**High Performance Computing Lab**

**Assignment No. 7**

**Name : Sourabh Shankar Patil**

**PRN : 21510045**

**Batch : B2**

**Title:** Installation of MPI & Implementation of basic functions of MPI

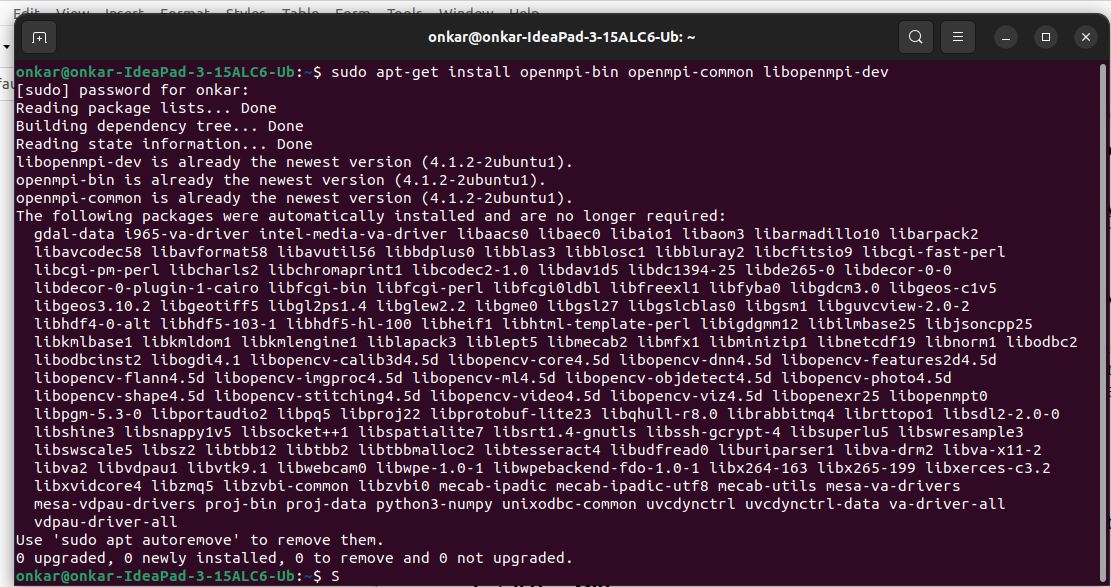
MPI (Message Passing Interface) is a standard used for parallel programming in distributed computing environments. It allows processes to communicate with one another by sending and receiving messages, making it ideal for tasks where multiple processors need to collaborate to solve a problem.

### **Installation of MPI:**

To work with MPI, you need to install an MPI implementation, such as OpenMPI or MPICH.

Install OpenMPI:

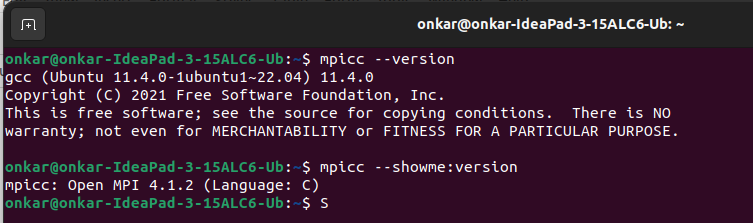
sudo apt-get install openmpi-bin openmpi-common libopenmpi-dev



Verify the installation: Check the version of MPI installed:

mpicc --version

**mpicc --showme:version**

****

### 2. **Implementation of Basic MPI Functions**

#### a. **MPI\_Init & MPI\_Finalize**

* MPI\_Init(&argc, &argv): Initializes the MPI environment, preparing it for message passing.
* MPI\_Finalize(): Cleans up the MPI environment when all processes have finished executing.

#### b. **MPI\_Comm\_rank & MPI\_Comm\_size**

* MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank): Retrieves the rank (ID) of the current process within the communicator group MPI\_COMM\_WORLD.
* MPI\_Comm\_size(MPI\_COMM\_WORLD, &size): Determines the total number of processes in the communicator group.

