

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»

Студентка гр. 0381

Ионина К.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

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.

Был взят класс `Matrix` из предыдущей лабораторной работы.

Написан класс `Buffer`, в котором содержатся методы `produce()` и `consume()`. Первый добавляет в очередь объекты класса `Matrix`. Второй извлекает из очереди необходимые данные.

Общий вид решения задачи заключается в следующем. Производитель отправляет объекты в специальную коллекцию – буфер фиксированного размера. Когда потребитель освобождается, он отправляет запрос на извлечение одного из объекта буфера. Если буфер пуст, потребитель блокируется и ждет, если буфер переполнен – ждет производитель.

Решение данной задачи реализовано с помощью синхронизации потоков. `Mutex` организывает взаимоисключающий доступ к данным. `Lock` запрашивает у `mutex` монопольное использование общих данных и либо происходит захват мьютекса этим потоком и ни один другой поток не может получить доступ к данным, либо поток блокируется (если `mutex` уже захвачен другим потоком). `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`.

В программе реализовано 3 главных потока. Генерация матриц, их сложение (выполняется с помощью параллельной работы N потоков) и вывод данных. Было создано 3 буфера. Первые два хранят сгенерированные матрицы, а третий – результат сложения этих матриц. Каждый буфер может иметь до 10 объектов класса Matrix.

Демонстрация работы программы (четырежды сложение матриц размера 5x5 тремя потоками) изображена на рис.1.

```
result.txt
1 5 5
2 18 56 30 40 54
3 22 146 142 152 192
4 94 106 94 16 90
5 174 26 70 50 156
6 88 40 118 118 146
7
8 5 5
9 18 56 30 40 54
10 22 146 142 152 192
11 94 106 94 16 90
12 174 26 70 50 156
13 88 40 118 118 146
14
15 5 5
16 18 56 30 40 54
17 22 146 142 152 192
18 94 106 94 16 90
19 174 26 70 50 156
20 88 40 118 118 146
21
22 5 5
23 18 56 30 40 54
24 22 146 142 152 192
25 94 106 94 16 90
26 174 26 70 50 156
27 88 40 118 118 146
28
```

Рисунок 1 – Демонстрация работы программы

Выводы.

В ходе выполнения лабораторной работы была изучена реализация взаимодействия потоков по шаблону «производитель-потребитель», принципы синхронизации потоков на языке C++ с помощью lock, mutex, condition_variable.