



PASCAL

程序设计

谢培均 编著



南京大学出版社

PASCAL程序设计

谢培均 编著

南京大学出版社

1988·南京

TP319

内容简介

本书结合微型机IBM PC及其兼容机上使用的IBM Pascal,全面系统地介绍了标准Pascal程序设计语言的语句、程序结构、数据类型以及编程方法;并扼要地叙述在微型机IBM PC和APPLE-II上编制、调试、运行Pascal程序的步骤和纠错方法,书末附有错误信息表和TURBO Pascal操作简介。本书内容简明,例题丰富,实用性强。可作为大专院校计算机专业及其他专业学生的教学用书或参考书,也可供从事计算机程序设计的工程技术人员和管理人员参考阅读。

PASCAL 程序设计

谢培均 编著

南京大学出版社出版

(南京大学校内)

江苏省新华书店发行 江苏省阜宁印刷厂印刷

开本: 787×1092 1/16 印张: 14.625 字数: 375千

1988年6月第1版 1988年6月第1次印刷

印数: 1—15000

ISBN 7-305-00067-1

TP·4

定价: 4.15元

前 言

随着计算机的应用深入到人类社会活动的各个领域，迫使越来越多的人掌握计算机应用最基本的工具——程序设计语言。

70年代以来，程序设计语言Pascal被公认为系统地体现了“结构程序设计原则”的，最受欢迎的，最广泛流行的程序设计语言之一，它以其基本结构简单，数据类型丰富，编写的程序易读、易维护，实现效率高，查错能力强，移植性好等特点，广泛用于数值计算和非数值计算的各种应用软件设计。它也用于系统软件设计，特别适用于程序设计教学，因此，目前国内外大多数高等院校计算机专业以及其他专业，选用Pascal语言作为教学的第一程序设计语言，以培养学生严谨的良好的程序设计风格。

Pascal语言受到许多计算机公司的重视，几乎所有大、中、小型计算机上都得到实现，而且在微型计算机上应用起来比较方便，大多数的微型机都配有Pascal编译程序。这些优越的客观环境促使越来越多的人学习Pascal程序设计。

本书是作者在多次Pascal教学实践的基础上，对所编讲义几经修改完成的。

本书注重实用性，结合国内流行的IBM PC微型机及其兼容机上使用的IBM Pascal，全面系统地介绍了标准Pascal内容。内容安排力求准确完整，算法分析比较简明扼要，概念叙述尽量通俗易懂。

本书各章节中共列举了120个例题，尤其是非数值计算方面例题较多，这些程序全部在IBM PC微型机上调试通过，大部分附有运行结果，提供读者阅读、掌握、上机调试，对深入理解Pascal语法规则和编程方法有所帮助。尤其是用自顶向下逐步精细化的设计方法与步骤，列举几个有实用意义的综合性示范程序，它们展示出Pascal程序清楚整齐的结构层次和良好风格。

本书提供了在IBM PC微型机和Apple—II等微型机上调试Pascal程序的一些必备的参考资料。书中反复强调学习程序设计，必须上机调试自己编写的程序，才能提高编程能力、操作能力和纠错能力。因此，在第一章中就详细介绍了Pascal上机调试步骤及错误处理，附录三提供了IBM Pascal编译错误信息，附录四介绍Apple—II等微型机上使用的UCSD Pascal及上机操作步骤和错误处理。附录五扼要介绍TURBO Pascal特点及其操作步骤。

