

Московский Авиационный Институт  
(Национальный Исследовательский Университет)  
Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу  
«Операционные системы»**

**Тема работы  
«Многопоточность»**

Студент: Лютоеv Илья Александрович  
Группа: М8О-207Б-21  
Вариант: 4  
Преподаватель: Миронов Евгений Сергеевич  
Оценка: \_\_\_\_\_  
Дата: \_\_\_\_\_  
Подпись: \_\_\_\_\_

Москва, 2022

## Задание

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы. Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы. В отчете привести исследование зависимости ускорения и эффективности алгоритма от входящих данных и количества потоков. Получившиеся результаты необходимо объяснить.

## Вариант 4

Отсортировать массив целых чисел при помощи TimSort ### Методы и алгоритмы решения TimSort является комбинацией сортировки вставкой и слиянием, для увеличения скорости работы распаралелим только сортировку вставкой. Мы можем это сделать так как сортировка вставкой производится на различных частях входного массива. Для распределения задач между потоками мы создали массив хранящий все ID наших потоков (число процессов вводится в качестве ключа программы при её запуске). ## Выполнение программы

```
lyutoev@fedora ~ ~/workshop/os/OS/lab3/cmake-build-debug ⌵ main ±+ ▮ ./main 4
<test1
```

minrun: 63

[illegible]

Time: 2687

## Оценка времени работы программы при разном количестве потоков

Для оценки времени работы я использовал утилиту time.  
Подадим на вход массив из миллиона элементов в случайном порядке.

Количество потоков	Время работы
1	0.48s
2	0.43s
3	0.36s
4	0.35s
5	0.37s
6	0.41s
7	0.41s
8	0.42s
9	0.39s
10	0.42s
11	0.45s
12	0.36s
13	0.39s
14	0.37s
15	0.45s
16	0.36s
17	0.41s
18	0.40s
19	0.39s
20	0.39s

Как мы видим после уже с тремя потоками время выполнения достигает своего минимума. Но стоит заметить, что использование потоков значительно снижает количество ## Вывод В ходе работы я научился работать с потоками и уменьшать время работы программы с их помощью.