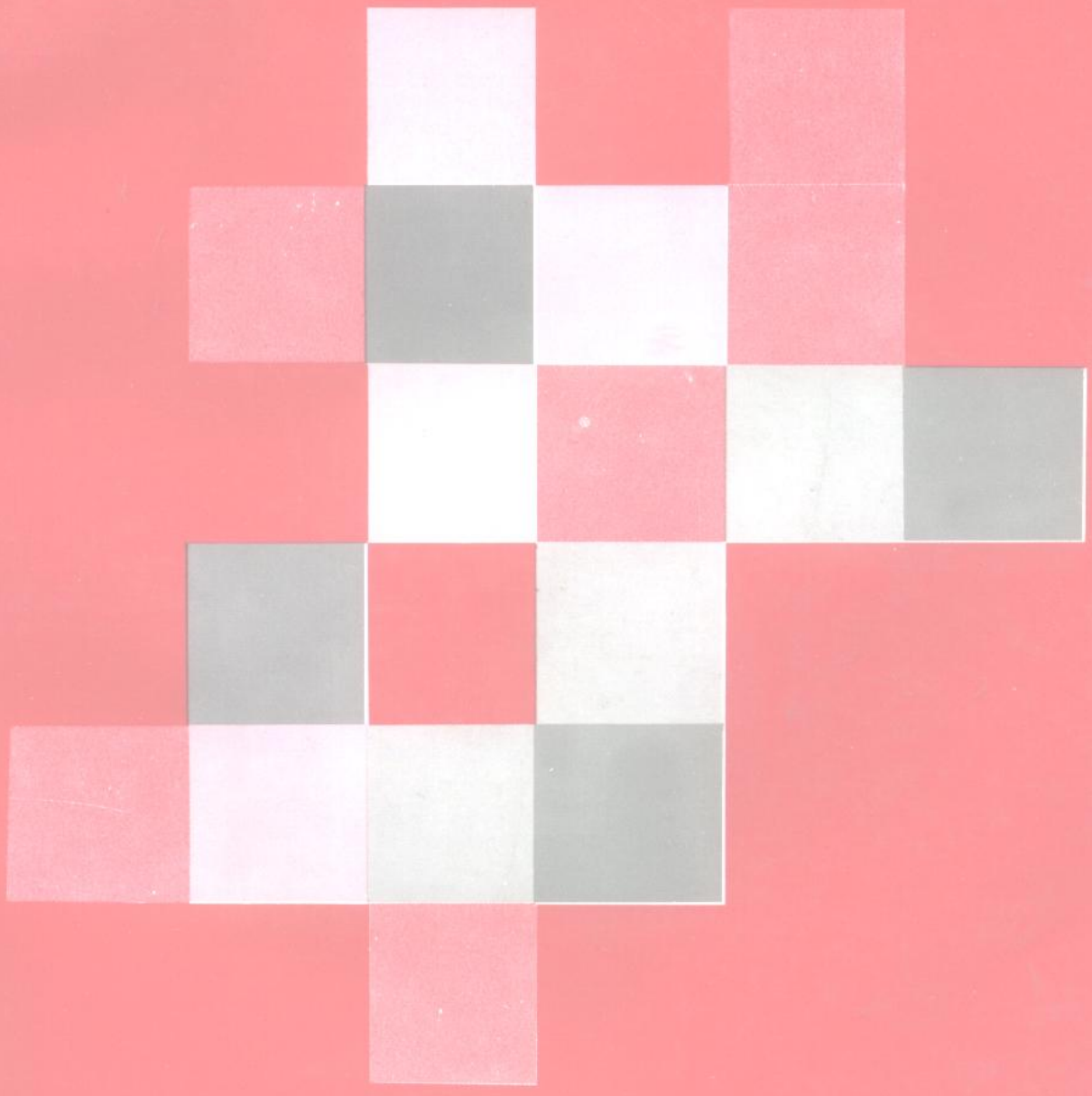
编译方法（修订版）

●胡笔蕊　杜永建　丁　樱　编著



電子工業出版社

编 译 方 法

（修订版）

胡笔蕊　杜永建　丁樱　编著

电子工业出版社

(京)新登字 055 号

内 容 提 要

本书旨在介绍最基本的编译原理与方法，主要内容有文法和形式语言、词法分析与有穷自动机、语法分析、中间语言、符号表、存贮分配、代码生成及优化等。为加强理论联系实际，本书以一个 PASCAL 语言书写的编译程序(与机型无关)为例，并穿插在各有关章节中。本书除配有习题及部分解答外，还附有实习题供学生上机实习。此外，还详细介绍了自行研制的编译计算机辅助教学系统(C-ICAIS)。

