

# **MPI Quick Reference in C**

#include <mpi.h>

#### **Environmental Management:**

int MPI Init(int \*argc, char \*\*argv[])

int MPI Finalize(void)

int MPI Initialized(int \*flag)

int MPI Finalized(int \*flag)

int MPI Comm size(MPI Comm comm, int \*size)

int MPI Comm rank(MPI Comm comm, int \*rank)

int MPI Abort(MPI Comm comm, int errorcode)

double MPI Wtime(void)

double MPI Wtick(void)

#### **Blocking Point-to-Point-Communication:**

- int MPI\_Send (void\* buf, int count,
   MPI\_Datatype datatype, int dest, int tag,
   MPI Comm comm)
- Related: MPI Bsend, MPI Ssend, MPI Rsend
- int MPI\_Probe (int source, int tag, MPI\_Comm
   comm, MPI Status \*status)

Related: MPI\_Get\_elements

- int MPI\_Sendrecv (void \*sendbuf, int
   sendcount, MPI\_Datatype sendtype, int
   dest, int sendtag, void \*recvbuf, int
   recvcount, MPI\_Datatype recvtype, int
   source, int recvtag, MPI\_Comm comm,
   MPI\_Status \*status)
- int MPI\_Sendrecv\_replace (void \*buf, int
   count, MPI\_Datatype datatype, int dest,
   int sendtag, int source, int recvtag,
   MPI\_Comm comm, MPI\_Status \*status)
- int MPI\_Buffer\_attach (void \*buffer, int

size)

#### **Non-Blocking Point-to-Point-Communication:**

- Related: MPI Ibsend, MPI Issend, MPI Irsend
- int MPI\_Iprobe (int source, int tag, MPI\_Comm
   comm, int \*flag, MPI\_Status \*status)
- int MPI\_Test (MPI\_Request \*request, int
   \*flag, MPI\_Status \*status)
- int MPI\_Waitall (int count, MPI\_Request
   request\_array[], MPI\_Status
   status array[])

Related: MPI\_Testall

int MPI\_Waitany (int count, MPI\_Request
 request\_array[], int \*index, MPI\_Status
 \*status)

Related: MPI\_Testany

int MPI\_Waitsome (int incount, MPI\_Request
 request\_array[], int \*outcount, int
 index\_array[], MPI\_Status status\_array[])

Related: MPI\_Testsome,

int MPI\_Request\_free (MPI\_Request \*request)

Related: MPI\_Cancel

int MPI\_Test\_cancelled (MPI\_Status \*status,
 int \*flag)

#### **Collective Communication:**

- int MPI\_Barrier (MPI\_Comm comm)
- int MPI\_Gather (void \*sendbuf, int sendcount,
   MPI\_Datatype sendtype, void \*recvbuf, int
   recvcount, MPI\_Datatype recvtype, int
   root, MPI Comm comm)

- int MPI\_Gatherv (void \*sendbuf, int
   sendcount, MPI\_Datatype sendtype, void
   \*recvbuf, int recvcount\_array[], int
   displ\_array[], MPI\_Datatype recvtype, int
   root, MPI Comm comm)
- int MPI\_Scatter (void \*sendbuf, int
   sendcount, MPI\_Datatype sendtype, void
   \*recvbuf, int recvcount, MPI\_Datatype
   recvtype, int root, MPI Comm comm)
- int MPI\_Scatterv (void \*sendbuf, int
   sendcount\_array[], int displ\_array[]
   MPI\_Datatype sendtype, void \*recvbuf, int
   recvcount, MPI\_Datatype recvtype, int
   root, MPI Comm comm)
- int MPI\_Allgather (void \*sendbuf, int
   sendcount, MPI\_Datatype sendtype, void
   \*recvbuf, int recvcount, MPI\_Datatype
   recvtype, MPI\_Comm comm)

Related: MPI\_Alltoall

int MPI\_Allgatherv (void \*sendbuf, int
 sendcount, MPI\_Datatype sendtype, void
 \*recvbuf, int recvcount\_array[], int
 displ\_array[], MPI\_Datatype recvtype,
 MPI\_Comm comm)

**Related:** MPI Alltoallv

- int MPI\_Reduce (void \*sendbuf, void \*recvbuf,
   int count, MPI\_Datatype datatype, MPI\_Op
   op, int root, MPI\_Comm comm)
- int MPI\_Allreduce (void \*sendbuf, void
   \*recvbuf, int count, MPI\_Datatype
   datatype, MPI Op op, MPI Comm comm)

Related: MPI Scan, MPI Exscan

- int MPI\_Reduce\_scatter (void \*sendbuf, void
   \*recvbuf, int recvcount\_array[],
   MPI\_Datatype datatype, MPI\_Op op,
   MPI\_Comm comm)
- int MPI\_Op\_create (MPI\_User\_function \*func,
   int commute, MPI Op \*op)
- int MPI Op free (MPI Op \*op)

#### **Derived Datatypes:**

- int MPI\_Type\_commit (MPI\_Datatype \*datatype)
- int MPI\_Type\_free (MPI\_Datatype \*datatype)

