# 编译原理实验指导书

## 实验1词法分析

实验项目名称：实验1 词法分析

实验目的及要求：

熟悉词法分析的原理和实现过程。编制一个读单词函数，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。（遇到错误时可显示“Error”，然后跳过错误部分继续显示）

输出形式：

mainsym，main

lparen，（

rparen，）

lbparen， {

intsym， int

ident， a

comma， ,

ident， b

semicolon， ;

ident， a

eql， =

number，10

semicolon， ;

ident， b

eql， =

ident， a

plus， +

number， 20

semicolon，;

rbparen，}

输入示例：

|  |
| --- |
| main()  {  int a,b;  a = 10;  b = a + 20;  } |

对应的输出如右图所示。

要求：

* 1. 识别保留字：if、int、for、while、do、return、break、continue

对应的类型分别为：

|  |  |  |
| --- | --- | --- |
| ififsys | intintsys | for forsys |
| whilewhilesys | do dosys | return returnsys |
| break breaksys | continue continuesys |  |

* 1. 运算符包括：+ - \* / =

> < >= <= !=

对应的类型分别为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| + plus | - minus | \* times | / slash | = eql |
| > gtr | <lss | >=geq | <=leq | != neq |

* 1. 分隔符包括：, ; { } ( )

对应的类型分别为：

|  |  |  |
| --- | --- | --- |
| , comma | ; semicolon | { lbparen |
| } rbparen | ( lparen | ) rparen |

* 1. 常数为无符号整型数

所有常数的类型表示为：number

* 1. 所有标识符的类型表示为：ident
  2. 可根据需要可自定义单词的类型名称

实验原理：

（设计词法分析处理流程，并画出相应的流程图。以下仅供参考，请独立完成设计。）

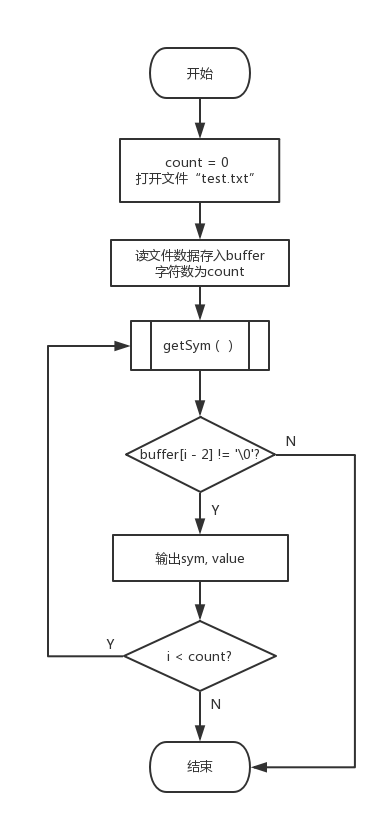


图1main函数流程

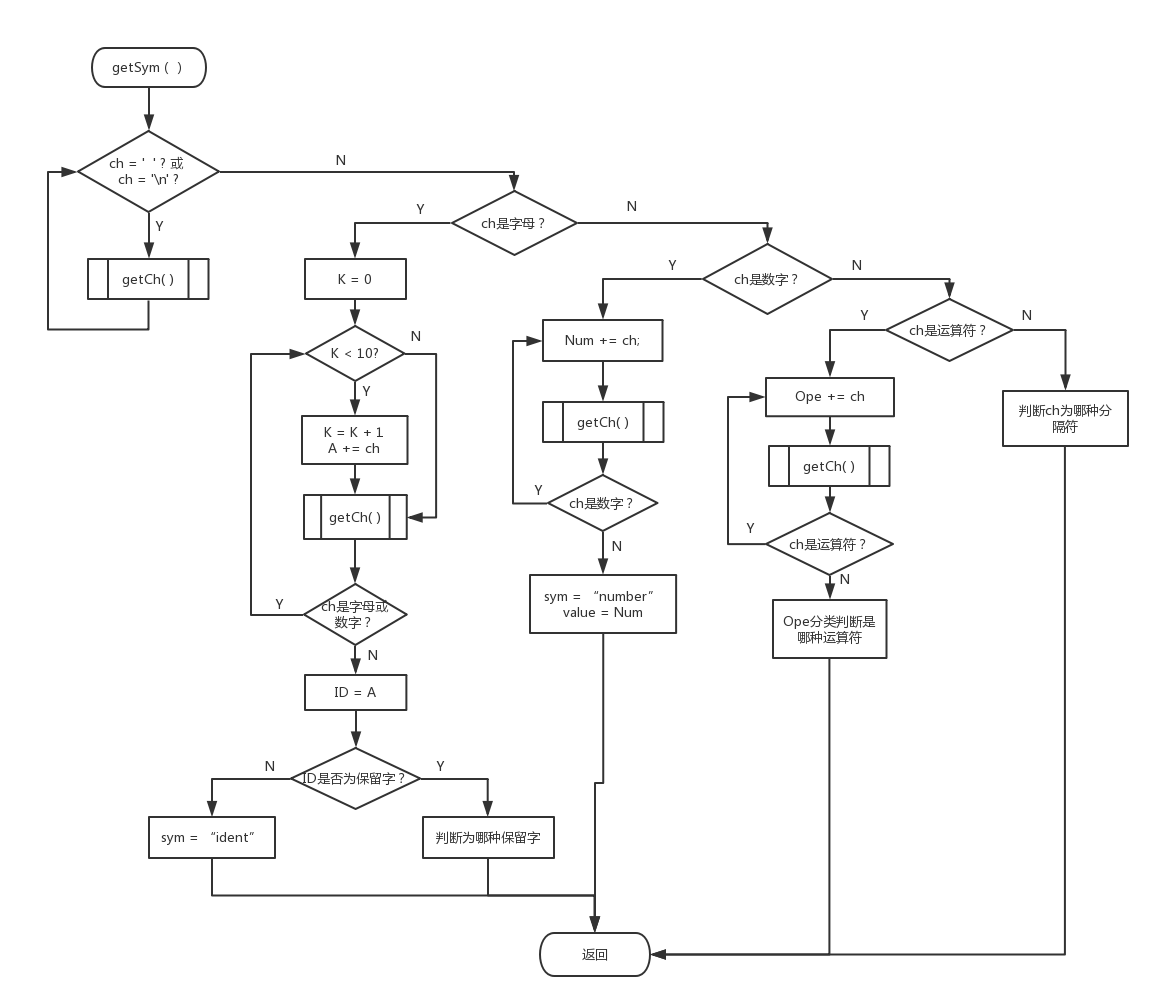


图2getSym( )函数流程

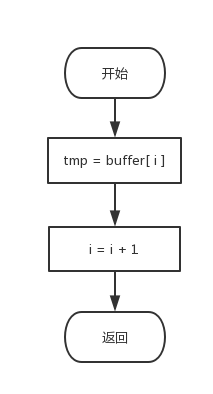


图3getCh( ) 函数流程

实验内容：

1.函数设计

（请根据流程图设计出所需函数，并在实验报告中写出各函数的名称和功能、参数类型和含义、返回值类型和含义、函数之间的调用关系）

例如：按照上述流程图构造如下函数

bool isLetter(char c)；判断参数c是否为字母，返回bool值；

bool isNumber(char c)；判断参数c是否为数字，返回bool值；

bool isOperator(char c)；判断参数c是否为运算符，返回bool值；

bool isResWord(string resWord)；判断参数resWord是否为保留字，返回bool值；

void getOperator(string ope)；获得参数ope对应的运算符；

void getResWord(string resWord)；获得参数resWord对应的保留字；

void getBorder(char c)；获得参数c对应的界符；

char getCh()； 从缓存区获取字符；

void getSym()；词法分析取得一个单词，并将当前单词及其类型存入全局变量。

调用关系（略）

2.功能实现

（主要函数功能的实现代码，代码要有较详细的注释）

实验结果与分析：

（运行结果截图，并分析算法复杂度、优势和不足）

## 实验2语法分析

实验项目名称：实验2.1 表驱动LL(1)语法分析

实验目的及要求：

熟悉表驱动LL(1)分析法的原理和实现过程。

文法为：

E🡪 E+T | E-T | T

T🡪 T\*F | T/F | F

F🡪（E）| i

根据LL(1)分析法，消除该文法的左递归和左公共因子，判别该文法是否为LL(1)文法；若是，设计出LL(1)分析表,并对下列表达式进行表驱动LL(1)语法分析，判断其语法是否正确。

|  |
| --- |
| 10;  1+y\*(x-6);  (r1+vt)/k+14-(r2+6/k);  ((100-13)/3+4;  sum+25-a2+(\*40/5). |

