**《编译系统设计实践》**

实验项目一：词法分析实验

学号： 031502248

姓名： 左文航

年级： 15级

学院： 数计学院

专业： 计算机

实验时间：2016－2017学年第一学期

任课教师：刘秉瀚

目录

[一、实验目的 1](#_Toc503539872)

[二、实验内容 1](#_Toc503539873)

[三、程序设计与实现 1](#_Toc503539874)

[1、功能描述： 1](#_Toc503539875)

[2、程序结构描述： 1](#_Toc503539876)

[实现 1](#_Toc503539877)

[3、符号表的设计和结构 3](#_Toc503539878)

[4、程序流程图： 4](#_Toc503539879)

[四、程序测试 5](#_Toc503539880)

[样例一 5](#_Toc503539881)

[样例二 5](#_Toc503539882)

[样例三 6](#_Toc503539883)

[样例四 6](#_Toc503539884)

[五、实验小结 7](#_Toc503539885)

[六、附 8](#_Toc503539886)

[1.char nextchar() 8](#_Toc503539887)

[2. void solve() 8](#_Toc503539888)

[3.int get\_state(char &ch) 9](#_Toc503539889)

[4.int get\_digit(char &now) 9](#_Toc503539890)

## 一、实验目的

将输入的源程序进行划分，给出基本符号（token）的序列，并掠过注解和空格等分隔符号。基本符号是与输入的语言定义的词法所规定的终结符。

## 二、实验内容

本实验要求学生编制一个读单词过程，从输入的源程序中，识别出各个具有独立意义的单词，即基本保留字、标识符、常数、运算符、分隔符五大类。并依次输出各个单词的内部编码及单词符号自身值。（遇到错误时可显示“Error”，然后跳过错误部分继续进行）

## 三、程序设计与实现

### 1、功能描述：

从输入的源程序中，识别出各个具有独立意义的单词，即基本保留字、标识符、常数、运算符、分隔符五大类,并且能跳过单行的注释,并依次输出各个单词的内部编码及单词符号自身值。遇到错误时显示“Error:Undefine xxx”，然后跳过错误部分继续进行。

### 2、程序结构描述：

### 实现

采用C++ devC5.11 工具实现 pascal语法

#### 各函数及功能说明

(1).Char nextchar()

模拟缓冲区,取读下一个字符。

(2).void fail(char \*str,char &now,int i)

发生错误,输出错误信息,并排错。

Str表示当前处理的文法的出错前的字符串的长度,now表示当前字符,i表示当前处理到输入文法的位置。

(3).int get\_state(char &ch)

读取当前的字符,忽略空白符、制表符和回车判断下一步是的操作,返回下一步操作的序号。

Ch表示当前输入的字符

(4).void Print(char \*str1,char \*str2)

输出识别到的单词和类别。

Str1表示识别到的单词,str2表示识别到的类别

(5).int get\_word(char &now)

获取关键字或者标识符,返回下一步操作序号。

Now表示当前读入的字符

(6).int get\_zhushi(char &now)

当前操作为注释,进行忽略,返回下一步操作序号。

Now表示当前读入的字符。

(7).int get\_relop(char &pre)

当前操作为获取逻辑运算符,返回下一步操作序号。

Pre表示当前读入的字符。

(8).int get\_digit(char &now)

当前操作为获取数字,返回下一步操作序号。

Now表示当前读入的字符。

(9).int get\_ariop(char &pre)

当前操作为获取算符运算符,返回下一步操作。

Pre表示当前读入的字符

(10).int get\_other(char &now)

获取其他的合法运算符,如: (,),等,返回下一步操作

Now表示当前读入的字符。

(11). void init\_1()

初始化需要用到的变量,并且预处理出关键字。

(12). bool isoperator(char ch)

判断是不是运算符

(13). bool isariop(char ch)

判断是不是逻辑运算符

(14). bool isword(char ch,bool f)

判断是不是单词,true:首字母

(15). bool isletter(char ch)

判断是不是字母

(16).bool isdigit(char ch)

判断是不是数字

(17). iskeyword(char \*str)

