Name: Abinash Satapathy

Reg. no.: 16BCE0081

Slot: L45 + L46

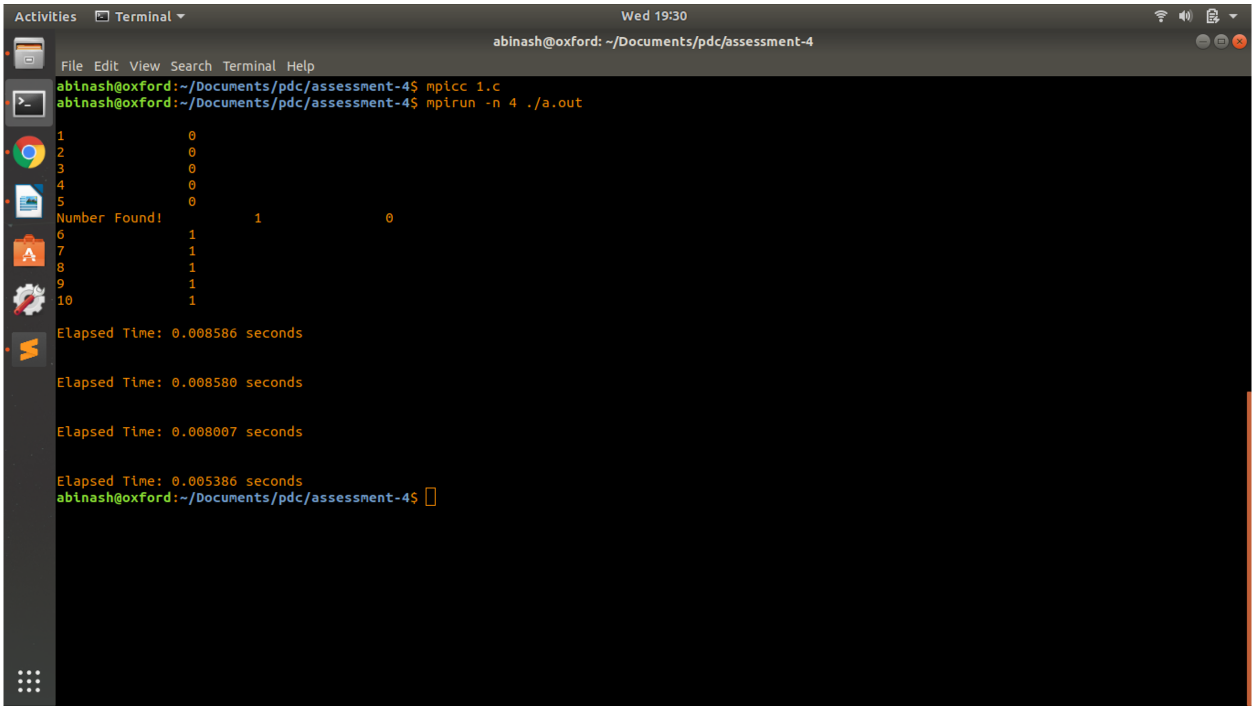
Subject: Parallel and Distributed Computing (CSE4001) Lab

Experiment – 4

---

1. **Write an MPI Program to perform binary search**

```
#include<stdio.h>
#include<time.h>
#include<mpi.h>
int main(int argc,char *argv[])
{
clock_t tic=clock();
int rank,size;
int a[10]={1,2,3,4,5,6,7,8,9,10},b[10];
int search=6,flag=0;
int i;
MPI_Init(&argc,&argv);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);
MPI_Comm_size(MPI_COMM_WORLD,&size);
MPI_Scatter(&a,5,MPI_INT,&b,5,MPI_INT,0,MPI_COMM_WORLD);
if (rank==0)
{
 for(i=0;i<5;i++)
{
if(b[i]==search)
{
 printf("\nNumber Found!\t\t%d\t\t%d",rank,i);
 flag=1;
}
printf("\n%d\t\t%d",b[i],rank);
}
}
if(rank==1)
 for(i=0;i<5;i++)
 {
if(b[i]==search)
{
 printf("\nNumber Found!\t\t%d\t\t%d",rank,i);
 flag=1;
}
printf("\n%d\t\t%d",b[i],rank);
}

MPI_Finalize();
clock_t toc=clock();
printf("\n\nElapsed Time: %f seconds\n",(double)(toc-tic)/CLOCKS_PER_SEC);
return(0);
}
```


