# 词法分析程序实验报告

----张宇奎 16211103

## 实验目的

通过本实验理解词法分析的整个过程,处理对象和处理的结果，更加清楚了解词法分析的过程。

通过动手实践,使学生对构造编译系统的基本理论、编译程序的基本结构有更为深入的理解和掌握，使学生掌握编译程序设计的基本方法和步骤，能够设计实现编译系统的重要环节。增强编写和调试程序的能力。

## 实验步骤

### 整理要求

**必做：**

1. 分析所给的PL0文法，确定关键字，分界符，运算符，常量。
2. 对输入的代码段进行分析，输出每个单词的分类，并输出该单词的值，整数必须用二进制表示。
3. 常数默认为整数，必须区分分界符和运算符。
4. 运算符必须区分单字符运算符和双字符运算符。
5. 对不规范的单词有错误提示。
6. 要求文件输入。

**选做：**

1. 常数里可以识别小数。
2. C0文法

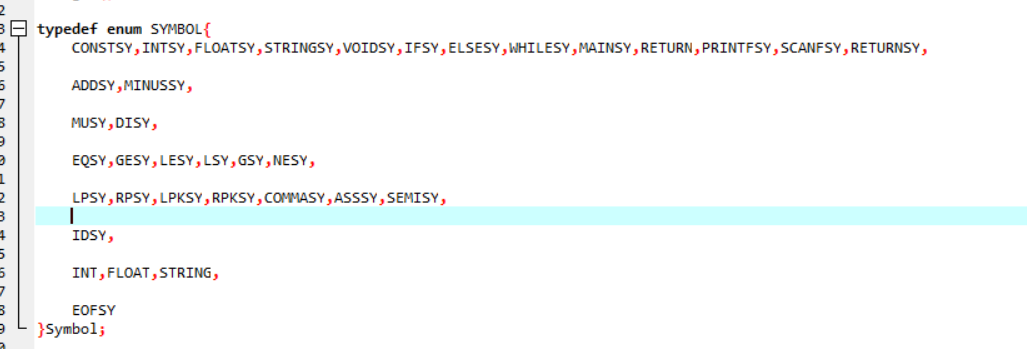
**额外完成部分:**

1. 支持 string 类型，在后续的内容中，string,int,float 将被作为三种可以在编译器中使用的类型。
2. 支持escapeCharacter,即转义字符，string 中的转义字符将会被正确的处理
3. 支持一种类型的注释/\* \*/这种风格的注释允许被出现在合法的程序中。

### 实际步骤

1. 根据一符一类原则，完成词法分析器的主要部分。

编译器支持的所有token



这一步实际上完成了大部分工作，其中较为复杂的逻辑有注释的处理和转义字符的处理。

1. 编写错误处理方法。使用链表的数据结构，这样完成的错误处理可以追踪到很后期的编译器阶段，完成多遍扫描的任务。
2. 编写内存分配模块，保证内存安全，出于实现上的方便以及编译器对于内存要求不是很多，就不进行内存的动态释放了。

## 代码组织

代码按照模块组织，词法分析本身是编译器的一个模块，因为这次实验单独抽取出来进行检查。

Makefile 中有生成的规则。

生成方法 :

make lex

使用方法 ：

lex <文件名>

## 实验感想

通过这次的实验，学习到了很多编译原理的知识。

词法分析器虽然简单，但却包含了许多编译器的思想和理念。

在完成实验的过程中也遇到了很多的问题，解决了不少bug,对于提升编程水平也有非常大的作用。