Московский Авиационный Институт

(Национальный исследовательский Университет)

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**Курсовая работа**

**по курсу «Операционные системы»**

**Реализация lock-free очередь. Сравнение с многопоточными версиями очереди, реализованными с помощью стандартного примитива блокировки мьютекс.**

|  |  |
| --- | --- |
| Студент: | Марков А.Н. |
| Группа: | М80-208Б-18 |
| Преподаватель: | Миронов Е.С. |
| Оценка: |  |
| Дата: |  |

Москва

2020

**Содержание**

1. Постановка задачи.
2. Общие сведения о программе.
3. Общий метод и алгоритм решения.
4. Демонстрация работы программы.
5. Сравнение времени работы lock-free очереди и очереди, реализованного с помощью мьютекса.
6. Выводы.

**Постановка задачи**

Реализовать lock-free очередь. Сравнить с многопоточными версиями очереди, реализованными с помощью стандартного примитива блокировки мьютекс.

**Общие сведения о программе**

Программа состоит из 6 файлов:

* main.cpp — основной исполняемый файл.
* lock\_free\_queue.hpp — файл с реализацией lock-free очереди.
* lock\_queue.hpp — файл с реализацией очереди с mutex.
* timer.hpp — вспомогательный файл с реализацией таймера для замера времени работы программы.
* sort.hpp — вспомогательный файл с реализацией пирамидальной сортировки, необходимой для lock-free очереди.
* binary\_search.hpp — вспомогательный файл с реализацией бинарного поиска, необходимого для lock-free очереди.

**Общий метод и алгоритм решения**

Неблокирующая синхронизация — это подход в параллельном программировании, в котором принят отказ от традиционных примитивов блокировки, таких, как семафоры, мьютексы.

Причина появления подобных алгоритмов — традиционный подход позволяет предоставить последовательный доступ при помощи такого механизма синхронизации, как блокировки, при таком подходе попытка одного из потоков получить блокировку, которая уже занята другим потоком, приводит к приостановке выполнения первого потока до момента освобождения блокировки во втором потоке, такая схема может привести к приостановке потоков на неопределенное время. Неблокирующие алгоритмы гарантируют, что такие остановки одного потока не приведут к простою остальных.