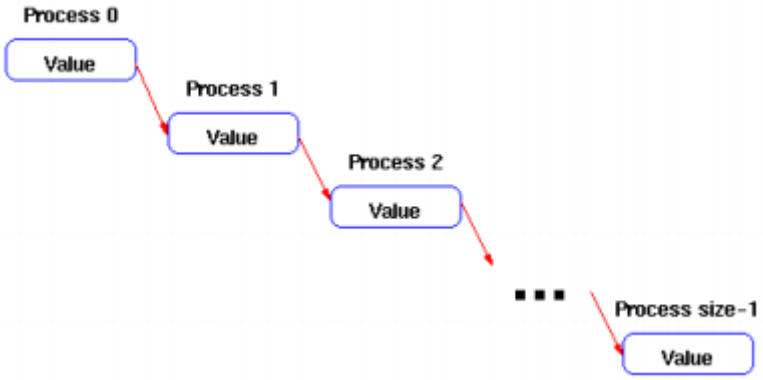
CSE4001 - Parallel and Distributed Computing, Fall 2019
Vellore Institute of Technology
Instructor: Prof Deebak B D - SCOPE

Lab report

Title of Lab: Sending in A Ring (Broadcast by Ring)
Assessment #: 10
Date: 18|10|2019

Author's name: Gagan Deep Singh
Registration ID: 17BCI0140
Lab section: Friday L59 + L60

AIM: Study the given C program that takes data from process zero and sends it to all of the other processes by sending it in a ring and its logical approach is based on MPI. That is, process i should receive the data and send it to process i+1, until the last process is reached. Analyse its key factors in terms of network application system.



Assume that the data consists of a single integer. Process zero reads the data from the user.

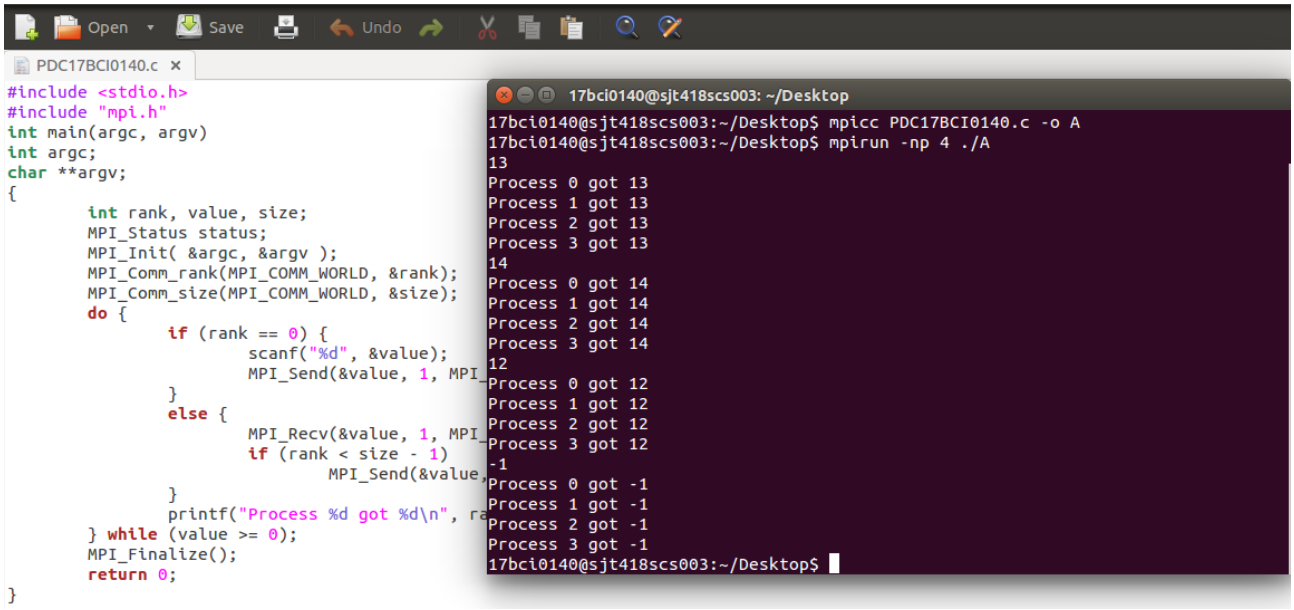
SOURCE CODE:

```
#include <stdio.h>
#include "mpi.h"
int main(argc, argv)
int argc;
char **argv;
{
        int rank, value, size;
        MPI_Status status;
        MPI_Init( &argc, &argv );
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
        MPI_Comm_size(MPI_COMM_WORLD, &size);
        do {
                if (rank == 0) {
                        scanf("%d", &value);
                        MPI_Send(&value, 1, MPI_INT, rank + 1, 0,
MPI_COMM_WORLD);
                }
                else {
                        MPI_Recv(&value, 1, MPI_INT, rank - 1, 0, MPI_COMM_WORLD,
&status);
                        if (rank < size - 1)
                                MPI_Send(&value, 1, MPI_INT, rank + 1, 0,
MPI_COMM_WORLD);
```

```
                }
                printf("Process %d got %d\n", rank, value);
        } while (value >= 0);
        MPI_Finalize();
        return 0;
}
```

Conceptual Discussion: Conceptual Discussion: Rank is the rank of each processes and size is the total number of processes. "MPI_Comm_rank(MPI_COMM_WORLD, &rank);" ranks the processes, here 0, 1, 2, 3. Now the program will take a value from the user by process with rank 0 and send this value to the process with rank 'rank + 1', the next process. All the processes with rank not equal to 0 will receive the value from the process with one less rank. Upon receiving the value if their rank is less than 'size - 1' i.e. till second last process, they will send the value to the next process. This will go on until the user enters a negative value. The first entered negative value will also be distributed but no further value will be taken, then the program will terminate.

Execution:



Result: In this type of MPI broadcast the resources, here a value entered by user is distributed in a sequential manner till it reaches the last process.