

11110011101011011000111001
0101111000011101101010010

○蒋立源 康慕宁 主编
○冯博琴 主审

编译原理

第2版

1111001
1010110
1100011
1001010
1111000
0111011
0101001
0010101
1100011
0011100
0111101

```
while(1);
}
/* Insert a node */
insert(head)
MEMBER *head
{
    MEMBER *newp; /* appl __ mem( );
    newp = appl __ mem( );
    if (!newp) return(0);
    newp->next = head;
    head = newp;
    suc = head->next;
    while(suc != NULL) {
        if(suc->num >= newp->num) {
            pre = suc;
            suc = suc->next;
        }
        pre->next = newp;
        newp->next = suc;
    }
}
MEMBER * appl __ mem( )
{
    /* memory allocation */
    MEMBER *newp;
    newp = (MEMBER *) malloc (SIZE);
    if(newp == 0) {
        printf("memory allocation error! \n");
        exit(0);
    }
    return(newp);
}
/* Input data */
main()
{
    MEMBER *newp;
    char s[20];
    printf("\ninput name:");
    gets(newp->name);
    printf("\nnumberr:");
}
```

西北工业大学

社

普通高等教育“九五”国家级重点教材

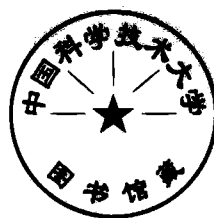
编 译 原 理

(第2版)

主编 蒋立源 康慕宁

主审 冯博琴

蒋立源 康慕宁 吴 健 编
高集荣 张延园 薛 贺



西北工业大学出版社

1999年9月 西安

(陕)新登字 009 号

JS13-8 66

【内容简介】 本书第2版为“九五”国家级重点教材,旨在系统地介绍编译系统的结构、工作流程以及编译程序的设计原理和基本实现技术。全书共分为十章。第1章对编译过程、编译程序的逻辑结构及编译程序各部分的主要功能进行了概括的说明。第2章介绍前后文无关文法和语言的基本知识。第3章以正规文法、正规式和有限自动机为工具,讨论了词法分析程序的设计原理和自动生成方法。第4章讲述了语法分析程序的设计技术。第5章以属性文法的理论为基础,以语法制导翻译为模式,介绍了将程序设计语言常见的语法成分翻译为中间代码的方法,并对几种十分流行的、行之有效的语法分析程序自动生成工具(LLama, YACC, OCCS等)的使用方法进行了扼要的讨论。第6章至第10章分别讨论了符号表的构造、目标程序运行时的存储组织与分配、代码优化、目标代码生成及源程序的查错与改错等问题。本书在基本概念和基本原理的阐述上力求准确清楚,在内容的组织上力求循序渐进,深入浅出,以便于自学;各章之后均附有一定数量的习题和上机实习题目。

本书可作为大学本科计算机类各专业的教材,也可作为从事计算机软件开发的科技工作者的参考书。

图书在版编目(CIP)数据

编译原理/蒋立源,康慕宁主编;吴健等编. -2版. -西安:西北工业大学出版社,
1999.9 重印
ISBN 7-5612-1147-3/TP·161(课)

I. 编… II. ①蒋… ②康… ③吴… III. 编译程序 IV. TP314

中国版本图书馆 CIP 数据核字 (1999) 第 27979 号

西北工业大学出版社出版发行
(邮编:710072 西安市友谊西路127号 电话:8491147)
全国各地新华书店经销
陕西省富平印刷有限责任公司印装

*

开本:787毫米×1092毫米 1/16 印张:23.375 字数:565千字
1993年8月第1版 1999年9月第2版第6次印刷
印数:27 001~31 500册 定价:27.00元

购买本社出版的图书,如有缺页、错页的,本社发行部负责调换。

第2版前言



本书自1993年出版以来,被许多兄弟院校用作相应课程的教材,迄今已印刷5次。为适应计算机教育迅速发展和提高的需要,我们借“九五”国家级重点教材立项之机,对本书进行了较全面的修改和补充。

这次修订,除保持第1版的风格和在文字上作进一步润饰之外,主要变动的方面如下:

1. 在对语法制导翻译及中间代码生成的论述上,采用了以属性文法的理论为基础,将语法制导翻译技术和语法分析器自动生成有机结合起来的方式,使编译程序的设计与实现有更为规范的途径。

2. 对目前十分流行,且使用效果良好的编译器自动生成工具,如LEX, LLama, YACC及OCCS等,在其功能和使用方法方面,内容有更多的扩充。

3. 在代码优化部分,增加了“语法制导翻译阶段的优化”和“线性窥孔优化”两节,同时对原书中有关代码优化的其它内容作了适当的精简。

4. 考虑到讲解具有回溯的自顶向下分析方法比较费时费力,且这种方法并无多少实用价值,故将相应的分析算法及实例予以删除。

5. 由于COMMON语句和EQUIVALENCE语句为FORTRAN语言所独有,且其处理技术并不具有普遍意义,故为节省篇幅,也将其予以删除。

6. 鉴于近几年来C语言已在国内广泛使用,并已成为许多领域的主流语言,故在第2版中将所有的分析算法、驱动程序以及语义动作代码等等,均改用C语言进行描述。

7. 在章节安排上作了局部的调整,对全书内容的表述格式进行了进一步的修改加工,使系统结构更趋合理,论述更为准确畅达。

8. 这次修订,内容的取舍仍以课内72学时为依据。若课程学时较少,在使用本书时,可酌情删减一些次要的教学内容,如第4章的优先分析方法、第8章中的某些代码优化方法,等等。

此外,其它小的修改还有多处,恕不一一列举。总之,我们希望,经

过这次修订之后,能使本书成为一种有较强的科学性和系统性,对课程的一些基本概念和基本原理,以及一些有典型意义、应用较为广泛的编译技术和方法进行较深入细致讨论且实用性较强的教材。然而,由于我们的水平有限,上述目标是否已经达到,还有什么缺陷与不妥之处需要完善和订正,敬请读者多加批评指教。

这次修订,由西北工业大学蒋立源和康慕宁担任主编,参加编写工作的还有吴健、高集荣、张延园和薛贺等同志。在编写和修订本书的过程中,曾先后参考了多种相关的书籍和资料。我们对这些书籍和资料的编著者表示衷心的感谢。

最后,我们特别对西安交通大学冯博琴教授表示深切的谢意,他对本书第1版和第2版的全部书稿进行了细致的审查,提出了许多宝贵中肯的意见,对本书质量的提高帮助甚大。陕西省老年书画学会常务副会长李峰山教授欣然为本书题写了书名,我们谨向李老先生表示崇高的敬意。

编 者

1999年2月于西安

第 1 版前言

“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。设置本课程的目的,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,使学生通过学习本课程,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。

根据上述要求,在航空高等学校第二教学指导委员会的组织下,我们参照中国计算机教育研究会向全国推荐的《编译原理教学大纲》,并结合编者多年来讲授本课程的教学实践经验,编写了本书。

全书共分为十章,第 1 章对编译过程、编译程序的逻辑结构及编译程序各组成部分的主要功能进行了概括的说明。第 2 章介绍前后文无关文法和语言的基本知识,它为学习后续各章奠定了理论基础。第 3 章以正规文法、正规式和有限自动机为工具,讨论了词法分析程序的设计原理。第 4 章讲述了语法分析程序的设计技术,其中既介绍了传统的算符优先分析方法,也介绍了目前十分流行的递归下降分析及 LR 分析法,同时还对一种行之有效的语法分析程序自动生成工具 YACC 的使用方法进行了简要的说明。第 5 章以语法制导翻译为模式,介绍了将程序设计语言常见的语法成分翻译为中间代码的方法。第 6 章至第 10 章分别讨论了符号表的构造、目标程序运行时的存储组织与分配、代码优化、目标代码生成及源程序的查错与改错等问题。各章之后均附有一定数量的习题供读者选做。

编译原理是一门理论性和实践性都比较强的课程。在本书的编写过程中,我们力图将其中的基本概念、基本原理和实现方法的思路阐述清楚,因为它们不仅是构造编译程序的依据,而且对开发其它系统软件和应用软件也是很有用的。同时,为了培养学生的实际工作能力,我们在有关的章节之后,还列出了一些上机实习题目,学生通过完成这些作业可进一步加深对课堂教学内容的理解。

