

# Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Низкоуровневое программирование

Лабораторная работа №1

Вариант №1 «Документное дерево»

Выполнила:

Руденко Мария, группа Р33312

Преподаватель:

Кореньков Юрий Дмитриевич

Санкт-Петербург

2023

**Задание:**

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

**Выполнение:**

Для работы с элементами были реализованы операции (commands/\*.c):

1. Добавление элемента
2. Выборка элемента(-ов)
3. Обновление
4. Удаление

Также реализованы операции закрытия и открытия файла(basic\_tools/basic\_file\_manager.c).

Программа состоит из модулей:

- basic\_types – базовые типы + crud
- basic\_tools – io + работа с файлом
- adv\_types – преобразование структур для записи
- adv\_tools – заполнение файла
- generator – генерация пустого файла
- crud\_tools – манипуляции с данными, базовые crud функции
- commands – команды для работы (применяются из консоли)
- ui – обработчик команд

Программа может быть запущена, как и в обычном режиме, так и в режиме парсинга данных из файла.

Ниже представлены разработанные структуры  
Определение типов данных

```
struct uint64_list{
    uint64_t value;
    struct uint64_list *next;
};
```

```
struct result_list_tuple {
    uint64_t id;
    struct tuple *value;
    struct result_list_tuple *next;
};
```

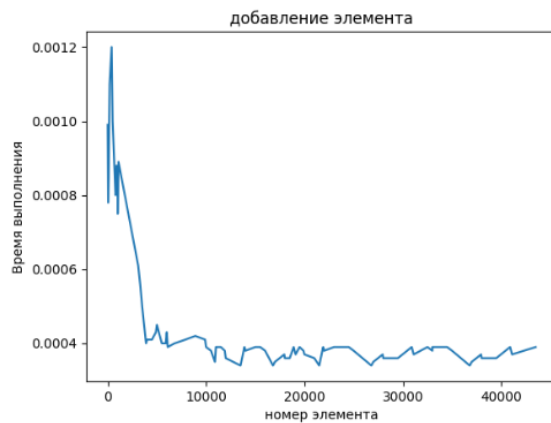
```
struct tree_subheader {
    uint64_t ASCII_signature;
    uint64_t root_offset;
    uint64_t first_seq;
    uint64_t second_seq;
    uint64_t cur_id;
    uint64_t pattern_size;
};
```

```
#pragma pack(push, 4)
struct key_header {
    uint32_t size;
    uint32_t type;
};
struct key {
    struct key_header *header;
    char *key_value;
};
#pragma pack(pop)

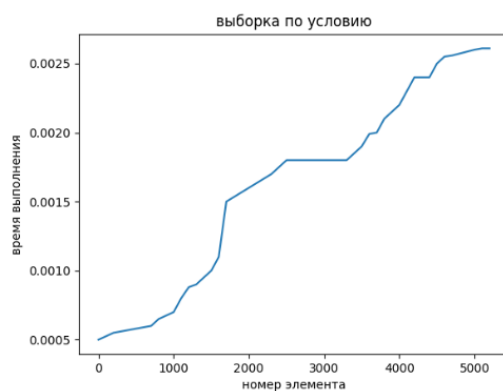
struct tree_header {
    struct tree_subheader *subheader;
    struct key **pattern;
    uint64_t *id_sequence;
};
};
```

