

Recitation 9

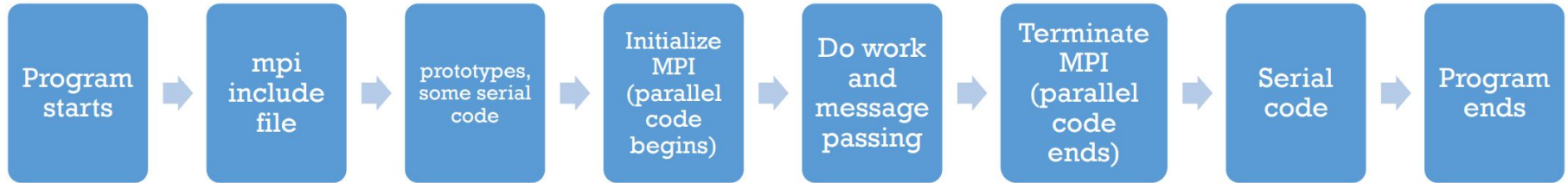
Zeinab Khalifa
October 29th, 2020

Carnegie Mellon University Qatar

What is MPI?

- It's a library of routines that can be used to create parallel programs
- Applications can be written in C, C++ and calls to MPI can be added where required.

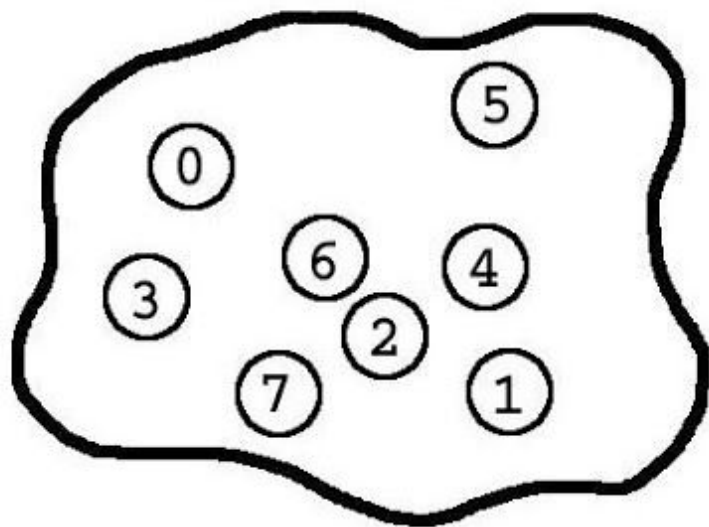
MPI Program Skeleton



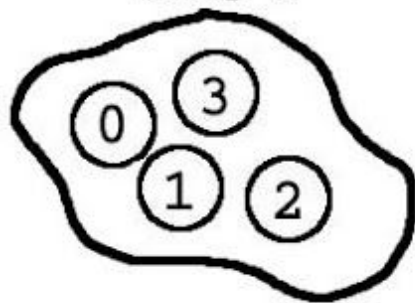
Communicators and Groups

- Communicator is a set of processes that may communicate with each other and may consist of processes from a single group or multiple groups.
- When an MPI application starts, the group of all processes is initially given a predefined name called **MPI_COMM_WORLD**

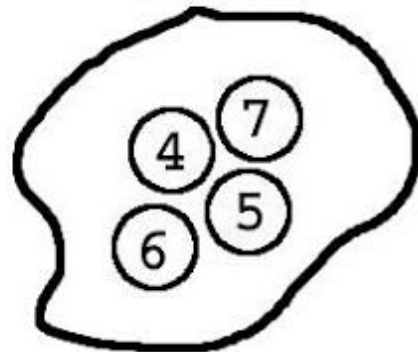
MPI_COMM_WORLD



Group A



Group B



Ranks

- Within a communicator, each process has its own and unique ID or rank
- These IDs are commonly used conditionally to control program execution
- Ranks start from 0

MPI Routines

- MPI_INIT – initialize the MPI library (must be the first routine called)
- MPI_COMM_SIZE - determines the number of processes in the group associated with the comm communicator
- MPI_COMM_RANK – get the rank of the calling process in the communicator
- MPI_SEND – send a message to another process
- MPI_RECV – send a message to another process
- MPI_FINALIZE – clean up all MPI state (must be the last MPI function called by a process)
- MPI_Wtime – determines elapsed wall clock time in seconds on the calling processor. We'll use this to measure the runtime of an MPI program

Let's write our first MPI program...

MPI Send

- `MPI_Send(void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)`
- This is a basic blocking send operation. It returns only after the application has sent the data to the recipient(s)
- MPI Datatype is very similar to a C datatype: `MPI_INT`, `MPI_CHAR`
- The count refers to how many datatype elements should be communicated
- tag is a user-defined “type” for the message
- dest is the rank of the target process in the communicator specified by comm.

MPI Recv

- `MPI_Recv(void *buf, int count, MPI_Datatype datatype, int src, int tag, MPI_Comm comm, MPI_Status *status)`
- This receives a message and blocks until the requested data is available in the application buffer
- source is rank in communicator comm
- status contains further information on who sent the message, how much data was actually received,..

More MPI programs...