**Лабораторная работа №3**

**Обобщенная передача данных**

**Цель:** изучить основные принципы обобщенной передачи данных в технологии MPI на примере использования в рамках языка С++.

**Лабораторные задания**

**Задание.** Модифицировать программу, написанную на Л.Р. №1 так чтобы она работала на основе обобщенной передачи сообщений. **Результаты работы сравнить (с результатами полученными в Л.Р. №2) и занести в отчет.**

#include <stdio.h>

#include "mpi.h"

#include <iostream>

#include <time.h>

#include <locale>

#include <vector>

using namespace std;

int generate\_int(int min\_v, int max\_v) {

return min\_v + std::rand() % (max\_v - min\_v);

}

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

{

int ProcNum, ProcRank;

int RecvRank = 0;

MPI\_Init(&argc, &argv);

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

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

MPI\_Status Status;

bool flag = true;

bool flag\_vec = true;

int counter = 0;

int rand\_send = 0;

int rand\_recv[8];

for (int i = 0; i < 8; i++)

{

rand\_recv[i] = 0;

}

srand(time(0));

while (flag) {

if (ProcRank != 0) {

rand\_send = generate\_int(-1, 15);

MPI\_Gather(&rand\_send, 1, MPI\_INT, rand\_recv, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

}

else {

MPI\_Gather(&rand\_send, 1, MPI\_INT, rand\_recv, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

for (int i = 0; i < 8; i++)

{

printf("%d-", rand\_recv[i]);

}

for (int i = 0; i < 8; i++)

{

if (rand\_recv[i] == -1) {

printf("\n THE END");

flag = false;

break;

}

else {

printf("\n %d", rand\_recv[i]);

}

}

}

MPI\_Bcast(&flag, 1, MPI\_C\_BOOL, 0, MPI\_COMM\_WORLD);

}

MPI\_Finalize();

return 0;

}