

基于 FAT32 文件系统的数据隐写实验

原创

渔网探索者 于 2020-06-12 09:45:26 发布 852 收藏 15

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/z1576393803/article/details/106708216>

版权

1 实验目的

本实验的目的是深入理解 FAT32 文件系统管理文件的方式，验证通过修改文件系统的保留扇区、FAT 分区表等关键结构，能够实现将数据隐写在分区，并且不会被新写入文件覆盖、不易被用户发现等功能。

2 实验内容及环境

1. 实验内容

本实验要求通过修改采用 FAT32 文件系统的分区结构，实现将数据隐写在分区中的空闲扇区，并且该隐写数据不会被新拷入文件覆盖。

2. 实验环境

(1) Win 10操作系统，以及实验

用的 WinHex 工具；

(2) Kingston 8G 优盘：该优盘采用 FAT32 文件系统；

3. 实验工具

WinHex 中文版：WinHex 是一款以通用的 16 进制编辑器为核心，专门

用来对付计算机取证、数据恢复、低级数据处理、以及 IT 安全性、各种日常紧急情况的高级工具；

用来检查和修复各种文件、恢复删除文件、硬盘损坏、数码相机卡损坏造成的数据丢失等。

3 实验步骤

本实验的思路是从优盘 FAT 表项中找一个未分配的簇，将其标志为“占用”或“坏簇”，然后在该簇内写入数据。为了保证数据的隐藏性，不修改优盘的目录结构。

具体修改位置包括：

(1) 引导扇区中的“空闲簇总数”和“下一个可用簇”；

(2) FAT1 表和 FAT2 表中找到“标识为 0”（4 字节）的簇，将其改为标志为“占用”或“坏簇”，并计算其簇号；

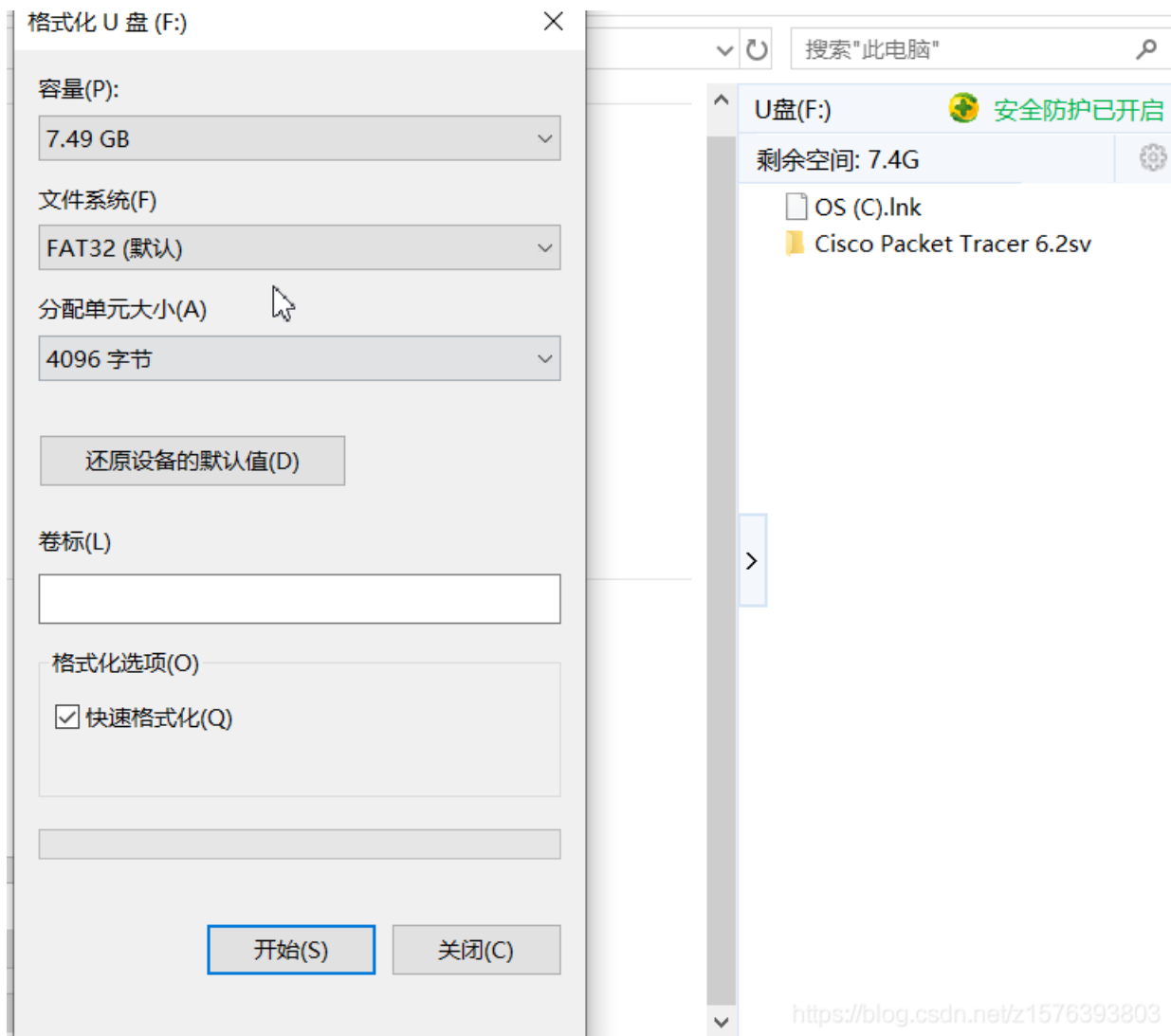
(3) 根据其簇号，找到其所在数据区部分，并对该簇内若干字节进行数据的填写；

1. 初始化优盘

将优盘插入主机的 USB 接口，待主机识别后，双击“我的电脑”，找到优盘所在分

区，右键点击，选择“格式化”如图所示。

格式化磁盘



2. 利用 WinHex 打开优盘

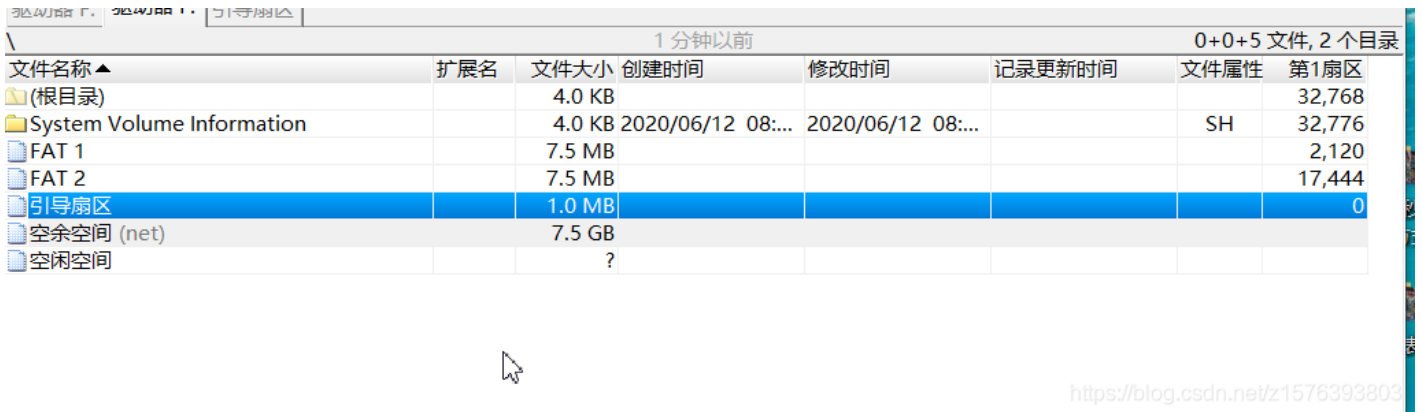
双击 WinHex.exe，进入操作界面。点击菜单→“工具”→“打开磁盘”，选择优盘所在分区

