责任编辑　冯建华
封面设计　王　多
技术设计　刘忠凤

卢开澄　郭宝安　戴一奇　黄连生　编著
计算机系统安全

重庆出版社出版、发行(重庆长江二路205号)
新华书店经销　陕西安康天宝印务公司印刷
*
开本 787×1092　1/16　印张 20　插页 7　字数 447 千
1999 年 5 月第一版　1999 年 5 月第一版第一次印刷
印数:1－3,000
*
ISBN 7－5366－4083－8/TP·42
定价：38.00 元

重庆出版社科学学术著作出版基金资助

第一批书目

蜱螨学	李隆术　李云瑞　著
变形体非协调理论	郭仲衡　梁浩云　编著
胶东金矿成因矿物学与找矿	陈光远　邵　伟　孙岱生　著
中国天牛幼虫	蒋书楠　著
中国近代工业史	祝慈寿　著
自动化系统设计系统学	王永初　任秀珍　著
宏观控制论	牟以石　著
法学变革论	文正邦等　著

第二批书目

中国自然科学的现状与未来	全国基础性研究状况调研组 中国科学院科技政策局
中国水生杂草	刁正俗　著
中国细颚姬蜂属志	汤玉清　著
同伦方法引论	王则柯　高堂安　著
宇宙线环境研究	虞震东　著
头位难产	凌萝达　顾美礼　主编
中国现代工业史	祝慈寿　著
中国古代经济史	余也非　著
劳动价值的动态定量研究	吴鸿城　著
社会主义经济增长理论	吴光辉等　著
中国明代新闻传播史	尹韵公　著
现代语言学研究——理论、方法与事实	陈　平　著
艺术教育学	魏传义　主编
儿童文艺心理学	姚全兴　著

从方法论看教育学的发展　　毛祖桓　著

第三批书目

奇异摄动问题数值方法引论　　苏煜城　吴启光　著
结构振动分析的矩阵摄动理论　　陈塑寰　著
中国古代气象史稿　　谢世俊　著
临床水、电解质及酸碱平衡　　江正辉　主编
历代蜀词全辑　　李　谊　辑校
中国企业运行的法律机制　　顾培东　著
法西斯新论　　朱庭光　主编
《易》与人类思维　　张祥平　著

第四批书目

计算流体力学　　陈材侃　著
中国北方晚更新世环境　　郑洪汉等　著
质点几何学　　莫绍揆　著
城市昆虫学　　蒋书楠　主编
马克思主义哲学与现时代　　李景源　主编
马克思主义的经济理论与中国社会主义　　项启源　主编
科学社会主义在中国　　李凤鸣　张海山　主编
马克思主义历史观与中华文明　　王戎笙　主编
莎士比亚绪论——兼及中国莎学　　王佐良　著
中国现代诗学　　吕　进　著
汉语语源学　　任继昉　著
中国神话的思维结构　　邓启耀　著

第五批书目

重磁异常波谱分析原理及应用　　刘祥重　著
烧伤病理学　　陈意生等　著
寄生虫病临床免疫学　　刘约翰　主编
国民革命史　　黄修荣　著
现代国防论　　王普丰　王增铨　主编
中国农村经济法制研究　　种明钊　主编
走向21世纪的中国法学　　文正邦　主编
复杂巨系统研究方法论　　顾凯平等　著

辽金元教育史	程方平　著
中国原始艺术精神	张晓凌　著
中国悬棺葬	陈明芳　著

第六批书目

非线性量子力学	庞小峰　著
线型结构动力学中的模态理论	渚得超　著
薄壳理论	薛大为　著
胆道流变力学	吴云鹏　主编
乙型肝炎的发病机理及临床	张定凤　陶其敏等　著
中国蚜小蜂科分类	黄　建　著
中国历史时期植物与动物变迁研究	文焕然等　著
紫色土的肥力与增产途径	侯光炯　著
中国新闻学术发展史	徐培汀　著
列宁哲学思想的历史命运	张翼星　著
唐高僧义净及其著作论考	王邦维　著
中国远征军史	时广东　冀伯祥　著
中国民间美术史	王朝闻　主编
历代蜀词全辑续编	李　谊　辑校