本书系航空教材编审委员会 1991~1992 年教材选题计划所列的部委规划教材,可作为计算机类专业编译原理课的教科书(课堂讲授约需 72~80 学时,此外还应有 25~30 学时的上机时间),也可供有关工程技术人员参考。

本书由西北工业大学蒋立源主编,参加编写工作的还有张延园、石志强、叶军和胡滨等同志。

西安交通大学冯博琴教授对本书进行了仔细的审阅,提出了许多宝贵的意见。在本书编写过程中,我们还得到了西北工业大学出版社的许多同志以及计算机系的张遵濂、徐秋元、韩兆轩、白中英、胡正国、赵政文等教授的支持、关心和帮助,在此一并表示衷心的感谢。

由于我们学力有限,书中定有不妥之处,恳请读者批评指正。

编 者

1992年3月于西安

目 录

第 1 章 绪论	1
1.1 编译过程概述	3
1.2 编译程序的逻辑结构	4
1.2.1 词法分析程序	5
1.2.2 语法分析程序	6
1.2.3 语义分析程序	6
1.2.4 中间代码生成	7
1.2.5 代码优化程序	7
1.2.6 目标代码生成程序	8
1.2.7 错误检查和处理程序	9
1.2.8 信息表管理程序	10
1.3 编译程序的组织	11
习题	12
第 2 章 前后文无关文法和语言	13
2.1 文法及语言的表示	13
2.2 文法和语言的定义	14
2.2.1 基本概念和术语	15
2.2.2 文法和语言的形式定义	16
2.3 句型的分析	22
2.3.1 规范推导和规范归约	22
2.3.2 语法树和二义性	24
2.3.3 短语和句柄	28
2.4 文法的化简和改造	30
2.4.1 无用符号和无用产生式的删除	30
2.4.2 ϵ -产生式的消除	32
2.4.3 单产生式的消除	34
2.5 文法和语言的 Chomsky 分类	35

习题	37
第3章 词法分析及词法分析程序	41
3.1 设计扫描器时应考虑的几个问题	41
3.1.1 词法分析阶段的必要性	41
3.1.2 单词符号的内部表示	42
3.1.3 识别标识符的若干约定和策略	43
3.1.4 源程序的输入及预处理	45
3.2 正规文法和状态转换图	48
3.2.1 由正规文法构造状态转换图	48
3.2.2 状态转换图的一种实现——状态矩阵法	52
3.3 有限自动机	58
3.3.1 确定的有限自动机	58
3.3.2 非确定的有限自动机	59
3.3.3 NFA 与 DFA 的等价性	60
3.3.4 具有 ϵ 动作的 FA	63
3.3.5 具有 ϵ 动作的 NFA 的确定化——子集法	65
3.3.6 DFA 状态数的最小化	68
3.4 正规表达式与正规集	70
3.4.1 正规表达式与正规集的定义	71
3.4.2 由正规文法构造相应的正规式	72
3.4.3 由正规式构造 FA——Thompson 法	75
3.5 词法分析程序的实现	77
3.5.1 词法分析程序的编写	78
3.5.2 词法分析程序的自动生成	82
习题	99
上机实习题	104
第4章 语法分析和语法分析程序	106
4.1 自顶向下的语法分析	107
4.1.1 消除文法的左递归	108
4.1.2 回溯的消除及 LL(1)文法	111
4.1.3 递归下降分析法	113
4.1.4 预测分析法	119
4.1.5 某些非 LL(1)文法的改造	124
4.2 自底向上的语法分析	126
4.2.1 简单优先分析法	127
4.2.2 算符优先分析法	136
4.2.3 优先函数	144

