

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Параллельный алгоритмы»**  
**Тема: Использование функций обмена данными «точка-точка» в**  
**библиотеке MPI.**

Студент гр. 9383

\_\_\_\_\_

Ноздрин В.Я.

Преподаватель

\_\_\_\_\_

Татаринов Ю.С.

Санкт-Петербург

2021

## Задание. Вариант 1. Испорченный телефон.

Процесс 0 генерирует строковое сообщение и передает его процессу со следующим номером. Процесс-получатель случайным образом меняет в сообщении один символ и передает его дальше. Последний процесс передает получившийся результат «ведущему».

### Выполнение работы.

Была написана программа согласно заданию. Для вычисления случайных чисел использовались методы srand() и rand(). srand() задает так называемый сид (seed), используемый для генерации чисел. rand() генерирует число от 0 до некоторого максимального числа.

```
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> jaa7"X4axaaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> jaa7"X4qxaaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> jaaU"a4]xaaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaU"aI]x5aaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaU"aI]xeaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaUaaI] ea9a
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaUa|IV ea9a
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaa IV e!9a
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaa aV o!Pa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> a.^aIIaaaaPa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 10 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> a.aaiZ8X,4aaa
```

Рисунок 1. Запуск программы на 10 процессах одиннадцать раз с сообщением “aaaaaaaaaaaa”.

```
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 1 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaaaaaaaaaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 2 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaaaaaaaaa&a
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 3 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaaaaaaSaa!
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 4 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> Gaaa:Daaaaaaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 5 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaZVataa6aaa
ice-jack@ice-pc ~/P/P/lab1 (master)> mpic++ main.cpp && mpirun -n 6 --oversubscribe a.out "aaaaaaaaaaaa"
aaaaaaaaaaaa ----> aaaaaaaUa-3aa
ice-jack@ice-pc ~/P/P/lab1 (master)> █
```

Рисунок 2. Запуск программы на 1, 2, 3, 4, 5, 6 процессах с сообщением “aaaaaaaaaaaa”.

Видно, что программа изменяет сообщение случайным образом. При этом реализация алгоритма такова, что количество измененных символов зависит от количества процессов. При этом так как “случайность” выбора символов и индексов для изменения задается сидом от текущего времени, при повторном запуске программы можно получить несколько одинаковых результатов. Потому что используемые случайные числа - псевдослучайные.

### **Выводы.**

Был получен опыт работы с функциями обмена данными точка-точка. Реализована программа, имитирующая работу так называемого испорченного телефона.

## ПРИЛОЖЕНИЕ

### Файл main.cpp

```
#include <iostream>

#include "mpi.h"

int main(int argc, char* argv[]){
    if (argc != 2) {
        std::cout << "argc = " << argc << std::endl;
        MPI_Status Status;
        MPI_Init(&argc, &argv);
        MPI_Finalize();
        return 5;
    }
    int size, rank, rand_index;
    char rand_char;
    size_t msg_size = strlen(argv[1]);
    char *sendbuf = new char[msg_size];
    char *recvbuf = new char[msg_size];
    MPI_Status Status;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    srand(time(NULL) + rank);
    if ( rank == 0 ) {
        sendbuf = argv[1];
        if (rank+1 < size) {
            std::cout << sendbuf << "\t---->\t";
            MPI_Send(sendbuf, msg_size, MPI_CHAR, rank+1, 0,
MPI_COMM_WORLD);
```

```

        MPI_Recv(recvbuf, msg_size, MPI_CHAR, size-1, MPI_ANY_TAG,
MPI_COMM_WORLD, &Status);
        std::cout << recvbuf << "\n";
    } else {
        std::cout << sendbuf << "\t---->\t" << sendbuf << "\n";
    }
} else {
    MPI_Recv(recvbuf, msg_size, MPI_CHAR, rank-1, MPI_ANY_TAG,
MPI_COMM_WORLD, &Status);
    for (int i = 0; i < msg_size; i++) {
        sendbuf[i] = recvbuf[i];
    }
    rand_index = rand()%msg_size;
    rand_char = 32 + rand()%94;
    sendbuf[rand_index] = rand_char;
    if (rank + 1 < size) {
        MPI_Send(sendbuf, msg_size, MPI_CHAR, rank+1, 0,
MPI_COMM_WORLD);
    } else {
        MPI_Send(sendbuf, msg_size, MPI_CHAR, 0, 0, MPI_COMM_WORLD);
    }
}
MPI_Finalize();
return 0;
}

```