读者通过阅读本书各章节及其示范程序，借助本书提供的参考资料上机调试Pascal程序就不会太困难，并从中逐步学会用Pascal语言编写程序。

在本书编写过程中，得到了合肥工业大学计算机信息系张莫成教授和计算中心孙家启副教授的热忱鼓励，他们审阅书稿并提出有益的建议和修改意见，在此表示感谢。

由于作者学识水平有限，书中的缺点和差错在所难免，诚恳地希望读者批评指正。

谢培均

一九八七年元月于南京

目 录

第一章 程序设计概述	(1)
§ 1.1 计算机系统简介.....	(1)
§ 1.2 Pascal 语言的特点.....	(2)
§ 1.3 Pascal 语言的基本符号.....	(3)
§ 1.4 Pascal 语言的程序结构.....	(4)
§ 1.5 Pascal 源程序的编译和运行.....	(6)
§ 1.6 Pascal 程序调试步骤.....	(7)
第二章 数据、表达式、赋值语句和输入输出	(11)
§ 2.1 数据的概念.....	(11)
§ 2.2 常量定义和变量说明.....	(11)
§ 2.3 标准类型.....	(13)
1. 整数类型.....	(13)
2. 实数类型.....	(15)
3. 布尔类型.....	(16)
4. 字符类型.....	(18)
§ 2.4 表达式与赋值语句.....	(19)
1. 表达式.....	(19)
2. 赋值语句.....	(22)
§ 2.5 输入和输出.....	(23)
1. 输入语句.....	(24)
2. 输出语句.....	(25)
3. 整数类型变量的输入/输出.....	(26)
4. 实数类型变量的输入/输出.....	(27)
5. 字符类型变量的输入/输出.....	(27)
6. 布尔类型变量的输入/输出.....	(28)
习题.....	(30)
第三章 控制语句	(33)
§ 3.1 复合语句.....	(33)
§ 3.2 条件语句.....	(34)
1. 如果语句.....	(34)
2. 情况语句.....	(39)
§ 3.3 重复语句.....	(41)
1. 直到语句.....	(41)
2. 当语句.....	(43)
3. 循环语句.....	(46)
4. 多重循环语句.....	(49)
§ 3.4 转向语句.....	(52)
习题.....	(54)

第四章 过程与函数	(55)
§ 4.1 过程.....	(55)
1. 无参过程.....	(56)
2. 有参过程.....	(56)
3. 值参数和变量参数.....	(57)
4. 全程变量和局部变量.....	(59)
5. 过程的嵌套.....	(61)
§ 4.2 函数.....	(64)
§ 4.3 过程参数和函数参数.....	(68)
§ 4.4 递归函数(子程序).....	(70)
习题.....	(74)
第五章 枚举类型、子界类型和集合类型	(75)
§ 5.1 枚举类型.....	(75)
§ 5.2 子界类型.....	(80)
§ 5.3 集合类型.....	(83)
习题.....	(91)
第六章 数组类型和串	(92)
§ 6.1 一维数组和多维数组.....	(92)
1. 一维数组.....	(92)
2. 多维数组.....	(100)
§ 6.2 高级数组、紧缩数组和布尔数组.....	(103)
1. 高级数组.....	(103)
2. 紧缩数组.....	(104)
3. 布尔数组.....	(106)
§ 6.3 串.....	(108)
1. L串固有函数.....	(109)
2. L串比较规则.....	(111)
3. 简单的字处理实例.....	(111)
习题.....	(114)
第七章 记录类型	(115)
§ 7.1 记录类型的数据.....	(115)
§ 7.2 开域语句.....	(118)
§ 7.3 记录的变体部分.....	(122)
习题.....	(130)
第八章 文件类型	(131)
§ 8.1 文件的说明.....	(132)
§ 8.2 文件的建立和读写操作.....	(133)
§ 8.3 行文文件.....	(137)
§ 8.4 类型文件.....	(140)
§ 8.5 直接存取文件.....	(146)
习题.....	(149)

第九章 指针类型	(150)
§ 9.1 指针.....	(150)
§ 9.2 链表.....	(152)
§ 9.3 二叉树.....	(162)
习题.....	(166)
第十章 程序设计	(167)
§ 10.1 自顶向下的逐步精细法.....	(167)
§ 10.2 调试和错误处理.....	(173)
§ 10.3 综合应用程序举例.....	(176)
附 录	(190)
附录一 ASCII 码.....	(190)
附录二 语法图.....	(193)
附录三 IBM Pascal 编译程序错误信息.....	(197)
附录四 Apple UCSD Pascal 的扩充.....	(210)
附录五 Turbo Pascal 操作简介.....	(222)
参考文献	(226)

第一章 程序设计概述

§ 1-1 计算机系统简介

现代电子计算机是一种广泛使用的重要工具，是一个非常复杂的系统。

计算机系统由硬件和软件两部分组成。计算机硬件是一些执行物理功能的机械和电子线路组合的模块相互连接而成，主要有两部分组成，即处理器(CPU)和输入输出设备(如图1-1所示)。

处理器包括存储器、控制器和运算器，它们完成类似人脑的三种功能：记忆、分析判断和运算，提供了快速处理、加工数据的物质基础，其中存储器以0和1的状态组合来存放数据的编码，它是一种记忆装置。由于计算机的主存储器容量有限，必须借助于外存储器(通常有磁盘和磁带)来记录大量信息。

