**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. 0304 _____ Асташёнок М.С.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург

2023

**Цели работы.**

Изучить принцип построения потокобезопасных структур данных без блокировками.

**Постановка задачи.**

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.

**В отчёте:** Сравнить производительность с реализациями структур данных из работы. В отчёте сформулировать инвариант структуры данных.

**Выполнение работы.**

Была реализована очередь без блокировок на основе очереди Майкла-Скотта. Класс самой очереди называется `LockFreeQueue`.

Внутри себя данная очередь устроена как односвязный список, который поддерживает две операции - положить элемент в конец (`push`) и взять элемент из начала (`pop`). Каждый элемент очереди хранит в себе данные узла и атомарный указатель на следующий элемент списка. В обоих методах очереди реализовано использование бесконечного цикла, в котором происходит попытка атомарной замены указателя. При работе данной очереди важно соблюдать инвариант - по окончанию выполнения любого из методов не должно быть состояния, когда голова очереди находится правее хвоста очереди.

У lock-free очереди, используемой несколькими потоками, существует несколько проблем. В случае с методом `push` - нельзя выполнить добавление элемента в очередь и перемещение хвоста атомарно. В случае с методом `pop` - мы не можем гарантировать соблюдение инварианта очереди, так как в процессе выполнения `push` могут быть случаи, что голова будет указывать на элемент правее, чем тот, на который указывает хвост. Для решения обеих этих

проблем был добавлен механизм “помощи” другим потокам, который внутри методов будет обнаруживать промежуточные состояния изменения очереди.

**Анализ**

Перед переходом к вычислениям стоит упомянуть, что работа выполнялась на машине с количеством ядер процессора равным 2 и максимальным числом потоков равным 4. Результат одного замера получается как среднее между 10-ю выполнениями полного цикла эксперимента (от создания производителей/потребителей до их окончания работы), генерация матриц в расчёт не бралась.

За основу бралось перемножение двух квадратных матриц размером 300x300, рассматривались перемножения при различных значениях количества производителей и потребителей. В таблицах 1 и 2 приведены результаты, получившиеся для очереди с “грубой” и “тонкой” блокировкой в прошлой лабораторной работе. Результат измерения работы очереди без блокировки приведён в таблице 3. За результат бралось время работы в микросекундах.

Таблица 1 - Очередь с «грубой» блокировкой

| **C \ P** | **5** | **10** | **15** | **20** |
|---|---|---|---|---|
| **5** | 227980 | 213609 | 210017 | 204707 |
| **10** | 214454 | 207959 | 204348 | 196114 |
| **15** | 195367 | 203966 | 199141 | 197595 |
| **20** | 198755 | 196691 | 195731 | 197219 |

Таблица 2 - Очередь с «тонкой» блокировкой

| **C \ P** | **5** | **10** | **15** | **20** |
|---|---|---|---|---|
| **5** | 197668 | 189643 | 191233 | 189798 |
| **10** | 185977 | 186604 | 182178 | 183587 |

| | | | | |
|---|---|---|---|---|
| **15** | 185757 | 184046 | 182673 | 189623 |
| **20** | 189059 | 187670 | 184066 | 189535 |

Таблица 3 - Очередь без блокировки

| **C \ P** | **5** | **10** | **15** | **20** |
|---|---|---|---|---|
| **5** | 146740 | 133544 | 135952 | 128624 |
| **10** | 137566 | 137174 | 135336 | 133427 |
| **15** | 135091 | 133476 | 132844 | 131220 |
| **20** | 130549 | 127438 | 127393 | 129454 |

Как можно заметить, очередь без блокировок во всех ситуациях работала быстрее, чем очереди с блокировками. Также по результатам видно, что очередь без блокировок работает быстрее, когда количество производителей больше, чем количество потребителей.

**Выводы.**

В ходе выполнения лабораторной работы были изучены потокобезопасные очереди без блокировок при их работе по шаблону производитель/потребитель. Для этого была реализована очередь на основе однонаправленного списка.

Было выявлено, что очередь без блокировок работает быстрее, чем очередь с блокировкой, причем лучшие показатели производительности были в случае, когда количество производителей превышало количество потребителей.