第七批书目

亚夸克理论	焦善庆　蓝其开　著
肝癌	江正辉　黄志强　主编
面向对象的程序设计方法	郑纬民　田金兰　著
多媒体应用的开发技术	徐光祐　主编
计算机系统安全	卢开澄等　编著
声韵语源字典	齐冲天　著
幼儿文学概论	张美妮　著
黄河上游地区历史与文物	半一之　主编
论公私财产的功能互补	忠　东　著

第八批书目

高等植物发育分子生物学	吴乃虎　主编
长江三峡库区昆虫	杨星科　主编

小波分析与信号处理 ——理论、应用及软件实现	李建平　主编
世界首例独立碲矿床的 成矿机理及成矿模式	银剑钊　著
现代国防质·量·度研究	郑开昭　著
内分泌外科学	朱　预　主编
当代社会主义的若干问题 ——国际社会主义的历史 经验和中国特色社会主义	江　流　徐崇温　主编
科技生产力：理论与运作	刘大椿　主编
世界语言词典	黄长著　著

第 8 章 软件保护技术

8.1 概　述

随着计算机应用的不断普及，计算机软件得到了空前迅猛的发展，在一套完整的计算机系统中已经占据了相当重要的地位。计算机用户在选择计算机时，不但要注意它的硬件性能，如主频、内存等；还必须了解其是否具有完善的软件支持，否则的话，运算速度再快的计算机也只是一堆“废铁”。目前，软件的开发规模越来越大，成本也越来越高，它是诸多软件编程人员集体智慧的结晶，其市场价值就应当受到人们的认可。但是，从另一方面讲，计算机软件又具有易被复制的特点，因此就有一些不法商人通过贩卖盗版软件牟取暴利，从而严重损害了软件开发者的权益。鉴于这种状况，软件开发者不得不在软件开发的同时，注意考虑如何对软件进行保护，从而防止他人复制和非法使用。当然，在高额利润的驱使下，必然会有人“反其道而行之”，去破解软件保护，从而造成了目前软件加密与解密之间的激烈对抗。

软件加密共分两大部分：一是制做反拷贝的指纹，二是设计反跟踪程序，两者缺一不可。如果仅仅将加密技术理解为反拷贝技术或反跟踪技术都未免失之偏颇。从技术上讲，这两部分内容基本上是相互独立的。但从整体上讲，两者之间又紧密相关，何种反拷贝技术与何种反跟踪技术相结合才能得到最好的加密效果是一个很复杂的问题。有的程序尽管在某一方面做得很好，但另一方面考虑不周，仍然有可能给解密者以可乘之机。本章除了分别介绍两部分内容之外，还会多次谈到它们的联系之处。

8.2 加密方法的分类

软件加密方法各式各样，从一个侧面很难将其清楚地分类，本文仅从软件的使用方法上将其简单地加以分类。

一、密码加密

密码加密指的是用户在安装或运行软件时，必须输入一正确的密码或序列号。如果输入错误，则程序一般会拒绝继续执行，给出密码错误的信息，然后退出或请求继续输入等。

软件需要的密码一般从购买软件时附带的资料上查到。这种软件通常没有什么特别的防拷贝手段，否则的话软件完全可以去检测防拷贝信息，而不用采取密码输入的方法了。因此，这种软件允许人们任意备份，这是它的一个优点，但也是其不安全的地方。因为，人们只要将提供的资料，如密码表等复印一下，即可制造出另外一套可用的软件。

为了解决这个问题，人们在此基础上也做了少许改进。如采用白底黄字来制造密码表，这种表复印出来将是一张白纸；但人们仍可通过将密码表抄写下来即可达到复制的目的。又如，人们将密码输入数字或字母的方式加以改变。密码表是一张划分了很多区域并填上不同颜色的图，输入时屏幕上显示一张相同但未填颜色的图，由用户输入指定几个区域的颜色代号。由于彩色复印机尚不普及，因此这种密码表也很难于复印。但这些方法并不能很好地保护软件。

由于难以解决好密码表被复制的不利因素，这类软件在反跟踪技术上也做得比较简单。所以，这类软件也易于被制成解密版，即在输入密码时可以随便输入都会继续执行下去。

密码加密并不是一种很有效的加密方法，它被大量用在一些游戏软件上，这些软件一般都价格便宜，有大量的用户。加密的目的只是提醒大家应当去使用和购买正版软件，却很难阻止盗版的发生。

