Parallel and Distributed Computing
CSE4001
Fall Semester 2020-21

Lab Assignment 8

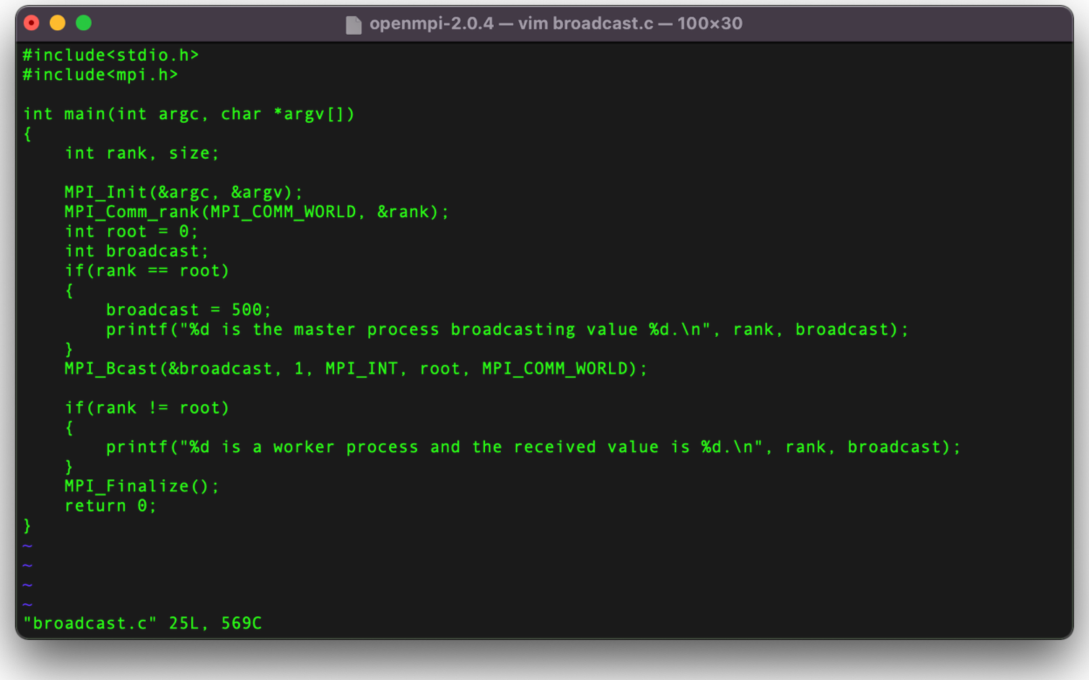
ISHAAN OHRI
18BCE0265

Aim:

Write a C program to demonstrate the use of MPI_Bcast()

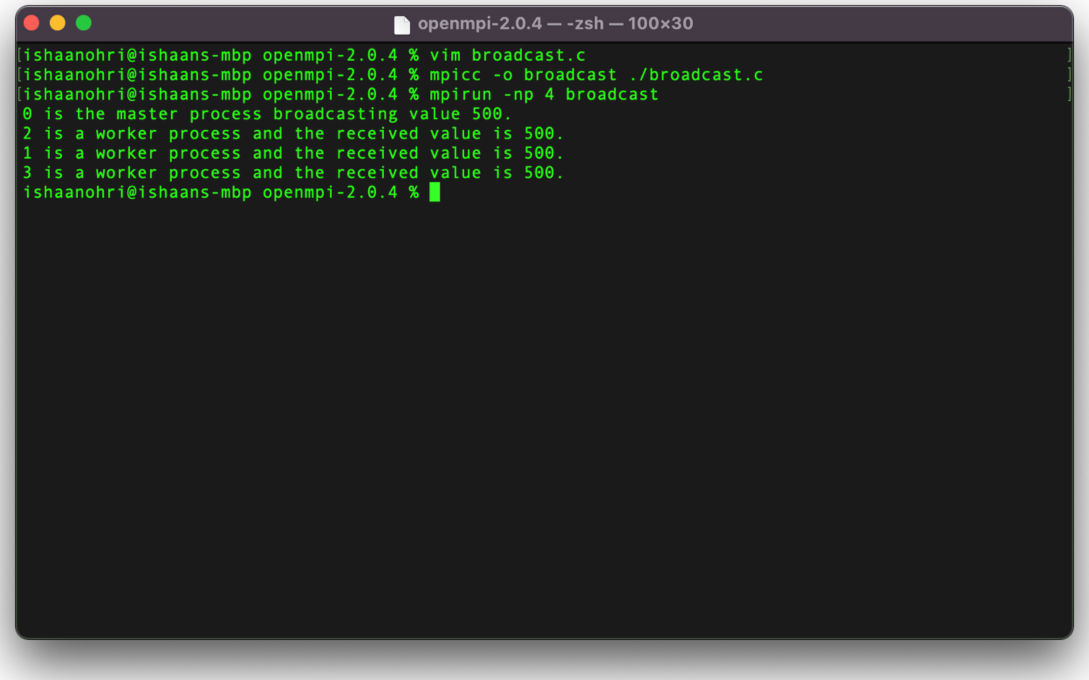
Case 1: **`broadcast = 500`**

Source Code:

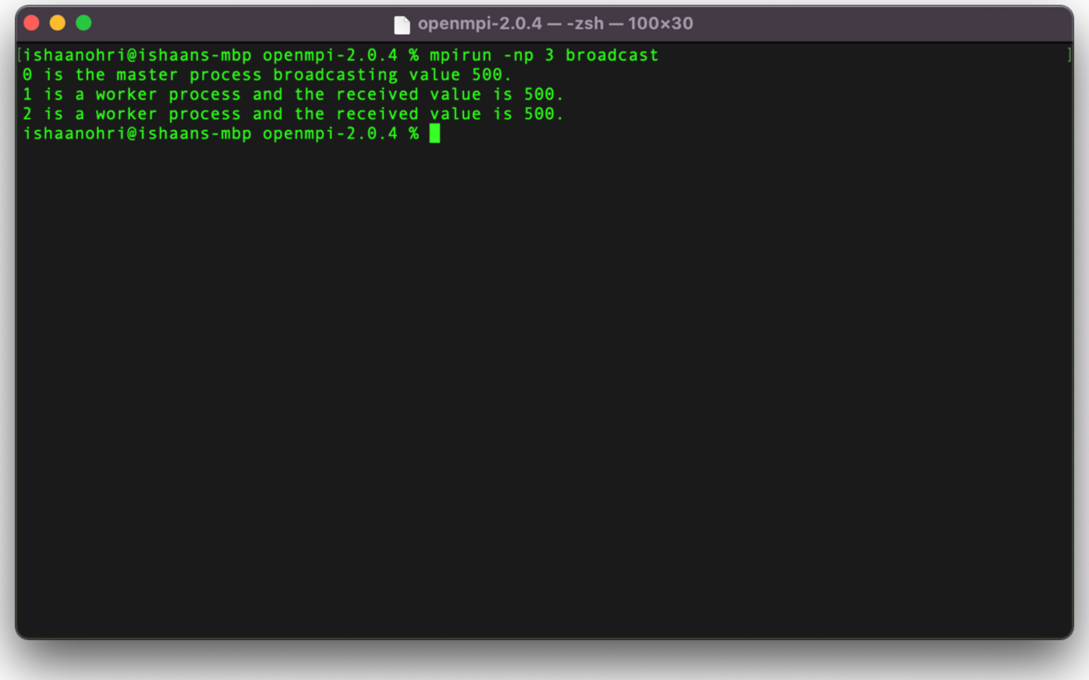


```
#include<stdio.h>
#include<mpi.h>

int main(int argc, char *argv[])
{
    int rank, size;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    int root = 0;
    int broadcast;
    if(rank == root)
    {
        broadcast = 500;
        printf("%d is the master process broadcasting value %d.\n", rank, broadcast);
    }
    MPI_Bcast(&broadcast, 1, MPI_INT, root, MPI_COMM_WORLD);

    if(rank != root)
    {
        printf("%d is a worker process and the received value is %d.\n", rank, broadcast);
    }
    MPI_Finalize();
    return 0;
}
~
~
~
~
"broadcast.c" 25L, 569C
```

Execution:



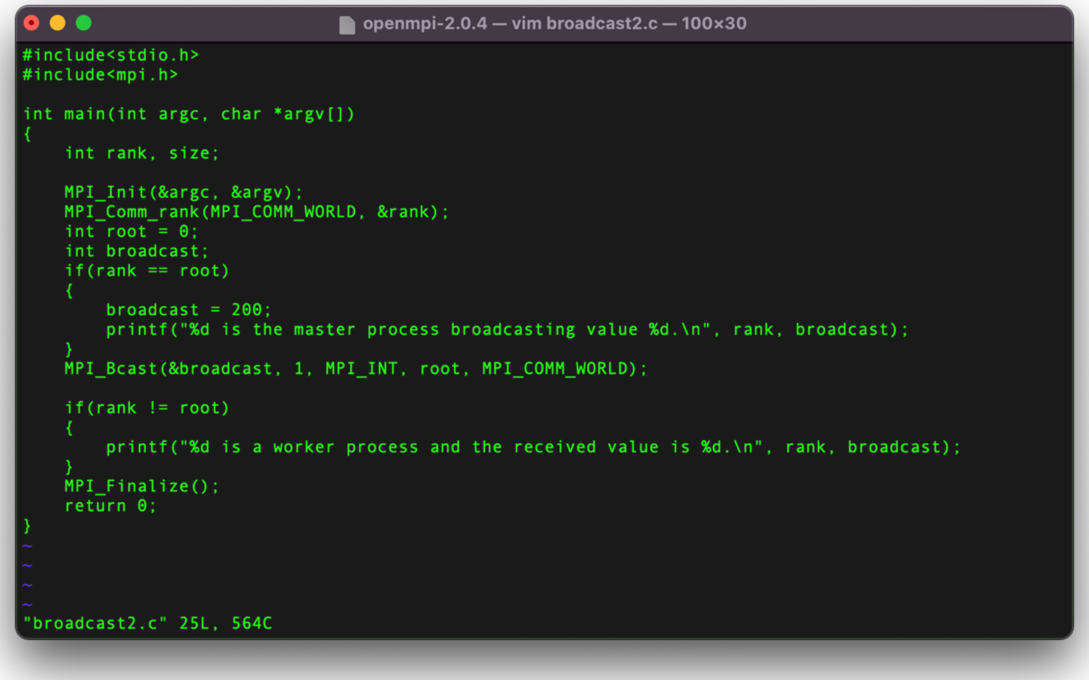
```
[ishaanohri@ishaans-mbp openmpi-2.0.4 % vim broadcast.c
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpicc -o broadcast ./broadcast.c
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpirun -np 4 broadcast
 0 is the master process broadcasting value 500.
 2 is a worker process and the received value is 500.
 1 is a worker process and the received value is 500.
 3 is a worker process and the received value is 500.
 ishaanohri@ishaans-mbp openmpi-2.0.4 % █
```



```
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpirun -np 3 broadcast
 0 is the master process broadcasting value 500.
 1 is a worker process and the received value is 500.
 2 is a worker process and the received value is 500.
 ishaanohri@ishaans-mbp openmpi-2.0.4 % █
```

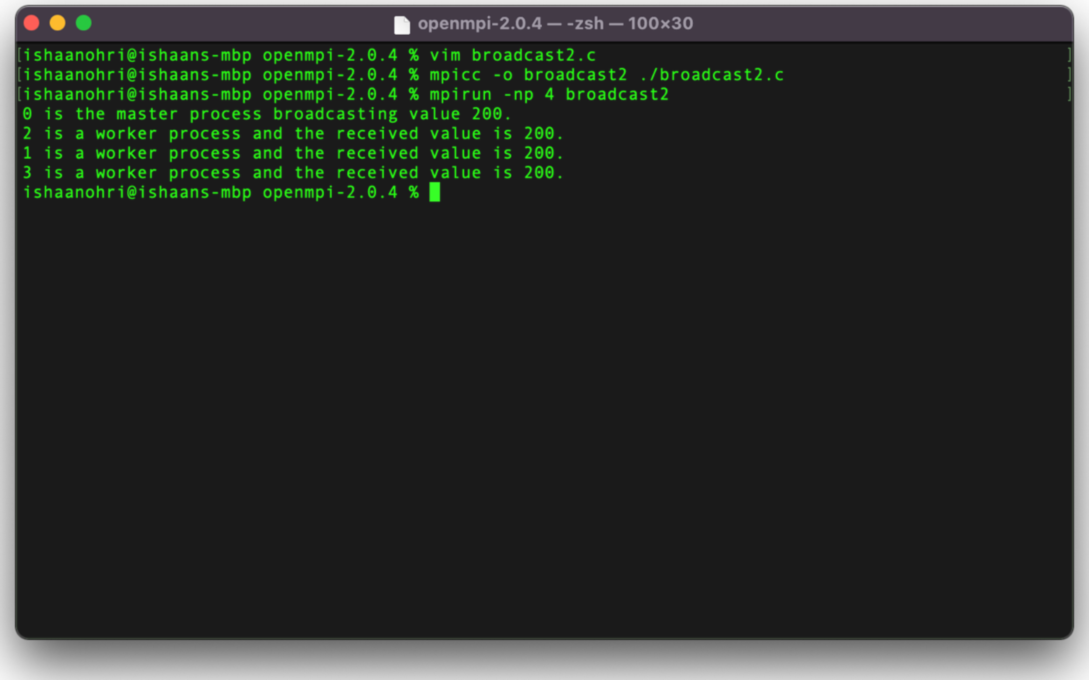
Case 2: `broadcast = 200`

Source Code:

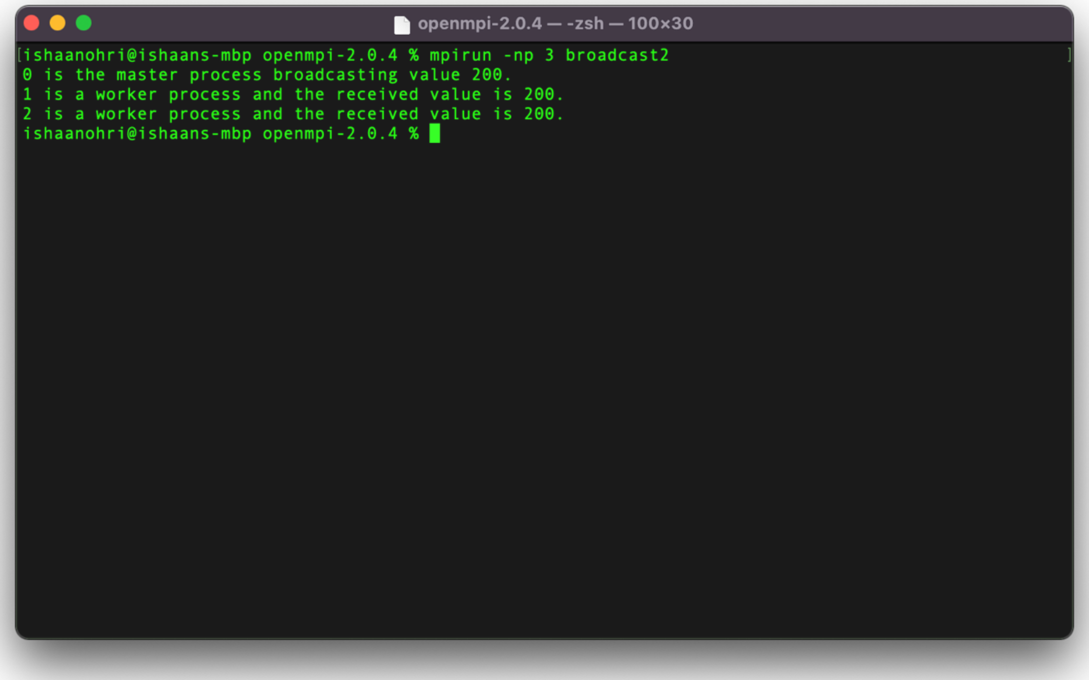


```
#include<stdio.h>
#include<mpi.h>

int main(int argc, char *argv[])
{
    int rank, size;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    int root = 0;
    int broadcast;
    if(rank == root)
    {
        broadcast = 200;
        printf("%d is the master process broadcasting value %d.\n", rank, broadcast);
    }
    MPI_Bcast(&broadcast, 1, MPI_INT, root, MPI_COMM_WORLD);

    if(rank != root)
    {
        printf("%d is a worker process and the received value is %d.\n", rank, broadcast);
    }
    MPI_Finalize();
    return 0;
}
~
~
~
~
"broadcast2.c" 25L, 564C
```

Execution:



```
[ishaanohri@ishaans-mbp openmpi-2.0.4 % vim broadcast2.c
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpicc -o broadcast2 ./broadcast2.c
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpirun -np 4 broadcast2
0 is the master process broadcasting value 200.
2 is a worker process and the received value is 200.
1 is a worker process and the received value is 200.
3 is a worker process and the received value is 200.
ishaanohri@ishaans-mbp openmpi-2.0.4 % 
```



```
[ishaanohri@ishaans-mbp openmpi-2.0.4 % mpirun -np 3 broadcast2
0 is the master process broadcasting value 200.
1 is a worker process and the received value is 200.
2 is a worker process and the received value is 200.
ishaanohri@ishaans-mbp openmpi-2.0.4 %
```

Remarks:

The MPI_Bcast() function helps a particular value to be broadcasted to all worker processes by the master process. It is helpful if a particular value needs to be transmitted to a lot of processes (Eg: 50), therefore saving us from writing 50 extra lines of code of MPI_Send() and MPI_Receive().

Over here we tried by sending a value of 500 in case 1 and 200 in case 2.

In the mpirun we specify the number of worker processes, over here it has been done for 4 and 3 worker processes.