



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»

Отчёт по лабораторной работе № 4 по курсу «Анализ алгоритмов»

Тема Параллельные вычисления на основе нативных потоков

Студент Гаврилюк В. А.

Группа ИУ7-51Б

Оценка (баллы)

Преподаватель Волкова Л. Л.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
1 Входные и выходные данные	3
1.1 Входные данные	3
1.2 Выходные данные	3
2 Преобразование данных	4
3 Пример работы программы	5
4 Тестирование	7
5 Исследование	8
5.1 Технические характеристики	8
5.2 Проведение исследования	8
ЗАКЛЮЧЕНИЕ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

ВВЕДЕНИЕ

При использовании нескольких потоков управления можно спроектировать приложение, которое будет решать одновременно несколько задач в рамках единственного процесса, где каждый поток решает отдельную задачу [1]. Одно из преимуществ такого подхода заключается в том, что решение некоторых задач можно разбить на более мелкие подзадачи, что может дать прирост производительности программы [1].

Однопоточный процесс, выполняющий решение нескольких задач, неявно вынужден решать их последовательно, поскольку имеет только один поток управления. При наличии нескольких потоков управления независимые друг от друга задачи могут решаться одновременно отдельными потоками. Две задачи могут решаться одновременно только при условии, что они не зависят друг от друга [1].

Цель работы – Получить навык организации параллельных вычислений на основе нативных потоков.

Для достижения цели необходимо выполнить следующие задачи:

- описать входные и выходные данные программы;
- разработать программное обеспечение (ПО), которое осуществляет выгрузку данных со страниц указанного интернет-ресурса;
- провести исследование скорости скачивания страниц в зависимости от количества созданных потоков.

1 Входные и выходные данные

1.1 Входные данные

Программа принимает на вход следующие данные:

- адрес главной страницы ресурса;
- максимальное количество страниц, с которых выгружаются данные;
- максимальное количество потоков;
- путь до директории хранения выгруженных файлов.

1.2 Выходные данные

Выходным данным программы является директория с файлами, которые содержат скачанные данные со страниц в формате HTML.

2 Преобразование данных

Процесс преобразования входных данных в выходные происходит следующим образом:

- программа получает на вход адрес главной страницы ресурса, максимальное количество страниц, с которых выгружаются данные, максимальное количество потоков и путь к директории для хранения выгруженных файлов;
- если директории не существует, то создаётся новая директория по указанному пути;
- в каждом потоке выполняется проверка, была ли обрабатываемая в данный момент страница уже посещена;
- если эта страница не обрабатывалась ранее, её содержимое выгружается и сохраняется в файл в формате HTML в указанную директорию;
- анализ содержимого выгруженной страницы и добавление ссылок в множество необработанных страниц.

3 Пример работы программы

На листинге 3.1 изображён пример запуска программы в терминале. При запуске в аргументах командой строки указываются входные параметры. Все аргументы являются опциональными, т. е. если какой-либо аргумент опущен, то его значение будет стандартным (значением по умолчанию) для данного параметра. Список значений по умолчанию:

- 1 для `workers` (максимальное количество потоков);
- 10 для `pages` (максимальное количество страниц);
- `https://edimdoma.ru` для `url` (адрес главной страницы ресурса);
- `../data` для `dir` (путь до директории хранения выгруженных файлов).

Листинг 3.1 – Пример запуска программы

```
$ go run ./cmd/main.go -workers=10 -pages=10
  -url=https://edimdoma.ru -dir=../data

[PARAMETERS]
- baseUrl: https://edimdoma.ru
- dirPath: ../data
- maxPages: 10
- maxWorkers 10

[RESULT]
- visited: 10
```

На рисунке 3.1 показан фрагмент реальной страницы ресурса, а на рисунке 3.2 фрагмент загруженного HTML файла. Так как внутри HTML страницы интернет-ресурса `https://edimdoma.ru` располагались элементы каскадной таблицы стилей (CSS), то многие элементы внешнего оформления страницы остались неизменными.

