

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

Мыратгелдиев А. М.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

Цель работы.

Научиться реализовывать lock-free структуру данных – lock-free очередь на основе атомарных операций.

Задание.

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.

Для решения поставленной задачи, был реализован класс *LockFreeQueue*.

Данный класс реализует два метода:

- *push* – для вставки элемента в очередь;
- *pop(val)* – для извлечения элемента из очереди в *val*;

Lock-free свойство очереди обеспечивается за счет использования атомарной операции CAS – compare and set (compare and exchange).

Исследуем скорость работы очереди без блокировок в зависимости от количества потребителей и количества производителей. Так как, программа потенциально может работать бесконечно, посчитаем количество произведенных и умноженных матриц за 2 секунды.

Производители/ Потребители	Грубая блокировка Produced/consumed	Тонкая блокировка Produced/consumed	Lock-free Produced/consumed
1/1	13540/2193	14483/2204	15468/2124
1/5	7345/7345	7364/7364	8513/8512
5/1	9008/1562	9421/1479	9721/1511
10/10	2696/2688	2893/2883	2921/2911

Из таблицы видно, что lock-free очередь работает эффективнее, чем другие реализации.

Выводы.

В данной лабораторной работе была реализована lock-free очередь на основе алгоритма Майкла и Скотта. Было выявлено, что очередь без блокировок работает эффективнее, чем очереди с блокировками. За счет использования атомарных операций мы смогли повысить уровень параллелизма.