

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 параллельной структуры данных с тонкой
блокировкой

Студент гр. 9304

Арутюнян С.Н.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

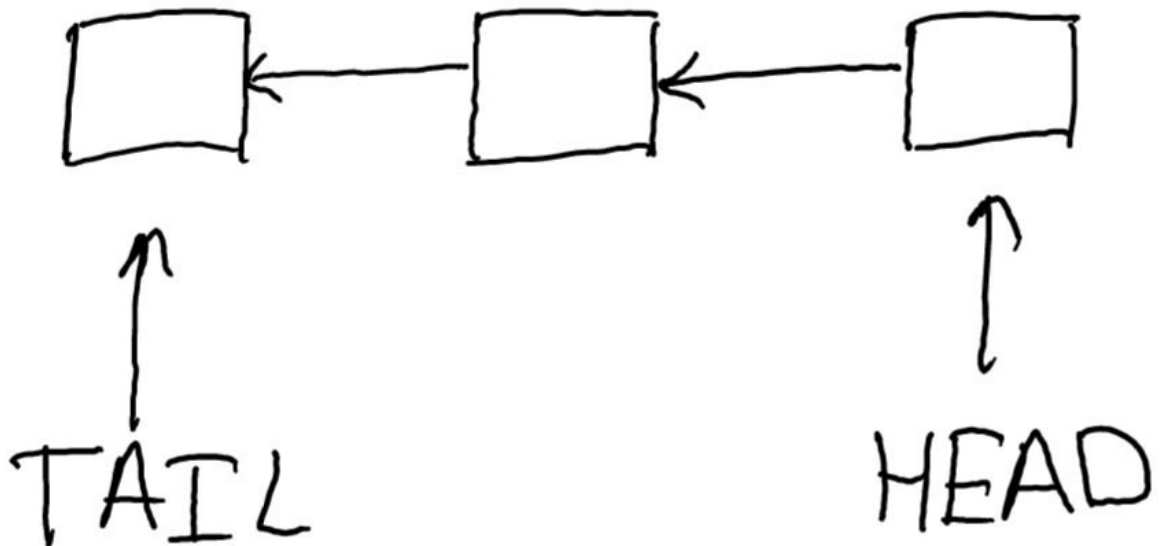
Ознакомиться с принципами работы параллельных структур данных с тонкой блокировкой.

Задание.

Обеспечить работу структуру данных из лаб.2 на основе lock-free. Протестировать доступ в случае нескольких потоков-производителей и потребителей. Сравнить производительность со структурой с грубой синхронизацией (т.е. с лаб.2). В отчёте сформулировать инвариант структуры данных.

Выполнение работы.

В лабораторной работе №2 мной использовалась очередь, синхронизируемая извне семафорами. В данной работе была реализована lock-free очередь на основе односвязного списка. Структура у такой очереди следующая:



Метод push(). Новые узлы добавляются в конец, т.е. TAIL начинает указывать на новый узел, а next последнего узла начинает указывать на новый узел.

Метод front(). В качестве текущего элемента очереди используется узел, на который указывает HEAD.

Метод pop(). В методе pop() удаляется элемент, на который указывает HEAD. На самом деле элемент не удаляется. HEAD просто сдвигается на HEAD->next, а удаление происходит чуть позже (см. ниже часть с описанием использования атомарных умных указателей). Это нужно для того, чтобы метод front() не читал уже освобожденные данные.

При выполнении лабораторной работы я столкнулся с АВА-проблемой. Есть множество различных решений этой проблемы, но я выбрал решение, основанное на умных атомарных shared-указателях. Это нужно для того, чтобы нам не приходилось явно удалять узлы списка, потому что в таком случае может возникнуть ситуация, когда во время выполнения метода front() другой поток удаляет узел, на который уже ссылается указатель из front(). Shared-указатели решают эту проблему с помощью счетчика ссылок на узлы. В таком случае pop может просто сдвинуть HEAD, после чего счетчик ссылок на первый узел уменьшится на 1. Если при этом front() уже успел сослаться на тот же метод, то счетчик ссылок будет равен как минимум 1, а значит узел не удалится.

Инвариант структуры данных

Инвариант такой очереди совпадает с инвариантом односвязного списка.

Сравнение производительности lock-free очереди со схемой “семафор + непараллельная очередь”

Время lock-free, с	Время “семафор + очередь”, с	Размер матриц
15.05	14.08	1000 x 1000
53.40	47.26	2000 x 2000

Выводы.

В ходе выполнения лабораторной работы была реализована lock-free очередь на основе atomic'ов, а также она была сравнена с очередью на основе массива с семафором в качестве примитива синхронизации.