MPI Reference

IDRIS

24/08/2022

1 Environment

- `MPI_Init` : Initialize the MPI environment
- `MPI_Comm_rank` : Rank of process
- `MPI_Comm_size` : Number of process
- `MPI_Finalize` : Disable the MPI environment
- `MPI_Abort` : Abort a MPI program
- `MPI_Wtime` : Elapsed time

2 Point to point communications

- `MPI_Send` : Send a message
- `MPI_Isend` : Non blocking send
- `MPI_Recv` : Receive a message
- `MPI_Irecv` : Non blocking receive
- `MPI_Sendrecv` et `MPI_Sendrecv_replace` : Send and receive
- `MPI_Wait` : Wait for the end of a non blocking communication
- `MPI_Wait_all` : Wait for the end of all non blocking communication

3 Collective communications

- `MPI_Bcast` : Global distribution
- `MPI_Scatter` : Selective distribution
- `MPI_Gather` et `MPI_Allgather` : Collection
- `MPI_Alltoall` : Collection and distribution
- `MPI_Reduce` et `MPI_Allreduce` : Reduction
- `MPI_Barrier` : Global synchronization

4 Derived datatype

- `MPI_Type_contiguous` : Contiguous datatypes
- `MPI_Type_vector` et `MPI_Type_create_hvector` : Datatypes with constant stride
- `MPI_Type_indexed` : Datatypes with variable stride
- `MPI_Type_create_subarray` : Subarray datatypes
- `MPI_Type_create_struct` : Heterogenous datatypes
- `MPI_Type_commit` : Commit datatype
- `MPI_Type_get_extent` : Get the extent
- `MPI_Type_create_resized` : Change the extent
- `MPI_Type_size` : Size of a datatype
- `MPI_Type_free` : Free the datatype

5 Communicator

- `MPI_Comm_split` : Partitioning of a communicator
- `MPI_Dims_create` : Process distribution
- `MPI_Cart_create` : Creation of a cartesian topology
- `MPI_Cart_rank` : Rank of a process in a cartesian topology
- `MPI_Cart_coords` : Coordinates of a process in a cartesian topology
- `MPI_Cart_shift` : Rank of neighbours in a cartesian topology
- `MPI_Comm_free` : Free a communicator

6 MPI-IO

- `MPI_File_open` : Open a file
- `MPI_File_set_view` : Change the view
- `MPI_File_close` : Close a file

6.1 Explicit offsets

- `MPI_File_read_at` : Read
- `MPI_File_read_at_all` : Collective read
- `MPI_File_write_at` : Write

6.2 Individual file pointers

- `MPI_File_read` : Read
- `MPI_File_read_all` : Collective read
- `MPI_File_write` : Write
- `MPI_File_write_all` : Collective write
- `MPI_File_seek` : Positioning the file pointer

6.3 Shared file pointers

- `MPI_File_read_shared` : Read
- `MPI_File_read_ordered` : Collective read
- `MPI_File_seek_shared` : Positioning the file pointer

7 Constant

- `MPI_COMM_WORLD`, `MPI_SUCCESS`
- `MPI_STATUS_IGNORE`, `MPI_PROC_NULL`
- `MPI_INTEGER`, `MPI_REAL`, `MPI_DOUBLE_PRECISION`
- `MPI_ORDER_FORTRAN`, `MPI_ORDER_C`
- `MPI_MODE_CREATE`,`MPI_MODE_RONLY`,`MPI_MODE_WRONLY`

8 Interfaces

Interfaces and arguments are available with the `man` command, for example, `man mpi_init`