**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Параллельные алгоритмы»**

Тема: **Реализация структур данных без блокировок**

|  |  |  |
| --- | --- | --- |
| Студент гр. 0303 |  | Парамонов В.В. |
| Преподаватель |  | Сергеева Е. И. |

Санкт-Петербург

2023

## Цель работы.

Исследовать структуры данных без блокировок. Воспользоваться структурой данных без блокировок для решения задачи производители-потребители.

**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.
* Сравнить производительность с реализациями структур данных из работы 2.
* Сформулировать инвариант структуры данных.

**Выполнение задач.**

1. **Изменения структуры решения в сравнении с лабораторной 2**:
2. Появился новый шаблонный класс nonLockThreadsQueue (многопоточная очередь без блокировок) унаследованный от baseThreadsQueue (Виртуальный базовый класс многопоточной очереди). Данный класс многопоточной очереди без блокировок реализован на базе алгоритма Michael & Scott queue. Для очистки памяти был реализован механизм “сбощика мусора”, который занимается удалением старых узлов в случае, если все потоки на некоторое время перестали вызывать метод pop.
3. Изменилась реализация классов “грубой” (roughThreadsQueue), “тонкой” (thinThreadsQueue) очередей и их базового класса (baseThreadsQueue) на использование односвязного списка.
4. **Исследование получаемых с использованием программы результатов:**
5. Инвариант для очереди на односвязном списке: голова списка всегда должна находиться до хвоста.
6. Исследуем скорость работы очереди с грубой блокировкой, с тонкой блокировкой и без блокировок в зависимости от количества производителей и потребителей, результаты измерений времени работы, усредненные для 100 запусков, представлены в таблице 1 (размеры матриц – (20, 40), (40, 20); количество генерируемых производителями и обрабатываемыми потребителями наборов данных – 200; макс. Размер очереди – 50):

Таблица 1 – Измерение времени работы разных очередей в зависимости от кол-ва потребителей и производителей

|  |  |  |
| --- | --- | --- |
| Количество производителей | Количество потребителей | Время выполнения задачи(мкс)  Для грубой блокировки  Для тонкой блокировки  Для без блокировок |
| 5 | 5 | 10586  **10568** 12429 |
| 2 | 5 | 11681 **11400** 12840 |
| 5 | 2 | 15866 **15084** 20331 |
| 10 | 10 | 10030 **9957** 17535 |
| 20 | 20 | 9848  **9632** 16861 |

Исходя из полученных данных в таблице 1 очередь с “тонкой” блокировкой быстрее, чем очередь с “грубой” блокировкой и намного быстрее очереди без блокировок. Это объясняется тем, что очередь без блокировок может выигрывать по времени выполнения в случае малой конкуренции процессов, так как при большой конкуренции в очереди с отсутствием блокировок все чаще потоки мешают друг другу выполнится полностью и происходит активная блокировка. Так же проблемой является необходимость очищения динамически выделяемых ресурсов в очереди без блокировок. Это увеличивает количество выполняемых операций и плохо сказывается на производительности.

**Заключение.**

В ходе работы была изучена разница между “грубой” и “тонкой” блокировкой. Были использованы очереди на основе блокировок и без блокировок для решения задачи производители-потребители. Было практически подтверждено, что в данной задаче с высокой конкуренцией очереди с блокировками оказались быстрее очереди без блокировок.