二、软盘加密

软盘加密曾是使用得最为广泛的一种加密方法。人们在运行程序时，要在程序提示下将加密盘(KEY DISK)插入软盘驱动器，待软件确认是正确的盘后才继续执行下去。

加密盘是购买软件时获得的。它是一种作上了特殊记号的软盘，只有相应的软件才能识别这个标记。这个标记不能按一般的 COPY 或 DISKCOPY 命令复制下来，它起到了类似人的“指纹”的作用。这种加密方法中最重要的就是如何制造出这种标记，这种技术也称为反拷贝技术。

除了能够制造出反拷贝的标记，在程序中还必须有验证这个标记的代码；这段代码一但被解密者发现，程序也就很容易被破解了。因此，必须采取一些编程技巧来阻止他人找到这段代码，这就是所谓的反跟踪技术，它与反拷贝技术有同等重要的地位。

使用磁盘加密方法与密码加密相比，最大的优点就是提供了“防拷贝”的硬件介质——钥匙盘。因为，使用钥匙盘输入密码，既隐蔽又方便了用户。而且，加密者在磁盘中可以储存较多的信息，包括程序的代码或数据，这样就可以使那些没有钥匙盘的用户破译程序的希望化为泡影。在使用过程中，人们也发现了使用磁盘加密的一些缺陷，例如兼容性问题。由于反拷贝技术或多或少都采用了与标准操作不同的地方，因此很容易产生不兼容的情况，使得用户很难放心地买到能在自己机器上正常使用的软件。另外，一个重要的问题则是寿命问题，它主要取决于钥匙盘的寿命。由于每次启动软件都要读取软盘，而且读取的位置往往又集中在某一磁道上，因此很容易将软盘磨损。同时，钥匙盘又防止了用户的备份工作，所以用户使用起来未免有些担心吊胆。

尽管如此，磁盘加密作为一种有效而且成本低廉的加密方法，仍然获得了很广泛的应用。人们对这类方法的研究已经非常深入了，提出了很多的方法。本章将会简要介绍其中的几种方法和各自的特色。

三、“软件狗”加密

“软件狗”是近些年来发展极为迅速的一种加密方法，大有取代磁盘加密的势头。“软件狗”本身就是一个插在并行口上的硬件电路，它对计算机通过并口发来的信息进行响应和处理。应用软件通过识别它的存在与否或利用它来进行一些数据变换而达到保护软件的目的。

与磁盘加密相比，这种加密方法有几大优点：

(1)速度快，适宜多次查询　这一点可以使得软件多次或定时地查询“软件狗”。而软盘加密通常都只在程序进入时查询一次，但可以使得在多台机器上同时使用一份软件。

(2)使用方便　使用时没有明显动作，使用户完全查觉不到它的存在，这给使用带来了方便。

(3)使用寿命长　正常使用下不必担心“软件狗”像加密盘一样容易出错，而且用户可以随意备份其它系统软盘，免去对磁盘寿命的担心。

(4)兼容性较好　由于“软件狗”与主机的通讯遵从并行口的标准，因此一般没有兼容性上的问题。而磁盘加密通常要使用一些非标准或不稳定的东西，可靠性经常出现问题，出现了大量困扰加密者的兼容性问题。

由于“软件狗”克服了磁盘加密的很多缺陷，才得到了越来越广泛的应用。但它本身也有一定的缺陷。由于每种软件都有各自的“软件狗”，因此，若要同时使用就必须在并行口上串接多个“软件狗”，这会使用户感觉使用起来很麻烦。而且，由于各种“软件狗”之间的电路各不相同，很有可能造成彼此不兼容的情况。

四、其它加密方法

除了以上三种主要的加密方法之外，还有一些其它的方法，如扩展卡加密。这是一种将硬件电路移到板上去做的加密方法，它的硬件电路可以比“软件狗”做得复杂得多。因此，安全性也较高。但是，若单独做成一块加密卡，既要占据一个扩展槽，安装起来又不方便，故难以广泛使用。一般都是与需要专用硬卡的软件(排版软件、多媒体软件等)做在一起，达到合二为一的目的。硬卡电路的设计要涉及较多的硬件知识，使用也有较大的局限性，在本文中就不再多作讨论了。

