



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 Тузов Даниил Александрович

Группа ИУ7-52Б

Преподаватель Строганов Дмитрий Владимирович

Москва, 2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Входные и выходные данные	4
2 Преобразование входных данных в выходные	5
3 Тестирование	6
4 Примеры работы программы	7
5 Описание исследования	8
ЗАКЛЮЧЕНИЕ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

ВВЕДЕНИЕ

В 4 лабораторной работе рассматривается тема параллельных вычислений на основе нативных потоков.

Параллелизм описывает последовательности действий, которые происходят одновременно^[1]. Нередко в современных системах используется распараллеливание вычислений, которое может привести к росту временной эффективности программы.

Целью работы является разработка ПО, выполняющего скачивание страниц, содержащих рецепты с сайта^[4]. Для достижения поставленной цели необходимо решить следующие задачи:

- рассмотреть структуру сайта;
- разработать ПО, выполняющее скачивание страниц в однопоточном режиме;
- разработать ПО, выполняющее скачивание страниц в многопоточном режиме;
- провести исследование временных характеристик, написанных программ;
- обосновать полученные результаты.

1 Входные и выходные данные

Входными данными в этой программе является страница электронного ресурса, с которого необходимо скачать страницы, и максимальное количество страниц, которые необходимо скачать. Выходными данными является директория с файлами, содержащими html код страниц.

2 Преобразование входных данных в выходные

Программа получает на вход ссылку на электронный ресурс и максимальное количество страниц выполняет код , указанный в листинге 1:

Листинг 1 – Скрипт на языке Python^[3] для получения файла ссылок для дальнейшей обработки

```
import requests as re
import bs4

filename = "urls.txt"
url = "https://www.povareschka.ru"
list_catalogs = set()
list_links = set()

page = re.get(url)
bs = bs4.BeautifulSoup(page.content, 'html.parser')
for groups in bs.find_all('menu')[1:-2]:
    for link in groups.find_all('a'):
        ref = link.get('href')
        if 'recepty' in ref:
            list_catalogs.add(url + ref)

for part in list_catalogs:
    page = re.get(part)
    bs = bs4.BeautifulSoup(page.content, 'html.parser')
    for groups in bs.find_all('main'):
        for link in groups.find_all('a'):
            ref = link.get('href')
            if 'recepty' in ref:
                list_links.add(url + ref)

f = open(filename, "w")
for i in list_links:
    f.write(i + '\n')
```

В результате работы формируется файл, содержащий в каждой строке ссылку на страницу с рецептом. В дальнейшем этот файл подается на вход программе на языке C++, которая выполняет сохранение страниц в файлы и формирует итоговую директорию.

3 Тестирование

При тестировании программы на вход подавался файл с единственной ссылкой на рецепт. В результате работы формировалась директория с единственным файлом, содержащим html код заданной страницы. На следующем этапе выполнялось сравнение кода элемента искомой страницы с кодом в полученном файле.

Тестирование выполнялось как для однопоточной программы, так и для многопоточной.

В многопоточной программе так же учитывалось количество создаваемых потоков.

4 Примеры работы программы

На рисунках 1 - 2 приведен пример работы программы.

