

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

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

Студент гр. 9303

Махаличев Н.А.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Изучение и практическое применение работы с процессами и потоками на языке C++.

Задание.

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

- 1) Входные матрицы вводятся из файла (или генерируются);
- 2) Результат записывается в файл

1.1. Выполнить задачу, разбив её на 3 процесса. Выбрать механизм обмена данными между процессами:

- 1) Процесс 1: заполняет данными входные матрицы (читает из файла или генерирует их некоторым образом). Опционально: в этом процессе могут быть 2 потока ввода/генерации данных;
- 2) Процесс 2: выполняет сложение;
- 3) 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.

Генерация входных данных происходит с помощью `generate-matrices.cpp`. Входными данными служат количество строк и столбцов матриц. В процессе выполнения программы в файлы `first_matrix.txt` и `second_matrix.txt` записываются элементы матриц — целочисленные значения, лежащие в диапазоне $[-5000; 5000]$.

Реализация с помощью процессов.

Процесс сложения двух матриц с использованием процессов реализован в файле `process.cpp`. На вход программе подается количество строк и столбцов матриц, над которыми происходит операция сложения.

Разбиение на процессы происходит с помощью функции `fork()` библиотеки `unistd.h`, который создаёт процессы-потомки. Для разделения ролей процессов используется условный оператор `switch`, реагирующим на значение идентификатора `PID`: если оно равно 0, то выполняется код потомка, иначе функцией `wait()` библиотеки `sys/wait.h` происходит ожидание завершения работы другого процесса.

Реализация с помощью потоков.

Процесс сложения двух матриц с использованием потоков реализован в файле `thread.cpp`. На вход программе подается количество строк и столбцов матриц, над которыми происходит операция сложения.

Создание потока происходит с помощью конструктора `thread()` библиотеки `thread`, аргументами которого являются функция, необходимая к исполнению, и её аргументы. Ожидание исполнения потока для продолжения исполнения программы выполняется с помощью метода `join()`.

Разбиение операции сложения на P потоков.

Процесс сложения двух матриц с использованием потоков (включая разбиение операции сложения на P потоков) реализован в файле `threads.cpp`. На вход программе подается количество строк и столбцов матриц, над которыми происходит операция сложения, и число P потоков сложения.

Так как матрицы хранятся в одномерном векторе, то для равномерного разделения данных рассчитываются индексы для каждого потока, в диапазоне которых он будет производит сложение.

Все созданные потоки хранятся в векторе, и после их инициализации итерационно для каждого потока вызывается метод `join()`, чтобы вывод результата произошел после завершения работы всех потоков сложения.

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

Результаты зависимости времени сложения от количества потоков P при сложении матриц 8000×8000 представлены в табл. 1.

Таблица 1 — Зависимость времени сложения от количества потоков P при сложении матриц 8000×8000 .

Количество потоков P	Время сложения, мс
1	1236
2	654
4	849
6	901
8	1121
10	1205

Результаты зависимости времени сложения от размера матриц при выполнении сложения двумя потоками представлены в табл. 2.

Таблица 2 — Зависимости времени сложения от размера матриц при выполнении сложения двумя потоками.

Размер матриц	Время сложения, мс
100x100	0
1000x100	1
1000x1000	9
5000x5000	250
10000x10000	1172
15000x15000	34004

Результаты зависимости времени сложения от характеристик вычислительной системы при выполнении сложения матриц 8000x8000 двумя потоками представлены в табл. 3.

Таблица 3 — Зависимости времени сложения от характеристик вычислительной системы при выполнении сложения матриц 8000x8000 двумя потоками.

Условия вычислительной системы	Время сложения, мс
4 потока, повышенная частота процессора	681
4 потока, пониженная частота процессора	1067
1 поток, повышенная частота процессора	1172
1 поток, пониженная частота процессора	1939

Выводы.

В процессе выполнения лабораторной работы были изучены и практически применены работы с процессами и потоками на языке C++. Были написаны коды программ, соответствующие следующим ситуациям разделения задач для сложения матриц:

- 1) Разбиение чтения, сложения и вывода матриц на 3 процесса;
- 2) Разбиение чтения, сложения и вывода матриц на 3 потока;
- 3) Аналогично 2, сложение разбивается на P потоков.

При исследовании зависимости времени сложения от различных условий: размер матриц, количество потоков при сложении, характеристик вычислительной системы, было выявлено, что:

- 1) Размеры матриц оказывают большое влияние на время выполнения операции сложения. Чем больше матрицы — тем дольше осуществляется сложение;
- 2) Характеристики вычислительной системы также сильно влияют на время сложения. В основном влияние оказывают частота процессора и количество его потоков/ядер;
- 3) Не всегда увеличение количества потоков приводит к ускорению выполнения задачи. Необходимо найти оптимальное количество потоков для задачи определённой сложности, иначе большая часть времени уйдёт на инициализацию этих потоков, чем на их работу.