

Lab 3: Collective Communication

Name: Sirichoke

Yooyen

ID: 6088232

Sec: 3

Save your file to “lab3_60xxxxxx.pdf” and upload it to MyCourses website.

Lab: 1. Pi Calculation

Result from running with number of intervals (n) = 5 and number of processes (np) = 5:

```
[[u6088232@cluster lab3]$ mpirun -np 5 Pi_calculcation
[Enter the number of intervals: (0 quits) 5
pi is approximately 3.1449258640033282, Error is 0.0033332104135351
[u6088232@cluster lab3]$ █
```

Result from running with number of intervals (n) = 4 and number of processes (np) = 3:

```
[[u6088232@cluster lab3]$ mpirun -np 3 Pi_calculcation
[Enter the number of intervals: (0 quits) 4
pi is approximately 3.1468005183939427, Error is 0.0052078648041496
[u6088232@cluster lab3]$ █
```

Result from running with number of intervals (n) = 7 and number of processes (np) = 6:

```
[[u6088232@cluster lab3]$ mpirun -np 6 Pi_calculcation
[Enter the number of intervals: (0 quits) 7
pi is approximately 3.1432933175274682, Error is 0.0017006639376751
[u6088232@cluster lab3]$ █
```

Lab: 2. Modified integersum.c

Source Code:

```

#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[]) {
    int rank, size;

    MPI_Status status;
    int interval;
    int number, start, end, sum, GrandTotal; int proc;
    int LEFT,RIGHT;
    MPI_Init( &argc, &argv );
    MPI_Comm_rank( MPI_COMM_WORLD, &rank ); MPI_Comm_size( MPI_COMM_WORLD, &size );
    if (rank == 0) { GrandTotal = 0;
    printf("Enter the number of left: (0 quits) "); fflush(stdout);
    scanf("%d", &LEFT);
    printf("Enter the number of right: (0 quits) ");
    fflush(stdout); scanf("%d", &RIGHT);
    MPI_Bcast(&LEFT, 1, MPI_INT, 0, MPI_COMM_WORLD); MPI_Bcast(&RIGHT, 1, MPI_INT, 0,
    MPI_COMM_WORLD);
    MPI_Reduce(&sum, &GrandTotal, 1, MPI_INT, MPI_SUM, 0,MPI_COMM_WORLD); printf("Grand
    total = %d \n", GrandTotal);
    }
    else {
    MPI_Bcast(&LEFT, 1, MPI_INT, 0, MPI_COMM_WORLD); MPI_Bcast(&RIGHT, 1, MPI_INT, 0,
    MPI_COMM_WORLD);
    interval=(RIGHT-LEFT+1)/(size-1);
    start=(rank-1)*interval+LEFT; end=start+interval-1;
    if (rank == (size-1)) { /* for last block */
    end = RIGHT;
    }
    sum=0; /*Sum locally on each proc*/
    for (number=start; number<=end; number++)

```

```

sum = sum+number;

printf("Rank %d: start %d, end %d, local sum %d\n",rank,start,end,sum); MPI_Reduce(&sum,
&GrandTotal, 1, MPI_INT, MPI_SUM, 0,MPI_COMM_WORLD);

} MPI_Finalize();

}

```

Result:

```

mo_intergersum.c Pi_calculation Pi_calculation.c Pi_calculcation
[u6088232@cluster lab3]$ mpicc -o mo_intergersum mo_intergersum.c
[u6088232@cluster lab3]$ mpirun -np 4 mo_intergersum
Enter the number of left: (0 quits) 5
Enter the number of right: (0 quits) 6
Rank 3: start 5, end 6, local sum 11
Rank 1: start 5, end 4, local sum 0
Grand total = 11
Rank 2: start 5, end 4, local sum 0
[u6088232@cluster lab3]$ █

```