4.2.4 LR 分析法	150
习题	177
上机实习题	185
第 5 章 语法制导翻译及中间代码生成	187
5.1 引言	187
5.2 属性文法与属性翻译文法	189
5.2.1 语义属性与属性文法	190
5.2.2 属性翻译文法	193
5.3 常见中间语言简介	197
5.3.1 逆波兰表示	197
5.3.2 四元式和三元式	200
5.3.3 其它表示法	202
5.4 简单算术表达式和赋值语句的翻译	203
5.5 布尔表达式的翻译	205
5.6 程序流程控制语句的翻译	210
5.6.1 常见控制结构的翻译	210
5.6.2 FOR 循环语句的翻译	216
5.6.3 语句标号及 GOTO 语句的翻译	219
5.6.4 情况语句的翻译	222
5.7 含数组元素的算术表达式及赋值语句的翻译	224
5.7.1 下标变量地址的计算	225
5.7.2 含有下标变量的赋值语句的翻译	227
5.8 过程说明和过程调用的翻译	229
5.8.1 过程说明的翻译	230
5.8.2 实参和形参间的信息传递	231
5.8.3 过程语句的翻译	233
5.8.4 关于形实结合的进一步讨论	234
5.9 说明语句的翻译	236
5.9.1 类型说明(变量及数组定义)语句的翻译	236
5.9.2 数据类型定义语句的翻译	239
5.10 语法分析程序的自动生成工具	241
5.10.1 LALR(1)分析器的自动生成工具——YACC 和 OCCS	242
5.10.2 LL(1)语法分析程序自动生成工具 LLama 简介	252
5.10.3 LLGen 简介	254
习题	255
上机实习题	258

第 6 章 符号表	259
6.1 符号表的组织	259
6.2 分程序结构语言符号表的建立	263
6.3 非分程序结构语言符号表的建立	268
习题	270
第 7 章 运行时的存储组织与分配	272
7.1 存储组织	273
7.1.1 运行时内存的划分	273
7.1.2 活动记录	273
7.2 运行时的分配策略	274
7.2.1 静态分配	275
7.2.2 栈式分配	278
7.2.3 堆式分配	281
习题	283
第 8 章 代码优化	286
8.1 语法制导翻译阶段的优化	286
8.2 线性窥孔优化	287
8.2.1 强度削弱	288
8.2.2 常数合并和常数传播	289
8.2.3 无用变量与无用代码的删除	290
8.2.4 窥孔优化实例	293
8.3 基于结构信息的优化	296
8.3.1 基本块及其优化	296
8.3.2 数据流分析方法	302
8.3.3 循环优化	311
习题	327
上机实习题	332
第 9 章 目标代码生成	333
9.1 目标代码的形式	333
9.2 一种假想的计算机模型	335
9.3 一种代码生成程序的雏型	338
9.3.1 待用信息	338
9.3.2 寄存器描述符与地址描述符	339
9.3.3 生成目标代码的算法	340
9.4 DAG 的代码生成	343

9.5 全局寄存器分配	345
习题	349
上机实习题	350
第 10 章 查错与改错	351
10.1 语法错误的校正	352
10.1.1 单词错误的校正	352
10.1.2 自顶向下分析中的错误校正	353
10.1.3 自底向上分析中的错误校正	356
10.2 语义错误的校正	359
10.2.1 遏止株连信息	359
10.2.2 遏止重复信息	359
习题	360
参考文献	361

第 1 章 绪 论

程序设计语言是用来编写程序的工具,可分为两大类。第一类称为低级语言,包括机器语言、汇编语言以及其它面向机器的程序设计语言。这类语言对计算机的依赖性强、直观性差、编写程序的工作量大,只有对相应计算机的结构比较熟悉,且经过一定训练的程序人员才能较好地使用此类语言。第二类称为高级语言,有几百种之多,但除了一些专用语言之外,得到广泛运用的只有其中少数几种,如 BASIC, FORTRAN, PASCAL, COBOL, C 等。高级语言不论在算法描述的能力上,还是在编写和调试程序的效率上,都远比低级语言优越。

然而,计算机硬件只懂自己的指令系统,即只能直接执行用相应机器语言编写的代码程序,而不能直接执行用高级语言或汇编语言编写的程序。因此,要在计算机上实现除机器语言之外的任一程序设计语言,就首先应使此种语言为计算机所“理解”。解决这一问题的方法有两种:一种是对程序进行翻译;另一种是对程序进行解释。

