**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. 1303 |  | Чубан Д.В. |
| --- | --- | --- |
| Преподаватель |  | Сергеева Е.И. |

Санкт-Петербург

2024

**Цель работы**

Изучить принцип построения потокобезопасных структур данных с блокировками.

**Задание.**

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)).

Количество производителей и потребителей должно быть изменяемым.

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.**

В качестве производителя выступает функция-генератор, генерирующий матрицы и помещающий их в очередь. В качестве потребителя выступает функция, получающая матрицы из очереди, перемножающая их и записывающая результат в файл.

Очередь с “Грубой” блокировкой.

Для реализации очереди с грубой блокировкой реализован класс *Queue* в пространстве имен *hardgrained*. Для хранения данных используется класс *std::queue* стандартной библиотеки. Но данная очередь не является потокобезопасной. Для реализации грубой блокировки в *Queue* добавлен мьютекс, который при взятии лока блокирует всю структуру данных. Кроме того, для пробуждения ожидающих потоков добавлены условные переменные *in* и *out*, которые отвечают за пробуждение производителей и потребителей соответственно.

Очередь с “Тонкой” блокировкой.

Реализация очереди с тонкой блокировкой мало отличается от вышеописанной реализации очереди с грубой блокировкой. Ключевым отличием является добавление дополнительного мьютекса, который позволяет разграничить потребителей и производителей. Работает это следующим образом – конкурирующие производители борются за взятие лока на добавление, при этом не влияя на потребителей, которые борются за второй лок – на взятие элемента из очереди. Таким образом, в момент, когда производитель добавляет задачу в очередь, потребители не простаивают и выполняют полезную работу.

Сравнение потокобезопасных очередей с блокировками.

Сравнение очередей осуществлялось при помощи измерений обработки очереди 500 задач по умножению матриц 100х100. Размер очереди фиксирован и равен 50.

Время выполнения программы с грубыми блокировками

| **Производители** | **Потребители** | **Real Time, сек** | **User Time, сек** | **Sys Time, сек** |
| --- | --- | --- | --- | --- |
| 10 | 10 | 0,857 | 8,247 | 0,139 |
| 50 | 50 | 0,859 | 8,862 | 0,141 |
| 50 | 1 | 5,912 | 6,127 | 0,110 |
| 1 | 50 | 0,834 | 8,838 | 0,130 |
| 500 | 500 | 1,041 | 8,166 | 0,506 |
| 500 | 10 | 1,019 | 8,121 | 0,329 |
| 10 | 500 | 0,876 | 8,222 | 0,122 |

Время выполнения программы с тонкими блокировками

| **Производители** | **Потребители** | **Real Time, сек** | **User Time, сек** | **Sys Time, сек** |
| --- | --- | --- | --- | --- |
| 10 | 10 | 0,859 | 8,155 | 0,150 |
| 50 | 50 | 0,864 | 9,025 | 0,111 |
| 50 | 1 | 5,718 | 5,905 | 0,199 |
| 1 | 50 | 0,839 | 8,752 | 0,168 |
| 500 | 500 | 1,188 | 8,822 | 0,455 |
| 500 | 10 | 1,018 | 8,208 | 0,261 |
| 10 | 500 | 0,869 | 7,942 | 0,202 |

**Выводы.**

В ходе лабораторной работы были изучены способы реализации потокобезопасных структур данных с блокировками. Были написаны 2 программы, выполняющие одну и ту же задачу, но использующие разные очереди: с «грубой» и «тонкой» блокировкой. Также было проведено исследование производительности задач в зависимости от количества потоков-производителей и потоков-потребителей. На малом количестве производителей и потребителей заметной разницы между «грубой» и «тонкой» блокировкой нет, при большой разнице в количестве потребителей и производителей очередь с «тонкой» блокировкой показывает более хорошие результаты – это связано с тем, что время блокировки очереди разбивается на чтение и на запись в очередь.