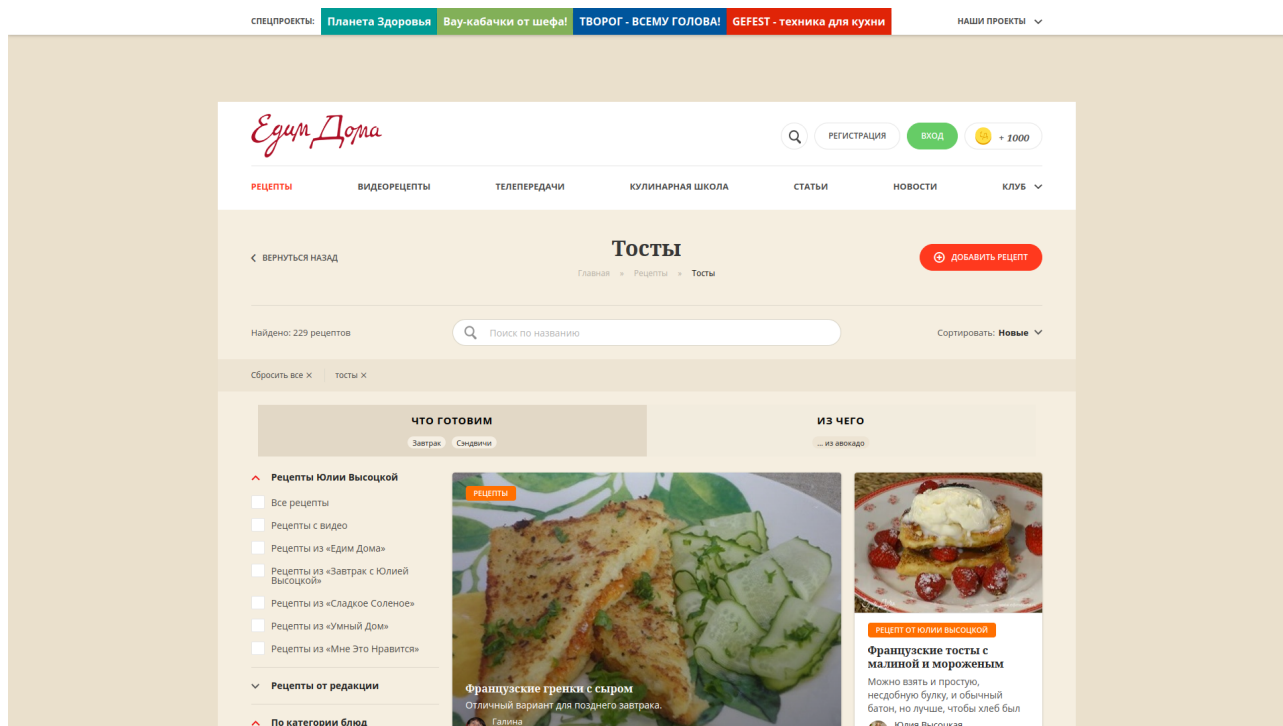


Рисунок 3.1 – Фрагмент страницы интернет-ресурса

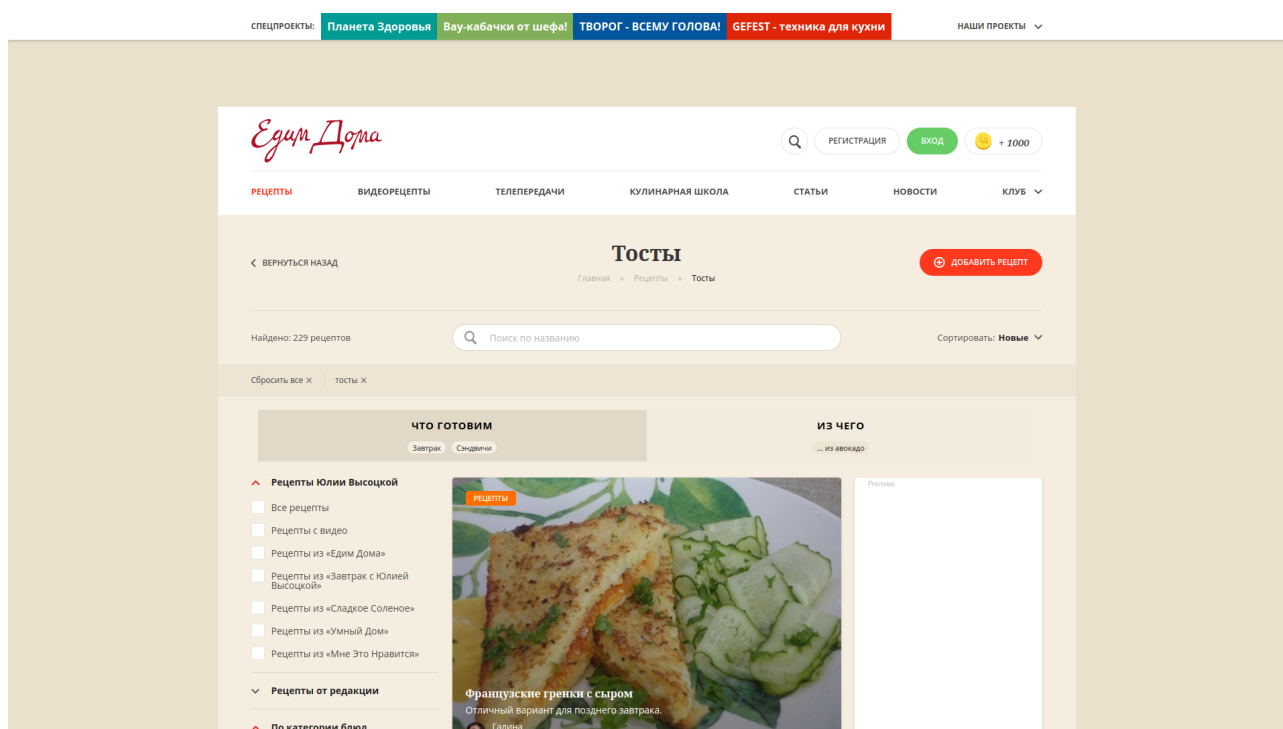


Рисунок 3.2 – Фрагмент загруженной HTML страницы

4 Тестирование

В таблице 4.1 представлены результаты функционального тестирования программы. Все тесты пройдены успешно.

Таблица 4.1 – Результаты функционального тестирования программы

Количество страниц	Количество потоков	Ожидаемый результат (кол-во загруженных страниц)	Полученный результат (кол-во загруженных страниц)
1	1	1	1
1	12	1	1
1	32	1	1
100	1	100	100
100	12	100	100
100	32	100	100
500	1	500	500
500	12	500	500
500	32	500	500

5 Исследование

5.1 Технические характеристики

Исследование проводилось на ЭВМ со следующими характеристиками:

- операционная система Ubuntu 22.04.5 LTS;
- объем оперативной памяти 16 ГБ;
- процессор Intel Core i7-8700K CPU 3.70GHz × 12 [2].

5.2 Проведение исследования

Было проведено исследование зависимости скорости загрузки данных со страниц интернет-ресурса от количества потоков, внутри которых происходит обработка и загрузка страниц. Максимальное количество выгружаемых страниц при каждом замере равно 85. В таблице 5.1 приведены результаты измерений.

Таблица 5.1 – Результаты измерений

Количество потоков (шт)	Скорость загрузки (кол-во страниц/сек)
1	1.10
2	2.21
4	3.43
8	5.10
16	6.31
32	7.64
48	6.92

На рисунках 5.1 — 5.2 продемонстрированы гистограммы, построенные на основе табличных данных.

