

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

Боблаков Д.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Ознакомиться с использованием примитивов и механизмов синхронизации в языке программирования C++. Реализовать корректную работу потоков используя шаблон “производитель-потребитель”.

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 реализации паттерна Producer-Consumer было создано два буфера. Первый буфер использовался для хранения сгенерированных матриц, второй буфер использовался для хранения матриц после суммирования и до вывода.

В функции `produce_m1` 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 коньюмера о том, что появились данные для обработки. В функции `consume_m1` происходит изъятие из буфера исходных матриц пары матриц (в секции защищенной мьютексом), после чего в секции, которая не защищена мьютексом вызывается функция `produce_m2`. В функции `produce_m2` 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 `consume_m2` 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 была реализована программа на языке программирования C++, получены навыки работы с потоками, а так же реализовано взаимодействия между ними по шаблону «производитель — потребитель».