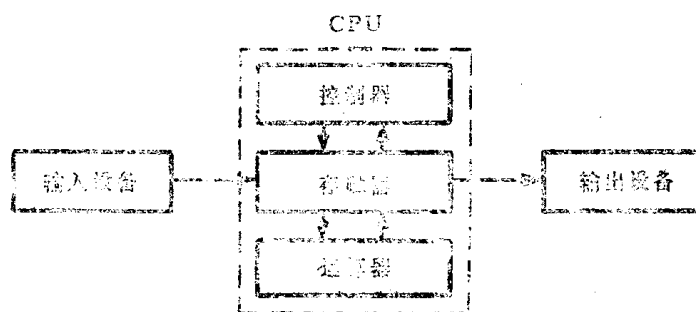


图 1-1 计算机硬件示意图

这两种存储器对计算机系统来说都是不可缺少的部分。输入输出设备中常用的输入设备是卡片输入机和终端键盘，常用的输出设备有打印机、绘图仪和显示器。特别目前普及微型计算机，使用终端键盘和显示器进行人-机对话方式的工作，是很灵活而方便的。

计算机软件由指令和数据组成：一条机器指令对应于计算机执行一个基本动作，一个指令序列称为程序。没有程序，计算机什么动作也没有，也就发挥不出每秒几万、几千万次的计算速度，软件就是由几万至几十万条机器指令所组成。

计算机软件大致可分四层(图1-2)，上层利用下层所提供的手段而构成。

操作系统担负起组织和管理各种软硬件资源，使之协调地、高效地对数据进行处理，使用户较方便地完成各项计算任务。

公用程序包括编辑程序，常用的数学函数库等。

编译程序：每一种语言基本上有一个编译程序。如BASIC编译和解释程序，FORTRAN编译程序，COBOL编译程序和Pascal编译程序等。

Pascal编译程序把用Pascal语言写的源程序翻译成等效的机器指令，组成目标程序。对目标程序的执行，就实现了源程序中所描述的计算工作。

应用程序：大多是用某种高级语言为科学技术、工业、商业和经济管理等领域编写的实用程序。

显然，任何一个程序的运行，既要依赖各种硬件资源，又要用到各种软件资源。计算机系统是庞大的、复杂的，读者可通过亲自应用，熟悉有关部分，而后逐渐做到对整个计算机

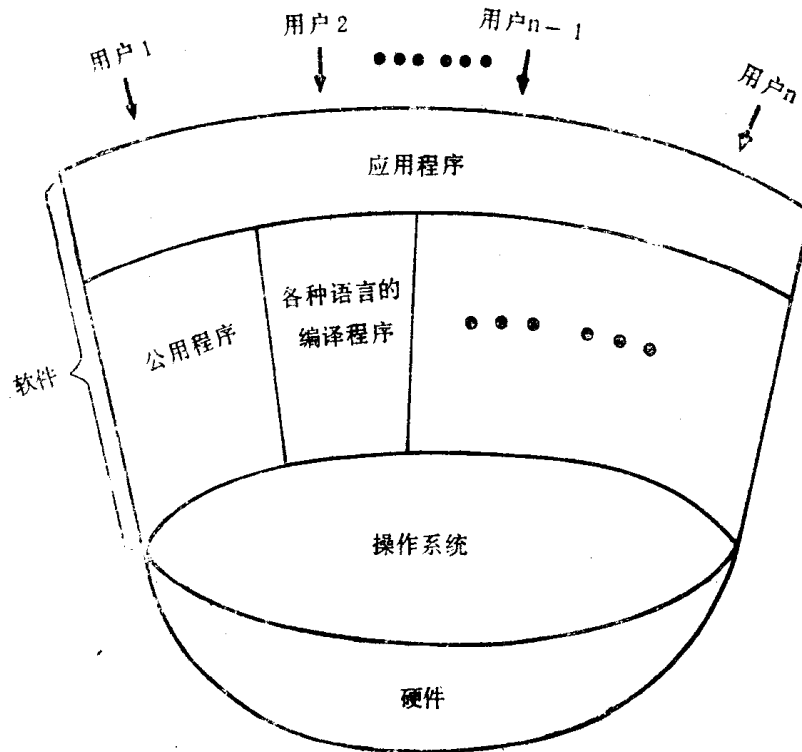


图 1-2 计算机软件示意图

系统运用自如。

§ 1-2 Pascal 语言的特点

60年代末，瑞士苏黎世联邦工业大学的Niklaus Wirth教授设计了一种新型程序设计语言，他为纪念一位最早实用计算器的发明者Blaise Pascal而命名为Pascal语言。

