

微机程序员问题解答

王 帆 编写

燕卫华 校

学苑出版社

计算机实用技术系列丛书(二)

微机程序员问题解答

王 帆 编写
燕卫华 审校

学苑出版社

(京)新登字 151 号

内 容 提 要

· 本书是针对解决程序员在工作中遇到的各种问题而编写的。使用本书可大大提高程序员的编程效率,节约大量查找技术资料的时间。

欲购本书的用户,请直接与北京海淀 8721 信箱书刊部联系,电话:2562329,邮
码:100080。

计算机实用技术系列丛书(二)

微机程序员问题解答

编 写:王 帆
审 校:燕卫华
责任编辑:甄国宪
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:施园印刷厂
开 本:787×1092 1/16
印 张:24.375 字 数:569 千字
印 数:1~1000 册
版 次:1993 年 12 月北京第 1 版第 1 次
ISBN7-5077-0760-1/TP·7
本册定价:17.00 元

学苑版图书印、装错误可随时退换

前 言

随着我国现代化建设的逐步开展,微型计算机迅速在我国得到了普及。我们现在面临的问题是如何充分利用现有的机器资源来充分发挥微型计算机的作用,提高我们的应用水平。

有关 IBM PC/XT/AT 及兼容机的技术细节有许多,如硬件技术、软件技术以及程序设计语言等。即使是一个受过正规训练的计算机技术人员,在进行应用开发时也需要不时地查阅各种技术资料,以查找对各种问题的解释和说明。这样一来,就大大降低了程序员的工作效率。这本《IBM PC/XT/AT 及兼容机程序员问题解答》就是针对解决程序员在工作中遇到的各种问题而编写的。使用本书,可大大提高程序员的编程效率,省去许多查阅技术资料的时间。

本书在编排上有两个独有的特点:其一就是按硬件种类分章,再按硬件性能分节。在每一小节中,再按条目进行具体的讨论,每个条目讨论一个特定的编程任务。其二就是每项内容均包括四个部分。开头介绍一些有关的基础知识,然后对所提出的问题按高级语言编程、中级的 BIOS 及 DOS 中断编程和初级的微处理器芯片编程三种方式进行讨论。书中给出的例子都是精心挑选的和经得起考验的,它们体现了许多值得学习的编程技巧。

书中除了某些较为高深的专业内容外,几乎包括了所有 IBM 微机资料中的技术细节。此外,在本书中,对同一问题我们还给出了不同的解决方法,并对其优劣进行了比较。还有,书中向读者提供了 ASCII 代码、指令执行时间等各种常用表格。因此,本书基本上能满足全部编程工作的需要。

我们希望本书能为广大的计算机工作者,尤其是微型机开发领域内的技术人员提供一个实用的工具,能够为发展我国的微型机应用做出一点贡献。由于我们的水平有限,书中的错误在所难免,敬请广大同仁批评指正。

一九九一年五月

于北京

23

目 录

第一章 系统资源	(1)
§ 1.1 确定系统资源的状况	(1)
§ 1.1.1 访问 8255 外设接口	(2)
§ 1.1.2 确定 IBM 微机的型号	(5)
§ 1.1.3 确定 PC-DOS 版本	(6)
§ 1.1.4 确定图形适配卡的数目及类型	(7)
§ 1.1.5 找出磁盘驱动器的数目及种类	(9)
§ 1.1.6 确定外部设备数目及类型	(10)
§ 1.1.7 确定 RAM 大小	(12)
§ 1.2 中断管理	(15)
§ 1.2.1 8259 中断控制器编程	(16)
§ 1.2.2 允许 / 禁止特定的硬件中断	(16)
§ 1.2.3 编制自用中断	(17)
§ 1.2.4 完善现有的中断	(20)
§ 1.3 程序管理	(21)
§ 1.3.1 存储器的分配与撤消	(22)
§ 1.3.2 运行嵌套程序	(25)
§ 1.3.3 程序中使用 DOS 用户接口指令	(27)
§ 1.3.4 程序执行完后常驻内存	(28)
§ 1.3.6 调入并运行程序覆盖	(31)
§ 1.3.6 将程序由 .EXE 型转换为 .COM 型	(34)
第二章 定时器与发声器	(37)
§ 2.1 定时器的置数和读数	(37)
§ 2.1.1 8253 / 8254 定时器编程	(37)
§ 2.1.2 时间的设置与读取	(40)
§ 2.1.3 日期的设置与读取	(42)
§ 2.1.4 实时时钟的设置与读取	(43)
§ 2.1.5 延时操作编程	(45)
§ 2.1.6 定时操作编程	(46)
§ 2.1.7 实时控制操作编程	(48)
§ 2.1.8 用定时器芯片产生随机数	(51)
§ 2.2 发声	(53)
§ 2.2.1 76496 发声器的编程方法 (仅对 PCjr)	(53)
§ 2.2.2 演奏	(55)
§ 2.2.3 与其它操作同时进行的演奏	(57)

§ 2.2.4	报警	(58)
§ 2.2.5	演奏音符串	(60)
§ 2.2.6	在进行其它操作的同时演奏音符串	(63)
§ 2.2.7	产生滑音音调	(67)
§ 2.2.8	产生音响效果	(68)
§ 2.2.9	同时发声	(70)
第三章	键盘	(72)
§ 3.1	键盘监控	(72)
§ 3.1.1	清除键盘缓冲区	(73)
§ 3.1.2	检查缓冲区中的键入字符	(75)
§ 3.1.3	等待键入字符但不在民间上显示	(76)
§ 3.1.4	等待键入字符并把它送向屏幕	(78)
§ 3.1.5	立即获取字符	(80)
§ 3.1.6	获取键入字符串	(81)
§ 3.1.7	检查/设置双态键和组合键的状态	(83)
§ 3.1.8	编写通用的键盘输入例程	(84)
§ 3.1.9	重新编制键盘中断	(88)
§ 3.2	存取特殊键	(92)
§ 3.2.1	Backspace, Enter, Escape 和 Tab 键的使用	(92)
§ 3.2.2	组合键的使用: Shift 键 Ctrl 键和 Alt 键	(92)
§ 3.2.3	双态键的使用: Numlock, Capslock, Ins 和 Scrollock	(93)
§ 3.2.4	数字副键盘和光标键的使用	(94)
§ 3.2.5	功能键的使用	(95)
§ 3.2.6	重新编制单个键	(96)
§ 3.2.7	对单个键定义键盘宏功能	(98)
§ 3.2.8	建立 Ctrl-Break 例程	(99)
§ 3.2.9	PrtSc 键的重新定义	(100)
§ 3.3	各种键盘编码及应用	(101)
§ 3.3.1	各种键的使用	(102)
§ 3.3.2	扫描码	(102)
§ 3.3.3	ASCII 码	(103)
§ 3.3.4	框图编码	(103)
§ 3.3.5	扩充码	(103)
第四章	视频显示	(105)
§ 4.1	视频显示控制	(105)
§ 4.1.1	6845 视频控制器编程	(106)
§ 4.1.2	屏幕显示模式的设置与检验	(108)

