**ARYAMAN MISHRA**

**19BCE1027**

**SAMPLE PROGRAM**

#include "mpi.h"

#include <stdio.h>

#include <stdlib.h>

float \*create\_rand\_nums(int n)

{

float \*rnd=(float \*) malloc(sizeof(float)\*n);

int i;

for(i=0;i<n;i++)

{

rnd[i]=rand();

}

return rnd;

}

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

{

MPI\_Init(&argc,&argv);

int id;

int p,i;

int num\_elements\_per\_proc=1;

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

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

float \*rand\_nums = NULL;

if(id==0)

{

rand\_nums = create\_rand\_nums(p);

}

float \*sub\_rand\_nums = (float \*)malloc(sizeof(float) \* p);

MPI\_Scatter(rand\_nums, num\_elements\_per\_proc, MPI\_FLOAT, sub\_rand\_nums, 1, MPI\_FLOAT, 0, MPI\_COMM\_WORLD);

if(id>0)

{

printf("\nData recieved: ");

for( i=0;i<p;i++)

{

printf("%.2f ",sub\_rand\_nums[i]);

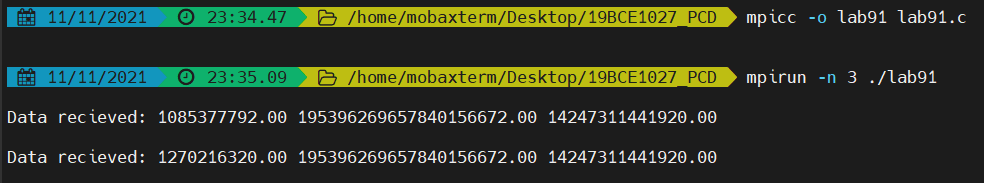
}

printf("\n");

}

MPI\_Finalize();

return 0;

}

**Create a random number and broadcast to each other by using Gather Function.**

#include "mpi.h"

#include <stdio.h>

#include <stdlib.h>

float \*create\_rand\_nums(int n)

{

float \*rnd=(float \*) malloc(sizeof(float)\*n);

int i;

for(i=0;i<n;i++)

{

rnd[i]=rand();

}

return rnd;

}

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

{

MPI\_Init(&argc,&argv);

int id;

int p,i;

int num\_elements\_per\_proc=1;

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

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

float \*rand\_nums = NULL;

if(id==0)

{

rand\_nums = create\_rand\_nums(p);

}

float \*sub\_rand\_nums = (float \*)malloc(sizeof(float) \* p);

MPI\_Gather(&rand\_nums, 1, MPI\_FLOAT, sub\_rand\_nums, 1, MPI\_FLOAT, 0, MPI\_COMM\_WORLD);

if(id>0)

{

printf("\nData recieved: ");

for( i=0;i<p;i++)

{

printf("%.2f ",sub\_rand\_nums[i]);

}

printf("\n");

}

MPI\_Finalize();

return **0;**

**}**

