**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 1**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

СОДЕРЖАНИЕ

[1. Постановка задачи 2](#_Toc335561691)

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

[3. Описание программы 2](#_Toc335561693)

[4. Реализация 2](#_Toc335561694)

[Заключение 2](#_Toc335561695)

[Литература 2](#_Toc335561696)

# 1. Постановка задачи

Скачайте, установите и настройте библиотеку mpich на домашнем компьютере. В файле с описанием выполнения задания приведите скрины, демонстрирующие шаги подключения библиотеки, компиляции и запуска тестовой MPI-программы. Укажите характеристики компьютера, операционную систему, используемое программное обеспечение.

# 2. Описание используемых функций

# 3. Описание программы

Тестовая программа от преподавателя.

#include <iostream>

#include <mpi.h>

#include <stdint.h>

using namespace std;

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

{

    int my\_rank;

    int size;

    MPI\_Init(&argc, &argv);

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

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

    printf("Hello world! I'm rank (processor number) %d of %d processor \n",my\_rank, size);

    MPI\_Finalize();

    return 0;

}

# 4. Реализация

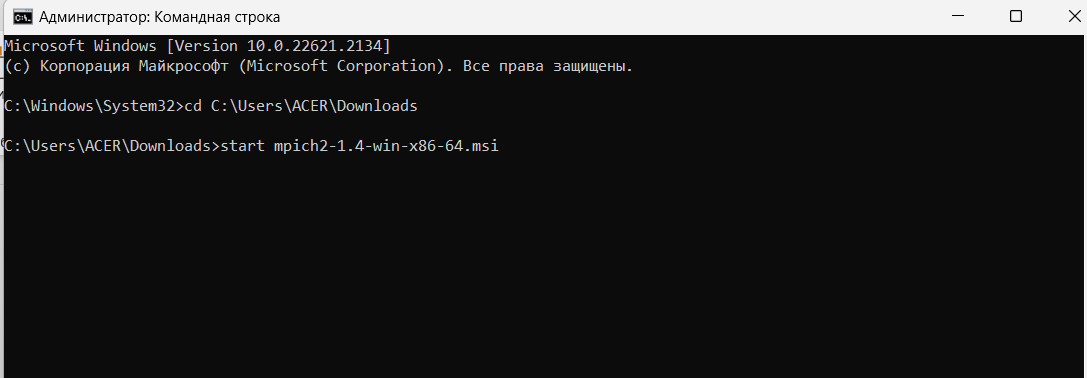
Подключение MPI к VS Code:

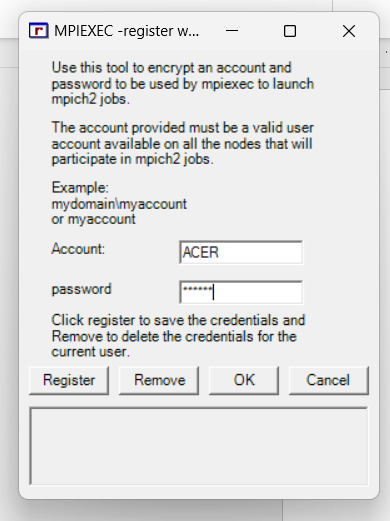
1. Устанавливаю на пк mpich2-1.4-win-x86-64.msi с правами администратора.
2. Регистрирую пользователя в файле wmpiregister.exe
3. Прописываю путь до папки bin в переменных средах
4. Создаю новый проект
5. В файле c\_cpp\_properties.json прописываю путь до MPICH2\bin
6. Создаю файл tasks.json, в котором прописываю пути до MPICH2\lib и MPICH2\include\

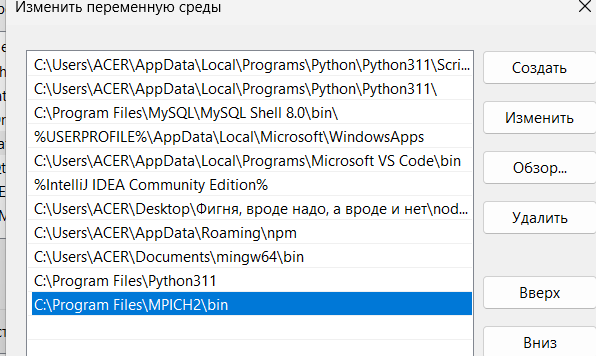
Скриншоты и характеристики представлены после списка литературы

