



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 № 5 по курсу «Анализ алгоритмов»

Тема Организация параллельных вычислений по конвейерному принципу

---

Студент Гаврилюк В. А.

---

Группа ИУ7-51Б

---

Оценка (баллы)

---

Преподаватель Волкова Л. Л.

---

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Входные и выходные данные</b>	<b>4</b>
1.1 Входные данные . . . . .	4
1.2 Выходные данные . . . . .	4
<b>2 Преобразование данных</b>	<b>5</b>
<b>3 Пример работы программы</b>	<b>7</b>
<b>4 Тестирование</b>	<b>12</b>
<b>5 Исследование</b>	<b>13</b>
5.1 Технические характеристики . . . . .	13
5.2 Проведение исследования . . . . .	13
<b>ЗАКЛЮЧЕНИЕ</b>	<b>15</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>16</b>

# 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 по конвейерному принципу.

Для достижения цели необходимо выполнить следующие задачи:

- описать входные и выходные данные программы;
- разработать программное обеспечение (ПО), которое осуществляет извлечение и сохранение данных о рецептах из файлов;
- в рамках исследования необходимо получить среднее время существования задачи, среднее время ожидания задачи в каждой из очередей, среднее время обработки задачи на каждой из стадий.

# 1 Входные и выходные данные

## 1.1 Входные данные

Программа принимает на вход следующие данные:

- путь до директории хранения файлов;
- размер очередей между стадиями конвейера;
- количество потоков на каждой стадии.

## 1.2 Выходные данные

В результате выполнения программы извлечённые данные помещаются в хранилище, в файл `log.csv` помещаются записи:

- времени создания задачи;
- времени постановки задачи в каждую из очередей;
- времени начала обработки на каждой из стадий;
- времени уничтожения задачи.

Также в консоль выводятся статистические данные:

- о среднем времени существования задачи;
- о среднем времени ожидания задачи в каждой из очередей;
- о среднем времени обработки задачи на каждой из стадий.

## 2 Преобразование данных

Процесс преобразования входных данных в выходные происходит следующим образом:

- программа получает на вход путь до директории хранения файлов, размер очередей между стадиями конвейера, количество потоков на каждой стадии, флаг удаления имеющихся данных из хранилища перед началом обработки;
- поток-генератор задач считывает названия файлов из указанной директории, для каждого имени генерирует задачу и помещает её в первую очередь;
- в каждом потоке первой стадии выполняется извлечение задачи из очереди, чтение содержимого из файла с именем, указанным в задаче, и задача с извлечёнными данными передаётся в следующую очередь;
- в каждом потоке второй стадии выполняется извлечение задачи из очереди, нахождение данных о рецепте, помещение задачи с информацией о рецепте в следующую очередь;
- в каждом потоке третьей стадии выполняется извлечение задачи из очереди, сохранение информации о рецепте в базу данных (БД) и помещение задачи в выходную очередь;
- поток-накопитель считывает задачи из выходной очереди завершающей стадии обработки, выполняет логирование указанных в разделе 1.2 данных и удаление задач;
- после завершения обработки в консоль выводятся статистические данные.

На рисунке 2.1 продемонстрирована схема алгоритма параллельной обработки по конвейерному принципу.