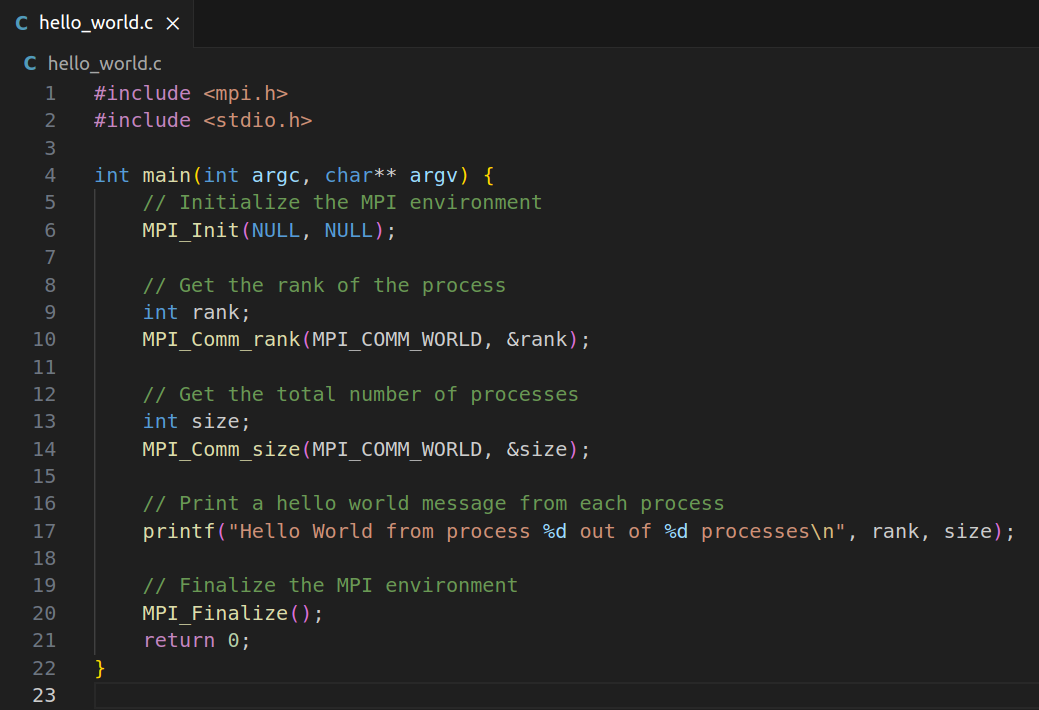
MPI provides fundamental functions for initializing, communicating, and terminating parallel processes, making it a powerful tool for distributed computing tasks. It supports scaling computations across multiple processes running on different machines.

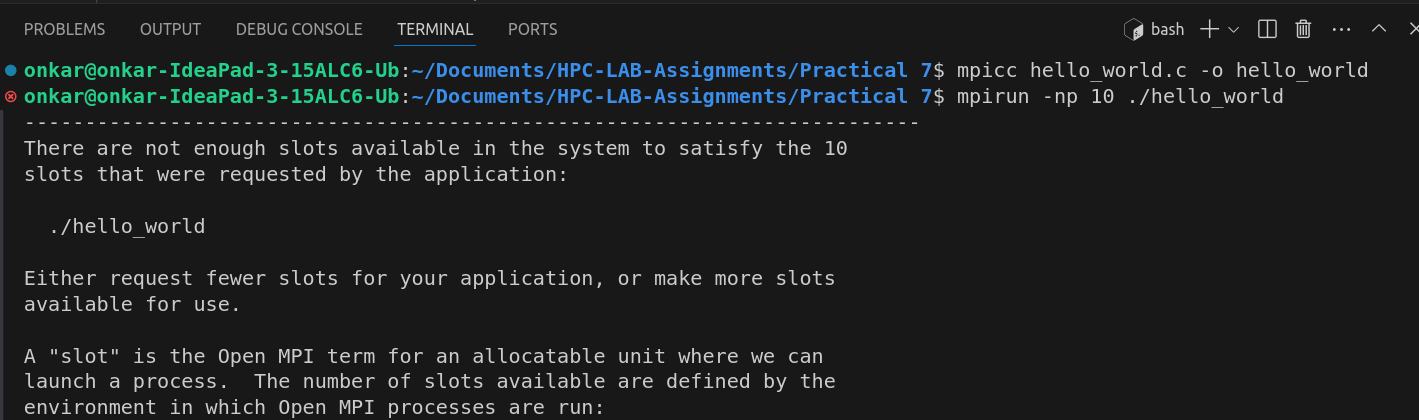
Implement following Programs using MPI:

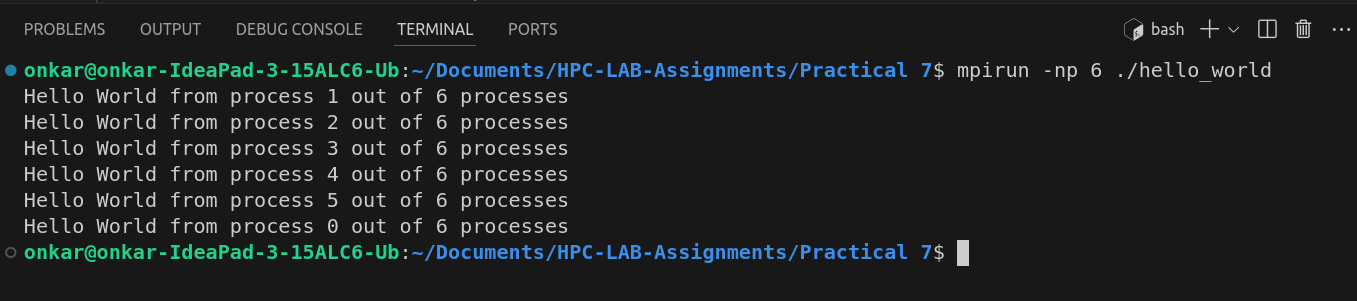
**Problem Statement 1: Hello World Program with 10 Processes**

