



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 № 5
по курсу «Анализ алгоритмов»
на тему: «Параллельные вычисления по конвейерному принципу»
Вариант № 0

Студент ИУ7-56Б
(Группа)

(Подпись, дата)

Александрова А. Д.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Кормановский М. В.
(И. О. Фамилия)

2024 г.

СОДЕРЖАНИЕ

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

ВВЕДЕНИЕ

Цель работы — получение навыка организации параллельных вычислений по конвейерному принципу.

Задачи работы:

- анализ возможностей организации параллельных вычислений по конвейерному принципу;
- разработка алгоритма, осуществляющего параллельную выгрузку страниц интернет ресурса по конвейерному принципу;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО.

1 Входные и выходные данные

Входными данными являются ссылка на интернет ресурс, содержащий кулинарные рецепты, и количество рецептов, которые необходимо сохранить. Выходными данными являются база данных, содержащая необходимое количество сохраненных рецептов, информация о среднем времени, проведенном задачей на каждом из этапов конвейера и в каждой очереди, и лог выполненных над каждой задачей действий.

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], а затем генерирует задачи для каждой из полученных ссылок. Для каждой задачи выполняется загрузка файла страницы, извлечение из него необходимой информации при помощи регулярных выражений [2] и запись в базу данных при помощи библиотеки SQLite [3], а затем логирование времени выполнения каждого из этапов и закрытие задачи. Каждый из перечисленных этапов (генерация, чтение, извлечение данных, запись и логирование с завершением) выполняется в отдельном

потоке по конвейерному принципу, поэтому между выполнением этих этапов задача помещается в соответствующие очереди. Временные метки начала и завершения выполнения каждого из этапов каждой задачей документируется в логе.

3 Примеры работы программы

На рисунке 3.1 представлен пример работы программы.

```
Enter a number of recipes to download: 10
DB init success
All tasks generated
All tasks read
All tasks extracted
All tasks written
All tasks read
Average existence time: 7.19892 s.
Average generation time: 2.24e-06 s.
Average reading time: 0.157249 s.
Average data extraction time: 1.32856 s.
Average data base writing time: 0.0802984 s.
Average logging and destruction time: 2.09e-06 s.
Average waiting in first queue time: 0.698853 s.
Average waiting in second queue time: 4.93373 s.
Average waiting in third queue time: 2.92e-06 s.
Average waiting in forth queue time: 4.612e-05 s.
```

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	База данных recipes с 1 записью	База данных recipes с 1 записью
2	0	Пустая база данных	Пустая база данных
3	10	База данных recipes с 10 записями	База данных recipes с 10 записями

5 Описание исследования

В ходе исследования необходимо получить лог выполненных действий и замерить среднее время обработки задачи на каждой из стадий, среднее время ожидания в каждой из очередей и среднее время жизни задачи.

В таблице 5.1 приведен лог работы программы для трех задач, где:

- в первую очередь процесс попадает по окончании процесса генерации и находится в ней в ожидании начала процесса чтения;
- во вторую очередь процесс попадает по окончании процесса чтения и находится в ней в ожидании начала процесса извлечения данных;
- в третью очередь процесс попадает по окончании процесса извлечения данных и находится в ней в ожидании начала процесса записи в базу данных;
- в четвертую очередь процесс попадает по окончании процесса записи в базу данных и находится в ней в ожидании начала процесса логирования и закрытия задачи.

Таблица 5.1 – Лог работы программы (начало)

Метка времени, мкс	ID задачи	Событие
26873813647100	1	started generation process
26873813658400	1	added in first queue

Таблица 5.1 – Лог работы программы (продолжение)

Метка времени, мкс	ID задачи	Событие
26873813661200	1	started reading process
26873813672800	2	started generation process
26873813685900	2	added in first queue
26873813711800	3	started generation process
26873813715200	3	added in first queue
26873946985400	1	added in second queue
26873946988900	2	started reading process
26873946989000	1	started extraction process
26874106097500	2	added in second queue
26874106099400	3	started reading process
26874233916000	3	added in second queue
26875073214900	1	added in third queue
26875073216100	2	started extraction process
26875073216900	1	started writing process
26875101161000	1	added in forth queue
26875101164000	1	started destruction process
26875101166200	1	ended destruction process
26876177932300	2	added in third queue
26876177934100	3	started extraction process
26876177935700	2	started writing process
26876235905400	2	added in forth queue
26876235908900	2	started destruction process
26876235911100	2	ended destruction process
26877299647100	3	added in third queue
26877299650000	3	started writing process
26877359864200	3	added in forth queue
26877359869600	3	started destruction process
26877359872000	3	ended destruction process

Среднее время обработки задачи на каждой из стадий, среднее время ожидания в каждой из очередей, среднее время жизни задачи при выполнении 50 задач указаны в таблице 5.2.

Таблица 5.2 – Среднее время прохождения этапов конвейера

Этап обработки задачи	Среднее время обработки, с
Среднее время существования задачи	$3.298 \cdot 10$
Среднее время генерации задачи	$4.712 \cdot 10^{-6}$
Среднее время чтения данных	$1.727 \cdot 10^{-1}$
Среднее время извлечения данных	1.278
Среднее время записи данных	$8.303 \cdot 10^{-2}$
Среднее время логирования и завершения задачи	$2.256 \cdot 10^{-6}$
Среднее время ожидания в первой очереди	4.126
Среднее время ожидания во второй очереди	$2.732 \cdot 10^1$
Среднее время ожидания в третьей очереди	$1.161 \cdot 10^{-4}$
Среднее время ожидания в четвертой очереди	$8.838 \cdot 10^{-6}$

В результате исследования сделан вывод, что выполнение различных этапов обработки разных задач происходит параллельно, но этапы выполнения одной задачи — последовательно. При этом время ожидания в очереди выполнения очередного этапа пропорционально времени выполнения этого этапа.

ЗАКЛЮЧЕНИЕ

Цель работы достигнута. Решены все поставленные задачи:

- анализ возможностей организации параллельных вычислений по конвейерному принципу;
- разработка алгоритма, осуществляющего параллельную выгрузку страниц интернет ресурса по конвейерному принципу;
- создание ПО, реализующего разработанный алгоритм;
- исследование характеристик созданного ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация библиотеки curl [Электронный ресурс]. — — Режим доступа: <https://curl.se/libcurl/c/libcurl.html> (Дата обращения: 2024-12-10).
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: 2024-12-10).
3. Документация библиотеки SQLite [Электронный ресурс]. — — Режим доступа: <https://www.sqlite.org/docs.html> (Дата обращения: 2024-12-10).