

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

Куршев Е.О.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Изучить принципы синхронизации потоков на языке C++, изучить 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.

Выполнение работы.

Для решения поставленной задачи были написаны два класса: Buffer и Matrix. Для класса Buffer были определены два метода: produce() и consume(). Метод produce() добавляет в очередь объект типа Matrix, а consume() удаляет из очереди элемент.

В шаблоне происходят следующие операции:

1. Производитель добавляет в очередь данные;
2. Потребитель забирает данные из очереди;
3. Если буфер полон, производитель блокируется и ждёт освобождения буфера;
4. Если буфер пуст, потребитель блокируется и ждёт, пока буфер станет не пустым.

В языке C++ шаблон реализован с помощью замка — lock, примитива синхронизации — mutex и условной переменной — condition_variable.

Примитив mutex обеспечивает взаимоисключающий доступ к данным, замок lock запрашивает у mutex использование общих данных для одного конкретного процесса, а condition_variable используется для блокировки потоков.

В программе используется три главных потока: поток генерации массивов, поток сложения матриц и поток вывода данных. В свою очередь, поток сложения матриц разбивается на N потоков, которые задаёт пользователь. Также создаются три буфера: первый — хранит первую матрицу, второй — вторую матрицу, третий — результат.

Выводы.

В ходе выполнения лабораторной работы были изучены примитивы и принципы синхронизации потоков на языке C++. Для потенциально бесконечных вычислений был реализован механизм синхронизации “производитель-потребитель”.