

CSE4001 - Parallel and Distributed Computing

Lab 21+22

Digital Assignment- 4

Submitted by: Alokam Nikhitha

Reg No:19BCE2555

QUESTION:

Write a C program to handle message passing in the MPI application interface, which allows processes to communicate with one another. Create two processes that will pass the number 17 from one to the other.

CODE:

```
#include <stdio.h>

#include <mpi.h>

int main(int argc, char** argv) {

    int process_Rank, size_Of_Cluster, message_Item;

    MPI_Init(&argc, &argv);

    MPI_Comm_size(MPI_COMM_WORLD, &size_Of_Cluster);

    MPI_Comm_rank(MPI_COMM_WORLD, &process_Rank);

    if(process_Rank == 0){

        message_Item = 17;

        MPI_Send(&message_Item, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);

        printf("Message Sent from Process 0: %d\n", message_Item);

    }

    else if(process_Rank == 1){

        MPI_Recv(&message_Item, 1, MPI_INT, 0, 1, MPI_COMM_WORLD,
        MPI_STATUS_IGNORE);

        printf("Message Received in Process 1: %d\n", message_Item);

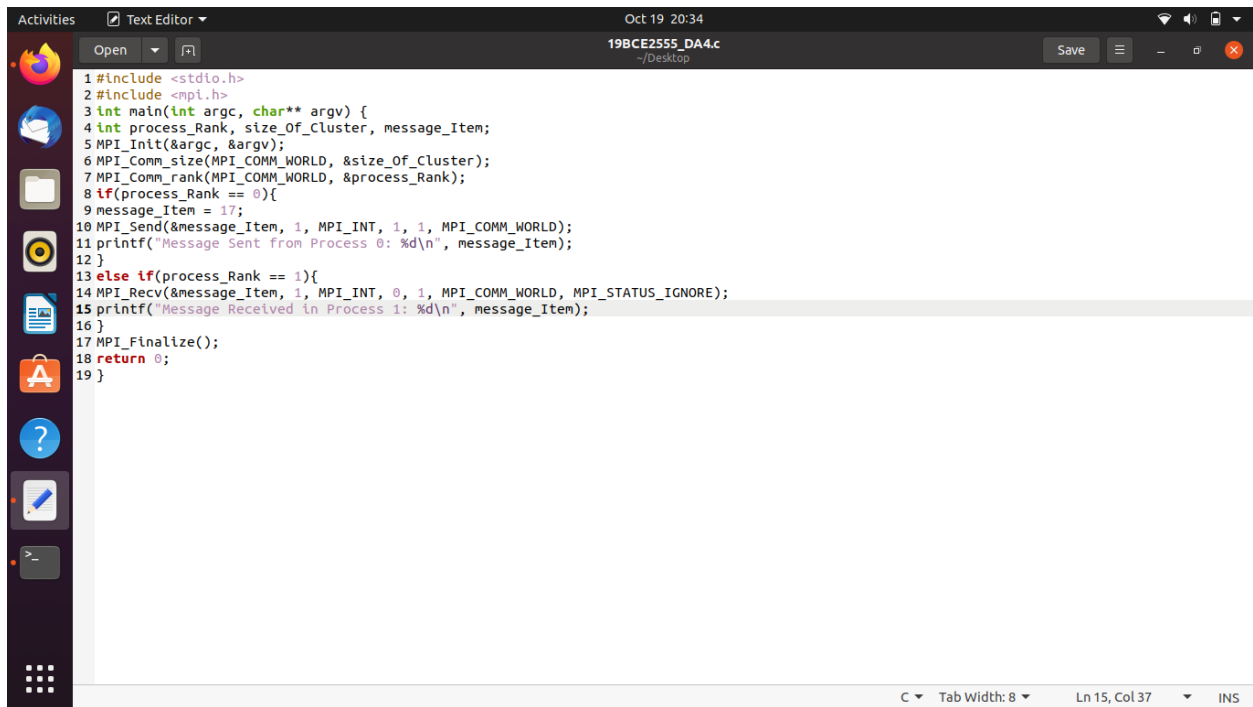
    }

    MPI_Finalize();

    return 0;

}
```