§ 4.1.3	设置字符的属性和颜色	(113)
§ 4.1.4	设置屏幕边界的颜色	(120)
§ 4.1.5	清除全部或部分屏幕内容	(121)
§ 4.1.6	视频适配器之间的转换	(123)
§ 4.2	光标控制	(125)
§ 4.2.1	在绝对位置处设置光标	(126)
§ 4.2.2	设置光标于相关位置	(128)
§ 4.2.3	光标的打开与关闭	(130)
§ 4.2.4	改变光标的形状	(131)
§ 4.2.5	光标位置的读取、存储及恢复	(132)
§ 4.2.6	产生交替光标类型	(134)
§ 4.3	屏幕字符显示	(135)
§ 4.3.1	在屏幕上“写”单个字符	(135)
§ 4.3.2	在屏幕上“写”一串字符	(142)
§ 4.3.3	读给定位的字符及属性	(144)
§ 4.3.4	建立特殊字符	(145)
§ 4.4	描绘点阵图形	(149)
§ 4.4.1	设置点阵图形的彩色	(150)
§ 4.4.2	在屏幕上描绘点阵(对于单色卡、彩色卡、PCjr)	(154)
§ 4.4.3	在屏幕上描绘点阵(对 EGA)	(158)
§ 4.4.4	确定屏幕一点的颜色	(165)
§ 4.4.5	在屏幕上画线	(169)
§ 4.4.6	屏幕区域填充	(173)
§ 4.4.7	用块字符描绘图形	(177)
§ 4.5	应用卷轴及分页技术	(179)
§ 4.5.1	文本屏幕垂直卷轴	(179)
§ 4.5.2	文本屏幕水平卷轴	(181)
§ 4.5.3	切换文本页	(182)
§ 4.5.4	文本页间卷轴	(187)
第五章	磁盘驱动器	(189)
§ 5.1	监视磁盘分配	(189)
§ 5.1.1	读取文件分配表	(190)
§ 5.1.2	确定可用的磁盘空间	(193)
§ 5.1.3	文件长度的获取与设置	(194)
§ 5.1.4	校正盘空间不足的错误	(196)
§ 5.2	磁盘目录操作	(197)
§ 5.2.1	根目录的读取与更改	(198)
§ 5.2.2	子目录的建立与删除	(201)

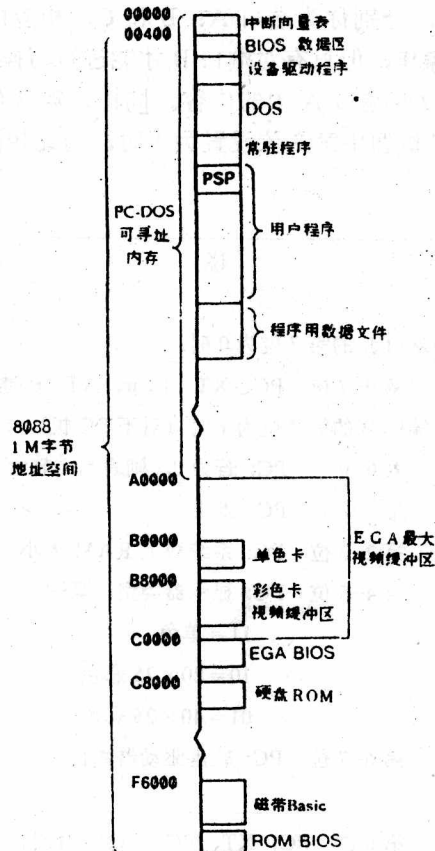
§ 5.2.3	子目录的读取与更改	(203)
§ 5.2.4	当前目录的获取与设置	(205)
§ 5.2.5	获取 / 设置文件的时间和日期	(206)
§ 5.2.6	写保护或隐藏文件	(207)
§ 5.2.7	文卷标号的读取与改写	(209)
§ 5.3	准备进行文件操作	(212)
§ 5.3.1	缺省驱动器的设置与检查	(213)
§ 5.3.2	文件的建立与删除	(214)
§ 5.3.3	文件的打开与关闭	(217)
§ 5.3.4	重新命名文件 / 移动文件的目录位置	(222)
§ 5.3.5	准备进行文件操作	(223)
§ 5.3.6	分析来自命令行的信息	(228)
§ 5.4	读写文件	(229)
§ 5.4.1	对 765 软盘控制器和 8237DMA 芯片进行编程	(230)
§ 5.4.2	特定扇区的读与写	(239)
§ 5.4.3	写入顺序文件	(242)
§ 5.4.4	读顺序文件	(248)
§ 5.4.5	写入随机文件	(253)
§ 5.4.6	读随机文件	(257)
§ 5.4.7	在读、写操作之后对数据进行验证	(261)
§ 5.4.8	磁盘错误的确定与排除	(261)
第六章	打印机	(265)
§ 6.1	控制打印机操作	(265)
§ 6.1.1	初始化打印口和重新初始化打印机	(266)
§ 6.1.2	测试打印机是否联机	(267)
§ 6.1.3	解释和排除打印机的错误	(269)
§ 6.1.4	两台或多台打印机的切换	(272)
§ 6.2	设置打印参数	(273)
§ 6.2.1	设定文本和图形打印方式	(274)
§ 6.2.2	行距的控制	(276)
§ 6.2.3	走纸的控制	(277)
§ 6.2.4	控制打印头的位置	(278)
§ 6.2.5	设置制表符位置	(278)
§ 6.2.6	改变打印字形	(279)
§ 6.2.7	IBM 各种打印机性能的比较	(280)
§ 6.3	将数据发送至打印机	(284)
§ 6.3.1	输出文本或图形数据	(285)
§ 6.3.2	文本的右对齐	(288)

