IBM PC实用技术高级专题

软件解密技术

丁红卫 编

北京希望电脑公司

IBM PC 实用技术高级专题

软件解密技术

丁红卫　　编

北京希望电脑公司
一九九二年三月

编者的话

编写本书的目的是想让广大计算机用户在阅读了本书之后能对软件进行加密保护，或者进行解密，着重于软件的解密。看完本书以后，您会感觉到，软件解密并不是什么很困难的事。

全书可分为两大部分。前四章简述了计算机体系结构以及磁盘数据的存贮格式等一些基础知识，这里还阐述了软件加密原理与原则；第五章集中分析了上百种不同层次、不同类别软件的解密方法。具体章节的内容如下：

- ▲ 第一章是绪论，这里介绍了计算机的总体结构，以及各主要组成部分的功能。这是基础知识，读者必须了解。
- ▲ 第二章逐条分析了 BIOS 及 DOS 中重要的中断调用。
- ▲ 第三章是 PC 机软盘驱动器概论。这里介绍了磁盘驱动器的结构及功能、磁盘记录数据的各种不同方法以及磁盘数据格式等内容。这一章是关键部分，它是我们掌握软件解密方法更直接的基础。
- ▲ 第四章介绍了软件加密的一般方法。针对 PC 机磁盘控制卡的某些局限，这里还提出了一些不常用但又安全可靠的加密保护方法。
- ▲ 整个第五章是解密实例。

本书大部分解密实例是用方便的调试工具 DEBUG 实现的。为了方便读者，书后以附录的形式介绍了调试程序 DEBUG，那里对 DEBUG 各条命令的功能及其使用方法和使用过程中所出现的问题都有说明，读者可自行参阅。

编者认为，一本再好的书，也不能穷尽所有的知识，因此，建议读者在阅读过程中积极动脑、善于动脑，抓住症结、学会方法，这样方可举一反三，一通百通。经验之谈是：解密需要耐心和时间。

由于编者水平有限，书中不足或错误之处在所难免，敬请广大读者批评指正。

编者

一九九一年一月

本书除教读者解密技巧之外，还涉及到一些常见的技术问题，这些都是广大用户心得精华所在，读完以后，你将在最短的时间里精通 PC 的各种解密技巧。

目　录

第一章 绪 论

对大多数的计算机用户来说，磁盘驱动器似乎是一种很复杂的电路装置，但又是所有计算机用户必定要用到的设备。在微型计算机系统中，一旦没有驱动器这种设备，计算机就会丧失处理事务的能力。事实上，只要一打开计算机的电源，计算机就从磁盘中去找所需要的数据，以使计算机能很容易地让人使用。

平常我们是靠着 MS-DOS(或叫 PC-DOS)的磁盘操作系统来使用驱动器，把有用的数据存入到磁盘上，或是将数据由磁盘上装入计算机中。有了 MS-DOS 操作系统，我们可以很容易地操作驱动器，进行读取(read)和写入(write)等工作，而不必去研究驱动器内复杂的构造及其工作原理。甚至一个对磁盘及驱动器结构一窍不通的人，只要告诉他几个 DOS 命令的用法，如：FORMAT、COPY、DISKCOPY ……等等，他就能很容易地使用计算机了。

但是，对于一个程序设计人员、软件工程师、或者研究拷贝与保护磁盘的“专家”来说，仅通晓 MS-DOS 显然是不够的，尤其是后者，更需要进一步详细地研究驱动器的使用方法，这就不仅限于 MS-DOS 的范围，还须弄清楚 I/O 的控制方法、磁盘的结构、数据的格式，更重要的还要知道如何用程序来控制驱动器的动作，以期能更好地控制驱动器。