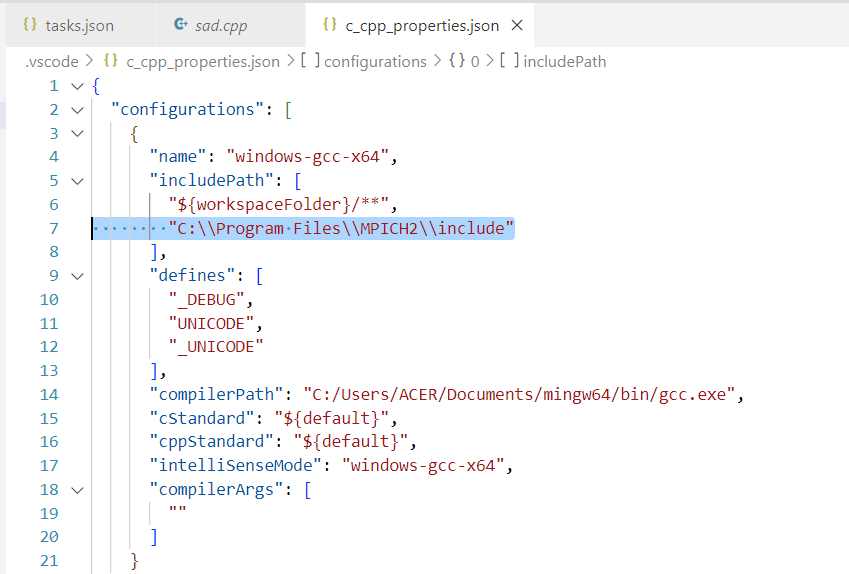
# Заключение

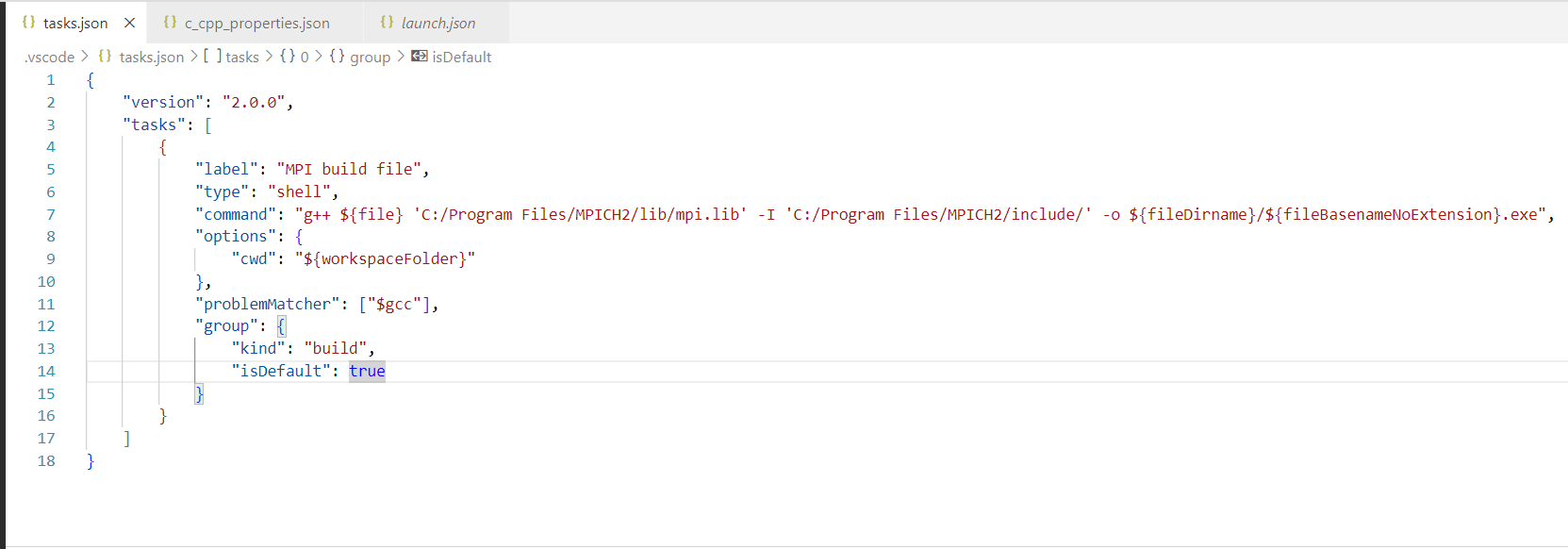
# Литература

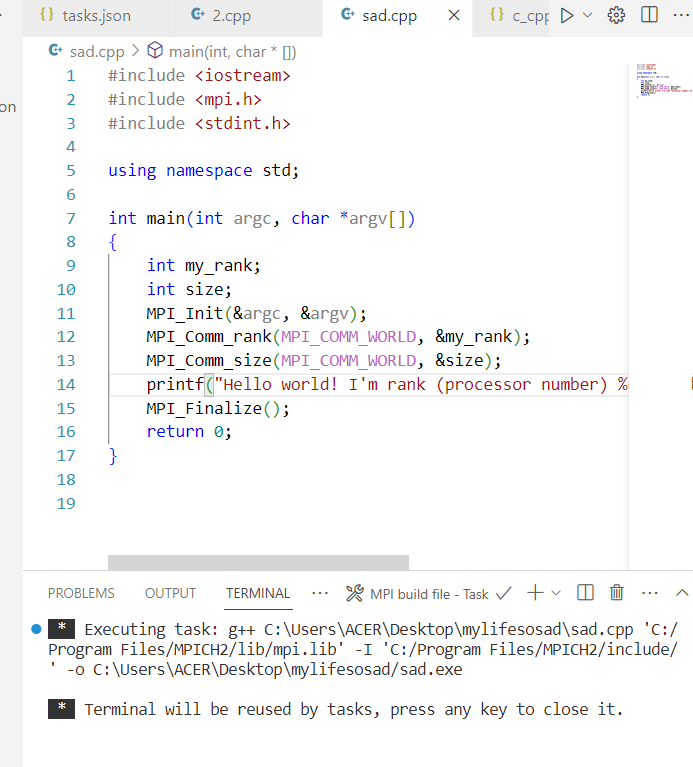
  
Рис.1 Загрузка mpich2

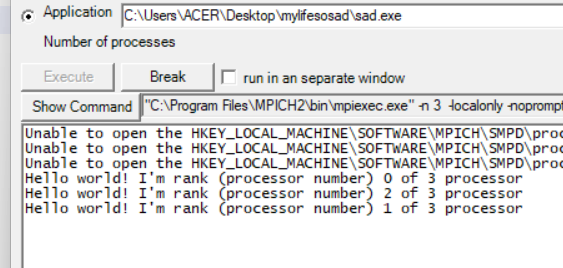
  
Рис.2 регистрация пользователя

  
Рис.3 изменение переменной среды

  
Рис.4 Изменение json-файла

  
Рис.5 Создание и изменение tasks.json

  
Рис.6 сборка программы

  
Рис.7 результат работы

**Характеристики компьютера и ПО**

* **Имя ОС: Майкрософт Windows 11 Домашняя для одного языка**
* **Имя системы: LAPTOP-00QBVBRB**
* **Изготовитель: Acer**
* **Модель: Nitro AN515-55**
* **Тип: Компьютер на базе x64**
* **Процессор: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, 2496 МГц, ядер: 4, логических процессоров: 8**
* **Установленная оперативная память: (RAM) 16,0 ГБ**

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 2**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

СОДЕРЖАНИЕ

[1. Постановка задачи 2](#_Toc335561691)

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

[3. Описание программы 2](#_Toc335561693)

[4. Реализация 2](#_Toc335561694)

[Заключение 2](#_Toc335561695)

[Литература 2](#_Toc335561696)

# 1. Постановка задачи

Дайте описание функции MPI\_Get\_processor\_name и приведите пример ее использования.

# 2. Описание используемых функций

Функция MPI\_Get\_processor\_name используется в библиотеке MPI для получения имени процессора, на котором выполняется процесс.

Вывод функции MPI\_Get\_processor\_name зависит от конкретной системы и конфигурации. Он может содержать имя компьютера, имя узла или другую информацию о процессоре.

# 3. Описание программы

После выполнения программа выводит имя системы.

