IBM-PC

高级软件开发资料和实例

尹彦芝

北京信通集团软件工程公司

一九九〇年

前　言

IBM PC机问世已有6年了，已经得到相当的普及。为了使PC机满足各种各样的应用要求，为了更充分地发挥PC机的性能，需要对PC机做一些高层次的开发工作。这种高层次的开发工作需要对机器的硬件和操作系统有更深入的了解，需要对汇编语言这样一种主要的软件开发语言有娴熟的掌握。而要做到这一点，光凭一般的手册是不够的，需要有能对系统作较深入透切分析的资料，需要有比较完整的开发成功的程序实例作参考。撰写本书的主要目的就在于此。另一方面，既然是作为开发成功的程序实例，所以本书所提供的程序大部分是一些有实际应用价值的完整程序，供用户选用，这是撰写本书的另一目的。

本书的各篇文章是从各个不同的角度对微机进行分析，彼此之间没有太多的联系，读者可以按其需要去选取。因为本书是面向高级软件开发人员，所以对一般资料上已介绍过的问题，对一些基本知识和技巧，本书不再叙述。

大体上讲，本书的内容可分为三部分。

第一部分主要是概括系统主要的内存单元和口地址的意义和用法，从几个方面对DOS进行比较深入的分析，如DOS的环境字符串，程序常驻内存的办法和问题，从汇编语言和C语言程序内执行一条DOS命令等。分析过程中不乏实例，有的还涉及到一些未公布的DOS内部秘密。

第二部分主要是介绍软件开发员经常碰到的显示处理问题。介绍了显示系统的发展和现状，主要篇幅放在开发加强图形适配器EGA上。因为这种适配器现在用得比较多，结构也比较独特，但介绍这种适配器的中文资料却很少。对于想了解显示系统（不仅仅是EGA）工作详情的用户来说，读一下这篇文章会很有益处的。

第三部分主要是收集了一批程序实例，其中“菜单制作程序”和“画图程序”是两个比较大型的汇编语言源程序实例。“菜单制作程序”可用来把任何一个应用程序变成一个以菜单形式进行操作的程序。“画图程序”使用户不用编程序，而只需要按规定输入他所需要的图形数据，就可以在屏幕上把图画出来并保存到盘上。为了便于读者理解，程序实例中作了尽可能详尽的注解，是软件开发员难得的参考资料。

由于作者水平有限，某些分析和叙述可能不正确，诚恳欢迎广大读者批评指正。

作者

1988年3月

QTML—多语种处理系统

本系统是清华大学软件开发中心和中国科学院信通集团软件工程公司研制开发的新型多国语言文字处理系统，可在IBM系列微机以及各种兼容机上运行。

采用两种工作方式：DOS方式和多语种混合方式。DOS方式下，保留了PC-DOS的几乎全部功能；多语种方式下，可同时处理中、日、英、法、德、意、葡、俄和希腊等九国文字及几百种图形符号。

实现了操作系统一级支持中日文汉字的处理，配有一、二级常用中文简体、繁体汉字两种版本，并具有几百种图形符号，符合GB2312标准；支持日本JIS一级全部汉字2965个，二级汉字1353个。

中文汉字可在拼音，词组、五笔，电报，区位和国标状态下输入，并可根据用户的需要增改输入方案（例如：仓颉、注音等）。

定义了假名键盘，全部日文假名字母都能直接输入；日文汉字输入有多种方法：用假名输入、用罗马字输入、按中国念法用汉语拼音输入；配有几万词汇的日语词典。

其他语种分别定义了各语种键盘，在标准键盘上可一键输入，简捷迅速，通过键盘切换各语种工作状态和DOS状态十分方便。

对各语种字符，使用了多种字库。可由键盘或程序设置多种打印字体，调整打印的字距和行距，实现16 * 16和24 * 24点阵混合打印。可直接与多种打印机相联，并根据需要输入打印控制码灵活地增加与本系统相联的打印机种类。