本书内容取舍灵活，可满足社会上各层次读者的需要，可作为工科院校计算机专业(包括本科、专科和成人高校)编译方法课程的教材或参考书。

编 译 方 法

(修订版)

胡笔蕊　杜永建　丁　樱　编著

责任编辑　吴　源

*

(本书 1988 年 11 月第一版由测绘出版社出版)

电子工业出版社出版(北京市万寿路)

电子工业出版社发行　各地新华书店经销

北京顺义李史山印刷厂印刷

*

开本：787×1092 毫米　1/16　印张：19.25　字数：480 千字

1994 年 9 月第 1 版　1994 年 9 月第 1 次印刷

印数：4000 册　定价：18.00 元

ISBN 7-5053-2396-2/TP・686

一 版 前 言

本书是计算机专业“编译方法”课程的教材，编者根据计算机编审小组审定的教学大纲及学时数(60～80学时)，并总结了多年教学实践经验，将原有讲义和讲稿整理而成。

在选材方面贯彻少而精的原则，既注意介绍些经典编译方法，也精选部分新编译技术，并尽量减少与机器有关的繁琐细节。基本概念讲透，力求使绝大部分学生能掌握最基本的编译原理和方法，同时还照顾到部分学生考研究生的需要。

在素材的编排上，考虑到教与学的方便，力求由浅入深，循序渐进，前后呼应，分散难点，通俗易懂而又不失科学严谨性。

为加强理论联系实际，使学生了解编译程序的概貌和培养其实际动手能力，以一个PASCAL语言书写的编译程序(与机型无关)作为附录，并以它为例穿插在各有关章节中。此外，每章均有习题，并配备了足够数量的实习题供学生上机实习用。

为适应当前计算机专业发展需要，本教材主要以PASCAL程序设计语言为对象介绍编译技术。

为满足当前社会上各层次读者的需要，内容取舍灵活。例如，第二章中“关系”部分，是为那些没有学过离散数学课程的成人高校学生而设置的，对一般院校学生可略去它。对于成人高校、大专班或学时数较少的学校，教学时可删去书中一些独立或较难的章节，而不影响整体。可供选用的章节计有：§2.8、§2.10、§3.3、§3.4、§4.4、§5.1、§5.5、§7.3中二和三以及§10.3等等。

本教材由武汉测绘科技大学计算机系胡笔蕊、杜永建编写，全书共计十一章，最后两章由杜永建执笔，其余均由胡笔蕊执笔。大部分实习题和习题由武汉测绘科技大学夏启明配置。插图由武汉测绘科技大学王伟绘制。

北京工业大学副教授刘椿年博士在百忙中仔细审阅了全书，并提出了许多宝贵意见，在此谨致真诚的谢意。在出版过程中，得到武汉测绘科技大学一些同志的大力帮助，也借此机会一并表示感谢。

由于编者水平有限，书中难免存在一些缺点错误，请广大读者批评指正。

编 者 1988.2

二 版 前 言

此次修订再版，对书中内容进行了较大调整。除在每章的最后增加了总结性的“要点”之外，新增加了编译计算机辅助教学一章；对原有的第一章、第五章和附录分别做了较多的修改；充实了各章习题并增加了习题解答，附在书末。

计算机辅助教学(CAI 或 ICAI)是当前计算机应用领域中非常活跃的一个分支，它对提高编译课程的教学质量，提高计算机专业学生的素质都很有帮助。本书除对 CAI 作了概略介绍之外，并对一个自行研制的编译计算机辅助教学系统——C－ICAIS 做了较为全面的介绍。

