

Министерство науки и высшего образования Российской Федерации
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 6

Выполнил:
Скориков Родион
Группа: Р33312
Преподаватель:
Кореньков Юрий Дмитриевич

Санкт-Петербург, 2024г

Цели

Целью данной работы является разработка модуля, реализующего хранение в файле данных информации, поддерживающего операции вставки, удаления, выборки и обновления элементов данных. По варианту форма данных – граф узлов с атрибутами.

Описание работы

Система состоит из следующих модулей:

query.c: Интерфейс к базе данных

database.c: Содержит методы работы с базой данных

file.c: Содержит методы чтения/записи в файл

graph.c: Содержит методы работы с представлением графа в программе

main.c: Этот файл содержит точку входа в программу.

Аспекты реализации

База данных разделена на страницы, представленные struct page

```
struct page {
    struct page_header header;
    struct content_header content_header;
    struct block blocks[BLOCKS_NUM];
};
```

Сама страница содержит свой заголовок

```
struct page_header {
    uint8_t is_occupied;

    uint64_t page_num;
    uint64_t next_same_page_num;

    uint64_t occupied_blocks_num;
    uint8_t occupied_blocks[BLOCKS_NUM];
};
```

Одинаковые страницы – страницы, хранящие один и тот же тип ноды/ребра.

Заголовок, описывающий тип данных

```
struct content_header {
    uint8_t content_type;
    union {
        struct node_type node_type;
        struct edge_type edge_type;
    };
};
```

И страница разделена на блоки фиксированного размера

```
struct block {
    union {
        struct node_block as_node;
        struct edge_block as_edge;
    };
};
struct node_block {
    struct node_header header;

    union {
        struct main_node_info as_main;
        struct additional_node_info as_additional;
    } info;

    uint8_t data[BLOCK_DATA_SIZE];
};
struct edge_block {
    uint32_t src_node_id;
    uint32_t dst_node_id;
    uint64_t next_src_node_edge_addr;
};
```

При этом блоки, хранящие ноду могут быть «главными» или «дополнительными» (если информация, хранящаяся в ноде не влезает в один блок)

```
struct main_node_info {
    uint32_t node_id;
    uint64_t first_edge_addr;
    uint64_t edges_num;
    uint64_t additional_block_addr;
    uint64_t data_size;
```

```
};

struct additional_node_info {
    uint64_t next_additional_block_addr;
    uint64_t data_size;
};
```