- 支持DBASE、LOTUS1-2-3、WS等应用软件以及各种编程软件。
- 系统配有改造过的字处理软件和打印输出软件，能直接编辑和打印九种文字的文件。

北京信通集团公司软件工程公司

地　址：北京海淀路31号　　电　话：2563689，289705，2567864

邮政编码：100080

目　录

第一部分　有关PC DOS一些问题的深入分析

第二部分　对显示设备的开发

第三部分　软件开发实例

主要的内存单元和口地址

本文分四部分详尽地介绍一些重要的程序员经常碰到的内存单元和口地址。这四部分是：ROM BIOS的数据区，BASIC程序的工作区和数据区，ROM BIOS程序区，输入/输出口。准确地理解这些内存单元和口地址的含义和用法，对于理解PC机的运行和开发PC机是非常有意义的。

一、ROM BIOS的数据区

从绝对地址0开始的1K字节里，存放着一张中断向量表。紧随中断向量表之后，即从地址00400H开始，存放着一些重要的数据。这些数据是在引导过程中由ROM BIOS程序装入的，它们控制着机器的许多操作。虽然这些数据在设计时是只供ROM BIOS使用的，但是用户程序也可以读这些数据来了解机器的状态。某些数据甚至还可以被修改，以便改变对机器的控制。即使用户不想使用这个数据区中的信息，但如能仔细地研究一下这些信息，定会加深对PC机工作过程的理解。

在以下的叙述中，一律假设段地址是40H。

0H 4个字

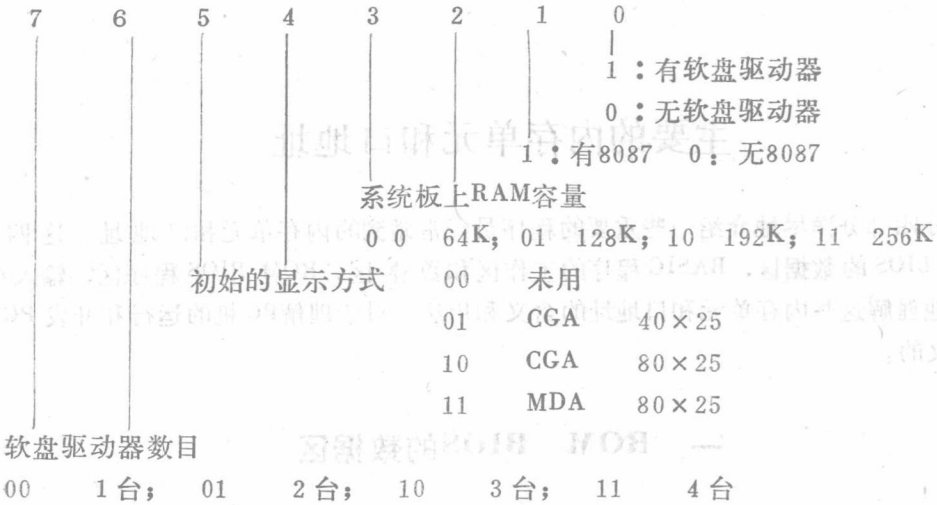
这4个字用来存放4个串行口的起始口地址。PC/XT的ROM BIOS只支持两个串行口，每个串行口占用连续8个口地址，3F8H至3FFH，或2F8H至2FFH，但在这4个字中存放的仅仅是每块串行通讯板的起始口地址。在引导过程中，ROM BIOS程序会按先3F8H后2F8H的顺序来检查系统中是否存在串行口及存在几个串行口。如果存在，则按先后顺序填入这个区域。这就是说，如果只存在一个串行口，则这个串行口就是COM1，不管其起始口地址是3F8H还是2F8H。如果存在两个串行口，则口地址3F8H相应于COM1，口地址2F8H相应于COM2。