根据当前程序设计语言发展近况，以及不断发展的计算机辅助软件工程 CASE 对编译环境的影响，我们对原书第一章做了较多修改，以期软件业的新动态能及时反映在培养未来计算机工程师的教科书中。

在第五章中，除了新增加一节介绍 LALR(1)分析法之外，对算符优先分析法的算法做了更为严格和实用的描述，这也是研制 C－ICAIS 的收获之一。

为切实加强理论联系实际，使读者更深刻理解编译程序的整体思想与实现过程，特将原有附录修改为 PL/0 编译程序以及上机实习作为第十二章列入正文。该章不仅概括了 PL/0 编译程序各组成部分的功能，还将它与各章中编译理论有机地联系了起来，无形中起到总结全书的作用。

为使学生有更多练习机会，我们充实了各章原有习题。附录中给出了部分较为典型题目或一个题目关键部分的参考答案，这样将有助于学生举一反三自我检查，也有利于减轻教师批改作业的负担。

参加这次修订工作的有胡笔蕊和丁樱。丁樱负责各章习题的充实与习题解答，同时完成了改动较大的第一章和第十二章的修改工作。其余修改工作均由胡笔蕊执笔完成。

本书的出版特别是二版定稿过程中，得到了测绘出版社华彬文同志的大力帮助，在此表示深深的谢意，特别感谢她对教育事业无私的支持。再版过程中，还得到了武汉测绘科技大学陈曼玲副教授和伍春香老师大力支持帮助，在此一并致以谢意。

燕山大学计算机工程系
胡笔蕊
于秦皇岛
1993 年 8 月

目　录

第一章 概 述

§1.1 编译程序

当前的电子计算机都是以系统的面貌出现的，它由硬件和软件组成。软件的功能与质量在很大程度上左右着整个计算机系统的功能。软件品种很多，对通用数字计算机来说，主要包括程序设计语言、系统软件与应用软件。操作系统、编译系统、诊断程序等等均属系统软件。

我们知道，程序设计语言是人类用来和计算机系统进行通信，并控制其工作的人工语言。它经历了几代的发展，最早只有机器语言——不须翻译可直接为计算机所接受，然后从汇编语言、结构化程序设计语言、逻辑程序设计语言发展到今天的面向对象程序设计语言，每类语言的产生都反映了当时人们对程序设计方法的认识和理论研究成果。程序设计语言品种繁多，就以高级程序设计语言来说，近 20 多年来在我们曾广为流行或正在流行的便有数十种，如早期的 ALGOL、FORTRAN、BASIC、COBOL 语言；始于 70 年代的结构化程序设计语言：PASCAL、ADA、C；人工智能语言 LISP、PROLOG；发展于 80 年代并将在 90 年代占主导地位的面向对象语言，如 SMALLTALK、C++、面向对象 PASCAL 等等。

这些高级程序设计语言的设计思想与方法、具备的功能以及应用范畴不尽相同，但有一个共同特点，即用它们编制程序比直接用机器语言编制可大大提高效率。可是迄今为止计算机主要是一种逻辑电子装置，它只能接受用二进制数表示的指令和数构成的程序。那么，它怎样接受源程序并完成计算呢？譬如，对于一个简单的赋值语句 y：=a+b*c，计算机怎样识别它，又怎样将它编制成能反映先乘后加的优先运算关系的一组机器指令，然后根据这组指令完成上述运算，并把结果保存在 y 单元中呢？这就有待于翻译。

通常分两种翻译方式：一种为“编译”方式；另一种为“解释”方式。所谓编译方式，是首先把源程序翻译成等价的目标程序，然后再执行此目标程序。而解释方式是边翻译边执行。它们之间的差别主要在于：解释程序不产生将被执行的目标程序，而是直接执行源程序本身。

如果源语言（编写源程序的语言）是 PASCAL、FORTRAN、COBOL、C 等这类高级语言，而目标语言是某计算机的汇编语言或机器语言，则这种翻译程序称为编译程序。