# 4. Реализация

Текст программы:

#include <iostream>

#include <mpi.h>

#include <stdint.h>

using namespace std;

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

{

    int my\_rank;

    int size;

    MPI\_Init(&argc, &argv);

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

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

    char processor\_name[MPI\_MAX\_PROCESSOR\_NAME];

    int name\_len;

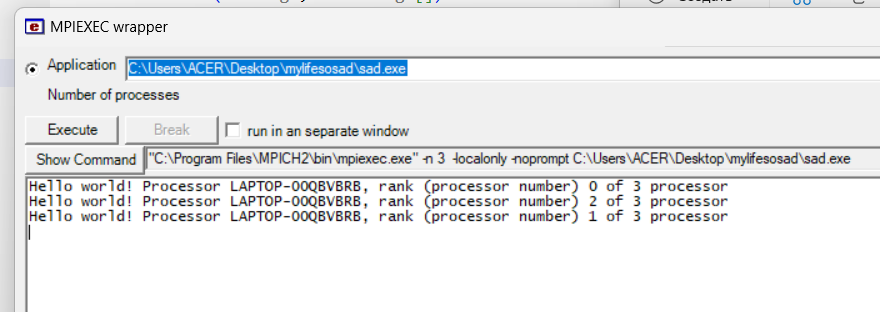
    MPI\_Get\_processor\_name(processor\_name, &name\_len);

    printf("Hello world! Processor %s, rank (processor number) %d of %d processor \n", processor\_name, my\_rank, size);

    MPI\_Finalize();

    return 0;

}

  
Рис.8 результат работы

# Заключение

При помощи функции MPI\_Get\_processor\_name(processor\_name, &name\_len) узнала имя узла на котором был запущен процесс.

# Литература

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 3**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

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

Дайте описание и продемонстрируйте выполнение функции MPI\_Wtick()

# 2. Описание используемых функций

Функция MPI\_Wtick() используется в библиотеке MPI для получения временного шага, то есть минимальной единицы времени, с которой может работать MPI.

# 3. Описание программы

После выполнения программа выводит имя системы.

# 4. Реализация

Текст программы:

#include <iostream>

#include <mpi.h>

#include <stdint.h>

using namespace std;

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

{

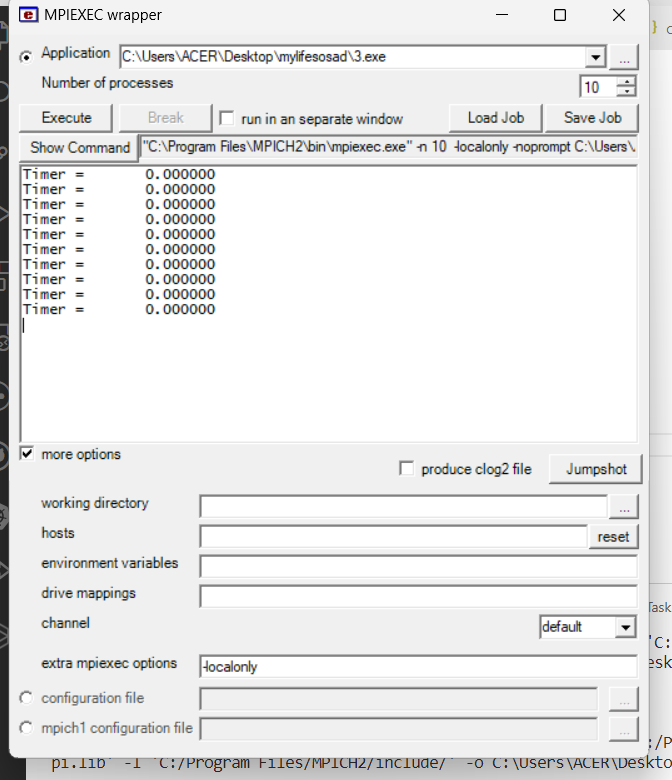
    MPI\_Init(&argc, &argv);

    printf("Timer = %14lf \n", MPI\_Wtick());

    MPI\_Finalize();

    return 0;

}

  
Рис.9 результат работы

# Заключение

При помощи функции MPI\_Wtick() узнала минимально значение кванта времени, с которой может работать MPI

# Литература

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 4**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

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 последовательную программу суммирования ряда чисел (1/(1+i)). Проведите исследование времени выполнения программы в зависимости от предела ряда. Предел ряда взять106, 107, 108, 109. Для определения времени выполнения программы используйте функцию MPI\_Wtime(), описание которой также приведите в отчете. Полученные данные приведите в табличном виде (предел ряда, результат суммирования, время выполнения).

# 2. Описание используемых функций

MPI\_Wtime() - это функция из библиотеки MPI, которая возвращает текущее время в секундах. Она может использоваться для измерения времени выполнения определенного участка кода или для синхронизации процессов.

