**PP LAB WEEK-6**

# DSE VI-A2 Divansh Prasad 210968140

1) Write a simple MPI program to find out pow(x,rank) for all processes where ‘x’ is the integer constant and ‘rank’ is the rank of the process.

#include <stdio.h>

#include <mpi.h>

#include <math.h>

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

int rank, size, x, result;

MPI\_Init(&argc, &argv);

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

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

if (rank == 0) {

printf("\nEnter the value of x: ");

scanf("%d", &x);

}

MPI\_Bcast(&x, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

result = pow(x, rank);

int \*results = NULL;

if (rank == 0) {

results = (int \*)malloc(size \* sizeof(int));

}

MPI\_Gather(&result, 1, MPI\_INT, results, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

if (rank == 0) {

printf("\nResults:\n");

for (int i = 0; i < size; i++) {

printf("Process %d: %d\n", i, results[i]);

}

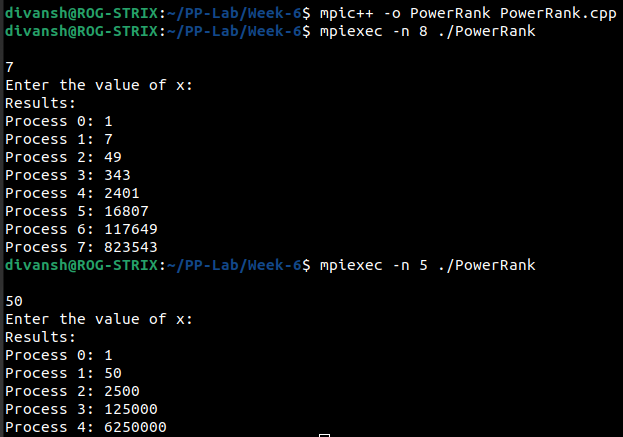
free(results);

}

MPI\_Finalize();

return 0;

}



2) Write a program in MPI where even ranked process prints “Hello” and odd ranked process prints “World”.

#include<stdio.h>

#include<mpi.h>

#include<stdlib.h>

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

{

int myid, numprocs=10, namelen;

char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];

MPI\_Init(&argc, &argv); // starts MPI

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &myid); // get current process id

MPI\_Comm\_size(MPI\_COMM\_WORLD, &numprocs); // get number of processeser

MPI\_Get\_processor\_name(processor\_name, &namelen);

printf("Rank %d: ", myid);

if (myid % 2 == 0) {

printf("Hello\n");

}

else {

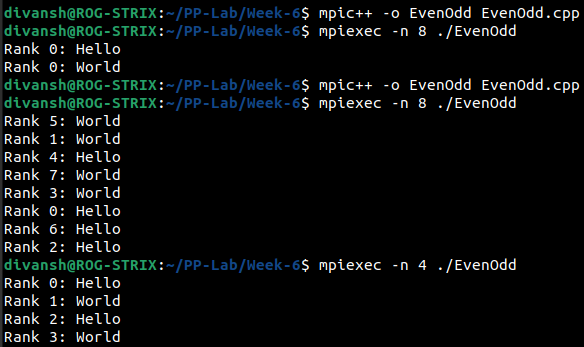
printf("World\n");

}

MPI\_Finalize();

return 0;

}



3) Write a program in MPi to simulate a simple calculator. Perform each operation using a different thread.

#include <stdio.h>

#include <mpi.h>

#include <stdlib.h>

#include <math.h>

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

int myid, numprocs, namelen, num1 = 69, num2 = 23;

char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];

MPI\_Init(&argc, &argv); // starts MPI

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &myid); // get current process id

MPI\_Comm\_size(MPI\_COMM\_WORLD, &numprocs); // get number of processes

MPI\_Get\_processor\_name(processor\_name, &namelen);

if (myid % 4 == 0)

printf("Sum: %d\n", num1 + num2);

if (myid % 4 == 1)

printf("Difference: %d\n", num1 - num2);

if (myid % 4 == 2)

printf("Product: %d\n", num1 \* num2);

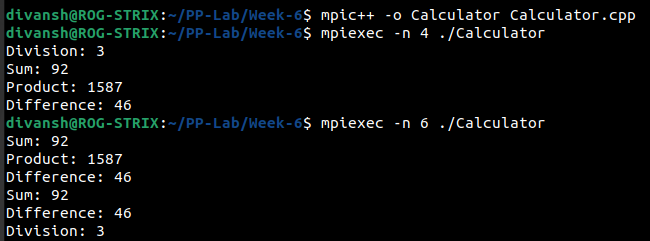
if (myid % 4 == 3)

printf("Division: %d\n", num1 / num2);

MPI\_Finalize();

return 0;

}



4) Write a program in MPI to toggle the characters of a given string indexed by the rank of the process.

#include <stdio.h>

#include <mpi.h>

#include <stdlib.h>

#include <ctype.h>

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

int myid, numprocs, namelen, len = 7;

char s[7] = "HeLlO";

char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];

MPI\_Init(&argc, &argv); // starts MPI

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &myid); // get current process id

MPI\_Comm\_size(MPI\_COMM\_WORLD, &numprocs); // get number of processes

MPI\_Get\_processor\_name(processor\_name, &namelen);

if (myid < len) {

if (islower(s[myid])) {

s[myid] = toupper(s[myid]);

} else if (isupper(s[myid])) {

s[myid] = tolower(s[myid]);

}

}

MPI\_Finalize();

printf("%s\n", s);

return 0;

}