§ 6.3.3 均衡间隙的文本打印	(291)
§ 6.3.4 打印特殊字符	(293)
§ 6.3.5 屏幕拷贝	(296)
第七章 输入/输出	(300)
§ 7.1 访问串行端口	(300)
§ 7.1.1 8250UART 芯片编程	(301)
§ 7.1.2 串行口初始化	(302)
§ 7.1.3 设置当前通信端口	(306)
§ 7.1.4 监视串行端口状态	(307)
§ 7.1.5 调制解调器的初始化和监视	(308)
§ 7.1.6 数据发送	(312)
§ 7.1.7 接收数据	(314)
§ 7.1.8 用通信中断发送/接收数据	(317)
§ 7.1.9 查找通信控制码	(320)
§ 7.2 建立设备驱动程序	(321)
§ 7.2.1 建立设备首标	(322)
§ 7.2.2 建立设备策略	(323)
§ 7.2.3 建立设备中断句柄	(324)
§ 7.2.4 访问设备驱动程序	(328)
§ 7.2.5 设备错误的检查与分析	(330)
§ 7.3 使用串行 I/O 设备	(333)
§ 7.3.1 读/写磁带录音机	(334)
§ 7.3.2 读取光笔位置	(334)
§ 7.3.3 从游戏端口读取模拟输入值	(337)
§ 7.3.4 从游戏端口取数字输入值	(340)
附录 A 二进制、十六进制及内存寻址	(343)
附录 B BASIC 语言的位操作	(347)
附录 C 汇编语言基础知识	(352)
附录 D 将汇编例程组合成 BASIC 程序	(356)
附录 E 使用 ANSI.SYS 设备驱动程序	(358)
附录 F 8088 指令系统	(359)
附录 G 80286 指令系统	(366)
附录 H IBM 微机词汇表	(371)

第一章 系统资源

§ 1.1 确定系统资源的状况

在调入程序时，应当首先了解程序所处的环境，如 IBM 微型计算机的型号、DOS 版本号、内存容量以及外部设备是否齐全等等。一般来讲，可以通过三种途径来了解系统资源的环境情况。一种办法是让用户自己去了解，但未必能得到答案，这并不是一个好办法。另一种方法是尽量从系统板上的 DIP 开关的设置来了解系统环境情况，但有时，这些设置并不充分，因此，这也不是实用的好方法。再一种办法就是直接对有疑问的硬件进行访问或查找 BIOS 数据区，这是一个非常实用的好方法。由于从 DIP 开关设置入手最利于了解系统资源环境，因此，本节从 8255 外围接口芯片开始讨论，由该芯片可以得知 DIP 开关的设置内容。



注：PSP——Program Segment Prefix 程序段前缀

EGA——Enhanced Graphics Adapter 增强型图形适配器

图 1-1 使用 1M 字节存储地址空间

程序仅能以两种方式对硬件进行访问，即可以读/写任何与硬件相连的端口地址（通

常只是 65535 个可用端口地址中的一小部分), 还可以读 / 写百万余个随机存储器 (RAM 地址空间中的任何地址。在第七章中, 我们较为详细地汇总了端口地址。图 1-1 给出了操作系统及程序所在内存的分布状况。

