



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»

Студент ИУ7-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Паламарчук А. Н.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Кормановский М. В.  
(И. О. Фамилия)

2024 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
1 Входные и выходные данные	3
2 Преобразование входных данных в выходные	3
3 Примеры работы программы	5
4 Тестирование	5
5 Описание исследования	6
<b>ЗАКЛЮЧЕНИЕ</b>	<b>7</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>8</b>

# ВВЕДЕНИЕ

Параллельные вычисления – это использование нескольких или многих вычислительных устройств для одновременного выполнения разных частей одной программы [1].

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

Задачи работы:

- анализ структуры сайта;
- разработка алгоритма обработки данных;
- разработка программного обеспечения, скачивающего html страницы в однопоточном и многопоточном режиме;
- исследование времени выгрузки html страниц от числа обрабатывающих потоков;

## 1 Входные и выходные данные

Входным данным является ссылка на сайт <https://www.gotovim.ru>, который содержит множество рецептов различных блюд. Выходными данными являются html файлы — скачанные страницы рецептов.

## 2 Преобразование входных данных в выходные

Программа отправляет запрос по исходной ссылке, разбирает полученную страницу и извлекает ссылки на категории блюд. Отправляются запросы по ссылкам на категории, из полученных страниц извлекаются ссылки на конкретные рецепты. Далее загружаются html страницы рецептов в директорию /data.

Листинг 2.1 – Функции извлечения ссылок на рецепты

```
void extract_urls(const std::string& page_content,
    std::set<std::string>& urls, const std::string& mask) {
    std::regex link_regex(mask);
```

```

std::sregex_iterator links_begin =
    std::sregex_iterator(page_content.begin(),
        page_content.end(), link_regex);
std::sregex_iterator links_end = std::sregex_iterator();
std::string new_link;
for (std::sregex_iterator i = links_begin; i != links_end;
    ++i) {
    new_link = ROOT_URL + i->str();
    urls.insert(new_link);
}
}

int get_category_urls(std::set<std::string>& category_urls,
    const std::string& url) {
    std::ostringstream readBuffer;
    CURL* curl = curl_easy_init();
    if (!curl) {
        std::cerr << "Ошибка инициализации CURL" << std::endl;
        return ERROR_INIT_CURL;
    }
    CURLcode res = perform_curl_request(curl, url, readBuffer);
    curl_easy_cleanup(curl);

    if (res == CURLE_OK)
        extract_urls(readBuffer.str(), category_urls,
            R"(\recepts\[a-z]+\[/[a-z]+\])");
    else {
        std::cerr << "Ошибка CURL: " << curl_easy_strerror(res)
            << std::endl;
        return ERROR_CURL;
    }

    return EXIT_SUCCESS;
}

int get_recepts_urls(std::set<std::string>& recept_urls, const
    std::string& url) {
    std::ostringstream readBuffer;
    CURL* curl = curl_easy_init();
    if (!curl) {
        std::cerr << "Ошибка инициализации CURL" << std::endl;

```

```

        return ERROR_INIT_CURL;
    }
    CURLcode res = perform_curl_request(curl, url, readBuffer);
    curl_easy_cleanup(curl);

    if (res == CURLE_OK)
        extract_urls(readBuffer.str(), recept_urls,
            R"(\sbs\[a-z]+\shtml)");
    else {
        std::cerr << "Ошибка CURL: " << curl_easy_strerror(res)
            << std::endl;
        return ERROR_CURL;
    }

    return EXIT_SUCCESS;
}

```

### 3 Примеры работы программы

На рисунке 3.1 представлен пример работы программы.

```

URLs прочитаны из файла

Команды:
1 - Задать максимальное количество страниц
2 - Выгрузка в последовательном режиме
3 - Выгрузка в параллельном режиме
4 - Построить график производительности от количества дополнительных потоков
0 - Завершить работу программы
>2
Введите кол-во выгружаемых страниц: 5
1) https://www.gotovim.ru/sbs/angliskeks.shtml download
2) https://www.gotovim.ru/sbs/apelsinkvas.shtml download
3) https://www.gotovim.ru/sbs/apelsinovliker.shtml download
4) https://www.gotovim.ru/sbs/apelsmuss.shtml download
5) https://www.gotovim.ru/sbs/apelsoknazimu.shtml download
Threads: 0, time: 6439 milliseconds

```

Рисунок 3.1 – Пример работы программы

### 4 Тестирование

Тестирование программы проводилось загрузкой 50 страниц по ссылкам на различные рецепты. Проверялось количество удачных загрузок и соответствие выходных файлов исходным html страницам.

