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»

Студент ИУ7-52Б _____ Новиков А. А.
(Группа) (Подпись, дата) (И. О. Фамилия)

Преподаватель _____ Строганов Д. В.
(Подпись, дата) (И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Многопоточность — это способность центрального процессора или одного из ядер в многоядерной архитектуре одновременно выполнять несколько процессов или потоков, поддерживаемых операционной системой.

В общем случае поток исполнения представляет собой последовательность инструкций, выполняемых на выделенном процессорном ядре и управляемых планировщиком операционной системы. Потоки могут приостанавливаться или блокироваться во время выполнения. Они создаются внутри процесса и совместно используют его ресурсы, такие как оперативная память и дескрипторы файлов. Такой механизм называется нативными потоками. Нативные потоки позволяют эффективно использовать системные ресурсы и выполнять несколько задач параллельно в рамках одного процесса, что существенно повышает производительность приложений [1].

Цель лабораторной работы — сравнить основные принципы последовательных вычислений с параллельными на основе нативных потоков. Для достижения поставленной цели необходимо выполнить следующие задачи:

— описать входные, выходные данные;

— реализовать два алгоритма для загрузки контента из $HTML$—страниц: последовательный и параллельный с использованием нативных потоков;

— протестировать разработанные алгоритмы;

— сравнить скорость выполнения программы в зависимости от количества используемых потоков.

1 Входные и выходные данные

Входные данные: базовый URL—адрес веб-сайта, число обрабатываемых страниц, количество используемых потоков (1 - для последовательного режима).

Выходные данные: файлы, содержащие полученный контент с соответствующих страниц рецептов.

2 Тестирование

В таблице 2.1 представлены функциональные тесты для разработанного программного обеспечения. Все тесты пройдены успешно.

Таблица 2.1 – Описание тестовых случаев

№	Входные данные	Ожидаемый результат	Результат теста
1	Корректный URL, 1 поток (тестирование последовательного режима)	Директория с текстами рецептов `recipes`	Пройден
2	Корректный URL, 16 потоков (тестирование параллельного режима)	Директория с текстами рецептов `recipes`	Пройден
3	Пустой URL	Вывод сообщения об ошибке, повторный запрос ввода	Пройден
4	Некорректный URL	Вывод сообщения об ошибке, повторный запрос ввода	Пройден
5	Не положительное число страниц	Вывод сообщения об ошибке, повторный запрос ввода	Пройден
6	Не положительное число потоков	Вывод сообщения об ошибке, повторный запрос ввода	Пройден

3 Описание исследования

В ходе исследования требуется сравнить скорость выполнения программы в зависимости от количества используемых потоков. Замеры проводились при обработки 5 страниц, каждая из которых содержала 30 рецептов.

3.1 Технические характеристики

Технические характеристики, используемого устройства:

— операционная система — Ubuntu Linux x86_64 [2];

— память — 16 Гб;

— процессор — AMD Ryzen 5 5500U (6x2.10 ГГц) [3].

3.2 Полученные результаты

В таблице 3.1 приведено время выполнения программного обеспечения в миллисекундах (далее — мс). На рисунке 3.1 показана зависимость времени работы от количества потоков без изменений количества обрабатываемых страниц сайта.

Таблица 3.1 – Время работы от количества потоков (в миллисекундах)

Количество потоков	Время работы
1	125,979
2	69,736
4	35,228
8	17,600
16	9,346
32	4,577

В таблице 3.2 приведено время выполнения в зависимости от количества обрабатываемых страниц. В отличие от предыдущего исследования, в таблице 3.2 отсутствует зависимость от количества потоков.

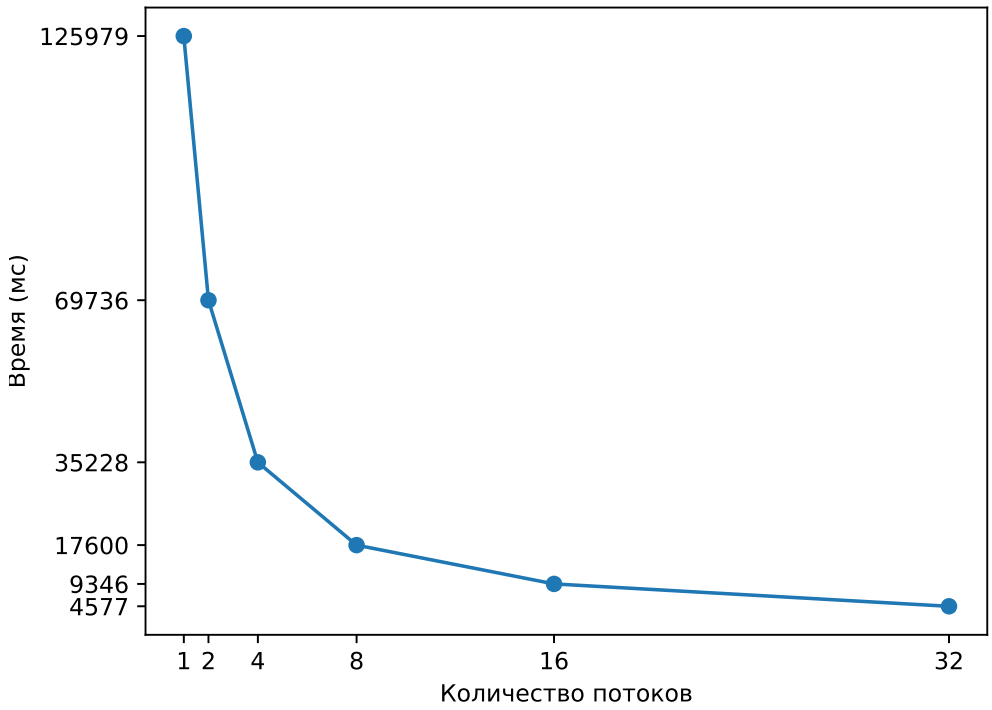

Рисунок 3.1 – Зависимость времени выполнения программы от количества потоков

Таблица 3.2 – Время работы программы в зависимости от количества обрабатываемых страниц (в мс)

Количество страниц	Время для реализации	
	1 поток	16 потоков
1	8,951	2,266
2	16,315	3,205
3	25,555	3,559
4	34,707	4,623
5	40,462	5,422

На рисунке 3.2 показана зависимость времени работы от количества обрабатываемых страниц.