如图所示。Winhex 选择目标磁盘，打开优盘

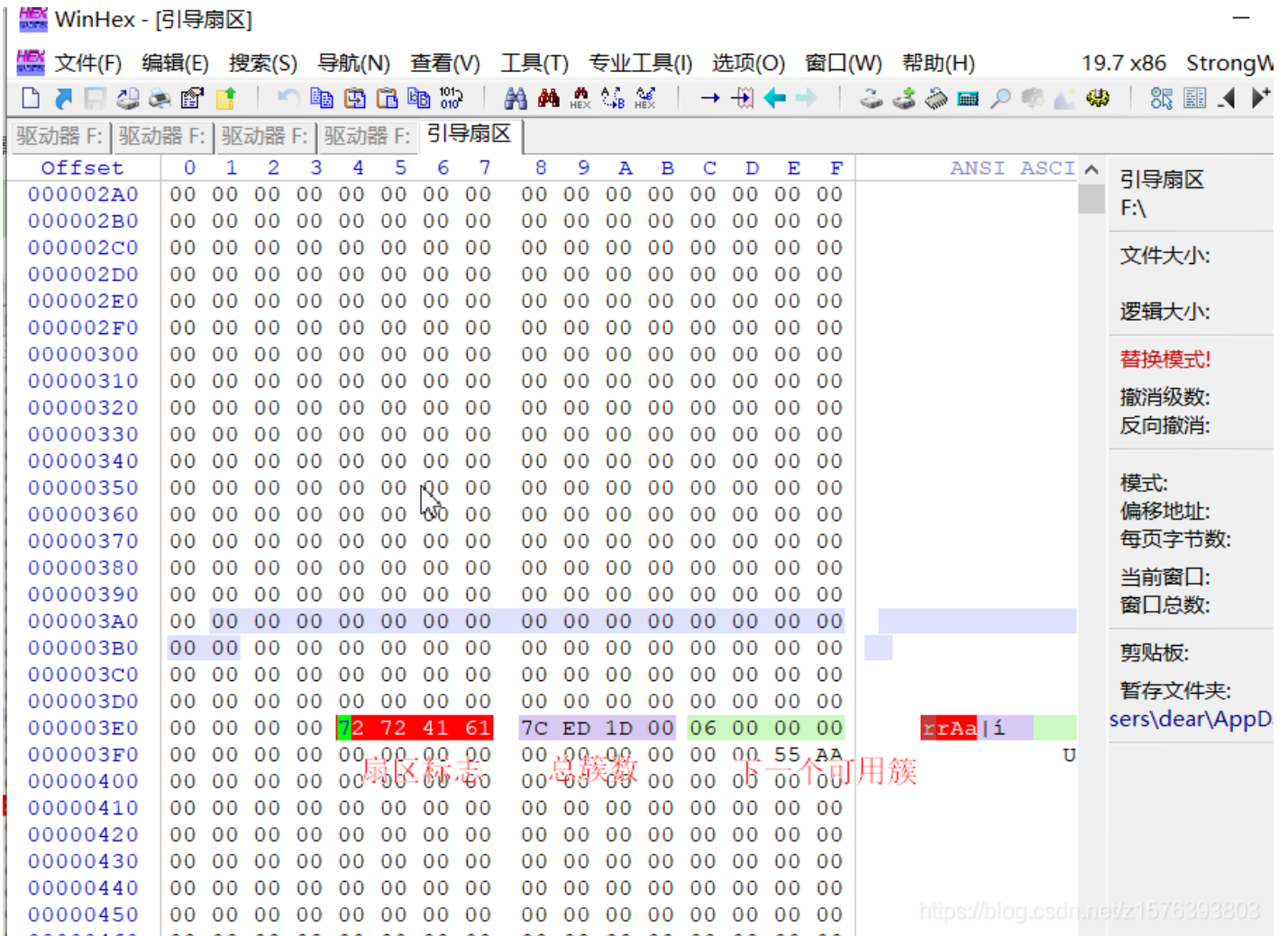


3.修改 FSINFO 保留扇区

1.为了找到可以写入数据的空闲空间，打开右击打开引导扇区



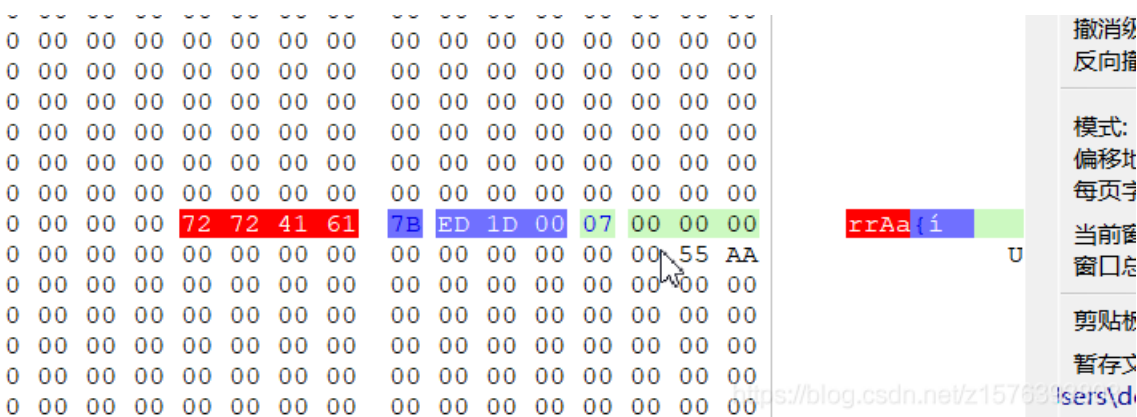
在引导扇区往下面找到 4 个字节“72 72 41 61”（扇区标志）来判断定位的 FSINFO 扇区是否正确。



根据 FAT32 文件系统结构，该扇区后连续的 4 个字节即为当前分区所有的可用簇总数，得值为“7C ED 1D 00”，由于需要将数据写入一个空闲簇，因此将其值减 1，得到“7B ED 1D 00”；同理，

