编号：_____

实验	一	二	三	四	五	六	七	八	总评	教师签名
成绩										

武汉大学国家网络安全学院

课程实验(设计)报告

题　　目：软件安全实验

专业(班)：信安 2 班

学　　号：

姓　　名：

课程名称：软件安全实验

任课教师：彭国军

2020 年　月　日

目 录

实验 1　磁盘格式与数据恢复

1.1 实验名称

磁盘格式与数据恢复

1.2 实验目的

1) 了解磁盘的物理和逻辑结构
2) 熟悉 FAT32 文件系统
3) 学会使用磁盘编辑软件
4) 了解文件删除、格式化的基本原理
5) 能够利用工具或者手工恢复被删除的文件

1.3 实验步骤及内容

第一阶段：

- 熟悉 WinHex 的使用。
  - 熟悉磁盘工具的使用。
  - 利用 WinHex 查看物理磁盘和逻辑磁盘。
  - 了解 WinHex 中相关工具的用法。

第二阶段：

- 分析本地硬盘的主引导扇区
- 利用磁盘编辑工具查看 MBR 磁盘分区并分析：
  - 主引导扇区由哪些部分组成？
  - 四个主分区项的内容各代表什么？
  - 分析主扩展分区表的结构。
  - 通过分区项来确定每个本地逻辑盘的位置以及大小，并画出本地硬盘的逻辑结构。
    - 每个本地盘的开始扇区位置，总扇区数，结束扇区位置，各扩展分区表扇区位置，保留空间数量。
- 利用磁盘编辑工具查看 GPT 磁盘分区并分析
  - GPT 分区结构与 MBR 的具体差异有哪些？
  - 主分区头所在扇区包括哪些重要内容，验证这些重要内容的有效性。
  - 通过分区节点分析自己硬盘的各分区信息。
  -

第三阶段：

- 熟悉 FAT32 文件格式。

- 用 WinHex 打开某个 FAT32 分区格式的逻辑盘。
- 查看该逻辑盘的起始扇区，分析起始扇区中的相关字段（BPB:BIOS Parameter Block）。
- 查看 FAT1 和 FAT2 的内容和大小。
- 查看该逻辑盘的根目录区。
- 查看某个文件的目录项结构和 FAT 链以及具体存储位置。
    - 在根目录下建立文本文件：test-学号后 3 位.txt，其中填充 60K 左右的文本字符保存（注意：先行存储其他数据使得该文件的首簇高位不为 0）。
    - 查看该文件的目录项，对其进行分析，并得到该文件所在位置以及大小。
    - 查看首簇位置，并得到簇链表。通过簇链表查看该文件内容。
    - 记录首簇位置（14H-15H,1AH-1BH）和文件大小（1CH-1FH）。

第四阶段：

- 手工恢复被删除的文件
    - 删除前面所建立的文件。(del&shift+del)
    - 利用 WinHex 在该文件所在盘符查找该.txt 文件的目录项。
    - 查看目录项的变化。
    - 利用该残余目录项来计算被删除的文件所在的位置。
    - 手工恢复该文件（文件名、首簇高位、簇链表修复）。

课后习题思考：

- 在磁盘分区过程中，用户提供了哪些信息？分析分区工具的工作原理。
- 高级格式化与低级格式化的具体原理和区别是什么？
- 查找资料，对 NTFS 分区的总体结构进行分析，尝试对 NTFS 下删除的文件进行手工恢复。
- 用数据粉碎工具（如金山、360、Strongdisk 等）粉碎指定文件，分析其数据粉碎原理。
- 通过分区表看到的分区字节数为何与资源管理器中看到的分区字节数有差异？
- 如果删除的文件是长文件名，如何恢复所有文件名。

1.4 实验关键过程、数据及其分析

1.4.1 WinHex 的使用

以管理员模式启动 WinHex，在选项界面中选择要查看的磁盘，如图 1.1 所示

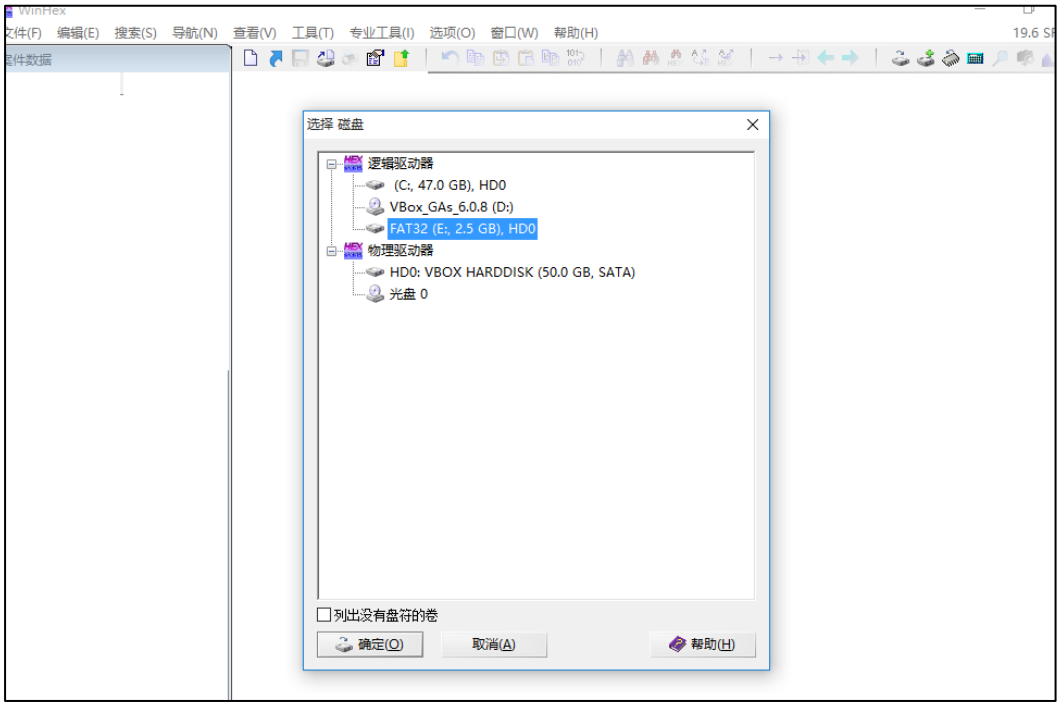


图 1.1 启动 WinHex

在 WinHex 操作界面中，选择自己要查看的逻辑盘和物理盘，确定即可打开相应的磁盘查看，如图 1.2 所示 ，左上角显示的是磁盘的文件目录，左下角显示的是磁盘的数据区数据信息。