上面的一些方法都是迄今为止经常见到的。由于软件加密的迫切性，新的技术还在不断地涌现出来，只有不断地学习、借鉴，才能制造出安全可靠的加密软件，保障自己的合法权益。

最后还想强调一点，软件反拷贝不单是技术问题，还牵涉到法律制约和心理作用等因素必须考虑，现举一个密码技术在反拷贝中的应用例子。例如：A 购得软件，为防止扩散，可将 A 的身份 $ID(A)$加密成密文 $C = E_k(ID(A))$，C 用二进制码表示，用它与执行程序的代码的关键部分作⊕运算。经过这样处理后的软件是不能正确运行的，所以也不担心被拷贝。用户 A 要使用该软件必须通过身份鉴别，将 $ID(A)$加密得到 $C = E_k(ID(A))$的二进制数，在执行程序的指定地点作⊕运算以恢复运行程序的正确性。可考虑在运行前屏幕上显示 A 的身份 $ID(A)$等办法。由于担心身份随软件的扩散而扩散，达到防止被拷贝。

8.3 指纹法

前面已经讲过，在磁盘加密方法中，最重要的就是制作防拷贝的“钥匙盘”，这些盘上所制作的特殊标记又被称为磁盘的“指纹”。在本节中，我们将介绍几种实用的“指纹”制作方法。

一、磁盘的基本结构

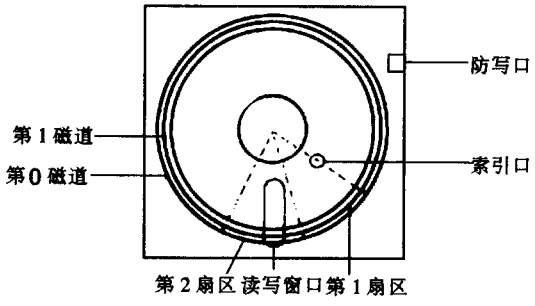

图 8.1

为了在磁盘上存储数据，就必须将磁盘格式化。每张磁盘在格式化后都被分为一个个同心圆，称为磁道。磁道的计数从外向内依次增加。在每个磁道上均匀地划分着数个存放数据的区域，称为扇区，扇区的大小可以调整(见图 8.1)。

对于不同类型的磁盘，相应的磁记录参数也有不同(见表 8.1)。

表 8.1　磁盘磁记录参数

参数 \ 磁盘类型	360KB	1.2MB	720KB	1.44KB
磁头数(H)	2	2	2	2
磁道数(T)	40	80	80	80
扇区数(S)	9	15	9	18
扇区大小(N)	2	2	2	2

其中，扇区大小表明了一个扇区所能存储数据的多少，一般是这样计算的，字节数 $n = 128 \times 2^{n}$。磁盘中的磁头、磁道都是由机械装置定位的，那么它又如何识别同一磁道内的各个扇区呢?这就需要有同步信息与标识信息来指明(见图 8.2)。

