Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

**Тема работы**

**“Потоки”**

Студент: Ханнанов Руслан Маратович

Группа: М8О-208Б-20

Вариант:8

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

<https://github.com/Naksen/OS>

**Постановка задачи**

Задача: Есть К массивов одинаковой длины. Необходимо сложить эти массивы. Необходимо предусмотреть стратегию, адаптирующуюся под количество массивов и их длину (по количеству операций).

**Общие сведения о программе**

Для реализации поставленной задачи нам нужны следующие библиотеки:  
<iostream> -  [заголовочный файл](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB) с классами, функциями и переменными для организации ввода-вывода в языке программирования [C++](https://ru.wikipedia.org/wiki/C%2B%2B).

<chrono> - для функций, работающих со временем.

<thread> - для работы с потоками.

<vector> - для работы с динамическими массивами - векторами.

<string> - для работы со строками.

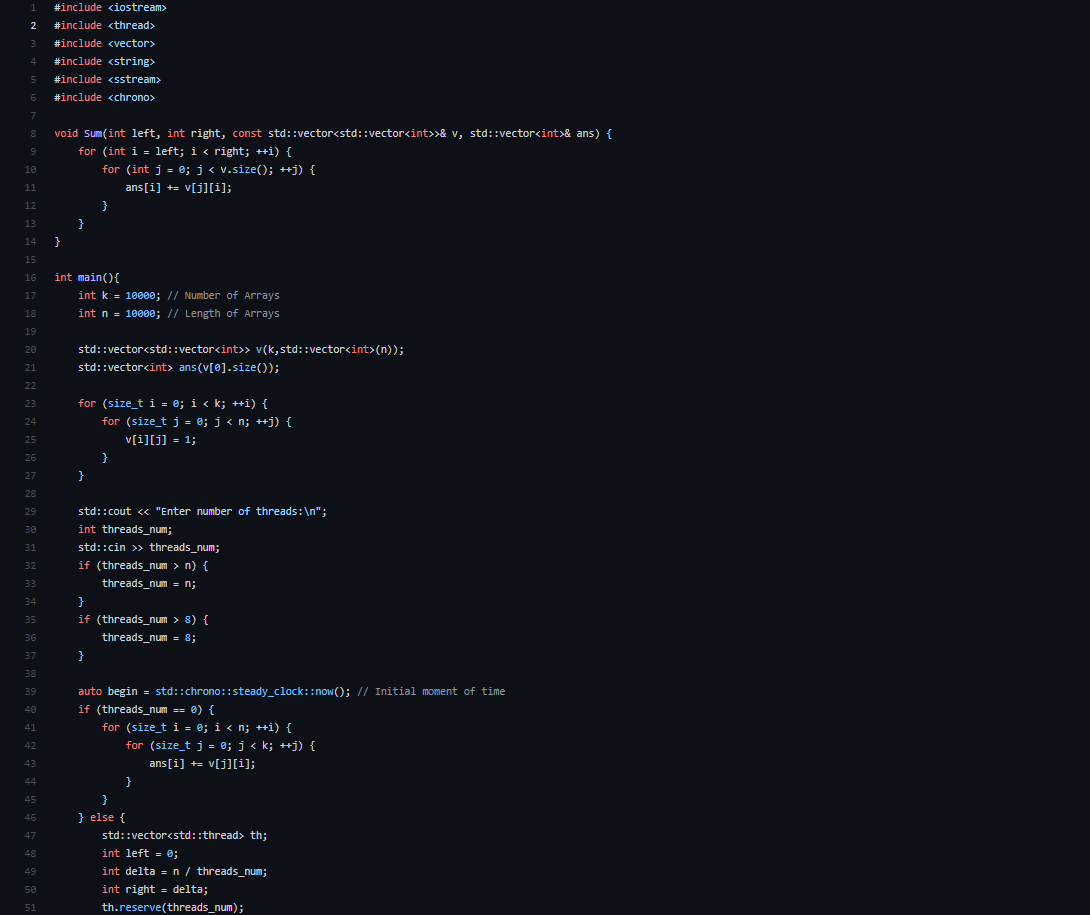
<sstream> - заголовочный файл с классами, функциями и переменными для организации работы со строками через интерфейс потоков.

