МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе № 3

по дисциплине «Параллельные алгоритмы»

Тема: Реализация структур данных без блокировок

Студент гр. 0303 Сологуб Н.А.

Преподаватель Сергеева Е.И.

Санкт-Петербург
2023

Цель работы.

Выполняется на основе работы 2.

Реализовать очередь, удовлетворяющую lock-free гарантии прогресса.

Протестировать доступ к реализованной структуре данных в случае нескольких потоков производителей и потребителей.

Выполнение работы.

1) Класс QueueLockFree

Был реализован класс QueueLockFree, который реализует очередь без блокировок. Метод **void produce(const Matrix& matrix)** добавляет матрицу в очередь. Метод **bool consume(Matrix& value)** извлекает элемент из очереди, записывает данные извлечённого узла в матрицу и возвращает true, если извлечь элемент удалось и false, если очередь пуста.

2) Время выполнения очереди без блокировки

Были проведены измерения времени очередей. При измерениях очередь обрабатывала 600 задач по умножению матриц 10×10. Результаты работы очереди при различных количествах производителей/потребителей представлены в табл. 1

Таблица 1 - зависимость времени работы очередей от количества производителей/потребителей

Количество потоков: Производители/Потребители	Время “грубой” блокировки, с	Время “тонкой” блокировки, с	Время без блокировки, с
2/2	0.38057	0.350042	0.337669
5/5	0.812425	0.805419	0.794337
12/12	0.856847	0.824529	0.819146
12/1	0.226718	0.224268	0.222423
1/12	0.83228	0.826941	0.813849

Из таблицы видно, что очередь без блокировок работает быстрее очередей с блокировками, что логично, так как потоки могут вызывать методы очереди без блокировки.

Выводы.

В ходе выполнения работы была реализована очередь без блокировки. Было выявлено, что очередь без блокировки работает быстрее очередей с блокировками.