

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Параллельные алгоритмы»
Тема: Реализация взаимодействия потоков по шаблону
“производитель-потребитель”

Студент гр. 9304

Атаманов С.Д.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать корректную работу потоков используя шаблон “производитель-потребитель”.

Задание.

На базе лаб. 1 (части 1.2.1 и 1.2.2) реализовать итерационное (потенциально бесконечное) выполнение подготовки, обработки и вывода данных. Обеспечить параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов.

Из лаб. 1:

1.2.1

Выполнить задачу, разбив её на 3 потока.

Поток 1: заполняет данными входные матрицы (читает из файла или генерирует их, некоторым образом).

Поток 2: выполняет сложение.

Поток 3: выводит результат.

1.2.2

Разбить сложение на P потоков.

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

Выполнение работы.

Основной шаблонный класс, на котором реализовывался шаблон “Производитель-потребитель” - Queue - модифицированная очередь `std::queue`, в которую были добавлены примитивы синхронизации, которые помогают организовать безопасный доступ к очереди из разных потоков. Первая очередь предназначена для загрузки в неё указателей на созданные матрицы. Вторая очередь предназначена для загрузки в неё готового результата, который впоследствии будет выведен на печать.

Взаимодействие потоков и очередей:

- Первые потоки (может быть несколько) потенциально-бесконечно генерируют матрицы и записывают их в первую очередь.
- Вторые потоки (может быть несколько) проверяют первую очередь на наличие в ней готовых для обработки данных. Если таковые имеются, то производится сложение матриц и запись последних во вторую очередь.
- Третий поток проверяет вторую очередь на наличие в ней готовых для записи в файл данных.

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

Исследование для одного потребителя и производителя.

Таблица 1 - Сравнение размера входных данных и времени вычисления для одного потока и 1 сгенерированной матрицы:

Время вычисления(миллисек.)	Размер входных данных
228	1000 x 1000
5472	5000 x 5000
22648	10000 x 10000

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 для шести потоков:

Время вычисления(миллисек.)	Размер входных данных
118	1000 x 1000
2745	5000 x 5000
15325	10000 x 10000

Исследование для нескольких потребителей и производителей.

В таблице 3 представлено общее время работы программы в зависимости от кол-ва производителей и потребителей. Матрицы суммируются в один поток.

Один производитель и два потребителя (миллисек.)	Два производителя и один потребитель (миллисек.)	Два производителя и два потребителя (миллисек.)	Размер генерируемых данных
34212	27193	20743	6 * (5000*5000)
68287	63298	45634	12 * (5000*5000)

Из данных измерений можно сделать вывод, что оптимальное время достигается при одинаковом соотношении производителей/потребителей.

Выводы.

В ходе выполнения лабораторной работы была реализована программа на языке программирования C++, получены навыки работы с потоками, а также реализовано взаимодействия между ними по шаблону «производитель — потребитель».