



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»

Лабораторная работа №4 по дисциплине «Анализ Алгоритмов»

Тема Параллельные вычисления на основе нативных потоков

Студент Куликов Е. А.

Группа ИУ7-56Б

Преподаватель Волкова Л. Л.

СОДЕРЖАНИЕ

| | |
|--|----------|
| ВВЕДЕНИЕ | 2 |
| 1 Входные и выходные данные | 2 |
| 2 Преобразование входных данных в выходные | 2 |
| 3 Примеры работы программы | 3 |
| 4 Тестирование | 4 |
| 5 Описание исследования | 5 |
| ЗАКЛЮЧЕНИЕ | 7 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 8 |

ВВЕДЕНИЕ

Цель работы — получение навыка организации параллельных вычислений на основе нативных потоков.

Задачи работы:

- анализ предметной области;
- разработка алгоритма обработки данных;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО.

1 Входные и выходные данные

Входными данными программы является ссылка на сайт с кулинарными рецептами, в данном варианте — на сайт <https://www.kuhnyatv.ru/recipes>, название папки, в которую нужно сохранять рецепты, количество рецептов, которые нужно сохранить и количество потоков которые нужно создать. Выходными данными программы является папка с заданным именем, внутри которой располагается заданное количество рецептов в сормате html-страниц.

2 Преобразование входных данных в выходные

Программа считывает указанные входные данные и в зависимости от указанного количества потоков выполняет поиск страниц с рецептами в последовательном режиме (при количестве потоков равном 0) или параллельном режиме, создавая указанное количество потоков.

Каждый поток (в последовательном режиме — главный поток, в параллельном — каждый из созданных потоков) выполняет поиск первой необработанной ссылки в общем для всех потоков множестве, помечает эту ссылку обработанной, обращается к сайту по этой ссылке (для этого используется библиотека `crg` [1]), получая html-страницу. На этой странице производится поиск всех ссылок (с помощью регулярных выражений библиотеки `regex` [2]), каждая из которых, если уже не находится в общем множестве, помещается в

него и помечается необработанной. Далее проверяется, является ли страница рецептом с помощью поиска на странице ключевых слов — "Продукты" и "Способ приготовления". Если страница является рецептом, ее исходный текст помещается в html-файл в указанную во входных данных папку и общий для всех потоков счетчик оставшихся страниц декрементируется. При равенстве счетчика нулю поиск останавливается.

3 Примеры работы программы

На рисунках 3.1- 3.2 представлен пример работы программы.

```
1) Enter dir name, recipes amount and threads amount, get recipes dir
2) Time measures
Enter option:
1
Enter directory name:
recipes
Enter recipes amount:
10
Enter threads amount:
0
Processing...
```

