

Министерство науки и высшего образования Российской Федерации

**Федеральное государственное автономное образовательное
учреждение высшего образования**

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Отчёт к практическому заданию №2

по «Низкоуровнему программированию»

Выполнил: Группа Р33312 Хайкин О.И.

Преподаватель:

Кореньков Ю.Д.

Санкт-Петербург, 2023

Цели

Использовать средство синтаксического анализа по выбору, реализовать модуль для разбора некоторого достаточного подмножества языка запросов по выбору в соответствии с вариантом формы данных. Должна быть обеспечена возможность описания команд создания, выборки, модификации и удаления элементов данных.

Вариант:

Базовый язык запросов: SQL

Задачи

Модификация репозитория

Первым делом была проведена модификация существующего с первого задания репозитория - выделение имплементации базы данных в отдельный модуль - database, выделение общего кода (API и имплементация утилит, ошибок, результатов и т.д.) в модуль common. Затем были созданы 2 новых модуля - parser и client. parser представляет из себя библиотеку, отвечающую за, соответственно, парсинг, а client отвечает за исполняемую программу, принимающую ввод пользователя и вызывающую парсинг-библиотеку.

Подключение Bison и Flex

Для парсинга были использованы GNU Bison и Flex. Их включение в проект осуществлено с помощью пакетов Bison и Flex для CMake.

Имплементация парсера и дерева разбора

После всех этих настроек, был имплементирован сам сабсет языка SQL и дерево разбора.

Подготовка отчёта

С информацией, полученной от стресс-тестов, можно было приступить к подготовке этого отчёта

Описание работы

Модуль парсера состоит из Flex и Yacc файлов для лексера и парсера соответственно, а также набор заголовков и с-файлов, обеспечивающих деревья разбора и задающие API для работы с парсером.

Аспекты реализации

Дерево разбора

Элементы AST-дерева построены с использованием наследования и полиморфизма. Интерфейс элемента задаётся следующим образом:

```
struct i_ast_node {
    str_t name;
    bool (*equals_impl)(struct i_ast_node *self, struct i_ast_node
*other);
    void (*print_at_level_impl)(struct i_ast_node *self, size_t
current_level);
    void (*destroy_impl)(struct i_ast_node *self);
};
```

Имплементации обеспечивают функционал, представляя различные типы вершин дерева.

API модуля

Вызов к bison-парсеру обернут в утилитные функции, создающие API, соответствующий код-базе из первой части работы - функции возвращают result_t, используют последний аргумент как “полезный” результат и т.д. Основные предоставляемые функции выглядят следующим образом:

```
result_t parse_stdin(struct i_ast_node **result);
result_t parse_string(str_t str, struct i_ast_node **result);
```

Функция parse_stdin использует режим ввода из стандартного потока, доступный в bison-парсере, и предназначена для клиента, который поддерживает пользовательский ввод.

Функция parse_string выполняет парсинг выражения из строки и предназначена в основном для тестов.

Результаты

Артефакты

В результате сборки программы создаются следующие артефакты:

- Файл библиотеки, предназначенный для линковки с пользовательской программой.
- Исполняемые файлы тестов, линкующиеся с файлом библиотеки
- Исполняемый файл клиента, линкующийся с файлом библиотеки

Пример работы клиента

Ввод

```
SELECT FROM aboba JOIN amogus ON aboba.color = amogus.color WHERE  
aboba.imposter = "SUS";
```

Вывод

STATEMENT:

SELECT STATEMENT:

FROM:

TABLE ID:

ID: aboba

JOINS LIST:

JOIN ON:

TABLE ID:

ID: amogus

PREDICATE:

OPERATOR EXPR:

MEMBER EXPR:

MEMBER ID:

TABLE ID:

ID: aboba

COLUMN ID:

ID: color

COMPARISON OPER: EQ

MEMBER EXPR:

MEMBER ID:

TABLE ID:

ID: amogus

COLUMN ID:

ID: color

WHERE:

PREDICATE:

OPERATOR EXPR:

```
MEMBER EXPR:
  MEMBER ID:
    TABLE ID:
      ID: aboba
    COLUMN ID:
      ID: imposter
COMPARISON OPER: EQ
LITERAL:
  STRING: "SUS"
```

Выводы

В результате выполнения задания проект пополнился двумя модулями - библиотекой для парсинга и демо-версией клиента, реализующих парсинг сабсета языка SQL и обеспечивающих возможность описания команд создания, обновления, выборки и удаления элементов, а также создания и удаления таблиц.