МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

**Институт фундаментальных наук**

**ДОМАШНЯЯ РАБОТА №3**

**ПО ДИСЦИПЛИНЕ**

**“Технологии параллельных вычислений”**

**Задача № 1**

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](#_Toc335561691)

[2. Описание используемых функций 2](#_Toc335561692)

[3. Описание программы 2](#_Toc335561693)

[4. Реализация 2](#_Toc335561694)

[Заключение 3](#_Toc335561695)

[Литература 3](#_Toc335561696)

# 1. Постановка задачи

Напишите программу, которая реализует следующий алгоритм: на 0-м процессе задается одномерный массив (a[i]=i, i=0..3\*size), который по одному элементу рассылается всем ненулевым процессам по кругу. Например, Вы запускаете программу на 3-х процессах, 1-й процесс в итоге должен получить 1, 4, 7, 2-й процесс - 2, 5, 8. Не забудьте сделать контрольный вывод пересланных данных.

# 2. Описание используемых функций

MPI\_Send() – Передача сообщения другим потокам.

MPI\_Recv() – Принимает сообщение из другого потока.

# 3. Описание программы

Программа создает массив на 0-ом процессе и отправляет по 3 элемента на каждый поток, с шагом равным количеству потоков. Остальные потоки создают массивы длиной 3 и записывают полученные значения в массив и выводят на экран.

# 4. Реализация

Программный код:

#include <stdio.h>

#include "mpi.h"

#include "stdlib.h"

int main(int argc, char \*argv[]){

    int rank;

    int size;

    MPI\_Status stat;

    MPI\_Init(&argc, &argv);

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

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

    if (rank == 0){

        int\* a = NULL;

        a = (int\*)malloc(sizeof(int)\*(size\*3));

        for(int i = 0; i<3\*size; i++)

        {

            a[i] = i;

        }

        for(int i = 0; i < 3\*size; i += size)

        {

            for(int j = 1; j<size; ++j)

            {

                MPI\_Send(&a[i+j], 1, MPI\_INT, j, 777, MPI\_COMM\_WORLD);

            }

        }

    } else {

        int\* b = NULL;

        b = (int\*)malloc(sizeof(int)\*3);

        for(int i = 0; i < 3; i++)

            MPI\_Recv(&b[i], 1, MPI\_INT, 0, 777, MPI\_COMM\_WORLD, &stat);

        printf( "rank = %d, b: ",rank );

        for(int i = 0; i<3; i++)

            printf( "%d ", b[i] );

        printf( "\n " );

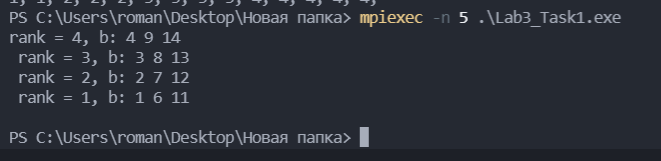
    }

    MPI\_Finalize();

    return 0;

}

Пример реализации:



# Заключение

Изучен способ передачи переменных из массива с определенным шагом в другие потоки.

# Литература