§ 1.1.1 访问 8255 外设接口

若要了解外部设备的当前状态, 最好可从 Intel 8255 外设接口芯片入手。该芯片用途相当广泛, 通过它可以获得系统板上 DIP 开关的设置情况。它接收键盘给计算机输入的信息, 并控制包括 8253 时钟芯片在内的一系列外部设备。在各种型号的 IBM 微型计算机中, 只有 AT 机不用 8255 芯片。它将本机配置的信息与实时时钟一起存储于由专门电池供电的芯片上。但 IBM AT 计算机仍然将 8255 的端口地址用于键盘操作及时钟芯片控制。

8255 有三个单字寄存器, 分别称为端口 A、B 和 C, 相应的端口地址为 60H-62H。对这三个端口都可以进行读操作, 但只有对端口 B 才能进行写操作。在 PC 机中, 端口 B 的第 7 位设置为 1 时, 可以改变端口 A 中的内容。同样, 第 2 位设置为 1 时, 可决定端口 C 中较低 4 位的内容。XT 机型中第 3 位设置为 1 时, 与此相同。整个寄存器的内容如下:

端口	说 明
端口 A (60H)	当端口 B 的第 7 位为 0 时: 第 0-7 位 PC、XT、PCjr、AT: 从键盘接收八位扫描码。 当端口 B 的第 7 位为 1 时 (对于 PC 机): 第 0 位 PC: 若为 0, 则表示无软盘驱动器。 第 1 位 PC: 未用 第 2-3 位 PC: 系统板上 RAM 大小 第 4-5 位 PC: 显示器类型, 其中: 11 = 单色 10 = 80×25 彩色 01 = 40×25 彩色 第 6-7 位 PC: 软盘驱动器数目
端口 B (61H)	第 0 位 PC、XT、PCjr: 8255 计时器芯片通道 2 的控制门。 第 1 位 PC、XT、PCjr: 输出到扬声器 第 2 位 PC: 选择端口 C 的内容 PCjr: 1 为字符模式 0 为图形模式

端口	说 明
端口 B (61H)	第 3 位 PC、PCjr: 若为 1, 表示关闭磁带机马达 XT: 选择端口 C 的内容 第 4 位 PC、XT: 若为 0, 表示 RAM 工作。 PCjr: 若为 1, 表示禁止蜂鸣器及磁带要马达。 第 5 位 PC、XT: 若为 0, 表示扩展槽出错信号工作。 第 6 位 PC、XT: 若为 1, 表示键盘时钟信号工作。 第 5-6 位 PCjr: 选择声源, 其中: 00=8253 芯片 01=磁带 10=I/O 11=76496 芯片 第 7 位 PC: 选择端口 A 的内容, 键盘应答。 XT: 键盘应答
端口 C (62H)	当 PC 机端口 B 的第 2 位为 1 或 XT 机端口 B 的第 3 位为 1 时: 第 0-3 位 PC: 配置开关 2 的下半部分 (RAM 在扩展槽中) 第 0 位 PCjr: 键盘输入值丢失 第 1 位 XT: 若为 1, 表示安装了运算协处理器。 PCjr: 若为 0, 表示安装了调制解调器卡。 第 2-3 位 XT: 系统板上 RAM 大小 第 3 位 PCjr: 0=128K RAM 第 4 位 PC、PCjr: 由磁带输入 XT: 未用 第 5 位 PC、XT、PCjr: 8253 通道 2 的输出 第 6 位 PC、XT: 若为 1, 表示扩展槽错误检验。 PCjr: 若为 1, 表示键盘数据。 第 7 位 PC、XT: 若为 1, 表示奇偶错误校验 PCjr: 若为 0, 表示已连接键盘电缆 当 PC 端口 B 的第 2 位为 0 及 XT 端口 B 的第 3 位为 0 时: 第 0-3 位 PC: 配置开关 2 的上半部分 (未用) 第 0-1 位 XT: 显示器类型, 其中: 11=单色 10=80×25 彩色 01=40×25 彩色 第 2-3 位 XT: 软盘驱动器数目 第 4-7 位 PC、XT: 与端口 B 的第 2 位为 1 时相同

注: 寄存器的某一位为 0 时, 相当于 DIP 开关设置到“OFF”位置。

AT 机上的设置信息以及实时时钟均采用 Motorola MC146818 芯片。尽管 AT 机采用相同的端口地址控制时间芯片并接收键盘输入数据，但它并没有使用 8255 芯片。MC146818 芯片有 64 个寄存器，依次标为 00-3FH。若要读取一个寄存器，应首先将寄存器号送入 70H 号端口地址，然后从 71H 号端口地址读出寄存器内容。下表给出了各种配置的大概设置情况：

寄存器号	用 途
10H	软盘驱动器类型，硬盘驱动器类型
12	
1	外设
15	系统板存储器（低位字节）
16	系统板存储器（高位字节）
17	全部扩展存储器（低位字节）
18	全部扩展存储器（高位字节）
30	超出 1M 的扩展存储器（低位字节）
31	超出 1M 的扩展存储器（高位字节）

高级编程

本书给出了访问上述端口的几个例子。首先，我们考察一个确定 PC 机安装的软盘驱动器数目的 BASIC 程序。在读取端口 A 的较高两位之前，必须将端口 B 中的第 7 位设置为 1，而在执行前又需将该位恢复为 0，否则的话，键盘会锁死，必须重新启动才行。BASIC 语言不允许出现以二进制表示的数字，因为那样做的结果会使位类型的分析复杂化。可以编制一个简单的子程序把一个大至 255（端口地址的最大值）的整数转化为一个八个字符的二进制字符串。然后，使用诸如 MID\$ 的字符串函数就能取出有关的位进行分析。有关 BASIC 位操作的基本知识，可参阅本书的附录 B。

```

100 A=INP (&H61)           ; 取端口B的数值
110 A=AOR128                 ; 第7位设置为1
120 OUT &H61, A              ; 将该字节返送回端口B
130 B=INP (&H60)             ; 取端口A的数值
140 A=A AND 127              ; 第7位设置为0
150 OUT &H61, A              ; 将原始值再存入端口B
160 GOSUB 1000               ; 将其转化成二进制字符串
170 NUMDISK $ =RIGHT$ (B$, 1) ;取第 0 位
180 IF D$ = "1" THEN NUMDISK =0:GOTO, 230 ;

```

无磁盘系统

```

190 C$ = LEFT$ (B$, 2)      ; 取字符串的较高两位
200 TALLEY = 0               ; 保持磁盘计数
210 IF RIGHT$ (C$, 1) = "1" THEN TALLEY = 2;
                             ; 分离高位字节
220 IF LEFT$ (C$, 1) = "1" THEN TALLEY = TALLEY + 1;
                             ; 加低位字节
230 TALLEY = TALLEY + 1      ; 从1 (而非从0) 开始计数, 得到驱动器数目
1000                         ; 下面的子程序将字节转化成二进制字符串
1010 B$ = " "               ; B$ 是字符串
1020 FOR N = 7 TO 0 STEP -1 ; 不断测试更小的2的幂
1030 Z = B - 2 * N           ; 将其与字节值相减
1040 IF Z >= 0 THEN B = Z: B$ = B$ + "1"
ELSE B$ = B$ + "0"         ; 组成字符串
1050 NEXT                   ; 对每一位重复操作
1060 RETURN                 ; 完成

```

初级编程

用汇编程序确定软盘驱动器数目的方法与上例相同, 但更简便一些。同样, 必须恢复端口 B 的原始值。

```

IN      AL, 61H              ; 取端口B的值
OR      AL, 100000000B       ; 强置第7位为1
OUT     61H, AL              ; 字节代换
IN      AL, 60H              ; 取端口A的值
MOV     CL, 6                 ; 设置AL右移位数
SHR     AL, CL                ; 将高两位右移六位
INC     AL                    ; 从1开始计数
MOV     NUM DRIES, AL         ; 得到驱动器数
IN      AL, 61H              ; 准备恢复端口B之值
AND     AL, 01111111B         ; 第7位置0
OUT     61H, AL              ; 字节代换

```

§ 1.1.2 确定 IBM 微机的型号

各类 IBM 微机 间存在着兼容性问题。若要使一个程序在各类 IBM 微机上运行并充分发挥其作用, 就必须确定运行程序的机器类型。这个信息可从存储器空间的倒数第二个字节获得。在 BIOS ROM 内, 其地址是 FFFFE, 并使用如下代码:

计算机	代码
PC	FF
XT	FE
PCjr	FD
AT	FC

高级编程

使用 BASIC 语言时, 只需用 PEEK 指令读取值:

```

100 DEF SEG = &HF000      ; 指向存储器顶部64K
110 X = PEEK (&HFFFE)     ; 取倒数第二字节
120 IFX = &HFD THEN...    ; 如果是FD则是PCjr

```

初级编程

使用汇编语言确定计算机的类型:

```

MOV  AX, 0F000H           ; 将ES指向ROM
MOV  ES, AX               ;
MOV  AL, ES: [0FFFEH]     ; 取字节
CMP  AL, 0FDH             ; 是PCjr?
JE   INITIALIZE JR       ; 若是, 转初始程序

```

§ 1.1.3 确定 PC-DOS 版本

随着 PC-DOS 版本的更新, 其功能不断增强, 一些程序的编制也更为简便。若要确保软件在任何 DOS 版本支持下都能运行, 应使程序只需 DOS1.0 所提供的功能。DOS 提供了能返回 DOS 版本号的中断功能, 可用来检验软件的兼容性。程序能在刚开始执行时便显示出错误信息, 以提醒用户更换版本。当然, 这种情况并不多见。

中级编程

INT 21H 的功能调用能返回 DOS 的版本号。调用返回时, “主版本号” (2.10 的 2) 在 AL 中, “次版本号” (2.1 的 0) 在 AH 中 (注意: 1 次版本号以 AH 形式给出, 而不是 1H。AL 值可为 0, 指 DOS2.0 以前的版本。这个中断将 0 返回到 BX 和 CX 中, 改变了其原来的值。

; ...确定 DOS 版本号:

```

MOV  AH, 30H              ; 取DOS版本号的功能调用号
INT

```

21H		; 得到功能的用号
CMP	AL, 2	; 检验版本是否为2.X
JL	WRONG_DOS	; 若小于2, 给出信息

§ 1.1.4 确定图形适配卡的数目及类型

应当了解一个程序是在单色卡、彩色图形卡, 还是在 EGA 环境下运行, 是否还有第二个适配卡。第四章中讲述了如何用开关来选择适配卡。位于 BIOS 数据区 0040:0010 处的设备状态字节反映了 DIP 开关 1 的设置情况, 而该设置决定现行的是哪个卡。理想的情况应是: 若用单色卡, 第 5-位将为 11; 若是 80×25 的彩图卡, 则为 10; 若是 40×25 的彩色图形卡, 则为 01; 若是 EGA 卡, 则为 00。但实际上如果 EGA 存在, 第 5-4 位并不为 00, 其具体值由其本身 DIP 开关的不同设置方式决定。因此, 必须首先用其它的方法判定 EGA 是否存在, 若不存在, 则 BIOS 的数据就会指明现行的是单色卡还是彩色卡。若要检验 EGA, 可测试 0040:0087 处的字节。若为 0, 则没有 EGA 卡; 若非零, 当第 3 位为 0 时, EGA 是现行适配卡; 若为 1 时, 现行卡是另一个适配卡。

