MPI Befehle

```
                    Initialisierung und Abschluss
                    =============================

------------------------------------------------------------------------------

C: int MPI_Init(int *argc,char ***argv);
F:     MPI_INIT(INTEGER IERROR)

------------------------------------------------------------------------------

C: int MPI_Finalize(void);
F:     MPI_FINALIZE(INTEGER IERROR)

------------------------------------------------------------------------------


##############################################################################

                    Bestimmung der Kommunikationsparameter
                    ======================================

------------------------------------------------------------------------------

C: int MPI_Comm_size(MPI_Comm comm,int *size);
F:     MPI_COMM_SIZE(INTEGER COMM,INTEGER SIZE,INTEGER IERROR)

------------------------------------------------------------------------------

C: int MPI_Comm_rank(MPI_Comm comm,int *rank);
F:     MPI_COMM_RANK(INTEGER COMM,INTEGER RANK,INTEGER IERROR)

------------------------------------------------------------------------------


##############################################################################

                    Globale Kommunikation
                    =====================

------------------------------------------------------------------------------

C: int MPI_Bcast(void* buffer, int count, MPI_Datatype datatype,
                 int root, MPI_Comm comm);

       Datentypen:  MPI_INT, MPI_FLOAT, MPI_DOUBLE, ...


F:     MPI_BCAST(CHOICE BUFFER,INTEGER COUNT,INTEGER DATATYPE,INTEGER ROOT,
                 INTEGER COMM,INTEGER IERROR)

       Datentypen:  MPI_INTEGER, MPI_REAL, MPI_DOUBLEPRECISION, ...

------------------------------------------------------------------------------

C: int MPI_Reduce(void* sendbuf,void* recvbuf,int count,
                  MPI_Datatype datatype,MPI_Op op,int root,MPI_Comm comm);

F:     MPI_REDUCE(CHOICE SENDBUF,CHOICE RECVBUF,INTEGER COUNT,
                  INTEGER DATATYPE,INTEGER OP,INTEGER ROOT,INTEGER COMM,
                  INTEGER IERROR)

       Operationen: MPI_SUM, MPI_PROD, MPI_MIN, MPI_MAX, ...

------------------------------------------------------------------------------
```

MPI Befehle

```
C: int MPI_Allreduce(void* sendbuf,void* recvbuf,int count,
                     MPI_Datatype datatype,MPI_Op op,MPI_Comm comm);

F:     MPI_ALLREDUCE(CHOICE SENDBUF,CHOICE RECVBUF,INTEGER COUNT,
                     INTEGER DATATYPE,INTEGER OP,INTEGER COMM,INTEGER IERROR)

------------------------------------------------------------------------------

Weitere: MPI_Scatter, MPI_Gather, MPI_Allgather

------------------------------------------------------------------------------


##############################################################################

                    Punkt-zu-Punkt-Kommunikation
                    ============================

------------------------------------------------------------------------------

C: int MPI_Send(void* buf,int count,MPI_Datatype datatype,
                int dest,int tag,MPI_Comm comm);

F:     MPI_SEND(CHOICE BUF,INTEGER COUNT,INTEGER DATATYPE,INTEGER DEST,
                INTEGER TAG,INTEGER COMM,INTEGER IERROR)

------------------------------------------------------------------------------

C: int MPI_Recv(void* buf,int count,MPI_Datatype datatype,
                int source,int tag,MPI_Comm comm,MPI_Status *status);

F:     MPI_RECV(CHOICE BUF,INTEGER COUNT,INTEGER DATATYPE,INTEGER SOURCE,
                INTEGER TAG,INTEGER COMM,INTEGER STATUS(MPI_STATUS_SIZE),
                INTEGER IERROR)

------------------------------------------------------------------------------


##############################################################################

 Vollständige Befehlsreferenz:

 z.B: http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp
           ?topic=/com.ibm.cluster.pe_linux43.mpisub.doc/am107_mpimsr.html
```