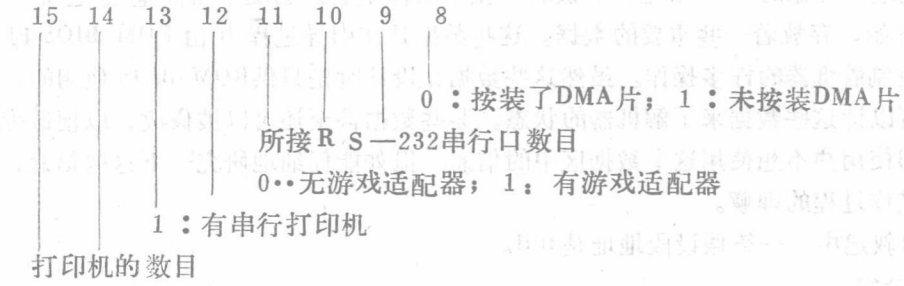
8H 4个字

这4个字用来存放4个并行口的起始口地址，PC/XT的ROM BIOS只支持三个并行口，每个并行口占用连接3个口地址，3BCH至3BEH，378H至37AH，278H至27AH，但在这儿存放的仅仅是起始口地址。在引导过程中，ROM BIOS程序按3BCH，378H，278H的顺序来检查系统中是否存在并行口及存在几个并行口，如果存在，则按先后顺序填入这个区域。与对串行口的处理一样，依次称为LPT_1，LPT_2，LPT_3。

10H 字

这个字保存着系统中设备的配置情况，可以通过中断11H读到这个字，各位的意义如下：





12H 字节

制造厂家的测试标志

13H 字

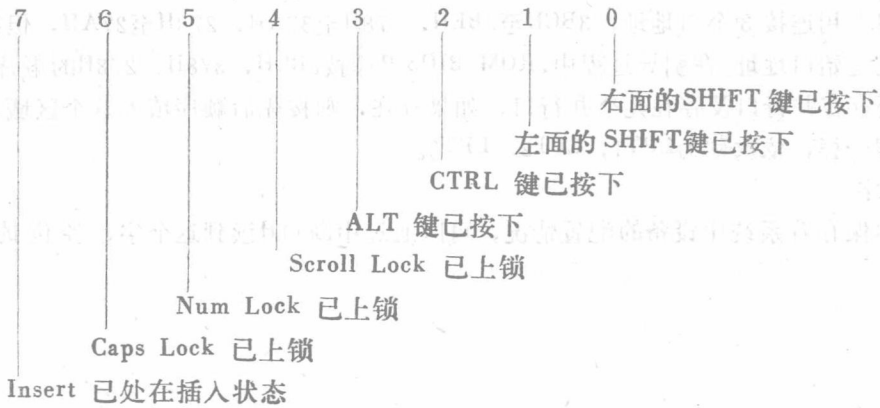
这个字含有系统中总的RAM的容量，以K字节为单位。当执行INT 12H时，返回的就是这个字的内容。

15H 字

扩展的（EXPanSion）RAM容量，以K字节为单位。

17H 字节

移位键的当前状态，下表列出了各位为1时的含意：



18H　字节

键盘状态的第二个字节，下表列出了各位为1时所表示的意义：

19H　字节

在PC机中，可以按如下的办法来打入任何一个ASC II字符：按下ALT键，从右面的小数字键盘上以10进制形式打入所需要字符的ASCII码，然后再松开ALT键。这样打入的数（0～255）就存在这个字节里。

1AH　字

键盘缓冲区的头的指针。这个指针一般用来取字符。

1CH　字

键盘缓冲区的尾的指针。这个指针一般用来存字符。

1EH　16个字

环形键盘缓冲区，可以缓冲16次击键，直到被INT 16H处理程序取走。

3EH　字节

这个字节用来表示软盘驱动器开始寻道以前是否需要先回到0 0 0道。位0到位3分别相应于软盘驱动器的0到3，位4到位7不用。如果某一位为0，则相应驱动器在寻道前会先回到0 0 0道。当驱动器操作出错时，往往首先就采取这种办法来试图解决。