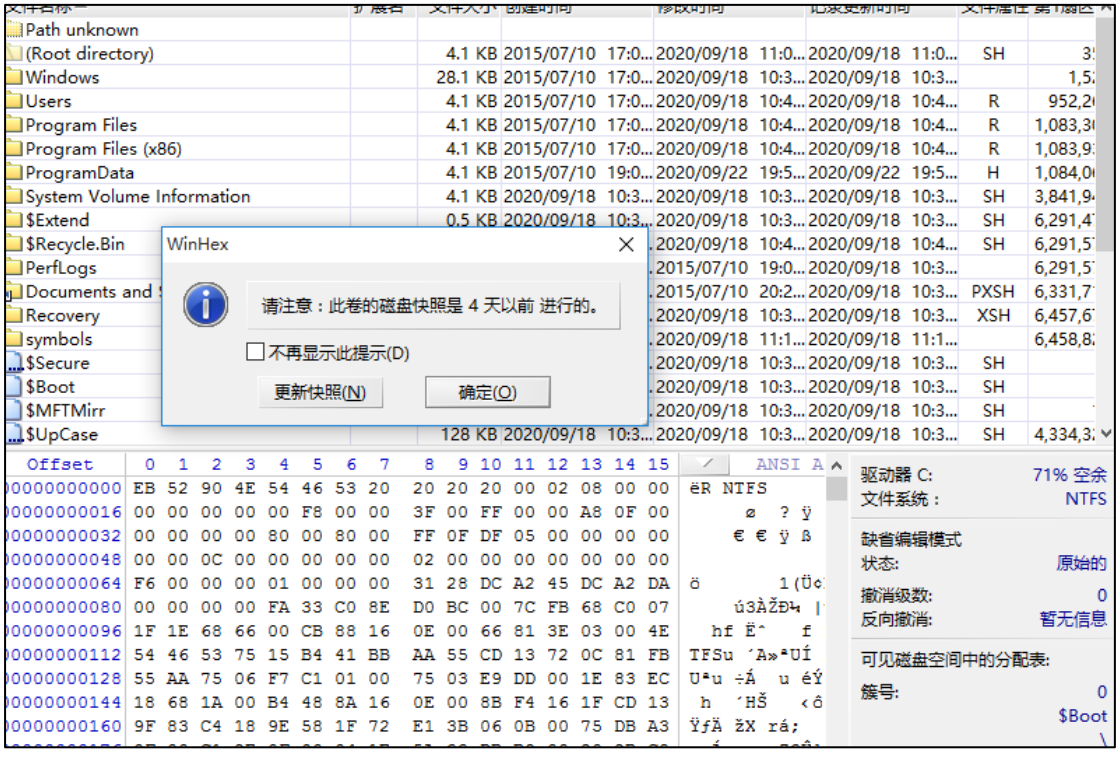


图 1.2 利用 WinHex 查看磁盘

由于磁盘快照生成较久，选择更新快照。

在 WinHex 中主要使用的工具是进入指定的偏移或地址。选择导航,可以选择相应的功能，如图 1.3 所示

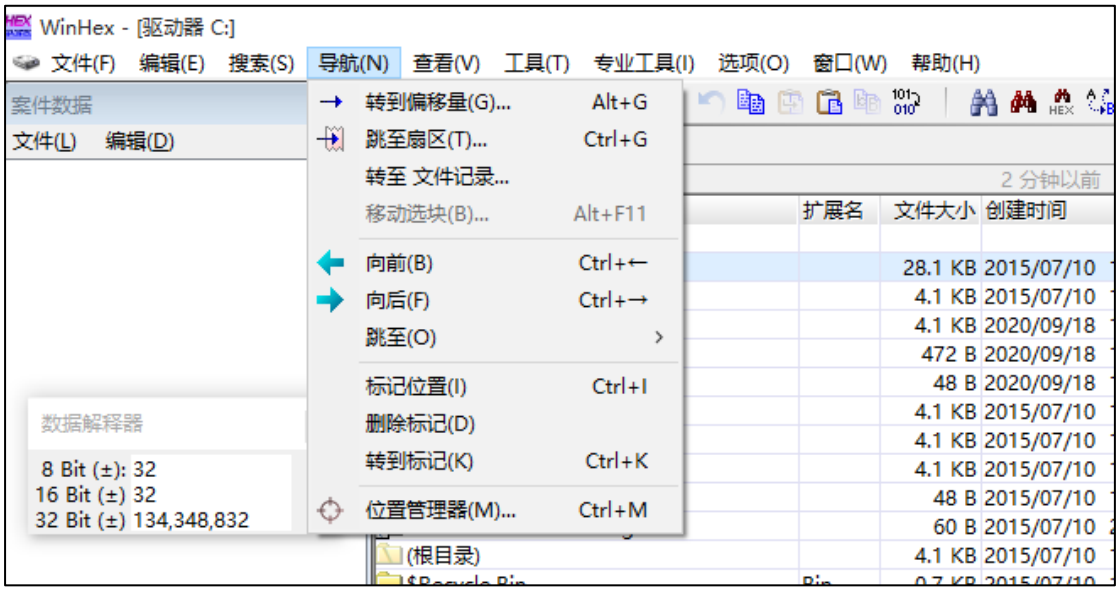


图 1.3 WinHex 导航工具的使用

转到偏移量可以选择基地址以及偏移量跳转到自己想要到达的地址;跳至扇区可以通过扇区为单位进行跳转。文件记录功能主要用于文件恢复过程中,利用其跳转到簇链表相应的位置。在对磁盘进行修改之后，由于 WinHex 没有更新磁盘信息，需要重新对磁盘进行快照，如图 1.4 所示。

