

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет информационных систем и технологий
Кафедра программного обеспечения и управления в технических системах

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

по дисциплине Параллельное программирование
название (при наличии)
Использование функции групповой рассылки данных
название работы (при наличии)

ВЫПОЛНИЛ
студент гр. ПО-61 Булычев И. Д.
(группа) (ФИО)
ПРОВЕРИЛА
к. т. н., доцент Мезенцева Е. М.
(должность) (ФИО)

Самара
2019

1 Цель лабораторной работы

1.1 Цель работы

Необходимо реализовать программу, процессы которой обмениваются данными, введенными пользователем.

1.2 Используемое программное обеспечение

Для выполнения лабораторной работы мною было использовано следующее программное обеспечение:

- ОС Ubuntu 18.10
- IDE IntelliJ Idea 2018.3
- JDK 1.8
- MPJ Express 0.44

2 Результаты выполнения лабораторной работы

2.1 Содержимое файла input.txt

```
1
20
300
4000
0
9999999999
```

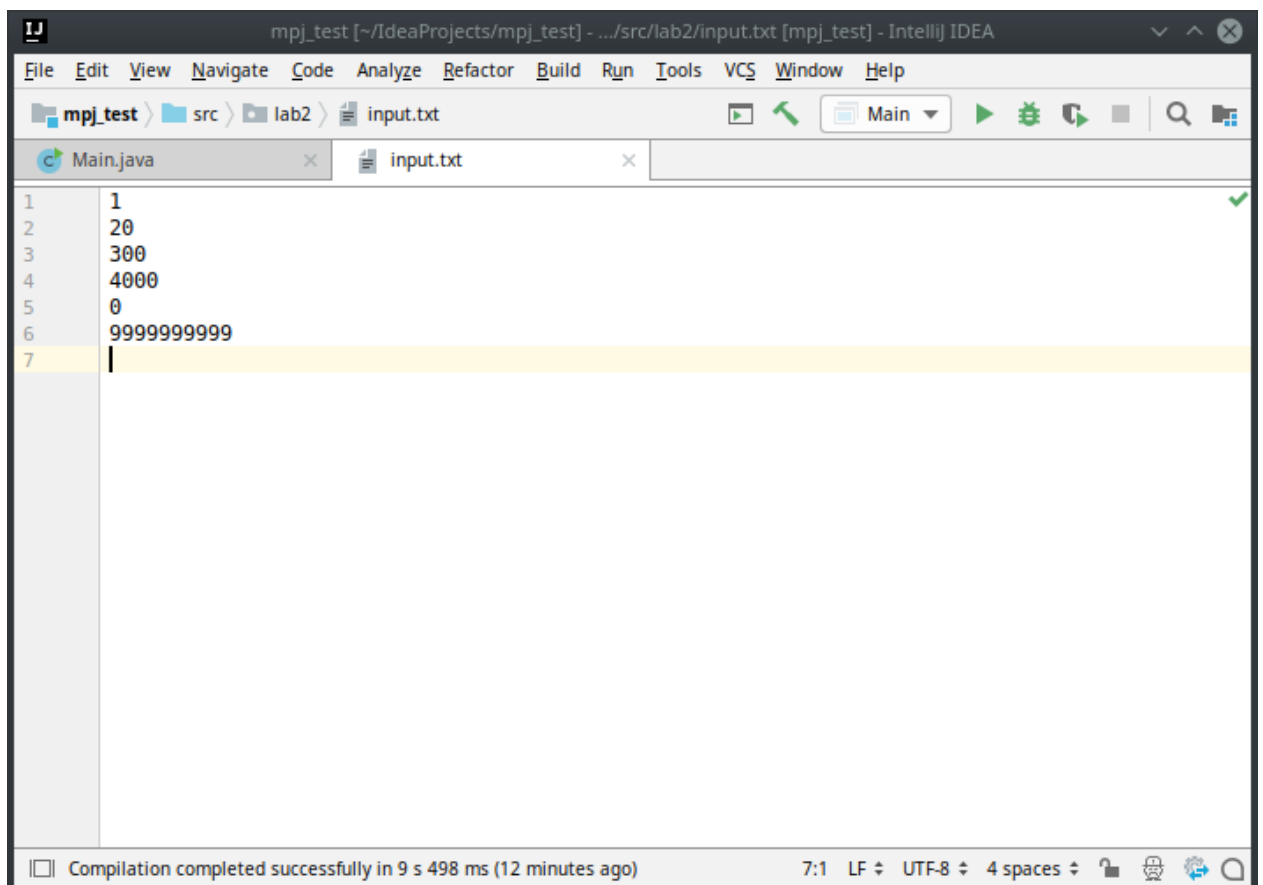


Рис. 2.1 — Файл input.txt

2.2 Листинг приложения

```
package lab2;

import mpi.MPI;

import java.io.File;
```

