МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №2

по дисциплине

“Низкоуровневое программирование

Вариант №7

Студент:

Лунева Арина Алексеевна

Группа P33312

Преподаватель:

Кореньков Юрий Дмитриевич

Санкт-Петербург, 2022

Цель работы: реализовать модуль для разбора некоторого достаточного подмножества языка запросов по выбору в соответствии с вариантом формы данных.

Задачи:

1. Изучить выбранное средство синтаксического анализа
2. Изучить синтаксис языка запросов и записать спецификацию для средства синтаксического анализа
3. Реализовать модуль, использующий средство синтаксического анализа для разбора языка запросов
4. Реализовать тестовую программу для демонстрации работоспособности созданного модуля, принимающую на стандартный ввод текст запроса и выводящую на стандартный вывод результирующее дерево разбора или сообщение об ошибке

Описание работы:

Тестовая программа принимает в стандартный поток ввода один запрос и выводит его синтаксический разбор.

Описание структур:

filter
```
struct filter {
    struct filter* next;
    struct comparator* comparator_list;
};
```

query
```
struct query {
    uint8_t command;
    struct filter* filters;
    struct value_setting* settings;
};
```

field_value_pair
```
struct field_value_pair {
    char* field;
    uint8_t val_type;
    uint64_t int_value;
    float real_value;
};
```

value_settings
```
struct value_setting {
    struct value_setting* next;
    struct field_value_pair field_value;
};
```

comparator
```
struct comparator {
    struct comparator* next;
    uint8_t operation;
```

```
    struct field_value_pair field_value;
};
```

extended_comparator

```
struct extended_comparator {
    struct extended_comparator* next;
    struct extended_comparator* connected;
    uint8_t operation;
    struct field_value_pair field_value;
};
```

Аспекты реализации:

lexer.l – файл лексера
parser.y – файл парсера
ast.h – заголовочный файл для описания структур

Операции над элементами:

- insert – добавление нового элемента
- find – поиск элемента (элементов)
- update – изменение элемента
- delete – удаление элемента

Операторы сравнения:

- $gt – greater than
- $gte – greater than or equal
- $lt – less than
- $lte – less than or equal
- $ne – not equal

Логические операторы:

- $or – логическое или

Примеры запросов:

`db.find({name: "Vanya", age: 12})`

```
command: 0
 filters:
  filter 0:
   comparator 0:
    field: name
    operation type: 0
    value: Vanya
  filter 1:
   comparator 0:
    field: age
    operation type: 0
    value: 12
```

db.insert({parent: 10}, {name: "Petya", surname: "Petrov", age: 11})

```
command: 2
 filters:
  filter 0:
   comparator 0:
    field: parent
    operation type: 0
    value: 10
 settings:
  field: age
  value: 11
  field: surname
  value: Petrov
  field: name
  value: Petya
```

db.update({name: "Vanya", age: {$ne: 13}}, {$set:{name: "Ivan"}})

```
command: 3
 filters:
  filter 0:
   comparator 0:
    field: name
    operation type: 0
    value: Vanya
  filter 1:
   comparator 0:
    field: age
    operation type: 5
    value: 13
 settings:
  field: name
  value: Ivan
```

db.delete({$or:[age: {$lt: 27}, age: {$gt: 40}]})

```
command: 1
 filters:
  filter 0:
   comparator 0:
    field: age
    operation type: 3
    value: 40
   comparator 1:
    field: age
    operation type: 1
    value: 27
```

Вывод: в результате выполнения лабораторной работы был реализован модуль, производящий синтаксический разбор и анализ языка MongoShell. Для написания лексера и парсера я ознакомилась с технологиями flex и bison.