Activities
Terminal
Wed 19:30
abinash@oxford: ~/Documents/pdc/assessment-4
File Edit View Search Terminal Help

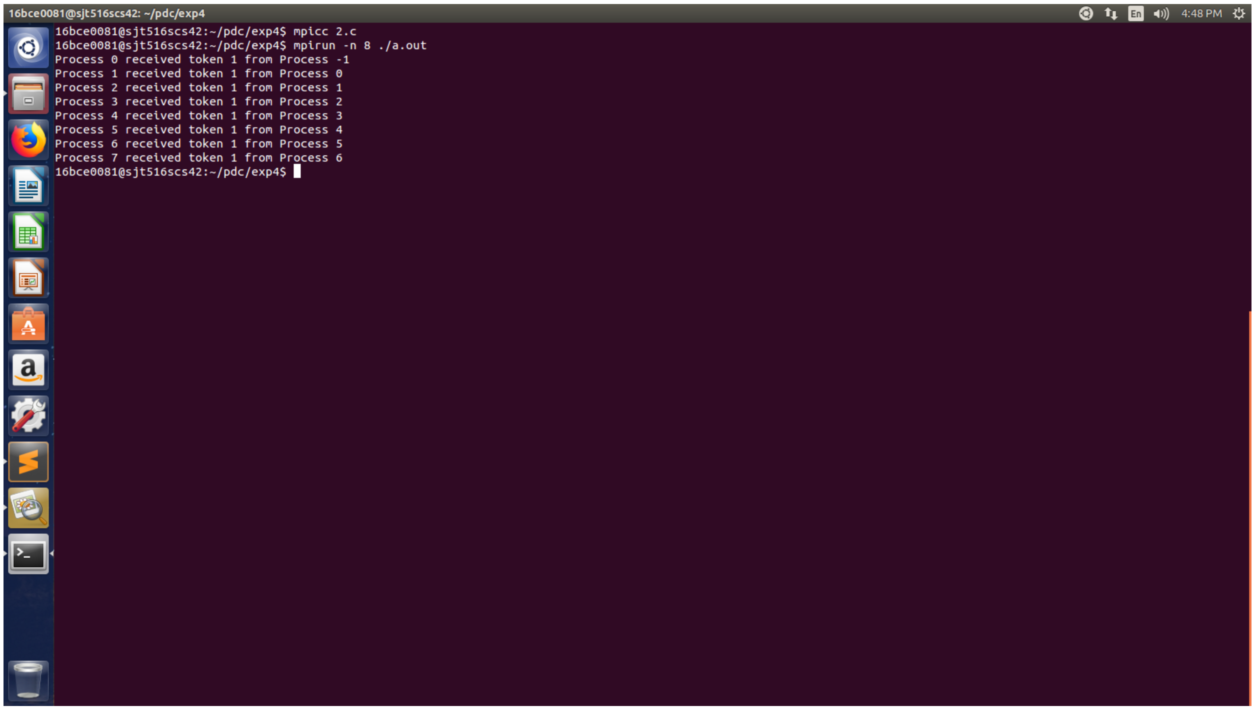
```
abinash@oxford:~/Documents/pdc/assessment-4$ mpicc 1.c
abinash@oxford:~/Documents/pdc/assessment-4$ mpirun -n 4 ./a.out

1               0
2               0
3               0
4               0
5               0
Number Found!           1               0
6               1
7               1
8               1
9               1
10              1

Elapsed Time: 0.008586 seconds


Elapsed Time: 0.008580 seconds


Elapsed Time: 0.008007 seconds


Elapsed Time: 0.005386 seconds
abinash@oxford:~/Documents/pdc/assessment-4$ 
```