所谓翻译,是指在计算机中放置一个能由计算机直接执行的翻译程序,它以某一种程序设计语言(源语言)所编写的程序(源程序)作为翻译或加工的对象,当计算机执行翻译程序时,就将它翻译为与之等价的另一种语言(目标语言)的程序(目标程序)。“源”和“目标”这两个术语总是相对于一类特定的翻译程序和翻译过程而言的。如果一个翻译程序的源语言是某种高级语言,其目标语言是相应于某一计算机的汇编语言或机器语言,则称这种翻译程序为编译程序。汇编程序也是一种翻译程序,它的源语言和目标语言分别是相应的汇编语言和机器语言。

由此可见,欲按编译方式在计算机上执行用高级语言编写的程序,一般需经过两个阶段:第一阶段称为编译阶段,其任务是由编译程序将源程序编译为目标程序,若目标程序不是机器代码,而是汇编语言程序,则尚须汇编程序再行汇编为机器代码程序;第二阶段称为运行阶段,其任务是在目标计算机上执行编译阶段所得到的目标程序。在执行目标程序时,一般还应有一些子程序配合进行工作,例如:常见的数据格式转换子程序、标准函数计算子程序、浮点解释子程序、数组动态存储分配子程序、下标变量地址计算机子程序等都属此类。这些子程序组

成一个子程序库,称为运行系统。运行子程序库中的各个子程序,大都按模块化的结构来编制。显然,库中的子程序愈丰富,各子程序的功能愈强,编译程序本身就愈简明紧凑。

编译程序与运行系统合称为编译系统。

源程序的编译(或汇编)和目标程序的执行不一定在同一种计算机上完成。当源程序由另一种计算机进行编译(或汇编)时,我们将此种编译(或汇编)称为交叉编译(或汇编)。

图 1-1 粗略地显示了按编译方式执行一个高级语言程序的主要步骤。

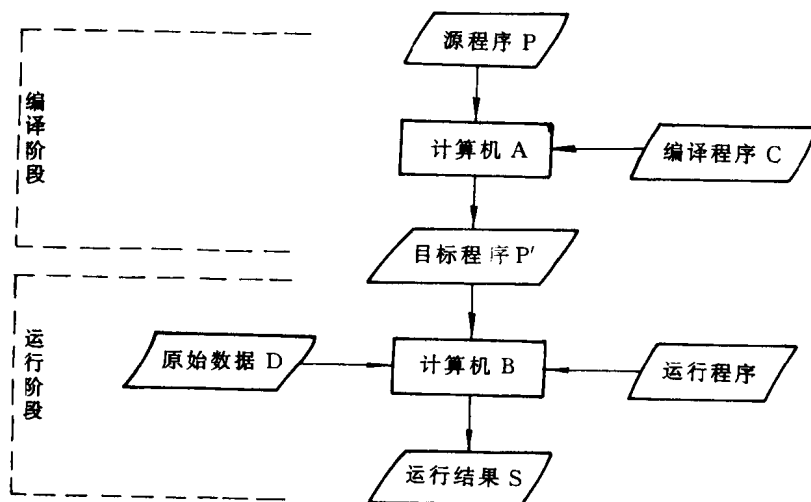


图 1-1 计算机执行高级语言程序的步骤

用高级语言编写的程序也可以通过解释程序来执行。解释程序也以源程序作为它的输入,它与编译的主要区别是在解释程序的执行过程中不产生目标程序,而是解释执行源程序本身。这种边翻译边执行的方式工作效率很低,但由于解释程序的结构比编译程序简单,且占用内存较少,在执行过程中也易于在源程序一级对程序进行修改,因此一些规模较小的语言,如 BASIC,也常采用此种方式。然而就目前的情况来看,纯粹的解释程序并不多见,通常的做法是把编译和解释做某种程度的结合。例如,有的先将源程序翻译为某种易于进行解释执行的内部中间语言形式,然后再对此中间语言程序进行解释执行;有的甚至在进行上述翻译时,还对一部分出现比较频繁的结构(如算术表达式等)产生目标代码。在采取上述这些措施后,解释程序执行效率不高的缺陷将有可能得到部分弥补。

编译程序已成为现今任何计算机系统的最重要系统程序之一。本课程的目的,在于向读者介绍设计和构造编译程序的基本原理和基本方法,其中许多方法也同样适用于构造解释程序或汇编程序。事实上,任何一个熟悉编译程序构造的人,是不难旁通解释程序或汇编程序的工作原理和实现方法的。因此,限于篇幅,对于有关构造解释程序和汇编程序方面的问题,本书将不再涉及。