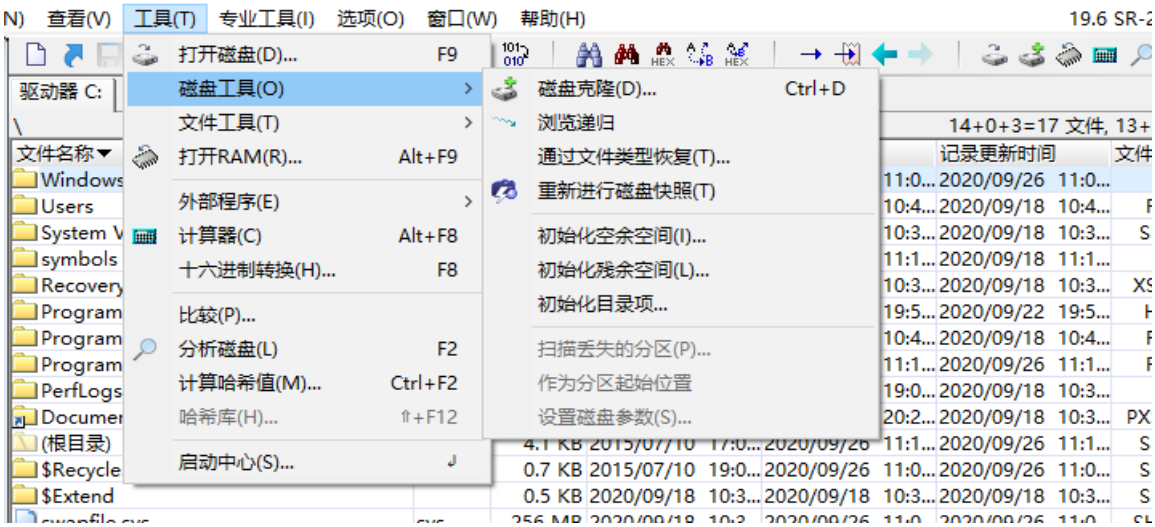


图 1.4 WinHex 重新快照

1.4.2 分析本地硬盘的主引导扇区【GPT＋MBR】

利用 WinHex 打开 MBR 分区形式的磁盘，如图 2.1 所示，从 00 开始的 512 个字节是硬盘的主引导扇区。

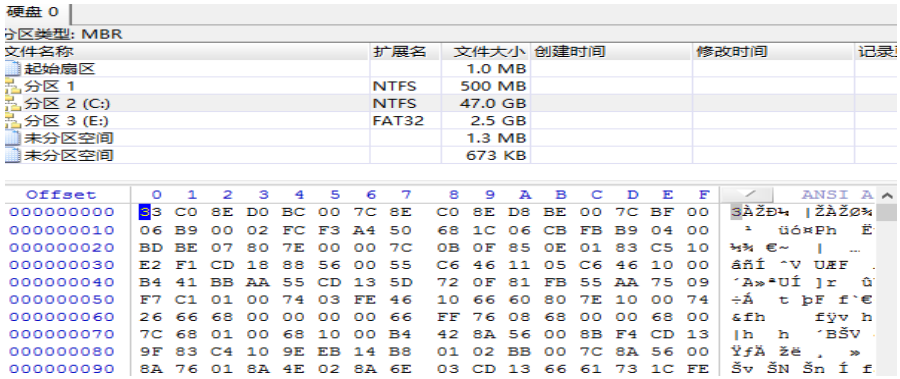


图 2.1 查看主引导扇区

主引导扇区是硬盘第一个扇区，它由主引导记录(MBR),硬盘主分区表(DPT)和引导扇区标记(ID)三部分组成。

主引导记录 MBR：从 0 至多占用的 0x1BDH 一共 446 个字节，存放系统主引导程序。

硬盘主分区表 DPT:占用 64 个字节，0x1BEH － 0x1FDH，它有四个分区项，每个项 16 个字节，最后 8=4+4 个字节存放相对扇区地址和该分区占用的扇区数量。

通过 WinHex 自带的分区表-模板，查看物理硬盘的结构，如图 2.2 所示

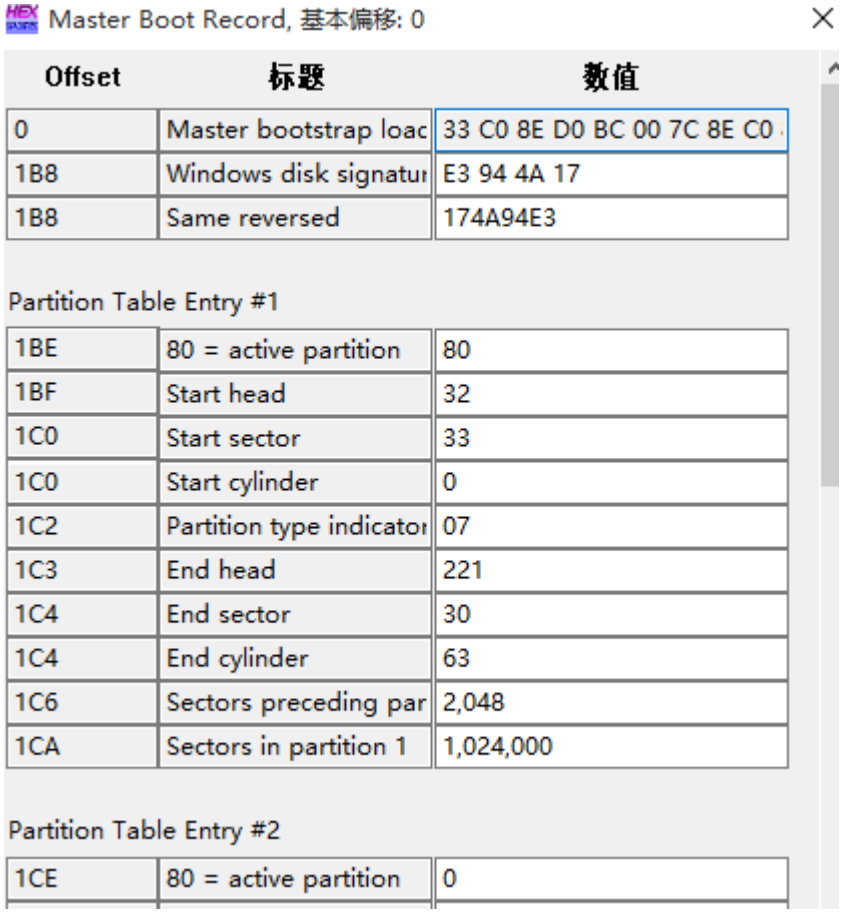


图 2.2 分区表模板查看

可以看到，前 1B8 个字节是其引导代码，从 1BE 开始是硬盘分区表，最后以 55AA 结尾。每一个分区表项的长度是 16 个字节。如图 2.3 所示。

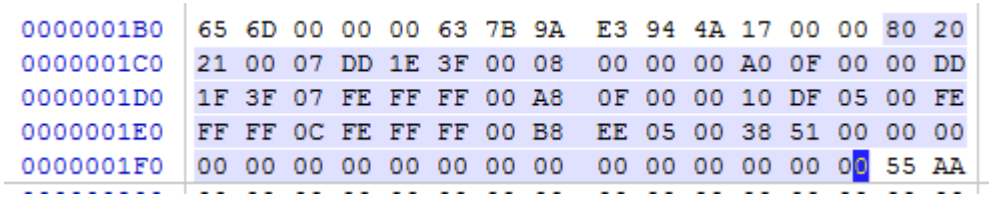


图 2.3 分区表地址数据

以第一个分区为例，分区表从 80 20 开始，到 0F 00 结束。00 0F A0 00 表示其占有的扇区数，00 00 08 00 代表其起始扇区(16 进制)。0x800=2048,这与分区信息是对应的，如图 2.4 所示。

文件名称	扩展名	文件大小	创建时间	修改时间	记录更新时间	文件属性	第1扇区
起始扇区		1.0 MB					0
分区 1	NTFS	500 MB					2,048
分区 2 (C:)	NTFS	47.0 GB					1,026,048
分区 3 (E:)	FAT32	2.5 GB					99,530,7...
未分区空间		1.3 MB					104,853,...
未分区空间		673 KB					104,856,...

图 2.4 分区第一扇区地址

下面根据 WinHex 中分区扇区数量计算分区大小并与 Windows 计算机磁盘管理进行验证。以 E 盘为例，根据图 2.3，其具有 0x00513800=5322752 个扇区，一个扇区 512 字节，因此具有 2661376KB=2599MB=2.53GB，与图 2.4 分区 E 是对应的。

扩展分区：主分区表只支持 4 个分区项目，当分区更多时无法满足，因此需要扩展分区表 EBR。要使用扩展分区，首先，主分区表中必须要有一个基本扩展分区项，用于指出所有扩展分区总体信息。所有的扩展逻辑盘全在这个基本扩展分区项指出的主扩展分区中。其结构如图 2.5 所示。

