

中国计算机应用软件人员
水平与职称考试指导丛书

语言处理程序

许耀昌 主编

中国科学技术出版社

中国计算机应用软件
人员水平与职称考试指导丛书

语言处理程序

许耀昌 主编

中国科学技术出版社

内 容 提 要

本书根据《计算机应用软件人员水平考试大纲》有关“语言处理程序”部分编写而成。书中介绍了有关语言处理程序最基本的原理和方法。主要内容有：编译过程、词法分析、语法分析、中间代码、符号表和模式表、目标代码结构、运行环境、代码优化、出错处理和代码生成。各章均设有练习，书末附有试题举例供读者参考。

本书选材贯彻少而精的原则，着重原理、方法和应用，便于读者自学。本书可供参加程序员级、高级程序员级应试人员参考，也可作为工科院校学生及计算机应用在职人员的教材和参考书。

(京)新登字175号

中国计算机应用软件人员

水平与职称考试指导丛书

语言处理程序

许耀昌 主编

责任编辑：胡永洁

封面设计：王序德

技术设计：范小芳

*

中国科学技术出版社出版(北京海淀区白石桥路32号)

新华书店北京发行所发行 各地新华书店经售

北京市燕山联营印刷厂印刷

*

开本：787×1092毫米 1/32 印张：8.25 字数：182千字

1992年7月第1版 1992年7月第1次印刷

印数：1—4200册 定价：5.00元

ISBN 7-5046-0667-7/TP·23

《中国计算机应用软件人员水平与职称考试指导丛书》

编辑工作委员会

主 编：洪民光 教授（中国计算机学会副理事长兼
技术咨询委员会主任）

（以下按姓氏笔划为序）

副主编：李三立 教授（中国计算机学会微机专委会副主任）

许耀昌 副教授（中国人工智能学会、计算机视觉学会副理事长）

孙延军 高工（中国计算机学会学术委员会副主任）

孙强南 高工（中国电子学会计算机工程与应用学会秘书长）

陈火旺 教授（中国计算机学会软件专委会副主任）

杜海萍 高工（《计算机世界》副主编）

龚炳铮 高工（中国计算机学会计算机应用专委会副主任）

编 委：	王人骅	教授	李秀志	副教授
	王 京	讲师	冯宗律	教授
	王顺晃	副教授	许光汉	副教授
	张海藩	教授	赵 磊	讲师
	张福渊	副教授	高锡武	副教授
	吴宝亮	副研究员	葛本倩	副教授
	麦中凡	教授	葛本修	副教授
	陈 凯	编辑	顾庆津	教授

编 者 的 话

本书是《中国计算机应用软件人员水平与职称考试指导丛书》之一。编者根据1990年全国计算机应用软件人员水平考试大纲编写而成。读者对象主要是参加程序员级、高级程序员级的社会各界应试人员。

本书内容适量，既注意介绍经典的编译原理和方法，也精选部分编译的新技术。书中着重介绍有关语言处理程序的基本原理和方法，并结合应用，力求使广大读者便于自学。此外，为便于读者复习，各章均设有练习。书末还附有试题举例，供读者参考。

本书第一章至第六章由许耀昌编写，第七章至第十章由姚军编写，怀力、梁以撒、刘捷、陆莹莹为本书的编写提供了许多有用资料，在此谨表感谢。全书由许耀昌主编和审阅。

由于编者水平有限，加上时间匆促，书中难免存在一些缺点错误，恳切希望广大读者批评指正。

编 者

1991.6

前 言

我国的计算机事业发展，至今已有 35 年的历史。在这段时间里，由于广大科技人员、工人和干部的辛勤劳动，我国的计算机科研、生产、教育、管理等都有了较大的发展。

我国的四个现代化，离不开以微电子产业、计算机产业和数据通信为核心的新兴产业。在“八五”期间及 90 年代的十年规划期间，计算机事业将进一步蓬勃发展。

为了发展计算机事业，就要大力推广计算机在各行各业中的应用。今天，凡是先进的科研、生产、教育都离不开应用计算机。当前关键的任务是培养大量能掌握、管理和使用计算机的技术人才。

为了推广计算机应用软件人才的培养，同时实行“以考代评”的评审制度，深化职称改革，国家劳动人事部、国家科委、机电部、国务院电子办等单位已组成中国计算机应用软件人员水平考试委员会，并从 1990 年开始实行统一的水平考试。据悉，从今年开始将实行水平考试和职称考试制度。