---

2. Write an MPI program to perform ring communication

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

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

        MPI_Init(NULL, NULL);
        int world_rank;
```

```
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    int token;
    if(world_rank!=0){
        MPI_Recv(&token, 1, MPI_INT, world_rank-1, 0,
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        printf("Process %d received token %d from Process %d\n", world_rank,
token, world_rank-1);
    }

    else
        token = 1;

    MPI_Send(&token, 1, MPI_INT, (world_rank+1)%world_size, 0,
MPI_COMM_WORLD);

    if(world_rank==0){
        MPI_Recv(&token, 1, MPI_INT, world_rank-1, 0,
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
        printf("Process %d received token %d from Process %d\n", world_rank,
token, world_rank-1);
    }

    return 0;
}
```



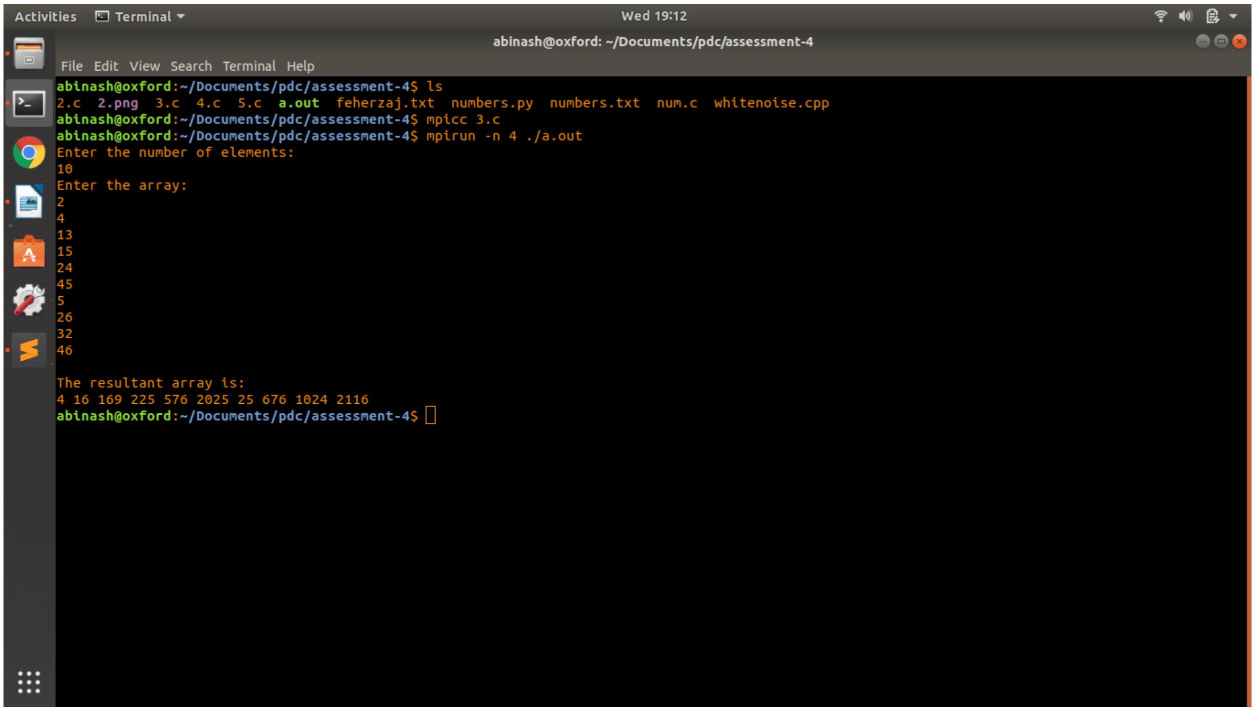
```
16bce0081@sjt516scs42:~/pdc/exp4$ mpicc 2.c
16bce0081@sjt516scs42:~/pdc/exp4$ mpirun -n 8 ./a.out
Process 0 received token 1 from Process -1
Process 1 received token 1 from Process 0
Process 2 received token 1 from Process 1
Process 3 received token 1 from Process 2
Process 4 received token 1 from Process 3
Process 5 received token 1 from Process 4
Process 6 received token 1 from Process 5
Process 7 received token 1 from Process 6
16bce0081@sjt516scs42:~/pdc/exp4$
```

---

3. **Write an MPI program to perform the squaring of numbers in array.**
   Input sequence: 2 4 8 16
   Output sequence: 4 16 64 256

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

int main(int argc, char **argv){
        int n, i;
        int size, rank;

        MPI_Status status;

        MPI_Init(&argc, &argv);
        MPI_Comm_size(MPI_COMM_WORLD, &size);
        MPI_Comm_rank(MPI_COMM_WORLD, &rank);

        MPI_Barrier(MPI_COMM_WORLD);

        if(rank==0){
```