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws Exception {
        MPI.Init(args); // initialization
        Scanner in = new Scanner(new File("src/lab2/input.txt"));
        int[] num = new int[1]; //array creation
        // 0 - main process
        do {
            if (MPI.COMM_WORLD.Rank() == 0) {
                num[0] = in.nextInt(); // reading an int from file
            }
            MPI.COMM_WORLD.Bcast(num, 0, 1, MPI.INT, 0); // cast an array to all
                other processes
            System.out.println("hello from process " + MPI.COMM_WORLD.Rank() + ",
                num = " + num[0]);
        } while (num[0] != 0);
        System.out.println("input equals zero, shutting down process " +
            MPI.COMM_WORLD.Rank());
        MPI.Finalize();
    }
}

```

2.3 Результат выполнения

```

hello from process 0, num = 1
hello from process 1, num = 1
hello from process 2, num = 1
hello from process 3, num = 1
hello from process 2, num = 20
hello from process 3, num = 20
hello from process 0, num = 20
hello from process 1, num = 20
hello from process 3, num = 300
hello from process 2, num = 300
hello from process 0, num = 300
hello from process 1, num = 300
hello from process 3, num = 4000
hello from process 2, num = 4000
hello from process 0, num = 4000
hello from process 1, num = 4000
hello from process 3, num = 0

```

input equals zero, shutting down process 3
 hello from process 2, num = 0
 input equals zero, shutting down process 2
 hello from process 0, num = 0
 input equals zero, shutting down process 0
 hello from process 1, num = 0
 input equals zero, shutting down process 1

The screenshot displays the IntelliJ IDEA IDE with a project named 'mpj_test'. The 'Main.java' file is open, showing the following code:

```

1 package lab2;
2
3 import mpi.MPI;
4
5 import java.io.File;
6 import java.util.Scanner;
7
8 public class Main {
9     public static void main(String[] args) throws Exception {
10         MPI.Init(args); // initialization
11         Scanner in = new Scanner(new File( pathname: "src/lab2/input.txt"));
12         int[] num = new int[1]; //array creation
13         // 0 - main process
14         do {
15             if (MPI.COMM_WORLD.Rank() == 0) {
16                 num[0] = in.nextInt(); // reading an int from file
17             }
18             MPI.COMM_WORLD.Bcast(num, offset: 0, count: 1, MPI.INT, root: 0); // cast an array to all other processes
19             System.out.println("hello from process " + MPI.COMM_WORLD.Rank() + ", num = " + num[0]);
20             while (num[0] != 0);
21             System.out.println("input equals zero, shutting down process " + MPI.COMM_WORLD.Rank());
22             MPI.Finalize();
23         }
24     }
25 }
  
```

The 'Run' window at the bottom shows the output of the program:

```

/usr/lib/jvm/java-8-oracle/bin/java ...
MPJ Express (0.44) is started in the multicore configuration
hello from process 0, num = 1
hello from process 1, num = 1
hello from process 2, num = 1
hello from process 3, num = 1
hello from process 2, num = 20
hello from process 3, num = 20
hello from process 0, num = 20
hello from process 1, num = 20
hello from process 3, num = 300
hello from process 2, num = 300
hello from process 0, num = 300
hello from process 1, num = 300
hello from process 3, num = 4000
hello from process 2, num = 4000
hello from process 0, num = 4000
hello from process 1, num = 4000
hello from process 3, num = 0
input equals zero, shutting down process 3
hello from process 2, num = 0
input equals zero, shutting down process 2
hello from process 0, num = 0
input equals zero, shutting down process 0
hello from process 1, num = 0
input equals zero, shutting down process 1
Process finished with exit code 0
  
```

At the bottom of the IDE, a status bar indicates: 'Compilation completed successfully in 9 s 498 ms (9 minutes ago)'.

Рис. 2.2 — Листинг и результат выполнения

3 Выводы по результатам выполнения лабораторной работы

В данной работе была написана программа, которая обеспечивает обмен данными между нулевым процессом и всеми остальными. Это делается с помощью MPI функции MPI_Bcast. Нулевой процесс читает число из файла, после чего рассылает его всем процессам. После этого каждый процесс выводит на экран полученное сообщение с указанием своего номера. Процесс ввода продолжается до получения нуля.