为了提高我国的计算机应用软件人员的素质，并协助程序员、高级程序员级的广大学员参加每年一度的全国联考，中国科学技术出版社组织了一批经验丰富、水平高、在计算机应用领域中有成就的专家、教授编写了“中国计算机应用软件人员水平和职称考试指导丛书”。这套丛书内容适当、概念清楚、理论联系实际、可读性好、针对性强。可以深信，丛书的出版将会满足广大读者的实际需要。

目 录

第一章 编译过程	1
1.1 翻译和解释	1
1.2 编译程序的结构	6
1.3 阶段的分组	10
1.4 编译器的伙伴	11
1.5 编译程序的生成	13
练习一	17
第二章 词法分析	18
2.1 词法分析器的作用	18
2.2 源程序的输入	21
2.3 单词符号的识别	23
2.4 状态转换图	24
2.5 正则表达式与有限自动机	27
2.6 词法分析器的自动生成	41
练习二	47
第三章 语法分析	48
3.1 文法和语言	48

3.2	语法树和二义性	54
3.3	符号串的分析	57
3.4	递归下降分析法	60
3.5	LL(1) 分析法	71
3.6	算符优先分析法	79
3.7	LR分析法	90
3.8	LALR分析法	110
3.9	分析表的自动生成器	116
	练习三	119

第四章 中间代码..... 121

4.1	逆波兰表示法	122
4.2	三元式	127
4.3	四元式	130
4.4	语法制导生成中间语言	134
	练习四	140

第五章 符号表和模式表..... 141

5.1	符号表的组织和使用的	141
5.2	顺序查表与造表法	146
5.3	对半查表与造表法	148
5.4	散列表法	150
5.5	分程序结构语言符号表的组织	154
5.6	非分程序结构语言符号表的组织	158
5.7	模式表	160

练习五	163
第六章 目标代码结构	165
6.1 程序的目标结构	165
6.2 说明和语句的目标结构	166
6.3 过程调用时数据传递的基本形式	172
6.4 循环语句的目标结构	176
练习六	180
第七章 运行时的存储组织及管理	181
7.1 存储组织	181
7.2 静态存储分配策略	186
7.3 栈式存储分配策略	189
7.4 堆式存储分配策略	195
练习七	198
第八章 错误处理	200
8.1 错误处理概述	200
8.2 词法及语法分析阶段错误的处理	202
8.3 语义错误处理	205
练习八	206

第九章 代码优化..... 207

9.1 优化概述 207

9.2 局部优化 208

9.3 循环优化 215

练习九..... 222

第十章 代码生成..... 225

10.1 引言..... 225

10.2 目标机器指令系统..... 226

10.3 一个简单的代码生成器..... 228

10.4 寄存器分配..... 237

10.5 窥孔优化..... 240

练习十..... 243

附录：试题举例..... 245

主要参考文献..... 250

第一章 编译过程

1.1 翻译和解释

用高级编程语言写的程序,在它们上计算机执行之前,必须翻译成等价的机器代码。50年代就出现了高级语言,早期的,如FORTRAN和COBOL语言;而Pascal及Ada则是近期出现的高级语言。把高级语言写成的程序转换成等价的机器语言的翻译程序,就称为编译程序(Compiler)。编译程序的翻译对象称为源程序(Source Program),翻译出来的程序称为目标程序(Object Program)。对一个程序进行编译,包括分析和综合这两大部分。分析是确定程序事先指定的效能,综合是指产生等价的机器代码。

需要指出,编译程序与源语言及计算机有关。

任何一个具体的编译程序都是某一特定机器上的关于某一特定语言的编译程序。我们用 $T_{M,L}$ 表示M机器上关于L语言的编译程序。显然 $T_{M_1,L}$ 与 $T_{M_2,L}$ 是两个不同的编译程序,是在两台不同计算机上同一种语言的编译程序。

T_{M_1,L_1} 与 T_{M_2,L_2} 也是两种不同的编译程序,是同一台计算机对两个不同语言的编译程序。

当既不强调机器,也不强调语言时,我们用T表示编译程序。对编译程序而言,源程序是被加工的对象,目标程序是加工后的结果。这就是说,源程序是编译程序的输入,目标程序是输出。从数学上看,一个编译程序T是一个映射,它把源程序P映射成Q,即 $Q = T(P)$ 。