判断是不是关键字

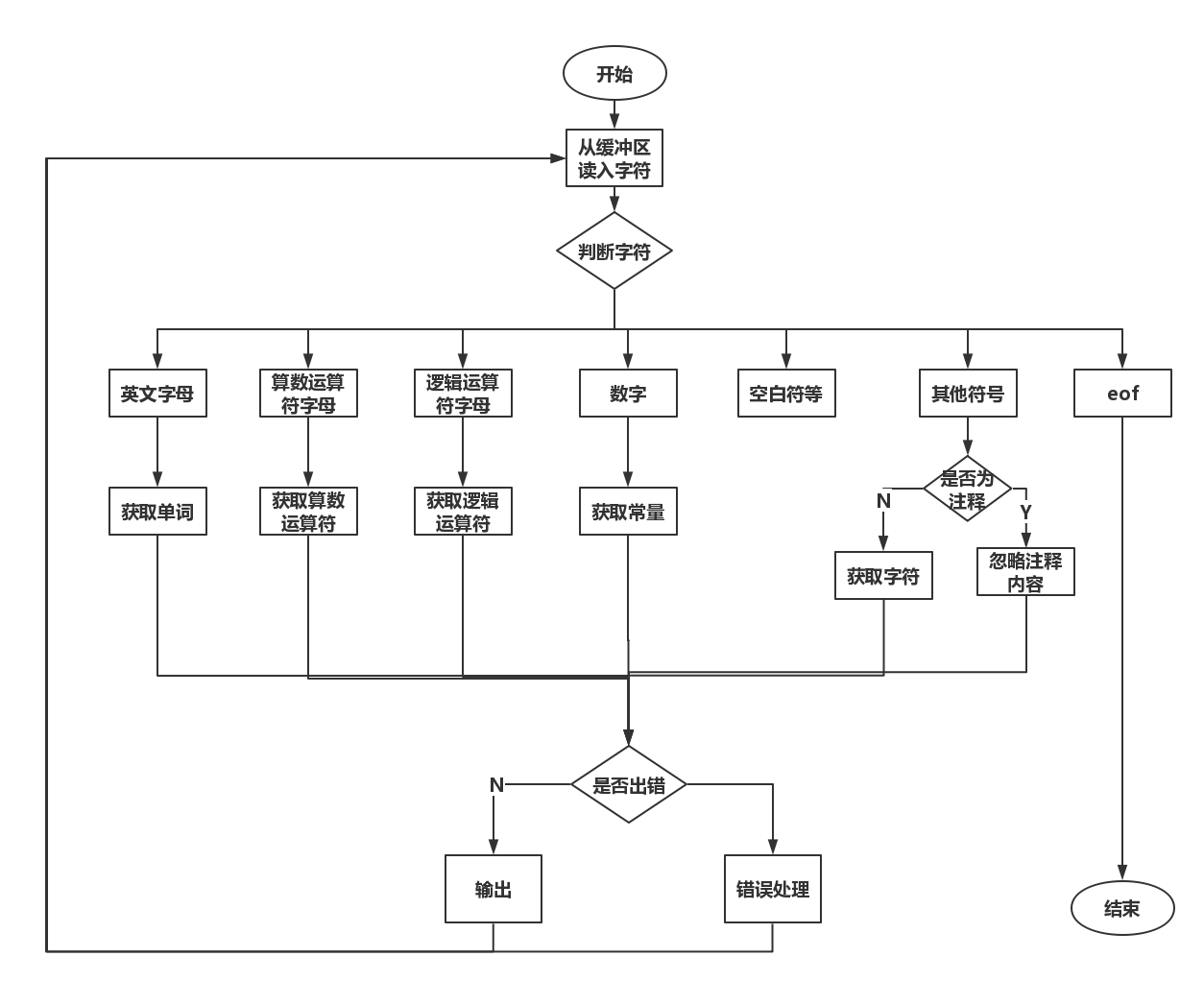
(18). bool isspace(char &now)

