**Program-9**

**Write MPI- C program which estimates the time it takes to send a vector of N double precision values through each process in a ring.**

#include <mpi.h>

#include <stdlib.h>

#include <stdio.h>

void ring\_io(int p,int id){

int dest,i,j,n,source,test,test\_num=10;

int n\_test[5]={100,1000,10000,100000,1000000};

int n\_test\_num=5;

MPI\_Status status; double tave,tmax,tmin,wtime,\*x;

if(id==0){

printf("\nTimings based on %d experiments\nN double precision values were sent\nin a ring transmission strating & ending at process 0\n and using a total of %d processes \n\n",test\_num,p);

printf(" N Tmin Tavg Tmax\n\n");

}

for(i=0;i<n\_test\_num;i++){

n=n\_test[i];

x=(double\*)malloc(n\*sizeof(double));

if(id==0){

dest=1;source=p-1;

tave=0.0;tmin=1.0E+30;tmax=0.0;

for(test=1;test<=test\_num;test++){

for(j=0;j<n;j++) x[j]=(double)(test+j);

wtime=MPI\_Wtime();

MPI\_Send(x,n,MPI\_DOUBLE,dest,0,MPI\_COMM\_WORLD);

MPI\_Recv(x,n,MPI\_DOUBLE,source,0,MPI\_COMM\_WORLD,&status);

wtime=MPI\_Wtime()-wtime;

tave+=wtime;

if(wtime<tmin) tmin=wtime;

if(tmax<wtime) tmax=wtime;

}

tave=tave/(double)(test\_num);

printf("%8d %14.6g %14.6g %14.6g\n",n,tmin,tave,tmax);

}else{

source=id-1;

dest=((id+1)%p);

for(test=1;test<=test\_num;test++){

MPI\_Recv(x,n,MPI\_DOUBLE,source,0,MPI\_COMM\_WORLD,&status);

MPI\_Send(x,n,MPI\_DOUBLE,dest,0,MPI\_COMM\_WORLD);

}

} free(x);

}return;

}

int main(int argc, char \*argv[]){

int error,id,p;

MPI\_Init(&argc,&argv);

MPI\_Comm\_size(MPI\_COMM\_WORLD,&p);

MPI\_Comm\_rank(MPI\_COMM\_WORLD,&id);

if(id==0){

printf("\nRING\_MPI:\nMPI version\nMeasure time req. to transmit data around\nring of processes \n\nNo. of processes is%d\n",p);

}

ring\_io(p,id);

MPI\_Finalize();