如果源语言是某一汇编语言，而目标语言是某计算机的机器语言，则称这种翻译程序为汇编程序。

图 1.1 是按编译方式翻译过程的示意图。图中 P 表示源程序（用高级语言写的）；Q′ 表示目标程序（由汇编语言构成的）；Q 表示目标程序（由机器语言构成的）；C 表示编译程序；A 表示汇编程序；R 表示运行（服务）程序；N_1 表示初始数据；N_2 表示计算结果。

图 1.1 中简要地说明了用高级语言编写的源程序在计算机上实现情况，即分为编译和运行两个阶段。实际上，为了得到最终运行结果，编译系统要完成的工作比这复杂。因为目标程序（后缀 .OBJ）由计算机指令、符号信息和其他一些二进制信息组成，它和源程序一样，

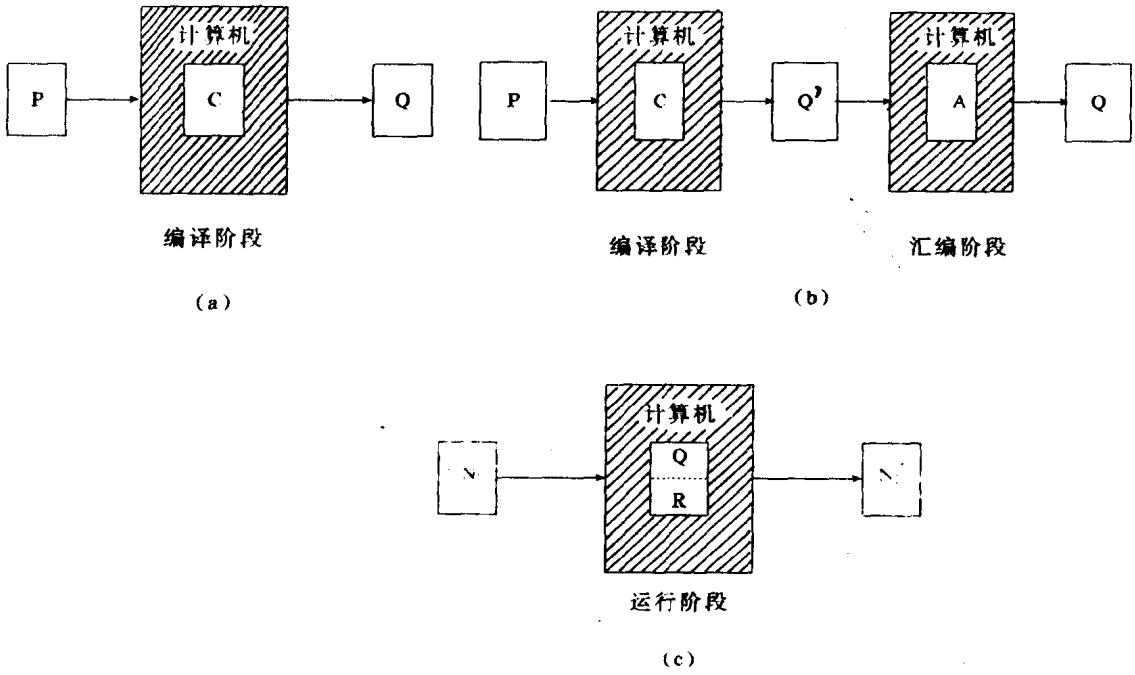


图 1.1　按编译方式翻译示意图

不能由计算机直接执行，必须由连(链)接程序(器)将目标程序和随同编译程序一同提供的系统库(.LIB)连接在一起(目标程序中的符号信息就是提供给连接程序使用的)，经连接后最终形成一个可执行程序(后缀为 .EXE)，可执行程序能在计算机上直接执行。此过程如图 1.2 所示。图中，L 代表连接程序；E 表示可执行程序；.LIB 表示系统库；P、C、Q 含义同图 1.1。