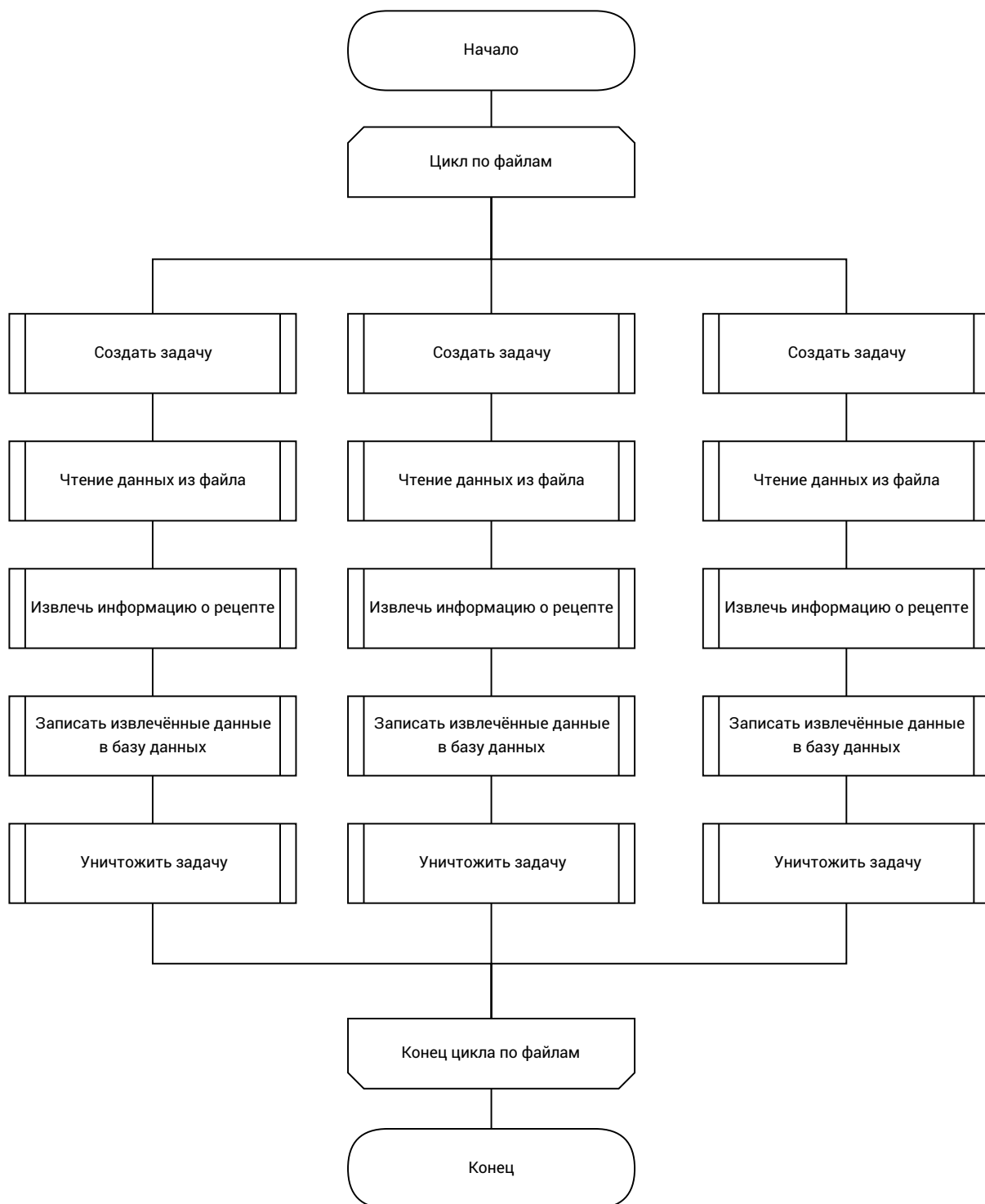


Рисунок 2.1 – Схема алгоритма параллельной обработки по конвейерному принципу

### 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:

- ../data для dir (путь до директории хранения считываемых файлов).
- 1 для chan (размер очереди между стадиями);
- 1 для workers (количество горутин на одной стадии).

Листинг 3.1 – Пример запуска программы

```
$ go run ./cmd/app/main.go --workers=10 --chan=10
-----
Statistics
-----
Workers: 10
Channel length: 10
Num of tasks: 100
Avg task lifetime: 43.545334ms
Total time: 123.339711ms
+-----+-----+-----+
| STAGE NAME          | AVG WAITING TIME | AVG PROCESSING TIME |
+-----+-----+-----+
| readFileContent     | 124.711us        | 2.484us              |
| parseRecipe         | 170.767us        | 81.219us             |
| saveRecipe          | 146ns            | 5.111us              |
+-----+-----+-----+
```

На листинге 3.2 представлен сокращённый пример дампа данных, полученных из БД.

### Листинг 3.2 – Пример дампа данных, полученных из БД

```
[{
  "_id": {
    "$oid": "675d4c565f65a245bcd9fe6e"
  },
  "id": 2,
  "issue_id": 9144,
  "url":
    "https://www.edimdoma.ru/retsepty/121730-salat-simvol-goda",
  "title": "Салат Символ года",
  "ingredients": [
    {
      "name": "крабовые палочки",
      "unit": "г",
      "quantity": 62.5
    },
    {
      "name": "кукуруза консервированная ТМ Фрау Марта",
      "unit": "банка",
      "quantity": 0.25
    },
    ...
    {
      "name": "маслины",
      "unit": "шт.",
      "quantity": 1.0
    }
  ],
  "steps": [
    "Подготавливаем все ингредиенты. Заранее отвариваем 3 яйца и
      150 г риса.",
    ...
    "Перекладываем все нарезанные ингредиенты в миску, добавляем
      отваренный рис и кукурузу ТМ Фрау Марта.",
    "Заправляем салат майонезом, соль и перец добавляем по
      вкусу.",
    ...
  ]
}]
```

На рисунке 3.1 приведён пример записей с информацией о рецептах в



базе данных.