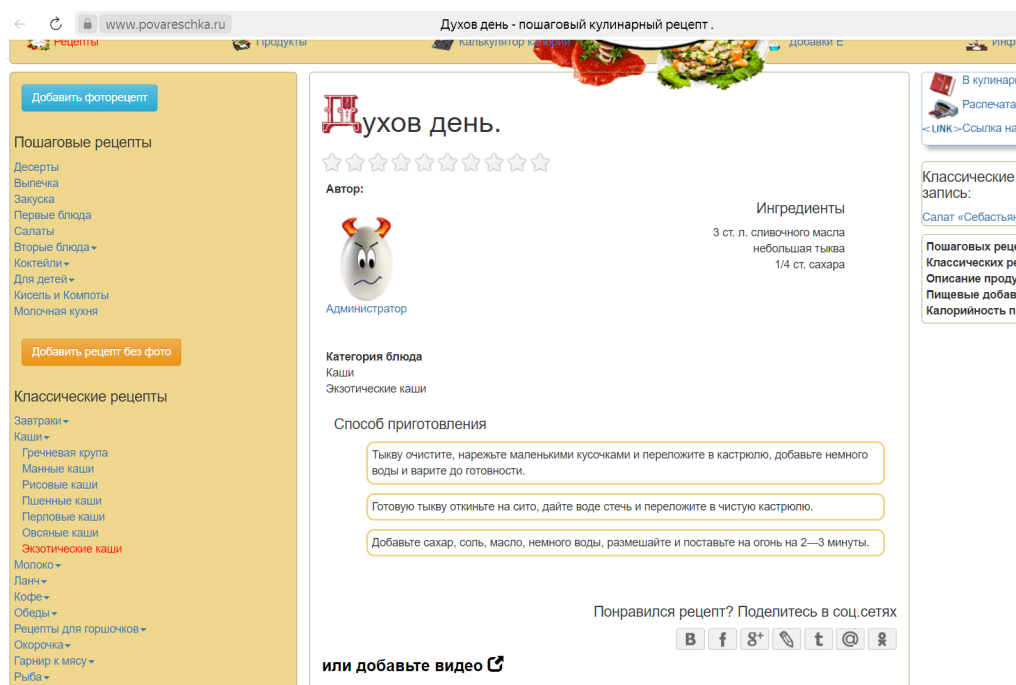


Рисунок 1 – Страница сайта, с которой необходимо скачать рецепт

```
Файл  Изменить  Просмотр
<a href="/user/0.html" rel="author" class="clearfix">
  
  <span itemprop="author" class="author clearfix">Администратор</span></a><br>
  <a href="/result/user/podpiska.php?id=0" rel="nofollow" class="btn btn-sm"></a>
</div>

<b>Категория блюда</b><br>
<div itemprop="recipeCategory"><div class="category"> Каши</div><div class="sub-category"> Экзотические каши</div></div>
</aside>

<section class="col-sm-8 col-xs-9 margin text-right list-unstyled">
  <h2>Ингредиенты</h2>
  <ul id="ing">
    <li itemprop="ingredients" class="ingredient">3 ст. л. сливочного масла</li><li itemprop="ingredients" class="ingredient">
      небольшая тыква</li><li itemprop="ingredients" class="ingredient">1/4 ст. сахара</li>
  </ul>
</section>

<section class="instructions col-sm-12 col-xs-12 marginbottom clearfix">
  <h2>Способ приготовления</h2>
  <ul itemprop="recipeInstructions" id="instruction">
    <li class="instruction">Тыкву очистите, нарежьте маленькими кусочками и переложите в кастрюлю, добавьте немного воды и варите до
      готовности.</li>
    <li class="instruction">Готовую тыкву откиньте на сито, дайте воде стечь и переложите в чистую кастрюлю.</li>
    <li class="instruction">Добавьте сахар, соль, масло, немного воды, размешайте и поставьте на огонь на 2-3 минуты.</li>
  </ul>
</section>
</article>
<div class="text-right margintop clear">
  <h4>Понравился рецепт? Поделитесь в соц.сетях</h4>
  <a onClick="share('vkontakte.ru/share.php?url=')" href="javascript: void(0)" class="b-share-icon b-share-icon_vkontakte"></a>
  <a onClick="share('www.facebook.com/sharer.php?url=')" href="javascript: void(0)" class="b-share-icon b-share-icon_facebook"></a>
  </div>
```

Рисунок 2 – Файл, содержащий html-код страницы

5 Описание исследования

В ходе исследования сравнивалось время работы программ с различным количеством потоков: 1, 2, 4, 8, 16, 32, 48, 64.

