基于lex的c语言的词法分析器

1. 实验目的
2. 能够掌握编译原理的基本理论,掌握有穷自动机。
3. 了解并掌握lex，并用lex自动生成词法分析程序。
4. 实验环境

Windows

1. 实验内容
2. 待分析的简单语言的词法
3. 保留字：

int，double，float，long，main，return，void，while，do，const，template，class，if，else，else if，case，include，using，namespace，try，catch，define，

关键字均为小写

1. 控制符

控制输出浮点数的位数

%.2f %.3lf %lld

1. 数字（整数和浮点数）

num=[+-]?({digit}+|0|{digit}\*[.]{digit}+)

1. 错误标识符

以数字开头的字符串

1. 标识符

字母或下划线开头的包括字母数字的字符串

Id=({letter}|[\_])({letter}|{digit}|[\_])\*

1. 运算符

"\*"|"/"|"+"|"-"|">"|"<"|">="|"<="|"=="|"!="|"="|"+="|"-="|"\*="|"\="|"%"|"&&"|"<<"|">>"|"++"|"--"|"?:"

1. 界符

","|";"|"("|")"|"{"|"}"|"#"|"<"|">"|"["|"]"

1. 空格

Whitespace=[" "\t]+（空格不做处理）

1. 注释

两类：//或者/\* \*/

Zhushi1= \/\/{phrase}\*[^\n]\*

zhushi2= \/\\*{phrase}\*\\*\/

1. 其他情况视为出错

2.各类单词符号对应的种别码

由于各类中的符号比较多，所以只分为十类。

表1 各类单词对应的种别码

|  |  |
| --- | --- |
| 单词 | 种别码 |
| 保留字 | 1 |
| 控制符 | 2 |
| 数字 | 3 |
| 标识符 | 4 |
| 错误标识符 | 5 |
| 运算符 | 6 |
| 界符 | 7 |
| 注释 | 8 |
| 其他为出错 | 9 |

1. 词法分析程序的功能

输入：程序文本

输出：三元组（count，class，token）；

其中count为单词所在的行，class为单词类型，token为识别的字符串

1. 实验源代码(lex源代码)

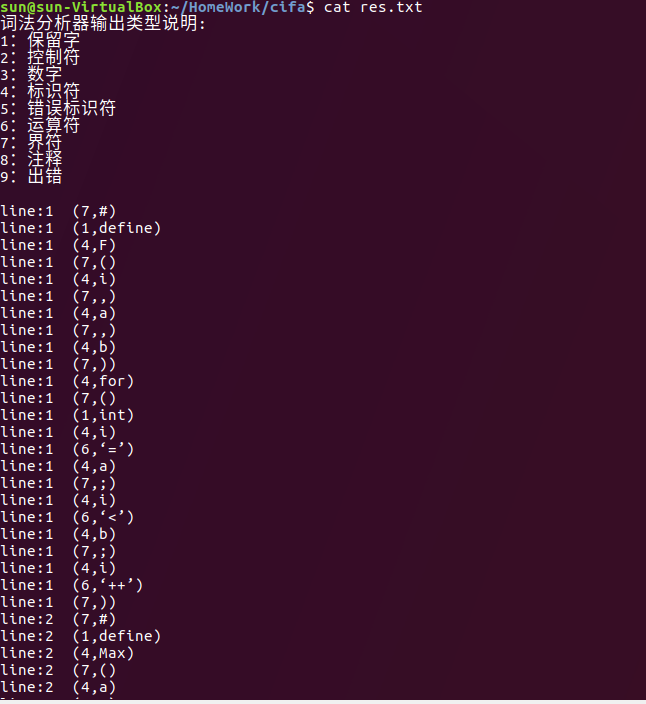
见压缩包文件夹“程序”中的

1. 运行结果

分析的代码：



分析结果如图：



1. 使用说明

在linux环境下安装lex 运行环境 sudo apt-get install yacc lex

1. 心得体会
2. 刚学lex的时候觉得很难，感觉步骤很繁琐，过程似乎也很复杂，但是入门了之后发现其实还是很简单学会的，然后有一种深深的满足感。
3. 自己用c++也写过词法分析器，但是实现同样的功能lex简洁得多。
4. 通过用lex自己对正则式的概念也更加理解深刻了一点，也很赞叹前人能开发lex这么一款软件觉得佩服。
5. 在编程过程中，自己遇到了很多小问题。第一：在lex的规则部分，之前一直不理解判断过程是按照顺序的，所以之前自己一直把保留字的判断放在标识符下面，结果就是所有的保留字都被判断成标识符。第二：在数字和运算符判断遇到了问题，比如a=a+3，自己写的分析器把+3判断成一个正数3，改进后也解决了这个问题。
6. 通过这个软件，自己对自动机的概念似乎也理解更透彻了。写完之后觉得这个软件还蛮好玩。