Лабораторная работа №3

Обобщенная передача данных

Максим Ромашкин

ПИН-34   Номер в списке: 18

Оглавление

[**Задание, код программы и пояснение:** 1](#_Toc95838769)

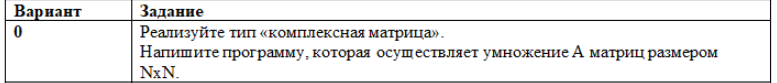
[**Листинг программы:** 1](#_Toc95838770)

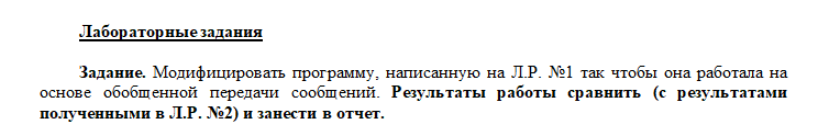
[**Результат программы:** 1](#_Toc95838771)

[**Выполнение примера** 1](#_Toc95838772)

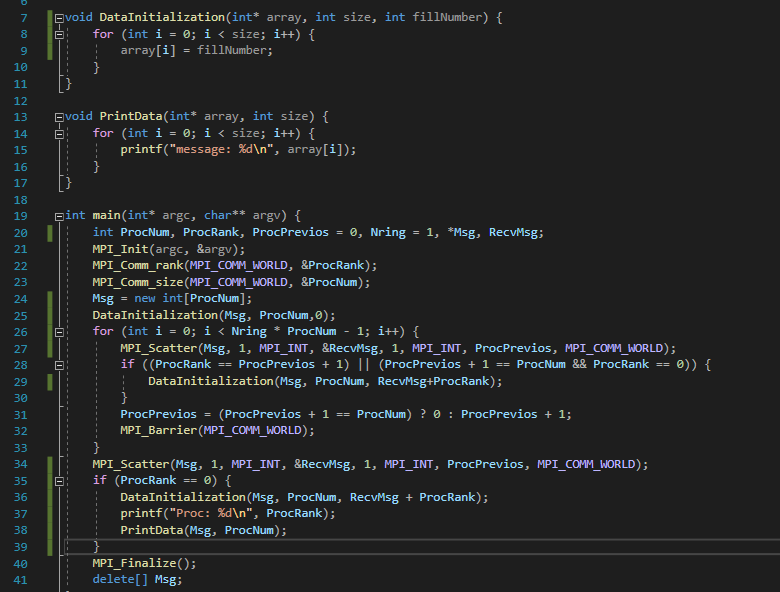
[**Ответы на контрольные вопросы** 1](#_Toc95838773)

# **Задание, код программы и пояснение:**





В каждом процессе инициализируется массив (размером ProcNum) и заполняется 0. Далее 0 процесс отправляет всем по одному 0 из этого массива. Далее следующий, то есть первый процесс, увеличивает каждый элемент своего массива на ProcRank и рассылает всем по элементу, то есть всем по 1. Далее второй процесс делает тоже самое, то есть рассылает всем по 3. И так далее. После цикла последний процесс, отправляет данные 0.



# **Листинг программы:**

#include <stdio.h>

#include <mpi.h>

#include <iostream>

using namespace std;

void DataInitialization(int\* array, int size, int fillNumber) {

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

array[i] = fillNumber;

}

}

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

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

printf("message: %d\n", array[i]);

}

}

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

int ProcNum, ProcRank, ProcPrevios = 0, Nring = 1, \*Msg, RecvMsg;

MPI\_Init(argc, &argv);

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

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

Msg = new int[ProcNum];

DataInitialization(Msg, ProcNum,0);

for (int i = 0; i < Nring \* ProcNum - 1; i++) {

MPI\_Scatter(Msg, 1, MPI\_INT, &RecvMsg, 1, MPI\_INT, ProcPrevios, MPI\_COMM\_WORLD);

if ((ProcRank == ProcPrevios + 1) || (ProcPrevios + 1 == ProcNum && ProcRank == 0)) {

DataInitialization(Msg, ProcNum, RecvMsg+ProcRank);

}

ProcPrevios = (ProcPrevios + 1 == ProcNum) ? 0 : ProcPrevios + 1;

MPI\_Barrier(MPI\_COMM\_WORLD);

}

MPI\_Scatter(Msg, 1, MPI\_INT, &RecvMsg, 1, MPI\_INT, ProcPrevios, MPI\_COMM\_WORLD);

if (ProcRank == 0) {

DataInitialization(Msg, ProcNum, RecvMsg + ProcRank);

printf("Proc: %d\n", ProcRank);

PrintData(Msg, ProcNum);

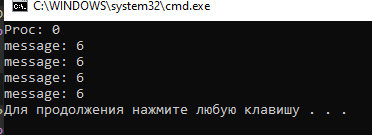
}

MPI\_Finalize();

delete[] Msg;

}

# **Результат программы:**



# **Выполнение примера**

В разработке не было полноценных примеров

# **Ответы на контрольные вопросы**

В разработке не было контрольных вопросов