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 потокобезопасных структур данных без блокировок.

Студент гр. 0304 _____ Алексеев Р.В.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург
2023

Цель работы.

Изучить способы реализации потокобезопасных структур без блокировок.

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.

1. Для реализации потокобезопасной очереди без блокировок создан класс Handler, который имеет ряд методов: *Handler()* - конструктор, *~Handler()* - деконструктор, освобождающий память, *void push()* - метод помещения нового элемента в очередь, *bool pop()* - метод извлечения элемента из очереди, *void deleteRecursive()* - рекурсивное удаление очереди.

Методы *push* и *pop* содержат CAS для реализации очередь без блокировок. Для добавления нового узла в очередь выполняется сравнение с обменом для изменения ссылки на следующий элемент хвостового узла и изменение самого хвостового узла. Для извлечения элемента, в случае, если очередь не пуста, выполняется сравнение с обменом головного узла, головной узел заменяется следующим за ним.

2. Были произведены замеры времени в зависимости от количества производителей и потребителей для тонкой и грубой блокировок и для очереди без блокировок.. Во всех случаях количество задач — 10000. Результаты представлены в таблицах 1 - 3.

Таблица 1 — Время выполнения программы с грубой блокировкой.

Количество производителей/потребителей	**Время, мс**

2/2	616
10/2	609
2/10	1223
50/50	1301
100/50	1413
50/100	1456
100/100	1352

Таблица 2 — Время выполнения программы с тонкой блокировкой.

Количество производителей/потребителей	**Время, мс**
2/2	642
10/2	638
2/10	1226
50/50	1475
100/50	1416
50/100	1512
100/100	1502

Таблица 3 — Время выполнения программы без блокировок.

Количество производителей/потребителей	**Время, мс**
2/2	758
10/2	1206
2/10	1098
50/50	1689
100/50	1873
50/100	1821
100/100	1963

По таблицам видно, что программа с тонкими блокировками работает быстрее программы без блокировок.

При большом количестве производителей программа работает медленнее чем в случае с большим количеством потребителей.

При большом количестве потоков, в реализации с блокировками потоки конкурируют за мьютекс, ожидая его освобождения, что приводит к временным потерям. В реализации без блокировок большое количество потоков приводит к большому количеству false в CAS, что также приводит к временным затратам. По таблицам можно сделать вывод, что захват мьютекса является более «легкой» операцией чем множество CAS. Также в реализации без блокировок требуется безопасно освобождать память, для чего используются указатели опасности, что также требует дополнительных временных затрат.

Выводы.

В ходе работы были изучены способы реализации потокобезопасных структур данных без блокировок.

Была реализована очередь без блокировок.

Была исследована зависимость количества и отношения производителей и потребителей ко времени выполнения программы. Было установлено, что реализации с тонкими блокировками требует меньше времени для выполнения умножений чем реализация без блокировок, что связано с большим количеством возвращаемых CAS false и с необходимостью безопасного освобождения памяти, для чего используются указатели опасности.