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

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Параллельные алгоритмы»

Тема: Реализация взаимодействия потоков по шаблону «производитель- потребитель»

Студент гр. 9303	 Махаличев Н.А.
Преподаватель	Сергеева Е.И.

Санкт-Петербург

Цель работы.

Изучение и практическое применение принципов синхронизации потоков на языке C++, реализация шаблона «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.

Выполнение работы.

Для облегчения работы был произведён полный рефакторинг кода первой лабораторной работы, код структурирован, классифицирован и написан, опираясь на основы ООП.

Для решения проблемы производитель-потребитель написан класс Buffer, в котором определены два метода: produce() и consume(). Первый метод отвечает за добавление в очередь очередного набора данных (в данном случае объектов класса Matrix), в то время как второй извлекает необходимые данные из полученной очереди. Сам шаблон заключается в следующем:

- 1. Производитель добавляет в очередь данные;
- 2. Потребитель забирает полученные данные;
- 3. Если буфер пустой, потребитель ждёт поступления данных;
- 4. Если буфер полон, производитель ждёт его освобождения.

Данный шаблон реализуется с помощью синхронизации потоков. Mutex организовывает взаимоисключающий доступ к данным, lock запрашивает у mutex монопольное использование общих данных, а condition_variable используется для блокировки потока до того момента, пока другой поток не изменит условие, используемое при блокировке, и не уведомит об этом condition variable.

Программа разделена на три главных потока: генерация массивов, сложение массивов (выполняется с помощью параллельной работы N потоков) и вывод данных. Для решения поставленной задачи создаются три буфера, каждый из которых может иметь в очереди до восьми объектов класса Matrix: первые два буфера хранят первую и вторую матрицу, которые будут складываться, а третий хранит их сумму, которую необходимо вывести.

Демонстрация работы программы, а именно пятикратное сложение матриц 7х10 двумя потоками представлено на рис. 1.

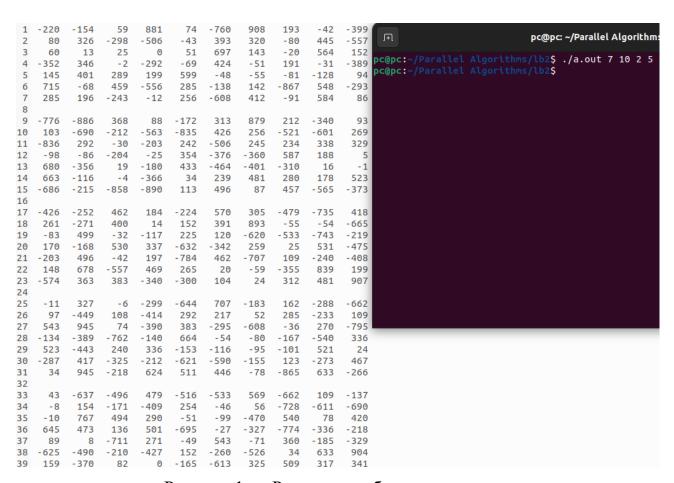


Рисунок 1 — Результат работы программы.

Выводы.

В процессе выполнения лабораторной работы были изучены и применены на практике принципы синхронизации потоков на языке C++ с помощью lock, mutex и condition_variable. Был написан код реализации шаблона «производитель-потребитель».