

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «Параллельные алгоритмы»**  
**Тема: Основы работы с процессами и потоками**

Студент гр. 9304

Попов Д.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

## **Цель работы.**

Ознакомиться с работой с процессами и потоками в языке программирования C++.

## **Задание.**

Выполнить поэлементное сложение 2х матриц  $M \times N$

Входные данные: Две матрицы вводятся из файла или генерируются.

Результат: Сумма двух матриц записывается в файл

1. Выполнить задачу, разбив её на 3 процесса. Выбрать механизм обмена данными между процессами.
  - a. Процесс 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом).
  - b. Процесс 2: выполняет сложение
  - c. Процесс 3: выводит результат
2. Выполнить задачу, разбив её на 3 потока.
  - a. Поток 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом).
  - b. Поток 2: выполняет сложение
  - c. Поток 3: выводит результат
3. Разбить сложение на P потоков.

Исследовать зависимость между количеством потоков, размерами входных данных и параметрами целевой вычислительной системы.

## **Выполнение работы.**

### **Сложение матриц с помощью 3 процессов.**

С помощью `fork()` порождаются процессы-потомки, которые почти идентичны потоку-родителю (не наследуются `pid` процесса, израсходованное время ЦП и т.п.). Весь код после `fork()` выполняется дважды, как в процессе-потомке, так и в процессе-родителе (чтобы не было дублирования, используется `switch` по `pid` процесса).

Для передачи данными между процессами использовалась разделяемая память, так как он самый быстрый (не приводит к переключению контекста между процессом и ядром). Для создание общей памяти использовалась функция mmap.

### **Сложение матриц с помощью 3 потоков.**

Создание потоков происходит с помощью конструктора thread(), в который передается функция и необходимые параметры.

Примитивом синхронизации выступает condition\_variable в связке с mutex, которые обеспечивают поочередное выполнение потоков.

### **Сложение матриц с помощью N процессов.**

Для сложения с помощью N потоков используется массив потоков. Каждый поток суммирует определенный диапазон значений.

### **Исследование зависимости между количеством потоком, размерами входных данных и параметрами вычислительной системы.**

В таблице 1 представлено сравнение размера входных данных и времени вычисления при выполнении в одном потоке:

Таблица 1 - Сравнение размера входных данных и времени вычисления

Время вычисления(миллисек.)	Размер входных данных
11	1000 x 1000
204	5000 x 5000
806	10000 x 10000
1843	15000 x 15000

В таблице 2 представлено сравнение размера входных данных и времени вычисления при распределении на 5 потоков.

Таблица 2 - Сравнение размера входных данных и времени вычисления

Время вычисления(миллисек.)	Размер входных данных
4	1000 x 1000
85	5000 x 5000
323	10000 x 10000
635	15000 x 15000

### **Выводы.**

В ходе выполнения лабораторной работы была реализована программа на языке программирования C++, получены навыки работы с потоками и процессами, а так же было установлено, что разбиение вычислительных задач на несколько потоков, положительно сказывается на скорости выполнения программы.