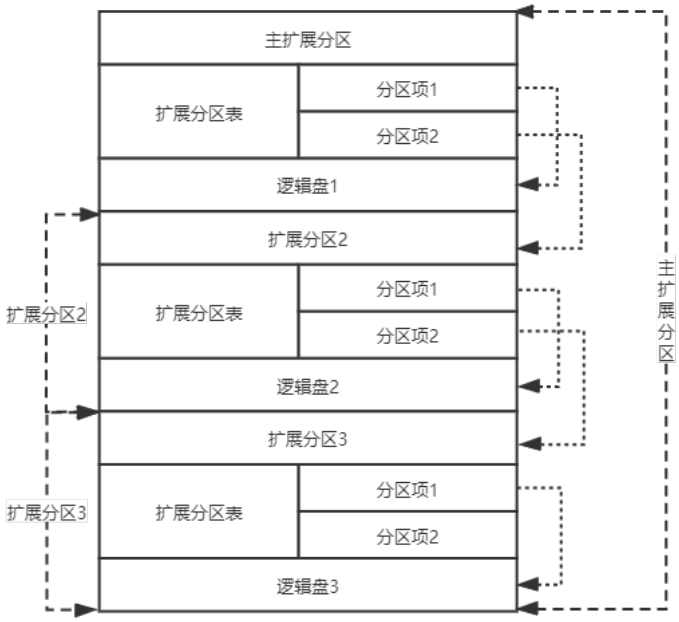


图 2.5 扩展分区和逻辑盘

查看 GPT 分区，首先添加一块硬盘，进入 cmd，输入命令 diskpart，显示挂载在主机上的磁

盘，进入磁盘转换。输入命令将其转换为 GPT 分区格式

图 2.6 转换磁盘分区格式

使用 WinHex 打开硬盘 1，如图 2.7 所示

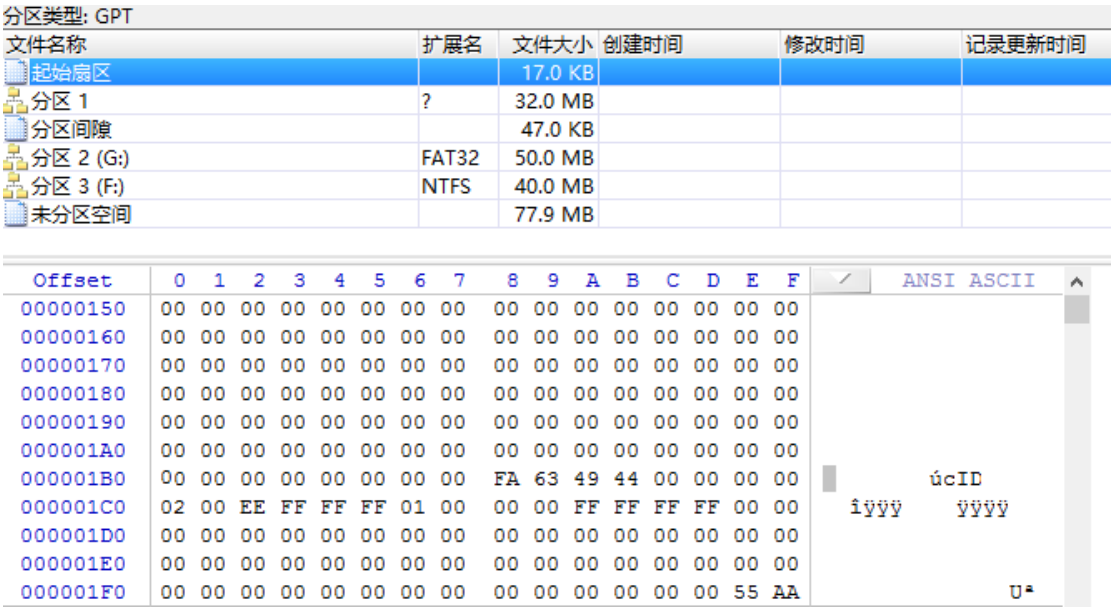


图 2.7 GPT 磁盘信息

利用 WinHex 自带的分区表模板，查看 GPT 磁盘的分区表信息，如图 2.8 所示

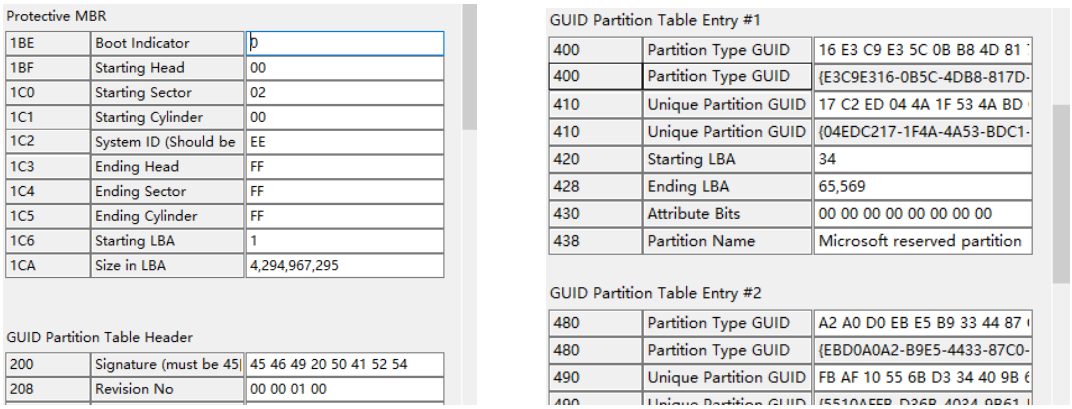


图 2.8 GPT 磁盘分区表

可以看到从 1BE-1CA 是磁盘保护性的主引导记录 MBR，从 0x200=512 开始是磁盘的分区表。从 0x400=1024 开始是 CPT 磁盘的第一个分区表项。
分区表项中存储了对应分区的起始 LBA，LBA 即逻辑地址块，是 GPT 分区形式的基本单位，每一个 LBA 的大小也是 512 个字节。
分区表项 1 中，起始 LBA 块是 34，终止 LBA 块是 65569，下面验证分区 1 的大小：
由分区表项信息，分区 1 一共有 65569-34+1=65536 个 LBA 块，因此有 32768KB=32MB，对比图 2.7 分区 1 信息，可以发现正确。

GPT 与 MBR 分区的具体差异：

1、MBR 分区表最多只能识别 2TB 左右的空间，大于 2TB 的容量将无法识别从而导致硬盘空间浪费；GPT 分区表则能够识别 2TB 以上的硬盘空间。

2、MBR 分区表最多只能支持 4 个主分区或三个主分区+1 个扩展分区(逻辑分区不限制)；GPT 分区表在 Windows 系统下可以支持 128 个主分区。

3、在 MBR 中，分区表的大小是固定的；在 GPT 分区表头中可自定义分区数量的最大值，也就是说 GPT 分区表的大小不是固定的。

1.4.3 FAT32 文件系统格式分析

用 WinHex 打开一个 FAT32 格式的逻辑盘，如图 3.1 所示

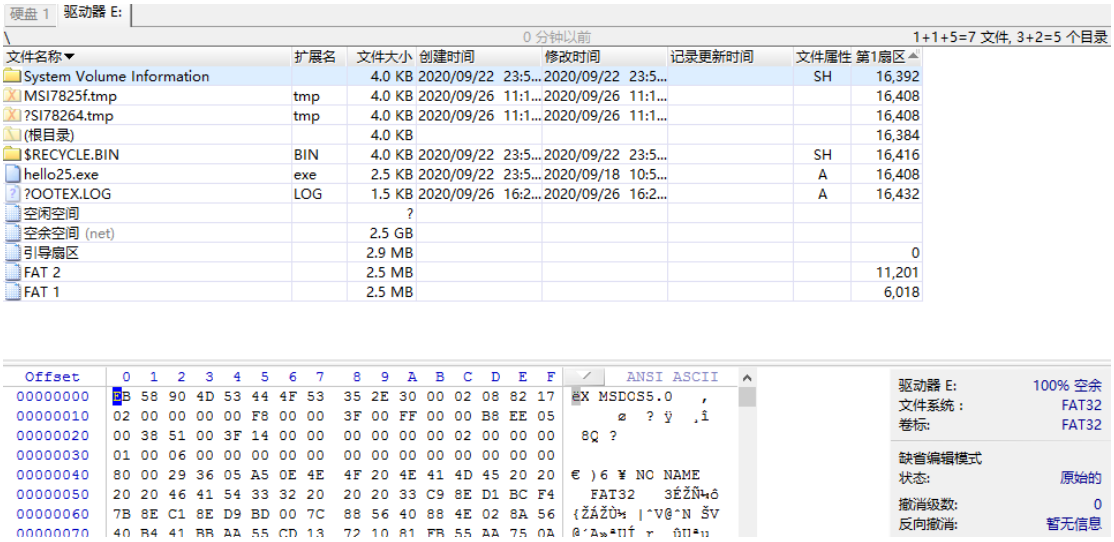


图 3.1 FAT32 磁盘

通过 WinHex 自带的引导扇区模板，查看该逻辑盘的起始扇区，分析起始扇区中的相关字段，如图 3.2 所示。

0-3 是其固有的字段，从 0xB 开始，到 0x20 是该逻辑盘对应的 BPB(BIOS Parameter Block，BIOS 参数块)字段。通过 BPB 可以查看这个磁盘的基本信息。如每个扇区的字节数，这里可以看到是 512 字节，每个簇有 8 个扇区。