★ recipes								
	_id ObjectId	id Int32	issue_id Int32	url String	title String	ingredients Array	steps Array	image_url String
1	ObjectId("67798ac4145338...")	10	9144	"https://www.edindoma.ru..."	"Лимонные прианки"	[ ] 7 elements	[ ] 6 elements	No field
2	ObjectId("67798ac4145338...")	2	9144	"https://www.edindoma.ru..."	"Салат с арбузом и брынз...	[ ] 8 elements	[ ] 6 elements	No field
3	ObjectId("67798ac4145338...")	4	9144	"https://www.edindoma.ru..."	"Праздничный салат "	[ ] 11 elements	[ ] 7 elements	No field
4	ObjectId("67798ac4145338...")	11	9144	"https://www.edindoma.ru..."	"Холодец"	[ ] 9 elements	[ ] 8 elements	No field
5	ObjectId("67798ac4145338...")	8	9144	"https://www.edindoma.ru..."	"Салат «Символ года»"	[ ] 9 elements	[ ] 5 elements	No field
6	ObjectId("67798ac4145338...")	5	9144	"https://www.edindoma.ru..."	"Васоль в чесночно-нибир...	[ ] 18 elements	[ ] 18 elements	"https://e2.edindoma.ru/...
7	ObjectId("67798ac4145338...")	9	9144	"https://www.edindoma.ru..."	"Салат «Морской прибой»"	[ ] 10 elements	[ ] 19 elements	No field
8	ObjectId("67798ac4145338...")	6	9144	"https://www.edindoma.ru..."	"Суперсырная хот-дог-пиц...	[ ] 18 elements	[ ] 16 elements	No field
9	ObjectId("67798ac4145338...")	17	9144	"https://www.edindoma.ru..."	"Сулфе с апельсиновой це...	[ ] 9 elements	[ ] 10 elements	No field
10	ObjectId("67798ac4145338...")	7	9144	"https://www.edindoma.ru..."	"Соление зеленые помидор...	[ ] 7 elements	[ ] 5 elements	"https://e1.edindoma.ru/...
11	ObjectId("67798ac4145338...")	3	9144	"https://www.edindoma.ru..."	"Заверша с запеченной ку...	[ ] 19 elements	[ ] 37 elements	"https://e1.edindoma.ru/...
12	ObjectId("67798ac4145338...")	14	9144	"https://www.edindoma.ru..."	"Тосты с плавленым сыром...	[ ] 6 elements	[ ] 9 elements	No field
13	ObjectId("67798ac4145338...")	20	9144	"https://www.edindoma.ru..."	"Меренговый рулет «Бурле...	[ ] 10 elements	[ ] 18 elements	"https://e3.edindoma.ru/...
14	ObjectId("67798ac4145338...")	19	9144	"https://www.edindoma.ru..."	"Облепиховый морс"	[ ] 8 elements	[ ] 5 elements	No field
15	ObjectId("67798ac4145338...")	1	9144	"https://www.edindoma.ru..."	"Компот из ревеня, яблоч...	[ ] 4 elements	[ ] 10 elements	"https://e3.edindoma.ru/...
16	ObjectId("67798ac4145338...")	12	9144	"https://www.edindoma.ru..."	"Селедка под шубой с мар...	[ ] 9 elements	[ ] 9 elements	No field
17	ObjectId("67798ac4145338...")	18	9144	"https://www.edindoma.ru..."	"Клеветный лимонад с ша...	[ ] 6 elements	[ ] 6 elements	No field
18	ObjectId("67798ac4145338...")	21	9144	"https://www.edindoma.ru..."	"Салат «Крабовый»"	[ ] 13 elements	[ ] 12 elements	No field
19	ObjectId("67798ac4145338...")	23	9144	"https://www.edindoma.ru..."	"Горюница с олятами и м...	[ ] 9 elements	[ ] 7 elements	No field
20	ObjectId("67798ac4145338...")	27	9144	"https://www.edindoma.ru..."	"Салат с брокколи «Ромул...	[ ] 8 elements	[ ] 5 elements	No field
21	ObjectId("67798ac4145338...")	29	9144	"https://www.edindoma.ru..."	"Компот из айвы"	[ ] 4 elements	[ ] 5 elements	No field
22	ObjectId("67798ac4145338...")	13	9144	"https://www.edindoma.ru..."	"Блинный пиро-улитка с ...	[ ] 11 elements	[ ] 18 elements	"https://e2.edindoma.ru/...
23	ObjectId("67798ac4145338...")	16	9144	"https://www.edindoma.ru..."	"Куриный томатный суп с ...	[ ] 13 elements	[ ] 6 elements	No field
24	ObjectId("67798ac4145338...")	28	9144	"https://www.edindoma.ru..."	"Суп из баклажанов с гри...	[ ] 10 elements	[ ] 3 elements	No field
25	ObjectId("67798ac4145338...")	22	9144	"https://www.edindoma.ru..."	"Закусочные тарты с крев...	[ ] 13 elements	[ ] 35 elements	"https://e2.edindoma.ru/...