输出形式：

|  |
| --- |
| 10  语法正确  1+y\*(x-6)  语法正确  (r1+vt)/k+14-(r2+6/k)  语法正确  ((100-13)/3+4  语法错误，出错位置是……  sum+25-a2+(\*40/5)  语法错误，出错位置是…… |

注意：

1. 表达式中可能含变量和无符号整数；
2. 可以直接调用实验1的词法分析程序，取得单词；
3. 如果遇到错误的表达式，应输出错误提示信息（该信息越详细越好，最好有详细的出错位置和出错性质）；
4. 测试用的表达式事先放在文本文件中，一行存放一个表达式，同时以分号分割。同时将预期的输出结果写在另一个文本文件中。

实验原理：

文法为：

E🡪 E+T | E-T | T

T🡪 T\*F | T/F | F

F🡪（E）| i

1.判断文法是否为LL（1）文法。

由于文法含有左递归，所以必须先消除左递归，使文法变为：

E -> TE’

E’-> +TE’| -TE’| ε

T -> FT’

T’-> \*FT’|/FT’| ε

F -> (E) | i

可推出ε的非终结符表为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E | E’ | T | T’ | F |
| 否 | 是 | 否 | 是 | 否 |

1. 各非终结符的FIRST集合如下：

FIRST(E) = {(, i};

FIRST(E’) = {+, -, ε};

FIRST(T) = {(, i};

FIRST(T’) = {\*, /,ε}；

FIRST(F) = {(, i};

1. 各非终结符的FOLLOW集合为：

FOLLOW(E) = {), #};

FOLLOW(E’) = {), #};

FOLLOW(T) = {+, -, ), #};

FOLLOW(T’) = {+, -, ), #};

FOLLOW(F) = {\*, /, +, -, ), #};

1. 各产生式的SELECT集合为：

SELECT(E -> TE’) = {(, i};

SELECT(E’-> +TE’) = {+};

SELECT(E’-> -TE’) = {-};

SELECT(E’->ε) = {), #};

SELECT(T -> FT’) = {(, i};

SELECT(T’-> \*FT’) = {\*};

SELECT(T’-> /FT’) = {/};

SELECT(T’->ε) = {+, -, ), #};

SELECT(F -> (E)) = {(};

SELECT(F ->i) = {i};

由上可知有相同左部产生式的SELECT集合的交集为空，所以文法是LL（1）文法。

2.构造预测分析表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **i** | **+** | **-** | **\*** | **/** | **(** | **)** | **#** |
| **E** | TE’ |  |  |  |  | TE’ |  |  |
| **E’** |  | +TE’ | -TE’ |  |  |  | ε | ε |
| **T** | FT’ |  |  |  |  | FT’ |  |  |
| **T’** |  | ε | ε | \*FT’ | /FT’ |  | ε | ε |
| **F** | i |  |  |  |  | (E) |  |  |

3.程序流程

（请设计程序的处理流程，并在实验报告中画出流程图。）

实验内容：

1.函数设计

（请根据流程图设计出所需函数，并在实验报告中写出各函数的名称和功能、参数类型和含义、返回值类型和含义、函数之间的调用关系）

2.功能实现

（主要函数功能的实现代码，代码要有较详细的注释）

实验结果与分析：

（运行结果截图，并分析算法复杂度、优势和不足）

实验项目名称：实验2.2算符优先语法分析

实验目的及要求：

熟悉算符优先分析法的原理和实现过程。

文法为：

E🡪 E+T | E-T | T

T🡪 T\*F | T/F | F

F🡪（E）| i

根据算符优先分析法，判别该文法是否为算符优先文法；若是，设计出算符优先分析表,并对下列表达式进行算符优先语法分析，判断其语法是否正确。

|  |
| --- |
| 10;  1+y\*(x-6);  (r1+vt)/k+14-(r2+6/k);  ((100-13)/3+4;  sum+25-a2+(\*40/5). |

输出形式：

|  |
| --- |
| 10  语法正确  1+y\*(x-6)  语法正确  (r1+vt)/k+14-(r2+6/k)  语法正确  ((100-13)/3+4  语法错误，出错位置是……  sum+25-a2+(\*40/5)  语法错误，出错位置是…… |