FAT32 Selection 是该 FAT32 文件系统的一个表，从表中可以看到每一个 FAT 它的扇区数是 5183，最后在偏移量 1FE 处也是以 55AA 结尾。

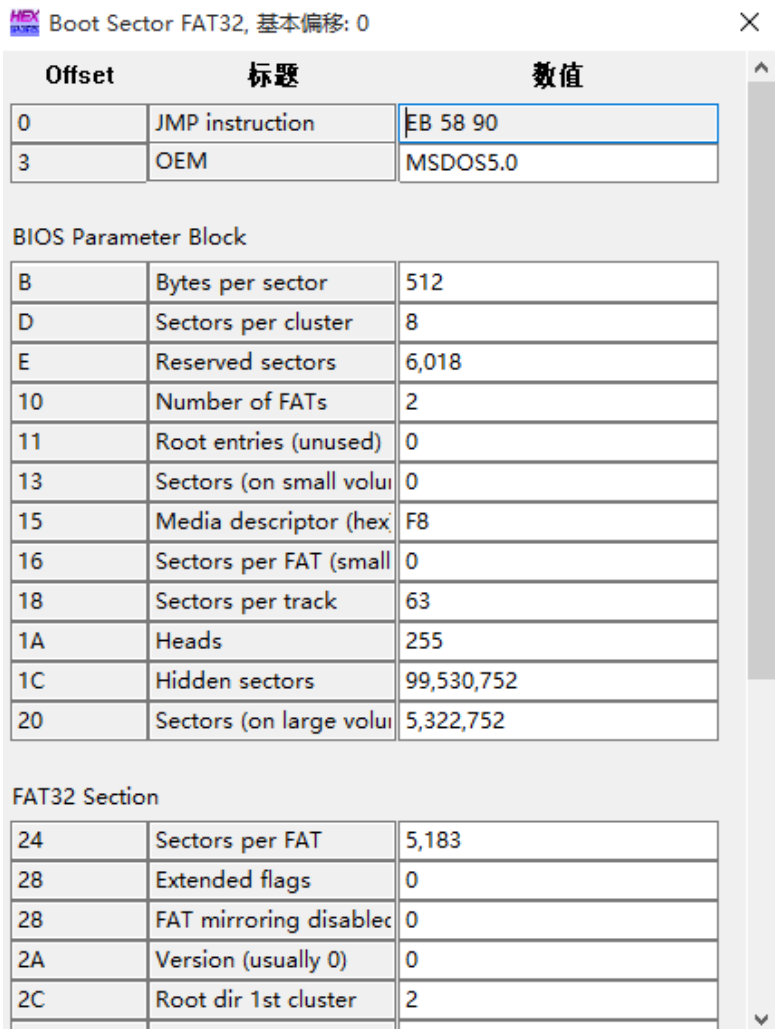


图 3.2 引导扇区模板

在 FAT32 文件系统的磁盘中，FAT1 和 FAT2 是存储簇链表的地方，由于 FAT 区十分重要，故存放两个，作为备份。

在根目录区，建立了每个文件的索引，并记录了每一个文件相关的信息。用 WinHex 模板打开根目录查看文件，如图 3.3 所示

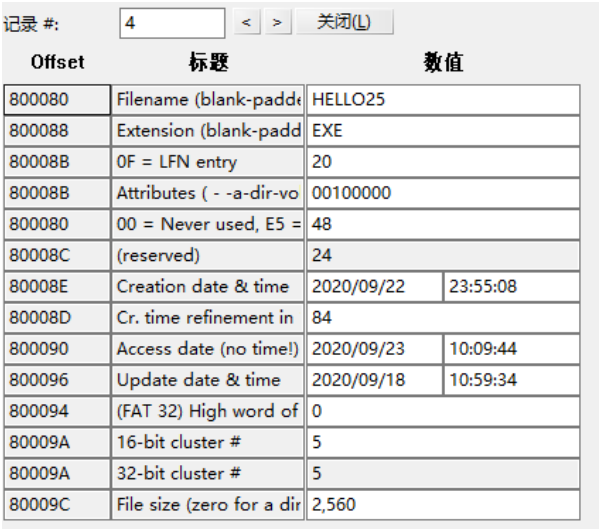


图 3.3 查看根目录

本磁盘中存储了 hello25.exe 文件，通过根目录模板，可以查看它的文件名是 HELLO25，扩展名是 EXE，创建时间是 9/22 23:55，簇高位号是 0，文件大小是 2560Bytes=2.5KB 等等信息。

在根目录下建立文本文件：test-学号后 3 位.txt，其中填充 60KB 左右的文本字符保存。更新磁盘快照，利用 WinHex 查看 txt 文件的簇，在左上角文件管理中右键选择该文件，导航->列出的簇，并取消缩短连续簇链表，如图 3.4 所示。

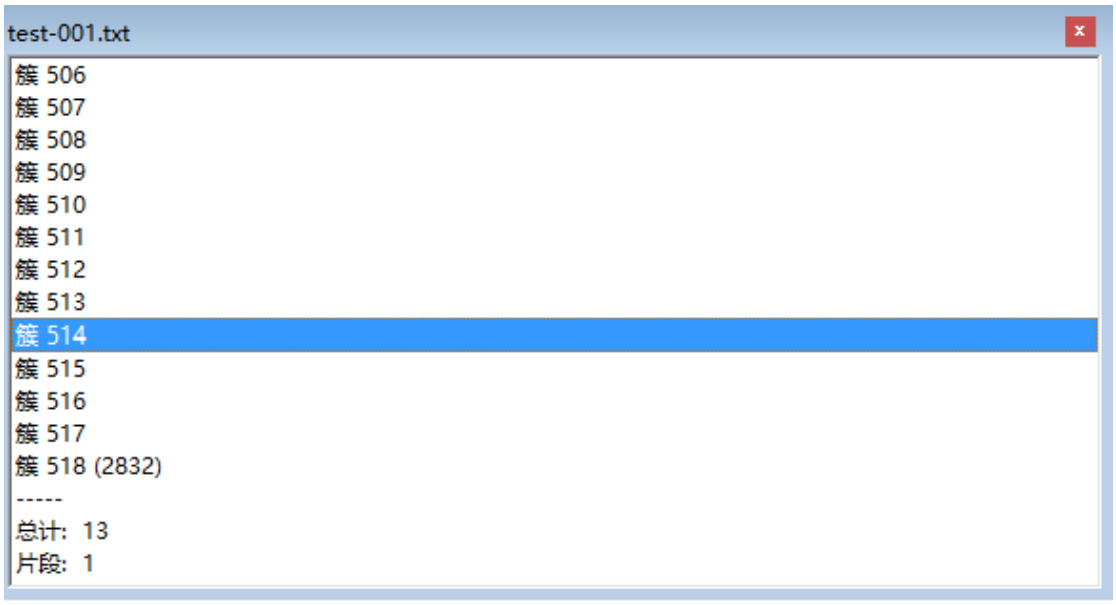


图 3.4 查看根目录

可以看到 txt 文件的首簇号是 506，总共用了 13 个簇来存储。根据 Windows 文件属性可以看到，文件实际大小是 50.7KB，但是占用了 52KB，而图 3.4 显示文件占用了 13 个簇，由前面的信息，本系统一个簇是 8 个扇区，4KB，13 个簇正是 52KB，这就是文件实际的占用大小。50.7KB 文件需占用 13 个簇才可以存下。