Рисунок 3.1 – Пример сохранённых данных о рецептах в БД

На рисунке 3.2 продемонстрирован пример содержимого файла log.csv в результате работы программы:

```
algorithm-analysis > 05_parallel_pipeline_processing > log.csv > data
1 TaskID, Created, Destructed, readFileContentQueued, readFileContentStarted, readFileContentFinished, parseRecipeQueued, parseRecipeStarted, parseRecipeFinis
2 10, 1735867689012361, 1735867689031070, 1735867689012361, 1735867689014040, 1735867689014761, 1735867689014762, 1735867689014853, 1735867689026666, 173586768
3 6, 1735867689012359, 1735867689032611, 1735867689012359, 1735867689013227, 1735867689014710, 1735867689014710, 1735867689014756, 1735867689026676, 1735867689
4 0, 1735867689012351, 1735867689037448, 1735867689012351, 1735867689012427, 1735867689014051, 1735867689014051, 1735867689014296, 1735867689027052, 1735867689
5 11, 1735867689012362, 1735867689040620, 1735867689012362, 1735867689014054, 1735867689014585, 1735867689014585, 1735867689014743, 1735867689028860, 173586768
6 8, 1735867689012360, 1735867689040888, 1735867689012360, 1735867689013283, 1735867689016476, 1735867689016476, 1735867689016650, 1735867689027415, 1735867689
7 9, 1735867689012361, 1735867689041917, 1735867689012361, 1735867689012367, 1735867689014900, 1735867689014901, 1735867689014947, 1735867689026722, 1735867689
8 3, 1735867689012357, 1735867689042026, 1735867689012357, 1735867689013287, 1735867689016395, 1735867689016395, 1735867689016642, 1735867689029105, 1735867689
9 12, 1735867689012363, 1735867689042936, 1735867689012363, 1735867689014592, 1735867689016456, 1735867689016457, 1735867689016642, 1735867689029943, 1735867689
10 4, 1735867689012357, 1735867689043096, 1735867689012357, 1735867689012389, 1735867689014037, 1735867689014037, 1735867689014219, 1735867689029712, 1735867689
11 2, 1735867689012355, 1735867689043311, 1735867689012355, 1735867689013185, 1735867689015493, 1735867689015493, 1735867689015507, 1735867689032865, 1735867689
12 1, 1735867689012355, 1735867689044804, 1735867689012355, 1735867689013277, 1735867689016645, 1735867689016646, 1735867689027057, 1735867689042873, 1735867689
13 15, 1735867689012363, 1735867689044908, 1735867689012363, 1735867689014926, 1735867689016711, 1735867689016712, 1735867689027418, 1735867689041053, 173586768
14 16, 1735867689012364, 1735867689044944, 1735867689012364, 1735867689015497, 1735867689016507, 1735867689016507, 1735867689026682, 1735867689039805, 173586768
15 23, 1735867689014763, 1735867689046550, 1735867689014763, 1735867689016657, 1735867689017694, 1735867689017694, 1735867689029716, 1735867689044168, 173586768
16 13, 1735867689012363, 1735867689046675, 1735867689012363, 1735867689014724, 1735867689016536, 1735867689016536, 1735867689026688, 1735867689045231, 173586768
17 14, 1735867689012363, 1735867689047046, 1735867689012363, 1735867689014767, 1735867689016640, 1735867689016640, 1735867689026729, 1735867689045396, 173586768
18 7, 1735867689012360, 1735867689047824, 1735867689012360, 1735867689013182, 1735867689016833, 1735867689016833, 1735867689028868, 1735867689046057, 1735867689
