

作者：游疆来、丁 宁

TURBO PASCAL V3.0.4.0

高级实用编程指南

北京科海培训中心

1990.1

TURBO PASCAL V3.0, 4.0

高级实用编程指南

游疆来 丁宁

北京科海培训中心

一九九〇年一月

前 言

近年来,随着计算机应用的逐渐普及,广大计算机用户已不再满足于对通用商业软件的使用,或编写单一的计算程序,而尝试着编制适合自己需要的应用程序。但往往由于缺乏经验或相应的资料,难以使自己编制的程序功能完善起来,使其具有一定的通用性。譬如在用户介面设计、键盘录入处理、系统功能调用等许许多多的问题上均感棘手,并且在选择编程语言时也遇到困难,因为程序的主体模块往往要用高级语言编写,而多数高级语言又缺乏低级调用的能力。

在应用中,我们发现Turbo Pascal是一个十分理想的应用程序开发环境,它不仅提供了高效的数值运算功能,还提供了系统的低级调用手段;另外在动态内存管理、文件操作、输入、输出管理等方面也是功能齐备。在国外,Turbo Pascal已成功地用于编制CAD、写作系统和文字处理等应用软件。

我们用Turbo Pascal编写了大量的程序,编程中,对一些基本的编程方法作过专门的收集和整理,制作了一些通用的工具程序。在此基础上,结合国外新近出版的有关资料,写成了此书。我们期望这本书能给广大计算机用户提供一定的帮助。

书中的每个专题都提供了完整的工具程序或示范程序。这些程序均用Turbo Pascal V3.0和V4.0写成,其中大部分程序都作了版本转换说明。在介绍编程方法的同时,也简要介绍了Turbo Pascal的有关功能,用户可以直接将书中的程序插入到自己的程序中,从而增添许多的方便。

全书共分十三章,其中包括系统功能调用、目录查寻、位操作、日期和时间的处理、键盘录入处理、视频显示技术、文本窗口、图形窗口、菜单编程、扩充内存和扩展内存的使用、特殊字符显示等等。第一、五、六、九、十二、十三章由丁宁编写;第二、三、四、七、八、十、十二章由游疆来编写。

在书的末尾附有三个附录,附录一提供了Turbo Pascal V3.0, V4.0基本功能的对照查找表;附录二是Turbo Graphix ToolBox的使用说明,这是作者通过仔细阅读其源程序而写出的,这个工具包是非常适合拥有CGA, Hercules显示适配器的用户使用的。附录三是我们请北京师范大学的於国荣编写的,这一部分综合讲述了目前较为流行的几种显示适配卡(包括CGA, MCGA, EGA, VGA, COLOR400(或EGC400), Hercules)的功能构成和基本使用方法。

在编写这本书的过程中,我们得到了多方面的帮助和支持。冶金部自动化所的杨飞强、清华大学的史元春和编辑夏非彼同志为书稿的抄写和程序的录入与审校付出了艰辛的劳动,我们在此表示衷心的感谢!

因受篇幅的限制,我们原准备的Turbo Pascal的InLine代码转换、外部过程的编制、中断处理程序、内存驻留程序的编写方法、编制DOS外壳(shell)等方面的内容

将作为此书的续篇与读者见面。

由于水平有限，加之时间十分仓促，书中难免存在不妥之处，望读者批评指教。

游疆来、丁宁

一九八九年十二月二十三日于
北京师范大学

目 录

第一章 系统功能调用	(1)
§ 1.1 系统功能介绍	(1)
§ 1.2 在Turbo Pascal中调用系统功能	(4)
§ 1.3 DOS功能调用实例	(6)
§ 1.3.1 获取DOS版本号	(6)
§ 1.3.2 报告当前驱动器	(7)
§ 1.3.3 报告当前目录	(9)
§ 1.3.4 获取扩展错误码	(11)
§ 1.3.5 获取校验状态	(16)
§ 1.3.6 设置校验状态	(17)
§ 1.3.7 获取磁盘自由空间	(18)
§ 1.3.8 获取或设置文件属性	(21)
§ 1.3.9 获取或设置系统日期和时间	(25)
第二章 目录查寻	(30)
§ 2.1 DOS目录查寻方法	(30)
§ 2.1.1 方法介绍	(30)
§ 2.1.2 简单的目录查寻	(32)
§ 2.2 文件信息的解码	(36)
§ 2.2.1 获取当前DTA地址	(37)
§ 2.2.2 建立新的 DTA	(38)
§ 2.2.3 对时间的解码	(39)
§ 2.2.4 对日期的解码	(40)
§ 2.2.5 返回文件长度	(42)
§ 2.2.6 返回文件名	(43)
§ 2.3 综合目录查寻	(44)
§ 2.3.1 对当前目录的查寻	(44)
§ 2.3.2 复杂的可变目录查寻	(46)
第三章 位操作与有效数据存贮	(56)
§ 3.1 二进制数位的应用和有效数据存贮	(56)
§ 3.2 二进制数位的操作	(58)
§ 3.2.1 位设置	(58)
§ 3.2.2 位测试	(59)
§ 3.3 位映象样板	(61)
§ 3.4 位操作应用举例	(62)

第四章 处理和存贮日期与时间	(67)
§ 4.1 日期和时间的存贮.....	(67)
§ 4.1.1 讨论.....	(67)
§ 4.1.2 对三字节日期的编码和解码.....	(67)
§ 4.2 朱利安日期.....	(69)
§ 4.2.1 日期的加减.....	(69)
§ 4.2.2 根据日期确定周日.....	(70)
§ 4.2.3 将朱利安日期转换成公制历法日期.....	(72)
§ 4.2.4 检查并校正日期.....	(73)
§ 4.3 系统日期和时间.....	(74)
§ 4.3.1 检查和设置系统日期与时间.....	(74)
§ 4.3.2 一个处理日期的键盘输入程序.....	(78)
§ 4.3.3 一个示范程序TestDate	(80)
§ 4.3.4 计算延时.....	(81)
第五章 键盘输入	(84)
§ 5.1 控制键状态.....	(84)
§ 5.1.1 获取、设置键盘状态字节的内容.....	(85)
§ 5.1.2 在程序中使用过程 KbdCtrl	(87)
§ 5.2 过滤击键码.....	(88)
§ 5.2.1 非打印字符.....	(89)
§ 5.2.2 检测键盘返回码.....	(96)
§ 5.2.3 简单键盘数据输入控制.....	(97)
§ 5.3 处理复杂的数据输入.....	(98)
§ 5.3.1 数据输入程序.....	(98)
§ 5.3.2 过程InKey的输入参数.....	(103)
§ 5.3.3 过程InKey的返回参数.....	(103)
§ 5.3.4 过程InKey的操作.....	(104)
§ 5.3.5 在程序中使用过程InKey.....	(105)
§ 5.3.6 数据输入介面.....	(105)
§ 5.3.7 使用Help屏幕	(107)
§ 5.3.8 退出数据输入项.....	(108)
§ 5.3.9 在程序中使用数据输入过程.....	(108)
§ 5.3.10 处理控制字符和功能键.....	(109)
§ 5.4 数据输入模板.....	(110)
§ 5.4.1 模板文件.....	(111)
§ 5.4.2 创建模板文件.....	(111)
§ 5.4.3 读取模板.....	(115)
§ 5.4.4 使用模板显示数据.....	(117)
§ 5.4.5 使用模板输入和修改数据.....	(119)

§ 5.4.6	在程序中使用模板控制数据	(122)
§ 5.4.7	开发更为完善的键盘输入介面	(125)
§ 5.5	扩展键盘控制和屏幕显示	(125)
§ 5.5.1	配置文件CONFIG.SYS	(125)
§ 5.5.2	设备驱动程序ANSI.SYS	(126)
§ 5.5.3	扩展键盘控制	(126)
§ 5.5.4	扩展屏幕控制	(129)
§ 5.5.5	设备驱动程序的管理	(130)
§ 5.6	编写键盘程序	(131)
§ 5.6.1	屏幕颜色变换器	(131)
§ 5.6.2	打印机控制器	(136)
第六章	文件保护	(138)
§ 6.1	保护数据文件	(138)
§ 6.1.1	数据文件保护技术	(138)
§ 6.1.2	复制数据文件备份	(139)
§ 6.1.3	删除数据文件备份	(142)
§ 6.1.4	使用数据文件备份恢复被损坏的文件	(143)
§ 6.1.5	在程序中使用数据文件保护过程	(146)
§ 6.2	自动备份文件	(149)
§ 6.2.1	检查文件大小	(149)
§ 6.2.2	提供自动备份功能	(150)
§ 6.2.3	在程序中调用自动备份过程	(152)
§ 6.3	限制程序的寿命	(153)
§ 6.3.1	使程序在指定日期后不能运行	(153)
§ 6.3.2	在程序中调用过程KillDate	(155)
§ 6.4	文件加密	(156)
§ 6.4.1	在文件中加入密码	(156)
§ 6.4.2	解除文件中的密码	(160)
第七章	视频显示	(164)
§ 7.1	ROM BIOS的显示功能调用	(164)
§ 7.1.1	显示状态控制	(165)
§ 7.1.2	光标控制	(167)
§ 7.1.3	显示页面管理	(171)
§ 7.1.4	读、写显示字符及属性	(172)
§ 7.1.5	控制屏幕或窗口的滚动	(177)
§ 7.2	面对显示适配器的编程	(179)
§ 7.2.1	确定显示卡的种类	(179)
§ 7.2.2	视频缓冲区	(180)
§ 7.2.2.1	往屏幕映象区写字符	(181)

§7.2.2.2 快速显示字符串.....	(183)
§7.2.2.3 显示第26行.....	(185)
§7.2.2.4 保留和恢复文本屏幕显示信息.....	(188)
第八章 文本窗口	(191)
§8.1 预定义参数、类型及全局变量说明.....	(213)
§8.2 面向窗口的基本的显示输出程序.....	(214)
§8.2.1 光标定位.....	(214)
§8.2.2 显示属性控制.....	(216)
§8.2.3 显示输出.....	(219)
§8.3 窗口操作.....	(222)
§8.3.1 窗口的设置与建立.....	(222)
§8.3.2 窗口的移动与撤除.....	(226)
§8.3.3 窗口的存贮与恢复.....	(227)
§8.4 窗口系统初始化.....	(229)
§8.5 文本窗口工具包的使用.....	(229)
第九章 程序菜单	(238)
§9.1 程序菜单的设计思想.....	(238)
§9.1.1 菜单的评价标准.....	(238)
§9.1.2 菜单过程的一般操作.....	(238)
§9.1.3 菜单的编程要求.....	(239)
§9.1.4 菜单文件.....	(239)
§9.2 简单的文本窗口操作.....	(240)
§9.2.1 存贮当前屏幕内容并创建窗口.....	(240)
§9.2.2 恢复存贮的屏幕内容.....	(241)
§9.3 菜单过程.....	(243)
§9.3.1 在屏幕中央显示菜单.....	(243)
§9.3.2 在屏幕上任何地方显示菜单.....	(253)
§9.3.3 在程序中使用菜单过程.....	(253)
第十章 图形窗口	(257)
§10.1 环境参数、类型及全局变量说明	(284)
§10.1.1 环境参量	(284)
§10.1.2 自定义数据类型	(285)
§10.1.3 全局变量	(285)
§10.2 辅助操作过程	(286)
§10.3 图形屏幕操作	(288)
§10.4 界域、窗口的设立	(293)
§10.5 边界剪切	(297)
§10.6 基本的图形操作	(299)
§10.7 显示不同比例的字符和文本	(305)

§ 10.8	窗口操作	(309)
§ 10.9	初始化	(316)
§ 10.10	图形窗口工具包的应用	(316)
§ 10.11	应用举例	(317)
第十一章	特长字符串处理	(325)
§ 11.1	定义特长字符串	(325)
§ 11.2	特长字符串的运算操作	(325)
§ 11.2.1	初始化特长字符串	(326)
§ 11.2.2	连接特长字符串	(326)
§ 11.2.3	插入特长字符串	(326)
§ 11.2.4	删除特长字符串	(327)
§ 11.2.5	匹配特长字符串	(327)
§ 11.2.6	检查特长字符串长度	(328)
§ 11.2.7	复制特长字符串	(328)
§ 11.3	带缓冲的字符串输入	(329)
§ 11.3.1	光标控制过程	(329)
§ 11.3.2	输入字符的控制	(332)
§ 11.3.3	字符串的快速显示	(334)
§ 11.3.4	移动输入的字符串	(336)
第十二章	扩充内存和扩展内存的使用	(343)
§ 12.1	使用扩充内存	(344)
§ 12.1.1	保护方式的寻址方法	(344)
§ 12.1.2	检查扩充内存量(对AT机)	(348)
§ 12.1.3	向扩充内存存取数据	(348)
§ 12.1.4	扩充内存应用举例	(350)
§ 12.2	使用扩展内存	(357)
§ 12.2.1	EMS规范介绍	(358)
§ 12.2.2	确认是否安装驱动程序和内存扩展卡	(360)
§ 12.2.3	扩展内存页面检查	(363)
§ 12.2.4	检查EMM版本号	(364)
§ 12.2.5	取得页框地址	(365)
§ 12.2.6	扩展内存页面分配	(366)
§ 12.2.7	页面映射	(368)
§ 12.2.8	扩展内存页面的释放	(369)
§ 12.2.9	保留映射状态	(370)
§ 12.2.10	恢复映射状态	(371)
§ 12.2.11	内存驻留程序使用扩展内存	(372)
§ 12.2.12	扩展内存应用举例	(373)
第十三章	特殊字符显示与屏幕效果	(379)

§13.1 产生放大字符	(379)
§13.1.1 字符图形表	(379)
§13.1.2 显示放大字符	(379)
§13.1.3 打印放大字符	(381)
§13.2 显示活动的字符	(384)
§13.2.1 显示向右滑动的字符	(385)
§13.2.2 显示旋转字符	(386)
§13.2.3 显示相互“碰撞”的字符	(387)
§13.2.4 放大字符的滑动	(389)
§13.2.5 在程序中使用显示活动字符过程	(391)
§13.3 反相显示的和闪烁的字符	(392)
§13.3.1 反相显示字符	(392)
§13.3.2 显示闪烁的字符	(392)
§13.4 控制光标的大小	(392)
§13.4.1 光标显示的参数	(393)
§13.4.2 设置光标大小	(393)
§13.5 显示棒图	(395)
§13.5.1 一个简单的棒图	(395)
§13.5.2 使用BarGraph过程	(398)
附录一: Turbo pascal V3.0、V4.0快速查询要览	(401)
附录二: 图形软件包使用说明	(422)
附录三: 计算机显示系统综述	(439)

第一章 系统功能调用

系统功能指的是DOS提供的一组设计精良的服务程序。这些程序能够有效地处理磁盘读写操作、内存管理、基本输入输出管理和日期、时间的管理等等。在应用程序编程中，由于有了这些可以直接调用的DOS服务程序，使得程序员能够从复杂的低级编程中解脱出来，从而大大地提高了编程的效率。

习惯上，人们总以为只有汇编语言才能调用系统功能，实际则不然，Turbo Pascal也具备调用系统功能的能力。在这一章中，我们首先向读者介绍DOS功能构成，然后通过给出一组调用系统功能的工具过程来说明具体编程的方法。

§1.1 系统功能介绍

DOS服务程序是磁盘操作系统的一个非常重要的组成部分。这部分程序是在系统初始化时从文件IBMDOS.COM中装入内存的。不同的DOS版本提供的系统功能是有差异的，但它们之间存在自下向上的兼容关系。早期DOS版本的功能调用一般都能被后来的版本所支持，但反过来则不一定成立。

DOS 1.xx支持的功能称为通用DOS功能，它是DOS功能中最基本的部分；DOS的2.xx扩充了一些非常有用的功能，比如支持子目录管理等；DOS 3.xx也作了进一步功能扩充，但扩充的功能主要是与网络有关的。

表 1.1 DOS 功 能 列 表

功能调用		功能描述	对应DOS版本
十进制	十六进制		
0	00h	程序中止	1. x x
1	01h	键盘输入	1. x x
2	02h	显示输入	1. x x
3	03h	辅助输入	1. x x
4	04h	辅助输出	1. x x
5	05h	打印机输出	1. x x
6	06h	直接控制台I/O	1. x x
7	07h	无回送直接控制台输入	1. x x
8	08h	无回送控制台输入	1. x x
9	09h	打印字符串	1. x x
10	0Ah	带缓冲的键盘输入	1. x x
11	0Bh	检测标准输入状态	1. x x
12	0Ch	清除标准键盘缓冲区并调键盘功能	1. x x
13	0Dh	磁盘复位	1. x x

续表 1.1

功能调用		功能描述	对应DOS版本
十进制	十六进制		
14	0Eh	选择磁盘驱动器	1. x x
15	0Fh	用FCB打开文件	1. x x
16	10h	用FCB关闭文件	1. x x
17	11h	寻找第一个匹配项	1. x x
18	12h	寻找下一个匹配项	1. x x
19	13h	删除文件	1. x x
20	14h	顺序读	1. x x
21	15h	顺序写	1. x x
22	16h	生成文件	1. x x
23	17h	文件更名	1. x x
24	18h	DOS保留	1. x x
25	19h	返回当前磁盘驱动器	1. x x
26	1Ah	设置磁盘传送地址	1. x x
27	1Bh	获得当前FAT信息	1. x x
28	1Ch	获得指定FAT信息	1. x x
29	1Dh	DOS保留	1. x x
30	1Eh	DOS保留	1. x x
31	1Fh	DOS保留	1. x x
32	20h	DOS保留	1. x x
33	21h	随机读	1. x x
34	22h	随机写	1. x x
35	23h	返回指定文件大小	1. x x
36	24h	设置随机记录号	1. x x
37	25h	放置中断向量	1. x x
38	26h	生成新的程序段	1. x x
39	27h	随机块读	1. x x
40	28h	随机块写	1. x x
41	29h	解释文件名	1. x x
42	2Ah	获得日期	1. x x
43	2Bh	设置日期	1. x x
44	2Ch	获得时间	1. x x
45	2Dh	放置时间	1. x x
46	2Eh	设置/清除磁盘写校验	1. x x
47	2Fh	获得磁盘传送地址	2. x x
48	30h	获得DOS版本号	2. x x
49	31h	终止进程并驻留内存	2. x x
50	32h	DOS保留	2. x x
51	33h	设置/获得Ctrl-Break状态信息	2. x x

续表 1.1

功能调用		功能描述	对应DOS版本
十进制	十六进制		
52	34h	DOS保留	2. x x
53	35h	获得中断向量	2. x x
54	36h	获得磁盘自由空间	2. x x
55	37h	DOS保留	2. x x
56	38h	获取/设置有关国别的信息	2. x x
57	39h	生成子目录 (MKDIR)	2. x x
58	3Ah	删除子目录 (RMDIR)	2. x x
59	3Bh	改变当前子目录 (CHDIR)	2. x x
60	3Ch	生成文件	2. x x
61	3Dh	打开文件	2. x x
62	3Eh	关闭文件	2. x x
63	3Fh	从文件或外设读取	2. x x
64	40h	写到文件或外设	2. x x
65	41h	删除文件	2. x x
66	42h	移文件指针	2. x x
67	43h	获取/设置文件属性	2. x x
68	44h	设备的I/O控制	2. x x
69	45h	复制文件编号	2. x x
70	46h	强制复制文件编号	2. x x
71	47h	获得当前目录	2. x x
72	48h	分配内存	2. x x
73	49h	释放分配的内存	2. x x
74	4Ah	修改已分配内存块	2. x x
75	4Bh	装入或执行程序	2. x x
76	4Ch	终止进程	2. x x
77	4Dh	获得子进程返回代码	2. x x
78	4Eh	寻找第一个匹配文件	2. x x
79	4Fh	寻找下一个匹配文件	2. x x
80	50h	DOS保留	2. x x
81	51h	DOS保留	2. x x
82	52h	DOS保留	2. x x
83	53h	DOS保留	2. x x
84	54h	获得校验状态	2. x x
85	55h	DOS保留	2. x x
86	56h	文件更名	2. x x
87	57h	获取/设置文件时间和日期	2. x x
88	58h	DOS内部使用	2. x x
89	59h	获得扩展错误代码	3. x x

续表 1.1

功能调用		功能描述	对应DOS版本
十进制	十六进制		
90	5Ah	生成临时文件	3. x x
91	5Bh	生成新文件	3. x x
92	5Ch	加锁/解锁文件存取	3. x x
93	5Dh	DOS保留	3. x x
94	5E00h	获得机器名字	3. x x
	5E02h	置打印机设置	3. x x
	5E03h	取打印机设置	3. x x
95	5F02h	取重定向列表项	3. x x
	5F03h	重定向设备	3. x x
	5F04h	中止重定向	3. x x
96	60h	保留	3. x x
97	61h	保留	3. x x
98	62h	取程序段前缀地址	3. x x

注：表1.1中的“对应DOS版本”一栏所列的版本号为最初引入某项功能的DOS版本号。

§1.2 在Turbo Pascal中调用系统功能

前面我们已经简要地介绍了系统功能的构成，这一节将要讲解如何在 Turbo Pascal 程序中调用这些功能。

系统功能是通过DOS中断Int 21H调用的。其调用格式如下：

AH= 功能号

AL= ...

:

Int 21H

Turbo Pascal中有两个标准预定义过程用于系统功能调用，它们是：Intr和MsDos。Intr是为所有软中断调用提供的，而MsDos则专门用于系统功能调用。因此，MsDos可以由Intr所代替。它们的定义格式如下：

Procedure Intr (IntrNo:byte; Var Regs:Registers) ;

Procedure MsDos (Var Regs:Registers) ;

其中，IntrNo为中断号（可以用10进制或16进制表示）。

Regs为一种寄存器组类型 (Registers) 变量

在Turbo Pascal V4.0的DOS单元中，定义了这种寄存器组类型Registers。对于使用较低版本的用户，则要在程序中自定义这一类型，定义说明如下：

Type

Registers= record

AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags:integer;

end;

或者:

Type

Registers = record Case Integer of

0: (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags: integer);

1: (AL, AH, BL, BH, CL, CH, DL, DH : byte);

end;

用过程Intr替代过程MsDos的方法是:

Intr (\$21, Regs);

在Turbo Pascal中调用DOS功能之前, 必须把有关参数赋给寄存器组记录的 有关字段(各字段的名称是与实际的寄存器名相对应的)。待服务程序执行完毕, 这些字段将包含着各种返回的参数值。

这里有两点需要说明:

- ①某些DOS功能需要的输入信息是一种ASCIIZ字符串(如DOS需要的文件名便是这种类型的字符串)。它与Turbo Pascal中的标准字符串类型略有不同。虽然这两种字符串都是由ASCII字符组成的, 但在Turbo Pascal字符串的头部留有一个字节指明当前字符串的长度; 而ASCIIZ字符串的头一个字节开始便是实际的字符串内容, 但在ASCIIZ字符串的尾部有一个零字节作为结束标志, 即字符串的内容直到遇到一个零字节(NULL字符)为止。

读者可以把ASCIIZ字符串看成是一个由字符构成的数组, 它的最后一个字符元素是零字节, 即chr(0), 或者把它看成是一个尾部加有零字节的Turbo Pascal字符串(如String0:=String+#0)。

对于由字符数组构成的ASCIIZ字符串, 可用如下语句获得其首地址:

DataSegment:=Seg(ByteArray[FirstElement]);

Offset:=Ofs(ByteArray[FirstElement]);

其中, FirstElement表示第一个数组元素的下标

对于由Turbo Pascal字符串尾部加零字节构成的ASCIIZ字符串, 可用如下语句获得其首地址:

DataSegment:=Seg(String0[1]);

Offset:=Ofs(String0[1]);

在取地址时跳过了记录字符串长度的第0字节。

- ②对于使用Turbo Pascal V4.0的用户, 注意在调用过程Intr和MsDos之前作如下说明:

Uses DOS;

因为过程Intr和MsDos是在Dos单元中定义的。

在DOS单元中还定义了几个以调用DOS功能实现的标准过程, 如用于处理系统日期与时间的过程GetDate, SetDate, GetTime和SetTime; 用于磁盘和文件管理的函数及过程DiskFree, DiskSize, GetFAttr, SetFAttr, GetFTime, SetFTime, FindFirst, FindNext等等。这些过程在本书中将要陆续介绍, 读者

也可以直接参阅Turbo Pascal V4.0的使用手册。

§1.3 DOS功能调用实例

在这一节中，我们将给出9个工具过程，它们都是通过调用DOS功能实现的。这些过程提供这样的一些服务：获取DOS版本号，报告当前驱动器和当前目录，获取扩展错误代码，获取和设置校验状态，获取磁盘上的自由空间大小，获取和设置文件属性，获取和设置系统日期与时间等等。每个过程都附有调用说明以及编程说明，并且还附有相应的演示程序。

§1.3.1 获取DOS版本号

Tool 1.1给出过程GetVersion，它调用DOS功能48 (\$30) 检查当前使用的DOS版本号。这个过程对于那些要求在特定DOS版本下执行的程序非常有用。如果读者恰巧有一个程序要在特定的DOS版本下运行，最好把这个过程放在该程序的前面。如果它发现使用了不适当的DOS版本，程序就显示提示信息并中止程序。

虽然DOS 1.x.x并不支持功能48，这个过程仍能在其环境下不太精确地工作。在DOS 2.x.x和以后的DOS版本中，版本号通过两个寄存器返回，寄存器AL的值表示主版本号，寄存器AH的值表示次版本号。例如，版本号2.11在过程返回时，AL寄存器中装有2，AH寄存器中装有11。如果过程在DOS 1.x.x下运行，它不会引起任何操作，寄存器返回值是零。因此，只有寄存器返回值为零，方表明处理的是DOS 1.x.x版本，但此时无法精确地知道其次版本号。对于新的OS/2系统，规定返回的主版本号是10。

实际在程序中调用过程GetVersion时，使用一个实型变量作为参数，过程返回时，这个实型参数包含DOS的版本号

下面说明过程GetVersion的工作步骤：

- 设置寄存器AH为功能号48，AL为零。
- 调用MsDos过程。
- 由寄存器AH和AL分别返回主、次版本号。

```
-----
PROCEDURE GetVersion (VAR DOSVersion:Real) ;
{ If this procedure is to be used with Turbo Pascal versions
  prior to 4.0, the following declaration must be included:
  TYPE
    Registers = RECORD CASE Integer OF
      1: (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags: Integer) ;
      2: (AL, AH, BL, BH, CL, CH, DL, DH           , Byte) ;
    END; }
VAR
  Regs           , Registers;
  Major, Minor   , Integer;
BEGIN { get DOS version }
  Regs. AH := 48;
  Regs. AL := 0;
```

```

MsDos (Regs) ;
Major := Regs. AL;
Minor := Regs. AH;
IF Major = 0 THEN DosVersion := 1.0
ELSE
    BEGIN
        DOSVersion := Minor/100;
        DOSVersion := DOSVersion + Major;
    END;
END; { get DOS version }

```

Tool 1.1 过程GetVersion获取DOS版本号

• 把主、次版本号两个整数转换为一个实数。

显然在过程GetVersion中可以用语句Intr(\$21, Regs)代替语句MsDos(Regs)，过程的功能不受影响。

Test Program 1.1演示了怎样在程序中调用过程GetVersion，它简单地显示通过GetVersion过程得到的有关DOS版本信息。

```

PROGRAM TestGetVersion;
{ To use this program with Turbo Pascal versions prior to
  4.0, remove the Uses statement below. }
Uses DOS;
VAR
    VersionNumber: Real;
SI GetVers, Inc}
BEGIN { program }
    GetVersion(VersionNumber) ;
    WriteLn('DOS Version on this computer is: ',
        VersionNumber:4:2) ;
END. { program }

```

Test Program 1.1 在程序中调用过程GetVersion

§ 1.3.2 报告当前驱动器

Tool 1.2给出过程CurrDrive，它调用DOS功能25 (\$19) 报告当前驱动器。这个过程在程序员希望知道计算机当前正在使用哪一个驱动器时非常有用

过程CurrDrive返回单个字符变量，作为确定当前驱动器的依据。当前驱动器的代号是：0 = A，1 = B，2 = C，等等。

下面说明过程CurrDrive的工作步骤：

- 设置寄存器AH为功能号25。
- 设置寄存器AL为零。
- 调用MsDos过程。