

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Параллельный алгоритмы»**  
**Тема: Виртуальные топологии.**

Студент гр. 9383

\_\_\_\_\_

Ноздрин В.Я.

Преподаватель

\_\_\_\_\_

Татаринов Ю.С.

Санкт-Петербург

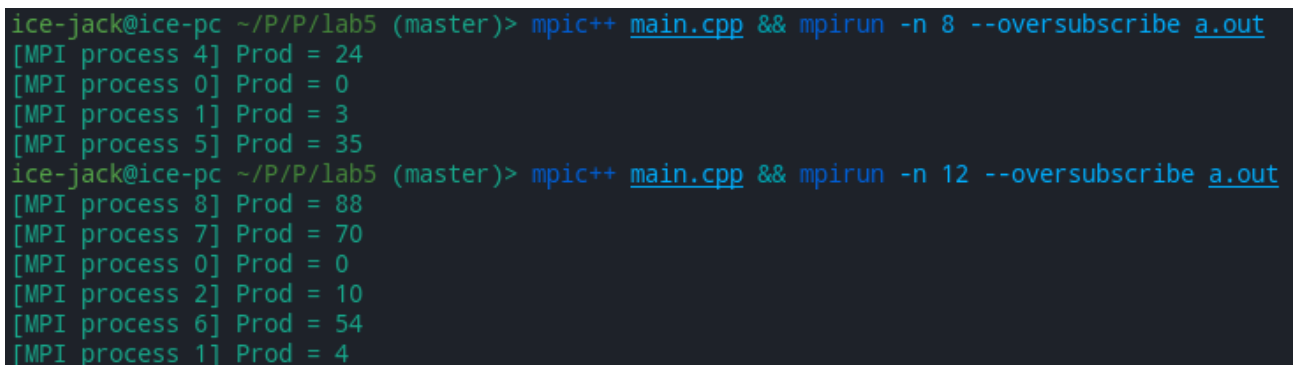
2021

### Задание. Вариант 7.

Количество процессов  $K$  равно 8 или 12, в каждом процессе дано вещественное число. Определить для всех процессов декартову топологию в виде трехмерной решетки размера  $2 \times 2 \times K/4$  (порядок нумерации процессов оставить прежним). Интерпретируя эту решетку как две матрицы размера  $2 \times K/4$  (в одну матрицу входят процессы с одинаковой первой координатой), расщепить каждую матрицу процессов на  $K/4$  одномерных столбцов. Используя одну коллективную операцию редукции, для каждого столбца процессов найти произведение исходных чисел и вывести найденные произведения в главных процессах каждого столбца.

### Выполнение работы.

Была написана программа, которая с помощью метода `MPI_Cart_create` создает декартову топологию, затем используется метод `MPI_Cart_sub` для расщепления топологии сначала на две матрицы процессов размером  $2 \times K/4$ , а затем, для разбиения этих матриц на  $K/2$  одномерных столбцов. Итого это будет  $2 \times K/4$  столбцов. Далее с помощью коллективной операции `MPI_Reduce` вычисляется произведение с помощью `MPI_PROD`.



```
ice-jack@ice-pc ~/P/P/lab5 (master)> mpic++ main.cpp && mpirun -n 8 --oversubscribe a.out
[MPI process 4] Prod = 24
[MPI process 0] Prod = 0
[MPI process 1] Prod = 3
[MPI process 5] Prod = 35
ice-jack@ice-pc ~/P/P/lab5 (master)> mpic++ main.cpp && mpirun -n 12 --oversubscribe a.out
[MPI process 8] Prod = 88
[MPI process 7] Prod = 70
[MPI process 0] Prod = 0
[MPI process 2] Prod = 10
[MPI process 6] Prod = 54
[MPI process 1] Prod = 4
```

Рисунок 1. Запуск программы на 8 и на 12 процессах.

В данном случае числа в процессах - есть их номер. Потому нулевой процесс возвращает значение 0, а остальные процессы возвращают некоторые произведения чисел. В случае с 8 процессами произведения следующие:  $0 \times 2$ ,  $1 \times 3$ ,  $4 \times 6$ ,  $5 \times 7$ . В случае с 12 процессами произведения следующие:  $0 \times 3$ ,  $1 \times 4$ ,  $2 \times 5$ ,  $6 \times 9$ ,  $7 \times 10$ ,  $8 \times 11$ .

### **Выводы.**

Изучены методы MPI позволяющие работать с декартовыми топологиями.  
Написана программа, согласно условию.

## ПРИЛОЖЕНИЕ

### Файл main.cpp

```
#include <iostream>

#include "mpi.h"

int main(int argc, char* argv[]){
    MPI_Init(&argc, &argv);
    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    if (size != 8 && size != 12) {
        std::cout << "This application is meant to be run with "
                    << "at 8 or 12 MPI processes.\n";
        MPI_Abort(MPI_COMM_WORLD, EXIT_FAILURE);
    }
    int number = rank;
    int dimension_number = 3,
        dims[dimension_number] = {2, 2, size/4},
        periods[dimension_number] = {false, false, false},
        reorder = false;
    MPI_Comm cartesian_communicator;
    MPI_Cart_create(
        MPI_COMM_WORLD,
        dimension_number,
        dims,
        periods,
        reorder,
        &cartesian_communicator
    );
```

```

int remain_dims[dimension_number] = {false, true, true};
MPI_Comm subgrid_communicator;
MPI_Cart_sub(
    cartesian_communicator,
    remain_dims,
    &subgrid_communicator
);
int subgrid_rank;
MPI_Comm_rank(subgrid_communicator, &subgrid_rank);
int reduction_result = 0;
int root_rank = 0;
MPI_Reduce(
    &number,
    &reduction_result,
    1,
    MPI_INT,
    MPI_SUM,
    root_rank,
    subgrid_communicator
);
if (subgrid_rank == root_rank)
    std::cout << "[MPI process " << rank << "] Sum = "
        << reduction_result << "\n";
MPI_Finalize();
return EXIT_SUCCESS;
}

```