本书采用循序渐进的方式，为您慢慢揭开驱动器神秘的面纱，并以 PC/XT 机为例，讲述如何保护驱动器，以保障程序设计人员的智慧财产权。而在研究驱动器的程序控制以前，必须先弄清一些基本概念，象驱动器的控制程序都是以 8088 机器语言来编写的，所以必须先了解 8088 汇编语言；又如，MS-DOS 中许多有用的子程序，可供程序设计人员写程序时参考和运用；这些都是读者必备的背景知识。读完本书，你会发现，保护磁盘内的程序不被拷贝，原来是那么简单的事！而且你将会对 PC/XT 的驱动器及一切有关的设备了若指掌！

让我们从基楚开始，先看看计算机与驱动器之间的关系吧。

§ 1.1 微型计算机简介

所有的计算机系统均是由三个主要部分组成的，我们称它三“要素”(Main element)，它们是：

1. 中央处理机 CPU(Central Processing Unit)；
2. 存贮器(memory)；
3. 输入／输出(Input/Output)设备(device),或称端口(port)；

这三个要素所构成的微型计算机方框图，见下页。

注：地址总线 = ADDRESS BUS
数据总线 = DATA BUS
控制总线 = CONTROL BUS

§ 1.1.1 CPU

CPU 是微型计算机的心脏，它具有加、减等数字运算、逻辑运算以及“时序”等方面

的功能。

CPU 的动作是受一系列指令(instruction)控制的，这些指令的集合称作程序(program)。程序存于内存当中，数据也是存在于内存当中的，不过数据的处理还取决于处理数据的程序。

