CSE4001 – Parallel and Distributed Computing
Slot: L29 + L30

Name: Rathnam Sasidhar Achari
ID: 17BCE0895

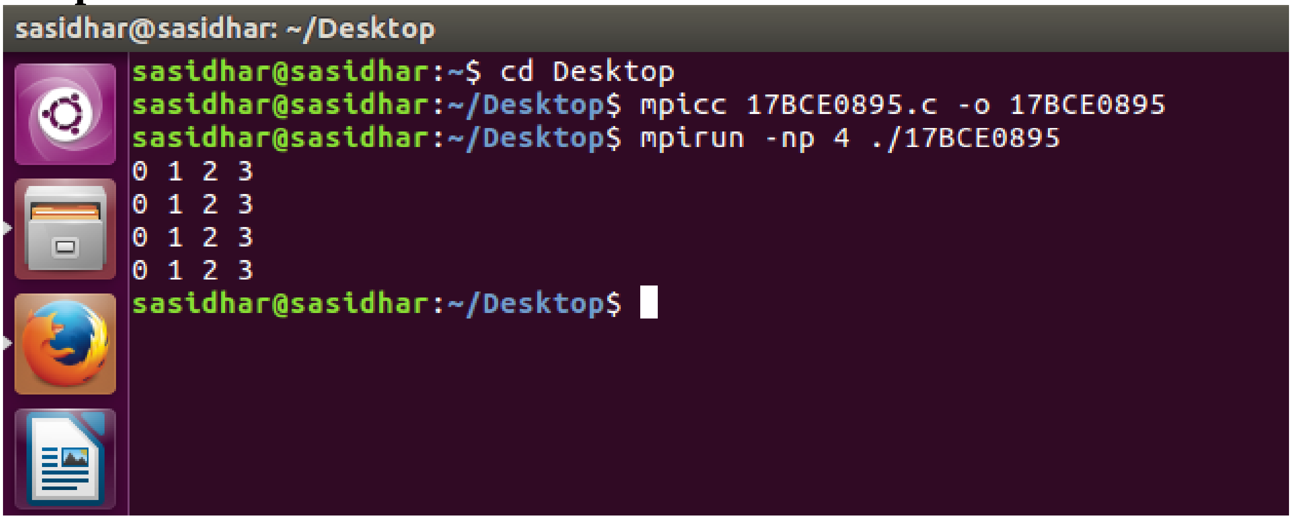
Lab Assignment – 5

Question-1: MPI Broadcast

Code:-

```
#include<stdio.h>
#include<stdlib.h>
#include<mpi.h>
#include<math.h>
int main(argc,argv)
int argc;
char *argv[];
{
int i,myid, numprocs;
int source,count;
int buffer[4];
MPI_Status status;
MPI_Request request;
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);
source=0;
count=4;
if(myid == source){
for(i=0;i<count;i++)
buffer[i]=i;
}
MPI_Bcast(buffer,count,MPI_INT,source,MPI_COMM_WORLD);
for(i=0;i<count;i++)
printf("%d ",buffer[i]);
printf("\n");
MPI_Finalize();
}
```

Output:



```
sasidhar@sasidhar: ~/Desktop
sasidhar@sasidhar:~$ cd Desktop
sasidhar@sasidhar:~/Desktop$ mpicc 17BCE0895.c -o 17BCE0895
sasidhar@sasidhar:~/Desktop$ mpirun -np 4 ./17BCE0895
0 1 2 3
0 1 2 3
0 1 2 3
0 1 2 3
sasidhar@sasidhar:~/Desktop$
```