3FH　字节

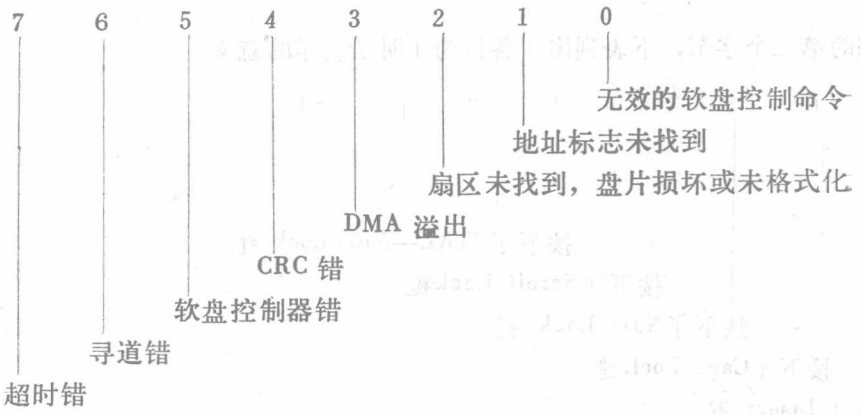
这个字节表示软盘驱动器的状态。位0到位3分别相应于软盘驱动器0至3。如果某一位为1，则表示相应的软盘驱动器的马达正在旋转，可以立即进行读、写和寻道操作，不需要等待。否则，应先启动马达并等待转速达到平稳。位7为1则表示某一台软盘驱动器正在执行写操作。

40H　字节

这个字节用来设置每次软盘操作完成后到软盘驱动器马达停止旋转之间的延迟时间，在PC/XT机中单位为55毫秒。

41H　字节

这个字节描述软盘出错的原因。某位为1则表示发生了对应的错误，为0则表示没有发生这个错误。



42H　7个字节

这7个字节存放的是从NEC软盘控制器芯片读回的软盘驱动器的状态，可参考该芯片的说明手册，对用户程序用处不大。其中的45H、46H，47H单元分别存放上一次读写或寻道操作的道号，头号和扇区号，48H单元存放每扇区的字节数，可能的值是0，1，2和3，分别代表每扇128字节，256字节，512字节和1024字节。当对硬盘进行读写时，前4个字节也用来存放4字节的错误状态信息。

49H　字节

从49H开始的30个字节的区域是用于显示控制。ROM BIOS程序用这个区域来保存重要的显示信息。用户程序可以查看这30个字节中任何一个字节的内容，但在大部分情况下，不可以往这些单元中写新的内容。否则，可能产生意想不到的结果。对用户程序很有用的且可以修改的就是保存光标位置的几个单元。

49H这个单元内含有当前的显示方式，显示方式号与ROM BIOS调用INT 10H中的显示方式号具有相同的意义。在本书的“开发加强显示适配器”一文中对各种显示方式作了详细的说明。

4AH　字

这个字保存当前显示方式（文本）下，每一行显示的字符的数目。随显示方式的不同，可以是20，40或80。

4CH　字

当处于文本显示时，在显示缓冲区中每一页的显示长度，以字节为单位。

4EH　字

当前显示页的起始地址相对于整个显示缓冲区的起始地址（随显示适配器的不同而不同，MDA是B000H，CGA是B800H）的偏移量。

50H　8个字

这8个字用来保存各页（从页0开始，最多8页）的当前光标位置。在每个字中，第一个字节保存的是列号（0至19，39或79），第二个字节保存的是行号。修改这些单元中的值就可以很方便地移动光标位置，但是这种修改要直到下一次屏幕输出时才会起作用。

60H　字

这个字用来保存光标的大小，第一个字节（60H）保存光标的起始扫描行号，第二个字节保存光标的结束扫描行号。若61H的位5为1，则不显示光标。与50H开始的光标位置不