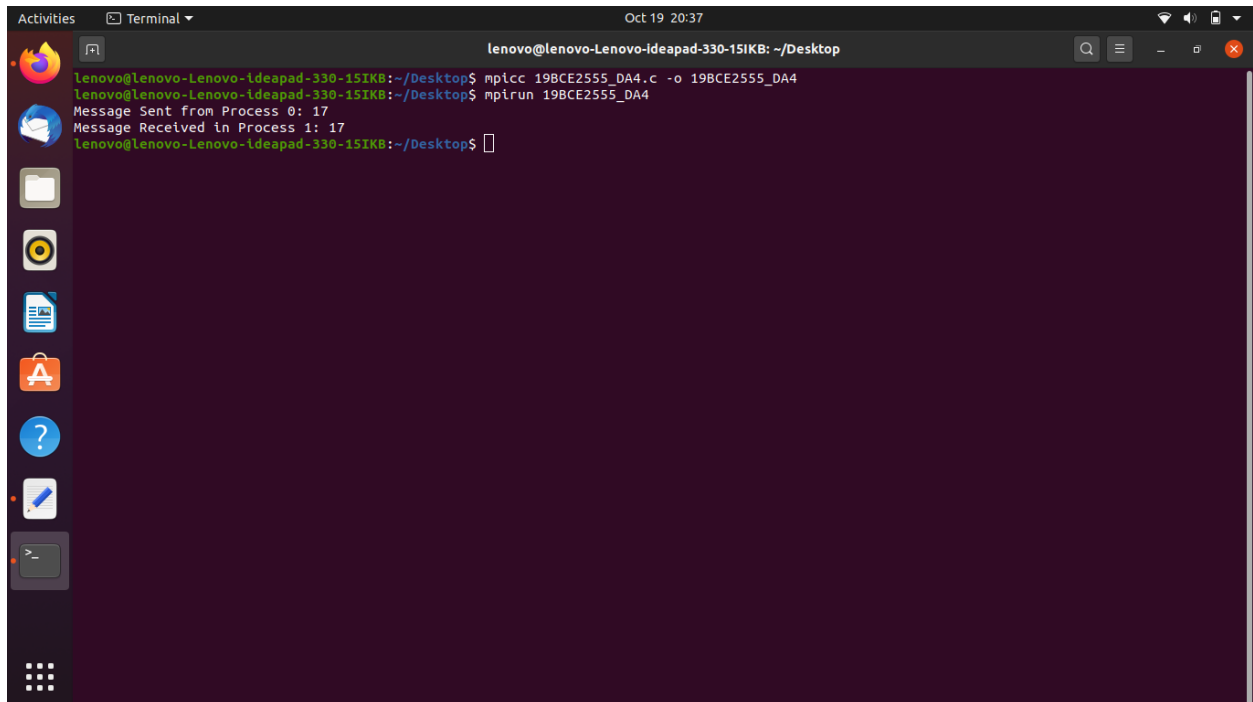
CODE SNIPPETS:



The screenshot shows a text editor window titled "19BCE2555_DA4.c" with the following C code:

```
1#include <stdio.h>
2#include <mpi.h>
3int main(int argc, char** argv) {
4    int process_Rank, size_Of_Cluster, message_Item;
5    MPI_Init(&argc, &argv);
6    MPI_Comm_size(MPI_COMM_WORLD, &size_Of_Cluster);
7    MPI_Comm_rank(MPI_COMM_WORLD, &process_Rank);
8    if(process_Rank == 0){
9        message_Item = 17;
10       MPI_Send(&message_Item, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);
11       printf("Message Sent from Process 0: %d\n", message_Item);
12   }
13   else if(process_Rank == 1){
14       MPI_Recv(&message_Item, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
15       printf("Message Received in Process 1: %d\n", message_Item);
16   }
17   MPI_Finalize();
18   return 0;
19 }
```