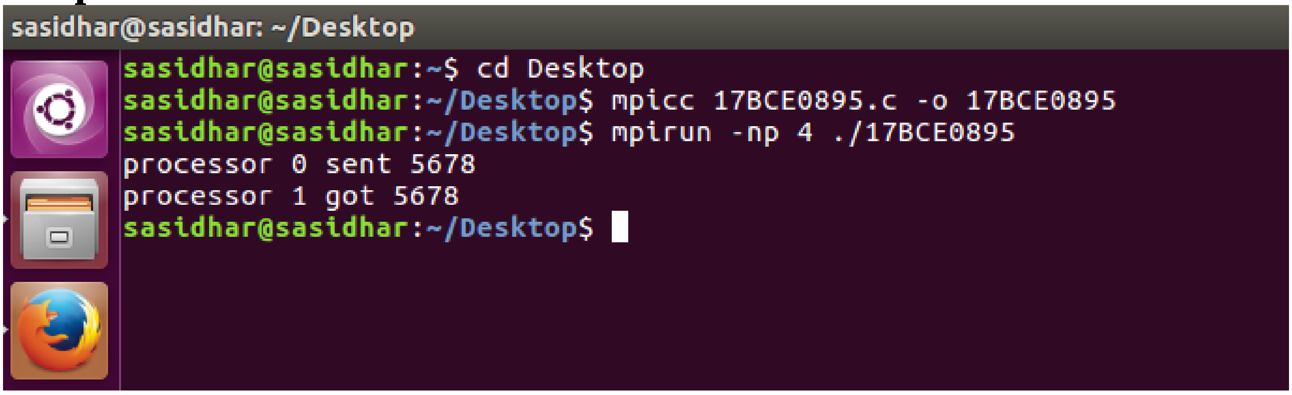
Question-2: MPI Send Receive

Code:-

```
#include<stdio.h>
#include<stdlib.h>
#include<mpi.h>
#include<math.h>
int main(argc,argv)
int argc;
char *argv[];
{
int myid, numprocs;
int tag,source,destination,count;
int buffer;
MPI_Status status;
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);
tag=1234;
source=0;
destination=1;
count=1;
if(myid == source){
buffer=5678;
MPI_Send(&buffer,count,MPI_INT,destination,tag,MPI_COMM_WORLD);
printf("processor %d sent %d\n",myid,buffer);
}
if(myid == destination){
```

```
MPI_Recv(&buffer,count,MPI_INT,source,tag,MPI_COMM_WORLD,&status);
printf("processor %d got %d\n",myid,buffer);
}
MPI_Finalize();
}
```

Output:



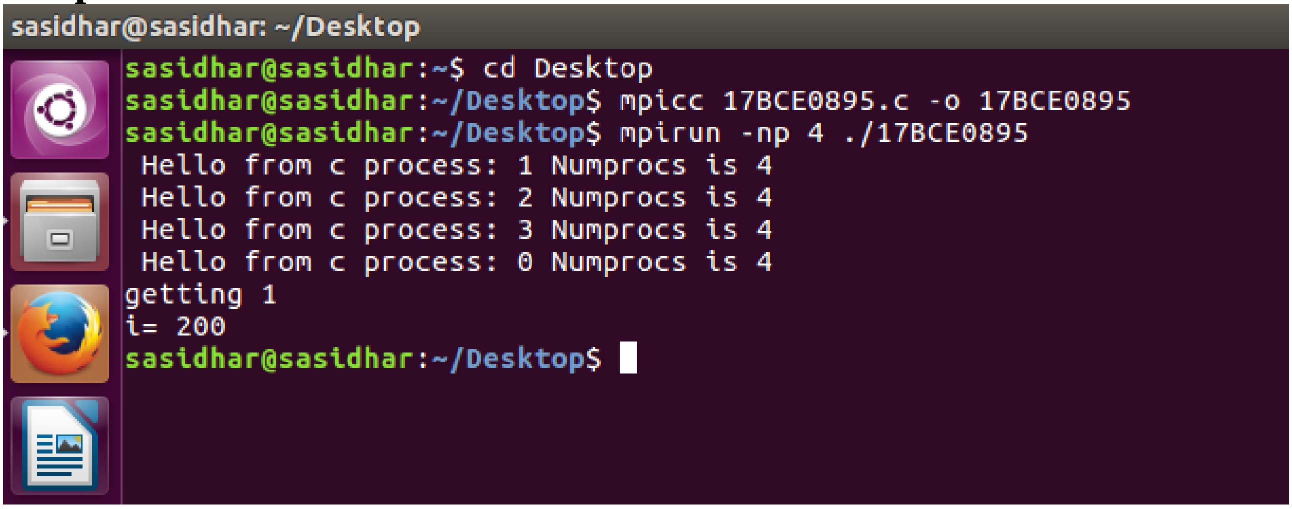
```
sasidhar@sasidhar:~$ cd Desktop
sasidhar@sasidhar:~/Desktop$ mpicc 17BCE0895.c -o 17BCE0895
sasidhar@sasidhar:~/Desktop$ mpirun -np 4 ./17BCE0895
processor 0 sent 5678
processor 1 got 5678
sasidhar@sasidhar:~/Desktop$
```

Question-3: MPI Calculate Size of Incoming Message

Code:-

```
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>
#include <math.h>
int main(int argc,char **argv)
{
int myid, numprocs;
MPI_Status status;
int mytag,ierr,icount,j,*i;
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);
printf(" Hello from c process: %d Numprocs is %d\n",myid,numprocs);
mytag=123;
if(myid == 0) {
j=200;
icount=1;
ierr=MPI_Send(&j,icount,MPI_INT,1,mytag,MPI_COMM_WORLD);
}}
if(myid == 1){
ierr=MPI_Probe(0,mytag,MPI_COMM_WORLD,&status);
ierr=MPI_Get_count(&status,MPI_INT,&icount);
i=(int*)malloc(icount*sizeof(int));
printf("getting %d\n",icount);
ierr = MPI_Recv(i,icount,MPI_INT,0,mytag,MPI_COMM_WORLD,&status);
printf("i= ");
for(j=0;j<icount;j++)
printf("%d ",i[j]);
printf("\n");
}
MPI_Finalize();
}
```

Output:



sasidhar@sasidhar: ~/Desktop

```
sasidhar@sasidhar:~$ cd Desktop
sasidhar@sasidhar:~/Desktop$ mpicc 17BCE0895.c -o 17BCE0895
sasidhar@sasidhar:~/Desktop$ mpirun -np 4 ./17BCE0895
 Hello from c process: 1 Numprocs is 4
 Hello from c process: 2 Numprocs is 4
 Hello from c process: 3 Numprocs is 4
 Hello from c process: 0 Numprocs is 4
getting 1
i= 200
sasidhar@sasidhar:~/Desktop$ 
```