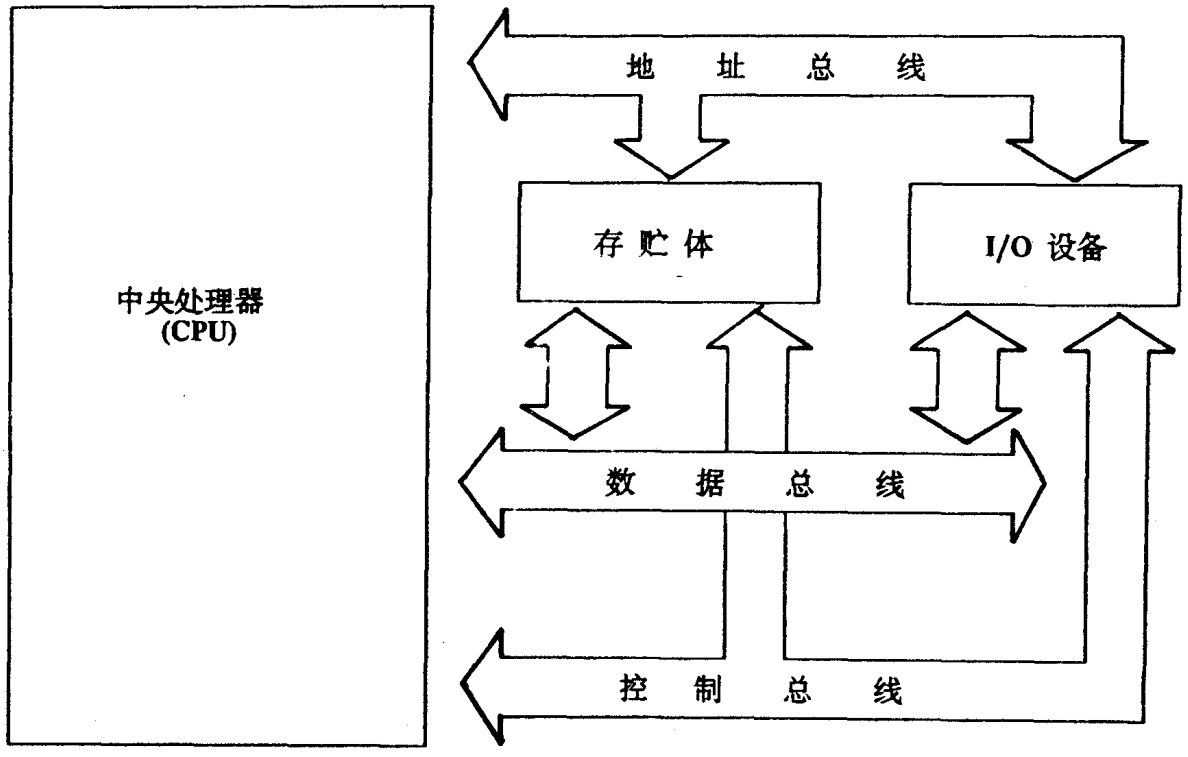

图 1-1

CPU 从输入端口(input port)读取数据和控制信号(即指令)，一次执行一条指令，然后把处理后的结果和控制信号由输出端口(output port)传送至外界。典型的 CPU 由下列三种功能单元(functional unit)组成：

. 寄存器(register)

. 算术／逻辑单元(arithmetic/logic unit，简称 ALU)

. 控制电路(control circuitry)

现在分别介绍如下：

▲ 寄存器是 CPU 内暂时用来寄存存贮器地址(memory address)、状态码(status code)、以及其它一些在程序执行时有用的数据的。不同的处理机有不同数目与长度的寄存器。象 8088、8086 等 CPU 具有 16 位的寄存器。

▲ 所有的 CPU 都有 ALU。ALU 内含一个加法器(adder)，它可以对存贮器、寄存器、或其他输入的数据做二进位的算术运算。

有些 CPU 能进行如乘、除法的复杂运算。ALU 的其他功能还包括布尔运算(Boolean logic)和数据移位操作(data shifting)。

另外 ALU 还包括标志位(flag bit)，它用来标志运算结果的信息，象正负号(sign)、零(Zero)、进位(Carry)和校验(Parity)等。

▲ 控制电路是用来协调各部分的活动。它利用时钟(CLOCK)来保持所有处理动作的

顺利进行。控制电路将指令解码(Decode)，并对 CPU 内部及外部的各单元发出控制信号以执行正确的动作。

§ 1.1.2 存贮器

微型计算机一般采用半导体介质来存贮程序和数据。存贮器可分为两种，随机存取存贮器(Random Access Memory，简称 RAM)和只读存贮器(Read Only Memory，简称 ROM)。为了扩充存贮容量，微型计算机系统时常使用如磁盘和磁带等数种可以大量存贮数据的磁性材料。

§ 1.1.3 输入／输出(I/O)设备

I/O 设备也称作外围设备(peripherals)，它们是供 CPU 和外界通信所使用的。例如在一台附有 CRT 终端的典型的微型计算机系统上，输入端口(或称通道 channel)是连接到键盘(key board)，而输出端口是连接到可以显示字符和图形的的硬件设备。本书所要详细研究的驱动器，也是一种 I/O 设备，它兼具输入与输出的功能。

§ 1.2 总线浅说

在上述三要素之间，有一种叫总线(bus)的结构，它是用来负责相互间传送数据的。总线可分为地址总线(address bus)、数据总线(data bus)和控制总线(control bus)。

CPU 就是靠这一组称作总线的水平电缆，连接至存贮器和 I/O 设备，请参看图 1－1。下面略述这些总线的用途：

△ 数据总线

数据总线(data bus)是用来在 CPU、存贮器和 I/O 设备之间传送数据(data)的。这些数据可以是 CPU 所发出的指令，也可以是 CPU 要传送至 I/O 端口、或 CPU 要从 I/O 端口接收的数据。8088 CPU 的数据总线是 8 位的，而 8086 CPU 的数据总线的 16 位的。