当存在 EGA 时, 可向 6845 芯片 (见 4.1.1) 的光标地址寄存器写入并读回一个数值以检验其是否相符。此法可判定是单色还是彩色卡。对于单色卡, 将 0FH 关入 3B4H 端口以指示光标寄存器, 然后由 3B5H 端口读/写光标地址。彩色卡上的相应端口分别为 3D4H 及 3D5H。当不存在适配卡时, 端口地址返回值为 0FFH; 但由于此值也可能是寄存器原有的, 所以只检验此值是不够的。

还有两个有关 EGA 的问题需要解释: 现有多少存储容量? 需连接哪种显示器? 要确定显示器类型, 可测试 0040:0087 处的第 1 位。当其为 1 时, 接的是单色显示器; 为 0 时, 接的则是彩色显示器。若程序用的是 350 线彩色模式, 则需要判断彩色显示器是 IRGB 型还 RGB 型, 而后者对应于增强型 IBM 彩色显示器。这些情况可由 EGA 本身的四个 DIP 开关的设置情况得知。当调用 INT 10H 的功能 12H 时, 这些设置信息就返回到 CL 中。若低 4 位为 0110, 则为增强型彩色显示器。此功能还给出了 EGA 的存储容量。若 BL 返回值为 0, 则有 64K 存储量; 为 1 则有 128K; 2 表示有 192K; 为 3 代表全部有内存 256K。

高级编程

下列程序段检验当前显示器的类型及模式, 并判定了主机使用的是何种显示适配器。

```

100''' 确定现行适配器:
110 DEF SEG = &H40          ; 指向BIOS数据区
120 X = PEEK (7H87)          ; 检验是否为EGA
130 IF X = 0 THEN 200        ; 若无EGA, 前跳
140 IF X AND 8 = 0 THEN...   ; EGA取得控制权

```

```

200 X=PEEK (&H10) ; 取设备状态字节
210 Y=X AND 48 ; 得到第4位及第5位的结合值
220 IF Y=48 THEN... ; 是单显 (00110000)
230 IF Y=32 THEN... ; 为80×25图形模式 (00100000)
240 IF Y=16 THEN... ; 为40×25图形模式 (00010000)

```

以下是当 EGA 或彩色卡为现行卡是检验单色卡的例子。若用端口地址 H3D4 和 H3D5, 该程序也可用于检验彩色卡。

```

100''' 确定单色卡是否存在:
110 OUT&H3B4, &HF ; 确立光标寄存器地址
120 X=INP (&H3B5) ; 读该地址并存值
130 OUT &H3B5, 100 ; 将判决值送入寄存器
140 IF INP (&H3B5) <> 100 THEN... ; 若值相同则单色卡存在
150 OUT&H3B5, X ; 若单色卡存在则恢复原值

```

初级编程

下面的例子与上述 BASIC 程序完成的功能相同。

```

; ... 确定现行适配器:
MOV AX, 40H ; 将ES指向BIOS数据区
MOV ES, AX;
MOV AL, ES:[87H] ; 检验EGA是否存在
CMP AL, 0;
JE NO_EGA ; 若0040: 0087是0, 没有EGA
TEST AL, 00001000B ; EGA存在, 测试第3位
JNZ EGA_NOT_ACTIVE ; 若在第3位为1, EGA不是现行的
EGA_NOT_ACTIVE: MOV AL, ES[10H] ; 取视频状态字节
AND AL, 00110000B ; 隔离第4位及第5位
CMP AL, 48 ; 是单色卡吗?
JE MONOCHROME ; 若是, 跳转; 否则为彩色卡

```

假定为单色卡, 确定是否装有 (非现行的) 彩色卡。

```

; ... 已安装彩色卡 (非现行) 否?
MOV DX, 3D4H ; 指向6845地址寄存器
MOV AL, 0FH ; 请求光标寄存器
OUT DX, AL ; 索引寄存器
INC DX ; 指向数据寄存器

```