总簇数值后连续 4 个字节为当前分区内“下一个可用簇”的字段，值为“06 00 00 00”。注意该簇 0x0006 (从后往前数)即为我们要写入数据的空闲簇。

如图 修改 FSINFO 扇区



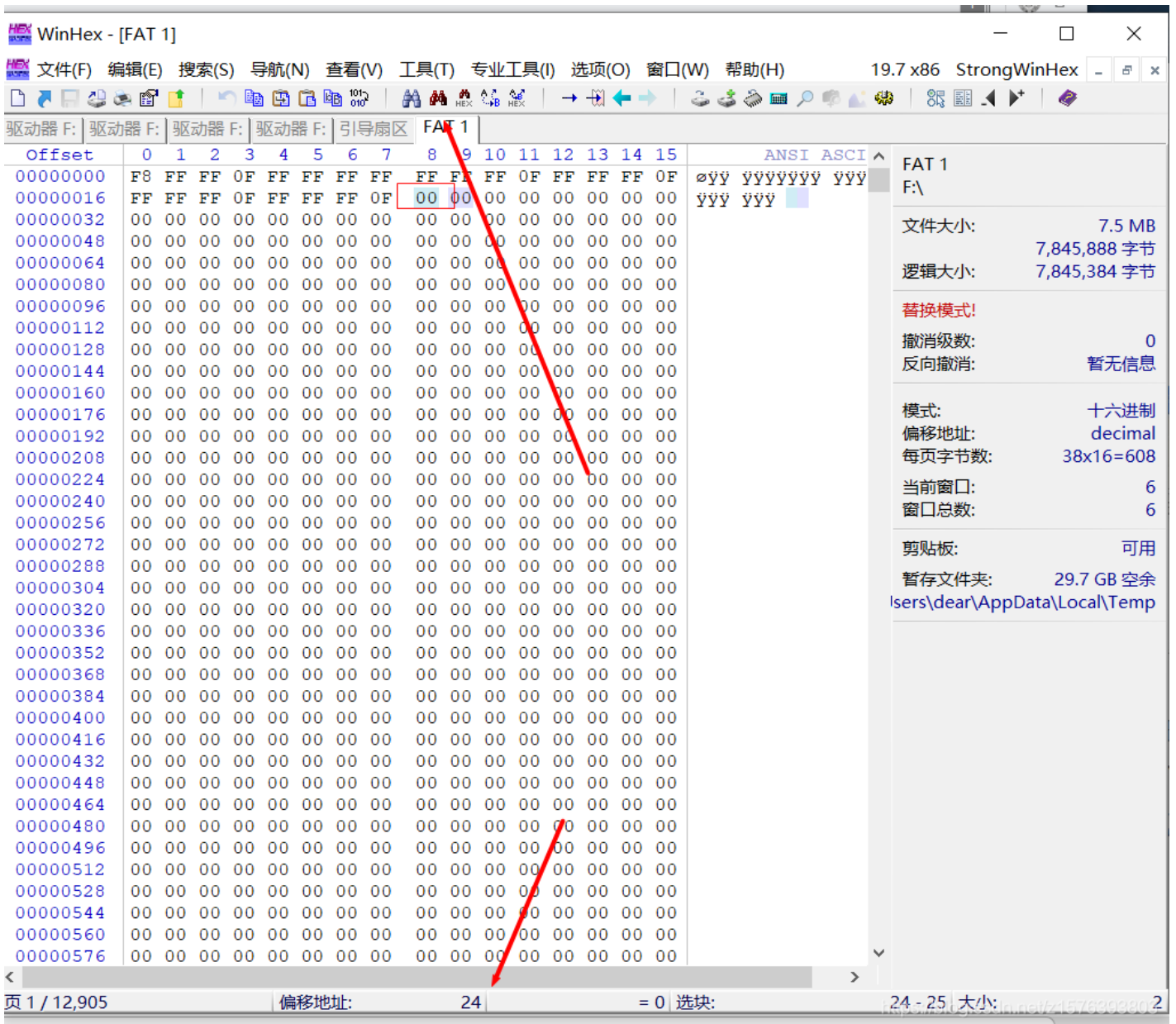
通过 FAT 表找到连续 4 个全零字节所在簇，我们这里找到的是 0x0006，将“下一个可用簇”的值 0x0006 改为 0x0007，将“空闲簇总数”的 0x001DED7C 改为 0x001DED7B。

4. 修改 FAT 表

接下来需要修改 FAT 表中的空闲簇，在 Winhex 文件显示窗口中点击“FAT1”来到

FAT1 表的初始位置。

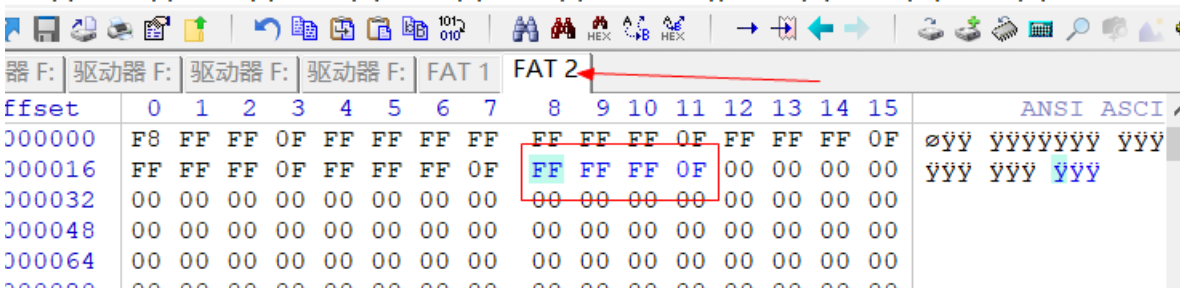
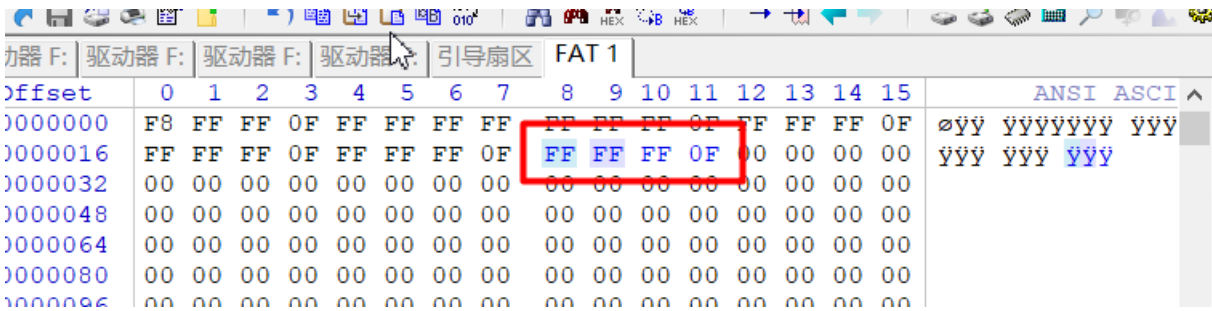
如图所示。



在 FAT1 表中，找到 0x0006 簇的状态字符，即相对起始处偏移 $0x0006 \times 4 = 24$ (十进制) 的指

示的 4 个字节，将其值“00 00 00 00”改为“FF FF FF 0F”或者“F7 FF FF FF”，前者

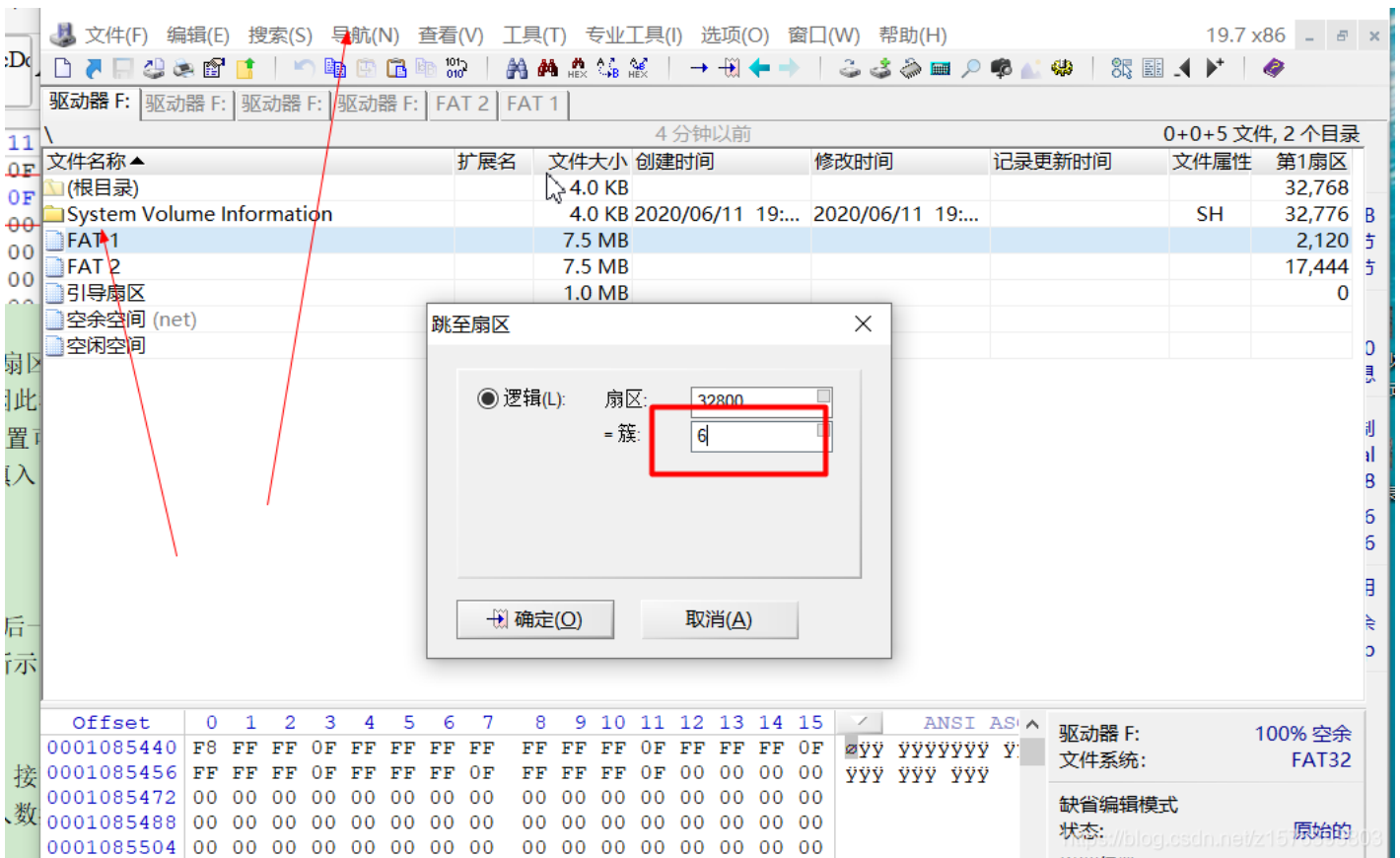
表示该簇已被占用，后者表示该簇已坏，将该簇状态标志为坏簇或占用簇主要为了让系统认为该簇已使用而不会被新写入的数据覆盖。FAT2 是 FAT1 的备份，为了保险起见，FAT2 也采用与 FAT1 同样的处理方式。



5. 填写数据

找到数据区“0x0006”簇开始的位置，将连续的 8 个扇区内容填入需要隐藏的数据，由于本实验只是为了验证写入数据不会被新文件覆盖，因此我们手工将每个扇区的开始部分改为连续的 0x66,即字符“f”。数据区的 0x0006 簇位置可利用 WinHex 菜单中提供的工具查找，选择菜单“à导航à跳至扇区”，在“簇”处填入 0x0006 转换为十进制的数字 6 即可，如图所示。

定位数据区指定簇



在本簇内开头和结尾写入0x66，右击鼠标→编辑→填充磁盘扇区→填充十六进制66 来验证此簇内不会被写入的数据覆盖。在结尾写入数据以区别与下一个可用簇的界线。

填充选择

填充十六进制数值(v)

66

随机字符(r)

范围(N): 0 至 255 (0..255)

模拟加密数据

加密 伪随机序列数-慢(r)

覆盖次数(P):

次数 #1

添加(d)

删除(l)

< 0x00

< 0xD

确定(O)

取消(A)

帮助(H)

在本簇开头写入66

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
001003FD0	50	FF	15	E8	71	40	00	89	35	88	0B	42	00	5E	C3	39	Pÿ	èq@ %5^ B ^Ã
001003FE0	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	iiiiiiiiiiiiiiii	
001003FF0	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004000	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004010	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004020	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004030	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004040	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004050	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004060	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004070	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004080	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	ffffffffffffffff	
001004090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	00	FF	FF	FF	FF	yyyyyyyyyy yyy	
0010040A0	B3	CC	D0	F2	20	20	20	20	20	20	20	10	00	42	FD	A8	* ÌÐò	Bý
0010040B0	CB	50	CB	50	00	00	0A	9D	CB	50	0E	01	00	00	00	00	ÈPÈP	ÈP
0010040C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		

在本簇结尾处写入66

新建文件夹 (2) 4.0 KB 2020/06/12 07:... 2020/06/11 19:... 32,808

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
16797536	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797552	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797568	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797584	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797600	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797616	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797632	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797648	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16797664	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16797680	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16797696	2E	20	20	20	20	20	20	20	20	20	10	00	AE	D1	3D	.	.	6
16797712	CC	50	CC	50	00	00	D2	3D	CC	50	07	00	00	00	00	00	i	P
16797728	2E	2E	20	20	20	20	20	20	20	20	20	10	00	AE	D1	3D	.	.
16797744	CC	50	CC	50	00	00	D2	3D	CC	50	00	00	00	00	00	00	i	P
16797760	41	42	4F	55	54	44	4C	47	43	50	50	20	18	B1	D1	3D	A	B
16797776	CC	50	CC	50	00	00	E4	A8	25	38	08	00	7A	02	00	00	i	P
16797792	41	42	4F	55	54	44	4C	47	48	20	20	20	18	B5	D1	3D	A	B
16797808	CC	50	CC	50	00	00	EA	95	9E	37	09	00	C1	04	00	00	i	P
16797824	43	4C	45	41	4E	55	50	20	42	41	54	20	18	BA	D1	3D	C	L
16797840	CC	50	CC	50	00	00	D4	4D	1E	27	02	00	12	01	00	00	i	P