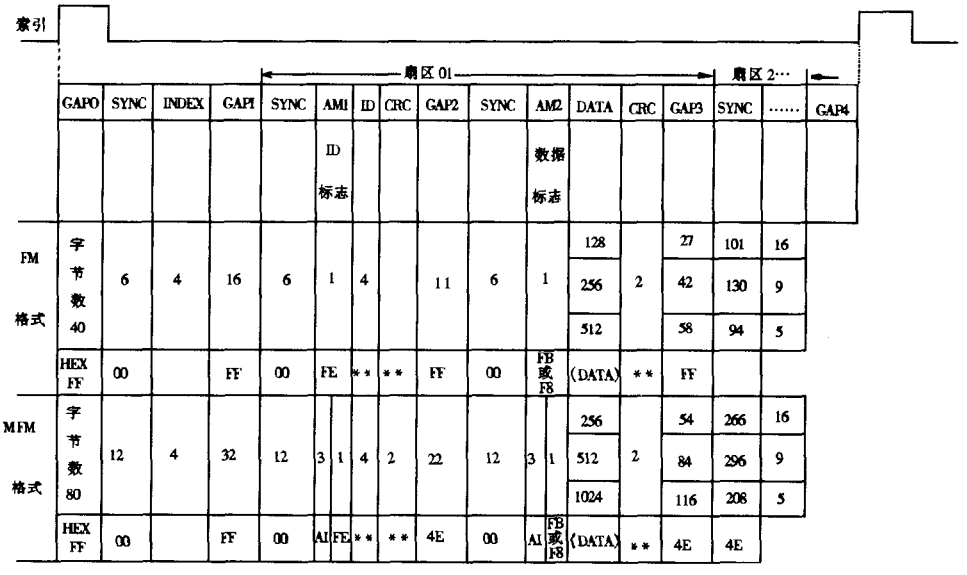

图 8.2

其中一些非数据区的含义是这样的：

1. 前置区

前置区是由图 8.2 中的 GAPO、SYNC、AM1 和 GAP1 组成的，其作用是为了弥补各磁盘机之间转速的误差而增加的一个缓冲区。

2. 识别区

识别区由同步字段 SYNC、地址字段 ID、CRC 和 GAP2 组成。这个区中最重要的字

段是 ID，它包括紧接着扇区的一些重要信息，即扇区所在的磁道号(Track)、扇区所在的磁头号(Head)、相应的扇区号(Sector)和扇区大小的标记 N。只有当磁盘所需读写的扇区的实际参数与 ID 中的数完全吻合时，才认为找到了相应的扇区，否则就会发出“扇区未找到”的错误信息。大多数的加密方法都要通过将此区域的 ID 值改为非标准的数据来完成防止拷贝的目的。

3. 数据区

这个区由 SYNC、DATA、AM2、DATA 和 CRC 及 GAP3 字段组成，它是用来存储数据的。

4. 后置区

后置区就是图 8.2 中的 GAP4，它位于磁道上最后一个扇区之后，一直延续到索引孔的位置，其作用也是为了适应磁盘机转速的不同而设计的。对于转速快的机器，格式化后的磁道，GAP4 区域就较短，反之则较长。

后置区长度的长短是在格式化时确定的，而且与磁盘机有关。这一特点也就成了制作软盘“指纹”的突破口，利用这个区作指纹的方法通常称为磁道接缝法。这种防拷贝方法抗攻击强度很高，可以阻止各种软件拷贝工具和硬件设备的成功拷贝。

格式化过程在填写这个区域的值时，都会一致地填入 4EH，填写的过程直到遇到索引孔为止。但是，这时不能保证最后一个字节的完全写入。因此，如果以后读出 GAP4 数据时，在索引孔位置处就会发生畸变，而且会影响后面数据的读出，这一点在 **8.3. 五**节再详细讨论。

二、磁盘的基本功能调用

为了制作加密盘，必须了解如何对磁盘进行读写。一般情况下，大多采用 BIOS 的 INT 13H 中断调用；既简单易学，又能保证较高的防拷贝强度。下面依次介绍 INT 13H 中最基本的 6 条中断调用。

1. 磁盘系统复位功能

入口参数：AH = 0, DL = 驱动器号(0 = A, 1 = B…)

出口参数：无

```
例：    MOV     AH，0       ；置功能号
        MOV     DL，0       ；选择 A 驱动器
        INT     13H
```

此功能是当磁盘读写出错时，为了再次调用该功能，且应当首先调用此功能使系统重置。使用此功能时，磁盘机并不做任何操作，而只是在系统中设置一标记。当启动读写功能时，一但发现此标记，磁盘机会首先将磁头拉到最外面(软盘边缘)用以校准第 0 磁道，然后执行相应的读写功能，这是常用到的一个功能。

2. 读取磁盘机状态功能

入口参数：AH = 1，DL = 驱动器号(0 = A, 1 = B…)

出口参数：AL = 磁盘状态

AH = 0

例：
```
MOV     AH, 1       ; 置功能号
MOV     DL, 0       ; 选择 A 驱动器
INT     13H
```

此功能获取上一次读、写等操作后磁盘机的状态，它可以使得把错误处理或错误报告写成完全独立于磁盘操作的部分。但对于一般软件保护的制作者来说，使用得并不太多。

3. 读扇区数据功能

入口参数：AH = 2
AL = 要读的扇区个数
CH = 磁道号
CL = 起始扇区号
DH = 磁头号
DL = 驱动器号(0 = A，1 = B…)
ES：BX = 读出数据的存放地址

出口参数：成功时：进位标志 CF = 0，AH = 0
失败时：进位标志 CF = 1，AH = 错误代码

例：
```
MOV     AX, 201H        ; 读取功能，长度为 1 扇区
LES     BX, BUFFER      ; 设数据存放地址
MOV     CX, 1           ; 读 0 磁道 1 扇区
MOV     DX, 0           ; 选择 A 驱动器 0 磁头。
INT     13H
```