Implement a simple hello world program by setting number of processes equal to 10

**Screenshots:**







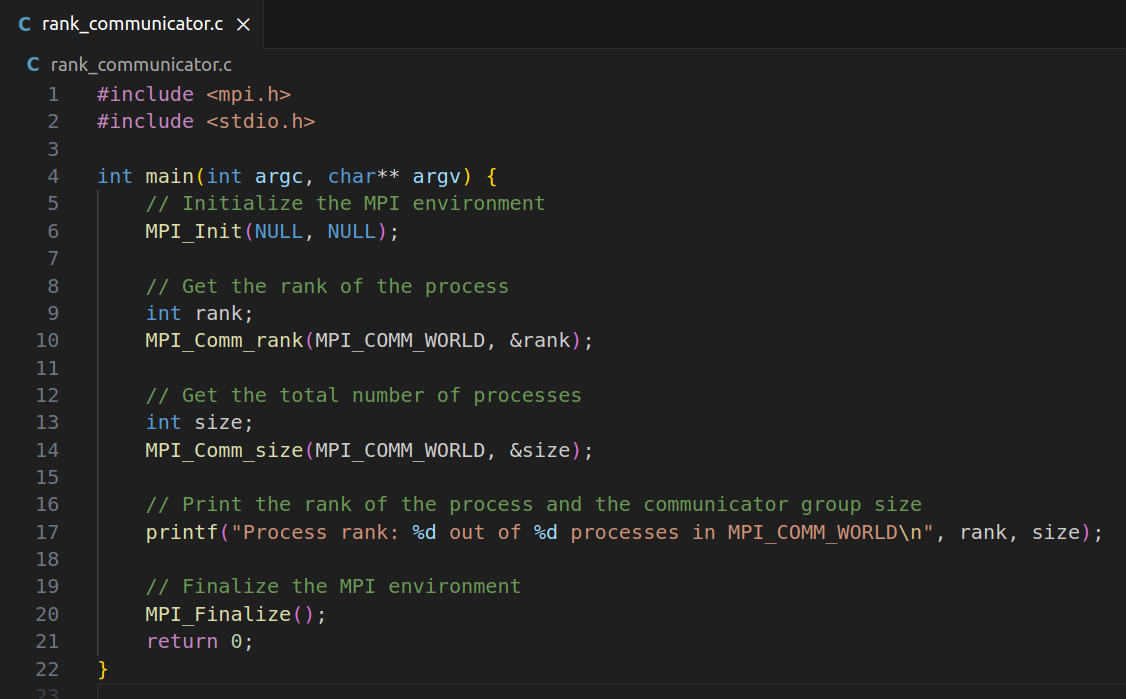
Information:

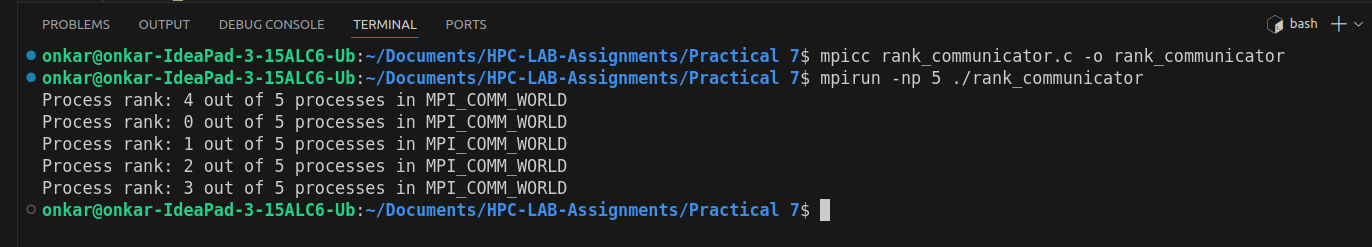
* The program initializes the MPI environment, determines the rank of each process, and prints a message from each process.
* The MPI\_Comm\_rank function provides the rank (ID) of each process, and MPI\_Comm\_size returns the total number of processes.
* MPI\_Finalize is used to clean up the MPI environment after all processes have finished execution.

**Problem Statement 2: Display Rank and Communicator Group of Five Processes**

Implement a program to display rank and communicator group of five processes

**Screenshots:**





Information:

* The rank of a process refers to its ID in the communicator group MPI\_COMM\_WORLD.
* MPI\_COMM\_WORLD is the default communicator that includes all processes.
* The communicator group size, obtained using MPI\_Comm\_size, ensures all processes are part of the same group and facilitates communication between them.

**Github Link:**

[https://github.com/onkaryemul/HPC-LAB-Assignments/tree/main/Practical%20](https://github.com/onkaryemul/HPC-LAB-Assignments/tree/main/Practical%206)7