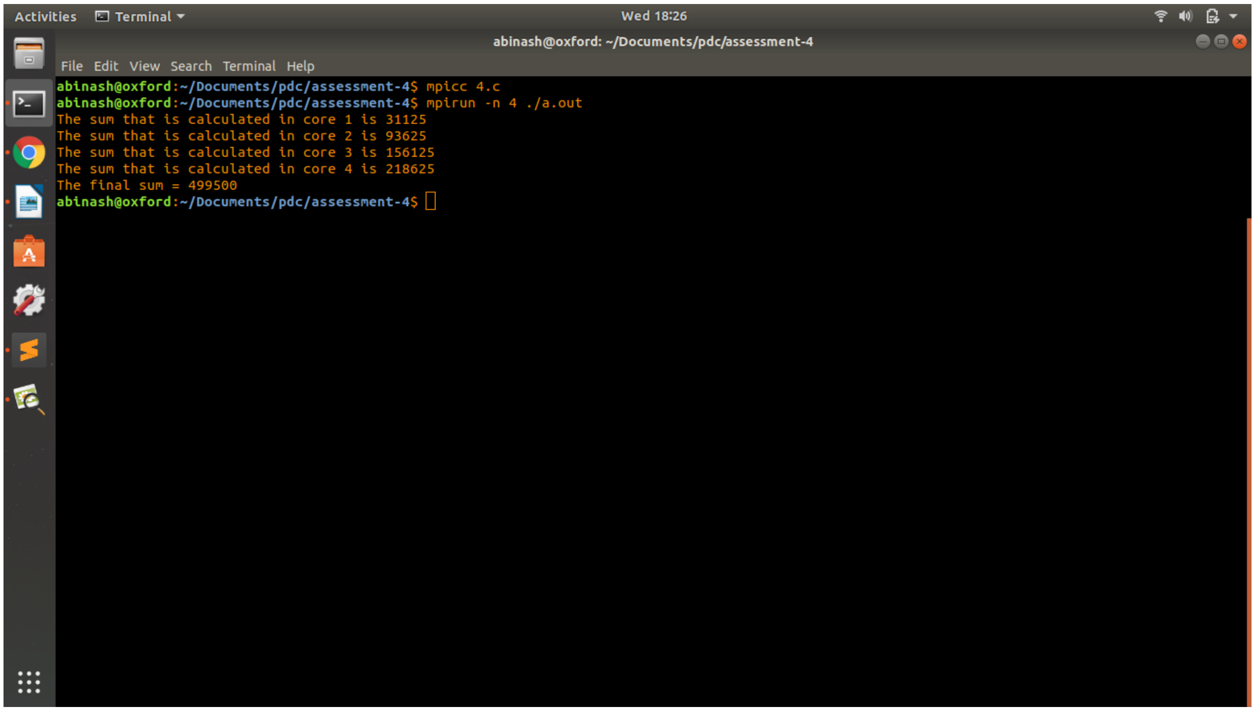
```
        int n, i;
      printf("Enter the number of elements:\n");
      scanf("%d", &n);
      int arr[n];
      printf("Enter the array:\n");
      for(i=0;i<n;i++)
          scanf("%d", &arr[i]);
        printf("\n");
        printf("The resultant array is:\n");
        for(i=0;i<n;i++)
            printf("%d ", (arr[i]*arr[i]));
        printf("\n");
    }

    MPI_Barrier(MPI_COMM_WORLD);

    MPI_Finalize();
    return 0;
}
```


