

词法分析程序

一、目的和内容

- 1、实验目的：通过完成词法分析程序，了解词法分析的过程
- 2、实验内容：用 C/C++ 实现对 PASCAL 的子集程序设计语言的词法识别程序
- 3、实验要求：将该语言的源程序,也就是相应字符流转换成内码，并根据需要是否对于标识符填写相应的符号表供编译程序的以后各阶段使用

二、程序设计语言的描述

程序设计语言的描述采用扩充的 BNF 表示：

<程序>::=<程序首部><分程序>.

<程序首部>::=PROGRAM 标识符;

<分程序>::=[<常量说明部分>][<变量说明部分>][<过程说明部分>]<复合语句>

<常量说明部分>::=CONST <常量定义> {, <常量定义>};

<常量定义>::=标识符=无符号整数

<变量说明部分>::=VAR <变量定义>{; <变量定义>};

<变量定义>::=标识符{, 标识符}: <类型>

<类型>::=INTEGER|LONG

<过程说明部分>::=<过程首部><分程序>;{<过程首部><分程序>;}

<过程首部>::=PROCEDURE 标识符; | PROCEDURE 标识符(标识符: <类型>);

<语句>::=<赋值语句>|<条件语句>|<当型循环语句>|<过程调用语句>

|<读语句>|<写语句>|<复合语句>| ε

<赋值语句>::=标识符:= <表达式>

<条件语句>::=IF <条件> THEN <语句>

<当型循环语句>::=WHILE <条件> DO <语句>

<过程调用语句>::=标识符|标识符(<表达式>)

<读语句>::=READ (标识符, {标识符})

<写语句>::=WRITE (<表达式>{, <表达式>})

<复合语句>::=BEGIN <语句>{;<语句>} END

<条件>::=<表达式><关系运算符><表达式>|ODD <表达式>

<表达式>::=[+|-]<项>{<加型运算符><项>}

<项>::=<因子>{<乘型运算符><因子>}

<因子>::=标识符|无符号整数|(<表达式>)

<加型运算符>::=+|-

<乘型运算符>::=*/

<关系运算符>::= =|<|>|<=|>=

其中：

< >：用左右尖括号括起的字符串表示非终结符号

::=：为定义为

{ }：表示该语法成分可以 0~n 次重复

[]：表示方括号内为可选项，即 0 或 1 次

三、程序设计语言单词的内部编码

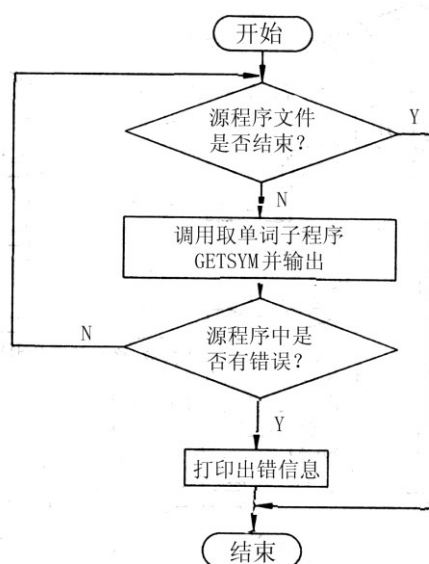
内码	单词	内码	单词	内码	单词	内码	单词
1	PROGRAM	2	CONST	3	VAR	4	INTEGER
5	LONG	6	PROCEDURE	7	IF	8	THEN
9	WHILE	10	DO	11	READ	12	WRITE
13	BEGIN	14	END	15	ODD	16	+
17	-	18	*	19	/	20	=
21	<>	22	<	23	<=	24	>
25	>=	26	.	27	,	28	;
29	:	30	:=	31	(32)
33	无符号整数	34	标识符	35	#		

四、词法分析程序的设计思想

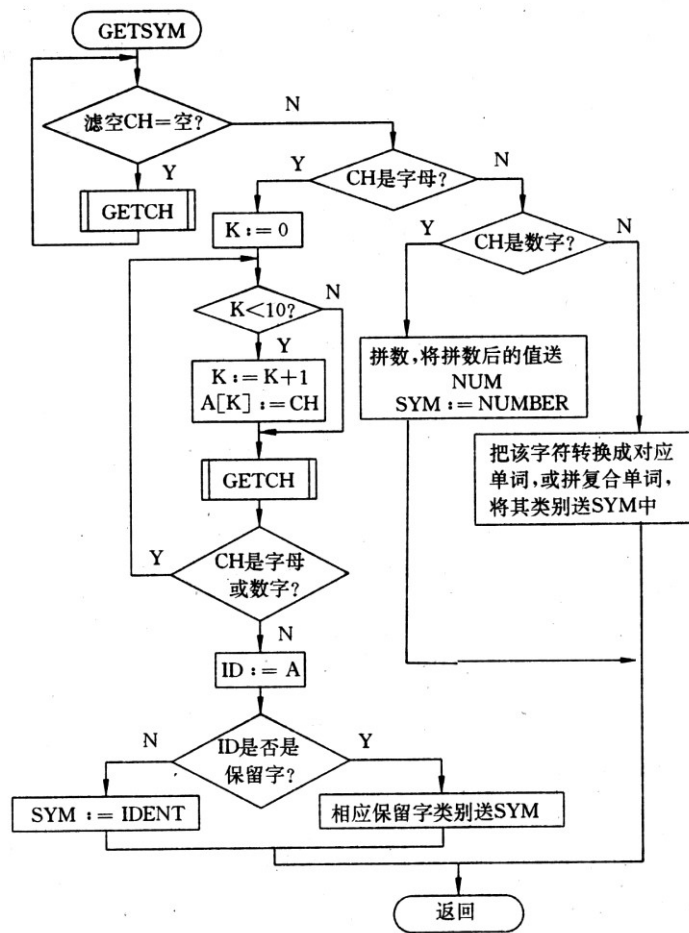
为了实现的编译程序实用和方便起见，我们规定源程序可采用自由书写格式，即一行内可以书定多个语句，一个语句也可以占领多行书写；标识符的前 20 个字符有效；整数用 2 个字节表示；长整数用 4 个字节表示。这样词法分析程序的主要工作为：

- (1) 从源程序文件中读入字符
- (2) 统计行数和列数用于错误单词的定位
- (3) 删除空格类字符，包括回车、制表符空格
- (4) 按拼写单词，并用（内码，属性）二元式表示

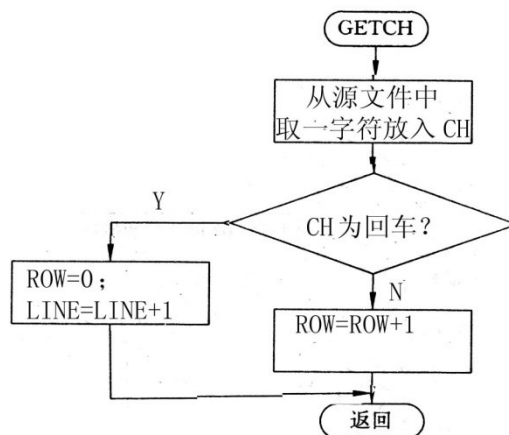
扫描程序流程图：



词法分析总程序流程图



取单词子程序流程图



五、用一个测试文件（PASCAL 程序）检测词法分析程序，测试文件如下：
PROGRAM new;

CONST a10=10,b=20,c=30;

VAR x,y,z:INTEGER

```
PROCEDURE fun1
```

```
BEGIN
```

```
x=9;
```

```
IF x<a THEN y=b;
```

```
END
```

请识别出该程序中的关键字、变量、数字、操作符等，并给出相应内码。