**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Параллельные алгоритмы»**

Тема: Реализация потокобезопасных структур данных с блокировками

|  |  |  |
| --- | --- | --- |
| Студент гр. 0303 |  | Афанасьев Д.В. |
| Преподаватель |  | Сергеева Е.И. |

Санкт-Петербург

2023

**Цель работы.**

Изучение основы потокобезопасных структур в языке программирования C++.

**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)).

Обеспечить параллельное выполнение потоков обработки готовой порции данных, подготовки следующей порции данных и вывода предыдущих полученных результатов. Использовать механизм “условных переменных”.

Подзадачи:

2.1 Использовать очередь с “грубой” блокировкой.

2.2 Использовать очередь с “тонкой” блокировкой

**Выполнение работы.**

Общая часть:

Перемножение матриц было позаимствовано из лабораторной №1 части 1.2.2. Был реализован класс Flows, который распределяет потоки. Были выделены потоки «производители», которые генерируют матрицы размером 512x512, и потоки «потребители», которые перемножают матрицы. Данные потоки передают данные через потокобезопасную очередь.

Подзадача 1:

Для создания потокобезопасной очереди использовались условные переменные и мьютекс для синхронизации потоков. Для очереди были реализованы операции push и pop.

Подзадача 2:

Для создания потокобезопасной очереди с «тонкой» блокировкой использовались условные переменные и мьютекс для синхронизации потоков, только мьютексы накладывался на узлы в очереди, а не на сами операции. Для очереди были реализованы операции push и pop.

Исследование проводилось путем запуска перемножение матриц с разным количеством потоков «производителей» и «потребителей». Матрицы были размером 512x512 и количество операций, которые выполняет поток «производитель» было 10. Результаты представлены в табл. 1 и табл. 2, соответственно. Исследование проводилось на процессоре с 12 ядрами и 20 виртуальными потоками.

Таблица 1 – Исследование очереди с грубой блокировкой

|  |  |  |
| --- | --- | --- |
| Количество потоков производителей | Количество потоков потребителей | Время выполнения, мс |
| 7 | 5 | 69940 |
| 20 | 10 | 90495 |
| 10 | 10 | 88934 |
| 10 | 2 | 17163 |

Таблица 2 – Исследование очереди с тонкой блокировкой

|  |  |  |
| --- | --- | --- |
| Количество потоков производителей | Количество потоков потребителей | Время выполнения, мс |
| 7 | 5 | 61914 |
| 20 | 10 | 87954 |
| 10 | 10 | 87457 |
| 10 | 2 | 17058 |

В результате исследования можно сделать вывод, что программа с очередью с тонкой блокировкой лучше по времени, чем программа с очередью с грубой блокировкой, поскольку уменьшается количество блокировок. Время выполнения особенно разниться при увеличении гонки за ресурс.

**Выводы.**

В ходе выполнения лабораторной работы были получены практические основы работы потокобезопасными структурами данных в языке C++. Было проведено исследовании в ходе, которого было выяснено, что очередь с тонкой блокировкой позволяет уменьшить количество блокировок.