19 27, 1735867689015951, 1735867689049284, 1735867689015951, 1735867689017696, 1735867689018007, 1735867689018007, 173586768902870, 1735867689046048, 173586768
20 21, 1735867689014286, 1735867689049434, 1735867689014286, 1735867689016545, 1735867689017180, 1735867689017180, 1735867689029110, 1735867689042483, 173586768
21 26, 1735867689014952, 1735867689052416, 1735867689014952, 1735867689017184, 1735867689017755, 1735867689017756, 1735867689029949, 1735867689049508, 173586768
22 28, 1735867689016634, 1735867689052517, 1735867689016634, 1735867689017758, 1735867689018068, 1735867689018068, 1735867689039810, 1735867689047980, 173586768
23 20, 1735867689012364, 1735867689054749, 1735867689012364, 1735867689016509, 1735867689018286, 1735867689018287, 1735867689041059, 1735867689051933, 173586768
24 19, 1735867689012364, 1735867689054811, 1735867689012364, 1735867689016480, 1735867689018484, 1735867689018484, 1735867689042879, 1735867689052030, 173586768
25 18, 1735867689012364, 1735867689058727, 1735867689012364, 1735867689016464, 1735867689018515, 1735867689018515, 1735867689045236, 1735867689057232, 173586768
26 24, 1735867689014856, 1735867689061317, 1735867689014856, 1735867689016737, 1735867689018739, 1735867689018739, 1735867689046056, 1735867689059096, 173586768
27 22, 1735867689014288, 1735867689061561, 1735867689014288, 1735867689016644, 1735867689018696, 1735867689018696, 1735867689045401, 1735867689059414, 173586768
28 30, 1735867689016636, 1735867689061604, 1735867689016636, 1735867689028739, 1735867689029204, 1735867689029204, 1735867689049512, 1735867689059182, 173586768
29 25, 1735867689014857, 1735867689062095, 1735867689014857, 1735867689016743, 1735867689019263, 1735867689019263, 1735867689047984, 1735867689059384, 173586768
30 17, 1735867689012364, 1735867689062158, 1735867689012364, 1735867689016399, 1735867689018493, 1735867689018493, 1735867689044175, 1735867689056896, 173586768
31 5, 1735867689012358, 1735867689062798, 1735867689012358, 1735867689013359, 1735867689019252, 1735867689019252, 1735867689046064, 1735867689059681, 1735867689
32 29, 1735867689016636, 1735867689064241, 1735867689016636, 1735867689018008, 1735867689018388, 1735867689018388, 1735867689042494, 1735867689062772, 173586768
33 33, 1735867689016666, 1735867689069013, 1735867689016667, 1735867689030429, 1735867689031182, 1735867689031182, 1735867689051943, 1735867689065432, 173586768
34 34, 1735867689016668, 1735867689070257, 1735867689016668, 1735867689030436, 1735867689031252, 1735867689031252, 1735867689056907, 1735867689065224, 173586768
```

Рисунок 3.2 – Пример содержимого файла log.csv

В качестве уникального идентификатора задачи в файле-журнале log.csv используется номер, присваиваемый в порядке создания, начиная с нулевого. Временные метки записываются в формате Unix в микросекундах.

На рисунках 3.3 и 3.4 изображены диаграммы Ганта для пяти задач при одном и пяти обработчиках на каждой стадии соответственно. Рисунок 3.3 демонстрирует параллельность обработки задач в рамках всего конвейера, но при этом последовательную обработку в рамках одной стадии. На рисунке 3.4 задачи выполняются параллельно как в рамках одной стадии, так и в рамках всего конвейера.