# 3. Описание программы

После выполнения программа выводит результат суммирования при определенном пределе ряда и время выполнения

# 4. Реализация

#include <iostream>

#include <mpi.h>

#include <stdint.h>

#include <cmath>

using namespace std;

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

{

    double begin, end, summ=0;

    MPI\_Init(&argc, &argv);

    int j = 6;//принимает значения (6-9)

    begin = MPI\_Wtime();

    for (int i = 0; i < pow(10, j); i++)

        summ += (float)1/(1 + i);

    end = MPI\_Wtime();

    cout << "10^" << j << " ";

    printf("Summ=%f", summ);

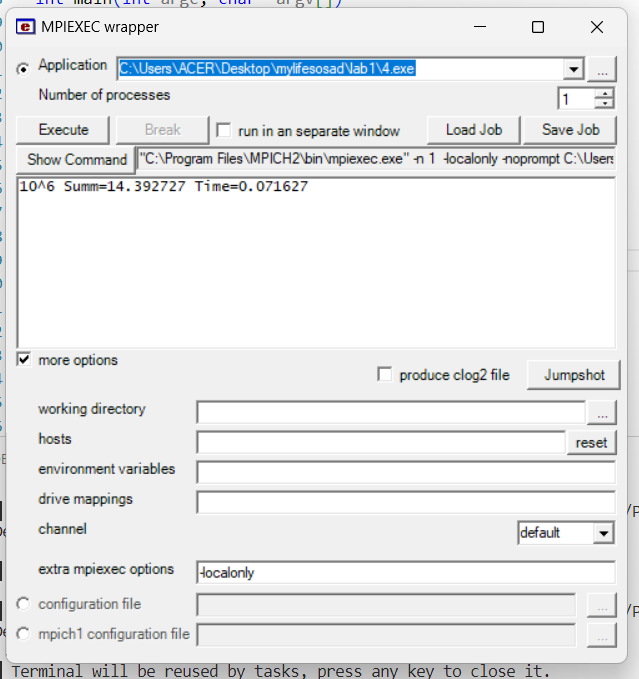
    printf(" Time=%f", end - begin);

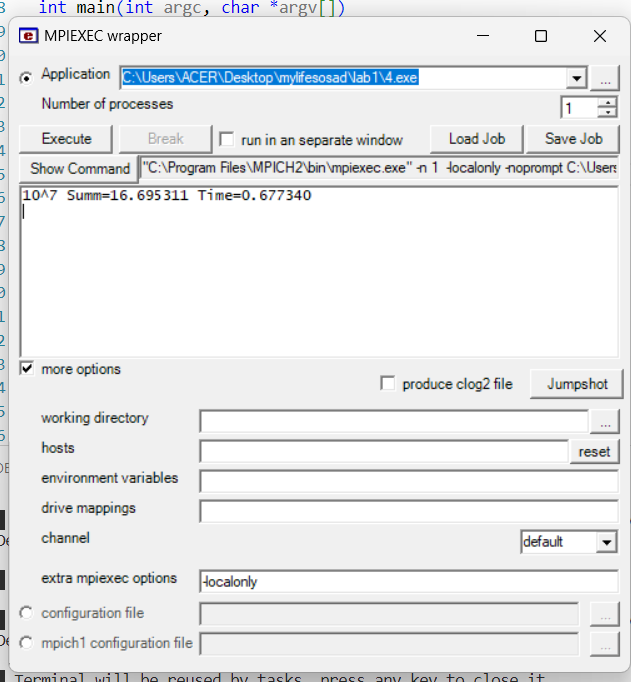
    cout << endl;

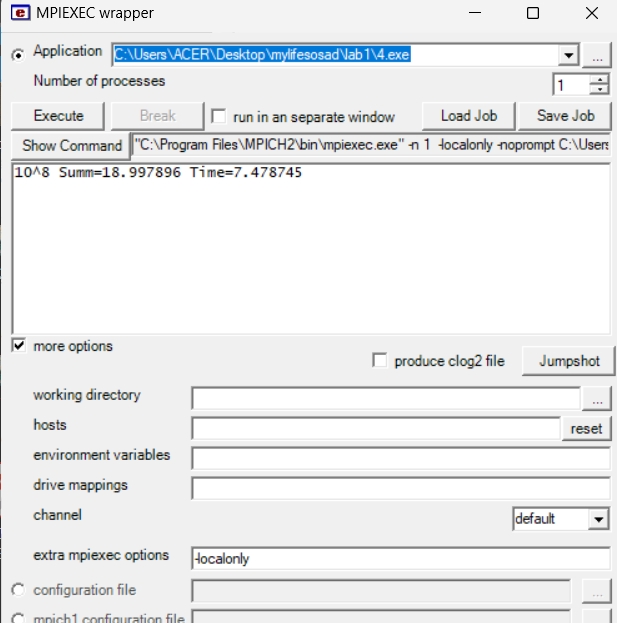
    MPI\_Finalize();