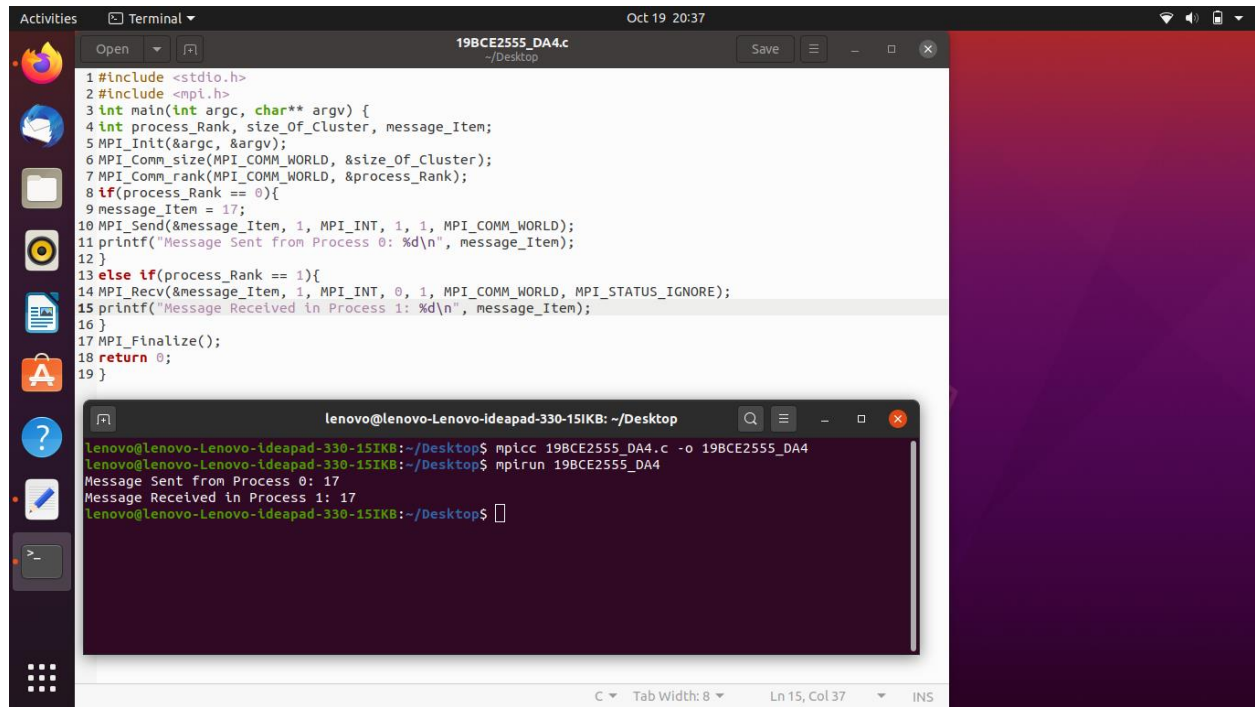
OUTPUT:



The screenshot shows a terminal window with the following output:

```
lenovo@lenovo-Lenovo-Ideapad-330-15IKB: ~/Desktop
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$ mpicc 19BCE2555_DA4.c -o 19BCE2555_DA4
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$ mpirun 19BCE2555_DA4
Message Sent from Process 0: 17
Message Received in Process 1: 17
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$
```

OUTPUT WITH CODE:



The screenshot displays a Linux desktop environment. On the left is a vertical dock with icons for various applications. The main workspace contains two windows. The top window, titled '19BCE2555_DA4.c', is a code editor showing the source code of a C program that uses MPI for inter-process communication. The bottom window is a terminal titled 'lenovo@lenovo-Lenovo-Ideapad-330-15IKB: ~/Desktop'. It shows the compilation of the program using 'mpicc' and its execution using 'mpirun'. The output of the program is visible in the terminal, showing messages sent and received by two processes.

```
1#include <stdio.h>
2#include <mpi.h>
3int main(int argc, char** argv) {
4int process_Rank, size_Of_Cluster, message_Item;
5MPI_Init(&argc, &argv);
6MPI_Comm_size(MPI_COMM_WORLD, &size_Of_Cluster);
7MPI_Comm_rank(MPI_COMM_WORLD, &process_Rank);
8if(process_Rank == 0){
9message_Item = 17;
10MPI_Send(&message_Item, 1, MPI_INT, 1, 1, MPI_COMM_WORLD);
11printf("Message Sent from Process 0: %d\n", message_Item);
12}
13else if(process_Rank == 1){
14MPI_Recv(&message_Item, 1, MPI_INT, 0, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
15printf("Message Recelved in Process 1: %d\n", message_Item);
16}
17MPI_Finalize();
18return 0;
19}
```

```
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$ mpicc 19BCE2555_DA4.c -o 19BCE2555_DA4
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$ mpirun 19BCE2555_DA4
Message Sent from Process 0: 17
Message Recelved in Process 1: 17
lenovo@lenovo-Lenovo-Ideapad-330-15IKB:~/Desktop$
```