Рисунок 3.1 – Ввод входных данных

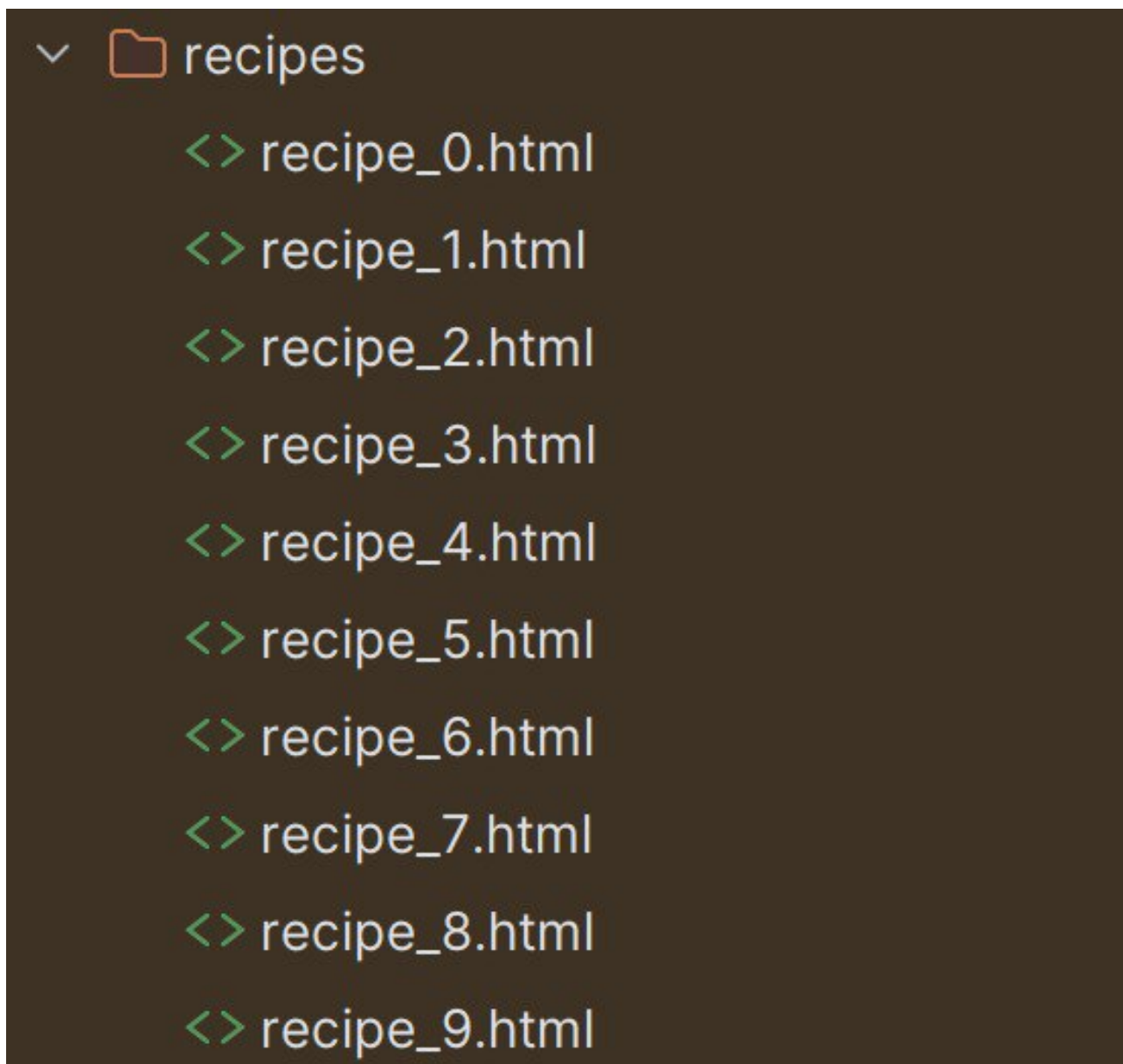


Рисунок 3.2 – Полученная папка с рецептами

4 Тестирование

В таблице 4.1 представлены функциональные тесты для разработанного ПО. Все тесты пройдены успешно. Порядок ввода входных данных: имя папки, количество рецептов, количество потоков.

Таблица 4.1 – Функциональные тесты

| № теста | Входные данные | Полученные выходные данные | Ожидаемые выходные данные |
|---------|----------------|----------------------------|---------------------------|
| 1 | recipes 10 0 | recipes/ с 10 рецептами | recipes/ с 10 рецептами |

Таблица 4.1 – Функциональные тесты (продолжение)

| № теста | Входные данные | Полученные выходные данные | Ожидаемые выходные данные |
|---------|----------------|----------------------------|---------------------------|
| 2 | recipes 10 4 | recipes/ с 10 рецептами | recipes/ с 10 рецептами |
| 3 | recipes 10 64 | recipes/ с 10 рецептами | recipes/ с 10 рецептами |

5 Описание исследования

В ходе исследования требуется исследовать зависимость производительности разработанного ПО (в терминах количества обработанных страниц в единицу времени) от количества дополнительных потоков. Изменять количество дополнительных потоков от 0 (вычисление в основном потоке), до $4 \cdot k$, где k — количество логических ядер используемой ЭВМ, по степеням числа 2. Количество логических ядер процессора, на котором проводилось исследование (13th Gen Intel(R) Core(TM) i5-13500H 2.60 GHz) равно 16, поэтому замеры проводились на интервале от 0 до 64 потоков.

В таблице 5.1 приведены замеры зависимости количества секунд, потраченных на один рецепт от количества потоков, на рисунке 5.1 приведен график, построенный по таблице.

