Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

по дисциплине «Параллельное программирование»

Бондарев Матвей Владимирович

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) «Искусственный интеллект и разработка программных продуктов»

Руководитель работы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.И. Лаева

*подпись*«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Авторы работы

студент группы № 932204

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.В. Бондарев

*подпись*

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ г.

Томск – 2024

Оглавление

[ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1](#_Toc161763160)

[Задание 3](#_Toc161763161)

[Листинг программы 3](#_Toc161763162)

[Результат работы программы 4](#_Toc161763163)

## Задание

Вариант, полученный для выполнения: 12

**На процессe size-1 задать вещественный массив X размерности n.   
Для инициализированных процессов организовать передачу этого массива по кольцу: size-1 -> size-2 -> … -> 2 -> 1 -> 0 с помощью блокирующих двухточечных функций обмена. На 0-ом процессе распечатать массив X.**

## Листинг программы

Программа была написана на языке C++

#include <iostream>  
#include <mpi.h>  
  
using namespace std;  
  
int main(int argc, char\*\* argv) {  
 MPI\_Init(&argc, &argv);  
  
 MPI\_Comm comm;  
 MPI\_Status status;  
  
 int rank, size;  
 MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);  
 MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);  
  
 comm = MPI\_COMM\_WORLD;  
  
 const int n = 5;  
 float X[n];  
  
 // Задаем массив Х на (size-1)-ом процессе  
 if (rank == size - 1) {  
 for (int i = 0; i < n; i++) {  
 X[i] = (float)rand() / RAND\_MAX;;  
 }  
 cout << "Rank: "<< rank <<" Array X: ";  
 for (int i = 0; i < n; i++) {  
 cout << X[i] << " ";  
 }  
 }  
  
 // Производим обмен в цикле между процессами  
 for (int i = size - 1; i > 0; i--) {  
 if (rank == i) {  
 MPI\_Send(X, n, MPI\_FLOAT, (i - 1), 0, comm);}  
 if (rank == i - 1) {  
 MPI\_Recv(X, n, MPI\_FLOAT, i, 0, comm, &status);}  
 }

// Принимаем массив Х на 0-й процесс и выводим его в консоль  
 if (rank == 0) {  
 cout << "Array X on process 0: ";  
 for (int i = 0; i < n; i++) {  
 cout << X[i] << " ";  
 }  
 cout << endl;  
 }  
  
 MPI\_Finalize();  
 return 0;  
}

## Результат работы программы

При запуске программы на 4-х процессах получаем вывод в консоль:

Array X on process 0: 0.840188 0.394383 0.783099 0.79844 0.911647   
Rank: 3 Array X: 0.840188 0.394383 0.783099 0.79844 0.911647

Как мы видим, в результате выполнения программы, был создан массив Х на процессе rank = size - 1, передан по циклу к процессу rank = 0, после чего был осуществлен вывод в консоль.