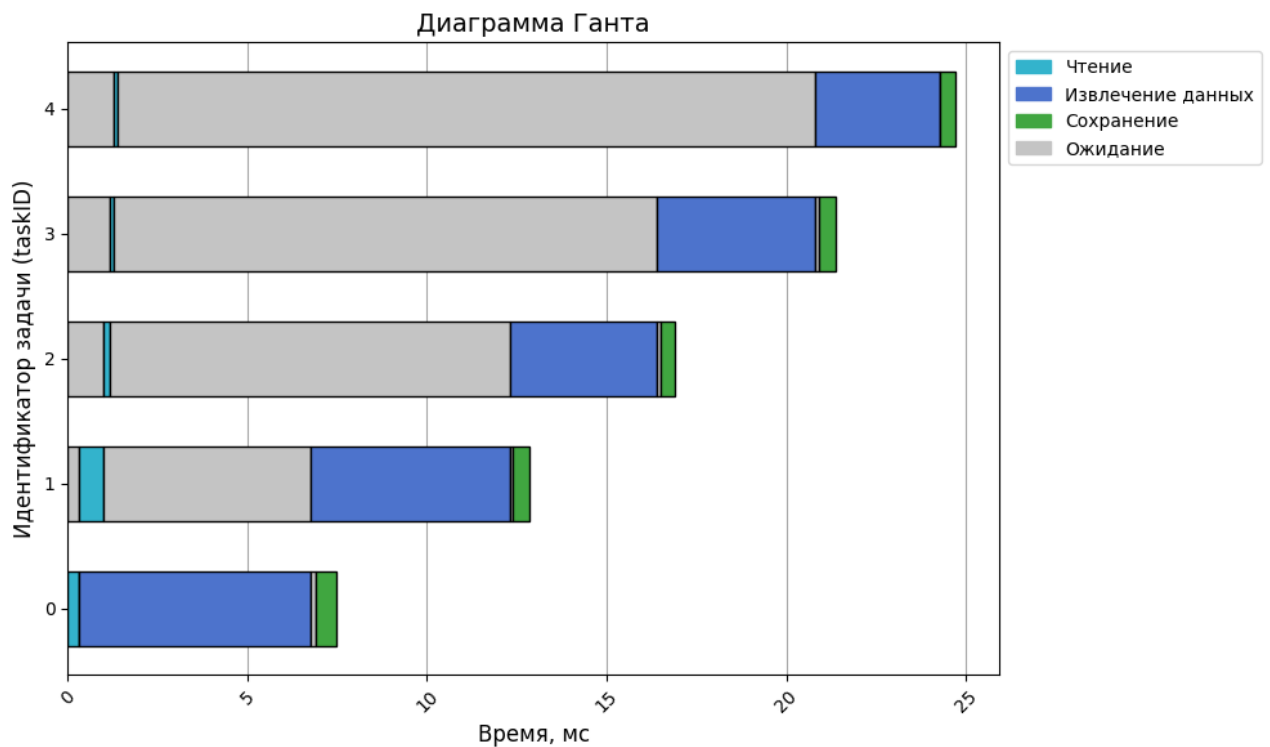


Рисунок 3.3 – Диаграмма Ганта для пяти задач при одном обработчике на каждой стадии

