###### 11b

###### Implement the program IPC/IPS using MPI library. Communication in between processes OS’s: Unix.

#include <stdio.h>

#include <mpi.h>

int main(int argc, char \*argv[]) {

int rank, size, value;

MPI\_Status status;

MPI\_Init(&argc, &argv);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

if (rank == 0) {

value = 42;

MPI\_Send(&value, 1, MPI\_INT, 1, 0, MPI\_COMM\_WORLD);

printf("Process %d sent value %d to process %d.\n", rank, value, rank + 1);

} else if (rank == size - 1) {

MPI\_Recv(&value, 1, MPI\_INT, rank - 1, 0, MPI\_COMM\_WORLD, &status);

printf("Process %d received value %d from process %d.\n", rank, value, rank - 1);

} else {

MPI\_Recv(&value, 1, MPI\_INT, rank - 1, 0, MPI\_COMM\_WORLD, &status);

printf("Process %d received value %d from process %d.\n", rank, value, rank - 1);

value \*= 2;

MPI\_Send(&value, 1, MPI\_INT, rank + 1, 0, MPI\_COMM\_WORLD);

printf("Process %d sent value %d to process %d.\n", rank, value, rank + 1);

}

MPI\_Finalize();

return 0;

}

###### run--

Ensure that you have the MPI library installed. You can install it by running the appropriate commands based on your operating system and package manager. For example, on Ubuntu, you can use the following command:

arduino

* sudo apt-get install libopenmpi-dev
* Save the code into a file, for example, mpi\_ipc\_ips.c.
* Open a terminal and navigate to the directory where you saved the file.
* Compile the program using the mpicc command:
* mpicc mpi\_ipc\_ips.c -o mpi\_ipc\_ips
* After successful compilation, you can execute the program using the mpirun command:

bash

* mpirun -np <number\_of\_processes> ./mpi\_ipc\_ips

Replace <number\_of\_processes> with the desired number of processes (e.g., 4).

* The program will start running, and you'll see the output in the terminal, showing the communication between processes and the values being passed.