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 № 2

по дисциплине

«Низкоуровневое программирование»

Вариант: 9 (Cypher)

Выполнила:

Канукова Ева

Группа: P33302

Преподаватель:

Кореньков Юрий Дмитриевич

Санкт-Петербург, 2023г

Цель

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

Программа состоит из следующих модулей:

- parser.y - описание грамматики языка запросов на языке Bison
- lexer.l - описание лексического анализатора на языке Flex
- main.c - точка входа в программу
- query.c - описание структур и методы для их формирования в ходе работы парсера
- printer.c - вывод получившейся структуры в консоль

Аспекты реализации

Поддерживаются операции - create, match-set, match-delete, match - для работы с узлами.

```
struct query {
    char var_name[MAX_STRING_SIZE];
    char label[MAX_STRING_SIZE];

    enum query_type type;
    union {
        struct create_query as_create;
        struct match_query as_match;
        struct set_query as_set;
        struct delete_query as_delete;
    };
};
```

Основная структура для описания запроса.

```
struct match_query {
    struct filter* filter;
    char return_value[MAX_STRING_SIZE];
};

struct delete_query {
    struct filter* filter;
    char delete_value[MAX_STRING_SIZE];
};

struct set_query {
    struct filter* filter;
    struct property* prop;
};


struct create_query {
    struct property* prop;
};
```

Описывают более конкретную структуру каждого запроса.

```
struct property {
    char name[MAX_STRING_SIZE];
    struct value value;
    struct property* next_prop;
```

```
};
```

Описывает поле узла с возможными значениями - bool, int, double, string.

```
enum property_operation {
    EQUALS,
    NOT_EQUALS,
    LESS_THAN,
    GREATER_THAN,
    CONTAINS
};
```

Поддерживаются операции: =, !=, >, <, CONTAINS.

В файле lexer.l описаны правила разбора запроса на токены.
Файл parser.y содержит описание синтаксиса запросов и логику их обработки.

Определенные правила добавляют в структуру query поля. Например:

```
create_query
    : CREATE node { set_query_type(&q, $1); }
    ;
```

Примеры работы программы

1. Создание элемента

```
PS C:\Users\eva\VScode\llp_lab2> ./main
create (p: Person {name: "Alina", age: 20, has_cat: true})
PARSED QUERY:

query type: CREATE

        var name: p
        node label: Person

        properties:
                name: name
                value: "Alina"

                name: age
                value: 20

                name: has_cat
                value: true
```

2. Обновление элемента

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p: Person) where p.name != "Andrew" set p.weight = 60.6
PARSED QUERY:

query type: SET

        var name: p
        node label: Person

where conditions:
                field name: name
                operation: NOT_EQUALS
                value: "Andrew"

changed properties:
                name: weight
                value: 60.600000
```

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p) where p.has_cat = true or p.has_dog = true or p.has_parrot = true set p.has_pet = true
PARSED QUERY:

query type: SET

        var name: p
        node label: none

where conditions:
                field name: has_cat
                operation: EQUALS
                value: true

        OR
                field name: has_dog
                operation: EQUALS
                value: true

        OR
                field name: has_parrot
                operation: EQUALS
                value: true

changed properties:
                name: has_pet
                value: true
```

3. Удаление элемента

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p) where p.age > 65 delete p
PARSED QUERY:

query type: DELETE

        var name: p
        node label: none

where conditions:
                field name: age
                operation: GREATER_THAN
                value: 65

deleted: p
```

4. Выборка элементов

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p:Pe) where p.name="E" and p.age=18 and p.c=3 or p.n<5 return p
PARSED QUERY:

query type: MATCH

        var name: p
        node label: Pe

where conditions:
                        field name: name
                        operation: EQUALS
                        value: "E"

                AND
                        field name: age
                        operation: EQUALS
                        value: 18

                AND
                        field name: c
                        operation: EQUALS
                        value: 3

        OR
                field name: n
                operation: LESS_THAN
                value: 5

return value: p
```

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p:Pe) where p.name="E" or p.age=18 or p.c=3 and p.n<5 or p.v contains "v" return p
PARSED QUERY:

query type: MATCH

        var name: p
        node label: Pe

where conditions:
                field name: name
                operation: EQUALS
                value: "E"

        OR
                field name: age
                operation: EQUALS
                value: 18

        OR
                        field name: c
                        operation: EQUALS
                        value: 3

                AND
                        field name: n
                        operation: LESS_THAN
                        value: 5

        OR
                field name: v
                operation: CONTAINS
                value: "v"

return value: p
```

5. Пример ошибки синтаксиса

```
PS C:\Users\eva\VScode\llp_lab2> ./main
match (p: Person) where p.age > 18
error: syntax error
```

Выводы

В ходе выполнения лабораторной работы изучены Bison и Flex, а также был реализован модуль для разбора запросов Cypher.