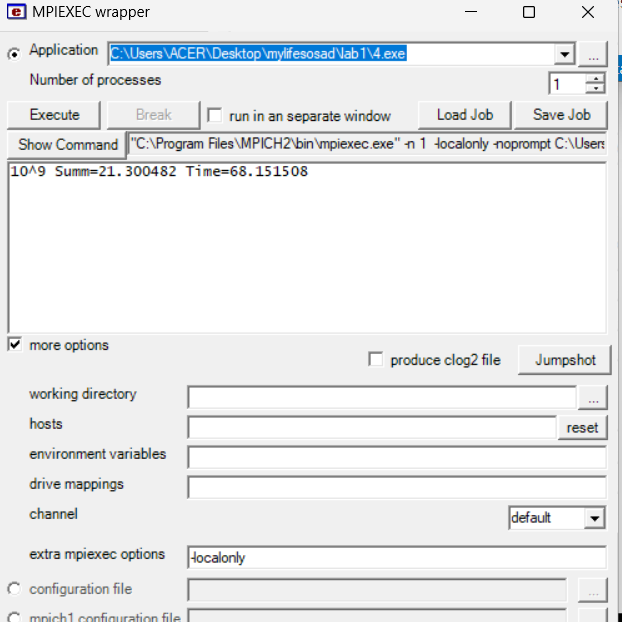
    return(0);

}

  
Рис.10 результат работы

  
Рис.11 результат работы

  
Рис.12 результат работы

  
Рис.13 результат работы

|  |  |  |
| --- | --- | --- |
| предел ряда | результат суммирования | время выполнения |
| 106 | 14.392727 | 0.071627 |
| 107 | 16.695311 | 0.677340 |
| 108 | 18.997896 | 7.478745 |
| 109 | 21.300482 | 68.151508 |

# Заключение

Оценивая результаты, представленные в таблице, можно сказать, что при увеличении предела ряда в 10 раз, время на выполнение суммирования в последовательной программе соответственно увеличиваться примерно в 10 раз.

# Литература

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 5**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

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 выводит на каждом процессе номер процесса (rank) и переменную n, которая вычисляется следующим образом: на четных – n=2\*rank, на нечетных – n=3\*rank. Запуск параллельной программы осуществите на различном количестве процессов (например, size=10,15,20).

# 2. Описание используемых функций

# 3. Описание программы

После выполнения программа выводит номер процесса и переменную в зависимости от условий.

# 4. Реализация

Текст программы:

#include <iostream>

#include <mpi.h>

#include <stdint.h>

#include <cmath>

using namespace std;

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

{

    int rank, size;

    int n;

    MPI\_Init(&argc, &argv);

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

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

    if (rank % 2 == 0) {

        n = 2 \* rank;

    } else {

        n = 3 \* rank;