下面根据文件占用的簇跳转到文件内容的位置。在 WinHex 中选择导航->跳转到 FAT 记录->填写首簇号 506，跳转到簇链表对应位置。由于是 FAT32 系统，簇链表中每个结点大小是 4 个字节，文件占用 13 个簇，在簇链表中便占据 52 个字节，如图 3.5 所示，高亮部分即 txt 文件的占用的簇链表。

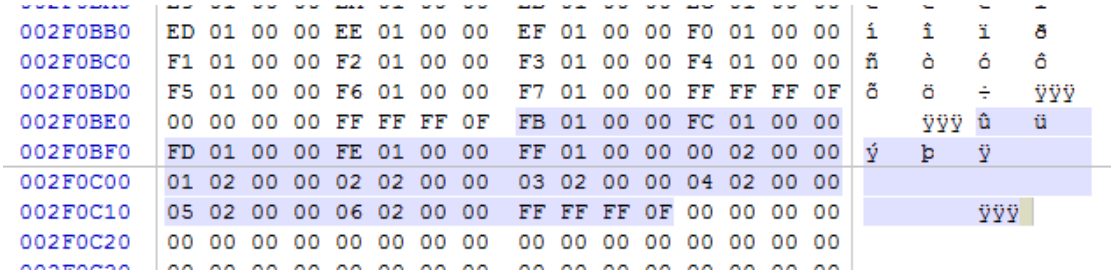


图 3.5 文件簇链表

注意到簇链表首结点值是 0x000001FB=507,而文件的首簇号是 506，这是因为在 FAT32 文件系统中，FAT 表是连续存储的，访问首簇号结点对应的值默认是首簇号+1.

记录下文件的簇链表，在 WinHex 中选择导航->跳至扇区，填写簇 506/507，可以发现跳转

到 txt 文件存储处，如图 3.6 所示。

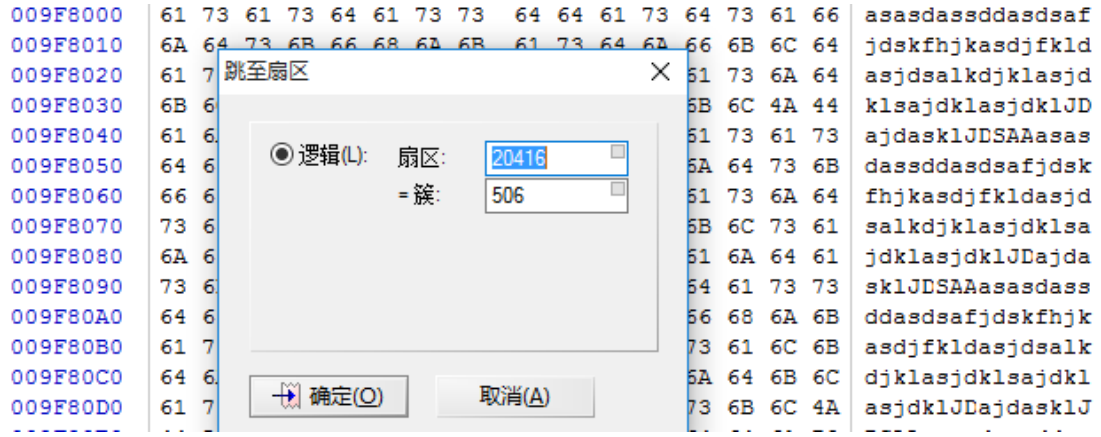


图 3.6 文件簇链表

根据目录项的定义，一个目录项占据 32 个字节，00H-07H 字节是文件的正名，1CH-1FH 是文件的占据的字节大小，打开 WinHex，利用模板跳转到 test-001.txt 的目录项所在的位置，如图 3.7 所示。

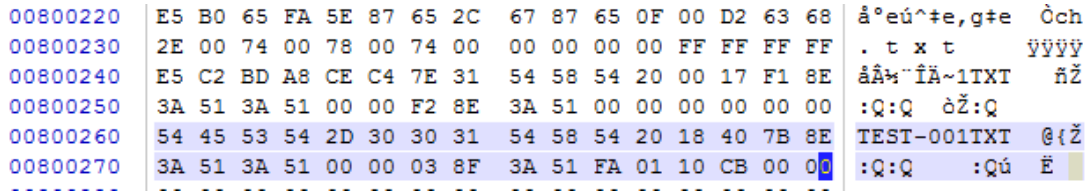


图 3.7 查看目录项

可以看到第 1AH-1BH=26-27 字节是(逆序) 01FA，14H-15H=20-21 字节是 0000，组成的首簇号是 0x000001FA=506，正是该文件之前查看的首簇号，说明观察正确。而 1CH-1FH=28-31 是文件占据的字节大小，0x00 00 CB 10=51984Byte=50.765625KB，与文件系统中显示的大小 50.8KB 保持一致。说明从目录项中正确读取到了文件信息。

1.4.4 手工恢复被删除的文件

首先使用 Del 键进行回收站删除，重新建立磁盘快照。在 WinHex 中查看文件目录，可以看到事实上文件及其对应的数据仍然还存在，如图 4.1 所示。

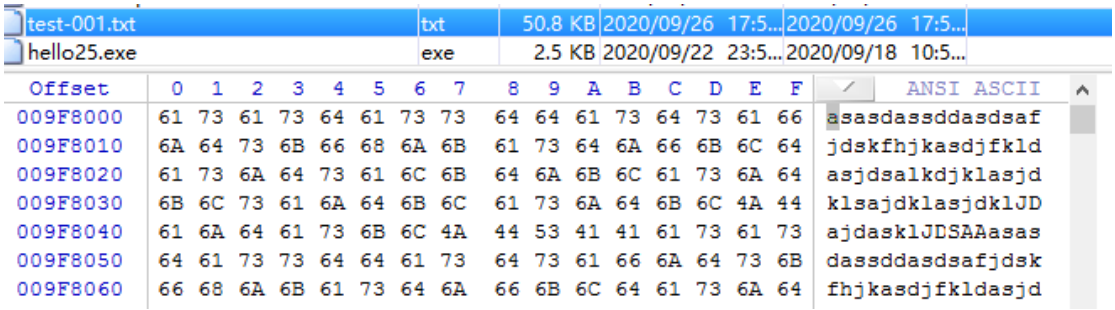
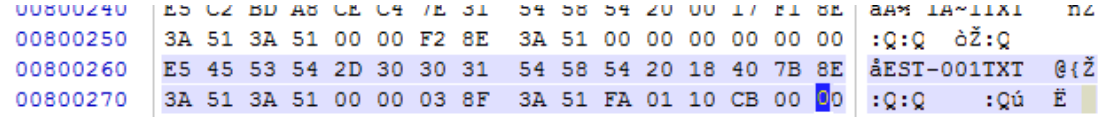