Pascal语言是70年代最有影响和最重要的一种程序设计语言；它是从ALGOL60语言演变发展来的，它增加了非数值计算和数据处理方面的功能，更重要的是Pascal语言体现了荷兰计算机科学家E.W.Dijkstra提出的“结构程序设计”的思想。这种思想为使程序具有结构合理，正确性高，易检验而规定了一套程序设计准则。Pascal语言规则中反应了这些要求和限制。Pascal是一种结构程序设计语言，它具有丰富的数据类型，简明通用的语句，清晰的程序结构，书写格式自由，直观易读。使程序设计教学成为逻辑的、系统的训练，适宜培养良好的程序设计风格。

Pascal也是一种系统程序设计语言，可以用来书写顺序型系统软件，如编译程序等。因此，目前国内外大多数高等院校都把它作为高级程序设计语言的教学内容。它除了广泛应用于高等院校、科学研究、企业等部门的大型计算机外，在微型计算机上Pascal语言也已成为最流行的高级语言，所有主要的微型计算机都配有Pascal编译程序。

IBM PC Pascal与ISO草案 (ISO/TC 91/SCSN595) 相符合，又具有自己的特征和扩展部分。由于Pascal语言的各项扩充，减少了编译过程所需要的时间与空间，故生成程序效率高。IBM PC Pascal能够象汇编语言一样，很容易完成某些操作。

Pascal程序设计语言，已列为高等院校计算机基础教学课程。它是应用计算机解决实际问题的—种必不可少的工具。因此掌握程序设计方法与技巧，并加强程序设计能力培养，与后续课程如软件技术基础、数据结构、算法分析、编译方法、数据处理等的学习关系很大。在实用的软件系统中，绝大部分的应用软件是用高级语言编制的。学习Pascal程序设计的目 的就是为了开发新的应用或为学习已有的应用软件提供必要的基础。

§ 1-3 Pascal 语言的基本符号

任何一种程序设计语言，都需要建立一套自己的基本符号，由这些基本符号，按照语法规则构成该语言语句。Pascal语言的基本符号由以下三部分组成：

1. 基本字符

包括下列三类字符：

(1) 字母：A, B, C, ..., X, Y, Z, a, b, c, ..., x, y, z。按照标准Pascal规定，一个字母可以是大写，也可以是小写，都被看着是相同的字符，但在串中两者不能互换。

(2) 数字：0, 1, ..., 9。

(3) 特殊符号：由运算符和保留字两部分组成。

运算符：+、-、*、/、=、<>、<、<=、>、>=、()、[]、{}、:、:=、...、;、...'。这些基本字符集，实际上是ASCII字符集(参考附录一)的一个子集。

保留字：由字母拼成的字，它们是具有固定意义的单独符号。如：

AND	ARRAY	BEGIN	CASE
CONST	DIV	DO	DOWNT0
ELSE	END	FILE	FOR
FUNCTIO	NGOTO	IF	IN
LABEL	MOD	NIL	NOT
OF	OR	PACKED	PROCEDURE
PROGRAM	RECORD	REPEAT	SET
THEN	TO	TYPE	UNTIL
VAR	WHILE	WITH	OTHERWISE

2. 标识符

标识符，就是用来表示程序、常量、变量、过程、函数和类型等名称的符号。标识符必须以字母开头，后面可以是字母、数字的任意组合。标识符的长度受到机器的限制。在Pascal语言中，通常把标识符前8个字符看作有效的，也就是说，用来表示不同对象的标识符，应当在前8个字符上有所区别。

标识符可以分为标准标识符和用户定义的标识符。

标准标识符是Pascal语言预先给标准函数、标准过程、标准类型、标准常量及标准文件定义的标识符，即语言已提供了标准标识符的名及其性质，如三角函数等，在程序中可直接使用，不必再事先定义。

标准常量: false, true, maxint.

标准类型: integer, real, boolean, char, text.

标准文件: input, output.

标准函数: abs arctan chr cos
 eof eoln exp ln
 odd ord pred round
 sin sgr sqrt succ trunc

标准过程: read readln writeln
 write page pack
 unpack reset rewrite
 new dispose get
 put

用户定义的标识符是根据标识符的定义方法,常常选用能说明标识符的意义或功能的标识符名。一般必须事先定义或说明一个标识符名,才能在程序中使用它。

下列标识符是合法的:

sum
prime
x
filename

下列标识符是不合法的:

char set {在标识符中不能出现空格符}
1ABC {标识符不能以数字开头}
money% {标识符不能有非字母、非数字型字符}
ARRAY {保留字不能作标识符}