一样，仅仅修改这个单元的内容不会自动引起光标大小的修改。

62 H　字节

这个字节保存当前的显示页号。

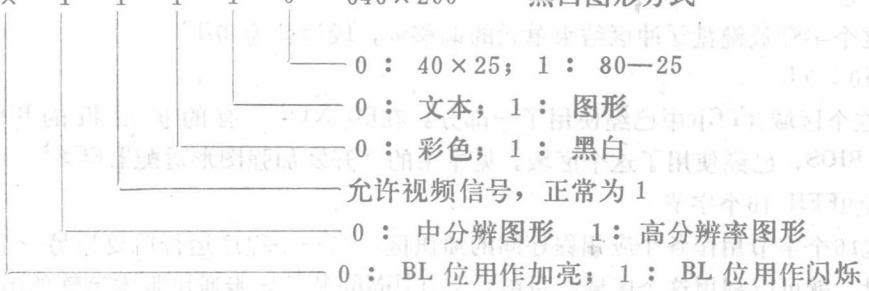
63 H　字

这个字保存分配给CRT控制器芯片6845的起始口地址，在MDA中为$3B_4H$，在CGA中为$3D_4H$。

65 H　字节

这个字节保存的是CRT方式寄存器中当前的内容，

7	6	5	4	3	2	1	0		
		1	0	1	1	0	0	40×25	黑白文本方式
		1	0	1	0	0	0	40×25	彩色文本方式
		1	0	1	1	0	1	80×25	黑白文本方式
		1	0	1	0	0	1	80×25	彩色文本方式
		×	0	1	1	1	0	320×200	黑白图形方式
		×	0	1	0	1	0	320×200	彩色图形方式
		×	1	1	1	1	0	640×200	黑白图形方式



66 H　字节

这个字节保存当前显示的颜色属性。

67 H　5个字节

这5个字节用于磁带的控制。

6 CH　4个字节

这4个字节保存的是当前的时间值，每次产生时间中断时，这个值就加1。在PC/XT中，每隔约55毫秒产生一次时间中断（每秒18.2次）。当计数达到24小时时，起始计数值又恢复到0，并使70H单元的内容置为1。通过ROM BIOS的INT 1A中断处理程序可以读到这4个字节的值。

70 H　1个字节

如上所述，每当6CH开始的4个字节计满24小时以后，就会使这个单元的内容置1，表示已经过了一天。每当用户程序通过INT 1AH读取时间时，则根据这个单元的值返回相应标志，同时使这个单位的内容恢复到0。因为这是假设每个发出INT 1AH指令的程序都能根据返回的标志确定是否增加天计数。应该注意的是，每当计满24小时以后，这个单元的内容是置1而不是加1，所以在发INT 1AH指令以前若已经过去了两天，则用户程序是无法知

道这件事实的。

71 H　字节

如果这个字节的位 7 为 1，则表示BREAK键组合已经按下了。

72 H　字

当系统正在进行热启动时，这个字的内容被置为1234H。

74 H　4 个字节

这 4 个字节依次存放硬盘系统的当前状态，硬盘驱动器台数，硬盘驱动器选择控制字节和硬盘控制器板的起始口地址的偏移量。这几个字节对用户程序关系不大，详细情况请见硬件技术参考手册。

78 H　4 个字节

这 4 个字节依次存放 4 个并行口的最大等待时间值。若输入输出操作不能在这个时间内完成，则认为是超时错。单位是1.6秒。

7CH　4 个字节

这 4 个字节依次存放 4 个串行口的最大等待时间值，单位为1.6秒。

80 H　字

这个字存放键盘缓冲区起始单元的偏移量，段地址为40H。

82 H　字

这个字存放键盘缓冲区结束单元的偏移量，段地址为40H。

84H 至0 F 0 H

这个区域在I Cjr中已经使用了一部分。在PC/XT中，有的扩展板的BIOS，如EGA的ROM BIOS，已经使用了这个区域。见本书的“开发加强图形适配器EGA”一文。

0F0H至0FFH 16个字节

这16个字节用作各个应用程序间的通讯区。当一个程序运行后要为另一个程序留下一些信息时，就可以利用这个区域。目前，只在IBM的某些异步通讯版本等软件中，用到了这个区域。

100 H字节

这个字节是由DOS和BASIC用来控制打印屏幕的操作，有三种可能的值

00　正确完成

01　打印屏幕的操作正在进行

FF　在打印屏幕过程中发生了错误

104 H　字节

这个字节用在只有一个软盘驱动器的系统中。在这种系统中，这台软盘既作为A 盘，又作为 B 盘。如果正在作为 A 盘，则DOS置这个字节为 0 。如果正在作为 B 盘，则DOS置这个字节为 1 。

10FH　字节

当BASIC正在执行SHELL时，这个字节为2。

110H　字

这个字由BASIC保存隐含的数据段段地址DS的值。在BASIC 程序中，允许用户通过“DEF SEG = 值”语句来建立他自己的数据段段地址。用户也可以用不带“= 值”的DEF