## Графики полученных данных:

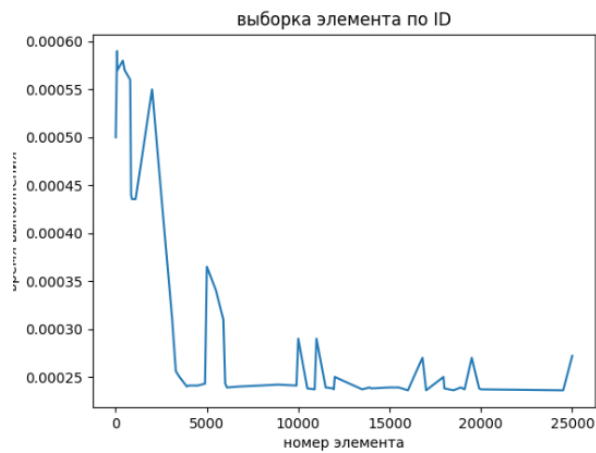
### Добавление нового элемента



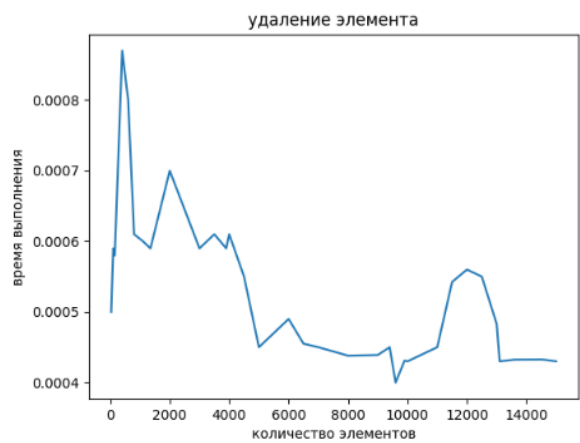
### Выборка по условию



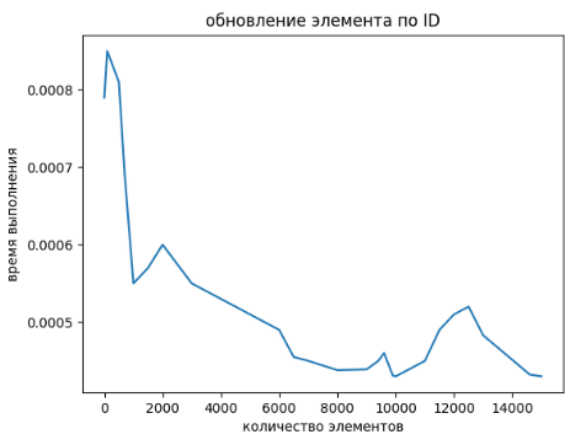
## Выборка по ID



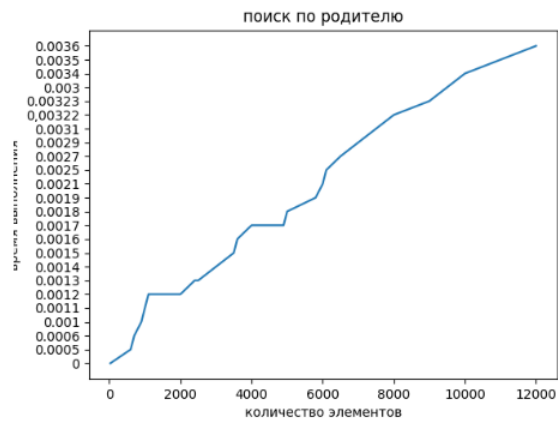
## Удаление элемента



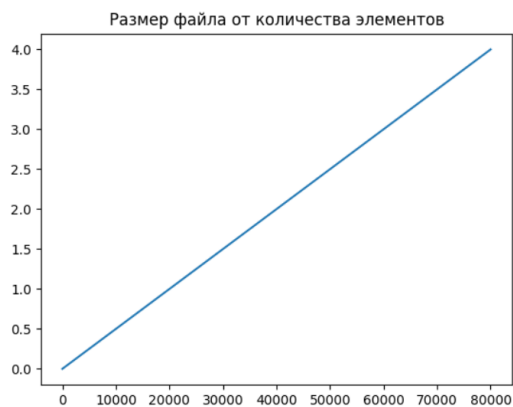
## Обновление элемента по ID



### Поиск элемента по родителю (поиск по отношению)



### Зависимость размера файла от количества элементов



Графики показали, что программа потребляет разумное количество ресурсов и выполняет свои функции за допустимое время. Программа соответствует желаемым требованиям.

### Вывод:

При выполнении лабораторной работы была разработана программа, которая поддерживает хранение и обработку данных в файле большого объема. Также был написан консольный интерфейс для взаимодействия с файлом. Также тесты показали, что программа потребляет разумное количество ресурсов.