- \*newtype)
- int MPI\_Type\_vector (int count, int
   blocklength, int stride, MPI\_Datatype
   oldtype, MPI Datatype \*newtype)
- int MPI\_Type\_indexed (int count, int
   blocklength\_array[], int displ\_array[],
   MPI\_Datatype oldtype, MPI\_Datatype
   \*newtype)
- int MPI\_Type\_create\_struct (int count, int
   blocklength\_array[], MPI\_Aint
   displ\_array[], MPI\_Datatype
   oldtype\_array[], MPI\_Datatype \*newtype)
- int MPI\_Type\_create\_subarray (int ndims, int
   size\_array[], int subsize\_array[], int
   start\_array[], int order, MPI\_Datatype
   oldtype, MPI\_Datatype \*newtype)
- int MPI\_Get\_address (void \*location, MPI\_Aint
   \*address)
- int MPI\_Type\_size (MPI\_Datatype \*datatype,
   int \*size)
- int MPI\_Type\_get\_extent (MPI\_Datatype
   datatype, MPI Aint \*lb, MPI Aint \*extent)
- int MPI\_Unpack (void \*inbuf, int insize, int
   \*position, void \*outbuf, int outcount,
   MPI Datatype datatype, MPI Comm comm)
- int MPI\_Pack\_size (int incount, MPI\_Datatype
   datatype, MPI Comm comm, int \*size)
- Related: MPI\_Type\_create\_hvector,

  MPI\_Type\_create\_hindexed,

  MPI\_Type\_create\_indexed\_block,

  MPI\_Type\_create\_darray,

  MPI\_Type\_create\_resized,

  MPI\_Type\_get\_true\_extent, MPI\_Type\_dup,

  MPI\_Pack\_external, MPI\_Unpack\_external,

  MPI\_Pack\_external\_size

#### **Groups and Communicators:**

- int MPI\_Group\_size (MPI\_Group group, int
   \*size)
- int MPI\_Group\_rank (MPI\_Group group, int
   \*rank)
- int MPI\_Comm\_group (MPI\_Comm comm, MPI\_Group
   \*group)

- int MPI\_Group\_translate\_ranks (MPI\_Group
   group1, int n, int rank1\_array[],
   MPI\_Group group2, int rank2\_array[])

- int MPI\_Group\_incl (MPI\_Group group, int n,
   int rank\_array[], MPI\_Group \*newgroup)
- Related: MPI Group excl
- int MPI\_Comm\_create (MPI\_Comm comm, MPI\_Group
   group, MPI Comm \*newcomm)
- int MPI\_Comm\_compare (MPI\_Comm comm1,
   MPI\_Comm comm2, int \*result)
   MPI\_IDENT, MPI\_COMGRUENT, MPI\_SIMILAR,
   MPI\_UNEQUAL
- int MPI\_Comm\_dup (MPI\_Comm comm, MPI\_Comm
   \*newcomm)
- int MPI\_Comm\_split (MPI\_Comm comm, int color,
   int key, MPI\_Comm \*newcomm)
- int MPI\_Comm\_free (MPI\_Comm \*comm)

#### **Topologies:**

- int MPI\_Dims\_create (int nnodes, int ndims,
   int \*dims)
- int MPI\_Cart\_create (MPI\_Comm comm\_old, int
   ndims, int dims\_array[], int
   periods\_array[], int reorder, MPI\_Comm
   \*comm cart)
- int MPI\_Cart\_shift (MPI\_Comm comm, int
   direction, int disp, int \*rank\_source,
   int \*rank\_dest)
- int MPI\_Cartdim\_get (MPI\_Comm comm, int
   \*ndim)
- int MPI\_Cart\_get (MPI\_Comm comm, int naxdim,
   int \*dims, int \*periods, int \*coords)
- int MPI\_Cart\_rank (MPI\_Comm comm, int
   coords array[], int \*rank)
- int MPI\_Cart\_coords (MPI\_Comm comm, int rank,
   int maxdims, int \*coords)

- int MPI\_Cart\_sub (MPI\_Comm comm\_old, int
   remain dims array[], MPI Comm \*comm new)
- int MPI\_Cart\_map (MPI\_Comm comm\_old, int
   ndims, int dims\_array[], int
   periods\_array[], int \*new\_rank)
- int MPI\_Graph\_create (MPI\_Comm comm\_old, int
   nnodes, int index\_array[], int
   edges\_array[], int reorder, MPI\_Comm
   \*comm graph)
- int MPI\_Graph\_neighbors\_count (MPI\_Comm comm,
   int rank, int \*nneighbors)
- int MPI\_Graph\_neighbors (MPI\_Comm comm, int
   rank, int maxneighbors, int \*neighbors)
- int MPI\_Graphdims\_get (MPI\_Comm comm, int
   \*nnodes, int \*nedges)
- int MPI\_Graph\_get (MPI\_Comm comm, int
   maxindex, int maxedges, int \*index, int
   \*edges)
- int MPI\_Graph\_map (MPI\_Comm comm\_old, int
   nnodes, int index\_array[], int
   edges array[], int \*new rank)
- int MPI\_Topo\_test (MPI\_Comm comm, int
   \*topo type)

#### Wildcards:

MPI ANY TAG, MPI ANY SOURCE

#### **Basic Datatypes:**

MPI\_CHAR, MPI\_SHORT, MPI\_INT, MPI\_LONG,
MPI\_UNSIGNED\_CHAR, MPI\_UNSIGNED\_SHORT,
MPI\_UNSIGNED, MPI\_UNSIGNED\_LONG MPI\_FLOAT,
MPI\_DOUBLE, MPI\_LONG\_DOUBLE, MPI\_BYTE,
MPI\_PACKED

#### **Predefined Groups and Communicators:**

MPI\_GROUP\_EMPTY, MPI\_GROUP\_NULL,
MPI\_COMM\_WORLD, MPI\_COMM\_SELF, MPI\_COMM\_NULL

### **Reduction Operations:**

MPI\_MAX, MPI\_MIN, MPI\_SUM, MPI\_PROD,
MPI\_BAND, MPI\_BOR, MPI\_BXOR, MPI\_LAND,
MPI\_LOR, MPI\_LXOR

## **Status Object:**

status.MPI\_SOURCE, status.MPI\_TAG,
status.MPI ERROR