Как для однопоточной реализации, так и для многопоточной тестировалось соответствие исходного перечня страниц и загруженных файлов, если все страницы были успешно загружены, тест считался пройденным, в обеих реализациях файлы были получены идентичные. Тестирование было пройдено успешно.

## 5 Описание исследования

Технические характеристики устройства:

- операционная система Manjaro Linux x86\_64;
- процессор Ryzen 5500U 6 ядер, тактовая частота 2.1 ГГц;
- оперативная память 16 Гбайт.

В ходе исследования был получен график зависимости времени выполнения от количества задействованных потоков от 0 (вычисление в основном потоке), до  $4 \cdot k$ , где  $k = 12$  — количество логических ядер.

Замер проводился 10 раз для каждого числа потоков, загружалось 50 ссылок и вычислялось среднее арифметическое время работы. На рисунке 5.1 представлен график полученной зависимости. Время указано в миллисекундах, количество потоков в штуках.

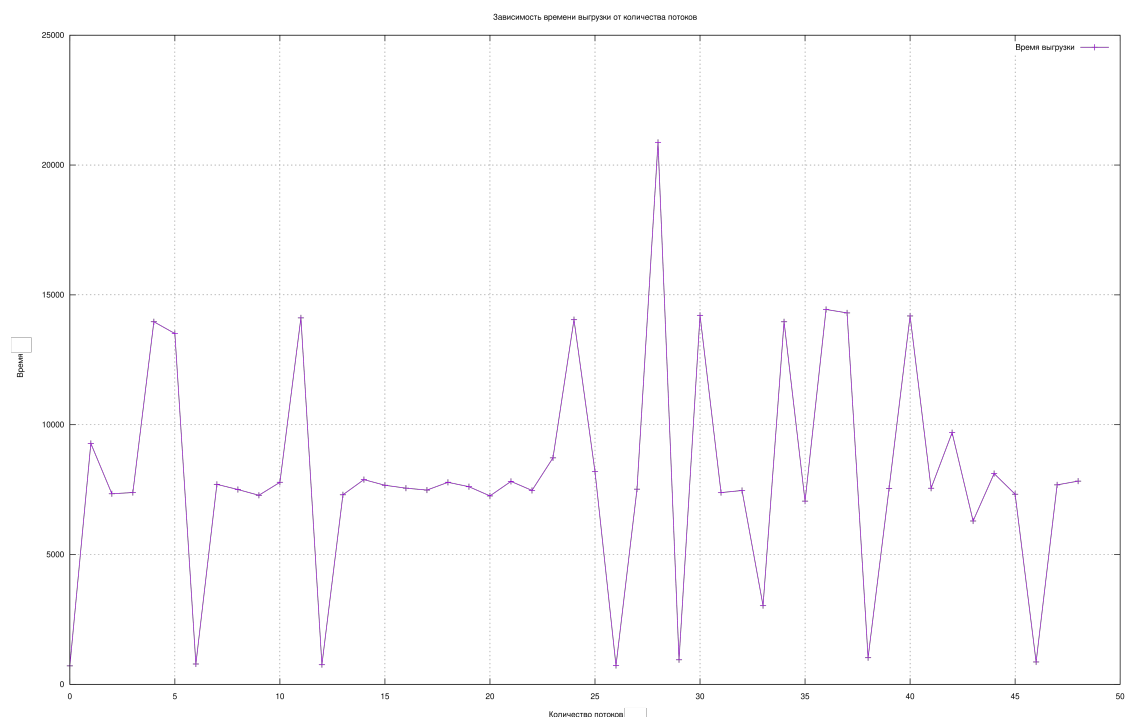


Рисунок 5.1 – График зависимости время выполнения от количества потоков

По результатам проведенного исследования сделан вывод о том, что время загрузки html страниц зависит не только от количества потоков, которые выполняют задачу, но и от скорости работы/загруженности сервера.

## ЗАКЛЮЧЕНИЕ

Получены навыки организации параллельных вычислений на основе нативных потоков. Цель работы достигнута. Решены все поставленные задачи:

- анализ структуры сайта;
- разработка алгоритма обработки данных;
- разработка программного обеспечения, скачивающего html страницы в однопоточном и многопоточном режиме;
- исследование времени выгрузки html страниц от числа обрабатывающих потоков;

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Гафаров Ф. М., Галимянов А. Ф.* Параллельные вычисления: учеб. пособие. — Казань : Изд-во Казан. ун-та, 2018. — С. 149.