ЛР-3

[Подзаголовок документа]

Тюльников Михаил пин-32

[Год]

# Лабораторное задание

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include "mpi.h"

#include <iostream>

void DataInitialize(int\* array, int size);

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

{

int M = 4;

int ProcNum, ProcRank;

MPI\_Status Status;

MPI\_Init(&argc, &argv);

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

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

int\* msg = new int[ProcNum], \*recvMsg = new int[ProcNum];

DataInitialize(msg, ProcNum);

DataInitialize(recvMsg, ProcNum);

for (int j = 0; j < M; j++) {

MPI\_Scatter(msg, 1, MPI\_INT, recvMsg, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

if (ProcRank == 0)

{

printf("Send %d from 0 to all process\n", msg[ProcRank]);

}

else

{

printf("Receive %d in process %3d\n", recvMsg[0], ProcRank);

recvMsg[0] = recvMsg[0] \* ProcRank;

printf("Send %d from %d to process %3d\n", recvMsg[0], ProcRank, 0);

}

MPI\_Gather(recvMsg, 1, MPI\_INT, msg, 1, MPI\_INT, 0, MPI\_COMM\_WORLD);

if (ProcRank == 0) {

int result = 0;

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

result += msg[i];

}

printf("Result: %d\n", result);

}

}

MPI\_Finalize();

delete[] msg;

delete[] recvMsg;

return 0;

}

void DataInitialize(int\* array, int size) {

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

array[i] = 1;

}

}

