

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

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Параллельные Вычисления»
Тема: «Запуск параллельной программы»

Студент гр. 1307

Тростин М. Ю.

Преподаватель

Санкт-Петербург

2025

Цели и задачи

Освоить процесс запуска программы на C++ с применением библиотеки MPICH2. Научиться получать сведения о количестве запущенных процессов и номере отдельного процесса

- 1) Создать и запустить программу на 2-х процессах с применением функций
 - int MPI_Init(int* argc, char*** argv)
 - int MPI_Finalize(void).
- 2) Создать и запустить программу на 3-х процессах с применением функций:
 - int MPI_Init(int* argc, char*** argv);
 - int MPI_Finalize(void);
 - int MPI_Comm_size(MPI_Comm comm, int* size)
 - int MPI_Comm_rank(MPI_Comm comm, int* rank)
- 3) Программа должна выводить на экран номер процесса и какой-либо идентификатор процесса. Создать и запустить программу на n-х процессах печати таблицы умножения.

Задание 1

Создать и запустить программу на 2-х процессах с применением функций

- int MPI_Init(int* argc, char*** argv)
- int MPI_Finalize(void).

Между функциями MPI_Init и MPI_Finalize стоит вывод текста.

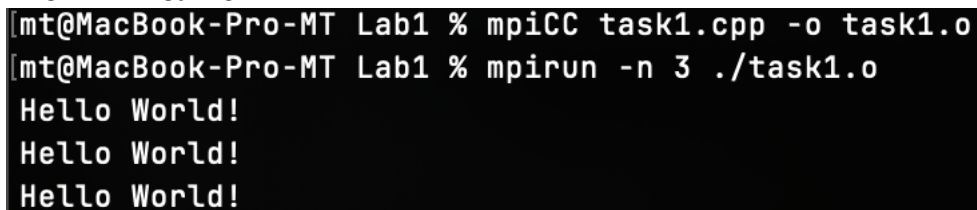
Количество выводов будет соответствовать количеству запущенных процессов

Исходный код task1.cpp

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

int main(int argc, char ** argv) {
    MPI_Init(&argc, &argv);
    printf("Hello World!\n");
    MPI_Finalize();
    return 0;
}
```

Скриншот выполнения



```
[mt@MacBook-Pro-MT Lab1 % mpiCC task1.cpp -o task1.o
[mt@MacBook-Pro-MT Lab1 % mpirun -n 3 ./task1.o
Hello World!
Hello World!
Hello World!
```

Задание 2

Создать и запустить программу на 3-х процессах с применением функций:

- int MPI_Init(int* argc, char*** argv);
- int MPI_Finalize(void);
- int MPI_Comm_size(MPI_Comm comm, int* size)
- int MPI_Comm_rank(MPI_Comm comm, int* rank)

Добавим к предыдущей программе получение и вывод номера процесса и кол-во запускаемых процессов

Исходный код task2.cpp

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

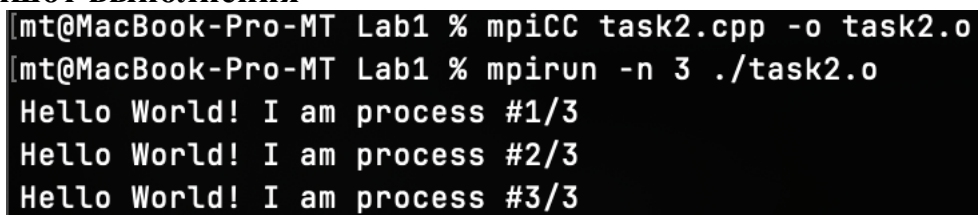
int main(int argc, char** argv) {
    MPI_Init(&argc, &argv);

    int rank, size;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    printf("Hello World! I am process #%d/%d\n", rank + 1, size);

    MPI_Finalize();
    return 0;
}
```

Скриншот выполнения



```
[mt@MacBook-Pro-MT Lab1 % mpiCC task2.cpp -o task2.o
[mt@MacBook-Pro-MT Lab1 % mpirun -n 3 ./task2.o
Hello World! I am process #1/3
Hello World! I am process #2/3
Hello World! I am process #3/3
```

Задание 3

Программа должна выводить на экран номер процесса и какой-либо идентификатор процесса. Создать и запустить программу на n-х процессах печати таблицы умножения.

Изменим предыдущую программу. Каждый запускаемый процесс будет определять числа для которых нужно вывести результаты умножения на числа с 1 по 9

Исходный код task3.cpp

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

int main(int argc, char ** argv) {
    MPI_Init(&argc, &argv);

    int rank, size;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    for(int i = 0; i <= (int)(10 / size); i++) {
        int number = i * size + rank + 1;
        if (number >= 10) break;
        for (int j = 1; j < 10; j++) printf("%d * %d = %d\n",
number, j, number * j);
        printf("\n");
    }

    MPI_Finalize();
    return 0;
}
```

Скриншоты выполнения

```
[mt@MacBook-Pro-MT Lab1 % mpirun -n 3 ./task3.o
```

```
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
```

```
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
```

```
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
```

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
```

```
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
```

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54

9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81

Выводы

В рамках выполнения данной работы был освоен процесс запуска программы на C++ с применением библиотеки MPICH2. А так же разобраны методы получения сведений о количестве запущенных процессов и номере отдельного процесса