1.1 编译过程概述

编译程序的主要功能是把用高级语言编写的源程序翻译为等价的目标程序。既然编译过程是一种语言的翻译过程,因此我们可将编译程序的工作过程与通常外语资料的翻译过程进行类比,这将有助于更直观地了解一个编译程序一般应由哪些部分组成,以及各个组成部分应如何进行工作等等。

抽象地看,任何一本外文资料都是由字母、标点符号(包括空格和其它符号)按相应语法规则所组成的字符串。因此,任何欲进行外文翻译的人,都应具备如下能力:①能认识外语的字母及标点符号;②能识别出文中的各个单词;③会查字典;④懂得此种外语的语法;⑤具有目标语言的修辞能力。至于如何进行翻译,概括地讲无非是做两方面的工作:一是进行分析,二是进行综合。所谓分析,就是从第一行的第一个字母开始,依次阅读原文中的各个符号,逐个识别出原文中的各个单词,然后根据语法规则进行语法分析,即分析原文中如何由单词组成短语和句子,以及句子的种类特点等。此外,在识别单词和进行语法分析的过程中,还要不时地查阅字典,做语法正确性的检查,进行相应的语义分析,并做一些必要的信息记录工作等等。所谓综合,就是根据上述分析所得到的信息,拟定译稿,进行修辞加工,最后写出译文。

类似地,编译程序在其工作过程中,也需做两方面的工作,即先分析源程序,然后再综合为目标程序。为了便于理解编译程序在此两方面应包括的工作环节,现将源程序的编译和外文资料的翻译两过程的主要工作进行对比如表1-1所示。

尽管编译过程与外文书刊翻译的工作过程比较类似,但由于编译程序所翻译的毕竟不是自然语言,因此,就必然有其自身特有的一些工作。

比如中间代码的产生,编译过程中信息表的构造与查询,以及运行时存储空间的分

配,对语法和语义错误进行必要的处理等杂务工作。诸如此类的工作还有很多,兹不一一列举。总之,编译程序是计算机的一个十分复杂的系统程序。为便于构造或分析一个编译程序,宜将整个编译程序分解为若干个组成部分,每一部分都用一段相对独立的程序去完成整个编译过程的一部分功能。就一个典型的编译程序而论,一般都含有下面八个部分:

- (1) 词法分析程序(也称为扫描器);
- (2) 语法分析程序(有时简称为分析器);
- (3) 语义分析程序;
- (4) 中间代码生成程序;
- (5) 代码优化程序;
- (6) 目标代码生成程序;
- (7) 错误检查和处理程序;
- (8) 各种信息表格的管理程序。

表 1-1 翻译和编译工作的比较

	翻译外文书刊	编译源程序
分 析	阅读原文	输入并扫视源程序
	识别单词	词法分析
	分析句子	语法分析
综 合	修辞加工	修饰优化
	写出译文	目标代码生成

在下一节中,我们将简要说明上述各个部分的功能,并指出如何将这八个部分组成一个完整的编译程序。

1.2 编译程序的逻辑结构

在 1.1 节中,我们已概括地介绍了编译程序的工作过程,并指出了一个典型的编译程序一般所包含的八个组成部分。图 1-2 表示了这八个部分间的控制流程和信息流程(分别用实线和虚线表示)。

