



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»

на тему: «Параллельные вычисления на основе нативных потоков»

Вариант № 9113

Студент ИУ7-56Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Александрова А. Д.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Кормановский М. В.  
(И. О. Фамилия)

2024 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Входные и выходные данные</b>	<b>3</b>
<b>2 Преобразование входных данных в выходные</b>	<b>3</b>
<b>3 Примеры работы программы</b>	<b>4</b>
<b>4 Тестирование</b>	<b>4</b>
<b>5 Описание исследования</b>	<b>5</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>6</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>7</b>

# ВВЕДЕНИЕ

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

Задачи работы:

- анализ возможностей организации параллельных вычислений на основе нативных потоков;
- разработка алгоритма, осуществляющего параллельную выгрузку страниц интернет ресурса с использованием нативных потоков;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО в зависимости от количества используемых потоков.

## 1 Входные и выходные данные

Входными данными являются ссылка на интернет ресурс, содержащий кулинарные рецепты, номер команды меню, которую необходимо выполнить, количество рецептов, которые необходимо сохранить, и количество потоков. Выходными данными является набор файлов, содержащий необходимое количество сохраненных рецептов.

## 2 Преобразование входных данных в выходные

Программа считывает входные данные и выполняет поиск необходимого количества ссылок на страницы интернет ресурса, содержащих кулинарные рецепты, при помощи библиотеки `curl` [1]. Далее загружает в память каждую полученную страницу и при помощи регулярных выражений [2] выполняет на ней поиск рецепта, который в свою очередь сохраняет в текстовом файле. При этом в зависимости от введенной команды меню будут использованы либо один, либо несколько потоков.

### 3 Примеры работы программы

На рисунке 3.1 представлен пример работы программы.

```
Menu:
0. Exit.
1. Regular download.
2. Multithreaded download.
3. Bench.
Select an option: 2
Enter a number of recipes to download: 3
Enter a number of additional threads: 3
```

Рисунок 3.1 – Пример работы программы

### 4 Тестирование

В таблице 4.1 представлены функциональные тесты для разработанного ПО. Все тесты пройдены успешно.

Таблица 4.1 – Функциональные тесты

№ теста	Входные данные	Полученные выходные данные	Ожидаемые выходные данные
1	1 1	1 файл с выгруженным рецептом	1 файл с выгруженным рецептом
2	1 10	10 файлов с выгруженными рецептами	10 файлов с выгруженными рецептами
3	1 10 8	10 файлов с выгруженными рецептами	10 файлов с выгруженными рецептами

## 5 Описание исследования

Необходимо исследовать, как производительность разработанного программного обеспечения (в терминах количества обработанных страниц за единицу времени) зависит от числа дополнительных потоков. Измеряется количество дополнительных потоков от 0 (вычисление в основном потоке), до  $4 \cdot k$ , где  $k$  — количество логических ядер используемой ЭВМ, по степеням числа 2. В данном случае — 0, 1, 2, 4, 8, 16, 32, так как количество логических ядер процессора (Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz), на котором выполняется исследование, равно 8.

Результаты замеров представлены на рисунке 5.1.

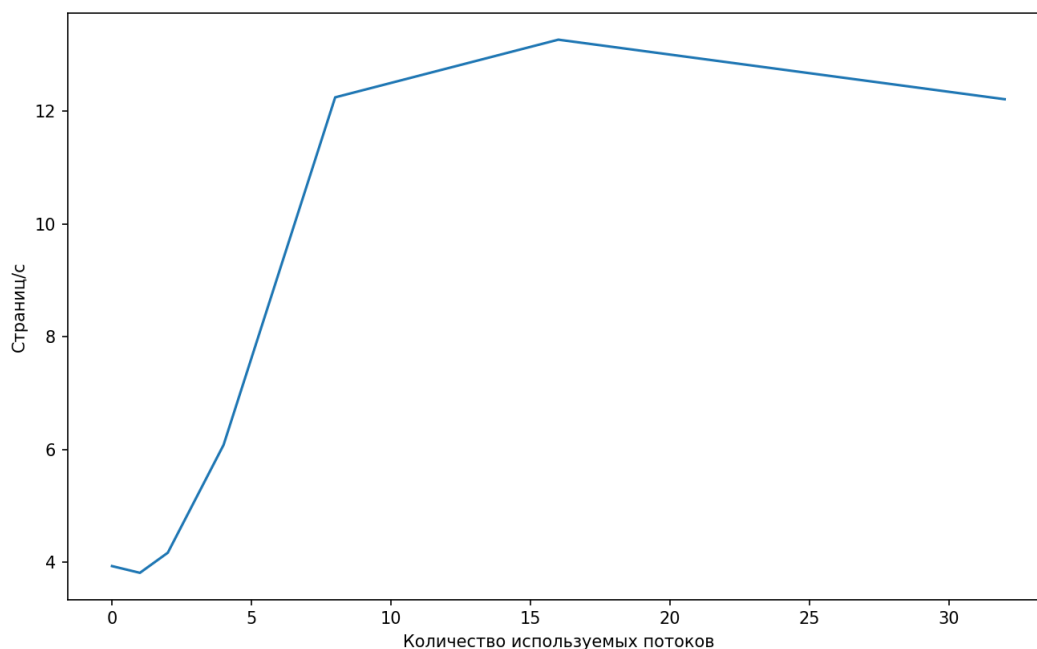


Рисунок 5.1 – График зависимости количества обработанных страниц в секунду от количества используемых дополнительных потоков

По результатам проведенных замеров сделан вывод, что использование дополнительных потоков дает значительное увеличение производительности программы. При этом использование только одного дополнительного потока не дает преимущества, так как накладные расходы на создание потока выше, чем выигрыш по времени за счет его использования. Также при увеличивающемся количестве потоков после определенного значения (в данном случае — 32), производительность снова уменьшается из-за затрат, необходимых для создания такого большого количества потоков.

## ЗАКЛЮЧЕНИЕ

Цель работы достигнута. Решены все поставленные задачи:

- анализ возможностей организации параллельных вычислений на основе нативных потоков;
- разработка алгоритма, осуществляющего параллельную выгрузку страниц интернет ресурса с использованием нативных потоков;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО в зависимости от количества используемых потоков.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация библиотеки curl [Электронный ресурс]. — — Режим доступа: <https://curl.se/libcurl/c/libcurl.html> (Дата обращения: 2024-11-27).
2. Документация по регулярным выражениям [Электронный ресурс]. — — Режим доступа: <https://www.ibm.com/docs/en/zos/3.1.0?topic=documentation-regular-expressions> (Дата обращения: 2024-11-29).