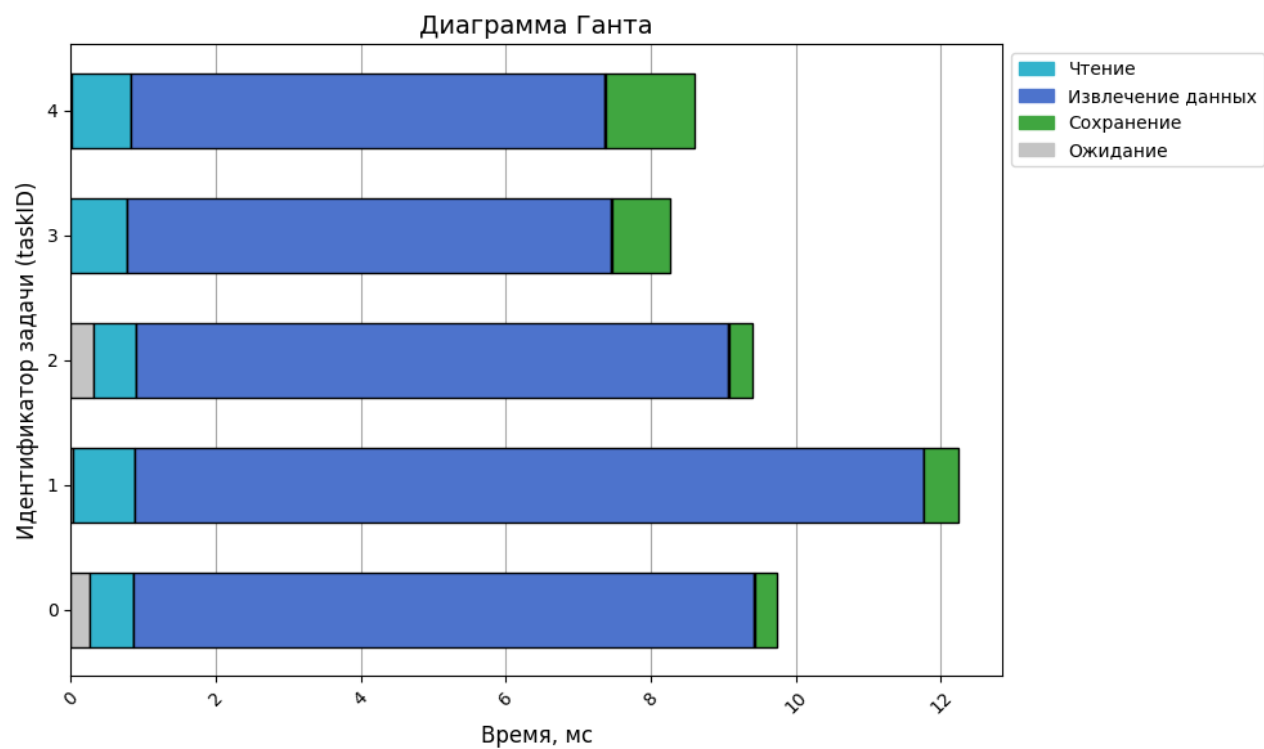


Рисунок 3.4 – Диаграмма Ганта для пяти задач при пяти обработчиках на каждой стадии

## 4 Тестирование

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

Таблица 4.1 – Результаты функционального тестирования программы

Кол-во файлов	Кол-во об-работчиков на каждой стадии	Ожидаемое кол-во полученных записей в БД	Ожидаемое кол-во записей в log.csv	Тест пройден
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 3.70ГГц × 12 [2].

### 5.2 Проведение исследования

В рамках исследования были получены следующие характеристики:

- среднее время существования задачи;
- среднее время ожидания задачи в каждой из очередей;
- среднее время обработки задачи на каждой из стадий.

Каждая характеристика была вычислена при одном и пяти обработчиках на каждой стадии. Количество обрабатываемых файлов равно 5, размер очередей равен 5. Количество замеров равно 20. В таблицах 5.1 — 5.3 приведены полученные значения.

Таблица 5.1 – Среднее время существования задачи при различном количестве обработчиков

Количество обработчиков на каждой стадии, шт	Среднее время существования задачи, мс
1	16.651
5	9.649

Таблица 5.2 – Время ожидания и обработки при одном обработчике на каждой стадии

Стадия	Среднее время ожидания задачи, мс	Среднее время обработки задачи, мс
Чтение данных	0.258	0.019
Извлечение данных	3.881	0.693
Запись извлечённых данных в БД	0.001	0.082

Таблица 5.3 – Время ожидания и обработки при пяти обработчиках на каждой стадии

Стадия	Среднее время ожидания задачи, мс	Среднее время обработки задачи, мс
Чтение данных	0.005	0.171
Извлечение данных	0.001	2.175
Запись извлечённых данных в БД	0.002	0.093

## ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы было реализовано ПО, выполняющее извлечение из HTML файлов и сохранение в базу данных информации о рецептах по конвейерному принципу. При этом в результате работы программы создаётся файл, содержащий временные метки создания, помещения в каждую из очередей, начала и завершения каждой стадии, уничтожения каждой задачи. В рамках исследования были получены следующие характеристики:

- среднее время существования задачи;
- среднее время ожидания задачи в каждой из очередей;
- среднее время обработки задачи на каждой из стадий.

Все задачи работы выполнены. Цель работы, которая заключалась в получении навыка организации параллельных вычислений по конвейерному принципу, достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Стивенс У. Р., Раго С. А.* UNIX. Профессиональное программирование. — 3-е изд. — СПб. : Питер, 2018. — С. 944.
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> (дата обращения: 5.12.2024).