



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## ОТЧЕТ

по лабораторной работе № 4

по курсу «Анализ алгоритмов»

на тему: «Параллельные вычисления на основе нативных потоков»

Студент ИУ7-56Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Барсков А. Д.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Волкова Л. Л.  
(И. О. Фамилия)

2023 г.

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Аналитическая часть</b>	<b>4</b>
1.1 Многопоточность . . . . .	4
1.2 Нечеткая кластеризация с-средних . . . . .	4

# Введение

Многопоточное программ

Целью данной работы является изучение принципов организации параллельных вычислений. Для достижения цели потребуется выполнить следующие задачи:

1. Исследовать и описать основы параллельных вычислений.
2. Разработать программное обеспечение, которое реализует алгоритм нечёткой кластеризации С-средних.
3. Разработать многопоточную версию данного алгоритма.
4. Сделать анализ временных характеристик при разных количествах потоков.

# 1 Аналитическая часть

## 1.1 Многопоточность

Многопоточность - это форма распараллеливания или разделения работы для одновременной обработки [**intel-multithreading**]. Вместо того, чтобы нагружать одно ядро большой нагрузкой, многопоточные программы разделяют работу на несколько программных потоков. Поток - некая сущность внутри процесса, получающая процессорное время для выполнения [**thread**]. Процесс - это экземпляр выполняемой программы [**winForProf**].

Главное преимущество многопоточной обработки заключается в том, что она позволяет писать программы, которые работают очень эффективно благодаря возможности выгодно использовать время простоя, неизбежно возникающее в ходе выполнения большинства программ. Кроме того, параллельное выполнение нескольких потоков инструкций позволяет разделить выполнение трудоемких задач и уменьшить время их выполнения [**mult-in-apps**].

Стек у каждого потока разный, но память для программного кода и куча разделяются среди всех потоков, функционирующих внутри одного процесса. Это позволяет потокам внутри одного процесса быстро взаимодействовать между собой, так как они разделяют одну и ту же память. Без должного контроля доступа к общей памяти может возникнуть состояние гонки, когда два или более потока пытаются одновременно изменить общие данные, что может привести к непредсказуемым результатам и ошибкам.

Для предотвращения состояний гонки часто применяются мьютексы и семафоры:

- Определение Мьютекса.
- Определение семафора.

## 1.2 Нечеткая кластеризация с-средних