoru, kao i vrednosti susednih elemenata matrice [2..N].
4. Isto kao i prethodni program, samo što je virtuelna topologija cilindar umesto matrice [2..N].
5. Isto kao i prethodni program, samo što je virtuelna topologija torus umesto cilindra [2..N].
6. Proširiti prethodni program tako da radi i ako broj raspoloživih procesa bude veći od proizvoda dimenzija matrica.
7. Sastaviti program koji stvara ciklični graf (prsten) od prisutnih procesa [2..N]. Svaki proces treba da svoj rang doda na broj primljen od „levog“ suseda i rezultat prosledi „desnom“ susedu. Proces sa rangom 0 treba da prvi pošalje svoj rang i ispiše primljeni rezultat nakon što zbir rangova zatvori krug.
8. Sastaviti program koji obavlja paralelno sabiranje elemenata niza čiji je broj elemenata celobrojni umnožak broja procesa-radnika u MPI svetu [3,8,15]. Proces-radnici treba da podele niz na jednake delove, sabere elemente svog dela i taj zbir pošalju „roditeljskom“ procesu. Proces može imati dve uloge – i procesa-radnika, koji vrši sabiranje elemenata niza, i „roditeljskog“ procesa, koji vrši sabiranje samo međurezultata. Graf procesa organizovati analogno priloženoj slici, uz uvažavanje činjenice da proces ne može biti sam sebi sused u virtuelnoj topologiji. Proces 0 treba da ispiše konačni rezultat.