源程序的执行需要分成两个阶段，即编译阶段和运行阶段，(图1-1)。

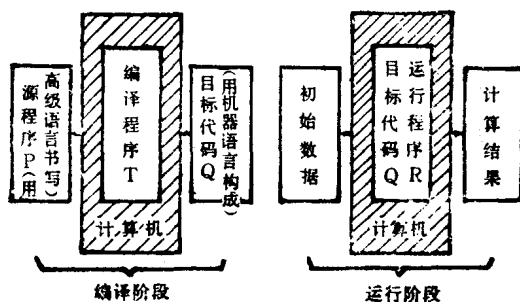


图 1-1 源程序执行的两个阶段

源程序的另一种执行途径是解释途径。解释与编译的区别在于：编译是把整个程序翻译成机器代码，然后再执行；解释程序往往是把程序翻译成中间语言，然后再逐句翻译并同时执行中间语言中的每一条语句。

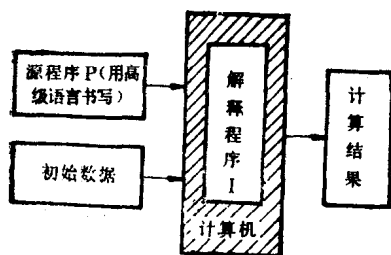


图 1-2 解释执行的示意图

解释程序 (图1-2)

同时处理源程序和数据，它逐条翻译并解释执行程序而不生成目标程序。

为了解释执行源程序，解释程序应该做以下几方面的工作。

第一，换码。换码的主要任务就是把用户的源程序改造成为另一种比较便于处理的中间形式。

第二，语句编辑。语句编辑的任务，是把换码后的内部形

式的语句按语句标号的前后次序由小到大排列好，以便按此次序进行解释执行。如果语言中规定源程序的语句必须有序，就无须做语句编辑工作。例如对于FORTRAN等语言就无须做此项工作。但对于像BASIC这类行结构语言，则必须做语句编辑工作。BASIC语言具有立即执行某些语句的功能，通常把这种功能称为键盘运算的功能。

第三，语法检查。语法检查的任务是检查源程序的语法结构是否正确。

第四，命令处理。命令处理的任务是对命令进行处理。

第五，键盘运算的处理。它的任务是对用户从终端上键入的要求立即执行的语句进行处理。

第六，解释执行。它的任务是解释执行内部形式的程序。

应该指出，解释程序决不能等到用户把它的源程序、命令等算题信息全部输入机器内后，再去进行换码编辑和语法检查，也不能等到换码编辑工作全部结束，再去单独对换码后的内部形式的程序进行语法检查，而必须采用边输入、边换码、边语法检查、边编辑的工作方式。因为，只有这样才能对用户的命令和键盘运算语句，逐句地进行解释执行。

如何把以上四项工作有机地结合起来呢？比较简单的办法是按行接收用户的输入信息。一行中的信息称为信息行。它可能是一个语句，也可能是一个命令。每接收一个信息行，就立即对这个信息进行换码，同时进行语法检查，以便让用户及时地知道当前输入的信息行中是否有语法错误。当一个信息行的换码、语法检查工作结束后，就立即对没有语法错误的语句进行编辑。编辑结束后再去接收下一个信息行。如此一行一行地处理下去，直至处理到命令或键盘运算

为止。解释执行模块再逐句地做解释执行，直至解释执行到程序结束语句为止。由于解释执行速度慢，因此大的计算程序都使用编译程序。

下面我们看一下在IBM/PC机上解释执行的过程。使用的是BASIC行编辑、会话型语言。

```
05 REM THIS IS A SORTING PROGRAM
10 INPUT M
15 DIM A(M)
20 FOR I=0 TO M
25 INPUT A(I)
30 NEXT I
35 FOR I=0 TO M-1
40 FOR J=I+1 TO M
45 IF A(I)>=A(J) THEN 65
50 LET K=A(J)
55 LET A(J)=A(I)
60 LET A(I)=K
65 NEXT J
70 NEXT I
75 FOR J=0 TO M
80 PRINT A(I)
85 NEXT I
90 END
```

当用户从终端输入完毕上述程序，并发布RUN运行命令后，解释程序便立即逐行进行翻译。在解释执行语句10时，它向用户索取数据M（屏幕上出现“？”），当用户应答后（例如键入一个数字，并回车），程序便继续解释执行语句15。如果用户将语句15误输入为：

15 DIM A (M)

解释程序此时将报告如下出错信息:

```
syntax error in 15
```

用户可立即进行修改。如果用户遗漏了输入语句65, 则解释程序执行到语句35时, 将再次报告出错信息:

```
FOR without NEXT in 35
```