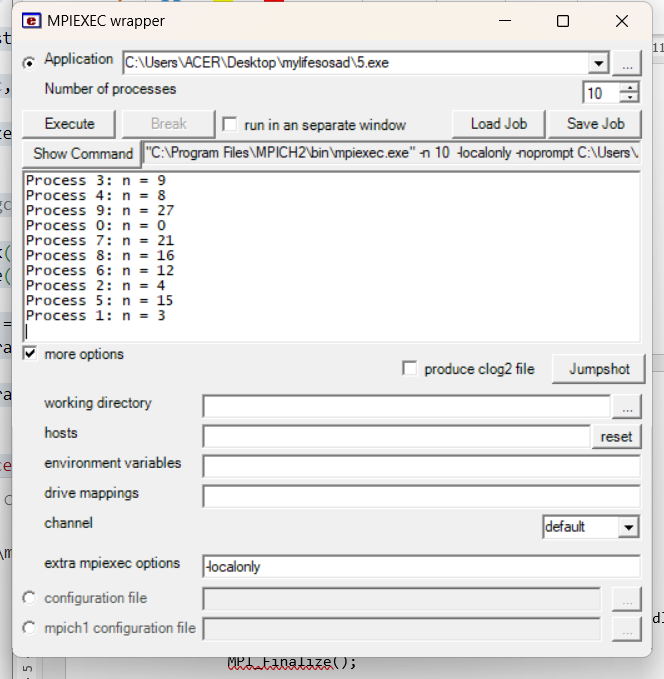
    }

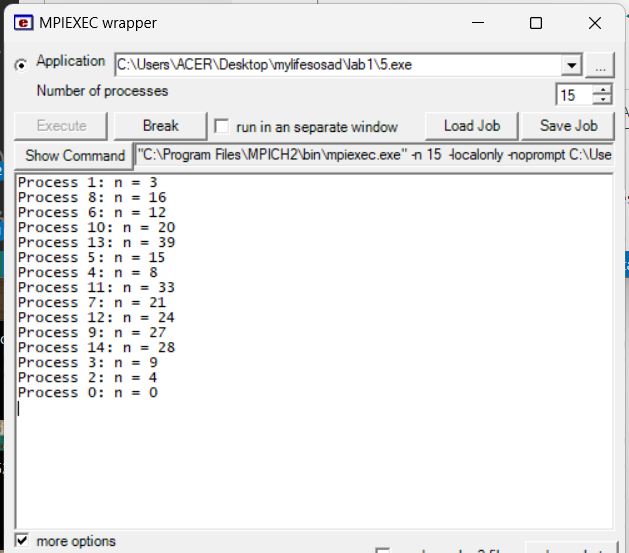
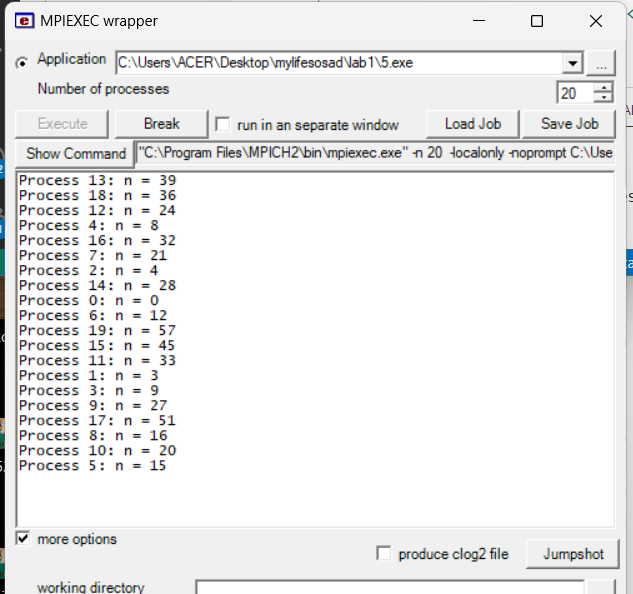
    cout << "Process " << rank << ": n = " << n << endl;

    MPI\_Finalize();

    return(0);

}

  
Рис.14 результат работы, size=10

  
Рис.15 результат работы, size=15  
  
Рис.16 результат работы, size=20

# Заключение

Выполнила поставленную задачу на различном количестве процессов.

# Литература

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 6**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

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 определяет среднее арифметическое 100 чисел, заданных случайным образом от 0 до 100 на каждом процессе. Вывод должен содержать номер процесса и вычисленное среднее значение случайных чисел для данного процесса

# 2. Описание используемых функций

Функция stand() используется, чтобы каждый процесс имел своё начальное значение для генератора случайных чисел.

# 3. Описание программы

После выполнения программа выводит среднее значение, разделив сумму на кол-во чисел.

# 4. Реализация

Текст программы:

#include <iostream>

#include <cstdlib>

#include <mpi.h>

#include <ctime>

using namespace std;

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

{

    int rank, size, numbers[100];

    double sum = 0, average = 0;

    MPI\_Init(&argc, &argv);

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

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

    srand(time(NULL) + rank);

    for (int i = 0; i < 100; i++) {

        numbers[i] = rand() % 101;

        sum += numbers[i];