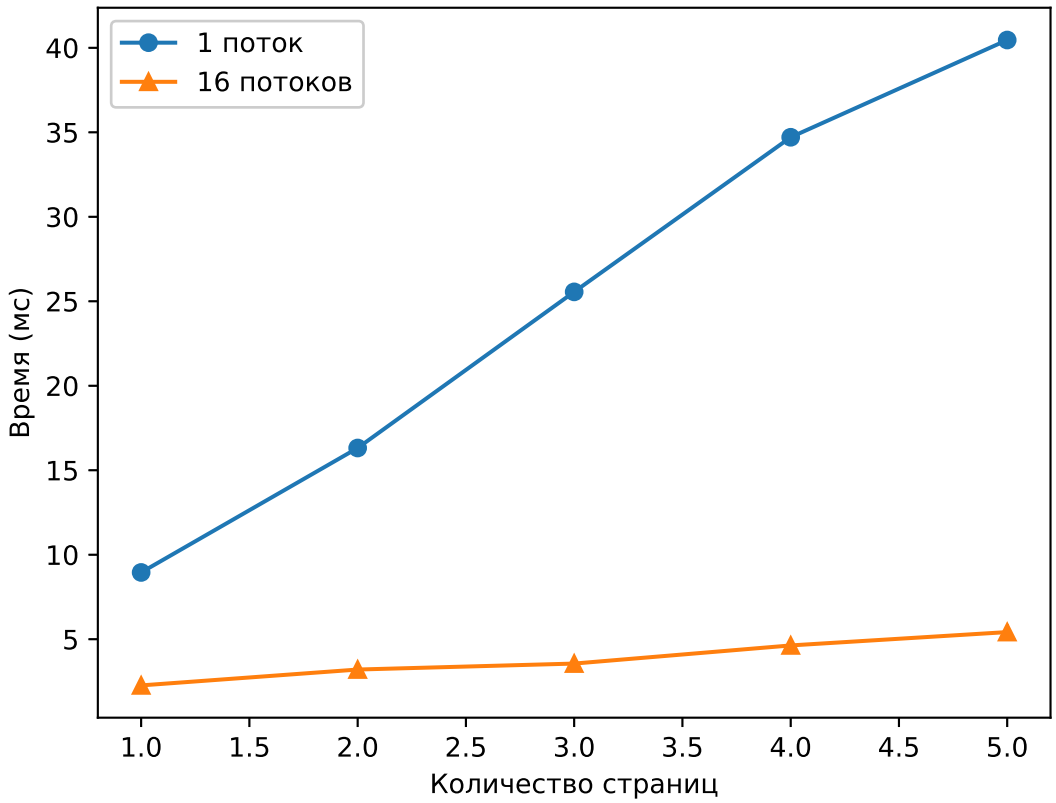

Рисунок 3.2 – Зависимость времени работы программы от числа страниц

3.3 Вывод

Исследование показало, что однопоточная реализация программы работает значительно медленнее по сравнению с многопоточной версией. При фиксированном количестве потоков и увеличении числа обрабатываемых страниц время выполнения однопоточной программы примерно в 8 раз больше, чем у 16-поточной версии.

При увеличении количества потоков и фиксированном числе обрабатываемых страниц время выполнения сокращается приблизительно в 2 раза при удвоении числа потоков.

ЗАКЛЮЧЕНИЕ

Цель работы достигнута: сравнены основные принципы последовательных вычислений с параллельными на основе нативных потоков. Было выявлено, что однопоточная реализация программы работает значительно медленнее по сравнению с многопоточной версией. При фиксированном количестве потоков и увеличении числа обрабатываемых страниц время выполнения однопоточной программы примерно в 8 раз больше, чем у 16-поточной версии. При увеличении количества потоков и фиксированном числе обрабатываемых страниц время выполнения сокращается приблизительно в 2 раза при удвоении числа потоков.

В ходе выполнения данной лабораторной работы были решены следующие задачи:

— описаны входные, выходные данные;

— реализованы два алгоритма для загрузки контента из $HTML$—страниц: последовательный и параллельный с использованием нативных потоков;

— протестированы разработанные алгоритмы;

— выполнено сравнение скорости выполнения программы в зависимости от количества используемых потоков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Потоки. Документация IBM. [Электронный ресурс]. — URL: `https://www.ibm.com/docs/ru/informix-servers/12.10?topic=processors-threads` (дата обращения 25.10.2024).
2. Ubuntu technical documentation for developers and IT pros [Электронный ресурс]. — URL: `https://ubuntu.com/tutorials` (дата обращения 25.10.2024).
3. AMD Ryzen 5 5500U Processor [Электронный ресурс]. — URL: `https://www.amd.com/en/products/apu/amd-ryzen-5-5500u` (дата обращения 25.10.2024).