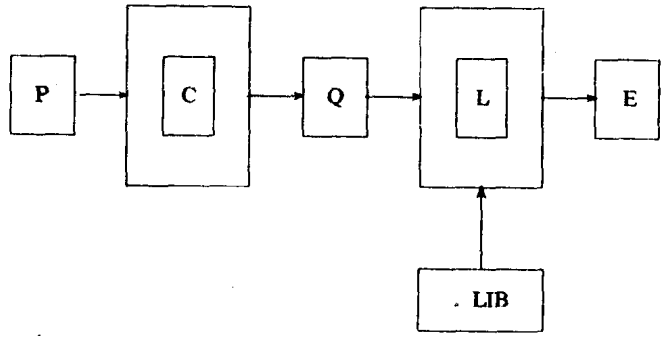


图 1.2　目标程序被链接的示意图

此外，前面提及的编译系统指的是编译程序、连接程序、系统库、源程序编辑程序和其他辅助开发的各种软件的总称。

图 1.1 和 1.2 仅仅示意了源程序被翻译的过程，实际的编译执行过程可用图 1.3 表示得更详尽些。

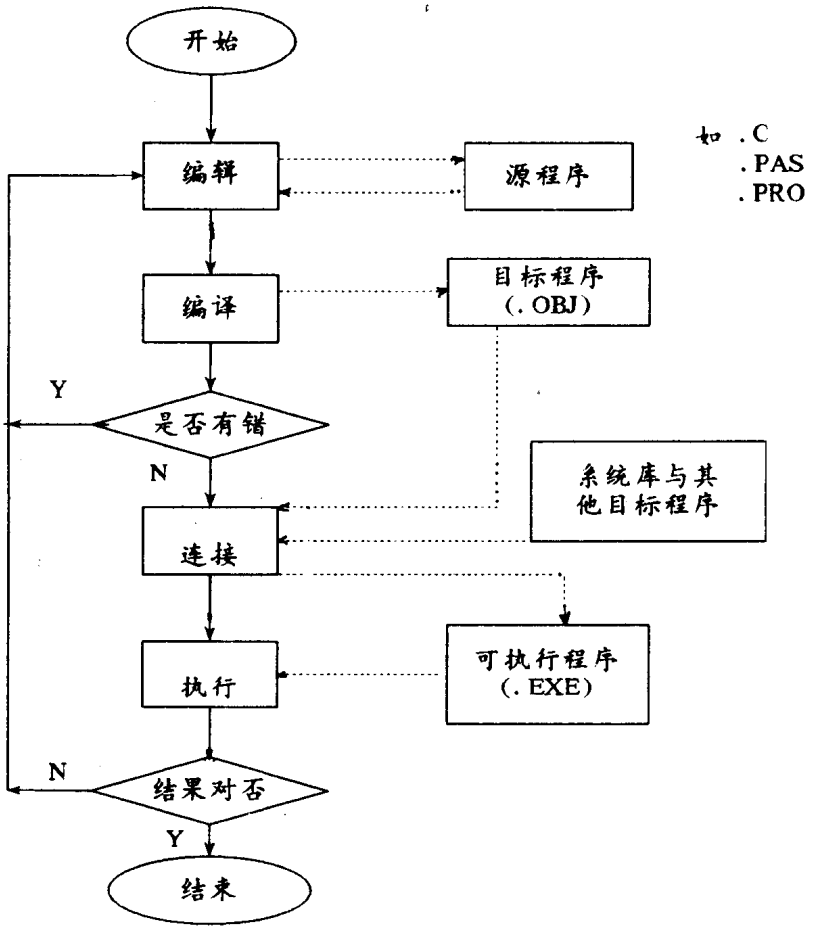


图 1.3　源程序被编译、执行的示意图

例如有一简单的 C 语言源程序：

```
#   include〈stdio. h〉
main(   )
{
  print f ("Hello world! \n");
}
```

其中 # include 是 C 编译程序中的一条编译指令，称预处理，它的作用是指示 C 编译程序将文件(常称头文件)stdio. h 的内容复制到该 # include 指令出现的位置上，因而文件 stdio 中的任何语句都被看成是如同被直接键入到源程序的指定位置上一样。C 编译程序对此经过预处理的源程序进行编译，得到目标文件(. OBJ)。然后连接程序根据此目标文件中保存的信息，从 C 系统库中将实现 print f 的代码加入到用户程序中，形成一个可执行文件(. EXE)。最后运行此可执行文件，便在屏幕上显示字符串：

Hello world!