程序中使用标识符,要使读者易于理解,便于记忆。常用英语单词表示。如Volume表示体积, Largest表示最大数。注意把字母O与数字0,字母l与数字1区分开。

3. 分隔符

在Pascal语言中,把空格符、行结束符和注释(用花括号{}括起来的一串字符称为注释)作为分隔符。在一个Pascal程序中规定:任何两个相邻的保留字、标识符、数或两个相邻的符号之间,或程序中第一个符号之前都必须插入一个或几个分隔符。但在一个保留字、标识符或数的内部不允许出现分隔符。一个程序可以用少量的分隔符写成,但为了改善程序的易读性,客观上需要自由地使用分隔符,使程序书写清晰美观。

§ 1-4 Pascal语言的程序结构

一个计算机程序是描述要实现的动作以及对这些动作进行操作的数据描述。

一个Pascal程序类似一篇英文文章,由字母、数字、标点符号和空格等按英语语法组成一样,Pascal程序也是由字母、数字和特殊符号按照它的一整套语法规则而组成的字符序列。

先通过一个简单的Pascal程序来初步认识一下完整的程序结构。

```

PROGRAM Example(input, output); {程序首部}
VAR x, square, cube; integer; {变量说明} {分程序}
BEGIN
    read(x); {语句部分}
    square:=x*x;
    cube:=square*x;
    write('square of', x, 'is', square);
    write('cube of', x, 'is', cube)
END.

```

从这个简单程序中，可以看出，一个Pascal程序由程序首部和分程序的两部分组成。分程序中包括程序说明部分和语句部分。

1. 程序首部

程序首部以保留字PROGRAM开头，它给出程序名Example，在圆括号内列出程序的参数(input, output)。程序参数用来表示程序与外界的联系。它从标准文件input读入把结果送到标准文件output中，文件的标识符input和output与具体运行环境有关。例如：在批处理系统中，input可以是一个卡片读入机，而output是行式打印机；在分时系统中，input和output可以是用户目录中已取名的文件；在微型计算机input可以是键盘，而output是显示器屏幕。通过在程序首部指定附加文件的方法，程序可以存取这些文件。

2. 程序说明部分

程序中的定义和说明部分是对数据的描述。Pascal语言中一些标准常量、标准类型、标准过程和标准函数，可以直接使用，而用户自己定义和说明部分由五个部分组成，即标号说明部分、常量定义部分、类型定义部分、变量说明部分以及过程和函数说明。每一部分都可能为空，若不为空则需在说明部分加以说明，然后才能在程序的语句部分使用。

程序的说明部分，必须严格按照如下顺序书写：

- ① <标号说明部分>；
- ② <常量定义部分>；
- ③ <类型定义部分>；
- ④ <变量说明部分>；
- ⑤ <过程与函数说明部分>。

上述简单程序中不出现标号、常量、类型、过程和函数说明，仅有变量说明部分。

3. 语句部分

程序中语句部分是对要实现的动作的描述。它是程序的执行部分，由一系列语句组成。每一个语句执行一定的功能，完成一定的任务，每个语句之间用分号(;)分隔开。语句部分以保留字BEGIN开头，以END结尾，在END后必须加上句号(.)表示整个程序结束。

上述程序的目的是读入一个数，计算它的平方和立方值。程序第一行是程序首部，给出程序名Example，并指出输入和输出文件。第二行是变量说明部分，命名三个数据项为x, square, cube，并说明它们都是整数。其他说明部分不出现。由保留字BEGIN和END括起

来的四个语句组成的语句部分,描述了四个动作:第一个动作读入一个数,并将它的值送入数据项 x ;第二个动作把 x 值自乘,并把其乘积送入数据项square;第三个动作计算 x 的立方值,并把它送入数据项cube中;最后是输出square和cube的值,整个程序以句号结束。

如果输入 $x = 2$,则程序执行结果输出square = 4, cube = 8。当输入其他数据时,同样能输出square和cube计算结果。通常对程序执行一次称为这个程序的一个进程。当然这个程序应该还能完成许多其他的进程,否则这个程序就没意义了。因此从一个程序具有多个进程的概念出发,程序中应有一些数据项在程序执行期间其值是可变的,这些数据称为变量,上例中,sgaure和cube都是变量。

一个程序的组成部分,可以用图1-3所示的语法图描述。从名字为“程序”的图开始,一条通过该图的路径定义了一个语法上正确的程序。语法图给出程序单元的语法——即它们必须遵循的格式。图中的矩形框内是程序单元名称,按该名称引用一个图,由这个图来定义矩形框的意义(分别在以后各章中加以定义说明)。椭圆形框书写的是保留字,圆形框是Pascal程序实际书写的符号。

