

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ**

## **ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

**“Низкоуровневое программирование”**

Вариант 3 (Граф узлов с атрибутами)

**Выполнил:**

Студент группы Р33302

Добрягин Михаил  
Александрович

**Преподаватель:**

Кореньков Юрий Дмитриевич



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург, 2023

## 1. Цель

Создать модуль, реализующий хранение в одном файле данных (выборку, размещение и гранулярное обновление) информации общим объёмом от 10GB соответствующего варианту вида.

### Задачи:

- 1) Спроектировать структуры данных для представления информации в оперативной памяти
  - a. Для порции данных. Поддерживать тривиальные значения по меньшей мере следующих типов: четырёхбайтовые целые числа и числа с плавающей точкой, текстовые строки произвольной длины, булевские значения
  - b. Для информации о запросе
- 2) Спроектировать представление данных с учетом схемы для файла данных и реализовать базовые операции для работы с ним:
  - a. Операции над схемой данных (создание и удаление колонок)
  - b. Базовые операции над элементами данных в соответствии с текущим состоянием схемы (над записями заданного вида)
    - i. Вставка элемента данных
    - ii. Перечисление элементов данных
    - iii. Обновление элемента данных
    - iv. Удаление элемента данных
- 3) Используя в сигнатурах только структуры данных из п.1, реализовать публичный интерфейс со следующими операциями над файлом данных:
  - a. Добавление, удаление и получение информации о элементах схемы данных, размещаемых в файле данных, на уровне, соответствующем виду записей
  - b. Добавление нового элемента данных определённого вида
  - c. Выборка набора элементов данных с учётом заданных условий и отношений со смежными элементами данных (по свойствам/полям/атрибутам и логическим связям соответственно)
  - d. Обновление элементов данных, соответствующих заданным условиям
  - e. Удаление элементов данных, соответствующих заданным условиям

- 4) Реализовать тестовую программу для демонстрации работоспособности решения

## 2. Основные структуры

Основные структуры, использующиеся для интерфейса выглядят следующим образом

```
typedef struct {
    Tag_type type;
    char* name; // id
    uint32_t properties_size;
    Type* property_types;
    char** property_names;
} Tag;

typedef struct {
    char* tag;
    Field id;
    uint32_t properties_size;
    Property* properties;
} Node;

typedef struct {
    char* tag;
    Field id;
    Field node1_id;
    Field node2_id;
    uint32_t properties_size;
    Property* properties;
} Edge;
```

Для хранения данных о размещении объектов и другой необходимой информации используется структура Metadata

```
typedef struct {
    uint32_t blocks_size;
    uint32_t draft_blocks_size;
    uint32_t blocks_capacity;
    uint64_t data_size;
    uint32_t headers_offset;
    uint64_t data_offset;
} Metadata;
```

Помимо неё также присутствует структура описывающая непосредственно каждый объект

```
typedef struct {
    uint32_t block_unique_id;
    Entity_type type;
    Block_status status;
    uint64_t data_offset;
    uint64_t data_size;
} Header_block;
```

### 3. Выполнение

Программа поделена на 3 уровня: клиентский, уровень базы данных, уровень непосредственной работы с файлом. Файловый уровень напрямую работает с файлами и собирает различную информацию по которой принимается решение о каких-либо действиях, не имеющих прямого отношения к специфике данных, с которыми мы работаем – например таких, как «расширение файла». Уровень базы данных вносит различные оптимизации, оперируя «метапараметрами». Клиентский уровень же позволяет представлять данные определённым образом и взаимодействовать со внешними программами.

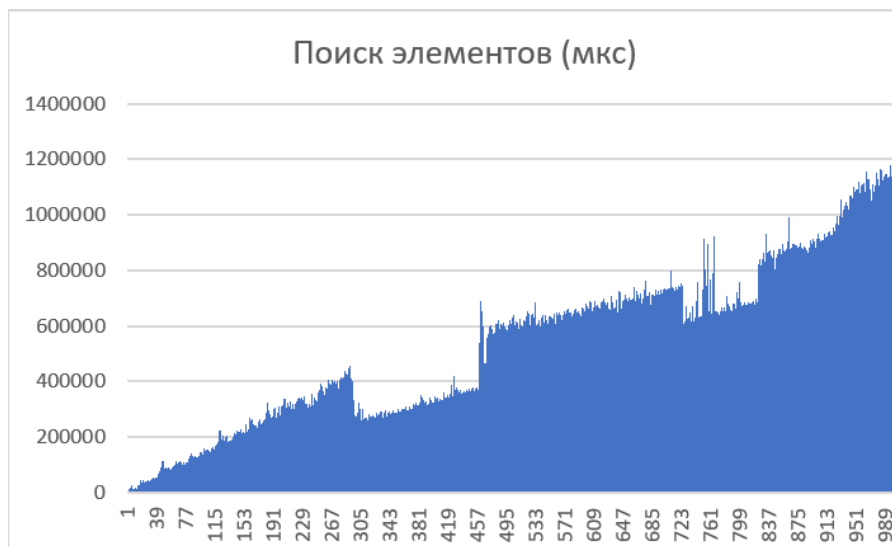
Код можно посмотреть по следующей [ссылке](#)

#### 4. Замер временных показателей программы

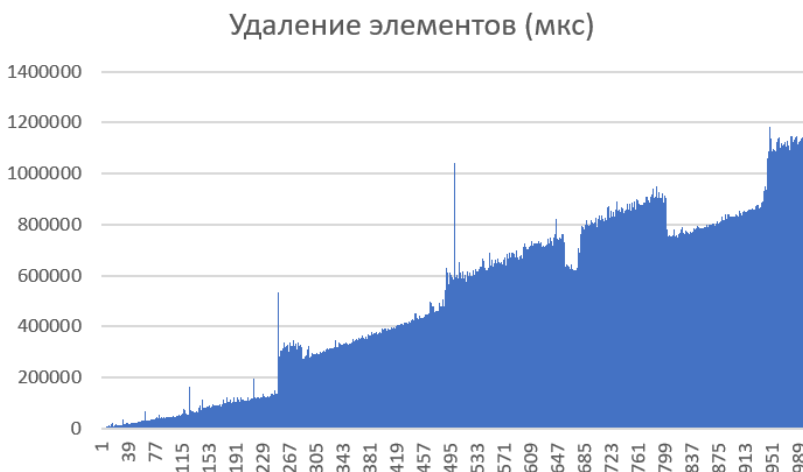
Вставка элементов в зависимости от кол-ва элементов



Поиск элементов в зависимости от их кол-ва



Удаление элементов в зависимости от их кол-ва в базе данных



Обновление элементов в зависимости от их кол-ва



## **5. Вывод**

В результате выполнения лабораторной работы был разработан модуль, реализующий хранение в одном файле данных в виде графа узлов с атрибутами, объем которых может достигать 10GB. Модуль поддерживает стандартные CRUD-операции, а также может работать с предсказуемой производительностью в POSIX системах.