上述出错信息, 提醒用户缺少与FOR配对的NEXT。这时用户可立即补充输入语句:

```
65 NEXT J
```

由上可见, 使用BASIC解释程序交互地调试小型程序是很方便的。除了BASIC语言, 像LISP、PROLOG、APL等语言, 也常采用边翻译边执行的解释方式。

与编译程序相比, 解释程序是逐行进行翻译和执行, 效率低。对初次编译的源程序来说, 编译程序可向用户报告它检测到的词法、语法方面的全部错误, 因此像FORTRAN、PASCAL等语言, 一般都采用编译方式进行翻译。有的语言, 如dBASE等, 既有解释型, 也有编译型的翻译程序, 目的是给用户提供一个更有力的编译环境。

除了编译和解释这两种翻译程序外, 还有一种称为汇编程序(Assembler)的翻译程序。

汇编程序是把用汇编语言写的源程序翻译成机器语言的程序。

汇编语言是一种面向机器的低级程序设计语言, 它是一种符号语言。作为一种符号语言, 它是由助记符、符号地址、标号及它们的使用规则所组成。作为一种面向机器的语言, 汇编语言和机器语言一样, 是为某种特定机器而设计的。不同类型机器上所配置的汇编语言各不相同, 不能

通用。

汇编语言把指令、地址、数据都加以符号化，因此它比机器语言易懂、易记、易修改。与高级语言相比，用汇编语言编制的程序具有节省内存、执行速度快，并可精细地控制和使用机器资源等优点，因而常常被用于系统程序、实时控制程序和常用标准子程序的设计。

汇编程序的翻译工作，一般可以用图1-3表示。

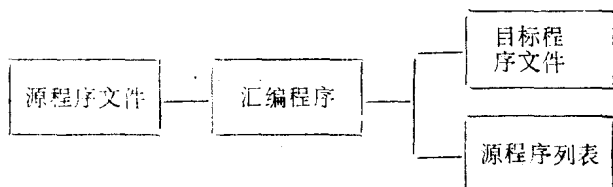


图 1-3 汇编程序的输出

源程序文件作为汇编程序的输入，产生两种输出文件——目标程序文件和源程序列表。

目标程序文件主要是被翻译出来的机器语言程序，它将由机器执行。源程序列表将列出源程序、目标程序的机器语言码以及符号表。符号表是汇编程序所提供的一种诊断手段，它包括源程序中所用的所有符号和名字，以及这些标号和名字指定的地址值，如果程序有错，则可很容易地在这个符号表中找出错误。

1.2 编译程序的结构

图1-4给出一个编译程序的基本结构。为了把源程序翻译成等价的目标程序，编译程序需要做下列六个方面的工作：

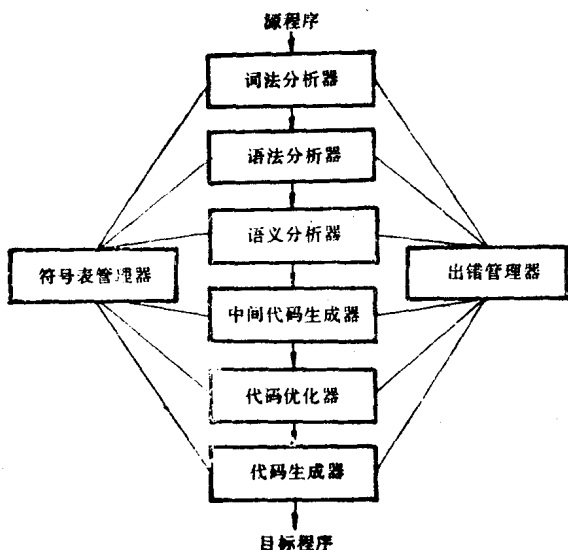


图 1-4 编译程序的结构

第一，词法分析。这部分的主要任务是分析由字符组成的源程序，把它们识别为一个一个的具有独立意义的“单词”，并识别出与其有关的属性，再转换成长度统一的标准形式，这种统一的标准形式，既刻划了单词本身，又刻划了它所具有的属性，称为属性字，以供其他部分分析使用。

第二，语法分析。这部分的主要任务是根据语言的语法规则逐一地分析词法分析时得到的属性字，以确定它们是怎样组成语句的，并说明语句怎样组成程序。分析中如发现有不合语法规则的地方，便把这些出错的地方及错误性质打印输出报告给算题人员；如果没有语法错误，则给出正确的语法结构，以供后面部分使用。

第三，语义分析。这部分的主要任务是根据语法结构分