→ #include "mpi.h"

• executable run individually on each node

→ mpirun would run program on all specified processors instead of manually executing them by a/a.out in each process

by default all process part of single communicator MPI_COMM_WORLD.
Any 2 process may communicate with each other in this scenario.

Within communicator, each process has a rank (ID).
ranked 0 to n-1.

```
int main (int argc, char **argv)
{  MPI_Init (&argc, &argv)

   MPI_Comm_rank (MPI_COMM_WORLD, &myrank);
                  ↑ communicator      ↑ variable where rank is to be saved
   if (myrank == 0)
      master();   ←
   else slave();  ←   — 2 c functions that run required programs

   MPI_Finalize()  ← to terminate/close MPI
}
```

MPI_Comm_size (MPI_COMM_WORLD, &grpSize);
saves no. of processes in this comm grp to grpSize

rank = 0, we send message to 1.
i.e. we get message from 0.

```
Communication;
    :

int myrank;
MPI_Comm_Rank ( MPI_COMM_WORLD, &myrank );
if (myrank == 0)
{   int x;
        MPI_Send (&x, 1, MPI_INT, 1, msgtag, MPI_COMM_WORLD);
} else if (myrank == 1)
{   int x;
    MPI_Recv (&x, 1, MPI_INT, 0, msgtag, MPI_COMM_WORLD, status);
}
```

here, if rank = 1

↓ int
to differentiate b/w different types of messages sent

likewise MPI_ANY_SOURCE
says that data can come from any source ← MPI_ANY_TAG [don't care which recieve picks data]

Synchronous send & recieve wait until data has been accepted/recieved by target.
→ Also, helps synchronise processes.

MPI_Send (Address of send buffer, No of items to be sent, data_type, rank of destination, msgtab, Comm)

MPI_Isend (..., request) used with MPI_Wait()
Blocking vs Non-blocking → returns immediately.
↳ waits until local operations (save data, network active ...) completed

has int root // process that contains data to be broadcast

→ M_bcast (...) ↓ sends to all processes in Comm group.
→ M_scatter (...): if I have list of data, I want to split & send to different processes. [will scatter into root as well]
→ M_gather (...) : opposite of scatter.
→ M_reduce (...) : opposite of bcast, but what fn used to re
Operation of reduction is to be mentioned