△ 地址总线

CPU 如果要选择某个特定的存贮体或 I/O 设备，只要指定存贮体或 I/O 的地址(address)即可。

△ 控制总线

控制总线(control bus)可将控制信号传给存贮体及 I/O 设备，以决定信息要进入或离开 CPU，并作正确的传送。

△ 总线周期

对总线有了大致的了解以后，笔者再解释一下什么叫总线周期(bus cycle)。

当计算机程序在执行时，数据在内存和 I/O 设备之间传送，有的从内存传送到 I/O 设备，也有的从 I/O 设备传送到内存。数据从系统的这部分传送至另一部分的每一步，称作一个总线周期，或称作机器周期(machine cycle)。这些周期(cycle)的定时工作是由 CPU 的时钟信号(clock signal)来完成的。有些动作象取指令(instruction fetch)、读取内存(memory read)、写入内存(memory write)、从输入端口读数据、向输出端口写数据等，要花掉一个或一个以上的总线周期才能完成。

总线周期的长度和时钟脉冲信号的频率有关。常见的时钟频率有 5MHz、8MHz、10MHz 几种，象 8086 的 8MHz 版本，其时钟周期为 125 毫微秒(nanosecond)，等于 0.125 微秒(microsecond)。

总线周期开始时，CPU 会发出一个机器码到地址总线上，以确定将被处理的内存地址

或者 I/O 设备，接着 CPU 在控制总线上发出一个真正的指令(activity command)。第三步，CPU 在数据总线接收或发送数据。

CPU 然后执行指令所要求的逻辑、算术或 I/O 操作！

CPU 是利用称作程序计数器(program counter)的寄存器来得知指令的顺序的。程序计数器中存在内存下一条指令的地址，当目前的指令执行完后，CPU 会到程序计数器所指的地址去取出下一个要执行的指令。取出后，程序计数器会自动地指向下一条指令的地址。最近 Intel 公司出品的 CPU，“程序计数器”这个术语大多已不再使用，而以“指令指针”(instruction pointer)来取代，然而其意义是一样的。

通常，在一给定的指令执行之后，指令指针的值要改变。CPU 会自动地由内存中取得(fetch)指令，然后将它解码(decode)，如此顺序地往下执行，直到程序结束，除非有特殊的指令要 CPU 去执行程序内存中的其他部分的指令。

某些情况会中断指令执行的正常顺序，例如，在一给定的总线周期中可能会加入一个“等待”状态(wait state)，以便提供更多的时间给内存或 I/O 设备，便于它们与 CPU 通信。当从内存传输数据的速度比 CPU 所要求的慢时，就必须用到等待状态。在这种情况下，当内存接收到读／写内存动作开始的 CPU 信号，就会要求等待状态。等到存贮体回答以后，就发出信号给 CPU，结束等待状态，再继续处理正常程序。

§ 1.3 中断

能改变指令执行顺序的另一种情况便是“中断”(Interrupt)。例如：假如一台计算机正在处理大量的数据，而这些数据有些部分是要输出到打印机的，CPU 虽然可以在单一的总线周期内将所给定数量的数据送到打印机上，但是打印机可能得花掉数个总线周期的时间才能将该数据所指的字符(character)打印出来。换句话来说，就是 CPU 送出字符给打印机以后，还要等待打印机把字符印好，在此期间 CPU 必须保持闲置(idle)，一直到打印机能够接受下一个字符为止。意即 CPU 要等到打印机把事做完以后，才可继续动作。中断的功能就是让 CPU 能将数据输出到打印机，然后回到其他数据的处理上。

当打印机准备好接收下一组数据时，就经过一条特殊的中断控制线(interrupt control line)向 CPU 发出信号。当 CPU 收到此中断信号(interrupt signal)之后，就暂停主程序的执行，并且自动撤换(switch)到输出到打印机的指令上，待输出到打印机的指令完成后，CPU 就自刚才暂停处理的地方继续主程序的执行！

通常会有数个中断设备(interrupting device)共用一个 CPU 的情形，为了能兼顾所有的中断设备，中断就得分出优先顺序。当同时遇到两个或更多的中断时,具有较高优先权的中断将优先得到服务(Service)。

§ 1.4 直接存贮器存取(DMA)