这段代码将 A 盘 0 面 0 磁道 1 扇区的内容读到用户指的缓冲区 BUFFER 中。读扇区功能是验证磁盘“指纹”最重要的功能。程序通过读取结果或失败时的错误代码来判定磁盘的合法性。当用此功能读取刚刚换上的软盘时，总会出现 AH = 6(磁盘被更换)的错误代码。因此，在使用时必须注意在失败后要重试一下，但不要忘记必须先使用 0 号功能。

4. 写扇区数据功能

入口参数：AH = 3
AL = 写入的扇区总数
CH = 磁道号
CL = 扇区号
DH = 磁头号
DL = 驱动器号(0 = A，1 = B…)
ES: BX = 被写入数据的存放地址

出口参数：成功时：CF = 0，AH = 0
失败时：CF = 1，AL = 错误代码

例：
```
MOV     AX, 301H        ; 写入功能，长度 1 扇区
```

```
LES     BX, BUFFER    ; 写入 BUFFER 中内容
MOV     CX, 1         ; 写入 0 磁道 1 扇区
MOV     DX, 0         ; 选择 A 驱动器 0 磁头
INT     13H
```

这段程序将用户指定的缓冲区 BUFFER 中的内容写到 A 盘 0 磁头 0 磁道 1 扇区中。此功能与 2 号读功能是相对应的，使用起来除功能号以外完全相同。这自然也存在上面提到的重试问题。但由于钥匙盘一段都是写保护的，所以在验证磁盘的时候，很少用到这个功能。

5. 验证扇区功能

入口参数：AH = 4

AL = 要验证的扇区数

CH = 磁道号

CL = 扇区号

DH = 磁头号

DL = 驱动器号(0 = A，1 = B…)

ES: BX = 扇区标识字段的起始地址

出口参数：成功时：CF = 0，AH = 0

失败时：CF = 1，AH = 错误代码

例：

```
MOV     AX, 401H      ; 验证一个扇区
LES     BX, IDBUF     ; 扇区 ID 的存放地址
MOV     CX, 1         ; 验证 0 磁道 1 扇区
MOV     DX, 0         ; 选择 A 驱动器磁头
INT     13H
```

这段代码验证 A 盘 0 面 0 磁道 1 扇区是否是正确的，数据是否可正确读出。此功能与 2 号功能类似，但并不需要缓冲区来存放读出的数据，可以通过结果来判断扇区的好坏。此功能使用得也比较少。

6. 格式化磁道功能

入口参数：AH = 5

AL = (一般要设置为 1)

CH = 磁道号

DH = 磁头号

DL = 驱动器号(0 = A，1 = B…)

ES: BX = 存放扇区标识 ID 的地址

出口参数：成功时：CF = 0，AH = 0

失败时：CF = 1，AH = 错误代码

例：

```
ID      DB    0, 0, 1, 2
        DB    0, 0, 2, 2
        DB    0, 0, 3, 2
```

```
        DB    0, 0, 4, 2
        DB    0, 0, 5, 2
        DB    0, 0, 6, 2
        DB    0, 0, 7, 2
        DB    0, 0, 8, 2
        DB    0, 0, 9, 2
MOV     AX, 501H              ; 格式化一个磁道
LES     BX, ID                ; 磁道标识在 ID 中
MOV     CX, 0                 ; 格式化第 0 磁道
MOV     DX, 0                 ; 选择 A 驱动器 0 磁头
INT     13H
```

此段代码按 360KB 软盘的格式，格式化 A 盘 0 面的第 0 磁道。INT 13H 的格式化功能是制作磁盘“指纹”的核心部分。大多数的“指纹”制作技巧都要用到这一功能。

格式化的主要功能就是将磁道上的数据组织成前面介绍的磁道结构的形式，包括向前置区、识别区、数据区和后置区填写内容。而其它功能只会造成对数据区的写操作，其它的区域只作读出用。当格式化命令填写扇区的 ID 信息时，会将用户提供的 ID 缓冲区中的数据不加验证地填入，这就使得可向其中写入一些非标准的数据并以此来制作磁盘“指纹”。下面将具体介绍其用法。

为了让磁盘操作处理更多的情况，INT 1EH 起到了补充的作用。这个中断向量并不指向一段中断程序，而是软盘驱动器使用的一张参数表，它由 11 项组成(见表 8.2)。

表 8.2 软盘驱动器使用参数

字节偏移	含义	值
0	高 4 位为磁头步进速率,低 4 位为卸载时间	DFH
1	高 7 位为磁头加载时间,低 1 位非 DMA 方式	02H
2	马达等待时间(延迟关闭)	25H
3	每扇区字节数	02H
4	每磁道扇区数	09H
5	扇区间的间隔长度	2AH
6	每扇区字节数(当偏移 3 为 0 时)	FFH
7	格式化时扇区间隔的填充字节	54H
8	格式化时扇区数据的填充字节	F6H
9	寻磁道后磁头的稳定时间	01H
0AH	命令等待时间	08H

表 8.2 与磁盘机的操作是紧密相关的，它定义了软盘在读、写、格式化时的各种参

数，如扇区字节数、每磁道扇区数等以及磁盘机运转时的一些机械指标，如磁头步进速率和马达等待时间等。通过对表 8.2 的修改，可以对磁盘操作做更加精细的控制并制作出高强度的反拷贝“指纹”。

上面这个参数表中，最重要的参数是偏移 3 和偏移 4 的值，即每扇区字节数和每磁道扇区数。在程序发出格式化命令后，整个磁道的格式就由这两个参数而定。当程序进行读、写操作时，参数 4 不再有用，这时由 CX、DX 及参数 3 构成的 T、H、S、N 四个值必须要与扇区前的地址标识中的 T、H、S、N 字段吻合才能正确读出数据。

三、制作额外磁道

正常的 1.2MB 软盘每面共有 80 个磁道(0～79)，过于靠近磁盘中心的磁道会由于周长过短，数据记录密度过大而造成数据的不稳定。但是，不论是从磁盘机的机械装置，还是 BIOS 软件，都允许格式化和读写这些磁道。实验表明，80 磁道、81 磁道的数据还是完全可以信赖的。对于 360KB 的软盘(每面 0～39 共 40 磁道)。人们曾经成功地格式化了磁盘的 47 磁道，已经达到了双密软盘 48TPI(Trac ks Per Inch)指标的极限。

要制作一个额外的磁道非常简单，请见代码 2.1：

本程序是在 1.2MB 密度磁盘的第 80 道上格式化的，读写方法与正常磁道类似。仅仅磁道号不同

```
code    segment    ' CODE'
        assume     cs: code, ds: code
        org        100h
start:
        jmp        begin
msg1    db         'Please Insert a DISK in drive A: ', 0ah, 0dh
        db         'Preass any key to continue. . . ', 0ah, 0dh, ' ¥'
msg2    db         'Format Success! ', 0ah, 0dh, '¥'
msg3    db         'Format Failed! ', 0ah, 0dh, '¥'
id      db         80, 0, 1, 2
        db         80, 0, 2, 2
        db         80, 0, 3, 2
        db         80, 0, 4, 2
        db         80, 0, 5, 2
        db         80, 0, 6, 2
        db         80, 0, 7, 2
        db         80, 0, 8, 2
        db         80, 0, 9, 2
        db         80, 0, 10, 2
        db         80, 0, 11 ,12
```

```
        db      80, 0, 12, 2
        db      80, 0, 13, 2
        db      80, 0, 14, 2
        db      80, 0, 15, 2
begin:
        mov     ah, 9
        lea     dx, msg1
        int     21h             ; 显示提示
                                ; 行字符串
        mov     ah, 7
        int     21h             ; 等待用户按键

        push    ds
        xor     ax, ax
        mov     ds, ax
        mov     si, 526h
        mov     byte ptr[si], 15
        pop     ds

        mov     cx, 5
loop1:
        push    cx
        mov     ax, 0501h
        lea     bx, id
        mov     ch, 80
        xor     dx, dx
        int     13h             ; 格式化磁道
        pop     cx
        jnc     succ            ; 成功则转 SUCC
        xor     ax, ax
        int     13h             ; 磁盘复位
        loop    loop1

        mov     ah, 9
        lea     dx, msg3
        int     21h             ; 显示格式化失败
        jmp     return
succ:
```