Experiment 2

1.Install mpich

sudo apt update

sudo apt install mpich

Step 1: Create the MPI Program

nano mpi_string_reverse.c

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

void reverse_string(char* str) {
    int n = strlen(str);
    for (int i = 0; i < n / 2; i++) {
        char temp = str[i];
        str[i] = str[n - i - 1];
        str[n - i - 1] = temp;
    }
}

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

    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    if (world_size < 2) {
```

```
        fprintf(stderr, "World size must be greater than 1 for %s\n", argv[0]);
        MPI_Abort(MPI_COMM_WORLD, 1);
    }

    if (world_rank == 0) {
        // Client (rank 0)
        char message[100] = "Hello from Client!";
        printf("Client sending message: %s\n", message);

        MPI_Send(message, strlen(message) + 1, MPI_CHAR, 1, 0, MPI_COMM_WORLD);
        MPI_Recv(message, 100, MPI_CHAR, 1, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        printf("Client received reversed message: %s\n", message);
    }
    else if (world_rank == 1) {
        // Server (rank 1)
        char message[100];
        MPI_Recv(message, 100, MPI_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        printf("Server received message: %s\n", message);

        // Reverse the string
        reverse_string(message);
        MPI_Send(message, strlen(message) + 1, MPI_CHAR, 0, 0, MPI_COMM_WORLD);
    }

    MPI_Finalize();
    return 0;
}
```

Step 2: Compile the MPI Program

```
mpicc -o mpi_string_reverse mpi_string_reverse.c
```

Step 3: Run the Program

mpirun -np 2 ./mpi_string_reverse

output:

Client sending message: Hello from Client!

Server received message: Hello from Client!

Client received reversed message: !tneilC morf olleH