直接存贮器存取(direct memory access，简称 DMA)是改进微型计算机效益的另一方法。

在一般的输入／输出(I/O)操作中，是 CPU 管理全部数据的传输操作，CPU 执行 I/O 指令时，是先由输入设备(input device)传送数据给 CPU，然后数据再从 CPU 传送到指定的内存地址，同样,来自内存要到输出设备的数据也要经过 CPU。某些外围设备(peripheral device)从内存传输数据的速度，比 CPU 在程序控制下进行传输的速度还要快。如果采用

DMA 的话，CPU 可让外围设备来控制总线，从而直接从内存传输数据，不必再经过 CPU。当 DMA 的传输完成后，外围设备释放总线，然后 CPU 就自刚才停止之处重新处理指令。

§1.5 寻址方式

CPU 用地址总线所提供的地址可以选择一个指定的内存地址或是一个 I/O 设备。依照所执行操作的不同，这些地址也有不同的产生方法，而产生这些地址的方法就称作寻址方式(addressing mode)。

最简单的寻址方式中，所要的数据项就包含在正要执行的指令之内。比较复杂的寻址方式中，指令内含有数据的的存贮器地址，或指令会参考一个内含有数据存贮地址的 CPU 寄存器。

最后，在某些处理机内部，这条指令会使控制电路产生一个复杂的地址。此地址是数个地址成份的总和，就象是多个寄存器加上含有指令本身在内的数据。

通常，功能愈强大的微处理机，它的寻址方式愈多。

§1.6 磁盘驱动器

前面曾经提过，驱动器是一种很重要的 I/O 设备，CPU 可利用磁盘控制卡(Floppy Disk Cotroller，简称 FDC)来控制磁盘的读写操作。

实际上，驱动器的一切动作都已固定在 FDC 上的一块叫做 uPD765A 的 IC 中，因此可简化原来操纵驱动器的许多复杂的硬件动作，并将其归纳为十五种不同的指令。如此以来，虽然用户仅需对 uPD765A 发出数个指令组，便可让驱动器动作，但是这也限制了原来驱动器的任意读写功能，使得保护磁盘的花样大为逊色，还不如 APPLE 计算机那样多姿多采了。

uPD765A 具有三种寄存器可和 CPU 沟通，在 PC/XT 的计算机上，这三种寄存器的 I/O 地址如下：

1. 数据输出寄存器(digital output register)I/O 地址为 3F2H；
2. 主状态寄存器 (main status register) I/O 地址为 3F4H；
3. 数据／状态寄存器(data/staus register)I/O 地址为 3F5H；

这些寄存器均是 8 位的。

“数据输出寄存器”仅能供输出，它是决定驱动器的开与关、重置等工作的。

“主状态寄存器”仅能供系统读取，它含有 FDC 状态数据，可随时供系统使用。

“数据／状态寄存器”可以读取也可被写入。实际上它是一个堆栈，而每次在数据总线上只含有一个寄存器，因此在读写此寄存器时，一定要依照固定的顺序来进行。

它在命令阶段(command phase)时是作为数据寄存器，可存贮数据、指令、参数、及驱动器的状态数据。在结果阶段(result phase)时是当成状态寄存器，用以得知磁盘动作执行的结果，状态寄存器有四个，分别为 ST0、ST1、ST2、ST3。

驱动器的动作可分成三个阶段：

1. 命令阶段(command plase)

将所需数据依次写到 FDC 的数据寄存器。

2. 执行阶段(execution phase)

FDC 执行所得到的命令，只要命令阶段中最后一个指令参数写到 FDC 后，就立即开始执行阶段。

3. 结果阶段(result phase)