驱动器 F: 98% 空余
文件系统: FAT32
缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息
可见磁盘空间中的分配表:
簇号: 6
空闲空间
磁盘快照创建于 9 分钟前
逻辑扇区号: 32,807
物理扇区号: 37,927
已用空间: 141 MB
147,529,728 字节

在下一个可用簇也写入数据，用来验证，这个簇在写入数据之后，写入的内容被覆盖

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI	ASCII
16794080	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794112	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
16794128	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794144	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794160	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794176	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794192	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794208	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794224	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794240	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794256	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794272	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794288	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794304	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794320	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794336	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794352	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794368	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794384	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f
16794400	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	66	f	f

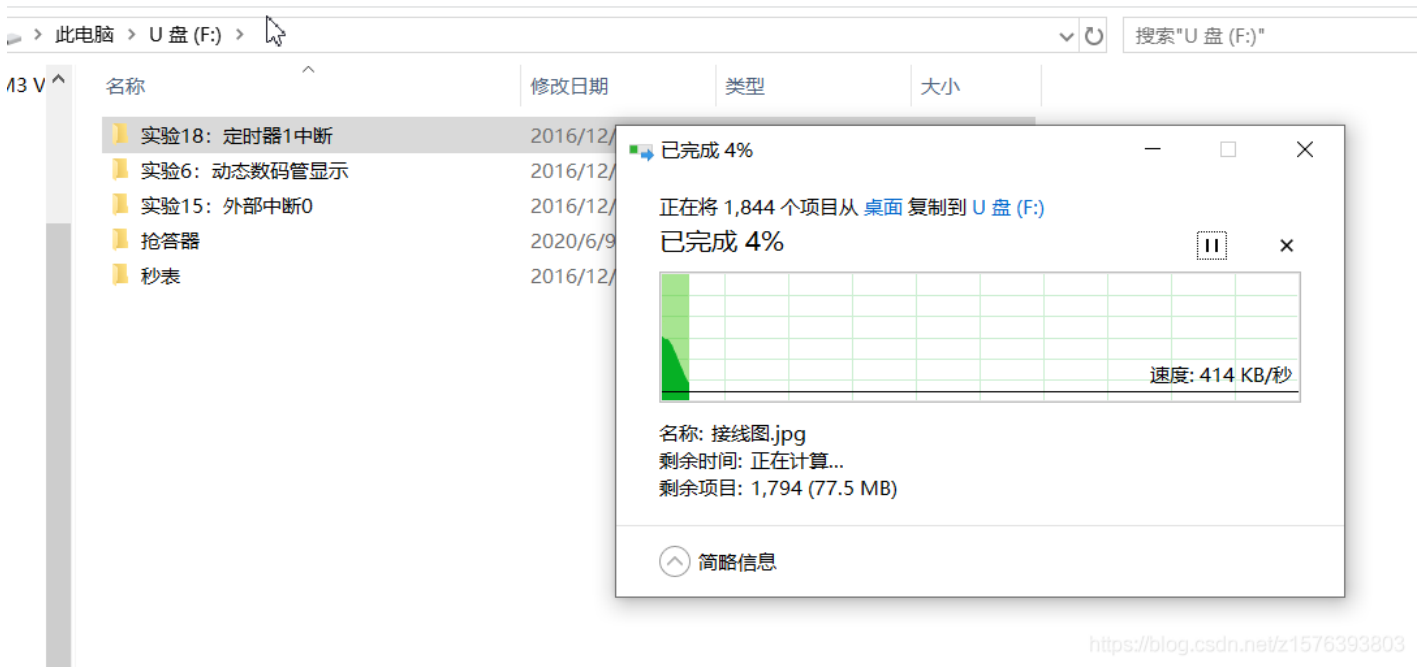
扇区 32,801 / 15,723,520 偏移地址: 16,794,320 = 102 选块: https://blog.csdn.net/z15763 无大小

