

# 编译原理实验指导

## 一、课程简介

1. 课程名称：编译原理（Principle of Compiler）
2. 课程总学时： 64 学时[理论： 48 学时；实验： 16 学时]

## 二、实验目的

编译原理是计算机类专业特别是计算机软件专业的一门重要专业课。设置该课程的目的在于系统地向学生讲述编译系统的结构、工作流程及编译程序各组成部分的设计原理和实现技术，使学生通过学习既掌握编译理论和方法方面的基本知识，也具有设计、实现、分析和维护编译程序等方面的初步能力。编译原理是一门理论性和实践性都比较强的课程。进行上机实验的目的是使学生通过完成上机实验题目加深对课堂教学内容的理解。同时培养学生实际动手能力。

## 三、实验环境

微机安装好 C 语言，或 C++，或 Visual C++开发环境。（或安装自己需要的开发环境）

## 四、实验任务

用 C/C++/Visual C++语言编写 PL0 语言的词法分析程序、语法分析程序、语义分析及中间代码生成程序。（或用自己选用的语言写也可以）

## 五、PL0 语言简介

PL0 语言功能简单、结构清晰、可读性强，而又具备了一般高级程序设计语言的必须部分，因而 PL0 语言的编译程序能充分体现一个高级语言编译程序实现的基本方法和技术。

# 1. PL/0 语言文法的 EBNF

## 1.1 符号说明:

‘<>’ 用左右尖括号括起来的中文字表示语法构造成分，或称语法单位，为非终结符。

‘::=’ 该符号的左部由右部定义，可读作“定义为”。

‘|’ 表示“或”，为左部可由多个右部定义。

‘{}’ 表示花括号内的语法成分可以重复。在不加上下界时可重复 0 到任意次数，有上下界时为可重复次数的限制。

‘[]’ 表示方括号内的成分为任选项。

‘()’ 表示圆括号内的成分优先。上述符号称“元符号”，定义文法用到上述符号作为文法符号时需要引号 ‘ ’ 括起。

## 1.2 PL/0 语言文法的 EBNF:

<程序>::=<分程序>.

<分程序> ::= [<常量说明>][<变量说明>][<过程说明>]<语句>

<常量说明> ::= CONST<常量定义>{, <常量定义>;}

<常量定义> ::= <标识符>=<无符号整数>

<无符号整数> ::= <数字>{<数字>}

<变量说明> ::= VAR <标识符>{, <标识符>;}

<标识符> ::= <字母>{<字母>|<数字>}

<过程说明> ::= <过程首部><分程序>{;<过程说明> };

<过程首部> ::= PROCEDURE <标识符>;

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

          |<复合语句>|<读语句>|<写语句>|<空>

<赋值语句> ::= <标识符>:=<表达式>

<复合语句> ::= BEGIN <语句> {;<语句> }END

<条件语句> ::= <表达式> <关系运算符> <表达式> | ODD<表达式>

<表达式> ::= [+|-]<项>{<加法运算符> <项>}

<项> ::= <因子>{<乘法运算符> <因子>}

<因子> ::= <标识符>|<无符号整数>|‘(<表达式>)’

<加法运算符> ::= +|-

<乘法运算符> ::= \*/

<关系运算符> ::= =|<|<=|>|>=

<条件语句> ::= IF <条件> THEN <语句>

<过程调用语句> ::= CALL 标识符

<当循环语句> ::= WHILE <条件> DO <语句>

<读语句> ::= READ('(<标识符>{,<标识符>}')

<写语句> ::= WRITE('(<表达式>{,<表达式>}')

<字母> ::= a|b|...|X|Y|Z

<数字> ::= 0|1|...|8|9

若不习惯看文法的巴科斯范式 EBNF, 可先将文法改写成常规的产生式形式 P75

## 2. PL/0 语言的词汇表

序号	类别	单词	编码
1	基本字	begin, call, const, do, end if, odd, procedure, read then, var, while, write	beginsym, callsym, constsym dosym, endsym, ifsym, oddsym , proceduresym, readsym, thensym , varsym, whilesym, writesym
2	标识符		ident
3	常数		number
4	运算符	+, -, *, /, odd =, <>, <, <=, >, >=, :=	plus, minus, times, slash, oddsym, eql, neq, lss, leq, gtr, geq, becomes
5	界符	( ) , ; .	lparen, rparen, comma, semicolon, period

## 六、实验项目

### 实验一. 词法分析

#### 1. 实验目的

- 根据 PL/0 语言的文法规范，编写 PL/0 语言的词法分析程序；或者调研词法分析程序的自动生成工具 LEX 或 FLEX，设计并实现一个能够输出单词序列的词法分析器。
- 通过设计调试词法分析程序，实现从源程序中分出各种单词的方法；加深对课堂教学的理解；提高词法分析方法的实践能力。
- 掌握从源程序文件中读取有效字符的方法和产生源程序的内部表示文件的法。
- 掌握词法分析的实现方法。
- 上机调试编出的词法分析程序。

#### 2. 实验准备

微机安装好 C 语言，或 C++，或 Visual C++，或自己需要用的语言。

#### 3. 实验时间

4 学时

#### 4. 实验内容

已给 PL/0 语言文法，输出单词符号（关键字、专用符号以及其它标记）。

#### 5. 实验要求

- 把词法分析器设计成一个独立一遍的过程。
- 词法分析器的输出形式采用二元式序列，即：  
(单词种类, 单词的值)

#### 6. 输入输出

输入：PL/0 源程序。例：

**const a=10;**

```
var b,c;  
begin  
  read(b);  
  c:=a+b;  
  write(c)  
end.
```

输出:

```
(constsym,const)  
(ident,a)  
(eql,=)  
(number,10)  
(semicolon,;)  
(varsym,var)  
(ident,b)  
(comma,,)  
(ident,c)  
(semicolon,;)  
(beginsym,begin)  
(readsym,read)  
(lparen,()  
(ident,b)  
(rparen,))  
(semicolon,;)  
(ident,c)  
(becomes,:=)  
(ident,a)  
(plus,+)  
(ident,b)  
(semicolon,;)  
(writesym,write)  
(lparen,()  
(ident,c)
```

(rparen,))  
(endsym,end)  
(period,.)

## 7. 实验报告

应包括以下内容：

- 题目
- 设计思想（包括 r、NFA、DFA、最小化的 DFA）
- 算法流程（程序的算法流程图）
- 源程序（要求有注释/无注释的源程序的得分为 0）
- 调试数据（输入/输出）
- 实验调试情况及体会

实验报告用 word 文档写好后上传到 <http://itc.hzau.edu.cn/> 平台