Все замеры проводились на ЭВМ, характеристики которой приведены ниже:

- процессор – 12th Gen Intel(R) Core(TM) i5-12450H 2.00 ГГц;
- оперативная память – 16,0 ГБ;
- тип системы – 64-разрядная операционная система, процессор x64;
- операционная система – Windows 11;
- версия ОС – 23H2;
- 12 логических ядер.

Для получения результата на вход подавалось 200 ссылок. Проводилось 20 замеров, результаты усреднялись.

Полученные результаты представлены на рисунке 3 и в таблице 1:

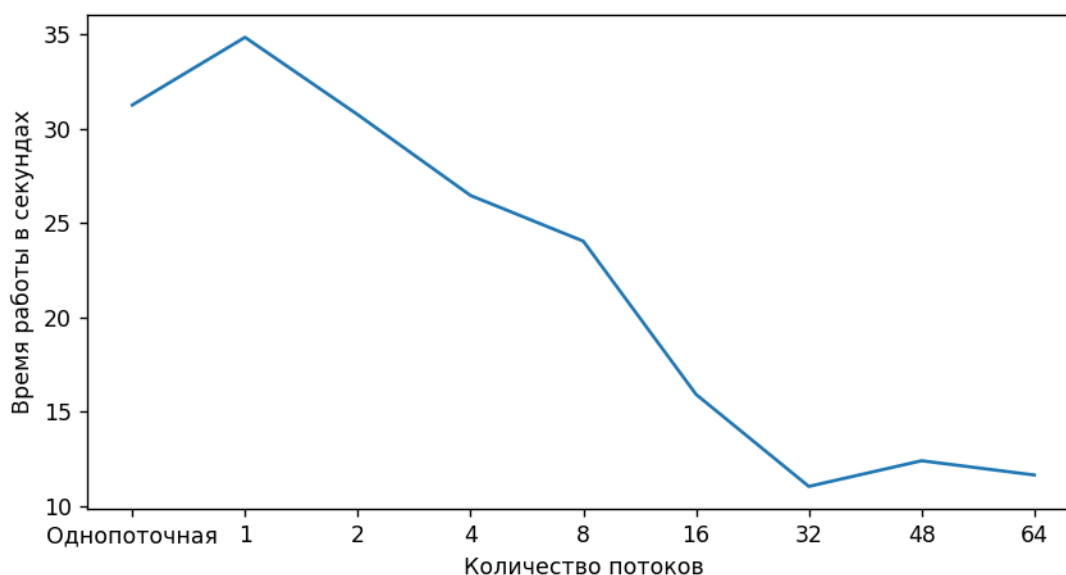


Рисунок 3 – Время работы программы

Таблица 1 – Время работы программы

Количество потоков	Время работы, с
Однопоточная реализация	
1	31.24
Многопоточная реализация	
1	34.82
2	30.73
4	26.45
8	24.03
16	15.92
32	11.05
48	12.4
64	11.66

В результате исследования можно сделать вывод, что использование параллельных вычислений может ускорить программу. Чем больше потоков, тем больше ускорение, однако с определенного момента выигрыш становится незначительным.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы поставленная цель была достигнута, а также были решены следующие задачи:

1. рассмотрена структура сайта;
2. написан скрипт на языке *Python*, который скачивает ссылки рецептов с заданного сайта;
3. написана однопоточная и многопоточная программы, скачивающие рецепты;
4. проведено сравнение временных характеристик работы программ: выявлено, что использование параллельных вычислений ускоряет программу, однако с определенного момента это ускорение незначительно;
5. обоснованы полученные результаты и сделан вывод.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Энтони Уильямс, С++. Практика многопоточного программирования. / Второе издание
2. Ковалев, Введение в многопоточность / [Электронный ресурс] // Режим доступа: <https://rekovalev.site/multithreading-3-cpp/#threads-create>
3. Язык Python / [Электронный ресурс] // Режим доступа: <https://docs.python.org/3/index.html>
4. Кулинарные рецепты / [Электронный ресурс] // Режим доступа: <https://www.povareschka.ru/>