**Intro to Distributed Memory**

**Parallel Computing with MPI**

NCSA Research Facilitation Services Workshop

**MPI Cheatsheet - Fortran**

## MPI Program Structure

| use mpi . . . integer :: mpierr . . call MPI\_INIT(mpierr) . . . call MPI\_FINALIZE(mpierr) |
| --- |

## Information about PEs

| call MPI\_COMM\_SIZE(MPI\_COMM\_WORLD, nranks, mpierr) |
| --- |

* Returns the total number of PEs in the World communicator into the integer variable **nranks**

| call MPI\_COMM\_RANK(MPI\_COMM\_WORLD, myrank, mpierr) |
| --- |

* Returns the ID of each PE into the variable **myrank**

## MPI Send message

| call MPI\_SEND(BUF, COUNT, DTYPE, DEST, TAG, COMM, IERR) |
| --- |

* **BUF**: initial memory address of the message to be sent
* **COUNT**: number of elements in the message
* **DTYPE**: data type of each element
* **DEST**: ID of the PE that will receive the message
* **TAG**: can be used to classify the message
* **COMM:** communicator over which the message is passed
* **IERR**: return error code

## MPI Receive message

| call MPI\_RECV(BUF, COUNT, DTYPE, SOURCE, TAG, COMM, STATUS, IERR) |
| --- |

* **BUF**: initial memory address of where the message received is stored
* **COUNT**: number of elements in the message
* **DTYPE**: data type of each element
* **SOURCE**: ID of the PE that sent the message
* **TAG**: can be used to classify the message
* **COMM:** communicator over which the message is passed
* **STATUS**: stores information about the received message
* **IERR**: return error code

## MPI Broadcast

| call MPI\_BCAST(SEND\_BUF, SEND\_COUNT, DTYPE, ROOT\_ID, COMM, IERR) |
| --- |

* **SEND\_BUF**: initial memory address of the message that is being sent to all PEs
* **SEND\_COUNT:** number of elements sent
* **DTYPE**: data type of each element
* **ROOT\_ID**: ID of the PE from which message is being broadcast
* **COMM**: communicator over which broadcasting will take place
* **IERR**: return error code

## MPI Reduce

| call MPI\_REDUCE(SEND\_BUF, RECV\_BUF, COUNT, DTYPE, OP, ROOT\_ID, COMM, IERR) |
| --- |

* **SEND\_BUF**: initial memory address of the message that is going to be reduced
* **RECV\_BUF**: memory address to where the reduced result is going to be stored
* **COUNT**: number of elements in each **SEND\_BUF**
* **DTYPE**: data type of each element
* **OP**: MPI reduce operation (e.g. MPI\_SUM)
* **ROOT\_ID**: ID of the PE that is going to receive the reduction
* **COMM**: Communication over which reduction will happen
* **IERR**: return error code