Граф в программе представлен следующими структурами:

```
struct node_type {
    uint8_t id;
    char name[MAX_NAME_SIZE];

    uint8_t attribute_num;
    uint8_t attribute_types[MAX_ATTR_NUM];
};

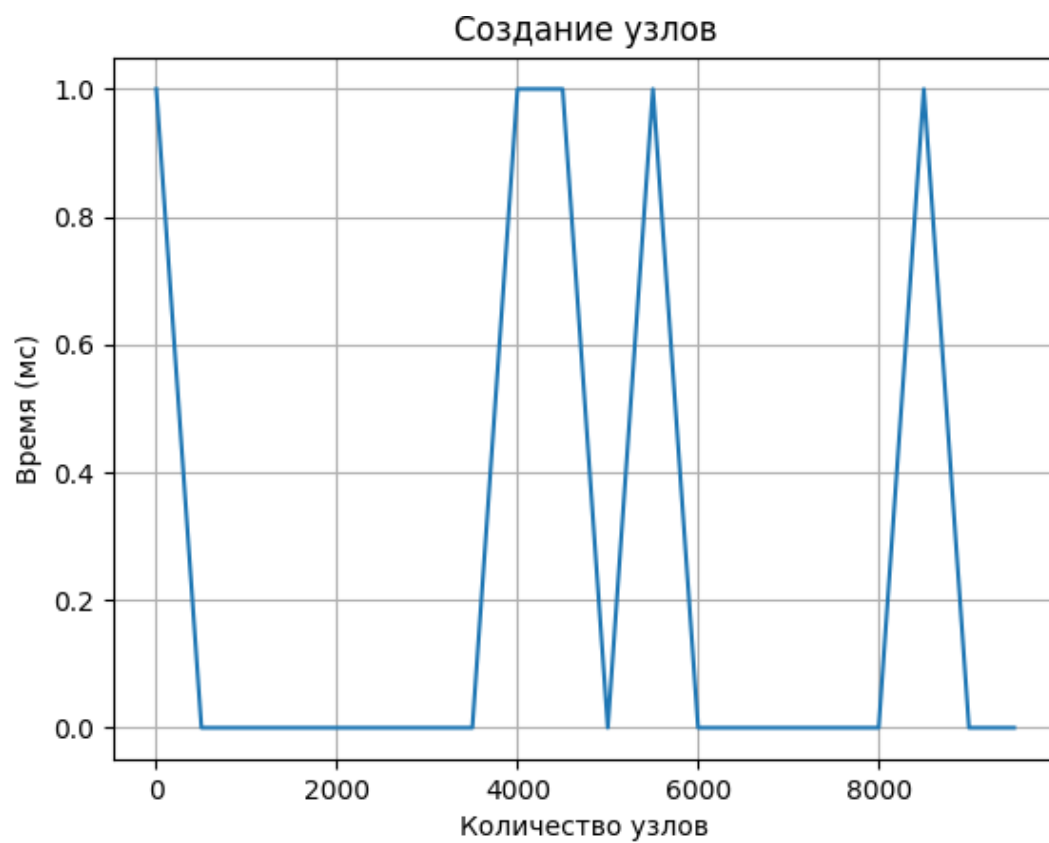
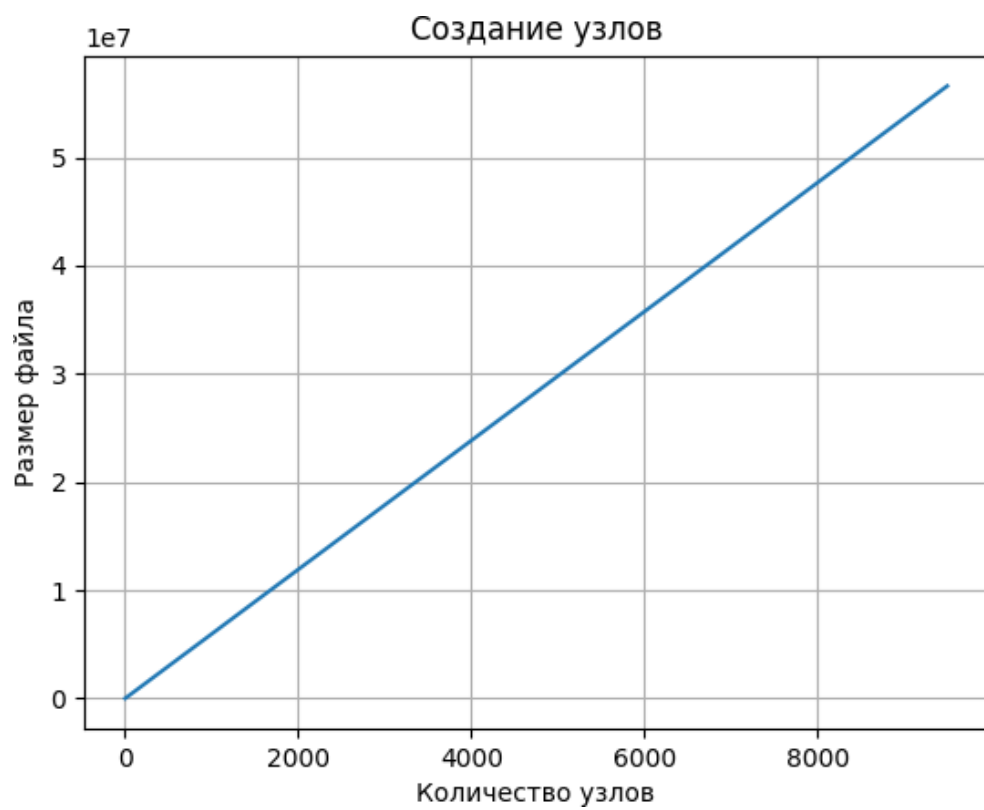
struct node {
    uint32_t id;
    struct node_type type;
    struct attribute* attributes;
    uint32_t edges_num;
    struct edge* edges;
};

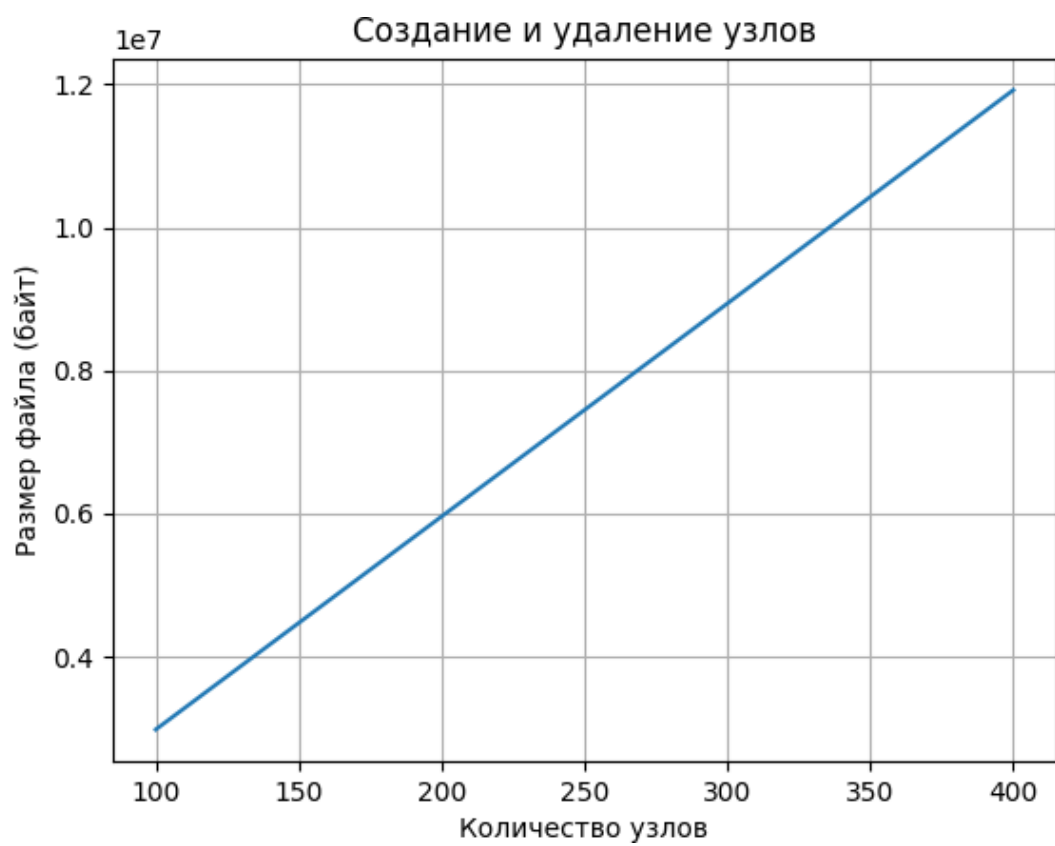
struct attribute {
    union {
        int32_t as_int32;
        float as_float;
        bool as_bool;
        char* as_string;
    } value;
};

struct edge_type {
    uint8_t id;
    char name[MAX_NAME_SIZE];
};

struct edge {
    struct edge_type type;
    uint32_t src_node_id;
    uint32_t dest_node_id;
};
```

Результаты







Вывод

В ходе выполнения лабораторной работы мною был разработан модуль, реализующий базу данных для бедных. Кроме этого, я в полной мере осознал всю жестокость и безжалостность языка С. В ходе выполнения лабораторной я поняла, насколько скудны мои знания в этом невероятно интересном языке. Также я поняла, какую огромную силу он дает, но взамен накладывает на тебя большую ответственность.