МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования  
 «Кемеровский государственный университет»**

**Институт фундаментальных наук**

**ДОМАШНЯЯ РАБОТА №13**

**ПО ДИСЦИПЛИНЕ**

**“Технологии параллельных вычислений”**

студента 3 курса

**Сулима Роман Иванович**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена:

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_201\_г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_

Кемерово 2021

СОДЕРЖАНИЕ

[1. Постановка задачи 2](#_Toc90677518)

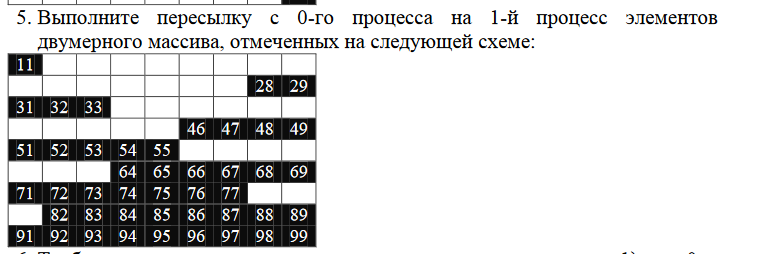
[3. Реализация 2](#_Toc90677519)

# 1. Постановка задачи

Посмотрите в прилагаемом материале еще раз примеры использования конструкторов размещения данных в ОП. Выполните упражнения, приведенные в конце параграфа - 1, 2, 5, 7 задачи. Оформляете в одном отчете - номер задачи, текст условия, текст программы с поясняющими комментариями в программе, скрин компиляции и скрин запуска. Удачи!

# 3. Реализация

Задание 5.



Программный код:

#include <mpi.h>

#include <stdio.h>

int main(int argc, char \*argv[])

{

    int rank;

    int size;

    int s = 0, n = 9;

    MPI\_Status stat;

    MPI\_Init(&argc, &argv);

    MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

    MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

    int count = 0, z = -3, x = 0;

    while (z < (n + (n-1))) {

        z += 4;

        count += 1;

    }

    z = -3;

    int\* bl=new int[count];

    int\* ds=new int[count];

    for (int i = 0; i < count; i++) {

        z += 4;

        bl[i] = z;

        ds[i] = x;

        x += n + n - 2;

*//printf("%d - %d\n", bl[i], ds[i]);*

    }

    if (bl[count - 1] + ds[count - 1] > n\*n)

        bl[count - 1] = n;

*// bl[count - 1] = z + 4;*

*// ds[count - 1] = n;*

*//int bl[5] = {1, 5, 9, 13, 17};*

*//int ds[5] = {0, 16, 32, 48, 64};*

    MPI\_Datatype mt;

    MPI\_Type\_indexed(count, bl, ds, MPI\_INT, &mt);

    MPI\_Type\_commit(&mt);

    int \*\*b = new int \*[n];

    b[0] = new int[n \* n];

    for (int i = 1; i < n; i++)

        b[i] = b[i - 1] + n;

    for (int i = 0; i < n; i++)

        for (int j = 0; j < n; j++)

            b[i][j] = 0;

    if (rank == 0)

    {

        for (int i = 0; i < n; i++)

            for (int j = 0; j < n; j++)

                b[i][j] = 10 \* (i + 1) + j + 1;

        MPI\_Send(\*b, 1, mt, 1, 777, MPI\_COMM\_WORLD);

    }

    if (rank == 1)

    {

        MPI\_Recv(\*b, 1, mt, 0, 777, MPI\_COMM\_WORLD, &stat);

    }

    for (int i = 0; i < 100000000 \* rank; i++)

        s += 1;

    printf("rank = %d\n", rank);

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

            printf("%3d ", b[i][j]);

        printf("\n");