必须指出,注释不仅可以写在程序首部之后,用来说明程序的主要功能、编写日期等用户要注释的内容,而且可以写在程序的任何位置。注释使程序更为清晰,在复杂的程序中加上注释是非常有益的。有了它,可以增加程序的可读性,便于修改与交流。注释对程序的执行毫无影响。

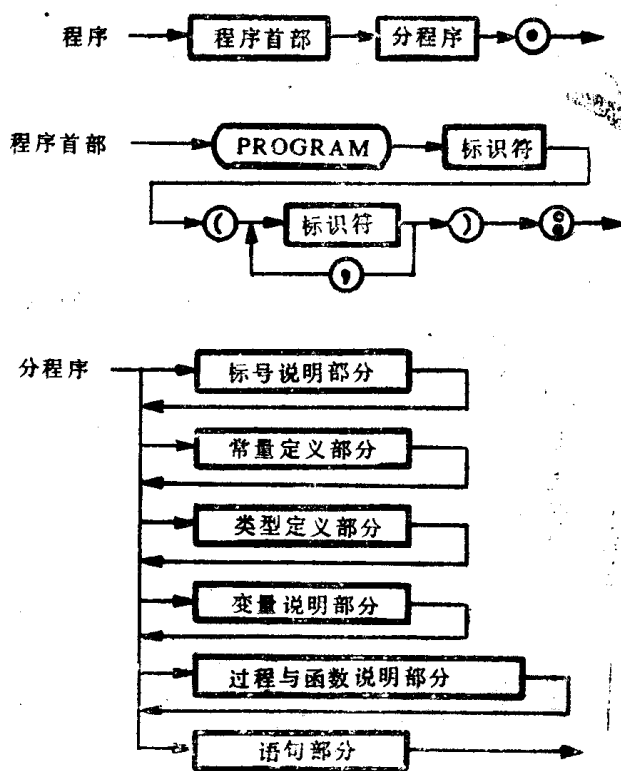


图 1-3 程序结构语法图

§ 1-5 Pascal 源程序的编译和运行

Pascal语言是一种高级语言,用它书写的程序称为源程序。把源程序送到计算机中执行时,分两步进行。第一步编译,调用Pascal编译程序将源程序翻译成机器指令编写的目标程序,如果编译无错误,就能进行第二步。第二步运行,即输入数据后,目标程序运行正常,则输出结果。

在IBM PC上,由图1-4可知计算机执行一个Pascal程序,首先是使用IBM PC提供的行编辑程序或文字处理程序,将源程序输入到计算机内,然后调用Pascal的编译程序,经过两次扫描,完成将源程序转换为目标程序的工作。计算机执行目标程序时,输入它所要求的数据,程序运行完毕则输出结果。

一般而言,一个程序在运行时,总可能发现有某些错误,大型程序更是如此。

错误可以归纳为三类:

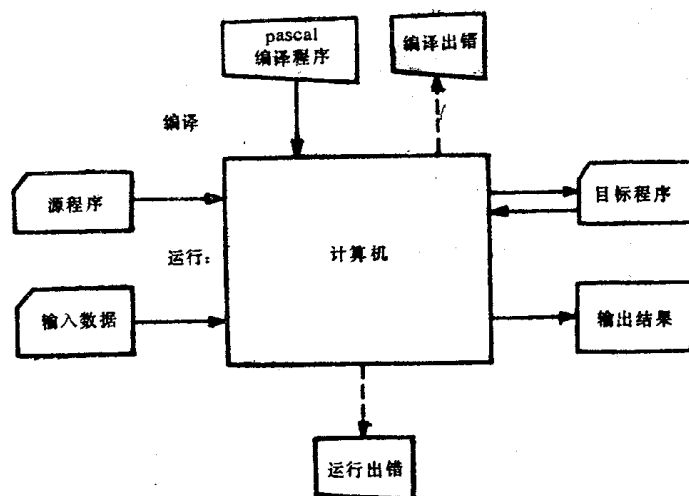


图 1-4 程序编译运行过程图

1. 在编译源程序时产生的错误。这类错误称为编译错误（或语法错误）。例如：一个 BEGIN 漏掉相应配对的 END；变量说明中，变量名为 Count，而在程序中都写成 Comnt；赋值语句 $X := 0$ 写成 $X = 0$ ，等等。

2. 执行目标程序时产生的错误。这类错误称为运行错误。例如：要计算函数 $\text{sqrt}(X)$ 或 $\ln(X)$ 的值，但 X 为负值，除数为 0，数值越界，等等。

3. 在编译和运行时，没有被发现的错误。这类错误称为逻辑性错误。例如 $\text{sqrt}(X)$ 写成 $\text{sqr}(X)$ ，写错了输入数据，或者把描述一个算法的程序写错了。这类错误将输出错误结果。

程序设计中难免发生错误，这就必须将设计的程序，上机运行，输入一组合适的数据，看能否产生预期的结果。如果发现错误，找出语法上或逻辑上的原因，重复检查、修改程序，再进行运算，直到输出正确结果为止。上述过程称为调试。调试是程序开发很重要的部分，为此下一节介绍如何调试 Pascal 程序。

§ 1-6 Pascal 程序调试步骤

在 IBM PC 微型机上，为了顺利地编译 Pascal 源程序，运行其目标程序，成功地获得正确的输出结果，应准备一张复制有以下文件的系统盘。

```

COMMAND.COM
PAS1.EXE
PAS2.EXE
PASKEY
PASCAL.LIB
LINK.EXE
EDLIN.COM

```

如内存容量在 360—512KB, 还可以使用虚拟存储器, 把下列各文件名从系统盘复制到虚拟盘C上, 调试速度就可以大大提高。

COMMAD. COM	MATH. LIB
PAS1. EXE	VDISK. SYS
PAS2. EXE	CONFIG. SYS
SHEN. MAP	LINK. EXE
PASCAL. LIB	

另外, 还需一张格式化后的空盘, 用来贮存用户的源程序和编译过程中产生的其他文件。

1. 建立Pascal源程序文件

使用文字处理程序WORDSTAR或行编辑程序EDLIN在用户盘上建立源程序文件, 取名为FILENAME, 其扩展名为.PAS。首先把系统盘放在A驱动器中。用户盘放在B驱动器中, 置当前盘符为B, 屏幕显示如下(下划横线者为操作者键入的字符, 下同):

```
B> A: EDLIN FILENAME. PAS
```

调入EDLIN程序, 建立FILEAME. PAS源程序文件。

2. 调用Pascal编译程序对源程序编译

(1) 启动编译程序PAS1

PAS1 是Pascal编译程序第一遍扫描。把它调入内存, 用来检查源程序。在当前驱动器为B的情况下, 键入

```
B> A: PAS1
```

经过短一段时间后, 编译程序就显示下面的指示:

```
IBM Personal Computer Pascal compile  
Version 3.20 (C) Copyright IBM Corp 1981  
Source filename [ .PAS ]:
```

其中source filename (原文件名) 是存放用户Pascal源程序的文件名。这时, 需输入原文件名FILNEME, 不必输入扩展名.PAS。编译程序会自动地找到.PAS。下一行提示:

```
Object filename [ FILENAME.OBJ ]
```

其中Object filename 是用户给目标文件取名字。如果希望目标文件仍以上面方括号中的字符为文件名, 不另取名字, 只需按RETURN键即可。下一行提示:

```
Source Listing [ NUL. LST ]:
```

其中 Source Listing 为源程序清单, 用户不需要列清单, 则按RETURN键, 这时, 给出省略文件名NUL. LST, 并通知编译程序, 不要生成源程序清单文件。如果用户需要这份文件, 则键入文件名, 扩展名LST可省略。

下一行提示:

```
Object Listing [ NUL. COD ]
```

其中Object Listing 为汇编的目标文件清单, 用户不需要, 则按RETURN键。

屏幕上显示的提示输入结束, 编译程序就开始对用户的源程序进行第一次扫描, 进行语法分析与说明部分的处理。对语句部分进行语法和语义的检查。如果有语法错误, 就会在屏

幕上显示如下的错误信息或警告信息:

```
ERROR DETECTED
Passone Had Warnings
```

用户必须重新调入EDLIN, 按附录三中列出的 IBM Pascal 编译错误信息提示, 在源程序中找到错误位置, 加以修改:

```
B> A: EDLIN FILENAME. PAS
```

并且重新启动Pas1:

```
B> A: PAS1
```

重复上述的各项输入。如果没有再发现任何错误, 则屏幕上显示:

```
Pass one no Errors Detected
```

编译的结果, 在用户盘上产生两个中间文件: PASIBF.SYM (变量表) 和 PASIBF.BIN (中间二进制代码)。

(2) 启动编译程序PAS2

PAS2 是Pascal编译程序第二遍扫描。把它调入内存, 对PASIBF.BIN中的机器语言进行优化, 产生两个目标文件.COD和.OBJ, 并删除PAS1生成的中间文件。

```
B> A: PAS2
```