Activities
Terminal
Wed 19:12
abinash@oxford: ~/Documents/pdc/assessment-4
File Edit View Search Terminal Help

```
abinash@oxford:~/Documents/pdc/assessment-4$ ls
2.c  2.png  3.c  4.c  5.c  a.out  feherzaj.txt  numbers.py  numbers.txt  num.c  whitenoise.cpp
abinash@oxford:~/Documents/pdc/assessment-4$ mpicc 3.c
abinash@oxford:~/Documents/pdc/assessment-4$ mpirun -n 4 ./a.out
Enter the number of elements:
10
Enter the array:
2
4
13
15
24
45
5
26
32
46

The resultant array is:
4 16 169 225 576 2025 25 676 1024 2116
abinash@oxford:~/Documents/pdc/assessment-4$ 
```

4. Write an MPI program to perform the sum of 1000 numbers using gather and scatter.

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

int main(int argc, char** argv){
        int i, j, k, p;
        int a[1000], b[2500], c[4], myrank, res, x, y;
        int interval, sum = 0;

        for(i=0;i<1000;i++)
                a[i] = i;

        MPI_Status status;
        MPI_Init(&argc, &argv);
        MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
        MPI_Comm_size(MPI_COMM_WORLD, &p);

        MPI_Scatter(a, 250, MPI_INT, b, 250, MPI_INT, 0, MPI_COMM_WORLD);

        res = 0;
        for(i=0;i<250;i++)
                res = res + b[i];

        MPI_Gather(&res, 1, MPI_INT, c, 1, MPI_INT, 0, MPI_COMM_WORLD);

        int final = 0;
        if(myrank==0){
                for(i=0;i<4;i++){
                        printf("The sum that is calculated in core %d is %d \n", (i+1), c[i]);
                        final = final + c[i];
                }
                printf("The final sum = %d\n", final);
        }

        MPI_Finalize();
        return 0;
}
```


Activities
Terminal
Wed 18:26
abinash@oxford: ~/Documents/pdc/assessment-4
File Edit View Search Terminal Help

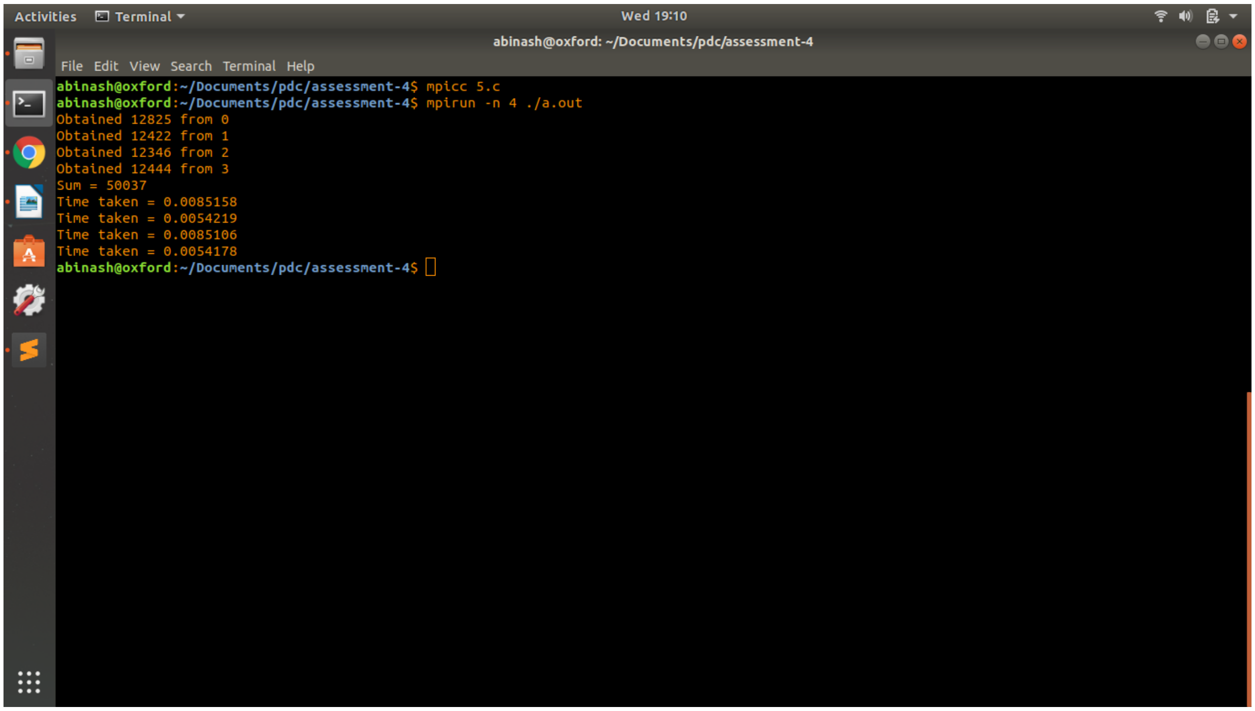
```
abinash@oxford:~/Documents/pdc/assessment-4$ mpicc 4.c
abinash@oxford:~/Documents/pdc/assessment-4$ mpirun -n 4 ./a.out
The sum that is calculated in core 1 is 31125
The sum that is calculated in core 2 is 93625
The sum that is calculated in core 3 is 156125
The sum that is calculated in core 4 is 218625
The final sum = 499500
abinash@oxford:~/Documents/pdc/assessment-4$
```

---

5. Write a MPI program to perform the sum of 1000 numbers using MPI broadcast and reduce function. Calculate the time using MPI wall time function.