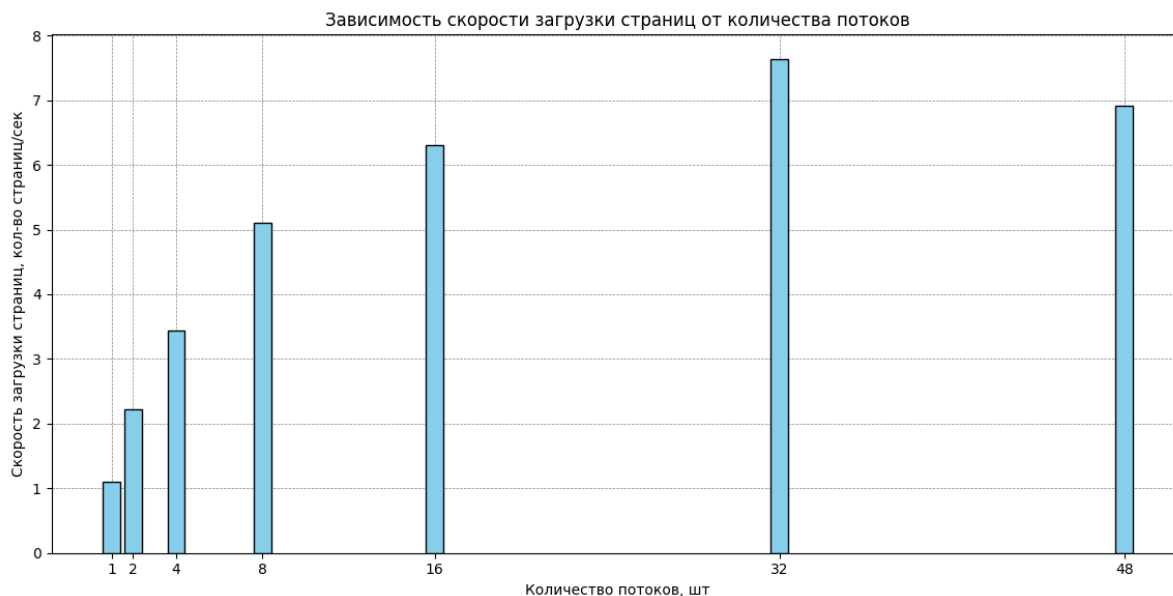


Рисунок 5.1 – Зависимость скорости загрузки страниц от количества потоков обработки (линейная шкала)

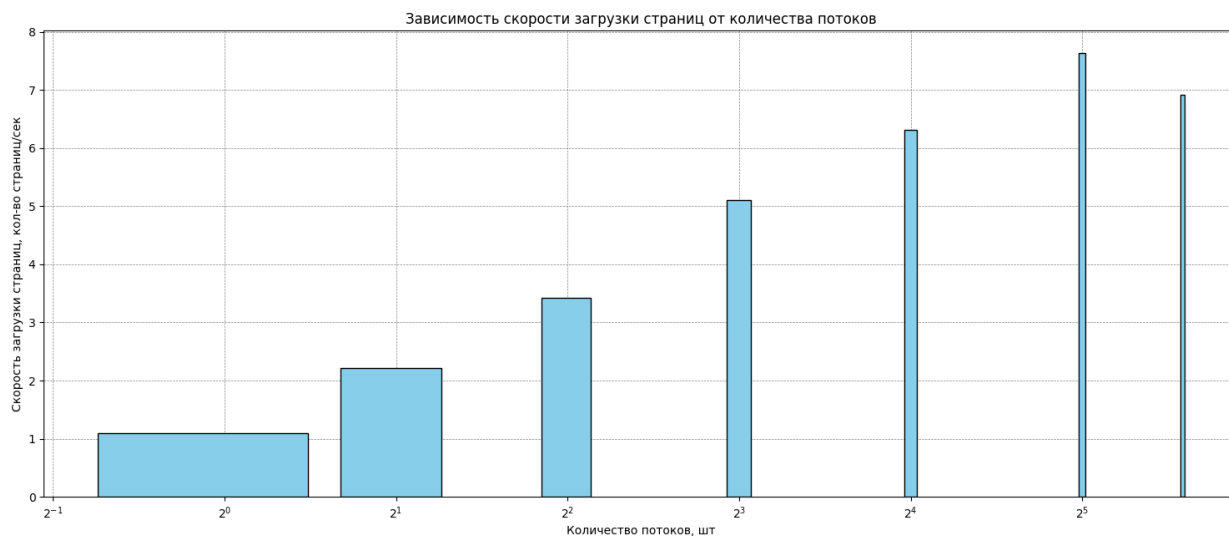


Рисунок 5.2 – Зависимость скорости загрузки страниц от количества потоков обработки (логарифмическая шкала)

Скорость загрузки страниц при использовании 48 потоков уменьшилась на 9.4% в сравнении со скоростью при 32 потоках. Такой результат может быть связан с накладными расходами на создание, управление потоками и переключение контекста выполнения. При этом результаты измерений показали, что скорость загрузки страниц при 32 потоках в 6.9 раз выше, чем при использовании одного.

ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы было разработано ПО, выполняющее выгрузку HTML страниц с интернет-ресурса в многопоточном режиме. Было проведено исследование зависимости скорости загрузки страниц от количества потоков обработки, которое показало, что при использовании 32 потоков скорость загрузки страниц в 6.9 раз выше, чем при использовании одного потока. Однако при увеличении числа потоков до 48 скорость загрузки снизилась на 9.4% по сравнению с 32 потоками. Такой результат может быть связан с накладными расходами на создание, управление потоками и переключение контекста выполнения.

Все задачи выполнены, цель лабораторной работы достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Стивенс У. Р., Раго С. А.* UNIX. Профессиональное программирование. — 3-е изд. — СПб. : Питер, 2018. — (Для профессионалов).
2. Intel Core i7-8700K Processor [Электронный ресурс]. — Режим доступа: <https://ark.intel.com/content/www/us/en/ark/products/126775/intel-core-i78700k-processor-12m-cache-up-to-4-70-ghz.html>).