    }

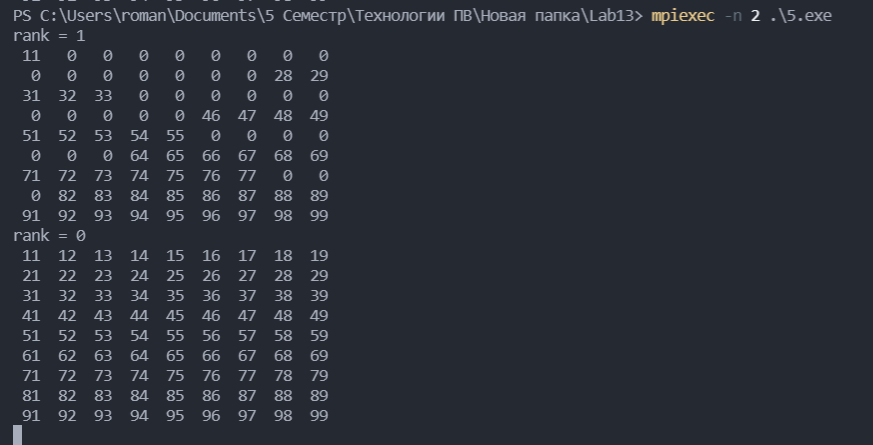
    MPI\_Finalize();

    return 0;

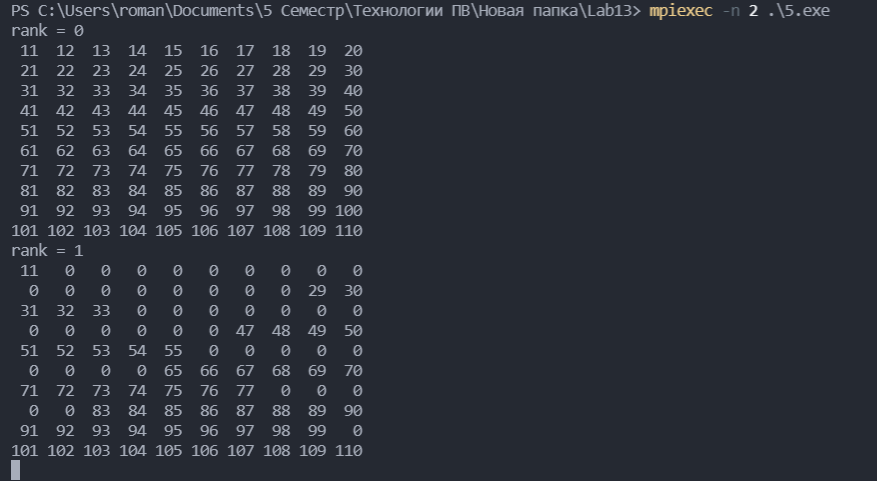
}

Выполнение:

При n=9



При n=10



Задание 7.

Требуется написать следующую параллельную программу: 1) на 0-м  
процессе задается квадратная матрица NxN по формуле:  
a(i,j)=10\*(i+1)+(j+1) 2) на 1-м процессе создается такая же матрица и  
заполняется нулями. 3) требуется передать с 0-го на 1-й процесс четные  
столбцы одной посылкой.

Программный код:

#include <stdio.h>

#include "mpi.h"

int main(int argc, char\* argv[]) {

    int rank, size, n = 10;

    MPI\_Init(&argc, &argv);

    MPI\_Status stat;

    MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

    MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

    MPI\_Datatype mt, mt1;

    MPI\_Type\_vector(n, 1, n, MPI\_INT, &mt);

    MPI\_Type\_commit(&mt);

    MPI\_Type\_create\_hvector(n/2, 1, 8, mt, &mt1);

    MPI\_Type\_commit(&mt1);

    int\*\* a = new int\* [n];

    a[0] = new int[n\*n];

    for (int i = 1; i < n; i++)

        a[i] = a[i - 1] + n;

    for (int i = 0; i < n; i++)

        for (int j = 0; j < n; j++)

            a[i][j] = 0;

    if (rank == 0){

        for (int i = 0; i < n; i++)

            for (int j = 0; j < n; j++)

                a[i][j] = 10 \* (i + 1) + (j + 1);

        MPI\_Send(&a[0][1], 1, mt1, 1, 111, MPI\_COMM\_WORLD);

    }

    if (rank == 1){

        MPI\_Recv(&a[0][1], 1, mt1, 0, 111, MPI\_COMM\_WORLD, &stat);

        for (int i = 0; i < n; i++){

            for (int j = 0; j < n; j++)

                printf("%3d ", a[i][j]);

            printf("\n");

        }

    }

    MPI\_Finalize();

    return 0;

}

Выполнение:

