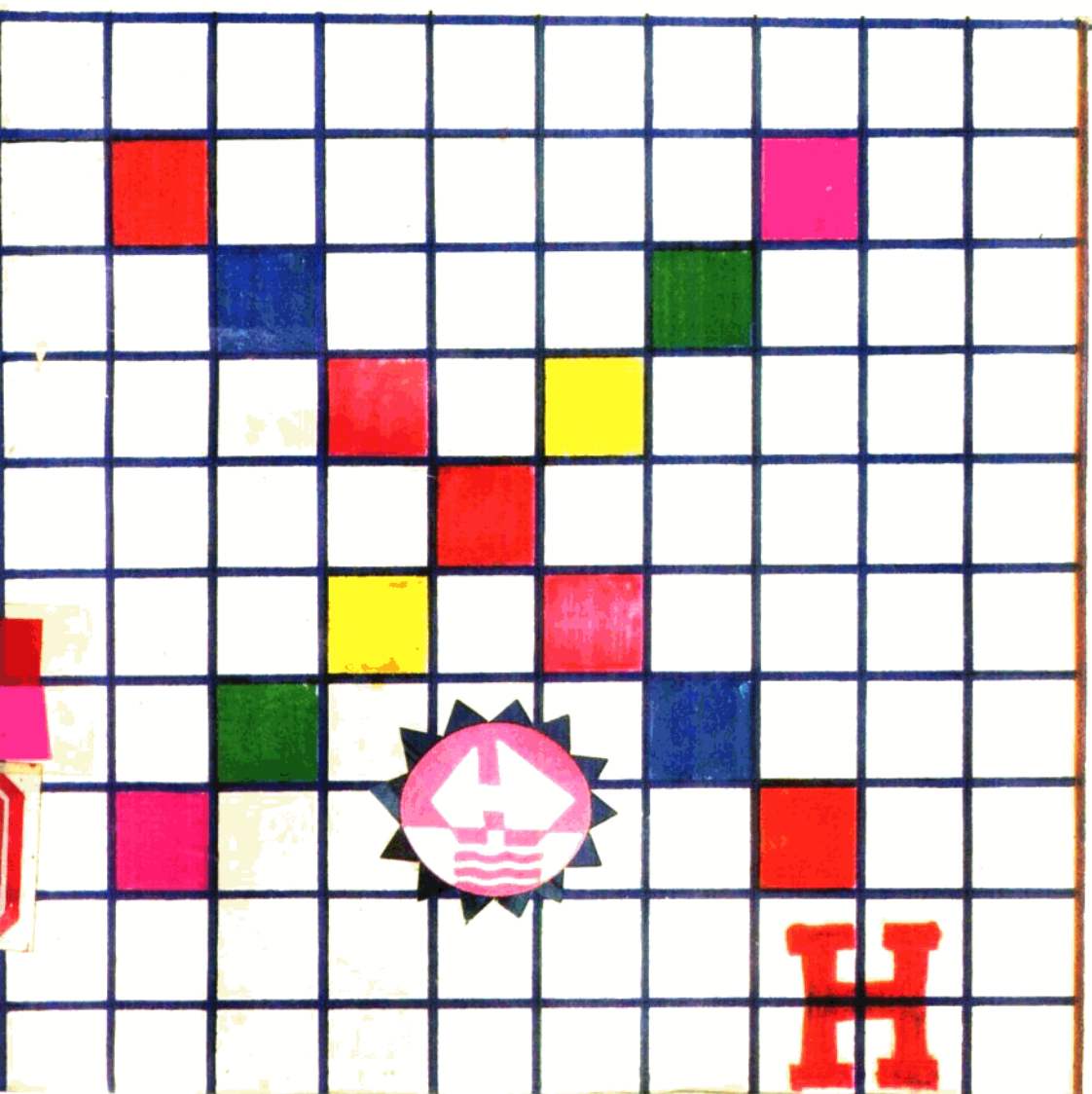


HOPE COMPUTER COMPANY LTD.

适用于IBM—PC 机 长城0520机 WANG—PC 机

PASCAL 程序设计 基础及实用技术

王 诚 赵毓升 奚和泉 编译



北京希望电脑公司

前 言

目前我国已有各类微机数十万台，其中功能比较齐全、用得又比较普及的还是 IBM PC 机、长城 0520 机和 WANG PC 机，它们占已有微机总数的绝大部分，今后每年仍会以数万台的增长速度在发展。

从这些微机的应用领域和使用水平看，总的状况是令人鼓舞的，但也有相当多的使用单位遇到困难。造成这一状况的根本原因，是使用单位的人员得不到必要的培训，自学又受到种种条件，特别是缺乏水平适中、简洁实用的技术资料的限制，在计算机能做些什么事情和怎样使用计算机两个方面缺乏基本和必要的知识。

要解决这一矛盾，我们认为至少应做好如下两项技术方面的工作：

第一，要大力普及一种结构性能良好，语句简明清晰，数据类型完备，易学、好用且运行效率高的程序设计语言，解决能使用计算机的一个关键问题。我们推荐用 PASCAL 语言，因为它完全满足上述要求，正如我国计算机专家金兰教授所说的：“应该创造条件逐步用 PASCAL 语言取代 BASIC 语言在我国目前占据的不适当地位，要把 PASCAL 语言作为推广计算机应用的启蒙语言，让人们从一开始就培养严谨的、良好的程序设计风格和习惯，然后再学习 FORTRAN、COBOL 等带有一定应用偏向的语言。这样做将对人们的程序设计基本训练产生深远的影响。”

第二，要开发出一批能在微机上使用的完全汉字化的新型高性能的关系数据库系统，使其最大限度地符合我国国情和目前广大用户的技术素质与水平。再进一步，最好使这些数据库系统带有一定的开发工具或自生成能力，使一般水平的用户能用这个系统实现自己单位的业务、事务管理，这是促使各单位的计算机真正运转起来并带来效益的重要条件。是让广大用户了解计算机能做些什么事情的最好办法。

基于上述分析，清华大学计算机系的一个科研课题组，在多年教学和科研工作的基础上又经过两年多的艰苦工作，已基本完成在上述机种上运行的一个数据库系统和管理信息系统的开发工具系统。这套软件是以 PASCAL 语言为主并配以少量汇编语言子程序实现的。

《PASCAL 程序设计基础和实用技术》这一资料，是在多年教学实践和科研实践的基础上编写的。我们期望它能对扩大微机应用的范围和使用水平有所贡献，同时也希望它成为将来使用、进一步开发完善我们的软件系统的用户的基础读物。

这本资料主要由两部分内容组成：

第一部分，简明、准确地讲解 PASCAL 程序设计的基础知识。在全面介绍标准 PASCAL 功能的基础上，重点突出了 MS PASCAL 语言扩展功能的用法，针对性很强。这部分内容是以教学讲义形式编写的，循序渐进，由浅入深，并配有很多原理性的小程序，对大多数微机用户和工程技术人员来说，比较容易学懂会用。

第二部分，详尽、分类地介绍了 PASCAL 程序设计中的许多实用技术。这些技术的提出和实现方法，多是以实现数据库系统为例进行讲解的，但必须指出，这是一些通用技术，在其它各应用领域也都有普遍实用意义。这一部分内容涉及面广，难度也比较大，但

非常实用，是在其它书籍资料中难以找到的。

因为本资料的内容是专门针对 IBM PC 机、长城 0520 机和 WANG PC 机编写的，针对性强，新素材多，简明实用，所以对直接使用这些机种的人员更对口。对不使用这些机种的人员来说，不必过多阅读 MS PASCAL 扩展功能的部分。对第二部分中给出的内容，可以看其思路和方法，不必仔细阅读程序。

本资料也可以作为听《PASCAL 程序设计》课并在 IBM PC 机上上机的学生的教材。此时，应先学标准 PASCAL 的功能，再学 MS PASCAL 的部分扩展功能，最后以练习或演示方式学习第二部分中的内容。

目 录

前言	1
第一章 PASCAL语言概述	3
1.1 PASCAL语言的特点和用途	3
1.2 运行MS PASCAL程序的方法和步骤	4
1.3 MS PASCAL语言的特性	6
第二章 简单数据类型和简单程序设计	42
2.1 整型数据及其运算	43
2.2 字节类型、字节类型数据及其运算	45
2.3 实型数据及其运算	46
2.4 PASCAL语言中的算术运算	48
2.5 算术运算程序设计举例	49
2.6 字符类型数据及其运算	65
2.7 布尔类型数据及其运算	66
2.8 枚举类型数据及其运算	66
2.9 子界类型数据及其运算	68
2.10 标准函数及其引用	69
2.11 关系运算和逻辑运算	71
2.12 过程类型	72
2.13 简单程序设计举例	73
第三章 程序设计中的流程控制	78
3.1 顺序结构程序设计	78
3.2 分支结构程序设计	79
3.3 循环结构程序设计	87
3.4 MS PASCAL语言在分支与循环控制结构中实现的扩展功能	91
3.5 循环结构程序设计举例	93
第四章 构造型数据类型	100
4.1 数组类型	101
(高级数组类型, 字符串类型, 变长字符串类型)	
4.2 记录类型和WITH (开域) 语句	109
4.3 集合类型	115
4.4 文件类型	121
4.5 地址类型数据及其引用	139
4.6 指针类型	149
4.7 相同类型和兼容类型问题 (• 有扩展内容)	158
第五章 过程和函数	161
5.1 可用的过程和函数	161

5.2	过程说明和过程调用	174
5.3	函数说明和函数引用	176
5.4	过程和函数的参数类型	176
5.5	程序中的模块与层次结构设计问题	182
5.6	程序设计举例	186
第六章	屏幕上格式数据的输入输出支持	212
6.1	终端硬件特性及其使用方法	212
6.2	实现屏幕格式数据输入输出的基本技术	232
6.3	在用户程序中使用支持软件	240
6.4	在屏幕上实现多窗口入出的技术	244
6.5	屏幕页的概念与用法	248
第七章	制表软件和报表支持功能	255
7.1	制表软件实现的基础	255
7.2	制表软件的进一步完善	261
7.3	屏幕编辑方式制表	268
7.4	程序中直接形成表格	275
7.5	用表格方式从终端录入数据	280
7.6	报表生成及打印	284
7.7	实现绘图功能	289
7.8	打印机的打印控制	299
第八章	索引文件的实现及应用	310
8.1	索引文件的实现原理	310
8.2	支持索引表文件操作的程序	314
8.3	在用户的程序中使用索引文件	331
8.4	实现与使用索引文件中的实际问题	338
第九章	和汇编语言程序的接口关系	339
9.1	问题的提出	339
9.2	汇编语言程序设计的简明知识	339
9.3	在 PASCAL 程序中调用汇编语言的子程序	349
9.4	实现音响控制和仿真电子琴的程序	354
9.5	协同运行多个程序和使用公用数据区	357
9.6	在程序中查询系统信息	370
第十章	文件和磁盘的应用技术	378
10.1	文件控制块 (FCB) 和文件用法	378
10.2	磁盘的盘区分配及读写	389
10.3	文件目录的直接操作	395
10.4	利用串行口实现计算机之间的通信	399
10.5	在 PASCAL 程序中读写 dBASE 的文件	404
附录 A	ASCII 字符集	411
附录 B	MS PASCAL 语言的语法图	412
附录 C	出错信息一览表	422

第一章 PASCAL语言概述

1.1 PASCAL语言的特点和用途

一个计算机系统由硬件系统和软件系统两部分组成。硬件系统是计算机系统实际装置的总称，软件系统通常是指计算机系统运行所需的各种程序及其有关数据。

计算机硬件系统一般包含中央处理机（通称CPU）、内存储器、外存储器（如磁盘机、磁带机）以及输入和输出设备（如终端、行式打印机等）。

计算机软件系统通常包括操作系统、系统实用程序（如编辑程序、装配程序、调试程序）、语言处理程序（如汇编程序、编译程序、解释程序等）以及某些专用程序。

近来，许多计算机系统中，一般还配有不同功能的数据库管理系统、网络通讯系统等软件。当然，在一个实用的计算机系统中，往往还有应用程序（用户程序）这一类软件。

计算机程序设计语言是用来编写计算机程序的工具。可以从不同的角度对程序设计语言进行分类。

根据语言提供功能的水平，可以把语言划分为低级语言和高级语言。低级语言一般指汇编语言，多数是面向具体的计算机的，较难掌握与使用。但用它设计的程序，能直接使用计算机系统中全部的硬件成分与功能；而高级语言一般是面向应用的，与具体的计算机硬件关系不那么密切，亦较标准与规范，且容易掌握与使用。如FORTRAN, BASIC, PASCAL, COBOL, PL/I以及C语言等都属于高级语言。

按照语言的应用范围，可以把语言划分为通用语言和专用语言。通用语言可适用于多种应用领域，如实现各种科学计算、不同部门的事务管理以及研制计算机的系统软件等。上述几种高级语言，还有LISP和PROLOG语言等都属于流行的通用语言。专用语言是指特定用于某一领域的程序设计语言。用在不同领域中的专用语言名目繁多，其专用性限定了它们的应用范围。因此，了解与使用专用语言的人数相对较少。

PASCAL语言，属于通用的高级程序设计语言。由于它完全符合结构程序设计的原则与要求、又具有语句简明灵活，表达能力强，数据类型丰富完备、用户可按需要自行定义等特点，使得它在科学计算、事务管理、系统软件设计和高等学校多门课程的计算机教学等领域得到越来越广泛的应用。再加上PASCAL语言的程序书写格式自由、程序设计的风度优美、可读性好、编译紧凑方便、目标程序运行效率高等特点，正日益受到人们的重视。

关于PASCAL语言的具体特征，我们不妨把PASCAL语言的设计者沃斯教授的一段话摘录在下面，供大家参考。

“我们把PASCAL的特征列举如下：

- (1) 变量说明是强制性的。
- (2) 某些基本字(例如，BEGIN, END, REPEAT等)是‘保留’的，不能作标识符。
- (3) 分号(;)看作是语句分隔符，而不是语句终止符(如同PL/I语言一样)。
- (4) 标准数据类型是整数、实数、字符和布尔类型。基本的数据构造工具有数组、记录(对应于

COBOL 和 PL/1 语言的‘结构’)、集合和文件。这些结构可通过组合和嵌套以形成集合的数组、记录的文件,等等。数据还可以动态地分配并由指针访问。

(5) 集合 SET 数据结构提供了类似于 PL/1 语言的‘位串’的能力。

(6) 数组可以具有任意维数与任意的界;数组的界是常量,即没有动态数组。

(7) 象 FORTRAN、ALGOL 和 PL/1 语言一样,有转移语句。标号是无符号整数,必须加以说明。

(8) 复合语句如同 ALGOL 语言,相应于 PL/1 语言里的 DO 群。

(9) ALGOL 语言的开关和 FORTRAN 语言的计算转语句的功能在 PASCAL 中由情况语句表示。

(10) 循环语句相应于 FORTRAN 的 DO 循环,它的步长只能是 1 (TO) 或 -1 (DOWNT0),并仅当控制变量的值落在界内时执行。因此,被控制的语句可能完全没有执行。

(11) 没有条件表达式和多重赋值。

(12) 过程和函数可以递归调用。

(13) 对变量不存在(象 ALGOL 语言里那样)‘OWN’属性。

(14) 参数有值调用和引用调用,没有名字调用。

(15) 就不存在匿名的分程序来说,这里的‘分程序结构’不同于 ALGOL 和 PL/1 语言的分程序结构,即每个分程序都给一个名字,并做成一个过程。

(16) 常量、变量等所有对象必须在它们被引用之前加以说明。但允许以下两个例外,

- 指针类型定义里的类型标识符;
- 当存在向前引用时的过程与函数调用。

许多人初次接触 PASCAL 语言时,往往感到它缺少某些‘中意的特色’——例如方幂运算符、串的并列、动态数组、布尔值的算术运算、自动的类型转换和省缺说明等,这都不是疏忽遗漏,而是有意删除的。在某些场合下,有了它们反而会招致程序设计解法上的效率不高问题;在另外一些场合,又会感到它们不利于达到程序设计清晰可靠和‘风度优美’的目标。最后,还得考虑对繁多的有效的程序设计功能进行严格挑选,以保证编译程序比较紧凑和高效。”

1.2 运行 MS PASCAL 程序的方法和步骤

为了运行 MS PASCAL 的程序,要求个人计算机的配置如下,

- 个人计算机主机 (CPU 和内存)。
- 终端。
- 双软盘驱动器, 或一个软盘驱动器和一个温氏硬盘。
- 最好再有一台打印机。
- 几片 5 1/4 英寸的软磁盘片。

· 除了有操作系统等系统软件外,当然还必须有 PASCAL 语言的编译程序,它由以下几个文件组成: (WANG PC 机)

- PAS1.EXE
- PASKEY
- FILKQQ.INC
- FILUQQ.INC
- PAS2.EXE
- PASCAL.LIB

在有些机器中,有时还有另外一些文件。

通常情况下, PASCAL的编译程序是以软盘片形式提供给用户的。我们建议用户,最好不要直接使用这几块软盘片去工作,而是首先拷出一份备份来。工作时使用拷贝出来的几块盘片,而把原版盘片保存好,以备由于误操作或正常磨损造成工作盘上的PASCAL编译程序出现问题时,能再进行软件复制用。在系统中配有硬盘的情况下,可把PASCAL的编译程序直接拷到硬盘上。

运行一个MS PASCAL的程序,通常的步骤及其含义如下:

1. 用个人计算机上的编辑程序,建立MS PASCAL语言的源程序,或从已存放在软盘上的文件中拷贝。按DOS操作系统规定,文件名应由1到8个字符组成,扩展名通常应由3个字符组成。任何一个PASCAL源程序文件的扩展名都应为'.PAS'。

2. 用PAS1.EXE对PASCAL源程序进行编译,将得到两个目标文件,其扩展名分别为'.SYM'和'.BIN',分别存放符号表和中间二进制代码的内容。按用户要求,还可以得到一个编译清单文件,扩展名为'.LST'。在编译的过程中,如果发现用户源程序中有错误,编译程序还会把出错情况显示在终端屏幕上。需要时,返回第一步重新编辑用户源程序。

3. 在PAS1.EXE成功地编译了用户的源程序之后,再用PAS2.EXE进行代码生成和代码优化,以便得到用户程序的二进制代码的、浮动地址形式的目标文件。如果PAS1.EXE发现用户源程序中有某些错误,它就不生成符号表和中间代码两个文件,此时PAS2.EXE将不能运行。正常情况下,PAS2.EXE都能正确运行,不会在用户程序中发现新的错误。

4. 用连接程序LINK.EXE建立用户程序的最终运行文件。它要把用户程序、PASCAL.LIB和用户的其它程序块连接成一个程序,在DOS操作系统支持下,能直接装入内存并加以运行。

5. 运行得到最终的目标程序。若发现运行中出现问题或结果不正确时,要到自己的源程序中查明问题,并返回第1步,用编辑程序改正源程序中的错误,并重复第2—第5步,直至得到满意的结果。必须说明,最终目标程序可以被反复运行,不是每运行一次都要重复第1或第2到第5步这个全过程的。

例如,假定用户的源程序名为myfile.PAS,上述五个步骤可如下进行(✓代表按RETURN回车键),

EDIT myfile.PAS,✓ (新建或修改用户的PASCAL源程序)

PAS1 myfile,✓

Object filename[MYFILE.OBJ]:✓

Source listing[NUL.LST]:myfile,✓

Object listing[NUL.COD]:✓

(带下横线部分的内容是PAS1自动给出的提示信息,其它内容是用户从终端上打入的。编译时,用户总得给出被编译的源程序的文件名,这里为myfile,它的扩展名部分.PAS可省略。目标文件通常是需要的,PAS1在方括号中提示该文件名为MYFILE.OBJ,若用户无特殊要求时,可直接按一次回车键回答,以表示认可。源程序的编译清单文件的文件名,PAS1提示的为NUL.LST,当用户直接用回车键回答此项提示时,表示不要求给出此文件;若要求PAS1编译过程中产生此文件,则必须给出一个文件名(如这里的myfile),然后再回车,就会得到myfile.LST文件。最后的一项提示,是目标

代码的反汇编清单文件,其扩展名为'COD'。绝大多数程序设计人员都不使用此文件,因此可直接用回车回答此项提示,那么就不产生此文件。而当用户想取消这三项提示,直接使用 PAS1 在方括号中提示的文件名时,可以用 PAS1 myfile, ✓的方式调用编译程序,即在文件名后打入一个分号后再回车)。

PAS 2 ✓

(打入 PAS 2 后直接回车是正常用法)

LINK myfile, ✓或

LINK myfile ✓或

LINK ✓ :

(第一种方式将取消 LINK 给出的全部提示信息,直接开始连接操作。第二种和第三种方式,用户必须回答 LINK 的每一项提示),例如:

Object Modules:myfile ✓

Run File: myfile ✓

List File: [MYFILE.MAP]: ✓

Libraries []: ✓

Publics [No]: ✓

Line Numbers [No]: ✓

Stack Size [Object file stack]: ✓

Load Low [Yes]: ✓

DSAllocation [No]: ✓

(对这些内容有兴趣的读者请查阅 DOS 手册中关于 LINK 的用法一章,我们在这里不再详细介绍通常情况下用第一种方式即可。如有多个 OBJ 类型的文件要连接到一起,可用空格或 '+' 号把多个文件名隔开)

myfile ✓

(运行得到的最终目标程序)

1.3 MS PASCAL语言的特性

一、概 述

在微机上运行的PASCAL语言有几个性能各异的编译程序,它们是由不同的软件公司设计的。我们这里讲的 MS PASCAL 语言的编译程序是由美国的 MICROSOFT 公司设计的,现在已有了几个版本,其功能是高版本的强,并且能兼容低版本的程序。

MS PASCAL 总的来说是符合 ISO (国际标准化组织) 草案 (ISO/TC 97/SC5 N595) 的,其目的是允许在微机上不加修改地编译并运行标准 PASCAL 语言的程序。

MS PASCAL 语言又扩充了许多新的功能,主要包括以下几部分:

- 编译命令
- 单元 (UNIT)
- 属性
- 高级数组类型
- 字符串
- 常量值设置
- 系统中实现的扩展

1. 编译命令 (COMPILER DIRECTIVES)

MS PASCAL提供了许多条编译命令。它们的作用是为用户提供各种控制、指示编译程序如何运行的手段,使其满足用户的不同要求。包括对出错检查码的控制,列表文件格式控制,条件编译的结构控制,把其他源文件穿插到被编译的源文件中的控制等等。编译命令又称作“元语言”命令 (METALANGUAGE COMMANDS, 或 META COMMANDS)。

2. 单元 (UNIT)

MS PASCAL提供了把一个很大的程序分解为多个程序单元,分别设计、编译,然后连接到一起的结构化性能很好的程序设计手段,这就是UNIT。UNIT就是可以分别设计、编译并可以方便地连接到主程序中的程序单元。比起使用外部过程来,它具有更好的结构化特性,更强的功能和更大的灵活性。一个UNIT通常由接口 (INTERFACE) 和实现 (IMPLEMENTATION) 两部分组成。

3. 属性 (ATTRIBUTES)

变量,过程和函数的属性给用户提供了对文件连接的控制手段。其中包括:

- 供全程标识符连接用的 PUBLIC和EXTERN。
- 供变量用的STATIC 和 REANONLY。
- 供过程和函数用的PURE。

4. 高级数组 (SUPER ARRAY)

MS PASCAL的高级数组类型,指的是要求给出数组下标的下界值,而不具体给出下标的上界值的变长数组。

高级数组类型不能用来在变量说明部分直接说明变量,它的主要用法是:

- 用作过程和函数的变量形式参数
- 用作指针的引用类型
- 用于“派生”出新的数组类型

5. 字符串 (STRING 和 LSTRING)

在标准PASCAL语言中,字符串变量的长度是不能变的。MS PASCAL则又提供了说明与使用变长字符串的能力。变长字符串是一种抽象的数据类型,其定义为:

SUPER PACKED ARRAY [0 ~ *] OF CHAR, 这个串的0号单元用于记忆该串的实际长度。MS PASCAL还提供了一组对字符串进行各种操作的过程和函数。串变量的赋值、比较和读写操作都可以直接进行。

MS PASCAL提供了STRING和LSTRING两种字符串变量,前者的下标上界最大为MAXINT,后者的下标上界最大为255,在用法上它们也有一些差别。

6. 常量值设置

MS PASCAL提供了设置常量值的功能。常量值设置是在编译过程中完成的。其中包括:

- 用已定义过的某些常量定义一个新的常量。
- 给出了串常量连接操作符,还允许对数组和记录常量进行运算。
- 在程序中增设了VALUE段,为有关变量设置初值。
- 过程和函数的形式参数可以用CONST作前缀,其效果与用VAR作为前缀类似,差别在于对应的实际参数可以是一个常量,也可以是变量,并且不能在过程和函数执行体内

修改它的值。

• 数值可以用二进制、八进制、十进制或十六进制表示。在READ和WRITE过程中也支持这种表示方法。

7. 系统中实现的扩展

下面给出的扩展功能,是与使用的计算机系统的硬件、系统软件有直接联系的。用了这些扩展功能的PASCAL程序难于移植到其它计算机系统中。这些扩展是:

- 字 (WORD) 类型
- 地址 (ADDRESS) 类型
- 输入/输出能力
- 交互式的READ
- 随机文件和文件方式
- 内部过程和函数

MS PASCAL 的这些扩展功能,大大增强了语言的表述能力,使得程序设计变得更容易。当然,要正确掌握这部分内容,必须花费一些时间和精力。对于要经常使用微机的人来说,学好这部分内容是非常有益的。

二、字符集和字汇表

在一个语言中能使用的全部字符构成该语言的字符集。一种计算机语言往往还要使用某些英文单词或它们的缩写形式,这些单词构成该语言的字汇表。MS PASCAL 使用的英文单词分为两类,一类是单词含义确定,用户不能变更的,我们称它们为保留字。另一类是单词含义已定,但用户还可以重新加以确定的,我们称它们为预先说明的标识符。我们明确告诫读者,最好不把预先说明的标识符重新定义后当作用户自己的标识符用。为了叙述和学习上更简明清晰,我们将把MS PASCAL划分为四个层次:元语言、标准PASCAL、扩展PASCAL和系统PASCAL。

(1) 元语言

元语言是用以设置编译程序的可选项和进行条件编译的,由全部的编译命令组成。

(2) 标准 PASCAL

MS PASCAL提供标准PASCAL语言的全部功能,使所有标准的ISO PASCAL 程序都能正确地进行编译和运行。

(3) 扩展 PASCAL

MS PASCAL提供了相对“安全”的扩展功能,从而使ISO PASCAL 的功能得到增强,因此程序可以设计得更好。这些扩展包含象BREAK语句,UNIT和结构常量那样的结构等等。

(4) 系统 PASCAL

MS PASCAL还提供了某些更完善的扩展,包括那些对系统程序设计有用的或必要的扩展。这些附加的扩展包括地址类型和RETYPE函数等等。

1. 字符集

基本的PASCAL词汇是由字母、数字、保留字和专用字符组成的。小写和大写字母可以互换,仅仅在字符串文字中不能这样做,是一个例外。

专用字符按照如下的类别分成组,

- 元语言

\$ 只用作元语言的前缀。

- 标准PASCAL

+ - * / = < > () [] , ; ' \ } / < > > = < = := 。在标识符中允许使用 “_” (下划线)。

- 替换

可用 (*...*) 代替 {...}；可用 (, 代替 [；可用 .) 代替]；? 或 @ 可用来代替 ^，以代表一个指针。

- 更高级的替换

“#” 用在整数常量中 (表示是16进制的数)。“_1” 表示行的结束。用 [...] 代替 BEGIN...END。

- 无用的字符 % & ? | ~ , \

空格 (空白) 符、制表符和换页符也是标准字符集的一部分。制表符 CHR (9) 看作是空格符并可以传送到清单文件上。换页符 CHR (12) 被转换成新页元命令 (\$ PAGE+)。

包括在 CHR (0) 到 CHR (31) 中的所有其它字符，上面列出的那些无用的字符，和从 CHR (127) 到 CHR (255) 这些字符用在源文件中时，就会产生错误。仅当它们出现在注释或字符串文字中的时候才是允许的。

(, 和] 配对或 [和 .) 配对是允许的 (但这里不推荐它)。{ 必须同 } 配对，(* 必须和 *) 配对。

2. 字汇表

PASCAL保留字

下面这些字在标准PASCAL中是保留的，

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

MS PASCAL的特性又增加了下列一些保留字，

特性 (FEATURE)

保留字

单元接口

IMPLEMENTATION
INTERFACE
UNIT
USES

模块

MODULE

扩展CASE

OTHERWISE

高级数组类型

控制流程

扩展运算符

地址类型

数值段

SUPER

BREAK

CYCLE

RETURN

XOR

ADR

ADS

CONSTS

VARS

VALUE

属性 (ATTRIBUTES)

“属性”是用来表示变量、过程或函数的特性的关键字。下列的一些属性是保留字:

EXTERN

PUBLIC

READONLY

EXTERNAL

PURE

STATIC

命令 (DIRECTIVES)

“命令”是用在过程或函数块中的字。

EXTERN

EXTERNAL

FORWARD

注意: EXTERN 既是属性又是命令。EXTERNAL 是 EXTERN 的同义词, 它具有同其他几个 PASCAL 兼容的能力。

预说明的标识符 (PREDECLARED IDENTIFIERS)

下列是一些预说明的标识符。它们可以由程序员重新定义, 但这种做法被认为有害无益, 应尽量避免。它们是:

• 标准的

ARCTAN

FALSE

OUTPUT

SIN

ABS

FLOAT

PACK

SQR

BOOLEAN

GET

PRED

SQRT

CHAR

INPUT

PUT

SUCC

CHR

INTEGER

READ

TEXT

COS

LN

READLN

TRUE

DISPOSE

MAXIN

REAL

TRUNC

EOF

NEW

RESET

UNPACK

EOLN

ODD

REWRITE

WRITE

EXP

ORD

ROUND

Writeln

• 扩展的内部特性

ABORT

EVAL

RESULT

BYWORD

HIBYTE

SIZEOF

DECODE

LOBYTE

UPPER

ENCODE

LOWER

• 字符串内部特性

CONCAT	POSITN	COPYLST	
DELETE	SCANEQ	COPYSTR	
INSERT	SCANNE		
• 系统内部特性			
FILLC	MOVER	FILLSC	MOVESR
MOVEI	RETYPE	MOVESL	
• 扩展 I/O 特性			
ASSIGN	DISCARD	READFN	SEQUENTIAL
CLOSE	FILEMODES	READSET	TERMINAL
• 系统 I/O 特性			
FCBFQQ			
• WORD (字) 类型特性			
MAXWORD	WORD	WRD	
• 高级数组类型特性			
LSTRING	NULL	STRING	

注释 (COMMENTS) 注释是括在 { } 或 (* *) 里的部分, 它可以延续多行。用 “!” 开始的是单行注释, 它可延续到行尾。使用相同定界符的注释不能嵌套。

分隔符 (SEPARATORS) 注释、空格或行结束符都被用作为分隔符, 它们只能出现在两个相邻的数或相邻的保留字和标识符之间, 是不能嵌入到数、保留字或标识符的内部。

在某些情况下, MS PASCAL 可以接受没有分隔符的信息, 且不给出错信息。例如:

100MOD#50 作为 100 MOD #50 来接收

100MOD127 看作是 100 跟着标识符 MOD127

分号被用作为语句之间的分隔符, 它不是每个语句的组成部分。

三. 用户标识符

用户标识符, 可以简称为标识符, 是程序设计人员为自己的程序、单元、模块、常量、类型、变量、记录中的域名或记录变体选择域以及语句标号等所选用的一个符号。

标识符的构成规则如下: 标准 PASCAL 语言的规定是, 标识符要由英文字母开头, 后面跟一个或多个英文字母或数字字符构成, 也可以什么都不跟, 仅由一个英文字母构成; 而 MS PASCAL 语言则对它稍加扩展, 规定开头的英文字母之后, 除了可以跟英文字母或数字字符之外, 还可以跟连字符。如 ABC、A12345、A1B2C3、WANG-PC 等等, 都是 MS PASCAL 的合法的标识符, 但最后一个, 即 WANG-PC 在标准 PASCAL 中是非法的, 因为标准 PASCAL 的标识符中不能出现连字符。2A1B、A+C、A.B、A 加 B 等等都是非法的标识符, 它们或者不以英文字母开头, 或在要给出的标识符内出现了不属于英文字母、数字字符和连字符等其它字符。值得特别提一下的是, 标识符内不能出现汉字, 因为微机内部汉字的内部编码不满足标识符的组成规则。除此之外, 我们还特别强调, 不能把 PASCAL 语言的保留字拿来当用户标识符使用, 应避免把 PASCAL 的预说明的标识符用作为用户标识符。

关于语句标号,标准PASCAL语言规定只能用1~9999之间的不带符号的正整数。而MS PASCAL语言则规定,语句标号也可以用标识符表示,并且这些标识符也遵从PASCAL语言对标识符辖域(有效范围)的有关规定。

标识符的长度(字符个数)是任意的,但必须限制在一行之内。MS PASCAL语言规定,一个标识符的前31(标准PASCAL语言规定为8)个字符才是有意义的,长于31个字符的标识符会产生警告性信息。在不造成重名的情况下,标识符以短些为好,形象化些(指名字和它代表的内容一致或接近)为好,这样更简明、清晰、好记,编译时更节省存储空间。

要传给连接程序(LINK)的标识符也可以有31个字符长。

在反汇编目标代码文件中,那些不被连接程序使用的变量名和过程名会被截成只有六个字符的长度。

运行系统所使用的外部标识符,都是由四个英文字母再跟上“QQ”两个字母组成的。因此,为避免混淆,用户不要用这种格式来定义自己的外部标识符。

四、三种可编译的源文件

MS PASCAL编译程序,可以对PASCAL语言的三种源文件进行编译。这三种源文件分别是程序(PROGRAMS)、模块(MODULES)和单元(UNITS)的实现部分(IMPLEMENTATIONS OF UNITS)。三种源文件都可以含有对单元的接口(INTERFACES)。下面我们分三小节来介绍这三种源文件的特性和各自的用法。

1. 程序(PROGRAM)

PASCAL程序是用PASCAL语言设计出来的、用来实现对给定的数据进行某些运算或操作处理功能的一段源文件。它的特点是本身比较完整独立,每个源程序在经过编译和连接操作之后,都可以在计算机的操作系统控制下独立自主地运行。一般来说,几个不同的程序之间的联系,不会在每个程序之内直接存在。即使在由几个程序“协同”完成一件复杂处理任务的情况下,它们的“协同”问题也是由计算机的使用人员在程序之外来解决的。各程序的编译、连接和运行的独立性是程序的一个重要特点。

程序总是由三个部分组成:

- 程序的首部
- 程序的说明部分
- 程序的执行部分(又叫程序体)

下面是一个简单的小程序。

```
PROGRAM ALPHA (OUTPUT, AFILE, PARAMETER),  
VAR AFILE:TEXT,  
    PARAMETER:STRING (10),  
BEGIN  
    REWRITE (AFILE),  
    WRITELN (AFILE, PARAMETER),  
END.
```

这个程序的第一行,构成程序的首部。ALPHA是这个程序的程序名,它是在比该程序中说明的全部静态标识符更高一层的层次上给出的一个标识符,与程序中预先说明的那

些标识符(如INTEGER、READ、PROCEDURE等等)处在同一层次。所以不能把预先说明的那些标识符用作程序名,否则将造成标识符重名的错误。但程序名却可以在该程序的内部作为另一个标识符重新加以说明和使用,因为这完全满足PASCAL程序的标识符分层说明及使用的有关规定。

一个PASCAL的源程序被编译与连接之后,就可以让它投入运行。此时首先要进行文件与变量的初始化操作,要调用一个ENTGQQ过程,程序名将作为一个公用(PUBLIC)标识符被记忆下来。其它的程序(或模块、单元)就可以在其内部用这个程序名或ENTGQQ过程来调用这个程序的程序体,把它作为一个公用过程(PUBLIC PROCEDURE)来使用。这个方法虽然是可行的,但我们不推荐采用这种办法来运行一个程序,这也是程序的独立性的一种表现。