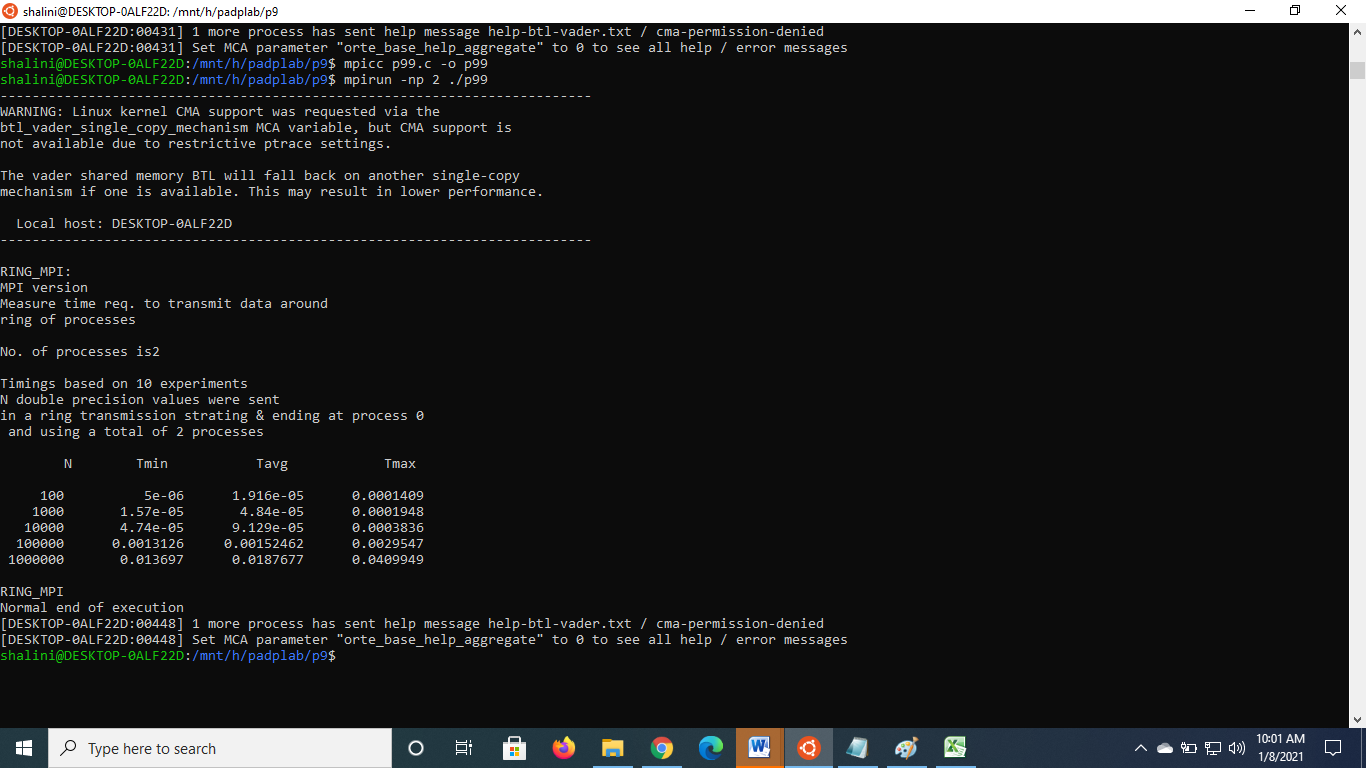
if(id==0){

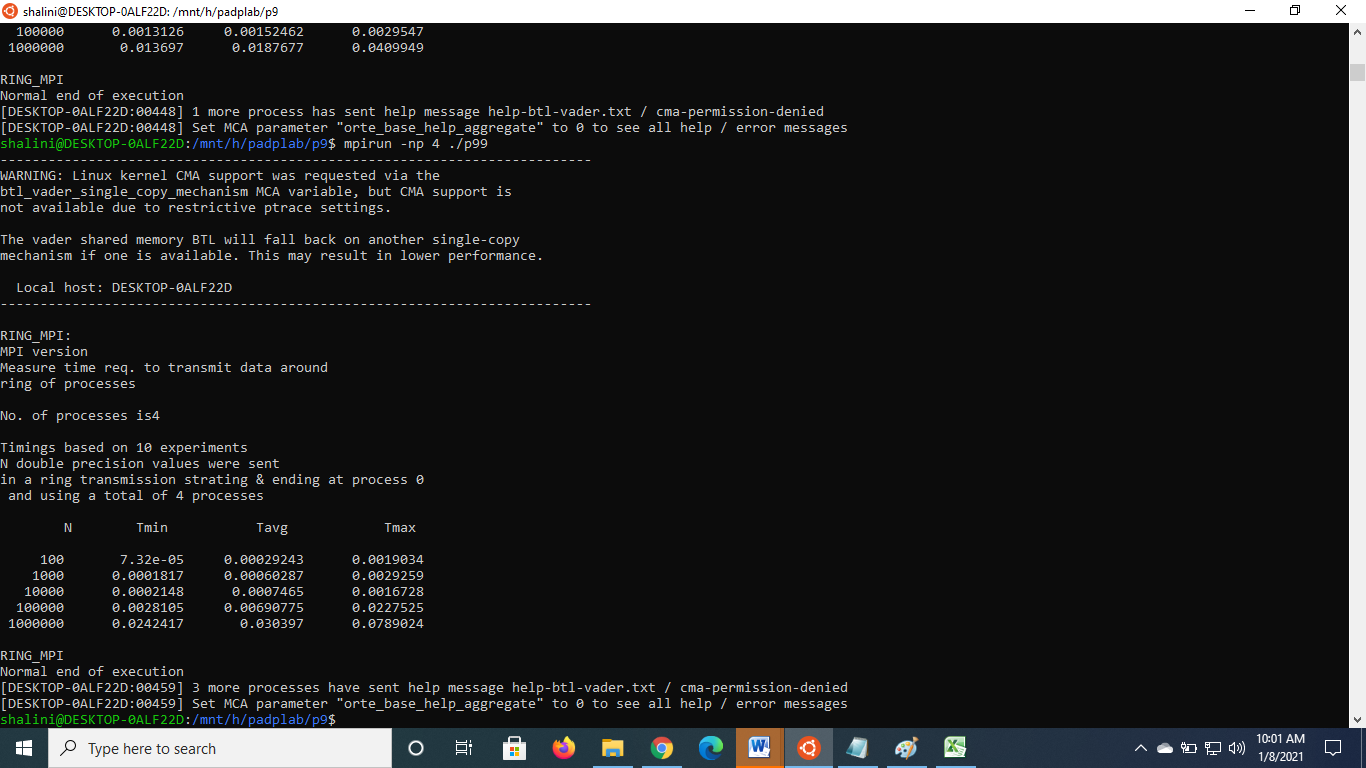
printf("\nRING\_MPI\nNormal end of execution\n");

