

### REVIEW - Introduction to MPI

#### True or false:

- MPI is a message-passing library specification not a language or compiler specification.
- In the MPI model processes communicate by shared memory.
- MPI is useful for an implementation of MIMD/SPMD parallelism.
- A group and context together form a communicator.
- A default communicator MPI\_COMM\_WORLD contains in its group all initial processes and its context is default.
- A process is identified by its rank in the group associated with a communicator.
- Maximal rank is equal to size.

### REVIEW - Introduction to MPI (cont.)

- List MPI functions to control starting and terminating procedures on all processors.
- List MPI functions for determining the number of participating processes (size) and the identifier of the current process (rank).
- Suppose that in a processes with rank 1 started the execution of MPI\_SEND (buf, 5, MPI\_INT, 4, 7, MPI\_COMM\_WORLD).  
(a) Which process has to start matching MPI\_RECV to finish this communication?  
(b) Write the adequate MPI\_RECV.  
(c) What will be received?

### REVIEW - Introduction to MPI (cont.)

- Implement a one-to-all MPI broadcast operation whereby a single named process (root) sends the same data to all other processes. Which process(es) has(have) to call this operation?
- Name the following definitions of the communication semantics:
  - a) may return before the operation completes, and before the user is allowed to re-use resources (such as buffers) specified in the call,
  - b) return from call indicates that resources can safely be re-used,
  - c) may require execution of an MPI procedure on another process, or communication with another process,
  - d) all processes in a group need to invoke the procedure.

### REVIEW - Introduction to MPI (cont.)

- When a process makes a call to MPI\_RECV, it will wait patiently until a matching send is posted. If the matching send is never posted, the receive will wait forever. a) Name this situation! b) How can you solve the problem?
- Give a functional equivalent program segment using non-blocking send to implement MPI's blocking send MPI\_SEND.

### REVIEW - Introduction to MPI (cont.)

- Name the following definitions of the communication semantics:
  - a) If a sender posts two messages to the same receiver, and a receive operation matches both messages, the message first posted will be chosen,
  - b) No matter how long a send has been pending, it can always be overtaken by a message sent from another process.
  - c) does MPI implementation by itself guarantee fairness?

### REVIEW - Introduction to MPI (cont.)

- Which types of parallel program composition are supported by a) MPI\_COMM\_DUP (comm, newcomm) and b) MPI\_COMM\_SPLIT (comm, color, key, newcomm)? c) Are these functions examples of collective operations?
- Suppose four processes a, b, c, d, with corresponding oldrank in comm: 0 1 2 3. Let color=oldrank%2 and key: 7 1 0 3. Write newgroups in newcomm sorted by newranks after the execution of MPI\_COMM\_SPLIT (comm, color, key, newcomm)
- Suppose an NxN array of integers. Construct a derived datatype newtype specifying the main diagonal of the array.