

Национальный исследовательский ядерный университет «МИФИ»
(Московский Инженерно–Физический Институт)
Кафедра №42 «Криптология и кибербезопасность»

Отчёт
по результатам выполнения
Лабораторной работы №4
«SQLite3»

Дисциплина:	Практические Аспекты Разработки Высокопроизводительного Программного Обеспечения (ПАРВПО)
Студент:	Гареев Рустам Рашитович
Группа:	Б22-505
Преподаватель:	Куприяшин Михаил Андреевич
Дата:	27.03.2025

Оглавление

Технологический стек.....	3
Ход работы.....	4
Заключение.....	7

Технологический стек

memory	8GiB Системная память
processor	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
siblings	8
cpu cores	4
bridge	11th Gen Core Processor Host Bridge/DRAM Registers
display	TigerLake-LP GT2 [Iris Xe Graphics]
gcc	version 13.3.0
OC	Ubuntu 24.04.2 LTS
IDE	Visual Studio Code 1.98.2

Ход работы

Для эксперимента было использовано 3 000 000 строк. Максимальная длина строки в первом эксперименте была равна 10 символам, во втором эксперименте — 100 символам, в третьем эксперименте — 1000 символам. Для поиска в строках использовалась подстрока «ab». Все строки генерируются случайным образом, создавая файлы и таблицы в SQLite3 объёмом несколько десятков и сотен мегабайт.

В результате были получены экспериментальные данные, приведенные ниже.

1. Запись данных.

Время записи строк в SQLite3:

k = 10:	5.27975 секунд;
k = 100:	11.8378 секунд;
k = 1000:	15.9916 секунд.

Время записи строк в файл:

k = 10:	4.54435 секунд;
k = 100:	10.4333 секунд;
k = 1000:	11.6453 секунд.

2. Чтение данных.

Время чтения из SQLite3:

k = 10:	0.204684 секунд;
k = 100:	0.240447 секунд;
k = 1000:	0.206547 секунд.

Время чтения из файла:

k = 10:	0.178554 секунд;
k = 100:	0.211918 секунд;
k = 1000:	26.6063 секунд.

3. Чтение строк с подстрокой "ab".

Время чтения строк с подстрокой из SQLite3:

k = 10: 0.352034 секунд;

k = 100: 1.91416 секунд;

k = 1000: 90.6293 секунд.

Время чтения строк с подстрокой из файла:

k = 10: 0.190586 секунд;

k = 100: 0.242088 секунд;

k = 1000: 26.7403 секунд.

На основании этих данных были построены диаграммы сравнения производительности SQLite3 и прямого ввода-вывода.