图 4.1 回收站删除后查看文件

跳转至目录表对应位置，如图 4.2 所示，



00800250	3A	51	3A	51	00	00	F2	8E	3A	51	00	00	00	00	00	00	:Q:Q òŽ:Q
00800260	E5	45	53	54	2D	30	30	31	54	58	54	20	18	40	7B	8E	åEST-001TXT @{Ž
00800270	3A	51	3A	51	00	00	03	8F	3A	51	FA	01	10	CB	00	00	:Q:Q :Qú Ë

图 4.2 回收站删除后查看目标表项

将图 4.2 和图 3.7 进行对比，发现目录项首字节在文件删除后被进行了修改成 E5，表示被删除标记。选择 E5 所在字节，右键编辑->填充选块，将 E5 随便修改成任意数据，选择文件->保存扇区，可以发现文件恢复，如图 4.3 所示，但是文件名会根据修改数据改变首字母。至此，完成了回收站文件删除后的文件恢复。

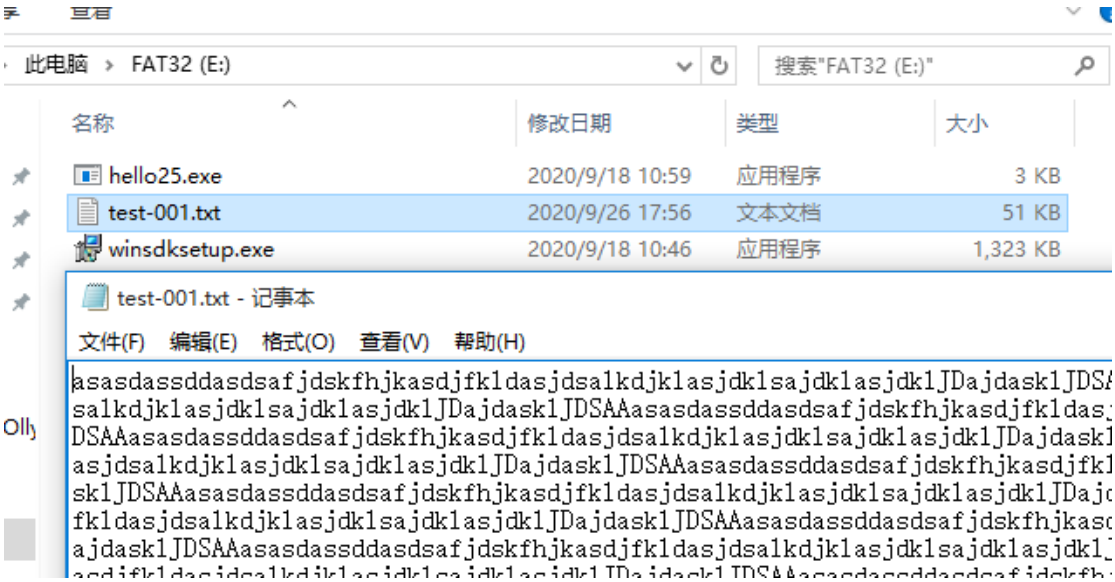
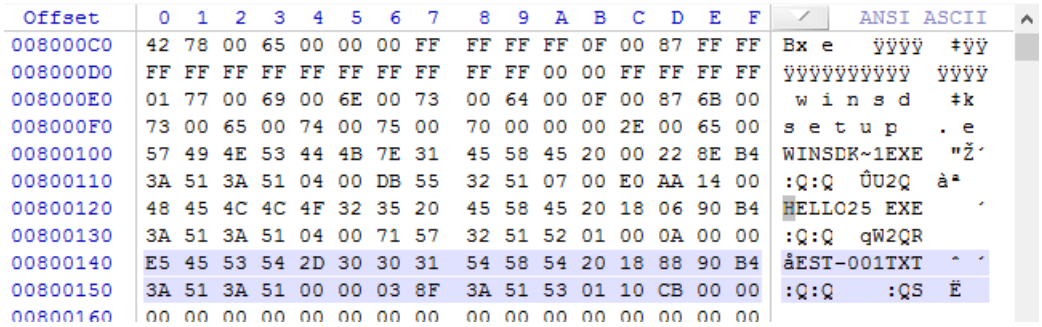


图 4.3 恢复文件

再使用 shift+delete 删除文件 EXP.TXT，转到目录项如图 4.4 所示

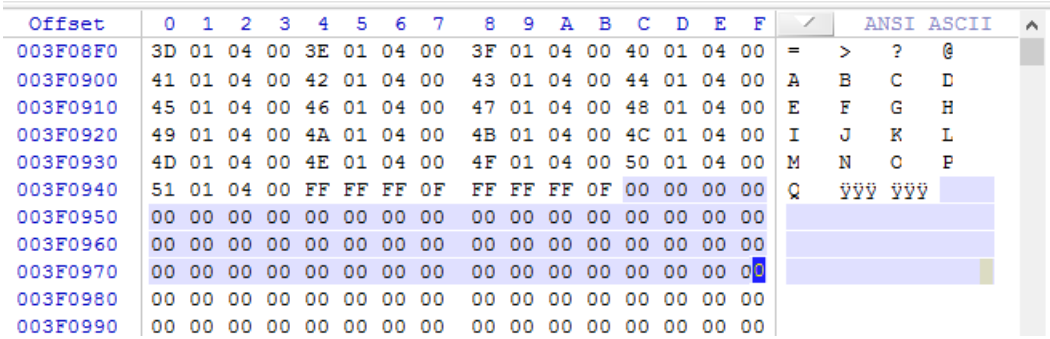


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
008000C0	42	78	00	65	00	00	00	FF	FF	FF	FF	0F	00	87	FF	FF	Bx e ÿÿÿÿ ‡ÿÿ
008000D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿ ÿÿÿÿ
008000E0	01	77	00	69	00	6E	00	73	00	64	00	0F	00	87	6B	00	w i n s d ‡k
008000F0	73	00	65	00	74	00	75	00	70	00	00	00	2E	00	65	00	s e t u p . e
00800100	57	49	4E	53	44	4B	7E	31	45	58	45	20	00	22	8E	B4	WINSDK~1EXE "Ž´
00800110	3A	51	3A	51	04	00	DB	55	32	51	07	00	E0	AA	14	00	:Q:Q ÛU2Q à‘
00800120	48	45	4C	4C	4F	32	35	20	45	58	45	20	18	06	90	B4	HELLO25 EXE ´
00800130	3A	51	3A	51	04	00	71	57	32	51	52	01	00	0A	00	00	:Q:Q qW2QR
00800140	E5	45	53	54	2D	30	30	31	54	58	54	20	18	88	90	B4	åEST-001TXT ˆ ´
00800150	3A	51	3A	51	00	00	03	8F	3A	51	53	01	10	CB	00	00	:Q:Q :QS Ë
00800160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

图 4.4 删除后目录项

可以看到目录项也被修改，首字节被修改为 E5，首簇号高 16 位清 0，但是复原后并不能打开文件，因为簇链表数据已经丢失.