是不是空白符

### 3、符号表的设计和结构

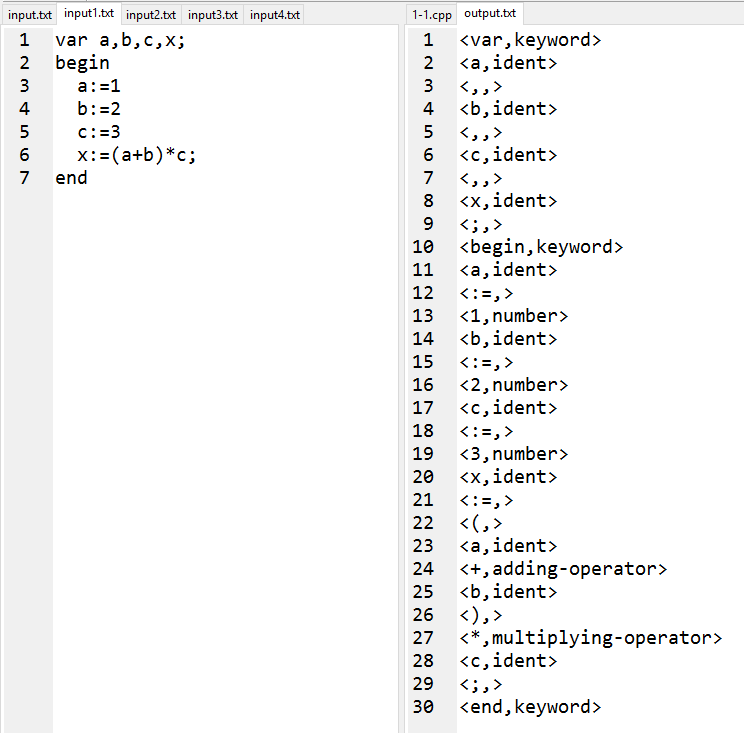
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 关键字 | |  | 算数运算符 |  |
| const | | 1 | + | 13 |
| var | | 2 | - | 14 |
| call | | 3 | \* | 15 |
| begin | | 4 | / | 16 |
| end | | 5 | 逻辑运算符 |  |
| if | | 6 | = | 17 |
| then | | 7 | <> | 18 |
| while | | 8 | < | 19 |
| do | | 9 | > | 20 |
| odd | | 10 | <= | 21 |
| procedure | | 11 | >= | 22 |
| to | | 12 | =: | 23 |
| 分隔符 |  | |  |  |
| ( | 24 | | 常量 | 28 |
| ) | 25 | | 标识符 | 29 |
| ; | 26 | |  |  |
| , | 27 | |  |  |