SEG语句使数据段段地址恢复到隐含值。因为这个字是很重要的，所以不要对它进行修改。

112 H　2个字

这两个字是BASIC的一个中断向量，指向BASIC的时钟中断处理程序。为了使得声音效果更好一些，BASIC使时钟中断频率提高到ROM BIOS初始的每秒18.2次的4倍。为了不影响其它程序的运行，在每4次时钟中断以后，BASIC再唤醒一次原来ROM BIOS的时钟中断处理程序。

116　2个字

这两个字是BASIC的一个中断向量，指向它的BREAK键处理程序。

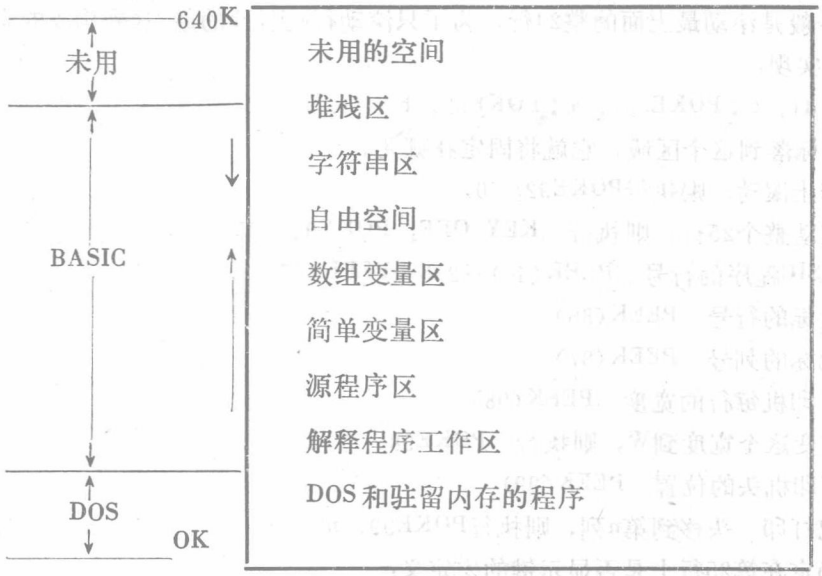
11AH　2个字

这两个字是BASIC的一个中断向量，指向它的软盘出错处理程序。

二、BASIC的一些重要内存单元

在下面的叙述中，把1981年和1982年生产的PC机称为老的PC机，它们的系统板上一般只有64K内存，它们的ROM的日期分别为4/24/81或10/19/81。以后生产的PC机称为新PC机，包括PC/XT和PC/AT。

BASIC程序运行时占据的空间如图所示，共可分为七个部分。第一部分是解释工作区，其余六部分是BASIC的程序、变量和堆栈区。



源程序区，简单变量区和数组变量区是互相紧接着存放的，三者联合起来从低地址往高地址增长。堆栈区和字符串区是从高地址往低地址增长。随着DOS版本的不同，BASIC装入时的起始地址也是不同的。在下面的所有叙述中，都假设DS：0是指向解释程序的头的，所有的偏移量都是相对于这个段地址的。为了保证做到这一点，应该先执行一条不带地址的DEF SEG命令。为了在BASIC中使用PEEK，POKE语句时更方便一些，下面的偏移量都是以10进制形式给出的

1．源程序起始地址　PEEK（48）＋256＊PEEK（49）

每个源程序行在这个区域内的存贮办法是：头两个字节是一个指针，表示下一行的起始地址。接着的两个字节是行号。再接着就是行内容本身，所有的BASIC命令都被一个字节的代码所代替。行的结束是以一个字节的00H来表示的，整个源程序的结束是以另加两个字节的00H来表示的。

2．简单变量区的起始地址　PEEK(856)＋256＊PEEK(857)

3．数组变量区的起始地址　PEEK(858)＋256＊PEEK(859)

4．自由空间的起始地址　PEEK(860)＋256＊PEEK(861)

5．字符串空间

字符串空间是从高地址增长的。它的最高地址是PEEK(815)＋256＊PEEK(816)，它当前的最低地址是PEEK(778)＋256＊PEEK(779)。这个最低地址也应该就是自由空间的结束地址。