编译完成后, 屏幕上显示如下信息:

```
Code Area Size = 016 (278)
Cons Area Size = 006E (710)
Data Area Size = 000D (13)
```

其中Code Area Size(代码表长度)表示源程序所占用的字节总数。Cons Area Size (常量区长度)表示源程序中常量所占用的字节总数。Data Area Size (数据区长度)表示静态分配的数据所占用的字节总数。

3. 连接 (LINKING)

调用LINK程序, 连接编译生成的目标程序与Pascal库文件。连接正确后产生扩展名为 .EXE的一个可执行文件, 称为运行程序。

```
B> A: LINK
```

屏幕显示如下:

```
Object modules [ .OBJ ]: -
```

当键入被连接的文件为FILENAME后, 第二行提示:

```
Run File [ FILENAME.EXE ]: -
```

如果用户要另取一个可执行代码文件的名字, 就须输入其名字, 否则, 按RETURN键即可。

第三行提示:

```
ListFile [ NUL.MAP ]: -
```

按RETURN键, 选取省略文件名NUL.MAP。

第四行提示:

```
Libraries [ .LIB ]: -
```


表示连接Pascal库文件，所有例行程序都在Pascal.LIB中。它们是运行用户程序时需要的。为了响应提示，按RETURN键。

当连接完成后，产生的带有扩展名.EXE的可运行程序FILENAME.EXE被存入B驱动器中用户盘上。

4. 运行用户程序

只要键入文件名，扩展名EXE可省略。

B> FILENAME

如果有输入，键入数据，则将输出结果。因此，调入已编译好的运行程序，进行运算是很方便的。

上述步骤，也可用批文件来完成。名为PASCOM.BAT的批文件，应有以下内容：

A:PAS1 %1.PAS,

A:PAS2

A:LLNK %1.OBJ, %1.EXE, NUL, A:

A:%1.EXE

<Ctrl-Z>

因此，上面操作步骤只须键入

B> A:PASCOM FILENAME

就能自动调入PAS1,PAS2和LINK进行编译和连接。如果编译发现错误，则如前所介绍，必须重新调入EDLIN进行修改，再启动PASCOM。图1-5表示编译文件名FILENAME程序的信息流程图。

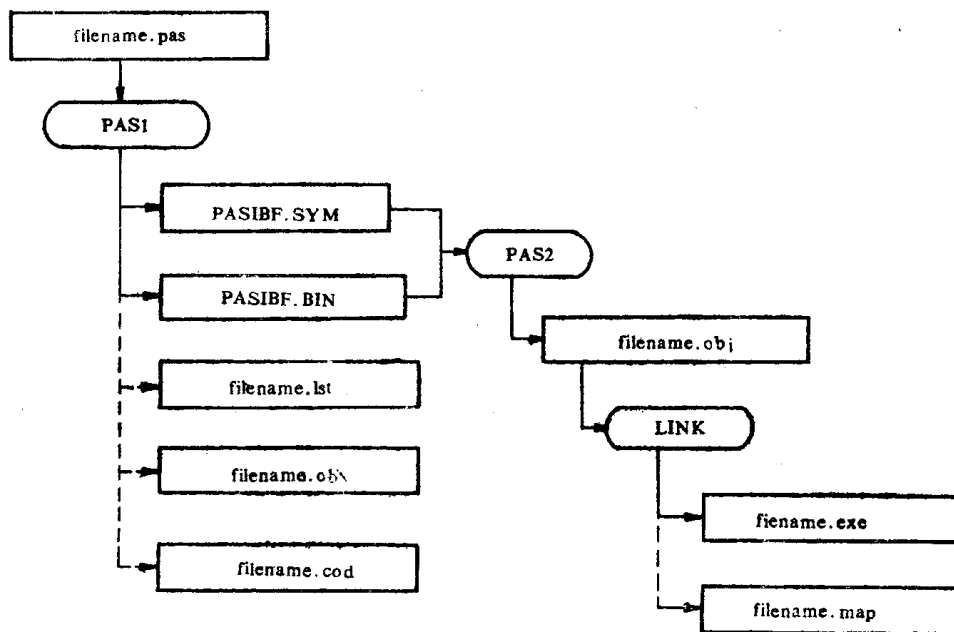


图 1-5 信息流程图