### 4、程序流程图：

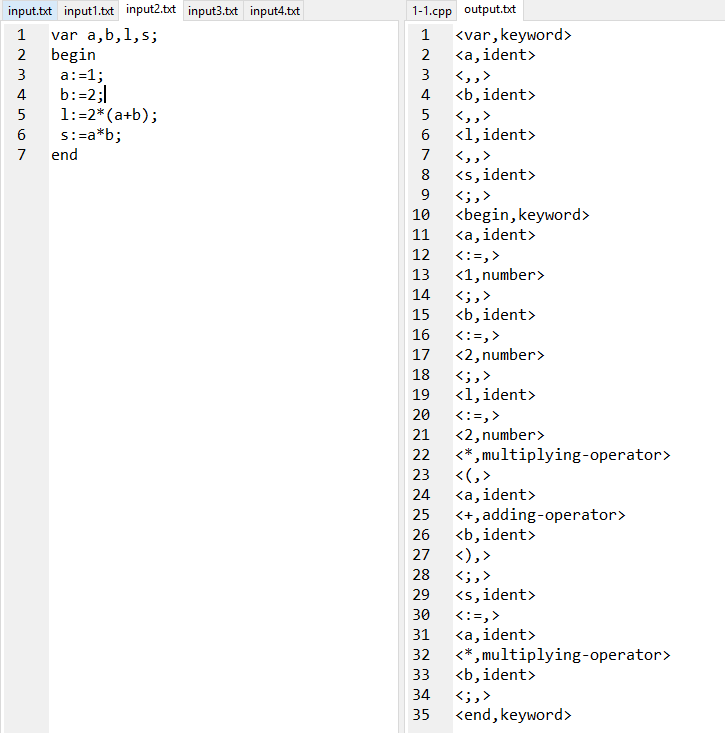


## 四、程序测试

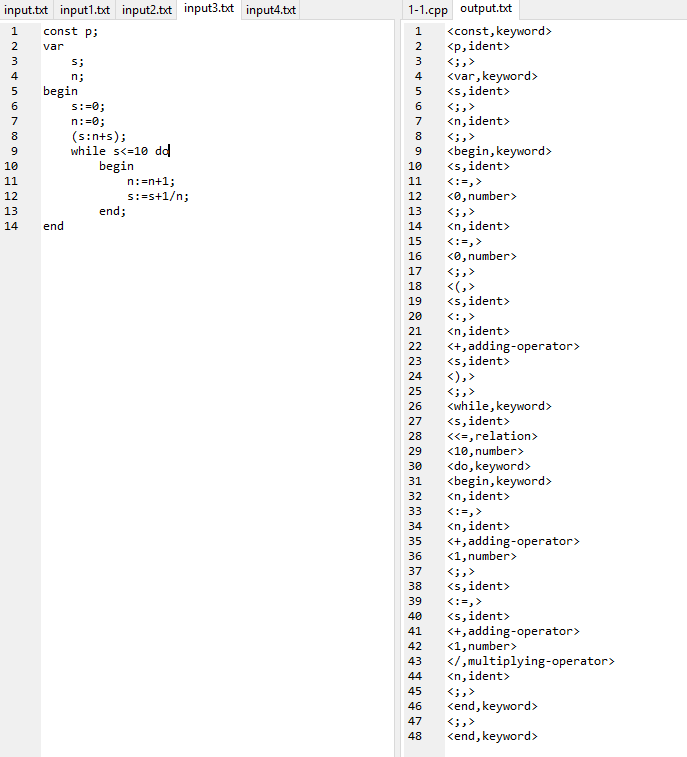
### 样例一



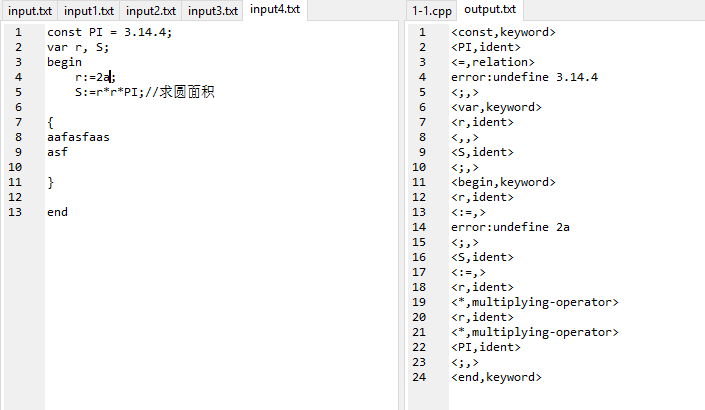
### 样例二



### 样例三



### 样例四



## 五、实验小结

通过课程设计，我们树立了正确的设计思想，巩固了所学编程语言基本知识，学习到如何综合运用所学的理论知识，进一步理解到高级语言在计算机中的执行过程，加深了对编译原理中第三章词法分析部分更加深入的了解，理解词法分析在编译程序中的作用；掌握了课程设计的一般方法与步骤，设计和构造编译程序的基本原理和常用的编译技术，以及编译程序的实现方法和技，提高了自己的编程能力。

因为是一个人做,所以所有事情都要亲力亲为,工作量比较大,一步一步的查找资料,然后再实现编程,花了不少时间,不过,收获也十分的大。

## 六、附

### 1.char nextchar()

//获取下一个字符,模拟缓冲区

char nextchar()

{

if (rollback != eof) //是不是需要回退一次

{

char ch = rollback;

rollback = eof;

return ch;

}

//缓冲区

if (work1\_buf[p] == eof)//eof

{

if (p >= MAXN)//缓冲区尾部,继续读入

{

int i = 0; p = 0;

while ((i<MAXN) && (~fscanf(fp\_input, "%c", &work1\_buf[i]))) ++i;

work1\_buf[i] = work1\_buf[i + 1] = eof;//缓冲区尾部给eof

}

else;

}

return work1\_buf[p++];

}

### 2. void solve()

void solve()

{

int state = 0;

char now;

while (true)

{

switch (state)

{

case 0: state = get\_state(now); break;//除去空白

case ARIOP: state = get\_ariop(now); break;//获取算数运算符

case RELOP: state = get\_relop(now); break;//获取逻辑运算符

case DIGIT: state = get\_digit(now); break;//获取数字

case WORD: state = get\_word(now); break;//获取关键字或者标识符

case zhushi:state = get\_zhushi(now); break;//获取注释

case OTHER: state = get\_other(now); break;//获取其他

case eof: return; //eof 退出

}

}

}

### 3.int get\_state(char &ch)

int get\_state(char &ch)

{

ch = nextchar();

if (isspace(ch)) return 0;//忽略回车和空格分隔符

else if (isariop(ch)) return ARIOP; //获取算数运算符

else if (isrelop(ch)) return RELOP; //获取逻辑运算符

else if (isdigit(ch)) return DIGIT;//获取数字常数

else if (isword(ch, true)) return WORD; //获取关键字或者标识符

else if (ch == eof) return eof;

else return OTHER;

}

### 4.int get\_digit(char &now)

//获取数字

int get\_digit(char &now)

{

char str[200] = { '\0' };

int i = 0;

bool flag = false;

str[i++] = now;

while (true)

{

now = nextchar();

if (isdigit(now)) str[i++] = now;

else

{

if (now == '.')

{

str[i++] = now;

flag = true;

}

else if (isspace(now) || now == ';' || now == ')' || isoperator(now))

{

rollback = now;

}

else

{

str[i++] = now;

fail(str, now, i);

}

break;

}

}

if (flag && !mistake)//获取小数

{

while (true)

{

now = nextchar();

if (isdigit(now)) str[i++] = now;

else if (isspace(now) || now == ';' || now == ')' || isoperator(now))

{

rollback = now;

break;

}

else

{

str[i++] = now;

fail(str, now, i);

break;

}

}

}

Print(str, (char\*)"number");

return 0;

}