```
#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>
#define maxsize 1000

int main(int argc, char **argv){
        double t1, t2;
        int myid, numprocs;
        int low, high, myresult=0, result;
        char fn[255];

        MPI_Init(&argc, &argv);
        MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
        MPI_Comm_rank(MPI_COMM_WORLD, &myid);
        t1 = MPI_Wtime();
        int i, x, data[1000];
        FILE *fptr;
        fptr = fopen("numbers.txt", "r");
```

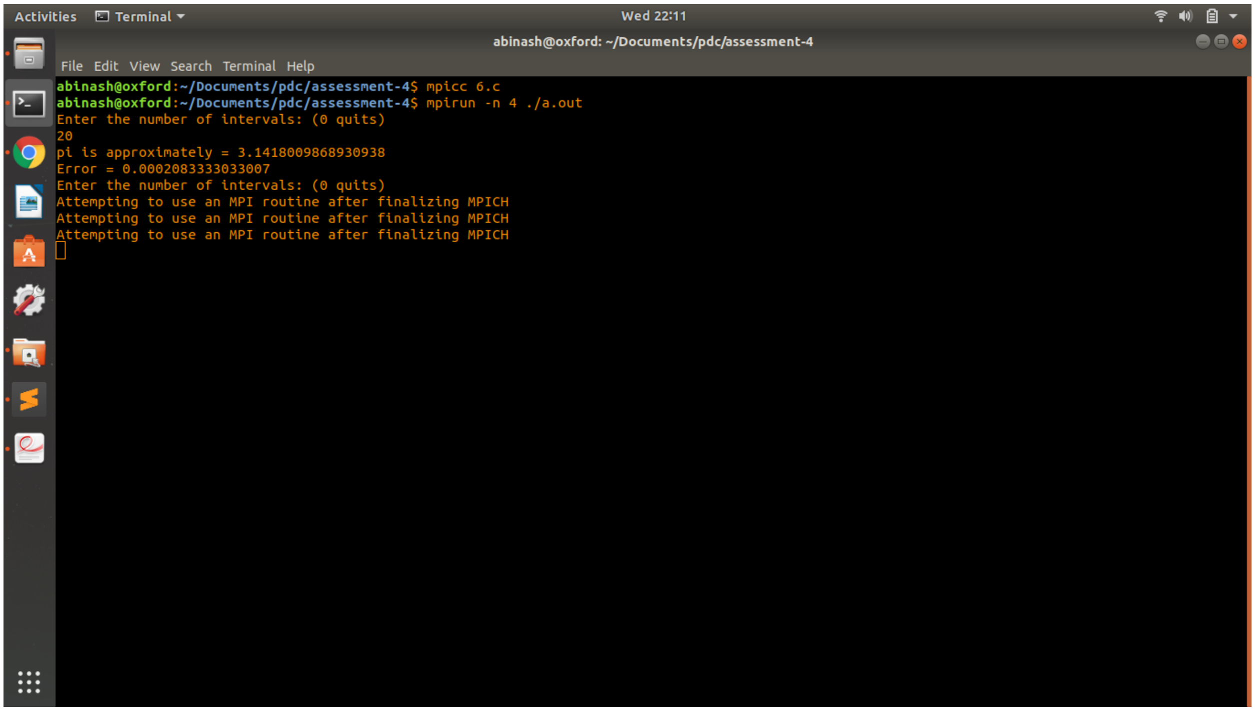
```
    if(fptr==NULL){
        printf("Error!\n");
        exit(1);
    }