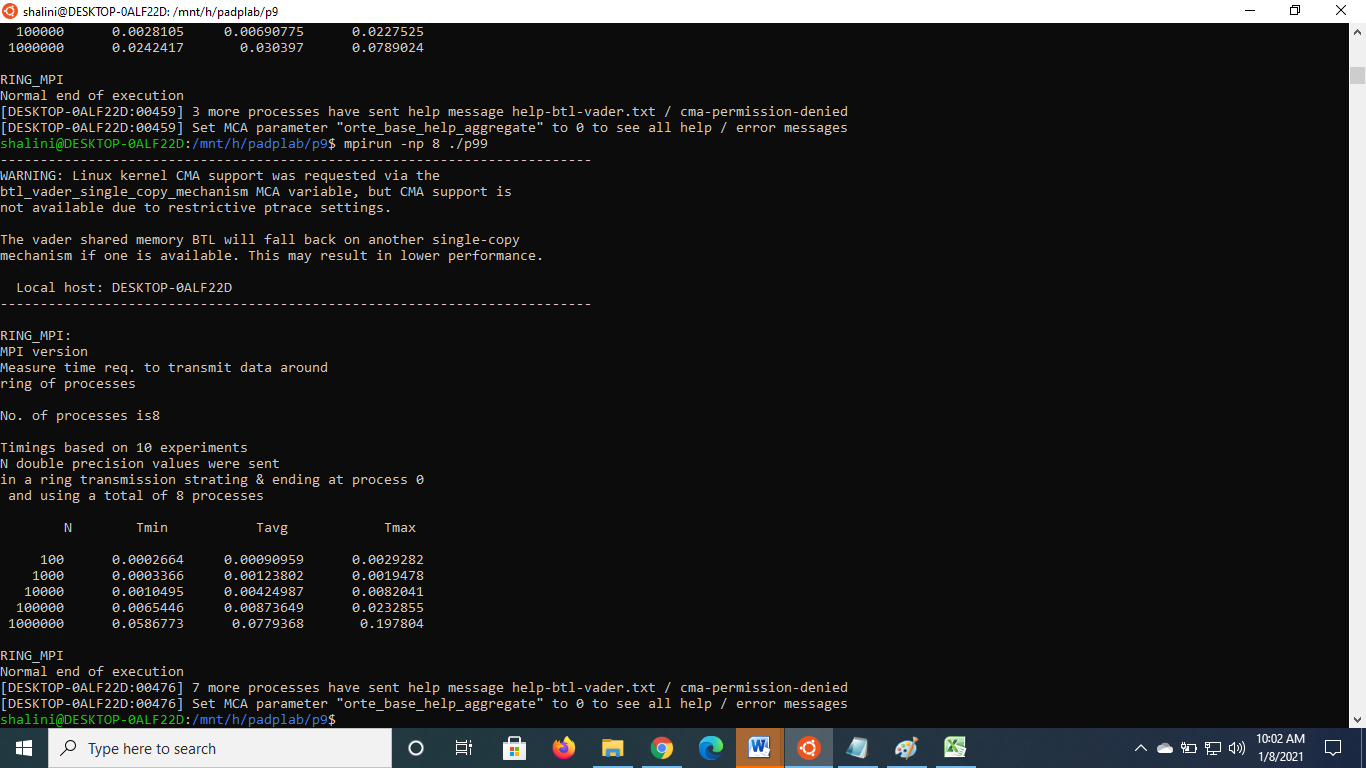
}

return 0;

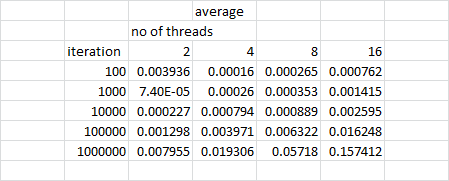
}



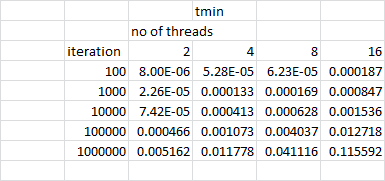




Tavg



Tmin



Tmax