磁盘动作执行完毕后，在状态寄存器或数据寄存器中，会产生一串状态参数来表示执行的结果，以供 CPU 读取。

接着我们来看磁盘 I/O 和 CPU 之间的沟通：

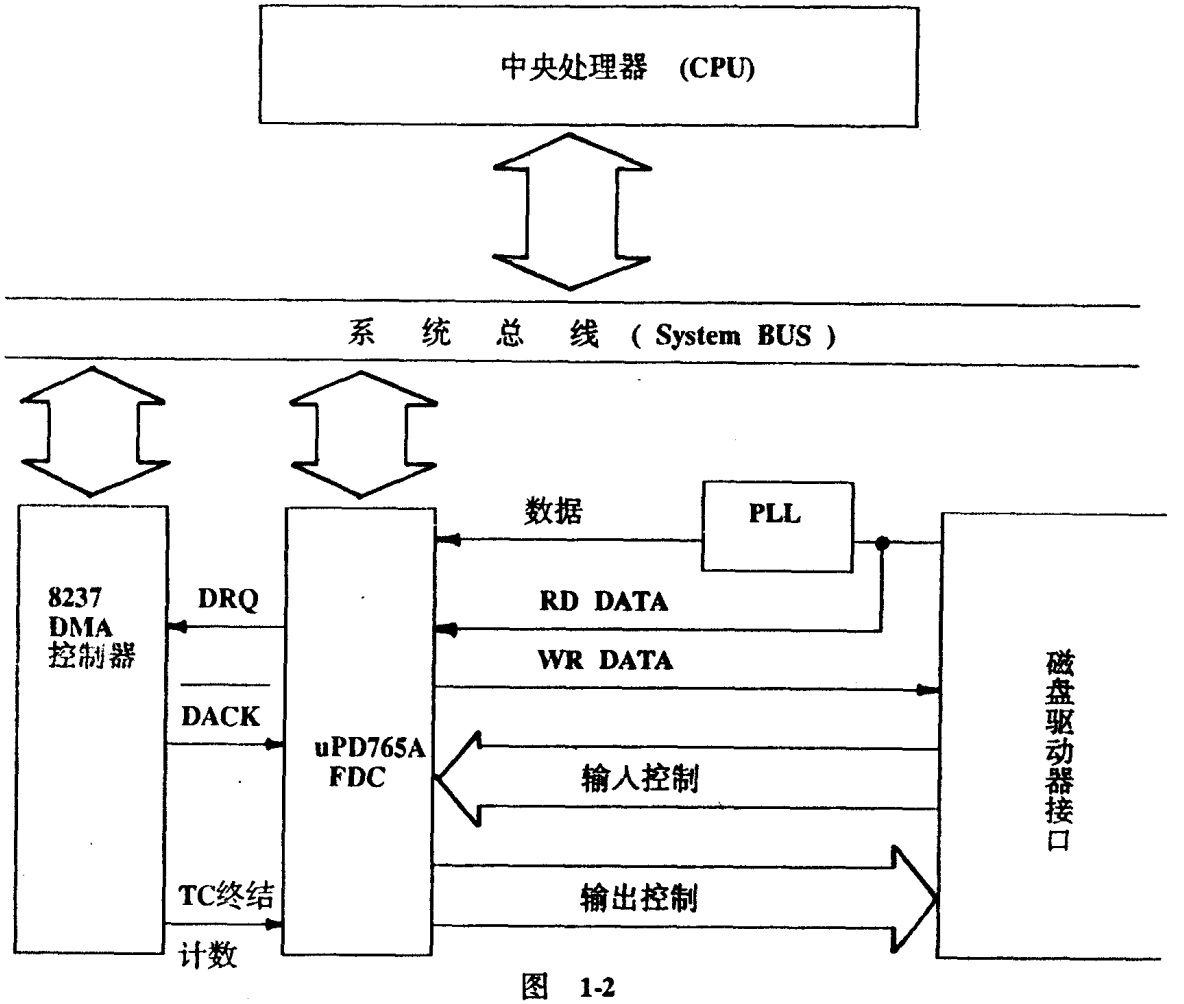

图 1-2

uPD765A 的 FDC 有两种传输方式，一种是 DMA 方式，另一种非 DMA 方式。在非 DMA 方式时，每次在 CPU 和 uPD765A 之间传输数据时，FDC 会对 CPU 产生中断。在 DMA 方式时，CPU 只是将一条指令送入 FDC，所有的数据传输将在 uPD765A 和 DMA 控制器的控制之下完成。

