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 3

Выполнил:

Скориков Родион

Группа: P33312

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];

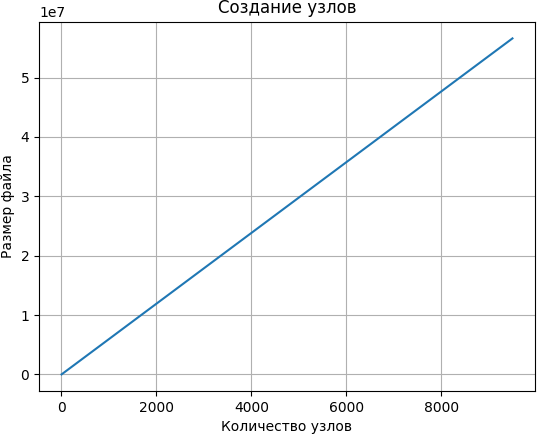
};

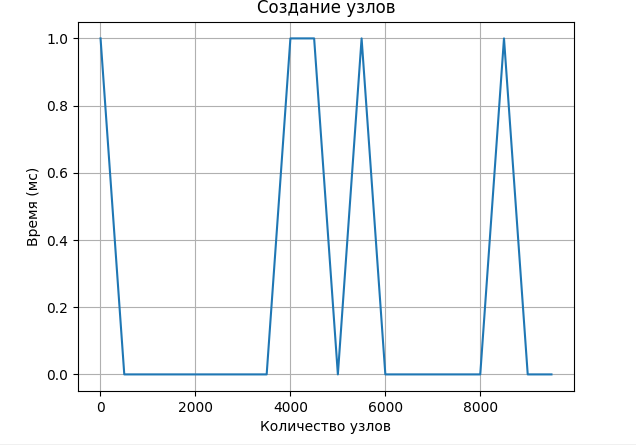
struct edge {

struct edge\_type type; uint32\_t src\_node\_id; uint32\_t dest\_node\_id;

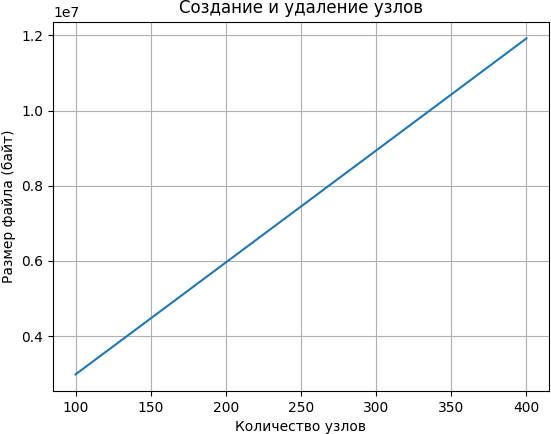
};

# 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 C. 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.