Installation of OPENMPI

1. Download openmpi-4.1.4.tar.bz2 from http://www.open-mpi.org in a folder say **LP5.**
2. Goto the terminal (Command prompt)
3. update using

    sudo apt-get update

    sudo apt install gcc {if not already installed}
4. Goto the directory which contains the downloaded file
5. Extract the files using

    tar -jxf openmpi-4.1.4.tar.bz2
6. The directory openmpi-4.1.4 is created
7. Configure, compile and install by executing the following commands

    ./configure --prefix=$HOME/opt/openmpi

    make all

    make install
8. Now openmpi folder is created in **'opt'** folder of Home directory.
9. Now the folder LP5 can be deleted (optional)
10. Update the PATH and LD_LIBRARY_PATH environment variable using

    echo "export PATH=\$PATH:\$HOME/opt/openmpi/bin" >> $HOME/.bashrc

    echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$HOME/opt/openmpi/lib">>$HOME/.bashrc

11. Compile the program using

    mpicc ***name of the program***

12. Execute the program using

    *mpirun -np N **./a.out***

Hello world program

nllabc2d22@nllabc2d-22:~/opt/openmpi/bin$ ***gedit hello.c***

```
#include <stdio.h>
#include "mpi.h"
int main(int argc, char* argv[])
{
   int rank, size, len;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);
   printf("Hello, world, I am %d of %d\n",rank, size);
   MPI_Finalize();
   return 0;
}
```

Compile the program

nllabc2d22@nllabc2d-22:~/opt/openmpi/bin$ ***mpicc hello.c***

Execute the program using 2 cores

nllabc2d22@nllabc2d-22:~/opt/openmpi/bin$ ***mpirun -np 2 ./a.out***

Hello, world, I am 0 of 2

Hello, world, I am 1 of 2

Execute the program using 4 cores

nllabc2d22@nllabc2d-22:~/opt/openmpi/bin$ ***mpirun -np 4 ./a.out***

Hello, world, I am 0 of 4

Hello, world, I am 3 of 4

Hello, world, I am 1 of 4

Hello, world, I am 2 of 4

Program to transfer data from core 0 to core 1.

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char* argv[])
{
   int rank, size, len;
   int num=10;
   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);
   if(rank == 0)
  {
        printf("Sending message containing: %d from rank %d\n", num,rank);
        MPI_Send(&num, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);
  }
  else
  {
        printf(" at rank %d\n",rank);
        MPI_Recv(&num, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        printf("Received message containing: %d at rank %d\n", num,rank);
  }
  MPI_Finalize();
  return 0;
}
```

```
Sending message containing: 10 from rank 0
 at rank 1
 at rank 3
Received message containing: 10 at rank 1
 at rank 2
```

/*** The cores 2 and will be in waiting mode … Press Ctrl+z to end the execution *******/***

Assignment program: Add 20 numbers in an array using 4 cores

```
#include <stdio.h>
#include "mpi.h"

int main(int argc, char* argv[])
{
   int rank, size;
   int num[20];  //N=20, n=4

   MPI_Init(&argc, &argv);
   MPI_Comm_rank(MPI_COMM_WORLD, &rank);
   MPI_Comm_size(MPI_COMM_WORLD, &size);
  for(int i=0;i<20;i++)
        num[i]=i+1;

 if(rank == 0){
        int s[4];
        printf("Distribution at rank %d \n", rank);
        for(int i=1;i<4;i++)
        MPI_Send(&num[i*5], 5, MPI_INT, i, 1, MPI_COMM_WORLD); //N/n i.e. 20/4=5
        int sum=0, local_sum=0;
        for(int i=0;i<5;i++)
        {
                local_sum=local_sum+num[i];
        }
        for(int i=1;i<4;i++)
        {
                 MPI_Recv(&s[i], 1, MPI_INT, i, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        }
        printf("local sum at rank %d is %d\n", rank,local_sum);
        sum=local_sum;
        for(int i=1;i<4;i++)
                 sum=sum+s[i];
        printf("final sum = %d\n\n",sum);
   }
```

```
    else
    {
        int k[5];
         MPI_Recv(k, 5, MPI_INT, 0, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        int local_sum=0;
        for(int i=0;i<5;i++)
        {
                local_sum=local_sum+k[i];
        }
        printf("local sum at rank %d is %d\n", rank, local_sum);
        MPI_Send(&local_sum, 1, MPI_INT, 0, 1, MPI_COMM_WORLD);

    }
    MPI_Finalize();

    return 0;
}
```

```
Distribution at rank 0
local sum at rank 1 is 40
local sum at rank 2 is 65
local sum at rank 3 is 90
local sum at rank 0 is 15
final sum = 210
```

/*** students can be asked to take dynamic values for N, n and array ************/***