C语言词法分析程序的设计与实现

娄司宇

2021.10.10

实验环境

本实验在Ubuntu 20.04下，使用g++ 9.3.0，cmake 3.21.0开发。 ## 程序设计 本程序主要由3个模块组成 ### token 每一个token应当是以<classification, attribute>的形式表示

classification 表示token的属性; attribute表示token对应的属性值

设计有以下分类

```
enum Classification
{
    TOKEN_INVALID,      //  token
    TOKEN_EOF,          //
    TOKEN_KEYWORD,      //
    TOKEN_IDENTIFIER,   //
    TOKEN_OPERATOR,     //
    TOKEN_STRING,       //
    TOKEN_CHARACTER,    //
    TOKEN_INTEGER,      //
    TOKEN_FLOAT         //
};
```

设计对应的属性值

```
union token
{
    //
    Keyword keyword;
    //       ,
    std::string* name;
    //
    Operator operator_;
    //        ,
    std::string* stringValue;
    //
    i64 integerValue;
    //
```

```
    double floatValue;
}u;
```

这样设计的好处在于每一个token仅会占用有限的空间，构造的时间、空间开销均比较小

token还维护了一个位置信息`Location`，储存了所属的文件、行、列，这对于调试以及记录信息有重大意义。

此外，token还应包含构造各类token的方法，具体实现见代码，限于篇幅不进行具体描述。

lexer

lexer是词法分析的核心部分。

一个lexer对象应当可以打开一个文件，每次调用`getNextToken()`方法后，从文件中读出一个token并返回。如果遇到一些错误，应当报

为了实现以上功能，lexer包含以下部分+buffer+识别各类token的子函数+符号表+输出错误信息的函数
+其他变量

具体信息见lexer的实现

scan

整个程序的入口部分，读取要处理的文件名，调用lexer对象得到单词流，并将得到的结果格式化输出到目标文件中。此外输出文件的统计信

lexer实现

输入缓冲区

符号表

识别token

错误处理

测试的说明和描述

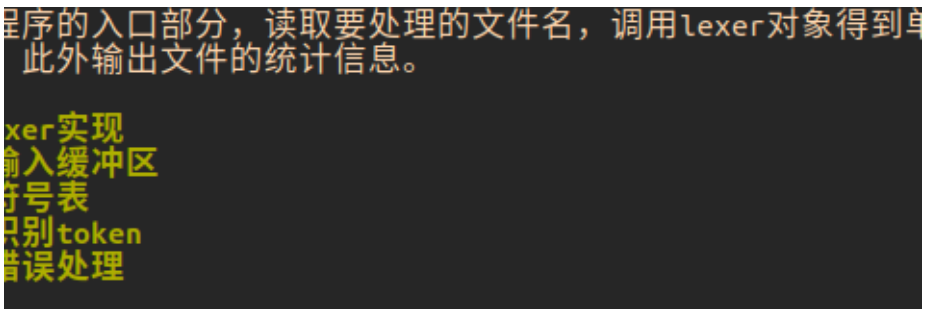

Figure 1: example img

析构函数！！