6．堆栈区

紧接着字符串空间的是堆栈区，直到可供BASIC使用的内存的最高端，除非通过CLEAR语句保留了一部分空间。

堆栈区的最高地址是　PEEK(44)＋256＊PEEK(45)

当前堆栈指针指向的地址是　PEEK(837)＋256＊PEEK(838)

7．最近发生错误的IBASC语句行号　PEEK(839)＋256＊PEEK(840)

最近发生的错误的错误号码　PEEK(40)

8．BASIC一般是滚动最上面的整24行，为了只滚动行a到行b的最左c列构成的窗口，则可以由下列语句实现：

POKE 41，c：POKE 91，a：POKE92，b

一旦光标落到这个区域，它就将固定在哪儿。

为了禁止滚动，则执行POKE92，0

为了滚动整个25行，则执行　KEY OFF：POKE92，25

9．当前BASIC程序的行号　PEEK(46)＋256＊PEEK(47)

10．当前光标的行号　PEEK(86)

当前光标的列号　PEEK(87)

11．当前打印机每行的宽度　PEEK(98)

为了改变这个宽度到W，则执行　POKE98，W

12．当前打印机头的位置　PEEK(99)

为了把打印　头移到第n列，则执行POKE99，n

13．为了确定在第25行上是否显示键的宏定义：

PEEK(113)＝0　　不显示

＝1　　调入BASIC时则显示

＝255　　执行KEY ON语句后显示

如果当前已经显示键的宏定义，为了不显示这个宏定义，则在执行“POKE(113)，0”语句以后，还要用CLS语句清除第25行，或用LOCATE语句往第25行上写空格。

14．上一次执行的语句的位置　PEEK(835)＋256＊PEEK(836)

这个位置的内容可能是一个表示行的分隔符的00H，也可能是同一行内的语句分隔符冒号“：”。

15．BASIC数据段的段地址，PEEK(848)+256*PEEK(849)

这个地址也就相当于BASIC解释程序工作区的起始地址。

16．为了测试当前的程序是否被保护，则

PEEK(1124) = 0　　不被保护

= 255　　被保护

为了防止当前的程序被LIST或被EDIT，则执行　POKE　1124，255

7．键盘宏定义

键盘宏定义的存贮区是从偏移量1619开始的160个存贮单元。共存贮10个功能键的宏定义，每个宏定义占16个字节，最后一个字节是00H，宏定义字符串最多只能15个字符。没有进行宏定义的功能键也是以00H填充。

18．BASIC的程序段前缀的段地址保存在：

PEEK(1794)+256*PEEK(1795)

知道这个地址以后，利用下列程序就可以找出环境字符串的段地址ENV：

```
10  DEF  SEG
20  PSP = PEEK (1794) + 256 * PEEK (1795)
30  DEF  SEG = PSP
40  ENV = PEEK (44) + 256 * PEEK (45)
```

知道了这个地址，利用下列程序，可以读出调用BASIC时的命令行参数：

```
10  DEF  SEG
20  PSP = PEEK (1794) + 256 * PEEK (1795)
30  DEF  SEG = PSP
40  FOR  I = 130  TO  129 + PEEK (128)
50  PRINT  CHR $ (PEEK (I) ) ;
60  NEXT
```

三、ROM BIOS的程序区

ROM程序是固定在内存的高端的，一般用户程序只调用其中的子程序，而不关心ROM程序中所包含的数据信息。但是，如果用户程序需要知道ROM的版本号，或需要知道运行此用户程序的究竟是什么类型的机器等，则可以，也必须利用ROM中的数据信息。在以下的叙述中，假设段地址是F000H。

1．ROM的版本号

从偏移量FFF5H到FFFCH的8个字节里，保存了ROM发表的日期。所有IBM PC机的ROM中都有这个信息，但兼容机中却只有少数的ROM中有这个信息。至今为此，IBM PC机已有许多个ROM版本，如下表所示：

