

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

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

Студент гр. 9303

Ефимов М.Ю.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Изучить работу процессов и потоков в языке C++.

Задание.

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

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 сложение

Процесс 3: выводит результат

1.2.1

Аналогично 1.1, используя потоки (threads)

1.2.2

Разбить сложение на P потоков.

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

Выполнение работы.

Для генерации данных был написан класс `Matrix` в конструкторе которого происходит наполнение данными двумерного массива.

1.1 Реализация с помощью процессов.

Реализация сложения двух матриц с помощью процессов выполнена в файле `processes.cpp`. Разбиение на три процесса происходит с помощью функции `fork()`, которая создаёт процессы-потомки. Для того, чтобы определить, какой процесс что выполняет используется идентификатор `PID`: если значение равно 0, то это потомок, и он выполняет свою часть, иначе, с помощью функции `wait()` происходит ожидание выполнения кода потомком.

1.2.1 Реализация с помощью потоков.

Реализация сложения двух матриц с помощью потоков выполнена в файл `threads.cpp`. Поток создается с помощью конструктора `thread()`, который принимает ссылки на функцию для выполнения. Ожидание исполнения потока для продолжения исполнения программы выполняется с помощью метода `join()`. Передача данных через каналы(`pipes`).

1.2.2 Разбиение операции сложения на P потоков.

Реализация сложения двух матриц с помощью нескольких потоков выполнена в функции `Sum`. В ней создаются потоки. Все созданные потоки хранятся в векторе, и после их инициализации для каждого потока вызывается метод `join()`. В функции `Simple` через индексы определяется, какой поток какой вектор складывает.

1.3 Исследование зависимости между количеством потоков, размерами входных данных и параметрами целевой вычислительной системы.

Данные приведены ниже

Количество потоков	Размер	Время
2	1000x1000	0.00430714
4	1000x1000	0.00398308
8	1000x1000	0.00234639
16	1000x1000	0.00304437

32	1000x1000	0.00269653
8	100x100	0.000528161
8	10000x10000	0.208908

Выводы.

В процессе выполнения работы были изучены процессы и потоки.

Было выявлено, что увеличение времени при увеличении объема данных линейно. В случае количества потоков ситуация другая. Не всегда увеличение потоков ведет к увеличению, даже если время уменьшается, не обязательно линейно. Узким местом в данном случае является обращение к RAM(оперативная память) .