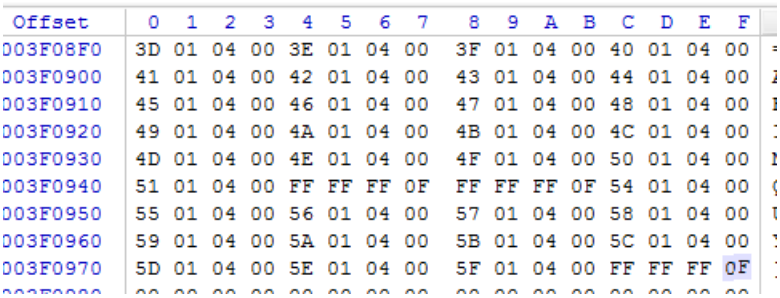
查看簇链表，如图 4.5 所示



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
003F08F0	3D	01	04	00	3E	01	04	00	3F	01	04	00	40	01	04	00	= > ? @
003F0900	41	01	04	00	42	01	04	00	43	01	04	00	44	01	04	00	A B C D
003F0910	45	01	04	00	46	01	04	00	47	01	04	00	48	01	04	00	E F G H
003F0920	49	01	04	00	4A	01	04	00	4B	01	04	00	4C	01	04	00	I J K L
003F0930	4D	01	04	00	4E	01	04	00	4F	01	04	00	50	01	04	00	M N O P
003F0940	51	01	04	00	FF	FF	FF	0F	FF	FF	FF	0F	00	00	00	00	Q ÿÿÿ ÿÿÿ
003F0950	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
003F0960	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
003F0970	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
003F0980	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
003F0990	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

图 4.5

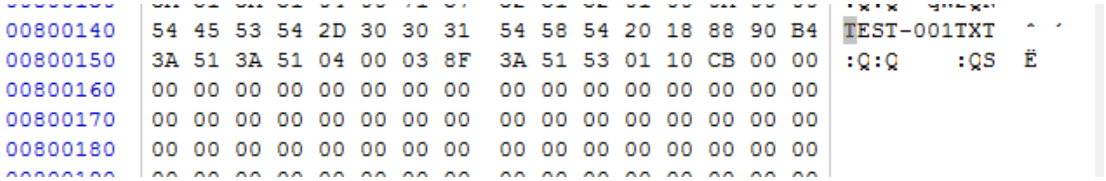
可以发现簇链表对应的位置已经全部清 0，要恢复文件，必须恢复簇链表再修改目录项。因此现在根据其相邻的目录项来确定其首簇号。可以看到前一个目录项的首簇号是 0x00040152，文件的字节大小是 0x A00，占用一个簇，因此恢复文件的首簇号是 0x00040153，因此依次将图 4.5 中的项目进行修复，如图 4.6 所示



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
003F08F0	3D	01	04	00	3E	01	04	00	3F	01	04	00	40	01	04	00
003F0900	41	01	04	00	42	01	04	00	43	01	04	00	44	01	04	00
003F0910	45	01	04	00	46	01	04	00	47	01	04	00	48	01	04	00
003F0920	49	01	04	00	4A	01	04	00	4B	01	04	00	4C	01	04	00
003F0930	4D	01	04	00	4E	01	04	00	4F	01	04	00	50	01	04	00
003F0940	51	01	04	00	FF	FF	FF	0F	FF	FF	FF	0F	54	01	04	00
003F0950	55	01	04	00	56	01	04	00	57	01	04	00	58	01	04	00
003F0960	59	01	04	00	5A	01	04	00	5B	01	04	00	5C	01	04	00
003F0970	5D	01	04	00	5E	01	04	00	5F	01	04	00	FF	FF	FF	0F

图 4.6

将目录项被修改的首字节和首簇高位修复。如图 4.7 所示



00800140	54	45	53	54	2D	30	30	31	54	58	54	20	18	88	90	B4	TEST-001TXT ^ ´
00800150	3A	51	3A	51	04	00	03	8F	3A	51	53	01	10	CB	00	00	:Q:Q :QS Ë
00800160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00800170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00800180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

图 4.7

保存，打开文件夹，发现文件恢复成功。如图 4.8 所示

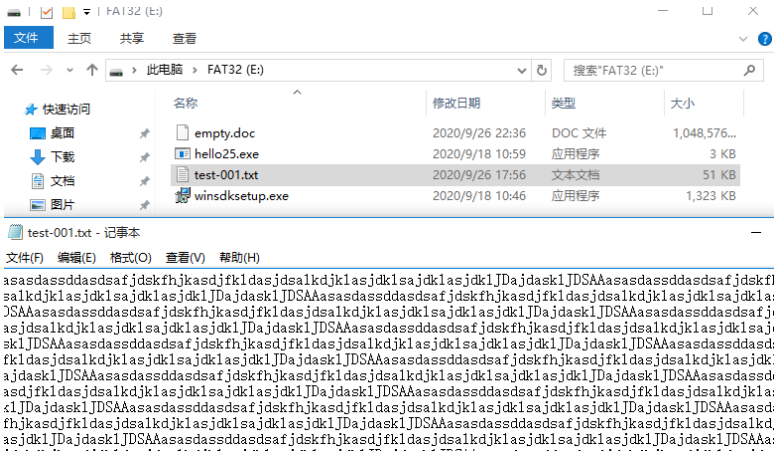


图 4. 8

1.4.5 课后习题思考

在磁盘分区过程中，用户提供了要分区的大小，分区的文件格式，盘符等信息。利用分区大小，系统构建新的逻辑盘，利用指定的文件格式，对分区进行格式化。

低级格式化就是将空白的磁盘划分出柱面和磁道,再将磁道划分为若干个扇区,每个扇区又划分出标识部分 ID、间隔区 GAP 和数据区 DATA 等。可见,低级格式化是高级格式化之前的一件工作, 高级格式化就是清除硬盘上的数据、生成引导区信息、初始化 FAT 表、标注逻辑坏道等。

数据粉碎一般会对文件所在的位置进行重复擦写，粉碎文件之后，利用 winhex 查看对应偏移量数据可以发现数据已经被完全改写。

文件名过长时恢复文件后需要手工查看文件数据区的文件名进行恢复

1、计算机是以二进制来记录数据的，所以单位从 K 到 M 再到 G 是以 1024 来进级的。而人们习惯了十进制，以 1000 来进级。这样造成了显示容量和实际一样。当然，也有些不良厂家以 1000 为进级计算标注容量，使实际容量缩水。
2、硬盘在分区和格式化时会占用一定的存储空间来保存分区等一些信息，所以实际可用容量就会小于分区的容量。
3、因为硬盘的分区一般都是按扇区为单位分的（每个扇区的大小一般从几 KB 到几百 KB，不同硬盘和格式不完全相同），所以人为输入的分区容量不一定是扇区的整数倍，系统会自动取近似的容量来划分分区。

1.5 实验体会和拓展思考

通过了解磁盘格式，使我对系统对于磁盘的管理有了全新的认识，之前我曾经以为磁盘分区与分类文件夹差不多，但是通过实验让我深刻理解了分区的机理和作用。对于簇链表的学习也使我理解了文件删除和恢复的机理，明白了恶意病毒破坏数据的手段。对于 FAT32,NTFS 等文件系统的学习使我了解到了不同文件系统各自的特点，以及在今后如何正确为自己的电脑设置合适的文件系统。
通过学习磁盘格式与数据恢复，使我对如何攻击与防御恶意软件破坏数据指明了一条方向，也促使我今后提升对于文件保护的意识。而数据恢复实验可以帮助我在紧急情况下恢复自己的重要数据