

Introduction to distributed and parallel processing

Laboratory 3

Write using MPI the following programs:

1. (4 pnts) Implement one to all broadcast communication for hypercube network using only send and receive functions. Use algorithm presented during lecture.
Function header:

```
void one2AllBroadcastHypercube(int &msg)
or
void one2AllBroadcastHypercube(int &msg, int srcProc)
```

where: in the first case we assume, that processor 0 broadcast its message. In the another case `srcProc` start the broadcast. Only starting process has an important value as `msg`, after end of the function call, all processors have in `msg` copy of sender's message.

2. (2 pnts) Implement one to all broadcast communication using group communication function from MPI.
3. (4 pnts) Implement all to all personalized communication for ring network using only send and receive functions. Use algorithm presented during lecture. Create function:

```
void all2AllPersonalizeRing(int *sendMsg, int *recvMsg).
```

Remember that in the array `sendMsg` on position `i` there is a message for `i`-th process. After returning from calling this function in the array `recvMsg` on position `j` there have to be message from process `j`.

For testing prepare array for every process in such a way:

```
processor "0" has sendMsg={0,1,2,3,..., p-1}
processor "1" has sendMsg={1000,1001,1002,...,1000+(p-1)}
processor "2" has sendMsg={2000,2001,2002,...,2000+(p-1)}
...
```

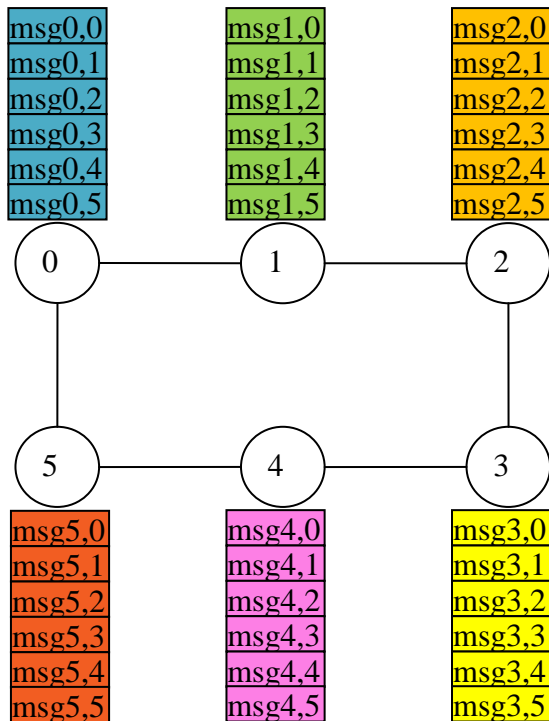
after come backing from the function:

```
processor "0" has recvMsg={0,1000,2000,3000,..., (p-1)*1000}
processor "1" has recvMsg={1,1001,2001,..., (p-1)*1000+1}
processor "2" has recvMsg={2,1002,2002,..., (p-1)*1000+2}
```

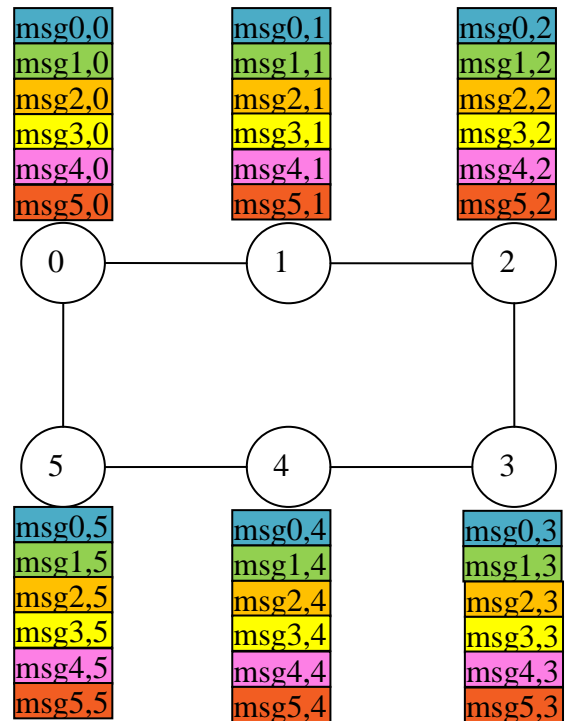
Use `printf(...)` command **before** and **after** every communication function similar like for laboratory 2. Use this also in the main program before and after calling the function.

Communication **all-to-all personalized (in a ring)**

0) Initial State



1) Final State



msg1,3 ← ---- A message from processor 1 to processor 3

State after shift phase