以上概略地介绍了编译系统的工作过程。本书将主要介绍其中的编译程序，至于翻译

的另一种方式——解释程序，只在下一节简略地作些介绍，当然，很多编译技术同样也适用于解释程序。

§1.2 解 释 程 序

前面已经讲过，解释程序是这样一种翻译程序，它将源程序作为输入接受并执行之，即边解释边执行；或者说逐行进行翻译，即对每一个语句，依次实现“分析语句”、“检验错误”、“执行规定的操作”等步骤。图 1.4 是按这种形式翻译的示意图，其中 I 表示解释程序。

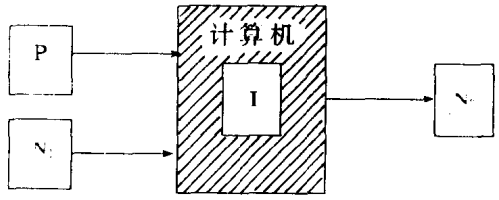


图 1.4 按解释方式翻译的示意图

解释程序的主要优点之一是易于为用户提供调试功能，用户对运行中的程序始终有控制权。解释程序对源程序的语法分析及出错处理都很及时，修改调试都很方便。

由于小型通用语言 BASIC 是一种行结构、会话型语言，所以常采用解释方式进行翻译。以下是一个熟知的简单的 BASIC 源程序(排序)，我们来看看它在 IBM PC 机上解释运行的情况，从中可体会到解释方式的方便之处。

```
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 I=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，它将向用户索取数据 M(屏幕显示“?”)，当用户应答(如 13↙)后，便继续解释执行语句 15。解释程序不仅在适当时刻，向用户询问信息、索取数据，还能及时指出源程序中的错误，例如，若用户原先将语句 15 误输为：

```
15  DIM A(M
```

解释程序此时将报告如下错误信息：

```
syntax error in 15
OK
15  DIM A(M
```

由于解释程序还提供了随时可进行修改的方便，这时用户可立即修改(移动光标，在M之后加“)”)。如果用户漏输了语句65，则解释执行到语句35时，将再次报告出错信息：

```
FOR without NEXT in 40
```

提醒用户，缺少与FOR匹配的NEXT。这时，用户可立即增设语句：

```
65  NEXT J
```

解释程序还允许用户在运行中修改数据，例如若数组A中的个别数据输入有误，如A(3)=－6.2被误为－2.6，用户可运用键盘运算