注意：

1. 表达式中可能含变量和无符号整数；
2. 可以直接调用实验1的词法分析程序，取得单词；
3. 如果遇到错误的表达式，应输出错误提示信息（该信息越详细越好，最好有详细的出错位置和出错性质）；
4. 测试用的表达式事先放在文本文件中，一行存放一个表达式，同时以分号分割。同时将预期的输出结果写在另一个文本文件中。

实验原理：

参见“实验2.1”的步骤完成实验原理分析，设计出相应的程序流程图。

实验内容：

1.函数设计

（请根据流程图设计出所需函数，并在实验报告中写出各函数的名称和功能、参数类型和含义、返回值类型和含义、函数之间的调用关系）

2.功能实现

（主要函数功能的实现代码，代码要有较详细的注释）

实验结果与分析：

（运行结果截图，并分析算法复杂度、优势和不足）

实验项目名称：实验2.3 SLR(1)语法分析

实验目的及要求：

熟悉SLR(1)分析法的原理和实现过程。

文法为：

E🡪 E+T | E-T | T

T🡪 T\*F | T/F | F

F🡪（E）| i

根据SLR(1)分析法，判别该文法是否为SLR(1)文法；若是，设计出SLR(1)分析表,并对下列表达式进行SLR(1)语法分析，判断其语法是否正确。

|  |
| --- |
| 10;  1+y\*(x-6);  (r1+vt)/k+14-(r2+6/k);  ((100-13)/3+4;  sum+25-a2+(\*40/5). |

输出形式为：

|  |
| --- |
| 10  语法正确  1+y\*(x-6)  语法正确  (r1+vt)/k+14-(r2+6/k)  语法正确  ((100-13)/3+4  语法错误，出错位置是……  sum+25-a2+(\*40/5)  语法错误，出错位置是…… |

注意：

1. 表达式中可能含变量和无符号整数；
2. 可以直接调用实验1的词法分析程序，取得单词；
3. 如果遇到错误的表达式，应输出错误提示信息（该信息越详细越好，最好有详细的出错位置和出错性质）；
4. 测试用的表达式事先放在文本文件中，一行存放一个表达式，同时以分号分割。同时将预期的输出结果写在另一个文本文件中。

实验原理：

参见“实验2.1”的步骤完成实验原理分析，设计出相应的程序流程图。

实验内容：

1.函数设计

（请根据流程图设计出所需函数，并在实验报告中写出各函数的名称和功能、参数类型和含义、返回值类型和含义、函数之间的调用关系）

2.功能实现

（主要函数功能的实现代码，代码要有较详细的注释）

实验结果与分析：

（运行结果截图，并分析算法复杂度、优势和不足）

## 实验3 语义分析及中间代码生成

实验项目名称：实验3语义分析及中间代码生成

实验目的及要求：

熟悉语义分析和中间代码生成的原理和实现过程。

基于实验2的任一语法分析方法，对算术表达式进行语义分析，实现表达式的计算和相应逆波兰式的生成。

输入形式：

|  |
| --- |
| 10  1+2\*(15-6)  (1+200)/3+14-(11+6/7)  ((100-13)/3+4  10+25-32+(\*40/5) |

输出形式：

|  |
| --- |
| 10  语法正确  逆波兰式为 10  计算结果为 10  1+2\*(15-6)  语法正确  逆波兰式为 1 2 15 6 - \* +  计算结果为 19  (1+200)/3+14-(11+56/7)  语法正确  逆波兰式为 1 200 + 3 / 14 + 11 56 7 / + -  计算结果为 62  ((100-13)/3+4  语法错误，出错位置是……  逆波兰式为……  10+25-32+(\*40/5)  语法错误，出错位置是……  逆波兰式为…… |

注意：

1. 表达式只含无符号整数；
2. 如果遇到语法错误的表达式，应输出错误提示信息；
3. 测试用的表达式事先放在文本文件中，一行存放一个表达式，同时以分号分割。同时将预期的输出结果写在另一个文本文件中。

实验原理：

（设计出每个产生式对应的语义子程序应实现的功能和实现逻辑。）

实验内容：

（主要功能的实现代码，代码要有较详细的注释）

实验结果与分析：

（运行结果截图，并分析算法复杂度、优势和不足）