



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного автономного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ ИУК «Информатика и управление»**

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»**

## **ЛАБОРАТОРНАЯ РАБОТА №1**

### **«Основные функции MPI»**

**ДИСЦИПЛИНА: «Параллельные процессы в информационных системах»**

Выполнил: студент гр. ИУК4-31М

\_\_\_\_\_  
(Подпись)

(Сафронов Н.С.)  
(Ф.И.О.)

Проверил:

\_\_\_\_\_  
(Подпись)

(Корнюшин Ю.П.)  
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

**Цель:** формирование практических навыков работы с программами на языке параллельного программирования MPI.

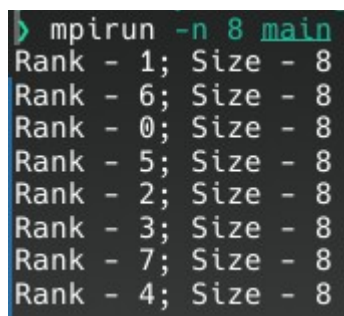
**Задачи:**

1. Получить представление о построении простых параллельных программ на языке параллельного программирования MPI.
2. Познакомиться с параллельными программами, настраиваемыми на размер вычислительной системы, как на параметр.
3. Получить практические навыки работы с функциями парных и коллективных взаимодействий между ветвями параллельной программы.

**Задание:**

Каждая ветвь параллельной программы выводит на экран свой идентификационный номер и размер заказанной параллельной системы, т.е. количество виртуальных компьютеров, в каждый из которых загружается ветвь п-программы.

**Результат выполнения работы:**



```
> mpirun -n 8 main
Rank - 1; Size - 8
Rank - 6; Size - 8
Rank - 0; Size - 8
Rank - 5; Size - 8
Rank - 2; Size - 8
Rank - 3; Size - 8
Rank - 7; Size - 8
Rank - 4; Size - 8
```

Рис. 1 – Результат выполнения задания

**Листинг:**

```
#include <mpi.h>
#include <stdio.h>

// Каждая ветвь параллельной программы выводит на экран свой идентификационный
// номер и размер заказанной параллельной системы, т.е. количество виртуальных
// компьютеров, в каждый из которых загружается ветвь п-программы.
int main(int argc, char** argv) {
    MPI_Init(&argc, &argv);

    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    printf("Rank - %d; Size - %d\n", rank, size);
```

```
    MPI_Finalize();  
    return 0;  
}
```

**Вывод:** в ходе выполнения лабораторной работы были сформированы практические навыки работы с программами на языке параллельного программирования MPI.