```
LET  A(3)=－6.2
```

改正它，解释程序将立即执行此键盘运算，然后用户可发布命令GOTO 40，重新排序。

由上可见，使用BASIC解释程序交互式地调试BASIC源程序是非常方便的。

虽然用解释方式实现的交互系统，具有良好的会话性、分时性和独占性，但和编译程序相比，其执行效率低。因为解释程序(以BASIC为例)是逐行进行翻译，若此语句在一个循环体内(例如在一个FOR——NEXT循环中)，每次执行此语句都必须重复上述翻译过程。此外，BASIC程序中的变量存放在一张变量表中，在运行该程序时，每当遇到一个变量，便要从头开始检索这张变量表。类似地，BASIC程序本身的存放与一张标号表相关，运行期间，每当遇到转向语句(例如GOTO或GOSUB语句)，为了确定转向位置，也需要从头开始检索标号表。而编译程序是一次性翻译，即所有翻译工作均在运行之前实现，因此不会出现上述重复翻译以及检索变量表和标号表的现象。此外，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一切错误。由于上述种种原因，像ALGOL、FORTRAN、PASCAL、C等这类高级程序设计语言，一般都采用编译的方式进行翻译。对于BASIC、PROLOG、dBASE、FOXBASE等语言，即有解释型的也有编译型的翻译程序。或者给用户创造一个更强有力的编译环境，即让解释程序和编译程序相互补充。IBM/PC BASIC编译程序就具有此优点，它被设计为能够支持大部分解释过的BASIC程序。因此，用户可以将BASIC解释程序作为调试程序的工具，然后再用BASIC编译程序编译原来那些程序以提高程序执行速度；用户也可以先使用BASIC编译程序对源程序进行初次编译，扫描出全部语法错误并纠正后，再用BASIC解释程序测试源程序逻辑上的正确性。总之，用户可灵活运用此有力的编译环境。

以上是从早期编程环境的角度来评价解释程序与编译程序的。在早期的编程工具支持下，编程工作是按如下方式进行的，即先调用编辑程序建立源程序，再退出编辑程序并调用编译程序产生目标程序，一旦发现源程序有语法错误或目标程序运行时有误，便退出编译程序或运行状态，再调用编辑程序；退出编辑程序后再调用编译程序……，如此反复进行。时至今日，随着软件产业界和学术界对软件工具和环境的研究，给人们带来了更为方便、灵活的集成开发环境。当前很流行的Turbo C、Turbo PASCAL、Turbo PROLOG等程序设计语言便是这样，它们不仅分别有一个快速、高效的编译程序，还具有一个易学、易用的工具箱式的编程环境，即提供了集高性能文件管理、编辑、编译、调试运行为一体的集成开发环境

(IDE)，并有统一的界面，即可以通过一个简单清晰的主屏幕，交互式地使用编辑、编译、运行等各项功能。不仅如此，用户还可以通过操纵主屏幕选择编译、运行阶段的各种模式、不同参数等。例如，可选择"使编译程序不停地进行"或"检查出××个错误暂停编译"等不同方式对源程序进行编译。图 1.5 是 Turbo C 的一个主屏幕，它由四个部分(四个窗口)：主菜单、编辑窗、信息窗和快速参考行组成。其中主菜单有八个选择项：File(文件)、Edit(编辑)、Run(运行)、Compile(编译)、Project(工程)、Options(选择)、Debug(跟踪)、Break/Watch(中断/监视)。Trubo PASCAL 和 Turbo Prolog 等程序设计语言也具有与图 1.5 相似的主屏幕。我们相信，随着计算机辅助软件工程(CASE)不断发展，将给我们创造更有利的编译环境，自然也将对编译程序自身产生不可避免的影响。

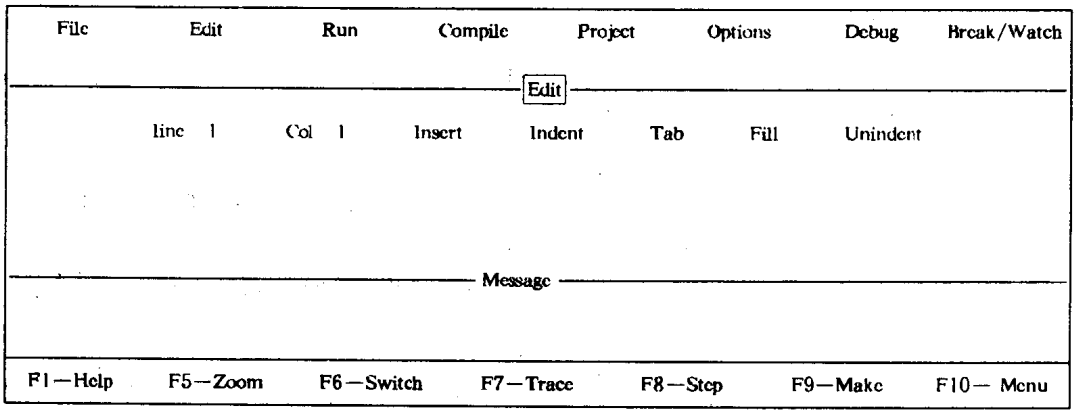


图 1.5 Turbo C 主屏幕

§1.3 编译程序的组成

