

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Московский авиационный институт  
(Национальный исследовательский университет)»

Факультет информационных технологий и прикладной математики  
Кафедра вычислительной математики и программирования

ОТЧЁТ  
по лабораторной работе № 2  
по дисциплине «Операционные системы»

Студент: Князьков Николай Дмитриевич  
Группа: М80–203БВ–24  
Вариант: 13  
Преподаватель: Соколов А. А.

Москва, 2025

## Лабораторная работа № 2

### 1.1 Постановка задачи

Цель работы: приобретение практических навыков работы с потоками в операционной системе и синхронизации между ними.

Задание: написать многопоточное приложение на языке Си, применяющее свёртку (фильтр) к матрице вещественных чисел. Требуется накладывать фильтр заданного размера на входную матрицу и повторять данную операцию заданное количество раз. Общее количество потоков задаётся параметром командной строки. Необходимо исследовать зависимость ускорения и эффективности от числа потоков и размеров данных.

### 1.2 Общие сведения о программе

Программа реализует параллельную обработку двумерной матрицы вещественных чисел с использованием потоков. Основная функция создаёт заданное количество потоков, каждый из которых обрабатывает определённую часть матрицы. Для вычислений используется операция свёртки, при которой каждый элемент выходной матрицы вычисляется как сумма произведений элементов входной матрицы и ядра фильтра.

Для синхронизации потоков используется барьер, который гарантирует завершение вычислений всеми потоками перед началом следующей итерации. Операция свёртки может повторяться несколько раз, при этом результат предыдущей итерации используется как входные данные для следующей.

### 1.3 Метод и алгоритм решения

Матрица разбивается по строкам между потоками. Каждый поток обрабатывает свой диапазон строк, вычисляя значения выходной матрицы. После завершения обработки все потоки ожидают друг друга на барьере. После синхронизации данные обновляются, и при необходимости начинается следующая итерация свёртки.

Для оценки производительности измеряется время выполнения про-

граммы при различном количестве потоков. По результатам вычисляется ускорение и эффективность параллельного алгоритма.

## 1.4 Основные файлы программы с кодом

Основная реализация находится в файле `lab2.c`. Ниже приведён фрагмент кода, демонстрирующий создание потоков и выполнение свёртки.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <pthread.h>
4
5 void* thread_func(void* arg) {
6     int id = *(int*)arg;
7     /* */
8     pthread_exit(NULL);
9 }
10
11 int main(int argc, char* argv[]) {
12     int num_threads = 4;
13     pthread_t threads[num_threads];
14     int ids[num_threads];
15
16     for (int i = 0; i < num_threads; i++) {
17         ids[i] = i;
18         pthread_create(&threads[i], NULL, thread_func, &ids[i])
19     };
20
21     for (int i = 0; i < num_threads; i++) {
22         pthread_join(threads[i], NULL);
23     }
24
25     return 0;
26 }
```

Листинг 1.1: `lab2`

## 1.5 Пример работы

Пользователь запускает программу, задавая количество потоков и параметры матрицы. После выполнения вычислений программа выводит результат свёртки и измеренное время выполнения. При увеличении числа потоков наблюдается уменьшение времени обработки до достижения аппаратных ограничений.

## 1.6 Выводы

В ходе выполнения лабораторной работы был реализован многопоточный алгоритм свёртки матрицы вещественных чисел. Использование потоков позволило значительно сократить время выполнения вычислений. Проведённый анализ показал зависимость ускорения и эффективности от числа потоков. Полученные результаты подтверждают целесообразность применения параллельных вычислений для обработки больших объёмов данных.