    for(i=0;i<1000;i++)
        fscanf(fptr, "%d", &data[i]);
    fclose(fptr);

    MPI_Bcast(data, maxsize, MPI_INT, 0, MPI_COMM_WORLD);

    x = maxsize/numprocs;
    low = myid * x;
    high = low + x;
    for(i=low;i<high;i++)
        myresult = myresult + data[i];
    printf("Obtained %d from %d\n",myresult, myid);

    MPI_Reduce(&myresult, &result, 1, MPI_INT, MPI_SUM, 0,
MPI_COMM_WORLD);

    if(myid==0)
        printf("Sum = %d\n", result);

    t2 = MPI_Wtime();

    MPI_Finalize();
    printf("Time taken = %.7f\n", (t2-t1));
    return 0;
}
```


Activities
Terminal
Wed 19:10
abinash@oxford: ~/Documents/pdc/assessment-4
File Edit View Search Terminal Help

```
abinash@oxford:~/Documents/pdc/assessment-4$ mpicc 5.c
abinash@oxford:~/Documents/pdc/assessment-4$ mpirun -n 4 ./a.out
Obtained 12825 from 0
Obtained 12422 from 1
Obtained 12346 from 2
Obtained 12444 from 3
Sum = 50037
Time taken = 0.0085158
Time taken = 0.0054219
Time taken = 0.0085106
Time taken = 0.0054178
abinash@oxford:~/Documents/pdc/assessment-4$
```

6. **Write an MPI program to calculate the the value of pi using broadcast and reduce functions.**

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

int main(int argc, char *argv[]){
        int done = 0, n, myid, numprocs, i, rc;
        double PI25DT = 3.141592653589793238462643;
        double mypi, pi, h, sum, x, a;

        MPI_Init(&argc, &argv);
        MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
        MPI_Comm_rank(MPI_COMM_WORLD, &myid);

        while(!done){
                if(myid==0){
                        printf("Enter the number of intervals: (0 quits)\n");
                        scanf("%d", &n);
                }
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
        if(n==0)
            break;

        h = 1.0/(double)n;
        sum = 0.0;
        for(i=myid+1;i<=n;i=i+numprocs){
            x = h*((double)i - 0.5);
            sum = sum + (4.0/(1.0+x*x));
        }

        mypi = h * sum;

        MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
MPI_COMM_WORLD);

        if(myid==0){
            printf("pi is approximately = %.16f\n", pi);
            printf("Error = %.16f\n", fabs(pi-PI25DT));
        }

        MPI_Finalize();

    }
    return 0;
}
```


Activities
Terminal
Wed 22:11
abinash@oxford: ~/Documents/pdc/assessment-4
File Edit View Search Terminal Help

```
abinash@oxford:~/Documents/pdc/assessment-4$ mpicc 6.c
abinash@oxford:~/Documents/pdc/assessment-4$ mpirun -n 4 ./a.out
Enter the number of intervals: (0 quits)
20
pi is approximately = 3.1418009868930938
Error = 0.0002083333033007
Enter the number of intervals: (0 quits)
Attempting to use an MPI routine after finalizing MPICH
Attempting to use an MPI routine after finalizing MPICH
Attempting to use an MPI routine after finalizing MPICH
```