

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

Попов Д.С.

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. Выбрать механизм обмена данными между потоками.

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.

Разделяемым ресурсом между потоками является структура *MatrixQueue*, которая является оберткой для двух *queue* из стандартной библиотеки шаблонов и имеет методы для безопасного взаимодействия с ними из разных потоков. В первую *queue* (далее *matrixQueue*) подаются данные, представленные указателями на структуру *Matrix*, в которой содержатся два

вектора — матрицы. Вторая *queue* (далее resultQueue) содержит результат сложения двух матриц в виде вектора.

Алгоритм работы:

- Первый поток безостановочно генерирует данные(матрицы) и складывает их в matrixQueue.
- Второй поток постоянно проверяет наличие данных в matrixQueue, и если таковые имеются, то производит сложение и передает результат в resultQueue.
- Третий поток выводит в файл результаты сложения матриц, которые имеются в resultQueue.

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:

Время вычисления(миллисек.)	Размер входных данных
39	1000 x 1000
269	5000 x 5000
1022	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 пяти потоков:

Время вычисления(миллисек.)	Размер входных данных
4	1000 x 1000
78	5000 x 5000
376	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
2955	3131	2089	6 * (5000*5000)
20633	19324	11089	10 * (10000*10000)

Из результатов видно, что оптимальное время исполнения достигается путем создания одинакового кол-ва потребителей и производителей. В противном случае, потребитель\производитель не успевает суммировать\генерировать данные.

Выводы.

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