接在程序名之后在一对括号中给出的是该程序的有关参数。程序参数是解决程序和计算机系统衔接、通信的重要手段。在标准PASCAL语言中规定,程序参数应该是文件类型的变量,是用来建立文件变量和计算机系统中的一个实际文件(通过文件名)的对应关系的一种途径。

程序参数和过程参数是很不一样的。第一,程序参数的说明必须在程序的变量说明部分进行,而不是在程序首部的程序参数表中进行。第二,这些参数所对应的实际内容不会被传送到程序体中。从实质上看,它们是在操作系统中进行处理的。

如果在程序参数表中没有给出程序中用到文件变量,操作系统就无法自行解决该文件与操作系统管理的实际文件的对应关系。这时,用户可以用PASCAL语言的扩展语句ASSIGN或READFN把一个字符串变量的值作为文件名指定给文件变量。

在程序参数表中,经常给出两个预先说明的文件INPUT和OUTPUT,分别表示终端的键盘输入和终端的屏幕输出设备。对这两个文件参数的处理办法与对别的程序参数处理的办法不一样,主要表现在,

- 它们是作为预先说明的参数处理的,不必在程序的变量说明部分再对它们进行说明。
- 当程序中用到从终端键盘输入、向终端屏幕输出时,INPUT和OUTPUT就必须出现在程序参数表中,否则编译过程中将会给出警告性出错信息。
- 在用正文(文本)文件的过程和函数进行终端入/出操作时,INPUT、OUTPUT这两个文件名可以在READ、WRITE等语句中省略,并且RESET(INPUT)和REWRITE(OUTPUT)两个打开文件的操作是由系统自动完成的。

• INPUT和OUTPUT也可以被重新定义,但在用文本文件的过程和函数执行终端入/出操作时,总是按对INPUT和OUTPUT的在预说明中的初始定义进行处理。

一个程序也可以没有自己的程序参数部分,这时,程序首部就仅由保留字PROGRAM后跟一个程序名组成。在这样的程序中,INPUT和OUTPUT文件是不能使用的。

INPUT和OUTPUT之外的其它程序参数,会在程序初始化期间,通过执行READFN语句取得各自的值。MS PASCAL对标准PASCAL的程序参数部分进行了扩展,也允许把文件类型之外的变量作为程序参数,但规定它们必须属于能用READFN来读它们的值的那些数据类型(如整型,字类型,字符类型,布尔类型,实型、枚举类型和子界类型、指针类型、字符串类型和文件类型),并且必须是一个完整的变量,不能是一个变量的一个组成成分(包括地址类型的.R和.S分量在内)。

在用READFN调用来取得程序的每个参数的值时,是要用到INPUT文件的。在读这

些参数值之前，首先要调用子例行程序PPMFQQ，让READFN去读从名字为PPMUQQ的DOS接口子程序那里返回的字符序列。PPMUQQ是从启动执行PASCAL目标程序的命令行中得到这些字符的。PPMFQQ和PPMUQQ两个子例行程序都要用到PASCAL的目标程序名。需要时，可以让系统给出对程序参数的提示。

以本小节开始给出的程序为例，看一下启动运行它的目标程序的几种方法。假定这一目标程序的文件名为ALPHA·EXE。

第一种方法，给出文件名后直接回车，要全部提示。

```
A)ALPHA✓  
AFILE:DATA.FIL✓  
PARAMETER:ABCDEFGHIIJ✓
```

第二种方法，在文件名后面给出参数AFILE的值以后回车，只要最后的一项提示。

```
A)ALPHA DATA·FIL✓  
PARAMETER:ABCDEFGHIIJ✓
```

第三种方法，在文件名后给出AFILE和PARAMETER的值以后再回车

```
A)ALPHA DATA.FIL ABCDEFGHIIJ✓
```

在用这第三种方法运行文件名为ALPHA·EXE目标程序时，第二个程序参数PARAMETER的值将从直接跟在DATA·FIL之后的空格读起，使给出的字符串的最后一个字符“J”不能被读到。

必须指明，如果给出一个LSTRING类型的程序参数，程序运行时会出现某种含意不清的局面。如果有下面一个程序：

```
PROGRAM PARMS (LN) ;  
VAR LN: LSTRING (10) ;  
BEGIN  
  :  
END.
```

在对它经过编译和连接之后，用户用下述办法启动执行它的目标程序（文件名为PARAMS·EXE），

```
A)PARAMS✓
```

此时就没法区分，用户是为LN参数送入了一个长度为零的值呢，还是希望得到一个对LN的提示信息后再送入LN的值。MS PASCAL总按后一种情况进行处理，就是说，在可以给出一个LSTRING类型的程序参数的值的位置上没有给出相应的值时，就在下一行给出对这个参数的输入提示。

当某些应用程序人员期望使用先进的命令行处理功能时，它可以通过调用UNIT U中的PPMUQQ过程来完成。为此，应该把程序首部做成无程序参数的形式。

2. 模块 (MODULE)

MS PASCAL语言中的模块，可以被看成是无执行体的程序。它们的主要用法，是用来把一些独立的模块简便快速地组合成一个更大的程序。每个模块可以被单独编译，但不能单独地执行连接操作，也不能被独立地加以运行。

模块总是由两部分组成：

- 模块的首部