一. 编译程序的组成部分

为了完成源程序的翻译工作，编译程序要做的事情很多，所以它一般比较庞大复杂。特别是随着程序设计语言功能的增强，其翻译程序也日趋复杂庞大。例如两种混合型的面向对象程序设计语言 C++和面向对象的 PASCAL。C++支持对象、类、方法、消息、子类、继承性以及多继承性；面向对象 PASCAL 支持对象、类、方法、消息、子类和继承性等面向对象概念，它们的编译系统也就相当庞大，例如 C++编译系统便占用十一张高密盘。编译程序除了庞大复杂外，由于各种程序设计语言的风格、功能差异甚大，针对它们进行翻译的编译程序自然不会雷同，即使同一种程序设计语言，由于采用的编译方法和扫描(对源程序从头到尾扫视一次)遍数不同，同时支撑环境也往往不同，所以各种编译程序有着千差万别，但每一个编译程序一般都需要做以下五个方面的工作：词法分析、语法分析、语义分析、代码生成、代码优化等。完成这五个方面工作的程序分别称为：词法分析程序、语法分析程序、语义分析程序、代码生成程序和代码优化程序。这些程序便是编译程序的主要组成部分，在此分别对它们做些简要说明，以后各章节将分别给出更详尽的论述。

1. 词法分析程序

词法分析程序也称扫描程序，它的主要任务是从构成源程序的字符串中识别出一个个具有独立意义的最小语法单位，即所谓单词(如 BEGIN、IF、一个标识符、一个常数、一个运

算符等），有时还指出单词的一些属性，为语法分析打下基础。

例如，以下是某源程序中由 37 个字符（含空格）组成的一个字符串。

IF a1＞0 THEN b1 ：＝(c1＋3) ＊ d1 ELSE b2 ：＝0

词法分析程序将从此字符串中依次识别具有独立意义的 18 个单词：

IF	a1	＞	0	THEN	b1	：＝	(	c1	＋
3	)	＊	d1	ELSE	b2	：＝	0		

一般地，词法分析程序工作方式如图 1.6 所示。

由图 1.6 可见，由左到右从源程序中逐个读取字符并拼写成单词时，可以检查出书写中的词法错误，这项工作由出错处理程序完成。在进行词法分析时，有时可能要与一些表格发生联系，由表格处理程序完成这项工作。

2. 语法分析程序

语法分析程序是编译程序的主要组成部分，它的主要任务是根据语言的语法规定进行语法分析，由单词符号串中识别出各种语法成分，判定当前正在编译何种说明和语句，并将控制转到相应的处理程序去工作。通过这种分析，确定整个单词符号串是否构成一个语法上正确的程序，若发现有不合语法规则的地方，语法分析程序便将出错性质、出错地点等信息通知用户，或对某些错误自动进行修正。

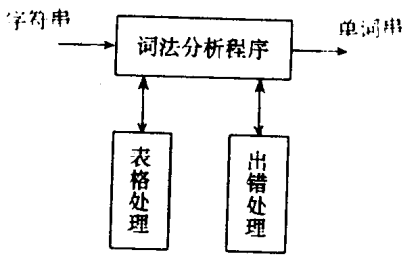


图 1.6　词法分析程序工作示意图

例如，语法分析程序将从如下单词符号串

w ：＝(a1＋b1) ＊ c1

中识别出这是一个〈赋值语句〉语法成分，而赋值等号右端为一个〈表达式〉语法成分。

又例如，对于单词符号串

FOR co ：＝1 TO cy DO

se ：＝3/(se＋1

语法分析程序不仅识别出此〈循环〉语法成份（其中还有〈赋值语句〉、〈表达式〉等语法成份），还将指出其中的语法错误（括号不匹配）。

语法分析过程中，将建立一些表格，也需要在一些表格中查找信息。此外，对于检查出的源程序中的语法错误，一方面向用户提供出错信息，另一方面对某些错误自动进行修正，使语法分析过程得以继续进行。上述工作分别由表格管理程序和出错处理程序完成，与词法分析部分相比，这些程序要复杂得多。

3. 语义分析程序

语义分析程序由许多子程序组成，是编译程序具体处理各种说明和语句的加工程序。它是根据各种语法成份的语义编写的，是完成语义解释的具体体现。例如，对如下条件赋值语句：

w ：＝IF u THEN (a＋b) ＊ c ELSE (a－b) ＊ c