    }

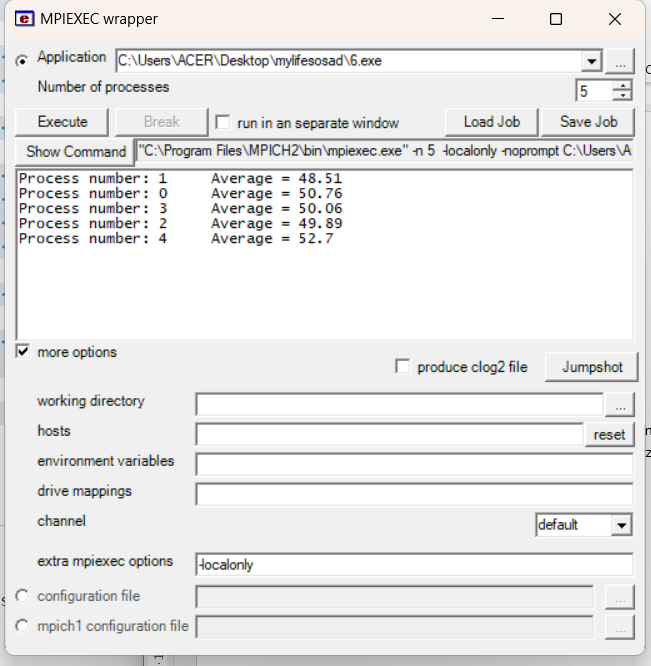
    average = sum / 100;

    cout << "Process number: " << rank << "     Average = "<< average << endl;

    MPI\_Finalize();

    return(0);

}

  
Рис.17 результат работы

# Заключение

Выполнила поставленную задачу.

# Литература

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИНСТИТУТ ЦИФРЫ**

**ДОМАШНЯЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ**

**“ Технологии параллельных вычислений”**

**Задача № 7**

студентки 3 курса

**Колесник Полины Олеговны**

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Преподаватель:

к-т физ.-мат.наук, доцент

С.В. Стуколов

Работа защищена

« »

“ ” 2023 г.

Кемерово 2023 г.

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, в которой на каждом процессе создается одномерный массив a[n], инициализация элементов которого происходит на каждом процессоре одинаково a[i]=i. Затем, каждый процесс производит вычисление сумм первых k элементов данного массива, где k – номер процесса, увеличенный на 1. Сделайте контрольный вывод результатов: номер процессора, результат вычислений

# 2. Описание используемых функций

# 3. Описание программы

На каждом процессоре создается массив из 10 элементов целого типа и заполняется числами 0-9. После чего выводится номер процессора и сумма элементов в соответствии с условием, поставленной задачи.

# 4. Реализация

Текст программы:

#include <iostream>

#include <mpi.h>

#include <stdint.h>

#include <cmath>

using namespace std;

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

{

    int rank, summ = 0, size, A[5];

    MPI\_Init(&argc, &argv);

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

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

    for(int i = 0; i < rank + 1; i++) {

        A[i] = i;

        summ += A[i];

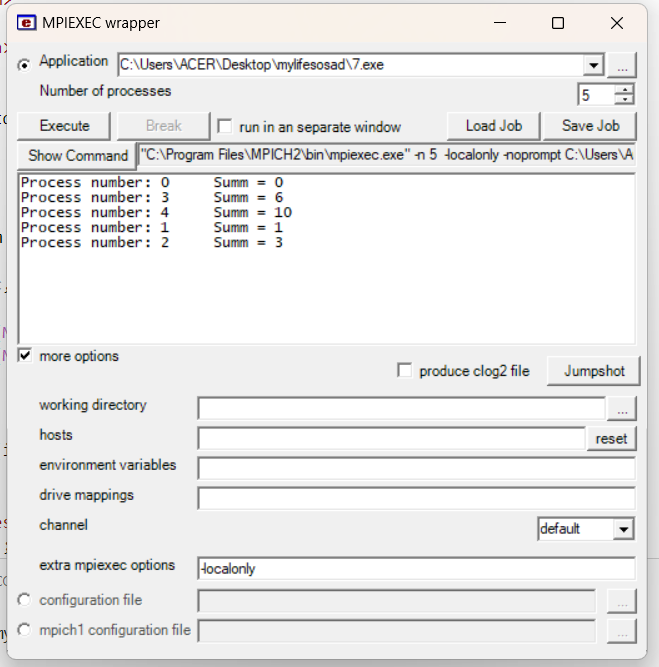
    }

    cout << "Process number: " << rank << "     Summ = "<< summ << endl;

    MPI\_Finalize();

    return(0);

}

  
Рис.18 результат работы

# Заключение

# Литература