在后面的章节中，将会更深入地讨论驱动器的工作原理。

图 1-2 是驱动器与 CPU 的连接框图。

第二章　BIOS 与 DOS 的中断调用

§2.1 前言

8086/8088CPU 中提供有一个很好用的 INT 指令，它很类似于 CALL 指令，然而 INT 指令可指定一个中断类型(TYPE)，此类型号(X)与调用地址有 4*X 的关系,相当灵活而又有规律,因此我们可以很容易地指定并设计一个中断向量(interrupt vector)以提供一功能。BIOS 与 DOS 中提供有很多有用的功能,可以用 INT 指令来调用。

BIOS 是 BASIC I/O SYSTEM 的缩写,它是一段 8K 长的程序,固化在 ROM 之中,它占存贮器的实际地址是 FE000-FFFFF。

BIOS 中有许多控制外围设备的程序,例如 RS-232C、驱动器、显示器、键盘、录音机、计数器等等，有了这些程序，我们才有办法使用计算机，大多我们自己编写 BIOS 程序，说是自己编写，其实又不得不“参考”计算机生产厂家的 BIOS，因为生产厂家设计的 BIOS 功能确实很强，很难再写出比它功能更强、更好的程序来。为了要提高相容性，BIOS 的设计方式要尽量和厂家的一样，又不能丧失任何一项厂家所提供的功能，否则，某些程序势必不能得到应用。可是和 IBM-PC/XT 的 BIOS 太相近了，人家又说你是“抄”的，侵犯版权，于是，国内计算机厂商对于如何编写合法而又功能齐全的 BIOS 程序，感到很伤脑筋。

既然各厂家编写的 BIOS 程序与原厂的 IBM PC/XT 机不同，又如何能做到与 PC/XT 计算机兼容呢？原来美国有很多软件制造厂出品的软件程序，并不能调用什么计算机的 BIOS 程序，程序所需用的外围设备控制程序完全由自己编写的，这样的程序当然就可以应用在任何 PC/XT 相容的计算机上了。

若所有的外围设备控制程序都要自己编写，而不用现成的 BIOS 程序的话，显然要花掉很多时间和不少心血。幸好世界上有一家独霸的软件公司，设计了一套十分强劲的磁盘操作系统，这就是众所周知的 MS-DOS，这套操作系统目前所有的计算机系统均已采用，而且是必备的系统软件之一。在 MS-DOS 内提供了较 BIOS 更为丰富的外围设备控制程序，只要设计程序时利用 MS-DOS 中的外围设备控制程序来控制 I/O 操作，就不要用到任何 BIOS 程序。那么，只要能使用 MS-DOS 的计算机系统，就能使用这个程序，而不必管 BIOS 是否为原厂家的了。

IBM 原厂所出售的 MS-DOS，已把其中的部分功能放到了 BIOS 中，故一定要原厂的 BIOS 才能使用 MS-DOS。不过你仍可向 Microsoft 公司购买完整 MS-DOS 的使用权，以用在你的“兼容性”计算机上。

笔者在此特别将 MS-DOS 及 BIOS 中这些有用的中断调用一一列出，不管你是要做拷贝、保护还阅读程序，这些调用都会经常遇到的。当然你在设计程序时，也可利用这些现成的功能，来加强你设计的程序的功能。

§2.2 BIOS 中断调用

8088 所提供的中断类型号(type)有 256 种，即 0H-FFH，其中 0H--1FH 的类型号是提供给 BIOS 使用，20H--3FH 的类型号提供给 MS-DOS 使用(目前实际只使用 20H--27H)。