

# Лабораторная работа №1 по дисциплине "Низкоуровневое программирование"

## Зависимости

- Clang
- CMake 3.24

## Сборка проекта, запуск

- Linux

```
bash ./Lab1/start.sh
```

- Windows

```
cd ..
mkdir build
cd build
cmake .. -D CMAKE_C_COMPILER="C:\msys64\mingw64\bin\clang.exe" -D CMAKE_CXX_COMPILER="C:\msys64\mingw64\bin\clang++.exe"
cmake --build . --target ALL_BUILD --config Release
```

## Цели

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

## Задачи

Порядок выполнения:

1. Спроектировать структуры данных для представления информации в оперативной памяти

- а. Для порции данных, состоящий из элементов определённого рода (см форму данных), поддержать тривиальные значения по меньшей мере следующих типов: четырёхбайтовые целые числа и числа с плавающей точкой, текстовые строки произвольной длины, булевские значения
- б. Для информации о запросе

2. Спроектировать представление данных с учетом схемы для файла данных и реализовать базовые операции для работы с ним:

- Операции над схемой данных (создание и удаление элементов схемы)
- Базовые операции над элементами данных в соответствии с текущим состоянием схемы (над узлами или записями заданного вида)
  1. Вставка элемента данных
  2. Перечисление элементов данных
  3. Обновление элемента данных
  4. Удаление элемента данных

3. Используя в сигнатурах только структуры данных из п.1, реализовать публичный интерфейс со следующими операциями над файлом данных:

- Добавление, удаление и получение информации о элементах схемы данных, размещаемых в файле данных, на уровне, соответствующем виду узлов или записей
- Добавление нового элемента данных определённого вида
- Выборка набора элементов данных с учётом заданных условий и отношений со смежными элементами данных (по свойствам/полям/атрибутам и логическим связям соответственно)
- Обновление элементов данных, соответствующих заданным условиям
- Удаление элементов данных, соответствующих заданным условиям

4. Реализовать тестовую программу для демонстрации работоспособности решения

# Аспекты реализации

## Структуры которые были риализованны при выполнении данной лабораторной работы

Node - элемент дерева

```
struct node {
    int32_t id{};
    string name;
    unordered_map<string, attributes> attribute;
};
```

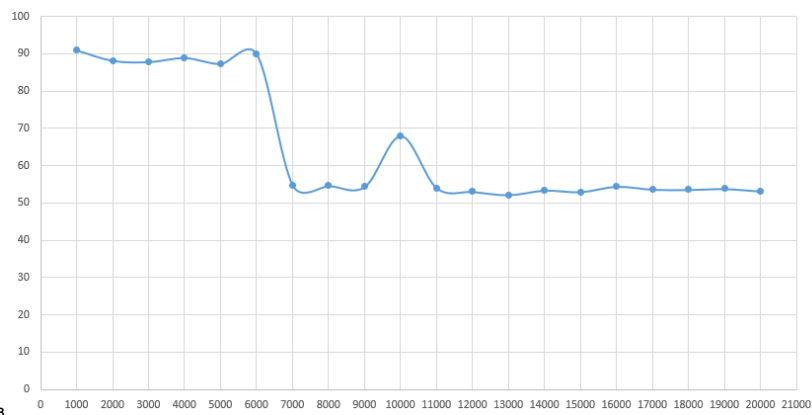
m\_data - мета информация элемета дерева

```
struct m_data {
    int32_t count;
    unordered_map<string, int32_t> pos;
    unordered_map<int32_t, string> path;
    unordered_map<int32_t, unordered_set<int32_t>> child;
    vector<int32_t> free;
};
```

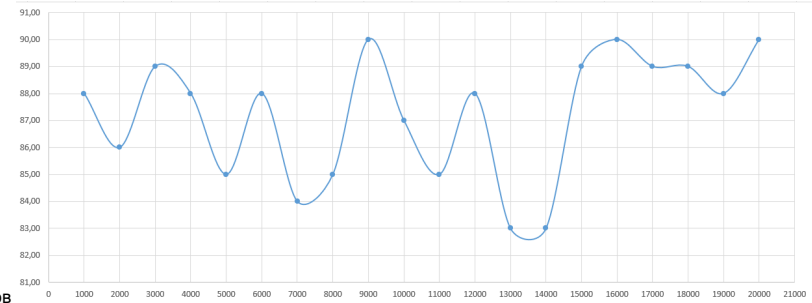
Attributes - атрибуты элемента дерева

```
struct attributes {
    int32_t i;
    bool b{};
    float f{};
    string s;
    enum type {INT, BOOL, FLOAT, STR};
    type label;
```

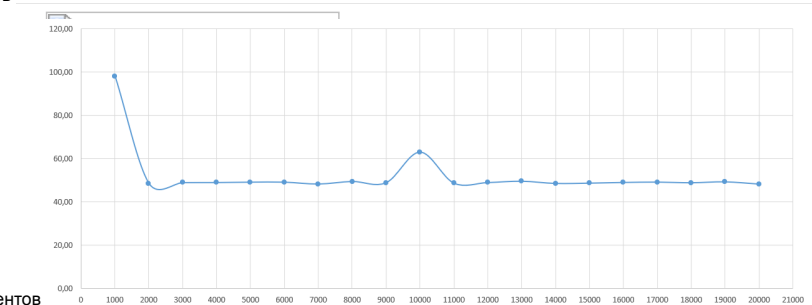
## Результаты



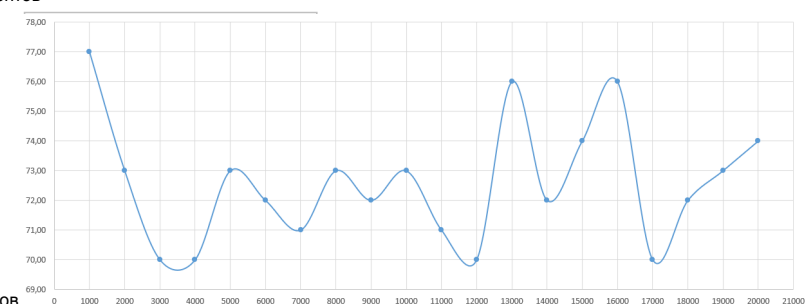
- Вставка 20000 элементов



- Выборка 20000 элементов



- Обновление 20000 элементов



- Удаление 20000 элементов

## Вывод

В результате выполнения лабораторной работы был разработан модуль, реализующий хранение в одном файле данных в виде дерева с атрибутами, объем которых может достигать 10GB. Модуль поддерживает операции select, insert, update, delete. Модуль может работать под управлением ОС семейств Windows и UNIX подобные.