

**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: Основа работы с процессами и потоками**

Студент гр. 1304

Кардаш Я.Е.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2024

## **Цель работы.**

Изучить основы работы с процессами и потоками на языке C++. Выполнить практическое задание по перемножению матриц с использованием нескольких процессов и нескольких потоков.

## **Задание.**

Лабораторная состоит из 3х подзадач, которые выполняют одинаковую задачу с использованием процессов или потоков. Реализация выполняется на C/C++

Выполнить умножение 2х матриц. Входные матрицы вводятся из файла (или генерируются). Результат записывается в файл.

1.1. Выполнить задачу, разбив её на 3 процесса. Выбрать механизм обмена данными между процессами. Процесс 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом). Процесс 2: выполняет умножение Процесс 3: выводит результат

1.2.1 Аналогично 1.1, используя потоки (std::threads)

1.2.2 Разбить умножение на P потоков (можно “наивным” способом по строкам-столбцам). Протестировать, сравнив результат вычислений с результатами из 1.2.1 / 1.1

В отчёте: исследовать зависимость между количеством потоков, размерами входных данных и параметрами целевой вычислительной системы. Сформулировать ограничения на количество потоков.

## **Выполнение работы.**

Исследуем зависимость между размером матриц и количеством потоков. Для сохранения размера блока при увеличении количества строк или столбцов матрицы следует также увеличить количество потоков, причем при увеличении количества строк и столбцов матрицы в  $n$  для сохранения размера блока требуется увеличить количество потоков в  $n^2$  раз.

Также при запуске программ с различным количеством потоков установлено, что при слишком большом количестве потоков время выполнения также растёт. Поскольку работа выполнялась на машине с 2 ядрами выделение

больше чем 2 потоков не давало прироста в скорости, что связано с количеством аппаратных потоков – превышение этого значения делает использование потоков неэффективным.

Исходя из этого можно сделать выводы:

Меньший размер блока приводит к меньшему времени вычисления при отсутствии других факторов

Число порожденных программой потоков должны быть  $\leq$  числу аппаратных потоков

Объем памяти, потребляемый потоками не может превышать доступную оперативную память

Большое число потоков (в особенности – большее чем число аппаратных) может привести к нехватке ресурсов при управлении этими потоками.

### **Выводы.**

Было исследовано, изучено основы работы с процессами и потоками на языке программирования C++.

Установлено, что способ вычисления при помощи P потоков дает преимущество по времени выполнения по сравнению с способами вычисления при помощи процессов и потоков.

Исследована зависимость количества потоков от размера матриц и установлено, что при фиксированном значении блоков, равном 100, количество

потоков растет со скоростью  $n^2$ , где  $n$  = размер матрицы / размер блока.

Также установлено, что на скорость выполнения, даже при одинаковом размере блоков и количестве потоков влияют входные данные.

Сформулированы ограничения на количество потоков, главным из которых можно считать то, что число порожденных потоков не должно превышать число аппаратных потоков.