Таблица 5.1 – Таблица замеров времени обработки одной страницы от количества потоков

| Количество потоков | Количество полученных за секунду рецептов |
|--------------------|---|
| 0 | 0.677015 |
| 1 | 0.689014 |
| 2 | 0.675797 |
| 4 | 0.618141 |
| 8 | 0.586283 |
| 16 | 0.564012 |
| 32 | 0.585115 |
| 64 | 0.602166 |

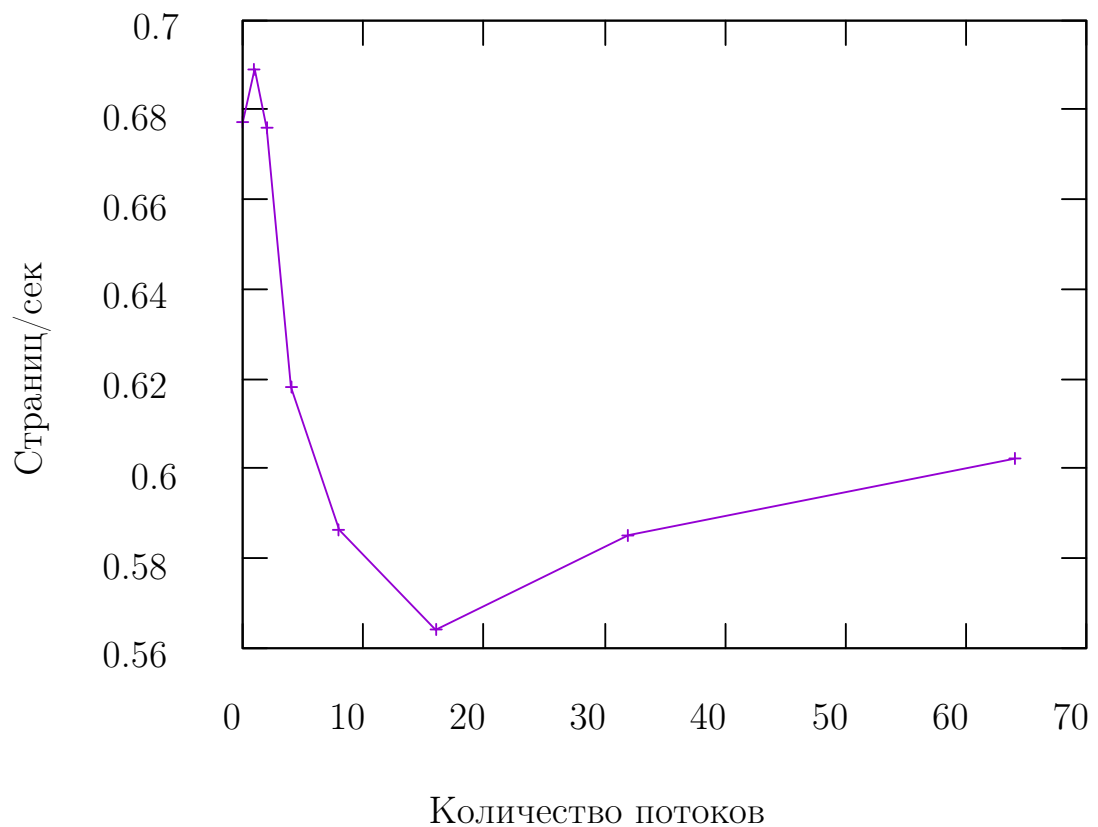


Рисунок 5.1 – График зависимости времени на один рецепт от количества потоков

По результатам проведенного исследования сделан вывод о том, что параллельная обработка данных может давать прирост к производительности, но только когда количество потоков не больше чем количество логических ядер процессора. График показывает что при большем количестве потоков степень время выполнения увеличивается, так как степень количество параллельных вычислений не изменяется, а количество издержек на создание потоков увеличивается.

ЗАКЛЮЧЕНИЕ

Цель работы достигнута. Решены все поставленные задачи:

- анализ предметной области;
- разработка алгоритма обработки данных;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация библиотеки `std::string` [Электронный ресурс]. — Режим доступа: <https://docs.libcpr.org/> (дата обращения: 29.10.2024).
2. Документация библиотеки `std::regex` [Электронный ресурс]. — Режим доступа: <https://learn.microsoft.com/ru-ru/cpp/standard-library/regular-expressions-cpp?view=msvc-170> (дата обращения: 29.10.2024).