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 _____ Арутюнян В.В.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург
2022

Цель работы

Ознакомиться с примитивами (std::mutex) и способами синхронизации в языке программирования C++.

Задание

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

Сложение матриц происходит в 3 основных этапа:

1. Первый поток:
    a. Считывание двух матриц из файла *input.txt* На отдельной строке была задана размерность матрицы (высота, ширина через пробел). Затем представлена матрица, элементы которой разделены пробелами, а строки матрицы выделяются переносами строк. Аналогично представлена вторая матрица в том же файле.
    b. Передача считанных матриц в первый буфер в виде пары матриц.
2. Второй поток:
    a. Считывание из первого буфера пары матриц.
    b. Поэлементное сложение матриц с помощью заданного количества потоков (параллельно).
    c. Запись полученной матрицы во второй буфер.
3. Третий поток:
    a. Считывание полученной матрицы из второго буфера.
    b. Вывод матрицы в файл *output.txt* в том же виде, как представлена матрица во входном файле.

Итерационное считывание обеспечивается повторным проведением описанных выше этапов.

Количество повторов, максимальный размер буфера и количество потоков, параллельно выполняющих суммирование матриц, задаются отдельными переменными.

Потокобезопасный буфер BlockingQueue

Работа буфера основывается на одном мьютексе, двух условных переменных и готовой очереди.

Мьютекс общий как для добавления элемента в буфер, так и для его взятия его из неё.

Условные переменные используются для ожидания потоков:

1. Ожидание потока при попытке положить элемент в полностью заполненный буфер.
2. Ожидание потока при попытке получить элемент из пустого буфера.

Выводы

В ходе выполнения лабораторной работы была изучена работа с примитивами и способами синхронизации в C++.

Реализовано итерационное выполнение подготовки, обработки и вывода данных.

Обеспечено параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов.