Национальная научно-образовательная корпорация ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа №2

по дисциплине
«Низкоуровневое программирование»

Вариант №2

Выполнил: Лебедев
Вячеслав Владимирович
Группа № P33312
Проверил:
Кореньков Юрий Дмитриевич

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: язык SQL

Решение

Программа состоит из двух частей:

- библиотеки, предоставляющей функции для синтаксического разбора содержимого файлов или строк
- тестовой программы командной строки, использующей библиотеку для разбора пользовательских команд и выводящую дерево разбора в стандартный вывод

Библиотека

Для синтаксического разбора были выбраны утилиты кодогенерации Flex и Bison, использующие спецификацию, описанную в файлах lexer.l и parser.y соответственно.

В файле lexer.l были описаны правила токенезации для создаваемого подмножества языка SQL. В основные разделы вошли: литеральные значения, пользовательские имена, ключевые слова, символы.

В файле parser.y описаны правила синтаксиса выбранного подмножества языка SQL. А также описаны правила построения синтаксического дерева, структура которого и функции для его создания/освобождения описаны в файлах ast.h, ast.c.

Сами узлы абстрактного синтаксического дерева представляют собой структуру, содержащую union на каждый тип узла. Узлы выделяются на куче, поэтому после работы с деревом, его необходимо освободить при помощи функции FreeAstNode, проходящую рекурсивно по дереву и освобождающую каждый узел.

Тестирующая программа

Программа получает на вход аргумент командной строки – название файла, из которого будет считываться скрипт для разбора, в том числе можно указать stdin для считывания со стандартного ввода.

Далее используется функция библиотеки для создания абстрактного синтаксического дерева на основе содержимого файла. Функция PrintAst рекурсивно проходит по дереву и выводит каждый узел на стандартный вывод.

Пример скрипта на подмножестве языка SQL, который покрывает все конструкции и запросы:

```
CREATE TABLE user (
    id INT32,
    account FLOAT32,
    is_banned BOOLEAN
);

SELECT user.id, user.account, user.is_banned FROM user
JOIN subscription ON subscription.user_id = user.id
JOIN service ON service.id = subscription.service_id
WHERE user.account >= 2.0 AND user.is_banned = FALSE;

INSERT INTO user (id, account, is_banned)
VALUES (1, 0.0, FALSE);

DELETE FROM user
WHERE user.account >= 2.0 AND user.is_banned = FALSE;

UPDATE user SET
account = 100.0,
is_banned = FALSE
WHERE user.account = 0.0 AND user.is_banned = TRUE;

DELETE TABLE user;
```

Результат работы программы:

```
LIST ITEM
  CREATE TABLE [TABLE: user]
    LIST ITEM
      COLUMN DECLARATION [NAME: id, TYPE: INT32]
    LIST ITEM
      COLUMN DECLARATION [NAME: account, TYPE: FLOAT32]
    LIST ITEM
      COLUMN DECLARATION [NAME: is_banned, TYPE: BOOLEAN]
LIST ITEM
  SELECT FROM [TABLE: user]
    SELECTOR LIST
      LIST ITEM
        COLUMN REFERENCE [TABLE: user, COLUMN: id]
```

```
      LIST ITEM
        COLUMN REFERENCE [TABLE: user, COLUMN: account]
      LIST ITEM
        COLUMN REFERENCE [TABLE: user, COLUMN: is_banned]
    JOIN LIST
      LIST ITEM
        JOIN ON [TABLE: subscription]
          COLUMN REFERENCE [TABLE: subscription, COLUMN: user_id]
          COLUMN REFERENCE [TABLE: user, COLUMN: id]
      LIST ITEM
        JOIN ON [TABLE: service]
          COLUMN REFERENCE [TABLE: service, COLUMN: id]
          COLUMN REFERENCE [TABLE: subscription, COLUMN: service_id]
    WHERE
      CONDITION [TYPE: AND]
        CONDITION [TYPE: COMPARE]
          COMPARE [TYPE: GREATER EQUALS]
            COLUMN REFERENCE [TABLE: user, COLUMN: account]
            FLOAT [2.000000]
        CONDITION [TYPE: COMPARE]
          COMPARE [TYPE: EQUALS]
            COLUMN REFERENCE [TABLE: user, COLUMN: is_banned]
            BOOL [FALSE]
LIST ITEM
  INSERT INTO [TABLE: user]
    COLUMNS LIST
      LIST ITEM
        STRING [id]
      LIST ITEM
        STRING [account]
      LIST ITEM
        STRING [is_banned]
    VALUES LIST
      LIST ITEM
        INT [1]
      LIST ITEM
        FLOAT [0.000000]
      LIST ITEM
        BOOL [FALSE]
LIST ITEM
  DELETE FROM [TABLE: user]
    CONDITION [TYPE: AND]
      CONDITION [TYPE: COMPARE]
        COMPARE [TYPE: GREATER EQUALS]
          COLUMN REFERENCE [TABLE: user, COLUMN: account]
          FLOAT [2.000000]
      CONDITION [TYPE: COMPARE]
        COMPARE [TYPE: EQUALS]
          COLUMN REFERENCE [TABLE: user, COLUMN: is_banned]
          BOOL [FALSE]
LIST ITEM
  UPDATE [TABLE: user]
    UPDATE LIST
      LIST ITEM
        SET [COLUMN: account]
          FLOAT [100.000000]
```

```
      LIST ITEM
        SET [COLUMN: is_banned]
          BOOL [FALSE]
    WHERE
      CONDITION [TYPE: AND]
        CONDITION [TYPE: COMPARE]
          COMPARE [TYPE: EQUALS]
            COLUMN REFERENCE [TABLE: user, COLUMN: account]
            FLOAT [0.000000]
        CONDITION [TYPE: COMPARE]
          COMPARE [TYPE: EQUALS]
            COLUMN REFERENCE [TABLE: user, COLUMN: is_banned]
            BOOL [TRUE]
LIST ITEM
  DELETE TABLE [TABLE: user]
```

Выводы

В процессе выполнения лабораторной работы удалось ознакомиться с такими средствами синтаксического анализа, как Flex и Bison, а также реализовать подмножество языка SQL и создание абстрактного синтаксического дерева на языке Си.