# 词法分析程序设计报告

## 需求分析

开发环境：Windows10 & WSL (Ubuntu 20.04.1 LTS) & VSCode & Remote – WSL & g++ (Ubuntu 9.3.0-10ubuntu2) 9.3.0

功能需求：设计实现C语言的词法分析程序

1. 识别单词符号
2. 识别并跳过注释
3. 统计行数，字符数，关键字数
4. 识别词法错误并报告
5. 对错误进行恢复

## 总体设计

1. 数据结构设计：
   1. Token类储存解析出的Token信息
   2. Lexer类负责执行词法分析
2. 总体结构设计：
   1. 初始化模块：

功能：执行Lexer功能的初始化

接口：Lexer::Lexer(std::string path)

* 1. 解析模块

功能：通过DFA对字符序列进行解析并记录Tokens

接口：Lexer::analyze()

* 1. 字符指针操作模块

功能：更新当前字符及行列数等信息

调用关系：由解析模块调用

接口：Lexer::rollBack(), Lexer::nextChar()

* 1. 输出模块

功能：输出解析结果

接口：Lexer::outputTokens(), Lexer::outputExceptions(), Lexer::outputCounts()

1. 用户接口设计：
   1. Lexer::Lexer(std::string path); //初始化Lexer
   2. void Lexer::analyze(); //执行词法分析
   3. void Lexer::outputTokens(); //输出解析出的token
   4. void Lexer::outputExceptions(); //输出词法错误
   5. void Lexer::outputCounts(); //输出统计结果

## 详细设计

1. Lexer类
   1. 接口描述
      1. Lexer::Lexer(std::string path)

参数为要进行词法分析的文件路径

读入该文件到缓冲区，设置初始行列数等信息

* + 1. void Lexer::nextChar()

将下一个字符添加至currentWord并更新currentChar及字符位置行列数

* + 1. void Lexer::rollBack()

回滚currentWord, currentChar及字符位置和行列数

* + 1. void Lexer::analyze()

对缓冲区中的文件进行解析

* + 1. void Lexer::outputTokens()

输出解析出的token

* + 1. void Lexer::outputExceptions()

输出解析出的Error和Warning

* + 1. void Lexer::outputCounts()

输出解析出的统计数据

* 1. 功能描述：进行词法分析
  2. 所用数据结构
     1. std::string filePath

要进行词法分析的文件路径

* + 1. std::string program

文件中所有的字符

* + 1. std::string currentWord

当前解析的单词

* + 1. char currentChar

当前解析的字符

* + 1. size\_t pos

当前字符所在的位置

* + 1. size\_t line

当前字符所在的行数

* + 1. size\_t col

当前字符所在的列数

* + 1. std::vector<Token> tokens

解析出的所有token

1. Token类
   1. 接口描述
      1. Token(size\_t line, size\_t col, std::string type, std::string value, std::string info)

参数为该token所在的行数，列数，类型，值以及附加信息

使用以上信息创建一个Token对象，记录数据

* + 1. std::ostream &operator<<(std::ostream &out, const Token &token)

重载了<<运算符方便输出Token

* 1. 功能描述：储存token信息
  2. 所用数据结构：
     1. size\_t m\_line

该token所在行数

* + 1. size\_t m\_col

该token所在列数

* + 1. std::string m\_type

该token的类型

* + 1. std::string m\_value

该token的值

* + 1. std::string m\_info

该token的附加信息

## 源程序清单

源程序：

include/Lexer.h

src/Lexer.cpp

main.cpp

构建：mkdir bin && cd bin && cmake .. && make

运行：./Lexer [filepath]

## 程序测试

测试环境：Windows10 WSL (Ubuntu 20.04.1 LTS)

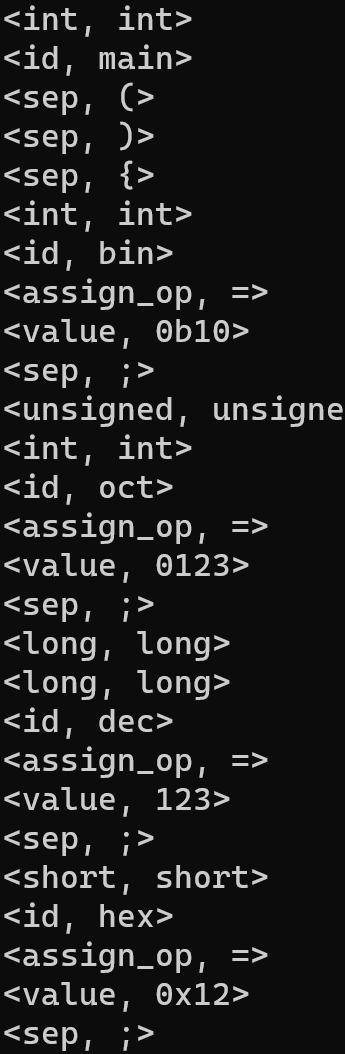
测试的功能：词法分析Token输出，字符统计输出，错误报告输出

测试用例：data/test.c

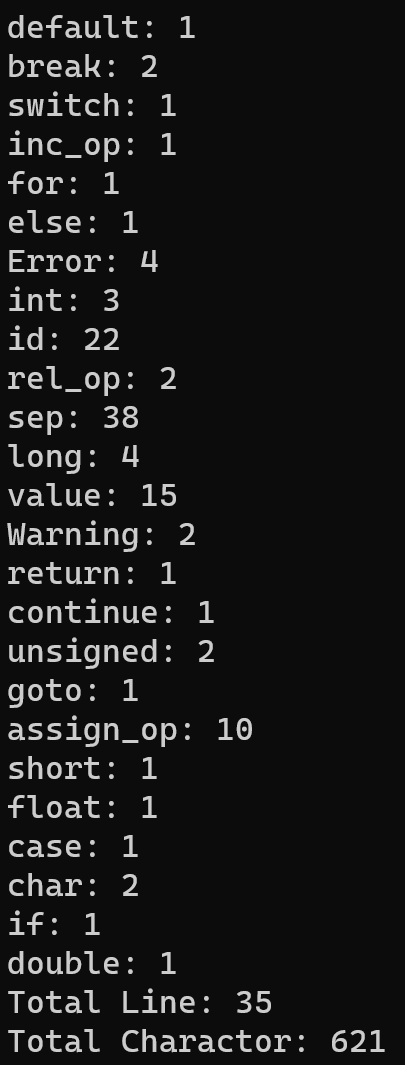
预期结果：能够对测试用例中所有的字符进行词法分析并能报告错误，进行字符统计

测试结果：

能够正确分析出Token



能够统计字符数，行数，关键字数等信息



能够报告源程序中存在的词法错误