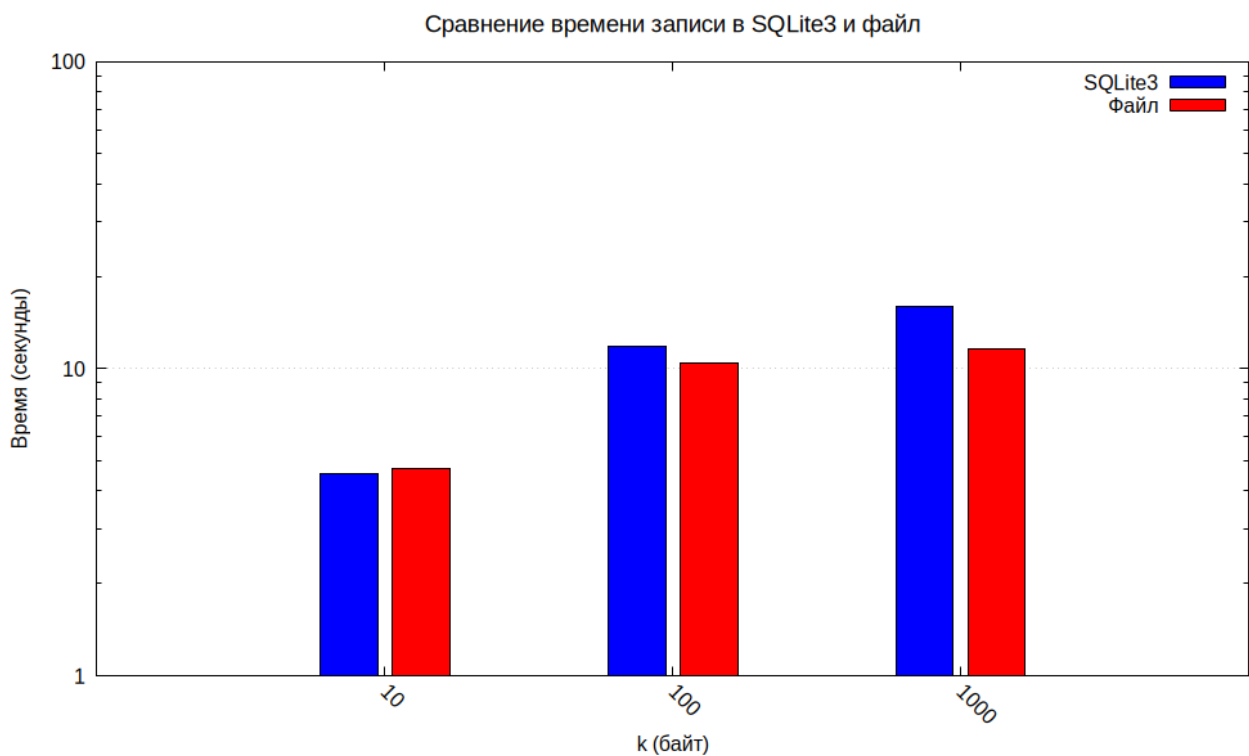


Рисунок 1 — Столбчатая сравнительная диаграмма времени записи

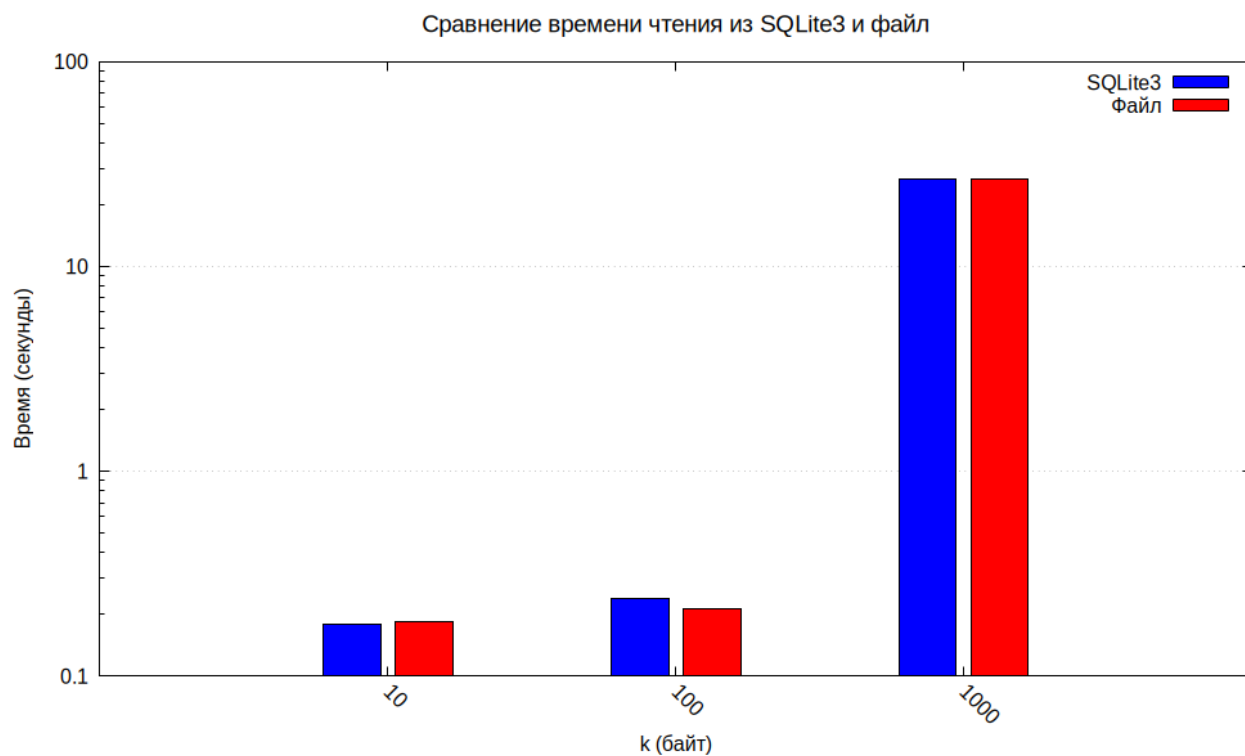


Рисунок 2 — Столбчатая сравнительная диаграмма времени чтения

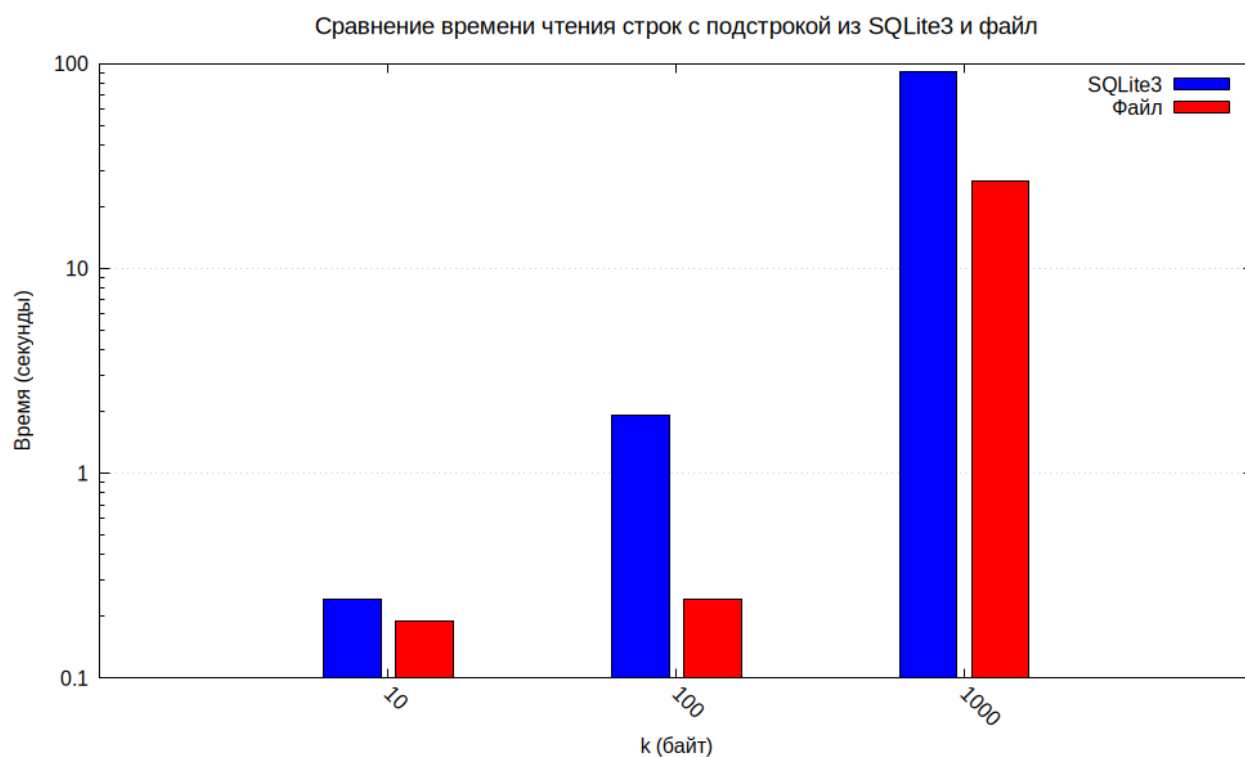


Рисунок 3 — Столбчатая сравнительная диаграмма времени чтения(поиск по подстроке)

Заключение

В ходе выполнения лабораторной работы был проведен сравнительный анализ производительности двух методов работы с данными — использования базы данных SQLite3 и прямого ввода-вывода в файл.

Результаты показали, что для записи и чтения данных из файла производительность была несколько выше, чем у SQLite3, что ожидаемо, учитывая простоту операций с файлами по сравнению с использованием базы данных.

Время работы увеличивалось с ростом размера данных, что было ожидаемо, особенно при записи и чтении больших объемов информации. В экспериментах с подстрокой результаты для SQLite3 показали наибольшее время, что можно объяснить дополнительными накладными расходами при обработке SQL-запросов. Однако с точки зрения масштабируемости, гибкости и удобства работы с большими объемами данных SQLite3 значительно выигрывает, особенно для задач, требующих эффективной фильтрации и обработки данных.

В общем, для задач с небольшими объемами данных и простыми операциями ввода-вывода прямой доступ к файлам может быть более эффективным. Однако для более сложных запросов и больших объемов данных SQLite3 предоставляет удобные средства для управления данными и их фильтрации, что делает этот инструмент полезным в ряде приложений.

Ссылка на гит-репозиторий : <https://github.com/sagilyp/PAPHSD-2/tree/main/lab4>