6. 保存与验证

点击菜单“文件→保存”将写入的数据保存。接下来需要往优盘中写入新的

文件，以用来验证之前写入的数据不会被覆盖。在拷入数据前，注意先将优盘从系统中正常退出，重新插拔。

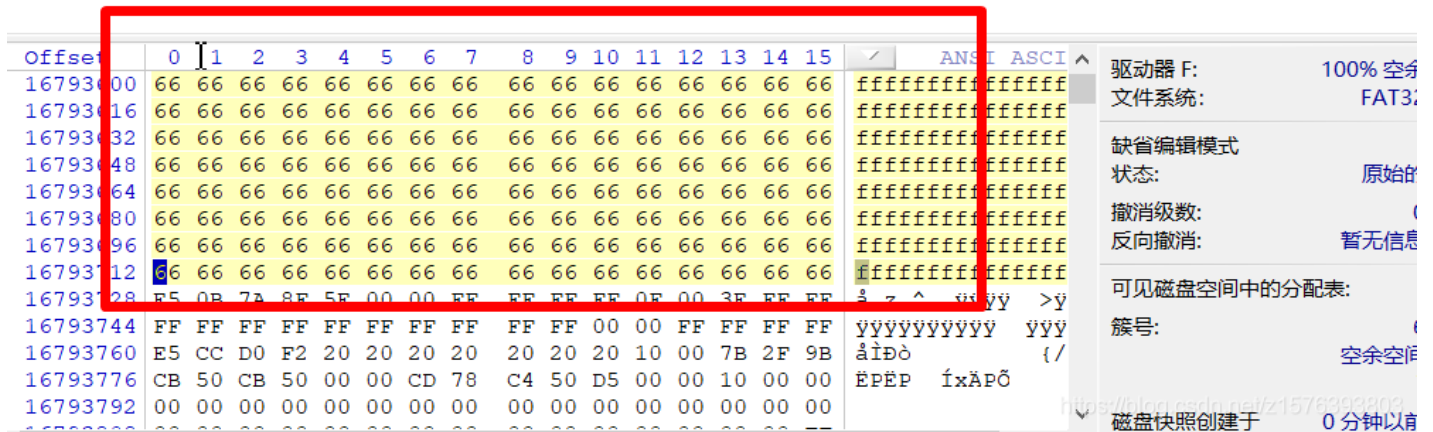
向U 盘随便复制进一些数据



再次用 WinHex 将优盘所在分区打开。定位到数据区的 0x006 簇的最后一个扇区，检查一下看看先前在 0x0066 簇写入的数据是否被覆盖，如图所示。

检查 0x006 簇数据

没有被删除



由图可以看出，先前写入的0x006 处的隐藏数据没有被新文件覆盖，从 0x007 处内容开始被删除

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
30	01	FF	C4	00	1F	00	00	01	05	01	01	01	01	01	01	00	ÿ	Ä
30	00	00	00	00	00	00	00	01	02	03	04	05	06	07	08	09		
30	0A	0B	FF	C4	00	B5	10	00	02	01	03	03	02	04	05	05	ÿ	Ä µ
30	05	04	04	00	00	01	7D	01	02	03	00	04	11	05	12	21		}
30	31	41	06	13	51	61	07	22	71	14	32	81	91	A1	08	23	1	A Qa "q 2 `;
30	42	B1	C1	15	52	D1	F0	24	33	62	72	82	09	0A	16	17	B±	Ä Ñø\$3br,
30	18	19	1A	25	26	27	28	29	2A	34	35	36	37	38	39	3A		%&'()*456789
30	43	44	45	46	47	48	49	4A	53	54	55	56	57	58	59	5A	C	DEFGHIJSTUVWXY
30	63	64	65	66	67	68	69	6A	73	74	75	76	77	78	79	7A	c	defghijstuvwxyz
30	83	84	85	86	87	88	89	8A	92	93	94	95	96	97	98	99	f	,;:~!@#\$%^&'()*+,-./:~
30	9A	A2	A3	A4	A5	A6	A7	A8	A9	AA	B2	B3	B4	B5	B6	B7	š	ç&µ¥;§" '@a"µ¶
30	B8	B9	BA	C2	C3	C4	C5	C6	C7	C8	C9	CA	D2	D3	D4	D5	,	°ÅÄÅÅÆÇÈÉÊËËÓÔ
30	D6	D7	D8	D9	DA	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	F1	Ö	×ØÙÚáääääæçèéê
30	F2	F3	F4	F5	F6	F7	F8	F9	FA	FF	C4	00	1F	01	00	03	ò	óôõ÷øùúÿ
30	01	01	01	01	01	01	01	01	01	00	00	00	00	00	00	01		
30	02	03	04	05	06	07	08	09	0A	0B	FF	C4	00	B5	11	00	ÿ	Ä µ

驱动器 F: 99% 空余
文件系统: FAT32

缺省编辑模式
状态: 原始的
撤消级数: 0
反向撤消: 暂无信息

可见磁盘空间中的分配表:
簇号: 7
接线图.jpg
实验18: 定时器1中断

磁盘快照创建于 0 分钟以前
逻辑扇区号: 32,808
物理扇区号: 37,928

4 实验结论

通过修改可用簇的值可以使写入的数据跳过本簇，不会覆盖里面的数据内容，在下一个簇开始写入数据