

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе № 2
по дисциплине «Параллельные алгоритмы»
ТЕМА: РЕАЛИЗАЦИЯ ПОТОКОБЕЗОПАСНЫХ СТРУКТУР ДАННЫХ С
БЛОКИРОВКАМИ

Студент гр. 0304

Асташёнок М.С.

Преподаватель

Сергеева Е.И.

Санкт-Петербург

2023

Цели работы.

Изучить принцип построения потокобезопасных структур данных с блокировками.

Постановка задачи.

Реализовать итерационное (потенциально бесконечное) выполнение подготовки, обработки и вывода данных по шаблону “производитель-потребитель” (на основе лаб. 1 (части 1.2.1 и 1.2.2)). Обеспечить параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов.

Использовать механизм “условных переменных”.

2.1

Использовать очередь с “грубой” блокировкой.

2.2

Использовать очередь с “тонкой” блокировкой.

Выполнение работы.

Были реализованы два класса - `HardBlockingQueue` и `FineBlockingQueue`. Первый - очередь с «грубой» блокировкой, второй - очередь с «тонкой» (мелкогранулярной) блокировкой. Каждый из них содержит в себе минимальный интерфейс для работы с очередью, а именно: метод `push` (положить элемент в конец очереди) и метод `popWithWaiting` (взять элемент из начала очереди). Рассмотрим каждую из вариаций очереди более подробно.

Очередь с «грубой» блокировкой представляет из себя обёртку над стандартной очередью, операции которой будут дополнительно синхронизироваться с помощью условной переменной и мьютекса. При добавлении элемента в очередь мьютекс будет блокироваться, а условная переменная получать информацию, что очередь в данный момент не пустая.

При получении и удалении элемента из очереди мьютекс также будет блокироваться, но при этом также будет происходить проверка, пуста ли очередь. Если очередь действительно пустая, то выполнение блокируется до момента, когда условная переменная получит сигнал, что в очереди появился элемент. Если очередь в момент получения элемента уже не пуста, то элемент извлекается сразу.

Очередь с “тонкой” блокировкой представляет собой однонаправленный список, который также поддерживает операции добавления элемента в конец очереди и извлечение элемента из начала очереди. Отличием от прошлой вариации очереди в том, что теперь у нас будут задействованы два мьютекса (на начало и на конец очереди) и условная переменная. При добавлении элемента блокируется мьютекс хвоста, после чего к хвосту добавляется новый элемент. Когда хвост изменился и элемент добавился - условная переменная получает информацию об этом событии. Извлечении элемента из очереди влечёт за собой блокировку мьютекса головы, после чего проверяется, пуста ли очередь в данный момент. Когда очередь не пуста - данные извлекаются.

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 равняется среднему между 10 выполнениями полного цикла эксперимента (от создания производителей/потребителей до их окончания работы). Генерация матриц в счет не бралась.

За основу бралось перемножение двух квадратных матриц размером 300x300, рассматривались перемножения при различных значениях количества производителей и потребителей. За результат бралось время работы в микросекундах. Результаты приведены в таблице 1 и 2.

Таблица 1 - Очередь с «грубой» блокировкой

С \ Р	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

С \ Р	5	10	15	20
5	197668	189643	191233	189798
10	185977	186604	182178	183587
15	185757	184046	182673	189623
20	189059	187670	184066	189535

Как можно заметить, по результатам очередь с «грубой» блокировкой в среднем работает медленнее, чем очередь с «тонкой» блокировкой.

Также можно заметить, что для обеих очередей увеличение количества как производителей, так и потребителей, ведет к ускорению работы программы.

Выводы.

В ходе выполнения лабораторной работы были изучены потокобезопасные очереди с различными видами блокировок при их работе по шаблону производитель/потребитель (Producer/Consumer). Для этого были разработаны и проанализированы две очереди с блокировками – «грубая» и «тонкая» (мелкогранулярная).

Было выявлено, что при равных обстоятельствах, очередь с «грубой» блокировкой работает медленнее, чем очередь с «тонкой» блокировкой. Это можно объяснить полной блокировкой очереди с «грубой» блокировкой при добавлении или удалении элемента, в то время, как в очереди с «тонкой» блокировкой действия удаления и добавления элемента в очередь блокируются по отдельности – т.е. деятельность производителей не блокируется действиями потребителей и наоборот.