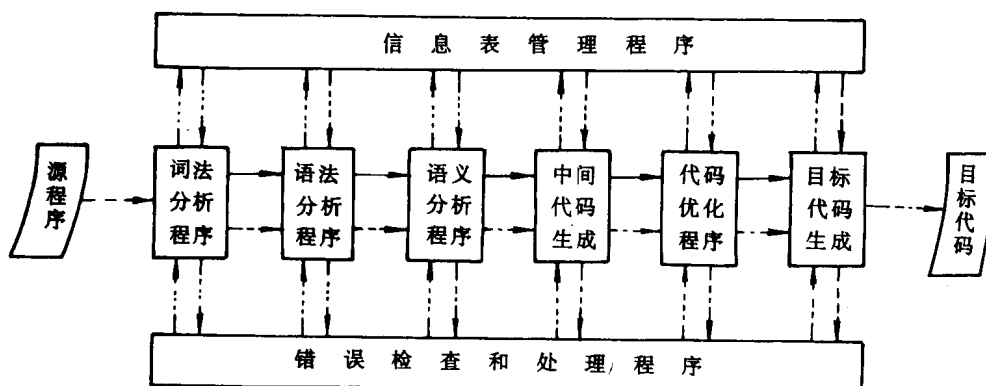


图 1-2 编译程序的逻辑结构

下面,我们用一个微型 PASCAL 语言 (PASCAL/M) 所编写的程序为例,分别介绍这八个部分的功能,并分别给出每一部分对此程序进行加工处理可能得到的结果。我们假定此语言只有如下四种语句:

- ① PROGRAM 语句;
- ② 说明语句;
- ③ BEGIN - END 语句;
- ④ 赋值语句。

每个 PASCAL/M 程序都以一个 PROGRAM 语句开头,此语句中的标识符用来给程序命名;PROGRAM 语句之后是说明语句,用来指明程序中所出现的各个变量的数据类型(假定 PASCAL/M 中只有整型变量);在一系列说明语句之后,再跟以一个 BEGIN - END 语句,在保留字 BEGIN 和 END 之间,应有一个或多个赋值语句,图 1-3 所示的 PASCAL/M 程序,对于我们后面的讨论来说,是一个恰当的例子。

```

1  PROGRAM source;
2      {this little source program is used to
3      illustrate compiling procedure. }
4  VAR x,y,z: integer;
5      a: integer;
6  BEGIN

```

```

7  {this program has only four statements. }
8  x := 23 + 5;
9  z := x DIV -3;
10 y := z + 18 * 3;
11 a := x + (y - 2) DIV 4;
12 END.

```

图 1-3 一个 PASCAL 源程序 source

1.2.1 词法分析程序

作为编译程序的输入,源程序仅仅是一个长长的字符串,扫描器将把这种形式的源程序转换为便于编译程序其余部分进行处理的内部格式。扫描器的工作任务如下:

- ① 识别出源程序中的各个基本语法单位(也称为单词或语法符号);
- ② 删除无用的空白字符、回车字符以及其它与输入介质相关的非实质性字符;
- ③ 删除注释;
- ④ 进行词法检查,报告所发现的错误。

现考虑图 1-3 所示的程序 source。扫描器依次查看缓冲区中源程序的各个字符,根据当前正查看之字符的种类,并参考扫描过程中前面所得到的信息,就能准确地判断当前正扫描的字符在源程序中所处的地位。概括而言,不外下述五种情况之一:

- ① 它是正处理的注释中的一个字符;
- ② 它是一个无用的空白字符;
- ③ 它是下一个单词的首字符;
- ④ 它是正识别的单词中的一个字符;
- ⑤ 它是一个不合词法规则的字符,或是一个不属于本语言字符集中的一个字符。

显然,如果扫描器根据上述不同的情况作不同的处置,并产生预定形式的输出,那么,它就能圆满地完成上面所提到的四项任务。

图 1-4 给出了扫描器对程序 source 进行处理后的一种可采用的输出形式。其中,程序里的各个单词已被一一识别出来,并用一个特定的标志符号“#”将相邻的两个单词加以分隔,程序中的非实质性符号已被全部删除。

```

# PROGRAM # source # ; # VAR # x # , # y # , # z # : # integer # ; # a # ;
# integer # ; # BEGIN # x # := # 23 # + # 5 # ; # z # := # x # DIV # -
# 3 # ; # y # := # z # + # 18 # * # 3 # ; # a # := # x # + # ( # y # -
# 2 # ) # DIV # 4 # ; # END # . #

```

图 1-4 source 经扫描器处理后的一种输出

这里我们只是假定把源程序表示为某种意义上的规整形式。为提供给语法分析程序处理,则应设计更为合适的单词内部格式。一种经常使用的方法是用形如 (Class, Value) 的序偶(即二元式)来作为一个单词的内部表示。其中,Class 为一整数码,用来指示该单词的类别;Value 则是单词的值。

对于程序 source 而言,如果我们假定可将相应语言的单词符号分为四类:① 保留字;②