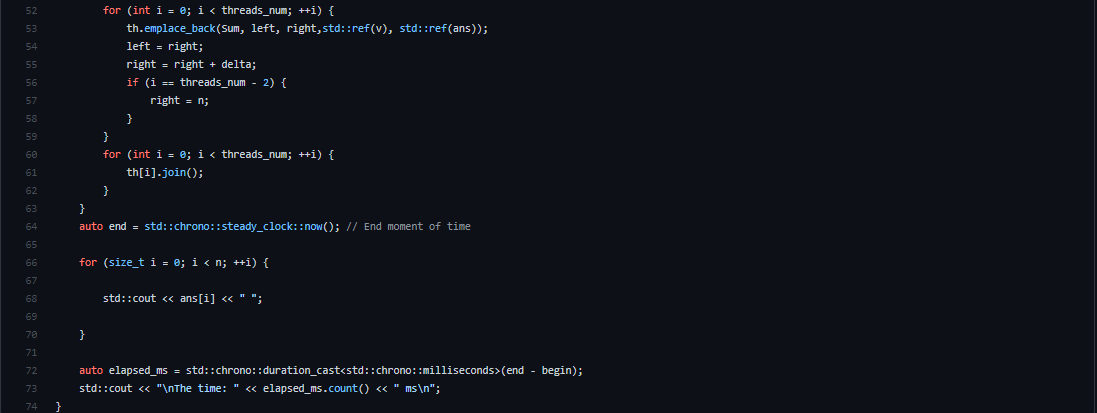
Для работы с потоками я использую библиотеку thread в C++. Я создаю вектор потоков и заполняю его ими по необходимости. В моей реализации потоки работают с регулярными фунцкиями, а именно с фунцкией sum, как раз и производящей нужные вычисления. Вызовом join я блокирую вызывающий поток до тех пор, пока он не выполнит работу. С помощью библиотеки chrono я замеряю время выполнения нужных вычислений для сравнения между собой запуска программы с разным количеством потоков.

**Общий метод и алгоритм решения**

Для реализации заданной задачи я использую массивы с уже заданными длинами, этим я избегаю “лишнего” ввода, который при необходимости можно добавить достаточно быстро. В программе есть такие переменные: k – количество массивов и n – их длина. Все массивы я заполняю единицами, для облегчения проверки. После программе подается на вход количество потоков, число которых не должно превышать 8, в таком случае оно уменьшается до заданного значения. Сделано это по причине отсутствия явного ускорения работы после излишнего увеличения числа потоков. Объем работы между потоками я разделяю поровну в зависимости от их количества. Таким образом каждый поток производит суммирования всех массивов только на диапазоне с определенными для него индексами, что позволяет избежать вмешательство одного потока к информации задействованной другим потоком. Каждый поток запускается с регулярной функцией sum, которая производит суммирования всех массивов на определенном диапазоне индексов.

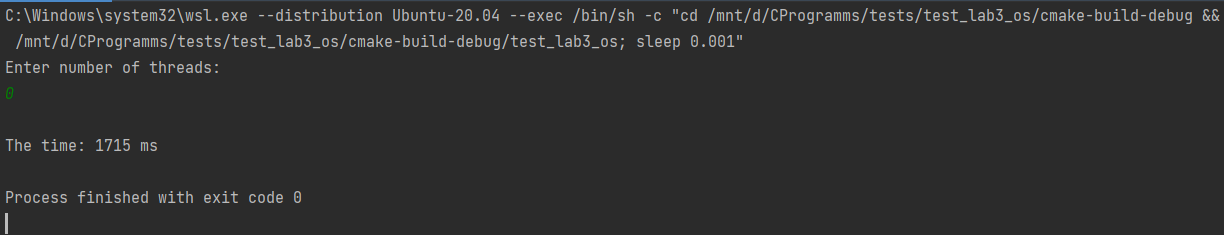
**Исходный код**



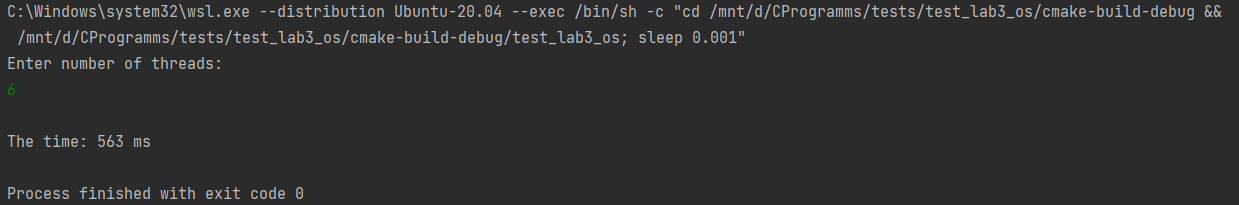


**Демонстрация работы программы**

Тест 1:



Тест 2:



**Выводы**

Благодаря данной лабораторной я успешно ознакомился с работой потоков в Linux и тем, как они устроены. Во время выполнения своего задания я изучил многие системные вызовы и научился применять их в своей программе, а также я узнал многие тонкости работы с потоками.