

**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

Борисовский В.Ю.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

## **Цель работы.**

Ознакомиться с использованием примитивов и механизмов синхронизации в языке программирования C++. А также решить проблему взаимодействия потоков по шаблону “Производитель-потребитель”, используя полученные знания.

## **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.**

Для выполнения работы было заведено два буфера, для обработки исходных матриц (генерация и передача для суммирования) и для обработки результирующих матриц (суммирование и вывод в консоль). В функции `produceOriginMatrices` производится генерация двух матриц до захвата мьютекса, так как данная операция может выполняться параллельно несколькими потоками. После захвата мьютекса, посредством функции `wait` у условной переменной происходит ожидание до тех пор, пока в буфере не освободится место, затем в буфер исходных матриц записывается пара сгенерированных матриц и посылается уведомление в условную переменную консьюмера о том, что появились данные для обработки. В функции `consumeOriginMatrices` происходит изъятие из буфера исходных матриц пары матриц (в секции защищенной мьютексом), после чего в секции, которая не защищена мьютексом вызывается функция `produceResultMatrices`. В функции `produceResultMatrices` в секции не защищенной мьютексом происходит суммирование пришедших в аргументе матриц (это действие может выполняться одновременно в нескольких потоках для разных пар матрицы).

Затем под защитой мьютекса результирующая матрица пишется в буфер результирующих матриц при условии что в нем есть свободное место и вызывает уведомления у условной переменной, которая ответственна за то, чтобы разбудить функцию консьюма вывода матрицы в консоль. В функции `consumeResultMatrices` под защитой мьютекса происходит извлечение матриц из результирующего буфера и их вывод в консоль. Таким образом матрицы генерируются и суммируются параллельно при этом ресурсы программы не тратятся в пустую на ожидание появления/освобождения места в буферах за счет использования условных переменных.

## **Выводы**

В ходе выполнения лабораторной работы было написана программа на языке программирования C++ для попарного сложения потока матриц.

С помощью примитивов и механизмов синхронизации в языке программирования C++ обеспечено параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов.