



中山大學
SUN YAT-SEN UNIVERSITY

中山大学计算机学院

本科生实验报告

编译器构造实验

实验	实验1-C语言词法分析器	专业 (方向)	计算机科学与技术 (超算方向)
学号	19335206	姓名	韦媛馨
Email	3366875159@qq.com	完成日期	2022/3/17

- 一、实验目的
- 二、C语言词法规则
 - 1. 标识符
 - 2. 数字常量
 - 3. 字符常量
 - 4. 字符串
 - 5. 关键字
 - 6. 运算符
 - 7. 界符
- 三、自动机设计
- 四、实验过程及关键代码
 - 1. 程序主体架构
 - 2. 关键函数
 - 3. 其他辅助函数
- 五、实验结果

一、实验目的

设计C语言词法分析器，

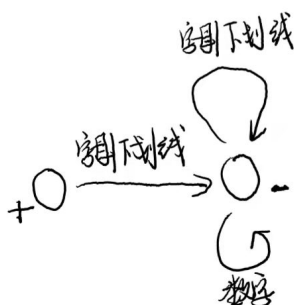
输入：一个C语言源程序文件 `demo.c`，过滤掉无用符号，判断源程序中单词的合法性，并分解出正确的单词，以二元组形式存放在文件中。

输出：一个文件 `tokens.txt`，该文件包括每一个单词及其种类枚举值，每行一个单词

二、C语言词法规则

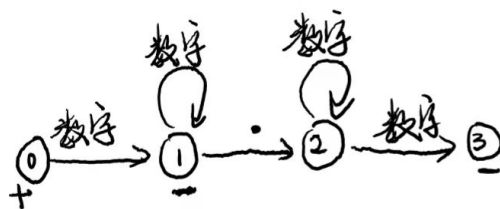
1. 标识符

标识符是由字母、下划线和数字组成的字符序列，允许由字母或下划线开头，但不允许由数字开头。其DFA如下：



2. 数字常量

常量是一种在程序中其值保持不变的量，C语言中常量分为数字常量和字符常量两类。数字常量又分为整形常量和浮点型常量。其DFA如下：



3. 字符常量

字符常量是由一对单引号(' ')括起来的一个字符组成的常量，如 'a'。

4. 字符串

字符串是由一对双引号(" ")括起来的一串字符组成的常量，如 "a", "abcDE"。字符串常量存放在内存中占有的字节数是字符个数加1，每个字符串存放在内存中都有一个结束符 '\0'。

5. 关键字

关键字是一种具有特定含义的标识符。关键字又称保留字，这些标识符是系统已经定义过的，不能再定义，需要加以保留。在本实验中，根据ANSI C标准，共定义了32种关键字，其中主要可分为三类：

- (1)标识类型的关键字，如 `int`, `char`, `float`, `register`, `auto` 等；
- (2)标识控制流的关键字，如 `goto`, `return`, `break`, `continue` 等；
- (3)其他关键字，如 `sizeof`, `void` 等

在程序中用一个数组 `ReserveWord[]` 来定义这些关键字：

```
1 static char ReserveWord[32][20] = {
2     "auto", "break", "case", "char", "const", "continue",
3     "default", "do", "double", "else", "enum", "extern",
4     "float", "for", "goto", "if", "int", "long",
5     "register", "return", "short", "signed", "sizeof", "static",
6     "struct", "switch", "typedef", "union", "unsigned", "void",
7     "volatile", "while"
8 };
```

6. 运算符

运算符是用来表示某种运算操作的一种符号，有的运算符用一个字符组成，也有的运算符由两个字符组成。在实验中定义了24种运算符，主要可分为以下几类：

- (1)算术运算，如 `+`, `-`, `*`, `/`, `+=`, `++`, `%` 等

- (2)比较运算，如>,>=,==,!=等
- (3)位移运算，如>>,<<
- (4)位运算，如&,&|
- (5)逻辑运算，如&&,&&||

7. 界符

界是用来分隔多个变量、数据项、表达式等的符号，实验中共定义了9种界符，用数组 `Delimiter[]` 表示：

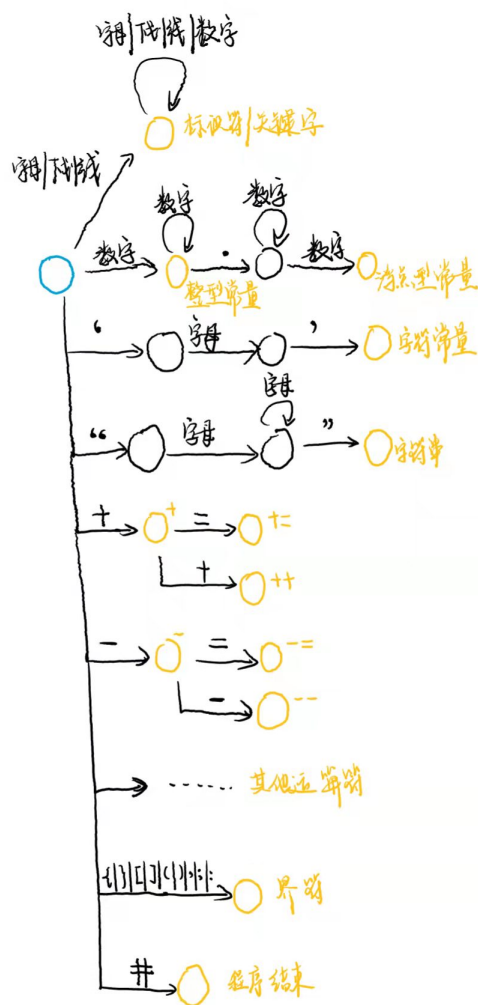
```
1 static char Delimiter[9][3]={
2     "{", "}", "[", "]", "(", ")", " ", " ", ";", ":"
3 };
```

根据上述介绍，实验中对C语言词法类型种别值的设计如下表：

标识符		数字常量		字符常量		字符串		关键字				运算符				界符	
A	00	123	01	'a'	02	"ok"	03	auto	04	int	20	+	36	<	48	{	60
Ch	00	100	01	'A'	02	"abc"	03	break	05	long	21	-	37	<=	49	}	61
temp	00	12.5	01	'c'	02	"lab"	03	case	06	register	22	*	38	=	50	[62
								char	07	return	23	/	39	==	51]	63
								const	08	short	24	+=	40	!=	52	(64
								continue	09	signed	25	-=	41	>>	53)	65
								extern	15	union	31	*=	42	<<	54	,	68
								float	16	unsigned	32	/=	43	&	55	;	69
								for	17	void	33	++	44	&&	56	:	70
								goto	18	volatile	34	--	45		57		
								if	19	while	35	>	46		58		
												>=	47	%	59		

三、自动机设计

DFA设计: (蓝色是开始状态, 橙色是终止态)



四、实验过程及关键代码

1. 程序主体架构

```

1 preprocess(source); //预处理, 取出无用的注释和字符
2 int p=0; //从头开始分析
3 int state=0; //设置初始状态
4 char token[20];
5
6 while(state!=-1){ //一直往下分析, 直到程序结束
7     state=scanner(source, token, &p); //实现有限自动机的功能
8     fprintf(fp_output, "<%s, %d>\n", token, state); //每次将识别出来的token及其种值码写入
9 }

```

2. 关键函数

```
1  /*扫描token并返回其种别码，实现DFA的功能*/
2  int scanner(char source[],char token[],int *pos){
3      int p=*pos;
4      int count=0;
5      int state;//种值码
6
7      ...//将token清零及过滤空格
8
9      //开头为字母：保留字或标识符
10     if(isLetter(source[p])){
11         token[count++]=source[p++];
12         while(isLetter(source[p])||isDigit(source[p])){
13             token[count++]=source[p++];
14         }
15         token[count]='\0';
16         state=searchReserve(token);//查表找到种别码
17     }
18
19     //开头为数字：数字常量
20     else if(isDigit(source[p])){
21         token[count++]=source[p++];
22         while(isDigit(source[p])){ //整型常量
23             token[count++]=source[p++];
24         }
25         if(source[p]=='.'){ //浮点型常量
26             do{
27                 token[count++]=source[p++];
28             }while(isDigit(source[p]));
29         }
30         state=1;
31     }
32
33     //开头为'：字符常量
34     else if(source[p]=='\'){
35         token[count++]='\'';
36         token[count++]=source[p+1];
37         token[count++]='\'';
38         p+=3;
39         state=2;
40     }
41
42     //开头为"：字符串
43     else if(source[p]=='\"'){
44         token[count++]=source[p++];
45         while(isLetter(source[p])){
46             token[count++]=source[p++];
47         }
48         token[count++]=source[p++];
49         token[count]='\0';
50         state=3;
51     }
52
53     //运算符：+ += ++
```

```

54     else if(source[p]=='+'){
55         token[count++]=source[p++];
56         if(source[p]=='='||source[p]=='+'){ //+= ++
57             token[count++]=source[p++];
58             state=source[p-1]==' '?40:44;
59         }
60         else { //+
61             state=36;
62         }
63     }
64
65     ...//其他运算符识别
66
67     //界符
68     else if(source[p]=='{'||source[p]=='}'||source[p]=='['||source[p]==']'||...){
69         token[0]=source[p++];
70         for(int i=0;i<9;++i){//查找界符表
71             if(strcmp(token,Delimiter[i])==0){
72                 state=i+60;//获得种别码
73                 break; //查到即退出
74             }
75         }
76     }
77
78     //程序结束
79     else if(source[p]=='#'){
80         token[count++]=source[p++];
81         state=-1;
82     }
83
84     //不能被以上词法分析识别，则出错
85     else{
86         printf("error:there is no exist %c \n", source[p]);
87         exit(-1);
88     }
89
90     *pos=p;
91     return state;
92 }

```

3. 其他辅助函数

`int isDigit()`: 判断是否为数字

`int isLetter()`: 判断是否为字母或下划线

`int searchReserve()`: 查找保留字并返回其种值码，若返回0则表示其为标识符而非保留字

`void preprocess()`: 编译预处理，取出无用的注释和字符

五、实验结果

cd到当前文件夹，编译：

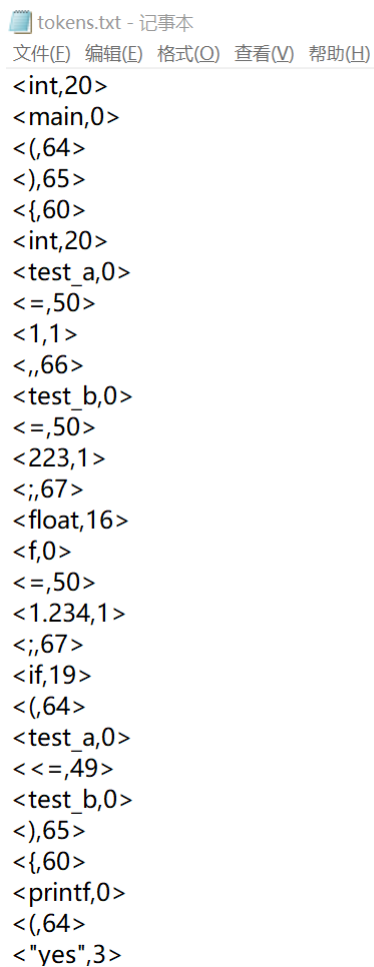
```
1 | gcc -o clang source.c
```

生成 `clang.exe`

运行（要确保输入文件 `demo.c` 位于同一目录下，否则会报错）：

```
1 | ./clang
```

运行结束后，在同一目录下会生成 `tokens.txt`，其内容如下：



```
tokens.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<int,20>
<main,0>
<(,64>
<),65>
<{,60>
<int,20>
<test_a,0>
<=,50>
<1,1>
<,,66>
<test_b,0>
<=,50>
<223,1>
<;,67>
<float,16>
<f,0>
<=,50>
<1.234,1>
<;,67>
<if,19>
<(,64>
<test_a,0>
<<=,49>
<test_b,0>
<),65>
<{,60>
<printf,0>
<(,64>
<"yes",3>
```

可见，该C语言词法分析程序根据所给的源代码程序，输出的是二元组：<单词符号, 种别码>；对于常数的形式，也是直接以字符串的形式表达。对照种别码可知，词法分析结果正确，该程序能正确识别出标识符、字符、字符串、整形常量、浮点型常量、运算符和界符，达到实验目的。