ROM发表日期	机器
04/24/81	最早的PC机

10/19/81	改进的PC（去掉了某些错误）
08/16/82	最早的PC/XT机（能处理576K内存）
10/27/82	改进的PC/XT机（能识别添加的硬盘ROM BIOS和处理640K内存）
11/08/82	最早的Portable PC机
06/01/83	最早的PCjr机
01/10/84	最早的PC/AT机

2．机器类型标志

偏移量FFFE 处的一个字节是说明PC机类型的ID标志

标志（16进制）	机器类型
FF	PC机及早期的PC/XT机
FE	PC/XT机
FD	PCjr机
FC	PC/AT机
2D	Compaq（等效于PC）
9A	Compaq-Plus（等效于PC/XT）

四、用户程序经常使用的口地址

下面是一张口地址分配表

说明	PC/XT	PC/AT
DMA控制器（8237A—5）	000—00F	000—01F
中断控制器（8259A）	020—021	020—03F
定时器（8253—5；AT机为8254.2）	040—043	040—05F
PPI（8255A—5）	060—063	
键盘（8042）		060—06F
DMA页寄存器（74LS612）	080—083	080—09F
NMI屏蔽寄存器	0A	070—07F
中断控制器2（8259A）		0A0—0BF
声音发生器（SN76496N）		
DMA控制器2（8237A-5）		0C0—0DF
清除/复位数值协处理机		0F0—0F1
数值协处理机		0F8—0FF
游戏控制器	200—20F	200—207
扩展箱	210—217	
并行打印机（第2台）		278—27F
串行口（第1个）	3F8—3FF	3F8—3FF
串行口（第2个）	2F8—2FF	2F8—2FF

Prototype板	300—31F	300—31F
硬盘控制板	320—32F	1F0—1F8
并行打印机（第1台）	378—37F	378—37F
SDLC	380—38F	
双同步通讯（第2台）		380—38F
双同步通讯（第1台）		3A0—3AF
单色显示/打印板	3B0—3BF	3B0—3BF
彩色/图形板	3D0—3DF	3D0—3DF
软盘控制板	3F0—3F7	3F0—3F7

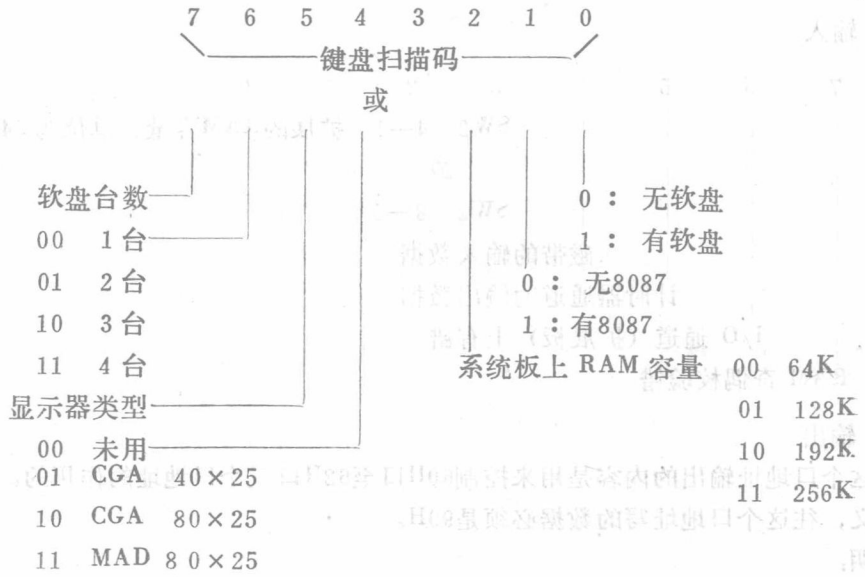
下面以PC/XT为主阐述程序员经常用到的一些输入输出口地址的用法。所有的输入输出都是一个字节。

1．PPI（8255A），60H至63H

在系统板上有一片8255A—5芯片，用来对键盘、声音、系统开关等进行管理。

60H口　输入

从这个口地址输入的一个字节，根据61H口位7的设置，其内容可能是键盘扫描码，也可能是反映系统板上开关SW1的通断状态。



61H口　输出/输入