

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. 9303

Халилов Ш.А.

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 задачи были написаны два класса: `SafeQueue`. для класса `SafeQueue` были определены два метода: `push()` — «производить» и `pop()` — «потреблять».

Метод `push()` добавляет в очередь объект типа `Matrix`, а `pop()` удаляет из очереди элемент.

В соответствии с нормами можно использовать:

1. Производитель производит данные по очереди;
2. Потребитель берет данные из очереди;
3. Если буфер заполнен, производитель блокируется и ждет возврата буфера;
4. Если буфер становится пустым, потребитель блокируется и ждет, пока буфер не пустой.

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.

Программа использует три основных потока: поток генерации массивы, поток сложения матриц и поток вывода данных. В свою очередь, поток сложения матриц разбивается на N потоков. Также создаются три буфера:

первый - хранит первую матрицу.

вторая — вторая матрица.

третья — 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”.